AM13E230x Microcontrollers

Technical Reference Manual

Literature Number: SPRUJF2
MARCH 2026

About This Manual

Notational Conventions

This document uses the following conventions.

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

Glossary

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

Related Documentation

Support Resources

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

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

Trademarks

TI E2E™ and EnergyTrace™ are trademarks of Texas Instruments.
All trademarks are the property of their respective owners.

ADVANCE INFORMATION

Table of Contents

ADVANCE INFORMATION

ADVANCE INFORMATION

ADVANCE INFORMATION

ADVANCE INFORMATION

ADVANCE INFORMATION

This page intentionally left blank.

Chapter 1

Introduction

This chapter introduces the features, subsystems, and architecture of the AM13E230x microcontroller platform.

Note

The AM13E230x microcontroller platform is hereinafter commonly referred to as *AM13E230x, platform, device,* or *chip*.

ADVANCE INFORMATION

1.1 Overview

The AM13E230x series of microcontrollers (MCUs) are cost-optimized 32-bit devices built for complex real-time control applications such as motor control. AM13E230x MCUs include a high-performance Arm® Cortex®-M33 core operating at up to 200MHz with a proprietary math accelerator for trigonometric operations (TMU) through a Custom Datapath Extension (CDE) and a TinyEngine™ Neural-network Processing Unit (NPU) Artificial Intelligence (AI) Engine to enable simultaneous motor control and AI model execution. Its rich analog integration offers industry-leading speed, precision, and reduced system cost. The digital peripherals offered on AM13E230x MCUs are specifically designed for motor control applications, and include tailored communications interfaces for appliance and industrial motor drive end equipment.

The AM13E230x MCUs provide up to 512KB of embedded flash program memory (2 banks of 256KB) with built-in error correction code (ECC) and up to 128KB SRAM (4 banks of 32KB) with hardware parity.

AM13E230x MCUs are enabled with robust, high-performance analog peripherals. Three 12-bit ADCs, four high-speed comparators (CMPSS_LITE) with built-in 10-bit reference DACs, and three PGAs (programmable gain amplifiers) provide real-time signal-chain performance. These MCUs also offer intelligent digital peripherals such as two advanced control timers, TIMG4 (16-bit) and TIMG12 (32-bit), one Windowed Watchdog Timer (WWDT), and a 12-channel Direct Memory Access (DMA) controller.

Complex real-time control applications are enabled with the MCUs' five MCPWM modules with 6 channels per module, two Enhanced Capture (ECAP) modules, and three Enhanced Quadrature Encoder Pulse (EQEP) modules. The integrated crossbar (XBAR) infrastructure enables flexible configuration and routing of external signals to internal ports and internal signals to external pins.

Data integrity and encryption features (AES, secure boot) provide security across the AM13E230x domains. The Keystore module provides secure management of the AES keys. A Cyclic Redundancy Checker (CRC) module provides internal diagnostics to the AM13E230x MCU.

Enhanced communication interfaces are supported through the single MCAN (CAN-FD) interface and two advanced SPG (Serial Peripheral Group) blocks. Each SPG block offers three UNICOMM (Unified Communications Module) blocks, for a total of 6 UNICOMM modules. Within an SPG block, two of the UNICOMM modules can be configured as UART, I2C, or SPI, and one module can be configured as UART/LIN or I2C/SMBUS. The high-speed External Peripheral Interface (EPI) connects AM13E230x to SDRAM or asynchronous RAM devices, and supports FPGA connectivity.

The AM13E230x series of microcontrollers (MCUs) offers a range of options in various memory, peripheral, and package configurations, enabling customers to find the most fitting solution that meets their specific project needs. The AM13E230x MCU platform combines the Arm® Cortex®-M33 platform with a holistic low-power system architecture, allowing system designers to increase performance while reducing energy consumption.

1.2 AM13E230x Architecture Overview

The AM13E230x Microcontroller is based on the Arm® Cortex-M33 core running at 200MHz. The device is organized into various functional blocks, each associated with specific clock and power domains. The functional blocks are interfaced through a network of interconnects and buses that connect across the clock and power domains. This section will detail a high-level overview of the clock and power domains, and the interconnects. More details about these device aspects can be found in the clock, power, and CPU sections of this document.

1.2.1 Bus, Power, Clock Organization

There are two main power domains on AM13E230x devices:

- **PD1** (Power Domain 1): Includes the CPU Subsystem, DMA, SRAM, and PD1 Peripheral Buses
- **PD0** (Power Domain 0): Includes PD0 peripherals and PD0 Bus

PD1 and PD0 are supplied by the 3.3V VDD input to the MCU, and the on-board STOP (1.0V) and Main (1.35V) LDOs, which supply power to modules with lower voltage requirements. The PD1 domain supports higher clock speeds for performance, and is disabled in certain operating modes to minimize power consumption. The PD0

ADVANCE INFORMATION

domain supports low-power performance and is always enabled in operating modes in which the core regulator is operating.

The device clocks are generated from several internal and external oscillators. External oscillators include a crystal oscillator input (25MHz) or a high-frequency clock input (4-48MHz). Internal oscillators include the System Oscillator (SYSOSC, up to 32MHz), and Low-Frequency Oscillator (LFOSC, 32KHz). PLL clocks are generated from the HPLL clock module, programmable up to 200MHz.

The M33 CPU is the fastest operating module in the device, and is clocked at the maximum frequency, 200MHz, referred to as MCLK. Peripheral blocks that do not require MCLK operate at speeds MCLK/2 (maximum frequency divided by 2), or MCLK/4 (maximum frequency divided by 4). Peripheral buses that cross clock domains are connected through SYNC bridges, which ensure that all clock domains are synchronous to the MCLK domain.

There are four main data buses on AM13E230x devices:

- S-AHB (System) bus matrix, interfaces the CPU to all peripheral buses
- C-AHB (CPU) bus matrix, interfaces the CPU to device memory systems (ROM, Flash memory)
- NPU bus, interfaces the Neural-Network Processing Unit to the device SRAM
- DMA bus, interfaces all peripheral buses to the device SRAM and Flash memory

The data buses allow communication between the Peripheral Interconnects on the device. The peripheral interconnects operate in different power and clock domains:

- Fast Peripheral Interconnect (PD1, MCLK)
- Peripheral Interconnect (PD1, MCLK/2)
- CPU Interconnect (PD1, MCLK/4)
- PD0 Interconnect (PD0, MCLK/4)

1.2.2 Device Block Diagram

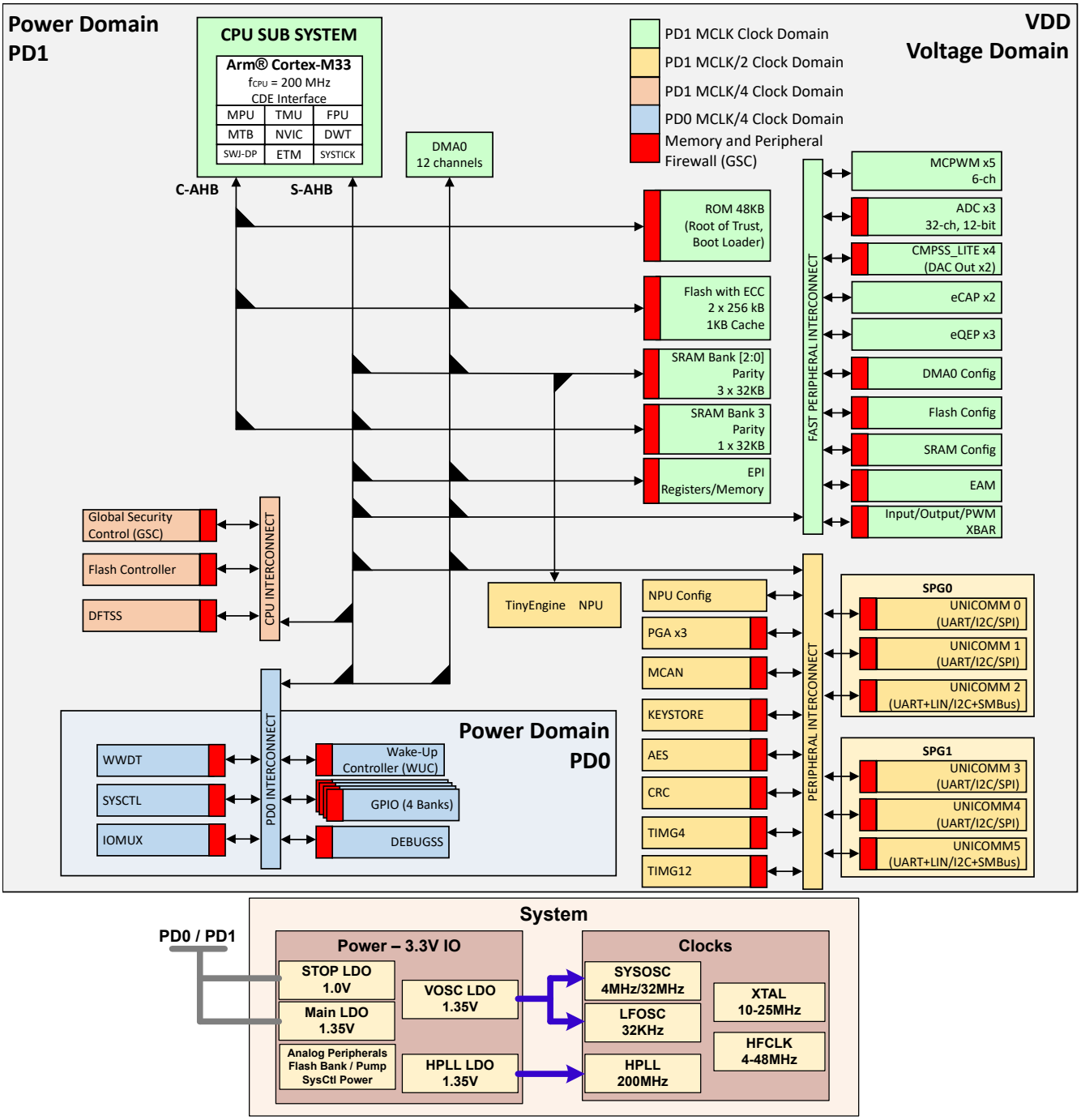


Figure 1-1. Device Block Diagram

ADVANCE INFORMATION

1.2.3 Module Allocation and Instances

Table 1-1. AM13E230x Module Allocation and Instances

Module Abbreviation	Module Full Name	Size / Device Instances	Power Domain	Notes
CPU & System				
M33 CPU	Arm® Cortex-M33 CPU	1	PD1	Includes Custom Datapath Extension (CDE) interface
TMU	Trigonometric Math Unit	1	PD1	
CPUSS Components	Central Processing Unit Subsystem Components		PD1	Includes: • Memory Protection Unit (MPU) • Floating-Point Unit (FPU) • Nested Vectored Interrupt Controller (NVIC) • Data Watchpoint and Trace Unit (DWT) • JTAG/Serial Wire Debug ports (SWJ-DP) • Micro Trace Buffer (MTB) • Embedded Trace Macrocell (ETM) • SysTick - 24-bit down counter
TinyEngine™ NPU	Neural-Network Processing Unit	1	PD1	Machine learning accelerator
DMA	Direct Memory Access Controller	1	PD1	12 channels
DEBUGSS	Debug Subsystem	1	PD0	
EAM	Error Aggregator Module	1	PD1	
Memories				
ROM	Read-Only Memory	48KB	PD1	
SRAM	Static Random-Access Memory	128KB	PD1	4 SRAM blocks, each with 32KB. All blocks have hardware parity support.
Flash		512KB	PD1	Up to 2 blocks of 256KB each with Error Correction Code (ECC)
General Connectivity & Communcation Peripherals				
GPIO	General Purpose Input/Output	Up to 107	PD0	
SPG0	Serial Peripheral Group 0	3 UNICOMM	PD1	UC0: UART / SPI / I2C
				UC1: UART / SPI / I2C
				UC2: UART + LIN / I2C + SMBus
SPG1	Serial Peripheral Group 1	3 UNICOMM	PD1	UC3: UART / SPI / I2C
				UC4: UART / SPI / I2C
				UC5: UART + LIN / I2C + SMBus
MCAN	Modular Controller Area Network	1	PD1	CAN-FD peripheral with 1KB of RAM
Timer Modules				
TIMG4		1	PD1	16-bit, general-purpose counter
TIMG12		1	PD1	32-bit, general-purpose counter
WWDT	Windowed WatchDog Timer	1	PD0	25-bit windowed watchdog timer driven from LFOSC
Analog Peripherals				
ADC	Analog to Digital Converter	3	PD1	6.9MSPS at 200MHz CPU speed

ADVANCE INFORMATION

Table 1-1. AM13E230x Module Allocation and Instances (continued)

Module Abbreviation	Module Full Name	Size / Device Instances	Power Domain	Notes
		32 input channels per ADC		
PGA	Programmable Gain Amplifier	3	PD1	4:1 input mux for effective 12-ch
CMPSS_LITE	Comparator Subsystem	4	PD1	2 of 4 CMPSS_LITE have DAC Output
VREF	Voltage Reference	1	PD1	ADC voltage reference. Internal reference and brought out to external pin.
Digital Control Peripherals				
MCPWM	Multi-Channel Pulse Width Modulattion Module	5	PD1	6 channels per module
ECAP	Enhanced Capture Module	2	PD1	32-bit counter for input capture or PWM output
EQEP	Enhanced Quadrature Encoder Pulse Module	3	PD1	32-bit counter for incremental encoders or capture
High-Speed Peripherals				
EPI	External Peripheral Interface	1	PD1	External memory/data interface with SDRAM/ASRAM, FPGA support
System Clock Modules				
SYSOSC	System Oscillator	1	PD0	4MHz / 32MHz
LFOSC	Low-frequency Oscillator	1	PD0	32KHz
XTAL	External high-frequency Crystal Connection	1	PD1	10MHz-25MHz
HFCLK	External high-frequency clock input	1	PD1	4MHz-48MHz
PLL	Phase-Locked Loop	1		200MHz
System Power Modules				
REFSYS	Reference System	1	PD0	Supplies required current and voltage references to all analog modules
POR	Power On Reset circuit	1	PD0	
BOR	Brown Out Reset circuit	1	PD0	
TEMPSENSE	Temperature Sensor	1	PD0	
Safety & Security Modules				
AES	Advanced Encryption Standard encryption/decryption	1	PD1	
CRCP	Cyclic Redundancy Checker	1	PD1	32-bit CRC
GSC	Global Security Controller	1	PD1	
Keystore	Keystore Controller	1	PD1	Controller that provides secure management of the AES keys
Crossbar (XBAR) Modules				
INPUTXBAR	Flexible Signal Multiplex Input Crossbar	1	PD1	Routes external input signals to a GPIO pin to internal peripheral blocks
OUTPUTXBAR	Flexible Signal Multiplex Output Crossbar	1	PD1	Routes signals from internal peripheral blocks to an external GPIO pin

ADVANCE INFORMATION

Table 1-1. AM13E230x Module Allocation and Instances (continued)

Module Abbreviation	Module Full Name	Size / Device Instances	Power Domain	Notes
PWMXBAR	PWM Signal Crossbar	1	PD1	Routes input signals from internal peripheral blocks to an MCPWM instance

1.3 Platform Memory Map

All AM13E230x devices share a common platform memory map. Peripherals are assigned a fixed address space and have the same address space on all devices within the family. The memory map is compliant with the standard ARM® Cortex®-M memory regions.

Note

All regions are non-secure. AM13E230x devices do not have ARM® TrustZone technology enabled.

Table 1-2. Top Level Memory Map

Memory Region	Start Address	End Address	Description
Code	0x0000.0000	0x1FFF.FFFF	Flash memory, ROM, SRAM Bank 3 (dual mapped in Code Region)
SRAM	0x2000.0000	0x3FFF.FFFF	SRAM Banks 0-3
Peripheral	0x4000.0000	0x5FFF.FFFF	Global peripheral memory-mapped registers and global nonexecutable data memory
Subsystem	0x6000.0000	0x7FFF.FFFF	Local CPU subsystem memory-mapped registers. Includes FLASH special regions.
External Memory	0x8000.0000	0xDFFF.FFFF	External memory via EPI
System PPB	0xE000.0000	0xFFFF.FFFF	ARM Private Peripheral Bus

1.3.1 Code Region

Table 1-3. Code Region Memory Map

Memory Region	Start Address	End Address
Flash Main Bank 0	0x0000.0000	0x0003.FFFF
Flash Main Bank 1	0x0004.0000	0x0007.FFFF
SRAM Bank 3	0x00C1.8000	0x00C1.FFFF
ROM	0x0100.0000	0x0100.BFFF

The code region contains the flash memory used to store executable code and data. Accesses to the flash memory from the CPU through the code region are processed through the AHB bus matrix to the flash read interface directly.

The code region also contains the read-only memory (ROM) used for the TI device boot code and the bootstrap loader. The ROM is only available during the initial device boot process.

SRAM Bank 3 is also dual mapped to this region and the SRAM Region.

1.3.2 SRAM Region

Table 1-4. SRAM Region Memory Map

Memory Region	Start Address	End Address
SRAM Bank 0	0x2000.0000	0x2000.7FFF
SRAM Bank 1	0x2000.8000	0x2000.FFFF
SRAM Bank 2	0x2001.0000	0x2001.7FFF

ADVANCE INFORMATION

Table 1-4. SRAM Region Memory Map (continued)

Memory Region	Start Address	End Address
SRAM Bank 3	0x2001.8000	0x2001.FFFF

The SRAM region contains the system memory (SRAM). The SRAM supports zero wait state access as documented in the device specific datasheet.

AM13E230x devices have up to 4 banks of SRAM. Each bank is 32KB and has hardware parity support. Accesses to SRAM banks 0-3 from the CPU are processed through the System AHB (S-AHB) bus matrix to the SRAM interface directly. SRAM bank 3 can also be accessed from the CPU through the CPU AHB (C-AHB) bus matrix.

See the device-specific data sheet for the amount of SRAM present on a given device.

1.3.3 Peripheral Region

The peripheral region contains the memory-mapped peripherals on the peripheral buses.

1.3.4 Subsystem Region

The Subsystem region contains local CPU subsystem (CPUSS) memory-mapped registers. This region includes flash regions for NONMAIN and FACTORY data.

1.3.5 External Memory Region

The External Memory Region contains address space for an external memory device connected to the AM13E230x using the EPI interface. For more information, refer to the External Peripheral Interface (EPI) section.

1.3.6 System PPB Region

The system private peripheral bus (PPB) region contains memory-mapped registers on the ARM private peripheral bus. These registers are tightly coupled to the CPU and are the interface for peripherals such as the memory protection unit (MPU), SysTick timer, and CPU power management and reset functions.

1.4 Boot Configuration

After a BOOTRST, the device executes the start-up boot routines to configure the device for operation before starting the main application. Boot routines are executed from read-only memory (ROM) before the main application is started. There are two boot routines: the Boot Configuration Routine (BCR) and Bootstrap Loader (BSL). The boot configuration routine sets up the device security policies, configures the device for operation, and optionally starts the BSL if it presents. The BSL, if started by the BCR, can be used to program or verify the device memory (flash and SRAM) through the use of a standard serial interface (UART, I2C, or MCAN).

After the start-up routines have successfully completed execution, the CPU is reset and the application is started by unconditionally fetching the stack pointer (SP) and reset vector from 0x0000.0000 and 0x0000.0004 of the flash memory for non-secure boot devices. To enable secure boot, a signed application image contains the SP and reset vector in the header which is interpreted by the ROM and updated in the GSC VTOR location.

1.4.1 Configuration Memory

The configuration memory is a dedicated region of flash memory (not the same as the code region) which stores the configuration data used by the BCR and BSL to boot the device. The region is not used for any other purpose. The BCR and BSL both have configuration policies which can be left at their default values (as is typical during development and evaluation), or modified for specific purposes (as is typical during production programming) by altering the values programmed into the configuration flash region.

The BCR and BSL configuration data structures are both contained within a single flash sector in the configuration memory region. To change any parameter in the boot configuration, it is necessary to erase the entire configuration memory and re-program both the BCR and BSL configuration structures with the desired settings.

The configuration data in this region is not affected by a mass erase command, but it is erased and re-programmed to factory defaults by a factory reset command sent to the BCR via the debug sub system mailbox (DSSM) over SWD.

The configuration memory is also erased by a factory reset command sent to the BSL using the UART, I2C, or MCAN BSL interface. However, unlike the DSSM factory reset, the BSL factory reset does not program TI factory defaults to the configuration memory following the erase. As such, it is the responsibility of the host which is connected to the microcontroller target (via the BSL interface) to re-program the configuration memory with a valid configuration before terminating the BSL session.

Note

If a factory reset command is executed through the BSL, and a valid configuration is not programmed back into the device before the BSL session is terminated, the device will assume a maximally restrictive state upon the next reset cycle, and it will not be possible to access the device via SWD or the BSL. Always ensure that a valid configuration is programmed back when using the BSL factory reset command.

The address ranges for the NONMAIN data structures are given in the *NONMAIN Registers* section. A detailed breakdown of the NONMAIN region is provided at the end of this section.

1.4.2 FLNONMAINECC Registers

Table 1-5 lists the memory-mapped registers for the FLNONMAINECC registers. All register offset addresses not listed in Table 1-5 should be considered as reserved locations and the register contents should not be modified.

Table 1-5. FLNONMAINECC Registers

Offset	Acronym	Register Name	Section
60100800h	BCR_CONFIG_ID	Predetermined Bootcode config signature ID	Section 1
60100804h	BOOTCFG0	Controls enable/disable of AHB-AP, ET-AP, PWR-AP. Values: DEBUG_EN, DEBUG_EN_PW, DEBUG_NS_EN, DEBUG_NS_EN_PW, DEBUG_DIS; Serial Wire Debug Port mode. Disabling this will disable all Debug ports including CFG_AP, SEC_AP. When disabled no DSSM commands are serviced	Section 2
60100808h	BOOTCFG1	Controls BSL pin invocation capability. Values: BSL_PIN_INVOKE_EN, BSL_PIN_INVOKE_DIS; Controls debug access release until INITDONE is issued	Section 3
6010080Ch	BOOTCFG2	Controls CSC policy in SYSCTL. YES enables CSC policy checking; Controls flash bank swap policy in SYSCTL	Section 4
60100810h	BOOTCFG3	Fast boot mode configuration - Skips certain boot time checks when enabled; Bootloader mode enable/disable control. Must be enabled for BSL functionality	Section 5
60100814h	BOOTCFG4	Mass erase mode configuration. Values: BC_CFG_MASS_ERASE_EN(0xAABB), BC_CFG_MASS_ERASE_EN_PW(0xCCDD), BC_CFG_MASS_ERASE_DIS(0x5522); Factory reset mode configuration. Values: BC_CFG_FACTORY_RESET_EN(0xAABB), BC_CFG_FACTORY_RESET_EN_PW(0xCCDD), BC_CFG_FACTORY_RESET_DIS(0x5522)	Section 6
60100818h	MASS_ERASE_0	Password for DSSM_BC_MASS_ERASE_REQUEST command - Word 0	Section 7
6010081Ch	MASS_ERASE_1	Password for DSSM_BC_MASS_ERASE_REQUEST command - Word 1	Section 8
60100820h	MASS_ERASE_2	Password for DSSM_BC_MASS_ERASE_REQUEST command - Word 2	Section 9
60100824h	MASS_ERASE_3	Password for DSSM_BC_MASS_ERASE_REQUEST command - Word 3	Section 10
60100828h	MASS_ERASE_4	Password for DSSM_BC_MASS_ERASE_REQUEST command - Word 4	Section 11
6010082Ch	MASS_ERASE_5	Password for DSSM_BC_MASS_ERASE_REQUEST command - Word 5	Section 12
60100830h	MASS_ERASE_6	Password for DSSM_BC_MASS_ERASE_REQUEST command - Word 6	Section 13
60100834h	MASS_ERASE_7	Password for DSSM_BC_MASS_ERASE_REQUEST command - Word 7	Section 14
60100838h	FACTORY_RESET_0	Password for factory reset command - Word 0	Section 15
6010083Ch	FACTORY_RESET_1	Password for factory reset command - Word 1	Section 16
60100840h	FACTORY_RESET_2	Password for factory reset command - Word 2	Section 17
60100844h	FACTORY_RESET_3	Password for factory reset command - Word 3	Section 18
60100848h	FACTORY_RESET_4	Password for factory reset command - Word 4	Section 19
6010084Ch	FACTORY_RESET_5	Password for factory reset command - Word 5	Section 20
60100850h	FACTORY_RESET_6	Password for factory reset command - Word 6	Section 21
60100854h	FACTORY_RESET_7	Password for factory reset command - Word 7	Section 22

Table 1-5. FLNONMAINECC Registers (continued)

Offset	Acronym	Register Name	Section
60100858h	DEBUG_LOCK_0	Password for debug access authentication - Word 0	Section 23
6010085Ch	DEBUG_LOCK_1	Password for debug access authentication - Word 1	Section 24
60100860h	DEBUG_LOCK_2	Password for debug access authentication - Word 2	Section 25
60100864h	DEBUG_LOCK_3	Password for debug access authentication - Word 3	Section 26
60100868h	DEBUG_LOCK_4	Password for debug access authentication - Word 4	Section 27
6010086Ch	DEBUG_LOCK_5	Password for debug access authentication - Word 5	Section 28
60100870h	DEBUG_LOCK_6	Password for debug access authentication - Word 6	Section 29
60100874h	DEBUG_LOCK_7	Password for debug access authentication - Word 7	Section 30
60100878h	DEBUG_NS_LOCK_0	Password for non-secure debug access - Word 0	Section 31
6010087Ch	DEBUG_NS_LOCK_1	Password for non-secure debug access - Word 1	Section 32
60100880h	DEBUG_NS_LOCK_2	Password for non-secure debug access - Word 2	Section 33
60100884h	DEBUG_NS_LOCK_3	Password for non-secure debug access - Word 3	Section 34
60100888h	DEBUG_NS_LOCK_4	Password for non-secure debug access - Word 4	Section 35
6010088Ch	DEBUG_NS_LOCK_5	Password for non-secure debug access - Word 5	Section 36
60100890h	DEBUG_NS_LOCK_6	Password for non-secure debug access - Word 6	Section 37
60100894h	DEBUG_NS_LOCK_7	Password for non-secure debug access - Word 7	Section 38
60100898h	RESERVED_REG0		
6010089Ch	RESERVED_REG1		
601008A0h	RESERVED_REG2		
601008A4h	RESERVED_REG3		
601008A8h	RESERVED_REG4		
601008ACh	RESERVED_REG5	Reserved - Word 5	
601008B0h	RESERVED_REG6	Reserved - Word 6	
601008B4h	RESERVED_REG7	Reserved - Word 7	
601008B8h	SECURE_BOOT_MODE	Controls application authentication; Reserved field	
601008BCh	USER_SECURE_APP_START_ADDR	Starting address for secure application	Section 40
601008C0h	USER_SECURE_APP_LENGTH	Length of secure application	Section 41
601008C4h	USER_SECURE_APP_CRC_0	Secure application CRC - Word 0	Section 42
601008C8h	RESERVED_REG8	Reserved field - Word 1	
601008CCh	RESERVED_REG9	Reserved field - Word 2	
601008D0h	RESERVED_REG10	Reserved field - Word 3	
601008D4h	RESERVED_REG11	Reserved field - Word 4	
601008D8h	RESERVED_REG12	Reserved field - Word 5	
601008DCh	RESERVED_REG13	Reserved field - Word 6	
601008E0h	RESERVED_REG14	Reserved field - Word 7	
601008E4h	BANK0_NM_USER_CONFIG	User Config Non Main Flash Static Write Protection. WEP_DISABLE=0xAABB; Reserved field	
601008E8h	BANK0_WRITE_ERASE_PROTECTION_A	Write protection for first 32 sectors	Section 44
601008ECh	BANK0_WRITE_ERASE_PROTECTION_B	Write Protection for 512KB-64KB	Section 45
601008F0h	BANK0_SECURITY_PROTECTION_A	Security protection for first 32 sectors	Section 46
601008F4h	BANK0_SECURITY_PROTECTION_B	Security protection for 512KB-64KB	Section 47
601008F8h	BANK0_PRIVILEGE_PROTECTION_A	Privilege protection for first 32 sectors	Section 48
601008FCh	BANK0_PRIVILEGE_PROTECTION_B	Privilege protection for 512KB-64KB	Section 49
60100900h	BANK0_RESERVED	Reserved field for BANK0 alignment	

Table 1-5. FLNONMAINECC Registers (continued)

Offset	Acronym	Register Name	Section
60100904h	BANK1_WRITE_ERASE_PROTECTION_A	Write protection for first 32 sectors	Section 50
60100908h	BANK1_WRITE_ERASE_PROTECTION_B	Write Protection for 512KB-64KB	Section 51
6010090Ch	BANK1_SECURITY_PROTECTION_A	Security protection for first 32 sectors	Section 52
60100910h	BANK1_SECURITY_PROTECTION_B	Security protection for 512KB-64KB	Section 53
60100914h	BANK1_PRIVILEGE_PROTECTION_A	Privilege protection for first 32 sectors	Section 54
60100918h	BANK1_PRIVILEGE_PROTECTION_B	Privilege protection for 512KB-64KB	Section 55
6010091Ch	BANK1_RESERVED	Reserved field for BANK1	
60100920h	RESERVED_REG15	Reserved field	
60100924h	RESERVED_REG16	Reserved field	
60100928h	RESERVED_REG17	Reserved field	
6010092Ch	RESERVED_REG18	Reserved field	
60100930h	BOOTCLK0	Clock configuration; PLL multiplier value; PLL divider value; PLL clock source configuration	Section 56
60100934h	BOOTCLK1	CPU delay cycles for PLL settling time; CPU delay cycles for XTAL monitoring; Reserved configuration field 0	
60100938h	RESERVED_REG_0	Reserved configuration field 0; Reserved configuration field 1	
6010093Ch	RESERVED_REG_1	Reserved configuration field 2; Reserved configuration field 3	
60100940h	RESERVED_REG_2	Reserved configuration field 4; Reserved configuration field 5	
60100944h	RESERVED_REG_3	Reserved configuration field 6; Reserved configuration field 7	
60100948h	RESERVED_REG_4	Reserved configuration field 8; Reserved configuration field 9	
6010094Ch	CRC	JAMCRC of BCR config structure	Section 58
60100C00h	BSL_CONFIG_ID	Predetermined Bootloader config signature ID	Section 59
60100C04h	BSLPINCFG0	UART receive pin number configuration; UART receive pin multiplexer selection; UART transmit pin number configuration; UART transmit pin multiplexer selection	Section 60
60100C08h	BSLPINCFG1	I2C data pin number configuration; I2C data pin multiplexer selection; I2C clock pin number configuration; I2C clock pin multiplexer selection	Section 61
60100C0Ch	BSLPINCFG2	MCAN receive pin number configuration; MCAN receive pin multiplexer selection; MCAN transmit pin number configuration; MCAN transmit pin multiplexer selection	Section 62
60100C10h	BSLCONFIG0	BSL invoke pin configuration data 0; BSL invoke pin configuration data 1; Memory readout control. ENABLE allows memory read operations	Section 63
60100C14h	PASSWORD_0	BSL access password - Word 0	Section 64
60100C18h	PASSWORD_1	BSL access password - Word 1	Section 65
60100C1Ch	PASSWORD_2	BSL access password - Word 2	Section 66
60100C20h	PASSWORD_3	BSL access password - Word 3	Section 67
60100C24h	PASSWORD_4	BSL access password - Word 4	Section 68
60100C28h	PASSWORD_5	BSL access password - Word 5	Section 69
60100C2Ch	PASSWORD_6	BSL access password - Word 6	Section 70
60100C30h	PASSWORD_7	BSL access password - Word 7	Section 71
60100C34h	APP_REV_POINTER	Pointer to application version information in MAIN flash	Section 72

Table 1-5. FLNONMAINECC Registers (continued)

Offset	Acronym	Register Name	Section
60100C38h	BSLCONFIG1	Security alert response: Factory Reset/Disable BSL/ Ignore; UART communication speed selection for ROM BSL	Section 73
60100C3Ch	I2C_SLAVE_ADDR	I2C slave address for ROM BSL I2C interface; Reserved field; Reserved field	
60100C40h	RESERVED_REG_0	Reserved field; Reserved field; Reserved field; Reserved field	
60100C44h	RESERVED_REG_1	Reserved field; Reserved field; Reserved field; Reserved field	
60100C48h	RESERVED_REG_2	Reserved field; Reserved field; Reserved field; Reserved field	
60100C4Ch	CRC	JAMCRC of BSL configuration structure	

Complex bit access types are encoded to fit into small table cells. Table 1-6 shows the codes that are used for access types in this section.

Table 1-6. FLNONMAINECC Access Type Codes

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

1 BCR_CONFIG_ID Register (Offset = 60100800h) [Reset = 00000000h]

BCR_CONFIG_ID is shown in Table 1-7.

Return to the Summary Table.

Predetermined Bootcode config signature ID

Table 1-7. BCR_CONFIG_ID Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	BCR_CONFIG_ID	R/W	0h	Predetermined Bootcode config signature ID

2 BOOTCFG0 Register (Offset = 60100804h) [Reset = 00000000h]

BOOTCFG0 is shown in Table 1-8.

Return to the Summary Table.

Controls enable/disable of AHB-AP, ET-AP, PWR-AP. Values: DEBUG_EN, DEBUG_EN_PW, DEBUG_NS_EN, DEBUG_NS_EN_PW, DEBUG_DIS; Serial Wire Debug Port mode. Disabling this will disable all Debug ports including CFG_AP, SEC_AP. When disabled no DSSM commands are serviced

Table 1-8. BOOTCFG0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	SWDP_MODE	R/W	0h	Serial Wire Debug Port mode. Disabling this will disable all Debug ports including CFG_AP, SEC_AP. When disabled no DSSM commands are serviced 5522h = DISABLED configuration AABBh = ENABLED configuration
15-0	DEBUG_ACCESS	R/W	0h	Controls enable/disable of AHB-AP, ET-AP, PWR-AP. Values: DEBUG_EN, DEBUG_EN_PW, DEBUG_NS_EN, DEBUG_NS_EN_PW, DEBUG_DIS 5522h = DISABLED configuration AABBh = ENABLED configuration

3 BOOTCFG1 Register (Offset = 60100808h) [Reset = 00000000h]

BOOTCFG1 is shown in Table 1-9.

Return to the Summary Table.

Controls BSL pin invocation capability. Values: BSL_PIN_INVOKE_EN, BSL_PIN_INVOKE_DIS; Controls debug access release until INITDONE is issued

Table 1-9. BOOTCFG1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	DEBUG_HOLD	R/W	0h	Controls debug access release until INITDONE is issued
15-0	BSL_PIN_INVOKE	R/W	0h	Controls BSL pin invocation capability. Values: BSL_PIN_INVOKE_EN, BSL_PIN_INVOKE_DIS 5522h = DISABLED configuration AABBh = ENABLED configuration

4 BOOTCFG2 Register (Offset = 6010080Ch) [Reset = 00000000h]

BOOTCFG2 is shown in Table 1-10.

Return to the Summary Table.

Controls CSC policy in SYSCTL. YES enables CSC policy checking; Controls flash bank swap policy in SYSCTL

Table 1-10. BOOTCFG2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	FLASH_BANK_SWAP_POLICY	R/W	0h	Controls flash bank swap policy in SYSCTL
15-0	CSC_EXISTS	R/W	0h	Controls CSC policy in SYSCTL. YES enables CSC policy checking 5522h = DISABLED configuration AABBh = ENABLED configuration

5 BOOTCFG3 Register (Offset = 60100810h) [Reset = 00000000h]

BOOTCFG3 is shown in Table 1-11.

Return to the Summary Table.

Fast boot mode configuration - Skips certain boot time checks when enabled; Bootloader mode enable/disable control. Must be enabled for BSL functionality

Table 1-11. BOOTCFG3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	BOOTLOADER_MODE	R/W	0h	Bootloader mode enable/disable control. Must be enabled for BSL functionality
15-0	FAST_BOOT_MODE	R/W	0h	Fast boot mode configuration - Skips certain boot time checks when enabled

6 BOOTCFG4 Register (Offset = 60100814h) [Reset = 00000000h]

BOOTCFG4 is shown in Table 1-12.

Return to the Summary Table.

Mass erase mode configuration. Values: BC_CFG_MASS_ERASE_EN(0xAABB), BC_CFG_MASS_ERASE_EN_PW(0xCCDD), BC_CFG_MASS_ERASE_DIS(0x5522); Factory reset mode configuration. Values: BC_CFG_FACTORY_RESET_EN(0xAABB), BC_CFG_FACTORY_RESET_EN_PW(0xCCDD), BC_CFG_FACTORY_RESET_DIS(0x5522)

Table 1-12. BOOTCFG4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	FACTORY_RESET_MODE	R/W	0h	Factory reset mode configuration. Values: BC_CFG_FACTORY_RESET_EN(0xAABB), BC_CFG_FACTORY_RESET_EN_PW(0xCCDD), BC_CFG_FACTORY_RESET_DIS(0x5522) 5522h = DISABLE configuration AABBh = ENABLE configuration CCDDh = ENABLE_WITH_PASSWORD configuration
15-0	MASS_ERASE_MODE	R/W	0h	Mass erase mode configuration. Values: BC_CFG_MASS_ERASE_EN(0xAABB), BC_CFG_MASS_ERASE_EN_PW(0xCCDD), BC_CFG_MASS_ERASE_DIS(0x5522) 5522h = DISABLE configuration AABBh = ENABLE configuration CCDDh = ENABLE_WITH_PASSWORD configuration

7 MASS_ERASE_0 Register (Offset = 60100818h) [Reset = 00000000h]

MASS_ERASE_0 is shown in Table 1-13.

Return to the Summary Table.

Password for DSSM_BC_MASS_ERASE_REQUEST command - Word 0

Table 1-13. MASS_ERASE_0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	MASS_ERASE_0	R/W	0h	Password for DSSM_BC_MASS_ERASE_REQUEST command - Word 0

8 MASS_ERASE_1 Register (Offset = 6010081Ch) [Reset = 00000000h]

MASS_ERASE_1 is shown in Table 1-14.

Return to the Summary Table.

Password for DSSM_BC_MASS_ERASE_REQUEST command - Word 1

Table 1-14. MASS_ERASE_1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	MASS_ERASE_1	R/W	0h	Password for DSSM_BC_MASS_ERASE_REQUEST command - Word 1

9 MASS_ERASE_2 Register (Offset = 60100820h) [Reset = 00000000h]

MASS_ERASE_2 is shown in Table 1-15.

Return to the Summary Table.

Password for DSSM_BC_MASS_ERASE_REQUEST command - Word 2

Table 1-15. MASS_ERASE_2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	MASS_ERASE_2	R/W	0h	Password for DSSM_BC_MASS_ERASE_REQUEST command - Word 2

10 MASS_ERASE_3 Register (Offset = 60100824h) [Reset = 00000000h]

MASS_ERASE_3 is shown in Table 1-16.

Return to the Summary Table.

Password for DSSM_BC_MASS_ERASE_REQUEST command - Word 3

Table 1-16. MASS_ERASE_3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	MASS_ERASE_3	R/W	0h	Password for DSSM_BC_MASS_ERASE_REQUEST command - Word 3

11 MASS_ERASE_4 Register (Offset = 60100828h) [Reset = 00000000h]

MASS_ERASE_4 is shown in Table 1-17.

Return to the Summary Table.

Password for DSSM_BC_MASS_ERASE_REQUEST command - Word 4

Table 1-17. MASS_ERASE_4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	MASS_ERASE_4	R/W	0h	Password for DSSM_BC_MASS_ERASE_REQUEST command - Word 4

12 MASS_ERASE_5 Register (Offset = 6010082Ch) [Reset = 00000000h]

MASS_ERASE_5 is shown in Table 1-18.

Return to the Summary Table.

Password for DSSM_BC_MASS_ERASE_REQUEST command - Word 5

Table 1-18. MASS_ERASE_5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	MASS_ERASE_5	R/W	0h	Password for DSSM_BC_MASS_ERASE_REQUEST command - Word 5

13 MASS_ERASE_6 Register (Offset = 60100830h) [Reset = 00000000h]

MASS_ERASE_6 is shown in Table 1-19.

Return to the Summary Table.

Password for DSSM_BC_MASS_ERASE_REQUEST command - Word 6

Table 1-19. MASS_ERASE_6 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	MASS_ERASE_6	R/W	0h	Password for DSSM_BC_MASS_ERASE_REQUEST command - Word 6

14 MASS_ERASE_7 Register (Offset = 60100834h) [Reset = 00000000h]

MASS_ERASE_7 is shown in Table 1-20.

Return to the Summary Table.

Password for DSSM_BC_MASS_ERASE_REQUEST command - Word 7

Table 1-20. MASS_ERASE_7 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	MASS_ERASE_7	R/W	0h	Password for DSSM_BC_MASS_ERASE_REQUEST command - Word 7

15 FACTORY_RESET_0 Register (Offset = 60100838h) [Reset = 00000000h]

FACTORY_RESET_0 is shown in Table 1-21.

Return to the Summary Table.

Password for factory reset command - Word 0

Table 1-21. FACTORY_RESET_0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	FACTORY_RESET_0	R/W	0h	Password for factory reset command - Word 0

16 FACTORY_RESET_1 Register (Offset = 6010083Ch) [Reset = 00000000h]

FACTORY_RESET_1 is shown in Table 1-22.

Return to the Summary Table.

Password for factory reset command - Word 1

Table 1-22. FACTORY_RESET_1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	FACTORY_RESET_1	R/W	0h	Password for factory reset command - Word 1

17 FACTORY_RESET_2 Register (Offset = 60100840h) [Reset = 00000000h]

FACTORY_RESET_2 is shown in Table 1-23.

Return to the Summary Table.

Password for factory reset command - Word 2

Table 1-23. FACTORY_RESET_2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	FACTORY_RESET_2	R/W	0h	Password for factory reset command - Word 2

18 FACTORY_RESET_3 Register (Offset = 60100844h) [Reset = 00000000h]

FACTORY_RESET_3 is shown in Table 1-24.

Return to the Summary Table.

Password for factory reset command - Word 3

Table 1-24. FACTORY_RESET_3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	FACTORY_RESET_3	R/W	0h	Password for factory reset command - Word 3

19 FACTORY_RESET_4 Register (Offset = 60100848h) [Reset = 00000000h]

FACTORY_RESET_4 is shown in Table 1-25.

Return to the Summary Table.

Password for factory reset command - Word 4

Table 1-25. FACTORY_RESET_4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	FACTORY_RESET_4	R/W	0h	Password for factory reset command - Word 4

20 FACTORY_RESET_5 Register (Offset = 6010084Ch) [Reset = 00000000h]

FACTORY_RESET_5 is shown in Table 1-26.

Return to the Summary Table.

Password for factory reset command - Word 5

Table 1-26. FACTORY_RESET_5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	FACTORY_RESET_5	R/W	0h	Password for factory reset command - Word 5

21 FACTORY_RESET_6 Register (Offset = 60100850h) [Reset = 00000000h]

FACTORY_RESET_6 is shown in Table 1-27.

Return to the Summary Table.

Password for factory reset command - Word 6

Table 1-27. FACTORY_RESET_6 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	FACTORY_RESET_6	R/W	0h	Password for factory reset command - Word 6

22 FACTORY_RESET_7 Register (Offset = 60100854h) [Reset = 00000000h]

FACTORY_RESET_7 is shown in Table 1-28.

Return to the Summary Table.

Password for factory reset command - Word 7

Table 1-28. FACTORY_RESET_7 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	FACTORY_RESET_7	R/W	0h	Password for factory reset command - Word 7

23 DEBUG_LOCK_0 Register (Offset = 60100858h) [Reset = 00000000h]

DEBUG_LOCK_0 is shown in Table 1-29.

Return to the Summary Table.

Password for debug access authentication - Word 0

Table 1-29. DEBUG_LOCK_0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DEBUG_LOCK_0	R/W	0h	Password for debug access authentication - Word 0

24 DEBUG_LOCK_1 Register (Offset = 6010085Ch) [Reset = 00000000h]

DEBUG_LOCK_1 is shown in Table 1-30.

Return to the Summary Table.

Password for debug access authentication - Word 1

Table 1-30. DEBUG_LOCK_1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DEBUG_LOCK_1	R/W	0h	Password for debug access authentication - Word 1

25 DEBUG_LOCK_2 Register (Offset = 60100860h) [Reset = 00000000h]

DEBUG_LOCK_2 is shown in Table 1-31.

Return to the Summary Table.

Password for debug access authentication - Word 2

Table 1-31. DEBUG_LOCK_2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DEBUG_LOCK_2	R/W	0h	Password for debug access authentication - Word 2

26 DEBUG_LOCK_3 Register (Offset = 60100864h) [Reset = 00000000h]

DEBUG_LOCK_3 is shown in Table 1-32.

Return to the Summary Table.

Password for debug access authentication - Word 3

Table 1-32. DEBUG_LOCK_3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DEBUG_LOCK_3	R/W	0h	Password for debug access authentication - Word 3

27 DEBUG_LOCK_4 Register (Offset = 60100868h) [Reset = 00000000h]

DEBUG_LOCK_4 is shown in Table 1-33.

Return to the Summary Table.

Password for debug access authentication - Word 4

Table 1-33. DEBUG_LOCK_4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DEBUG_LOCK_4	R/W	0h	Password for debug access authentication - Word 4

28 DEBUG_LOCK_5 Register (Offset = 6010086Ch) [Reset = 00000000h]

DEBUG_LOCK_5 is shown in Table 1-34.

Return to the Summary Table.

Password for debug access authentication - Word 5

Table 1-34. DEBUG_LOCK_5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DEBUG_LOCK_5	R/W	0h	Password for debug access authentication - Word 5

29 DEBUG_LOCK_6 Register (Offset = 60100870h) [Reset = 00000000h]

DEBUG_LOCK_6 is shown in Table 1-35.

Return to the Summary Table.

Password for debug access authentication - Word 6

Table 1-35. DEBUG_LOCK_6 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DEBUG_LOCK_6	R/W	0h	Password for debug access authentication - Word 6

30 DEBUG_LOCK_7 Register (Offset = 60100874h) [Reset = 00000000h]

DEBUG_LOCK_7 is shown in Table 1-36.

Return to the Summary Table.

Password for debug access authentication - Word 7

Table 1-36. DEBUG_LOCK_7 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DEBUG_LOCK_7	R/W	0h	Password for debug access authentication - Word 7

31 DEBUG_NS_LOCK_0 Register (Offset = 60100878h) [Reset = 00000000h]

DEBUG_NS_LOCK_0 is shown in Table 1-37.

Return to the Summary Table.

Password for non-secure debug access - Word 0

Table 1-37. DEBUG_NS_LOCK_0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DEBUG_NS_LOCK_0	R/W	0h	Password for non-secure debug access - Word 0

32 DEBUG_NS_LOCK_1 Register (Offset = 6010087Ch) [Reset = 00000000h]

DEBUG_NS_LOCK_1 is shown in Table 1-38.

Return to the Summary Table.

Password for non-secure debug access - Word 1

Table 1-38. DEBUG_NS_LOCK_1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DEBUG_NS_LOCK_1	R/W	0h	Password for non-secure debug access - Word 1

33 DEBUG_NS_LOCK_2 Register (Offset = 60100880h) [Reset = 00000000h]

DEBUG_NS_LOCK_2 is shown in Table 1-39.

Return to the Summary Table.

Password for non-secure debug access - Word 2

Table 1-39. DEBUG_NS_LOCK_2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DEBUG_NS_LOCK_2	R/W	0h	Password for non-secure debug access - Word 2

34 DEBUG_NS_LOCK_3 Register (Offset = 60100884h) [Reset = 00000000h]

DEBUG_NS_LOCK_3 is shown in Table 1-40.

Return to the Summary Table.

Password for non-secure debug access - Word 3

Table 1-40. DEBUG_NS_LOCK_3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DEBUG_NS_LOCK_3	R/W	0h	Password for non-secure debug access - Word 3

35 DEBUG_NS_LOCK_4 Register (Offset = 60100888h) [Reset = 00000000h]

DEBUG_NS_LOCK_4 is shown in Table 1-41.

Return to the Summary Table.

Password for non-secure debug access - Word 4

Table 1-41. DEBUG_NS_LOCK_4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DEBUG_NS_LOCK_4	R/W	0h	Password for non-secure debug access - Word 4

36 DEBUG_NS_LOCK_5 Register (Offset = 6010088Ch) [Reset = 00000000h]

DEBUG_NS_LOCK_5 is shown in Table 1-42.

Return to the Summary Table.

Password for non-secure debug access - Word 5

Table 1-42. DEBUG_NS_LOCK_5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DEBUG_NS_LOCK_5	R/W	0h	Password for non-secure debug access - Word 5

37 DEBUG_NS_LOCK_6 Register (Offset = 60100890h) [Reset = 00000000h]

DEBUG_NS_LOCK_6 is shown in Table 1-43.

Return to the Summary Table.

Password for non-secure debug access - Word 6

Table 1-43. DEBUG_NS_LOCK_6 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DEBUG_NS_LOCK_6	R/W	0h	Password for non-secure debug access - Word 6

38 DEBUG_NS_LOCK_7 Register (Offset = 60100894h) [Reset = 00000000h]

DEBUG_NS_LOCK_7 is shown in Table 1-44.

Return to the Summary Table.

Password for non-secure debug access - Word 7

Table 1-44. DEBUG_NS_LOCK_7 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DEBUG_NS_LOCK_7	R/W	0h	Password for non-secure debug access - Word 7

39 SECURE_BOOT_MODE Register (Offset = 601008B8h) [Reset = 00000000h]

SECURE_BOOT_MODE is shown in Table 1-45.

Return to the Summary Table.

Controls application authentication; Reserved field

Table 1-45. SECURE_BOOT_MODE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved field
15-0	SECURE_BOOT_MODE	R/W	0h	Controls application authentication AABBh = BC_CFG_SECURE_BOOT_CRC_EN configuration FFFFh = DISABLED configuration

40 USER_SECURE_APP_START_ADDR Register (Offset = 601008BCh) [Reset = 00000000h]

USER_SECURE_APP_START_ADDR is shown in Table 1-46.

Return to the Summary Table.

Starting address for secure application

Table 1-46. USER_SECURE_APP_START_ADDR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	USER_SECURE_APP_START_ADDR	R/W	0h	Starting address for secure application

41 USER_SECURE_APP_LENGTH Register (Offset = 601008C0h) [Reset = 00000000h]

USER_SECURE_APP_LENGTH is shown in Table 1-47.

Return to the Summary Table.

Length of secure application

Table 1-47. USER_SECURE_APP_LENGTH Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	USER_SECURE_APP_LENGTH	R/W	0h	Length of secure application

42 USER_SECURE_APP_CRC_0 Register (Offset = 601008C4h) [Reset = 00000000h]

USER_SECURE_APP_CRC_0 is shown in Table 1-48.

Return to the Summary Table.

Secure application CRC - Word 0

Table 1-48. USER_SECURE_APP_CRC_0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	USER_SECURE_APP_CRC_0	R/W	0h	Secure application CRC - Word 0

43 BANK0_NM_USER_CONFIG Register (Offset = 601008E4h) [Reset = 00000000h]

BANK0_NM_USER_CONFIG is shown in Table 1-49.

Return to the Summary Table.

User Config Non Main Flash Static Write Protection. WEP_DISABLE=0xAABB; Reserved field

Table 1-49. BANK0_NM_USER_CONFIG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved field
15-0	BANK0_NM_USER_CONFIG	R/W	0h	User Config Non Main Flash Static Write Protection. WEP_DISABLE=0xAABB

44 BANK0_WRITE_ERASE_PROTECTION_A Register (Offset = 601008E8h) [Reset = 00000000h]

BANK0_WRITE_ERASE_PROTECTION_A is shown in Table 1-50.

Return to the Summary Table.

Write protection for first 32 sectors

Table 1-50. BANK0_WRITE_ERASE_PROTECTION_A Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	BANK0_WRITE_ERASE_PROTECTION_A	R/W	0h	Write protection for first 32 sectors

45 BANK0_WRITE_ERASE_PROTECTION_B Register (Offset = 601008ECh) [Reset = 00000000h]

BANK0_WRITE_ERASE_PROTECTION_B is shown in Table 1-51.

Return to the Summary Table.

Write Protection for 512KB-64KB

Table 1-51. BANK0_WRITE_ERASE_PROTECTION_B Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	BANK0_WRITE_ERASE_PROTECTION_B	R/W	0h	Write Protection for 512KB-64KB

46 BANK0_SECURITY_PROTECTION_A Register (Offset = 601008F0h) [Reset = 00000000h]

BANK0_SECURITY_PROTECTION_A is shown in Table 1-52.

Return to the Summary Table.

Security protection for first 32 sectors

Table 1-52. BANK0_SECURITY_PROTECTION_A Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	BANK0_SECURITY_PROTECTION_A	R/W	0h	Security protection for first 32 sectors

47 BANK0_SECURITY_PROTECTION_B Register (Offset = 601008F4h) [Reset = 00000000h]

BANK0_SECURITY_PROTECTION_B is shown in Table 1-53.

Return to the Summary Table.

Security protection for 512KB-64KB

Table 1-53. BANK0_SECURITY_PROTECTION_B Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	BANK0_SECURITY_PROTECTION_B	R/W	0h	Security protection for 512KB-64KB

48 BANK0_PRIVILEGE_PROTECTION_A Register (Offset = 601008F8h) [Reset = 00000000h]

BANK0_PRIVILEGE_PROTECTION_A is shown in Table 1-54.

Return to the Summary Table.

Privilege protection for first 32 sectors

Table 1-54. BANK0_PRIVILEGE_PROTECTION_A Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	BANK0_PRIVILEGE_PROTECTION_A	R/W	0h	Privilege protection for first 32 sectors

49 BANK0_PRIVILEGE_PROTECTION_B Register (Offset = 601008FCh) [Reset = 00000000h]

BANK0_PRIVILEGE_PROTECTION_B is shown in Table 1-55.

Return to the Summary Table.

Privilege protection for 512KB-64KB

Table 1-55. BANK0_PRIVILEGE_PROTECTION_B Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	BANK0_PRIVILEGE_PROTECTION_B	R/W	0h	Privilege protection for 512KB-64KB

50 BANK1_WRITE_ERASE_PROTECTION_A Register (Offset = 60100904h) [Reset = 00000000h]

BANK1_WRITE_ERASE_PROTECTION_A is shown in Table 1-56.

Return to the Summary Table.

Write protection for first 32 sectors

Table 1-56. BANK1_WRITE_ERASE_PROTECTION_A Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	BANK1_WRITE_ERASE_PROTECTION_A	R/W	0h	Write protection for first 32 sectors

51 BANK1_WRITE_ERASE_PROTECTION_B Register (Offset = 60100908h) [Reset = 00000000h]

BANK1_WRITE_ERASE_PROTECTION_B is shown in Table 1-57.

Return to the Summary Table.

Write Protection for 512KB-64KB

Table 1-57. BANK1_WRITE_ERASE_PROTECTION_B Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	BANK1_WRITE_ERASE_PROTECTION_B	R/W	0h	Write Protection for 512KB-64KB

52 BANK1_SECURITY_PROTECTION_A Register (Offset = 6010090Ch) [Reset = 00000000h]

BANK1_SECURITY_PROTECTION_A is shown in Table 1-58.

Return to the Summary Table.

Security protection for first 32 sectors

Table 1-58. BANK1_SECURITY_PROTECTION_A Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	BANK1_SECURITY_PROTECTION_A	R/W	0h	Security protection for first 32 sectors

53 BANK1_SECURITY_PROTECTION_B Register (Offset = 60100910h) [Reset = 00000000h]

BANK1_SECURITY_PROTECTION_B is shown in Table 1-59.

Return to the Summary Table.

Security protection for 512KB-64KB

Table 1-59. BANK1_SECURITY_PROTECTION_B Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	BANK1_SECURITY_PROTECTION_B	R/W	0h	Security protection for 512KB-64KB

54 BANK1_PRIVILEGE_PROTECTION_A Register (Offset = 60100914h) [Reset = 00000000h]

BANK1_PRIVILEGE_PROTECTION_A is shown in Table 1-60.

Return to the Summary Table.

Privilege protection for first 32 sectors

Table 1-60. BANK1_PRIVILEGE_PROTECTION_A Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	BANK1_PRIVILEGE_PROTECTION_A	R/W	0h	Privilege protection for first 32 sectors

55 BANK1_PRIVILEGE_PROTECTION_B Register (Offset = 60100918h) [Reset = 00000000h]

BANK1_PRIVILEGE_PROTECTION_B is shown in Table 1-61.

Return to the Summary Table.

Privilege protection for 512KB-64KB

Table 1-61. BANK1_PRIVILEGE_PROTECTION_B Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	BANK1_PRIVILEGE_PROTECTION_B	R/W	0h	Privilege protection for 512KB-64KB

56 BOOTCLK0 Register (Offset = 60100930h) [Reset = 00000000h]

BOOTCLK0 is shown in Table 1-62.

Return to the Summary Table.

Clock configuration; PLL multiplier value; PLL divider value; PLL clock source configuration

Table 1-62. BOOTCLK0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	SYSPLL_SOURCE	R/W	0h	PLL clock source configuration AAh = BC_CFG_PLL_SOURCE_XTAL_20MHZ configuration FFh = DEFAULT configuration
23-16	SYSPLL_CONFIG_RDIV	R/W	0h	PLL divider value
15-8	SYSPLL_CONFIG_QDIV	R/W	0h	PLL multiplier value
7-0	SYSPLL_CONFIG	R/W	0h	Clock configuration AAh = PLL_80 configuration BBh = PLL_CUSTOM configuration FFh = NOPLL configuration

57 BOOTCLK1 Register (Offset = 60100934h) [Reset = 00000000h]

BOOTCLK1 is shown in Table 1-63.

Return to the Summary Table.

CPU delay cycles for PLL settling time; CPU delay cycles for XTAL monitoring; Reserved configuration field 0

Table 1-63. BOOTCLK1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESVD_0	R/W	0h	Reserved configuration field 0
15-8	XTAL_STARTUP_MONITOR_TIME	R/W	0h	CPU delay cycles for XTAL monitoring
7-0	SYSPLL_SETTLING_TIME	R/W	0h	CPU delay cycles for PLL settling time

58 CRC Register (Offset = 6010094Ch) [Reset = 00000000h]

CRC is shown in Table 1-64.

Return to the Summary Table.

JAMCRC of BCR config structure

Table 1-64. CRC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CRC	R/W	0h	CRC-32 of BCR config structure

59 BSL_CONFIG_ID Register (Offset = 60100C00h) [Reset = 00000000h]

BSL_CONFIG_ID is shown in Table 1-65.

Return to the Summary Table.

Predetermined Bootloader config signature ID

Table 1-65. BSL_CONFIG_ID Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	BSL_CONFIG_ID	R/W	0h	Predetermined Bootloader config signature ID

60 BSLPINCFG0 Register (Offset = 60100C04h) [Reset = 00000000h]

BSLPINCFG0 is shown in Table 1-66.

Return to the Summary Table.

UART receive pin number configuration; UART receive pin multiplexer selection; UART transmit pin number configuration; UART transmit pin multiplexer selection

Table 1-66. BSLPINCFG0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	UART_TXD_PF_MUX_SEL	R/W	0h	UART transmit pin multiplexer selection
23-16	UART_TXD_PAD_NUM	R/W	0h	UART transmit pin number configuration
15-8	UART_RXD_PF_MUX_SEL	R/W	0h	UART receive pin multiplexer selection
7-0	UART_RXD_PAD_NUM	R/W	0h	UART receive pin number configuration

61 BSLPINCFG1 Register (Offset = 60100C08h) [Reset = 00000000h]

BSLPINCFG1 is shown in Table 1-67.

Return to the Summary Table.

I2C data pin number configuration; I2C data pin multiplexer selection; I2C clock pin number configuration; I2C clock pin multiplexer selection

Table 1-67. BSLPINCFG1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	I2C_SCL_PF_MUX_SEL	R/W	0h	I2C clock pin multiplexer selection
23-16	I2C_SCL_PAD_NUM	R/W	0h	I2C clock pin number configuration
15-8	I2C_SDA_PF_MUX_SEL	R/W	0h	I2C data pin multiplexer selection
7-0	I2C_SDA_PAD_NUM	R/W	0h	I2C data pin number configuration

62 BSLPINCFG2 Register (Offset = 60100C0Ch) [Reset = 00000000h]

BSLPINCFG2 is shown in Table 1-68.

Return to the Summary Table.

MCAN receive pin number configuration; MCAN receive pin multiplexer selection; MCAN transmit pin number configuration; MCAN transmit pin multiplexer selection

Table 1-68. BSLPINCFG2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	MCAN_TX_PF_MUX_SEL	R/W	0h	MCAN transmit pin multiplexer selection
23-16	MCAN_TX_PAD_NUM	R/W	0h	MCAN transmit pin number configuration
15-8	MCAN_RX_PF_MUX_SEL	R/W	0h	MCAN receive pin multiplexer selection
7-0	MCAN_RX_PAD_NUM	R/W	0h	MCAN receive pin number configuration

63 BSLCONFIG0 Register (Offset = 60100C10h) [Reset = 00000000h]

BSLCONFIG0 is shown in Table 1-69.

Return to the Summary Table.

BSL invoke pin configuration data 0; BSL invoke pin configuration data 1; Memory readout control. ENABLE allows memory read operations

Table 1-69. BSLCONFIG0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	READOUT	R/W	0h	Memory readout control. ENABLE allows memory read operations
15-8	PIN_DATA_1	R/W	0h	BSL invoke pin configuration data 1
7-0	PIN_DATA_0	R/W	0h	BSL invoke pin configuration data 0

64 PASSWORD_0 Register (Offset = 60100C14h) [Reset = 00000000h]

PASSWORD_0 is shown in Table 1-70.

Return to the Summary Table.

BSL access password - Word 0

Table 1-70. PASSWORD_0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	PASSWORD_0	R/W	0h	BSL access password - Word 0

65 PASSWORD_1 Register (Offset = 60100C18h) [Reset = 00000000h]

PASSWORD_1 is shown in Table 1-71.

Return to the Summary Table.

BSL access password - Word 1

Table 1-71. PASSWORD_1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	PASSWORD_1	R/W	0h	BSL access password - Word 1

66 PASSWORD_2 Register (Offset = 60100C1Ch) [Reset = 00000000h]

PASSWORD_2 is shown in Table 1-72.

Return to the Summary Table.

BSL access password - Word 2

Table 1-72. PASSWORD_2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	PASSWORD_2	R/W	0h	BSL access password - Word 2

67 PASSWORD_3 Register (Offset = 60100C20h) [Reset = 00000000h]

PASSWORD_3 is shown in Table 1-73.

Return to the Summary Table.

BSL access password - Word 3

Table 1-73. PASSWORD_3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	PASSWORD_3	R/W	0h	BSL access password - Word 3

68 PASSWORD_4 Register (Offset = 60100C24h) [Reset = 00000000h]

PASSWORD_4 is shown in Table 1-74.

Return to the Summary Table.

BSL access password - Word 4

Table 1-74. PASSWORD_4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	PASSWORD_4	R/W	0h	BSL access password - Word 4

69 PASSWORD_5 Register (Offset = 60100C28h) [Reset = 00000000h]

PASSWORD_5 is shown in Table 1-75.

Return to the Summary Table.

BSL access password - Word 5

Table 1-75. PASSWORD_5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	PASSWORD_5	R/W	0h	BSL access password - Word 5

70 PASSWORD_6 Register (Offset = 60100C2Ch) [Reset = 00000000h]

PASSWORD_6 is shown in Table 1-76.

Return to the Summary Table.

BSL access password - Word 6

Table 1-76. PASSWORD_6 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	PASSWORD_6	R/W	0h	BSL access password - Word 6

71 PASSWORD_7 Register (Offset = 60100C30h) [Reset = 00000000h]

PASSWORD_7 is shown in Table 1-77.

Return to the Summary Table.

BSL access password - Word 7

Table 1-77. PASSWORD_7 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	PASSWORD_7	R/W	0h	BSL access password - Word 7

72 APP_REV_POINTER Register (Offset = 60100C34h) [Reset = 00000000h]

APP_REV_POINTER is shown in Table 1-78.

Return to the Summary Table.

Pointer to application version information in MAIN flash

Table 1-78. APP_REV_POINTER Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	APP_REV_POINTER	R/W	0h	Pointer to application version information in MAIN flash

73 BSLCONFIG1 Register (Offset = 60100C38h) [Reset = 00000000h]

BSLCONFIG1 is shown in Table 1-79.

Return to the Summary Table.

Security alert response: Factory Reset/Disable BSL/Ignore; UART communication speed selection for ROM BSL

Table 1-79. BSLCONFIG1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	UART_BAUD_RATE	R/W	0h	UART communication speed selection for ROM BSL 1h = BAUDRATE_4800 configuration 2h = BAUDRATE_9600 configuration 3h = BAUDRATE_19200 configuration 4h = BAUDRATE_38400 configuration 5h = BAUDRATE_57600 configuration 6h = BAUDRATE_115200 configuration 7h = BAUDRATE_1000000 configuration 8h = BAUDRATE_2000000 configuration 9h = BAUDRATE_3000000 configuration
15-0	SECURITY_ALERT_LEVEL	R/W	0h	Security alert response: Factory Reset/Disable BSL/Ignore AABBh = FACTORY_RESET configuration CCDDh = DISABLE_BSL configuration FFFFh = DO_NOTHING configuration

74 I2C_SLAVE_ADDR Register (Offset = 60100C3Ch) [Reset = 00000000h]

I2C_SLAVE_ADDR is shown in Table 1-80.

Return to the Summary Table.

I2C slave address for ROM BSL I2C interface; Reserved field; Reserved field

Table 1-80. I2C_SLAVE_ADDR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	RESERVED	R	0h	Reserved field
23-16	RESERVED	R	0h	Reserved field
15-0	I2C_SLAVE_ADDR	R/W	0h	I2C slave address for ROM BSL I2C interface

1.5 Factory Constants

All devices include a memory-mapped FACTORY region which provides read-only data describing the capabilities of a device as well as any factory-provided trim information for use by application software.

Key data provided in the FACTORY memory region includes:

- The device unique 96-bit identity
- The default BSL pins
- The total MAIN region flash memory size (in KB)
- The total DATA region flash memory size (in KB), if present
- The flash bank count
- The total SRAM memory size (in KB)
- The temperature sensor calibration value
- The SYSPLL startup parameters

1.5.1 FLASH Registers

Table 1-81 lists the memory-mapped registers for the FLASH registers. All register offset addresses not listed in Table 1-81 should be considered as reserved locations and the register contents should not be modified.

Table 1-81. FLASH Registers

Offset	Acronym	Register Name	Section
60111000h	TRACEID	Defined by TI, during ATE, based on wafer	Go
60111004h	DEVICEID	This is the JTAGIDCODE that comes from the Ramp system	Go
60111008h	USERID	Defined by TI, depending on device spin	Go
6011100Ch	BSLPIN_UART	BSL UART Pin Configuration	Go
60111010h	BSLPIN_I2C	BSL I2C Pin Configuration	Go
60111014h	BSLPIN_CAN	BSL UART Pin Configuration	Go
60111018h	BSLPIN_INVOKE	BSL Pin Invocation Configuration	Go
6011101Ch	ROM_SRAMFLASH	Number of banks , sram size and flash size	Go
60111020h	PLLSTARTUP0_4_8MHZ		Go
60111024h	PLLSTARTUP1_4_8MHZ	System PLL Paramater 1 MMR --- Data from Flash Table Lookup	Go
60111028h	PLLSTARTUP0_8_16MHZ		Go
6011102Ch	PLLSTARTUP1_8_16MHZ	System PLL Paramater 1 MMR --- Data from Flash Table Lookup	Go
60111030h	PLLSTARTUP0_16_32MHZ		Go
60111034h	PLLSTARTUP1_16_32MHZ	System PLL Paramater 1 MMR --- Data from Flash Table Lookup	Go
60111038h	PLLSTARTUP0_32_48MHZ		Go
6011103Ch	PLLSTARTUP1_32_48MHZ	System PLL Paramater 1 MMR --- Data from Flash Table Lookup	Go
60111040h	TEMP_SENSE0	Temperature sensor room temperature calibration code. This is ADC conversion results of temperature sensor output voltage. Included in BOOTCRC calculation.	Go
60111044h	BOOTCRC	BOOTCRC records the 32-bit CRC of all locations in OPEN including reserved locations.	
60111048h	RESERVED00		
6011104Ch	ROM_VERSION		Go
60111050h	ROM_INM_VERSION		Go
60111054h	SYSPLLPARAM2		Go
60111058h	SYSPLLLDOCTL		Go
6011105Ch	SYSPLLLDOPROG		Go
60111060h	EXTREF_MP3	External Reference MP3 Temp Sense	Go
60111064h	INTREF_MP3	Internal Refernce Temp Sene	Go
60111068h	TSCODE_MP1	Tempsense Code MP1	Go
6011106Ch	TSCODE_MP3	Tempsense code MP3	Go
60111070h	TSTEMP	Temperature sense TEMP	Go
60111074h	SRAMFLASH	Number of banks , sram size and flash size	Go
60111078h	RESERVED07		
6011107Ch	RESERVED08		
80110120h	TRIMRANGE	Contains the address of the TRIMCRC location. _Boot code uses this to access TRIMCRC content for verification.	Go

ADVANCE INFORMATION

Table 1-81. FLASH Registers (continued)

Offset	Acronym	Register Name	Section
80110124h	GDR		Go

Complex bit access types are encoded to fit into small table cells. Table 1-82 shows the codes that are used for access types in this section.

Table 1-82. FLASH Access Type Codes

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

1 TRACEID Register (Offset = 60111000h) [Reset = 00000000h]

TRACEID is shown in Figure 1-2 and described in Table 1-83.

Return to the Summary Table.

unique per part shipped, done per established TI process

Figure 1-2. TRACEID Register

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

Table 1-84. TRACEID Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DATA	R	0h	

ADVANCE INFORMATION

2 DEVICEID Register (Offset = 60111004h) [Reset = 1BBB702Fh]

DEVICEID is shown in Figure 1-3 and described in Table 1-84.

Return to the Summary Table.

per Connectivity format, provisioned from RAMP and is die rev specific

Figure 1-3. DEVICEID Register

31	30	29	28	27	26	25	24
VERSION				PARTNUM			
R-1h				R-BBB7h			
23	22	21	20	19	18	17	16
PARTNUM							
R-BBB7h							
15	14	13	12	11	10	9	8
PARTNUM				MANUFACTURER			
R-BBB7h				R-17h			
7	6	5	4	3	2	1	0
MANUFACTURER							ALWAYS_1
R-17h							R-1h

Table 1-86. DEVICEID Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	VERSION	R	1h	Revision of the device. This field should change each time that the logic or mask set of the device is revised.
27-12	PARTNUM	R	BBB7h	Part number of the device.
11-1	MANUFACTURER	R	17h	TI's JEDEC bank and company code, which is: 00000010111b
0	ALWAYS_1	R	1h	This is always 1

ADVANCE INFORMATION

3 USERID Register (Offset = 60111008h) [Reset = 00000000h]

USERID is shown in Figure 1-4 and described in Table 1-85.

Return to the Summary Table.

per Connectivity format, defines the variant feature set

Figure 1-4. USERID Register

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

Table 1-88. USERID Register Field Descriptions

Bit	Field	Type	Reset	Description
31	START	R	1h	
30-28	MAJORREV	R	0h	Monotonic increasing value indicating a new revision of the SKU significant enough that users of the device may have to revise PCB or or software design
27-24	MINORREV	R	0h	Monotonic increasing value indicating a new revision of the SKU that preserves compatibility with lesser minorrev values. New capability may be introduced such that lesser minorrev numbers may not be compatible with greater if the new capability is used.
23-16	VARIANT	R	0h	Bit pattern uniquely identifying a variant of a part. This is used to indicate memory or package variations of the same part number. This number shall be selected at random among the remaining numbers for the relevant combination of IDCODE.device and USERCODE.part such that the order of creation cannot be inferred by the number. The variant number does not encode specifics of the variant directly.
15-0	PART	R	0h	Bit pattern that uniquely identifying a part. This is used to identify the specific part based on the die identified in DEVICEID.device. This number shall be selected at random among the remaining numbers for DEVICEID.device such that the order of creation cannot be inferred by the number. This value does not encode the part number directly.

4 BSLPIN_UART Register (Offset = 6011100Ch) [Reset = 00000000h]

BSLPIN_UART is shown in Figure 1-5 and described in Table 1-86.

Return to the Summary Table.

BSL UART Pin Configuration

Figure 1-5. BSLPIN_UART Register

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

Table 1-90. BSLPIN_UART Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	UART_TXD_PF	R	07h	UART TXD Pin Function Selection Value
23-16	UART_TXD_PAD	R	00h	UART TXD Pin used by BSL
15-8	UART_RXD_PF	R	07h	UART RXD Pin Function Selection Value
7-0	UART_RXD_PAD	R	01h	UART RXD Pad used by BSL

5 BSLPIN_I2C Register (Offset = 60111010h) [Reset = 00000000h]

BSLPIN_I2C is shown in Figure 1-6 and described in Table 1-87.

Return to the Summary Table.

BSL I2C Pin Configuration

Figure 1-6. BSLPIN_I2C Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
I2C_SCL_PF								I2C_SCL_PAD							
R-0h								R-0h							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
I2C_SDA_PF								I2C_SDA_PAD							
R-0h								R-0h							

Table 1-92. BSLPIN_I2C Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	I2C_SCL_PF	R	04h	I2C SCL Pin Function Selection Value
23-16	I2C_SCL_PAD	R	17h	I2C SCL Pin used by BSL
15-8	I2C_SDA_PF	R	4h	I2C SDA Pin Function Selection Value
7-0	I2C_SDA_PAD	R	16h	I2C SDA Pin used by BSL

6 BSLPIN_CAN Register (Offset = 60111014h) [Reset = 00000000h]

BSLPIN_CAN is shown in Figure 1-7 and described in Table 1-88.

Return to the Summary Table.

BSL UART Pin Configuration

Figure 1-7. BSLPIN_CAN Register

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

Table 1-94. BSLPIN_CAN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	CAN_TX_PF	R	0Ah	MCAN TX Pin Function Selection Value
23-16	CAN_TX_PAD	R	0Ch	MCAN TX Pin used by BSL
15-8	CAN_RX_PF	R	0Ah	MCAN RX Pin Function Selection Value
7-0	CAN_RX_PAD	R	0Bh	MCAN RX Pin used by BSL

7 BSLPIN_INVOKE Register (Offset = 60111018h) [Reset = 00000000h]

BSLPIN_INVOKE is shown in Figure 1-8 and described in Table 1-89.

Return to the Summary Table.

BSL Pin Invocation Configuration

Figure 1-8. BSLPIN_INVOKE Register

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

Table 1-96. BSLPIN_INVOKE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-15	RESERVED	R	0h	
14-13	GPIO_REG_SEL	R	AAh	GPIO Module Selection
12-8	GPIO_PIN_SEL	R	BBh	GPIO Pin Number in GPIO Module
7	GPIO_LEVEL	R	06h	GPIO Level Configuration for BSL Invocation
6-0	BSL_PAD	R	86h	BSL Invocation Pin Number

8 ROM_SRAMFLASH Register (Offset = 6011101Ch) [Reset = 00000000h]

ROM_SRAMFLASH is shown in Figure 1-9 and described in Table 1-90.

Return to the Summary Table.

Number of banks , sram size and flash size

Figure 1-9. ROM_SRAMFLASH Register

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

Table 1-98. ROM_SRAMFLASH Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	DATAFLASH_SZ	R	00h	The encoding of the field is that the value of the field is an integer to be interpreted as number of KBs. For eg: if the value of the field id 4, then it is 4KB, if the value is 32, then 32KB, and so on.
25-16	SRAM_SZ	R	020h	The encoding of the field is that the value of the field is an integer to be interpreted as number of KBs. For eg: if the value of the field id 4, then it is 4KB, if the value is 32, then 32KB, and so on.
15-14	RESERVED	R	0h	
13-12	MAINNUMBANKS	R	2h	Defines the number of main flash banks
11-0	MAINFLASH_SZ	R	200h	The encoding of the field is that the value of the field is an integer to be interpreted as number of KBs. For eg: if the value of the field id 4, then it is 4KB, if the value is 32, then 32KB, and so on.

9 PLLSTARTUP0_4_8MHZ Register (Offset = 60111020h) [Reset = 00000000h]

PLLSTARTUP0_4_8MHZ is shown in Figure 1-10 and described in Table 1-91.

Return to the Summary Table.

Figure 1-10. PLLSTARTUP0_4_8MHZ Register

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

Table 1-100. PLLSTARTUP0_4_8MHZ Register Field Descriptions

Bit	Field	Type	Reset	Description
31	CAPBOVERRIDE	R	1h	Override Enable For Cap B 0h = 0 1h = 1
30-29	RESERVED	R	0h	
28-24	CAPBVAL	R	1h	Override Value for Cap B
23-22	RESERVED	R	0h	
21-16	CPCURRENT	R	Ah	Charge Pump Current
15-14	RESERVED	R	0h	
13-8	STARTTIMELP	R	16h	Startup time from Low Power Exit to Locked Clock in resolution of 1usec
7-6	RESERVED	R	0h	
5-0	STARTTIME	R	16h	Startup time from Enable to Locked Clock in resolution of 1usec

10 PLLSTARTUP1_4_8MHZ Register (Offset = 60111024h) [Reset = 00000000h]

PLLSTARTUP1_4_8MHZ is shown in Figure 1-11 and described in Table 1-92.

Return to the Summary Table.

Figure 1-11. PLLSTARTUP1_4_8MHZ Register

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

Table 1-102. PLLSTARTUP1_4_8MHZ Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	LPFRESC	R	FFh	Loop Filter Res C
23-18	RESERVED	R	0h	
17-8	LPFRESA	R	4h	Loop Filter Res A
7-5	RESERVED	R	0h	
4-0	LPFCAPA	R	1Fh	Loop Filter Cap A

11 PLLSTARTUP0_8_16MHZ Register (Offset = 60111028h) [Reset = 00000000h]

PLLSTARTUP0_8_16MHZ is shown in Figure 1-12 and described in Table 1-93.

Return to the Summary Table.

Figure 1-12. PLLSTARTUP0_8_16MHZ Register

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

Table 1-104. PLLSTARTUP0_8_16MHZ Register Field Descriptions

Bit	Field	Type	Reset	Description
31	CAPBOVERRIDE	R	1h	Override Enable For Cap B 0h = 0 1h = 1
30-29	RESERVED	R	0h	
28-24	CAPBVAL	R	1h	Override Value for Cap B
23-22	RESERVED	R	0h	
21-16	CPCURRENT	R	Ah	Charge Pump Current
15-14	RESERVED	R	0h	
13-8	STARTTIMELP	R	16h	Startup time from Low Power Exit to Locked Clock in resolution of 1usec
7-6	RESERVED	R	0h	
5-0	STARTTIME	R	16h	Startup time from Enable to Locked Clock in resolution of 1usec

12 PLLSTARTUP1_8_16MHZ Register (Offset = 6011102Ch) [Reset = 00000000h]

PLLSTARTUP1_8_16MHZ is shown in Figure 1-13 and described in Table 1-94.

Return to the Summary Table.

Figure 1-13. PLLSTARTUP1_8_16MHZ Register

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

Table 1-106. PLLSTARTUP1_8_16MHZ Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	LPFRESC	R	FFh	Loop Filter Res C
23-18	RESERVED	R	0h	
17-8	LPFRESA	R	4h	Loop Filter Res A
7-5	RESERVED	R	0h	
4-0	LPFCAPA	R	Fh	Loop Filter Cap A

13 PLLSTARTUP0_16_32MHZ Register (Offset = 60111030h) [Reset = 00000000h]

PLLSTARTUP0_16_32MHZ is shown in Figure 1-14 and described in Table 1-95.

Return to the Summary Table.

Figure 1-14. PLLSTARTUP0_16_32MHZ Register

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

Table 1-108. PLLSTARTUP0_16_32MHZ Register Field Descriptions

Bit	Field	Type	Reset	Description
31	CAPBOVERRIDE	R	1h	Override Enable For Cap B 0h = 0 1h = 1
30-29	RESERVED	R	0h	
28-24	CAPBVAL	R	1h	Override Value for Cap B
23-22	RESERVED	R	0h	
21-16	CPCURRENT	R	Ah	Charge Pump Current
15-14	RESERVED	R	0h	
13-8	STARTTIMELP	R	16h	Startup time from Low Power Exit to Locked Clock in resolution of 1usec
7-6	RESERVED	R	0h	
5-0	STARTTIME	R	16h	Startup time from Enable to Locked Clock in resolution of 1usec

14 PLLSTARTUP1_16_32MHZ Register (Offset = 60111034h) [Reset = 00000000h]

PLLSTARTUP1_16_32MHZ is shown in Figure 1-15 and described in Table 1-96.

Return to the Summary Table.

Figure 1-15. PLLSTARTUP1_16_32MHZ Register

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

Table 1-110. PLLSTARTUP1_16_32MHZ Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	LPFRESC	R	FFh	Loop Filter Res C
23-18	RESERVED	R	0h	
17-8	LPFRESA	R	4h	Loop Filter Res A
7-5	RESERVED	R	0h	
4-0	LPFCAPA	R	Fh	Loop Filter Cap A

15 PLLSTARTUP0_32_48MHZ Register (Offset = 60111038h) [Reset = 00000000h]

PLLSTARTUP0_32_48MHZ is shown in Figure 1-16 and described in Table 1-97.

Return to the Summary Table.

Figure 1-16. PLLSTARTUP0_32_48MHZ Register

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

Table 1-112. PLLSTARTUP0_32_48MHZ Register Field Descriptions

Bit	Field	Type	Reset	Description
31	CAPBOVERRIDE	R	1h	Override Enable For Cap B 0h = 0 1h = 1
30-29	RESERVED	R	0h	
28-24	CAPBVAL	R	1h	Override Value for Cap B
23-22	RESERVED	R	0h	
21-16	CPCURRENT	R	Ah	Charge Pump Current
15-14	RESERVED	R	0h	
13-8	STARTTIMELP	R	16h	Startup time from Low Power Exit to Locked Clock in resolution of 1usec
7-6	RESERVED	R	0h	
5-0	STARTTIME	R	16h	Startup time from Enable to Locked Clock in resolution of 1usec

16 PLLSTARTUP1_32_48MHZ Register (Offset = 6011103Ch) [Reset = 00000000h]

PLLSTARTUP1_32_48MHZ is shown in Figure 1-17 and described in Table 1-98.

Return to the Summary Table.

Figure 1-17. PLLSTARTUP1_32_48MHZ Register

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

Table 1-114. PLLSTARTUP1_32_48MHZ Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	LPFRESC	R	FFh	Loop Filter Res C
23-18	RESERVED	R	0h	
17-8	LPFRESA	R	4h	Loop Filter Res A
7-5	RESERVED	R	0h	
4-0	LPFCAPA	R	Fh	Loop Filter Cap A

17 TEMP_SENSE0 Register (Offset = 60111040h) [Reset = 00000000h]

TEMP_SENSE0 is shown in Figure 1-18 and described in Table 1-99.

Return to the Summary Table.

Temperature sensor room temperature calibration code. This is ADC conversion results of temperature sensor output voltage. Included in BOOTCRC calculation.

Figure 1-18. TEMP_SENSE0 Register

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

Table 1-116. TEMP_SENSE0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DATA	R	0h	

ADVANCE INFORMATION

18 BOOTCRC Register (Offset = 60111044h) [Reset = 00000000h]

BOOTCRC is shown in Figure 1-19 and described in Table 1-100.

Return to the Summary Table.

BOOTCRC records the 32-bit CRC of all locations in OPEN including reserved locations.

Figure 1-19. BOOTCRC Register

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

Table 1-118. BOOTCRC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DATA	R	0h	

ADVANCE INFORMATION

19 ROM_VERSION Register (Offset = 6011104Ch) [Reset = 01010000h]

ROM_VERSION is shown in Figure 1-20 and described in Table 1-101.

Return to the Summary Table.

Figure 1-20. ROM_VERSION Register

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

Table 1-120. ROM_VERSION Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	ROM_PG_VER	R	01h	
23-16	ROM_MAJOR_VER	R	01h	
15-8	ROM_MINOR_VER	R	00h	
7-0	ROM_PATCH_VER	R	00h	

20 ROM_INM_VERSION Register (Offset = 60111050h) [Reset = 01010000h]

ROM_INM_VERSION is shown in Figure 1-21 and described in Table 1-102.

Return to the Summary Table.

Figure 1-21. ROM_INM_VERSION Register

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

Table 1-122. ROM_INM_VERSION Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	ROM_INM_PG_VER	R	01h	
23-16	ROM_INM_MAJOR_VER	R	01h	
15-8	ROM_INM_MINOR_VER	R	00h	
7-0	ROM_INM_PATCH_VER	R	00h	

ADVANCE INFORMATION

21 SYSPLLPARAM2 Register (Offset = 60111054h) [Reset = 00000000h]

SYSPLLPARAM2 is shown in Figure 1-22 and described in Table 1-103.

Return to the Summary Table.

Figure 1-22. SYSPLLPARAM2 Register

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

Table 1-124. SYSPLLPARAM2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R	0h	
3	RNGIXVCOIBIASCFG	R	1h	
2	RESERVED	R	0h	
1-0	LPFCAPC	R	00h	

22 SYSPLLLDOCTL Register (Offset = 60111058h) [Reset = 00000000h]

SYSPLLLDOCTL is shown in Figure 1-23 and described in Table 1-104.

Return to the Summary Table.

Figure 1-23. SYSPLLLDOCTL Register

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

Table 1-126. SYSPLLLDOCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	
15-0	LDOCTLLOWV	R	00h	

23 SYSPLLLDOPROG Register (Offset = 6011105Ch) [Reset = 00000000h]

SYSPLLLDOPROG is shown in Figure 1-24 and described in Table 1-105.

Return to the Summary Table.

Figure 1-24. SYSPLLLDOPROG Register

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

Table 1-128. SYSPLLLDOPROG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R	0h	
2-0	LDOVOUTPROGLOWV	R	4h	

24 EXTREF_MP3 Register (Offset = 60111060h) [Reset = 00000000h]

EXTREF_MP3 is shown in Figure 1-25 and described in Table 1-106.

Return to the Summary Table.

External Reference MP3 Temp Sense

Figure 1-25. EXTREF_MP3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TSOFFSET_EXTREF_MP3															
R-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TSGAIN_EXTREF_MP3															
R-0h															

Table 1-130. EXTREF_MP3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	TSOFFSET_EXTREF_MP3	R	0h	Temperature sensor offset is recorded here
15-0	TSGAIN_EXTREF_MP3	R	0h	Temperature sensor gain is recorded here

25 INTREF_MP3 Register (Offset = 60111064h) [Reset = 00000000h]

INTREF_MP3 is shown in Figure 1-26 and described in Table 1-107.

Return to the Summary Table.

Internal Refernce Temp Sene

Figure 1-26. INTREF_MP3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TSOFFSET_INTREF_MP3															
R-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TSGAIN_INTREF_MP3															
R-0h															

Table 1-132. INTREF_MP3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	TSOFFSET_INTREF_MP3	R	0h	Temperature sensor offset is recorded here
15-0	TSGAIN_INTREF_MP3	R	0h	Temperature sensor gain is recorded here

26 TSCODE_MP1 Register (Offset = 60111068h) [Reset = 00000000h]

TSCODE_MP1 is shown in Figure 1-27 and described in Table 1-108.

Return to the Summary Table.

Tempsense Code MP1

Figure 1-27. TSCODE_MP1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TSCODE_INTREF_MP1															
R-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TSCODE_EXTREF_MP1															
R-0h															

Table 1-134. TSCODE_MP1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	TSCODE_INTREF_MP1	R	0h	Temperature sensor value at 30C stored at this location
15-0	TSCODE_EXTREF_MP1	R	0h	Temperature sensor value at 30C stored at this location

27 TSCODE_MP3 Register (Offset = 6011106Ch) [Reset = 00000000h]

TSCODE_MP3 is shown in Figure 1-28 and described in Table 1-109.

Return to the Summary Table.

Tempsense code MP3

Figure 1-28. TSCODE_MP3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TSCODE_INTREF_MP3															
R-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TSCODE_EXTREF_MP3															
R-0h															

Table 1-136. TSCODE_MP3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	TSCODE_INTREF_MP3	R	0h	Temperature sensor value at 125C stored at this location
15-0	TSCODE_EXTREF_MP3	R	0h	Temperature sensor value at 125C stored at this location

28 TSTEMP Register (Offset = 60111070h) [Reset = 00000000h]

TSTEMP is shown in Figure 1-29 and described in Table 1-110.

Return to the Summary Table.

Temperature sense TEMP

Figure 1-29. TSTEMP Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TSTEMP_MP3																TSTEMP_MP1															
R-0h																R-0h															

Table 1-138. TSTEMP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	TSTEMP_MP3	R	0h	Test temperature (125C) is recorded here
15-0	TSTEMP_MP1	R	0h	Test temperature (30C) is recorded here

29 SRAMFLASH Register (Offset = 60111074h) [Reset = 00000000h]

SRAMFLASH is shown in Figure 1-30 and described in Table 1-111.

Return to the Summary Table.

Number of banks , sram size and flash size

Figure 1-30. SRAMFLASH Register

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

Table 1-140. SRAMFLASH Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	DATAFLASH_SZ	R	00h	The encoding of the field is that the value of the field is an integer to be interpreted as number of KBs. For eg: if the value of the field id 4, then it is 4KB, if the value is 32, then 32KB, and so on.
25-16	SRAM_SZ	R	80h	The encoding of the field is that the value of the field is an integer to be interpreted as number of KBs.
15-14	RESERVED	R	0h	
13-12	MAINNUMBANKS	R	2h	Defines the number of main flash banks
11-0	MAINFLASH_SZ	R	200h	For eg: if the value of the field id 4, then it is 4KB, if the value is 32, then 32KB, and so on.

ADVANCE INFORMATION

30 TRIMRANGE Register (Offset = 80110120h) [Reset = 00000000h]

TRIMRANGE is shown in Figure 1-31 and described in Table 1-112.

Return to the Summary Table.

Contains the address of the TRIMCRC location. _Boot code uses this to access TRIMCRC content for verification.

Figure 1-31. TRIMRANGE Register

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

Table 1-142. TRIMRANGE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	TRIM_END	R	012Ch	Lower 16 bits of TRIM end address. Upper 16 bits are assumed to be 0x8011.
15-0	TRIM_START	R	0238h	Lower 16 bits of TRIM start address. Upper 16 bits are assumed to be 0x8011.

ADVANCE INFORMATION

31 GDR Register (Offset = 80110124h) [Reset = 00000000h]

GDR is shown in Figure 1-32 and described in Table 1-113.

Return to the Summary Table.

Figure 1-32. GDR Register

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

Table 1-144. GDR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	VERSION_FT	R	0h	
23-16	VERSION_PROBE	R	0h	
15-8	TEST_RECORD_FT	R	FFh	
7-0	TEST_RECORD_MP	R	FFh	

1.6 Memory Configuration

On AM13E230x devices, the MEMCFG registers control the configurations of the RAM and ROM. Some of the key features of the MEMCFG registers are:

1. Configuring SRAM for write access based on ECC or data bits
2. Configuring Wait-States on SRAM based on the SRAM bank
3. Configuring Wait-States on ROM

1.6.1 MEMCFG Registers

This Section describes the MEMCFG Registers.

1.6.1.1 MEMCFG Base Address Table

Table 1-145. MEMCFG Base Address Table

Bit Field Name		DriverLib Name	Base Address
Instance	Structure		
MemcfgRegs	MEM_CFG_REGS	MEMCFG	0x4002_A000

ADVANCE INFORMATION

1.6.1.2 MEM_CFG_REGS Registers

Table 1-115 lists the memory-mapped registers for the MEM_CFG_REGS registers. All register offset addresses not listed in Table 1-115 should be considered as reserved locations and the register contents should not be modified.

Table 1-146. MEM_CFG_REGS Registers

Offset	Acronym	Register Name	Write Protection	Section
1000h	TEST	RAM TEST Register		Go
1010h	ROM_WS_CONFIG	ROM wait state configuration		Go
1014h	TMUROM_TEST	ROM wait state configuration		Go

Complex bit access types are encoded to fit into small table cells. Table 1-116 shows the codes that are used for access types in this section.

Table 1-147. MEM_CFG_REGS Access Type Codes

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

1 TEST Register (Offset = 1000h) [Reset = 00000000h]

TEST is shown in Figure 1-33 and described in Table 1-117.

Return to the Summary Table.

RAM TEST Register

Figure 1-33. TEST Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
TEST_SRAM3		TEST_SRAM2		TEST_SRAM1		TEST_SRAM0	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	

Table 1-149. TEST Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R-0	0h	Reserved
7-6	TEST_SRAM3	R/W	0h	Selects the defferent modes for SRAM3: 00: Functional Mode. 01: Writes are allowed to data bits only. No write to ECC bits. 10: Writes are allowed to ECC bits only. No write to data bits. 11: Same as functional mode Reset type: SYSRSn
5-4	TEST_SRAM2	R/W	0h	Selects the defferent modes for SRAM2: 00: Functional Mode. 01: Writes are allowed to data bits only. No write to ECC bits. 10: Writes are allowed to ECC bits only. No write to data bits. 11: Same as functional mode Reset type: SYSRSn
3-2	TEST_SRAM1	R/W	0h	Selects the defferent modes for SRAM1: 00: Functional Mode. 01: Writes are allowed to data bits only. No write to ECC bits. 10: Writes are allowed to ECC bits only. No write to data bits. 11: Same as functional mode Reset type: SYSRSn
1-0	TEST_SRAM0	R/W	0h	Selects the defferent modes for SRAM0: 00: Functional Mode. 01: Writes are allowed to data bits only. No write to ECC bits. 10: Writes are allowed to ECC bits only. No write to data bits. 11: Same as functional mode Reset type: SYSRSn

ADVANCE INFORMATION

2 ROM_WS_CONFIG Register (Offset = 1010h) [Reset = 00000000h]

ROM_WS_CONFIG is shown in Figure 1-34 and described in Table 1-118.

Return to the Summary Table.

ROM wait state configuration

Figure 1-34. ROM_WS_CONFIG Register

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

Table 1-151. ROM_WS_CONFIG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R-0	0h	Reserved
0	ROM_WS_ENABLE	R/W	0h	Wait state configuration ROM: 0: 1wait state disable. 1: 1 wait enable. Reset type: SYSRSn

ADVANCE INFORMATION

3 TMUROM_TEST Register (Offset = 1014h) [Reset = 00000005h]

TMUROM_TEST is shown in Figure 1-35 and described in Table 1-119.

Return to the Summary Table.

TMUROM TEST Register

Figure 1-35. TMUROM_TEST Register

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

Table 1-153. TMUROM_TEST Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R-0	0h	Reserved
3-0	TMUROM_PAR_FORCE	R/W	5h	TMUROM Parity Error Force: 0xA: Error forced Others: No error forced Reset type: SYSRSn

ADVANCE INFORMATION

Table 2-1. Peripheral Registers Memory Map

Structure	DriverLib Name	Base Address
MCLK/2 Domain		
EPI_REGS_GPCFG, EPI_REGS_SDRAMCFG, EPI_REGS_HB8CFG, EPI_REGS_HB16CFG	EPI0_BASE, EPI0SDRAM_BASE, PI0HB8_BASE, EPI0HB16_BASE	0x4001_A000
PGA_REGS	PGA0_BASE	0x400F_C000
PGA_REGS	PGA1_BASE	0x400F_D000
PGA_REGS	PGA2_BASE	0x400F_E000
MCAN_REGS	MCAN0_BASE	0x4011_0000
TIMG4_REGS	TIMG4_BASE	0x4018_0000
TIMG12_REGS	TIMG12_BASE	0x4018_8000
AES_REGS	AES_BASE	0x401B_0000
CRCP_REGS	CRC_BASE	0x401B_2000
KEYSTORE_REGS	KEYSTORE_BASE	0x401B_6000
UNICOMMUART_REGS	UC0_UART_BASE	0x4060_0000
UNICOMMUART_REGS	UC1_UART_BASE	0x4060_1000
UNICOMMUART_REGS	UC2_UART_BASE	0x4060_2000
UNICOMMI2CC_REGS	UC0_I2CC_BASE	0x4060_8000
UNICOMMI2CC_REGS	UC1_I2CC_BASE	0x4060_9000
UNICOMMI2CC_REGS	UC2_I2CC_BASE	0x4060_A000
UNICOMMI2CT_REGS	UC0_I2CT_BASE	0x4061_0000
UNICOMMI2CT_REGS	UC1_I2CT_BASE	0x4061_1000
UNICOMMI2CT_REGS	UC2_I2CT_BASE	0x4061_2000
UNICOMMSPI_REGS	UC0_SPI_BASE	0x4061_8000
UNICOMMSPI_REGS	UC1_SPI_BASE	0x4061_9000
UNICOMM_REGS	UNICOMM0_BASE	0x4063_0000
UNICOMM_REGS	UNICOMM1_BASE	0x4063_2000
UNICOMM_REGS	UNICOMM2_BASE	0x4063_4000
SPG_REGS	SPG0_BASE	0x4063_F000
UNICOMMUART_REGS	UC3_UART_BASE	0x4064_0000
UNICOMMUART_REGS	UC4_UART_BASE	0x4064_1000
UNICOMMUART_REGS	UC5_UART_BASE	0x4064_2000
UNICOMMI2CC_REGS	UC3_I2CC_BASE	0x4064_8000
UNICOMMI2CC_REGS	UC4_I2CC_BASE	0x4064_9000
UNICOMMI2CC_REGS	UC5_I2CC_BASE	0x4064_A000
UNICOMMI2CT_REGS	UC3_I2CT_BASE	0x4065_0000
UNICOMMI2CT_REGS	UC4_I2CT_BASE	0x4065_1000
UNICOMMI2CT_REGS	UC5_I2CT_BASE	0x4065_2000
UNICOMMSPI_REGS	UC3_SPI_BASE	0x4065_8000
UNICOMMSPI_REGS	UC4_SPI_BASE	0x4065_9000

Table 2-1. Peripheral Registers Memory Map (continued)

Structure	DriverLib Name	Base Address
UNICOMM_REGS	UNICOMM3_BASE	0x4067_0000
UNICOMM_REGS	UNICOMM4_BASE	0x4067_2000
UNICOMM_REGS	UNICOMM5_BASE	0x4067_4000
SPG_REGS	SPG1_BASE	0x4067_F000
MCLK/1 Domain		
ADC_LITE_REGS	ADC0_BASE	0x4000_0000
ADC_LITE_REGS	ADC1_BASE	0x4000_2000
ADC_LITE_REGS	ADC2_BASE	0x4000_4000
ADC_LITE_RESULT_REGS	ADC0RESULT_BASE	0x4000_A000
ADC_LITE_RESULT_REGS	ADC1RESULT_BASE	0x4000_B000
ADC_LITE_RESULT_REGS	ADC2RESULT_BASE	0x4000_C000
MCPWM_6CH_REGS	MCPWM0_BASE	0x4001_0000
MCPWM_6CH_REGS	MCPWM1_BASE	0x4001_1000
MCPWM_6CH_REGS	MCPWM2_BASE	0x4001_2000
MCPWM_6CH_REGS	MCPWM3_BASE	0x4001_3000
MCPWM_6CH_REGS	MCPWM4_BASE	0x4001_4000
DMA_REGS	DMA0_BASE	0x4002_0000
FLASH_CTRL_REGS	FLASH_BASE	0x4002_8000
MEM_CFG_REGS	MEMCFG_BASE	0x4002_A000
EAM_REGS	EAM_BASE	0x4002_C000
ECAP_REGS	ECAP0_BASE	0x4044_0000
ECAP_REGS	ECAP1_BASE	0x4044_1000
EQEP_REGS	EQEP0_BASE	0x4044_8000
EQEP_REGS	EQEP1_BASE	0x4044_9000
EQEP_REGS	EQEP2_BASE	0x4044_A000
CMPSS_LITE_REGS	CMPSS0_BASE	0x4046_0000
CMPSS_LITE_REGS	CMPSS1_BASE	0x4046_1000
CMPSS_LITE_REGS	CMPSS2_BASE	0x4046_2000
CMPSS_LITE_REGS	CMPSS3_BASE	0x4046_3000
INPUT_XBAR_REGS	INPUTXBAR_BASE	0x4046_8000
EPWM_XBAR_REGS	PWMXBAR_BASE	0x4046_9000
OUTPUTXBAR_REGS	OUTPUTXBAR_BASE	0x4046_A000
SYNC_SOC_REGS	SYNC_BASE	0x4046_B000
OUTPUTXBAR_FLAG_REGS	OUTPUTXBAR0_FLAGS_BASE	0x4047_0000
OUTPUTXBAR_FLAG_REGS	OUTPUTXBAR1_FLAGS_BASE	0x4047_1000
OUTPUTXBAR_FLAG_REGS	OUTPUTXBAR2_FLAGS_BASE	0x4047_2000
OUTPUTXBAR_FLAG_REGS	OUTPUTXBAR3_FLAGS_BASE	0x4047_3000
OUTPUTXBAR_FLAG_REGS	OUTPUTXBAR4_FLAGS_BASE	0x4047_4000
OUTPUTXBAR_FLAG_REGS	OUTPUTXBAR5_FLAGS_BASE	0x4047_5000
OUTPUTXBAR_FLAG_REGS	OUTPUTXBAR6_FLAGS_BASE	0x4047_6000
OUTPUTXBAR_FLAG_REGS	OUTPUTXBAR7_FLAGS_BASE	0x4047_7000
INPUT_FLAG_XBAR_REGS	INPUTXBAR_FLAGS_BASE	0x4049_0000
Power Domain 0 (Always ON)		
SYSCTL_REGS	SYSCTL_BASE	0x400A_F000
DEBUGSS_REGS	DEBUGSS_BASE	0x400C_7000
IOMUX_REGS	IOMUX_BASE	0x400C_C000
WWDT_REGS	WWDT_BASE	0x400D_0000
GPIO_REGS	GPIO0_BASE	0x400F_0000
GPIO_REGS	GPIO1_BASE	0x400F_2000

Table 2-1. Peripheral Registers Memory Map (continued)

Structure	DriverLib Name	Base Address
GPIO_REGS	GPIO2_BASE	0x400F_4000
GPIO_REGS	GPIO3_BASE	0x400F_6000
MCLK/4 Domain		
NVMNW_REGS	NVMNW_BASE	0x4004_2000
GSC_REGS	GSC_BASE	0x4004_6000

Chapter 3

Power Management and Clock Unit (PMCU)

The power management and clock unit (PMCU) is a unified system module which provides all power management, clock configuration, and reset control functionality for the device. All power management unit (PMU) and clock module (CKM) policies for device operation are configured through memory-mapped registers in the system controller (SYSCTL).

3.1 PMCU Overview

The power management and clock unit (PMCU) provides all power, clocking, reset, and system control services for the device. The PMCU contains three submodules to provide this functionality: the power management unit (**PMU**), the clock module (**CKM**), and the system controller (**SYSCTL**).

The PMU is an analog submodule that generates the internal regulated supplies for the device and supervises the condition of the external supply. The PMU also contains voltage and current reference circuits used by the on-chip regulators and analog peripherals.

The CKM is an analog submodule that provides clock sources (internal and external oscillators) and presents these clock sources to SYSCTL. SYSCTL distributes these clock sources to the CPU, buses, and peripherals on the device.

The SYSCTL is a digital submodule that provides the control logic for all functions in the PMCU. In addition, SYSCTL contains the memory-mapped registers used by software to configure power management and clocks, assess the status of the device, and control resets. SYSCTL also provides general-purpose memory that is retained in SHUTDOWN mode and can be used to store status information in SHUTDOWN mode when SRAM and register contents are lost.

Figure 3-1 shows the interfaces between the PMCU and the device supplies, clocks, and signals. Configuration of the PMCU by software is always done through memory-mapped registers in the SYSCTL submodule.

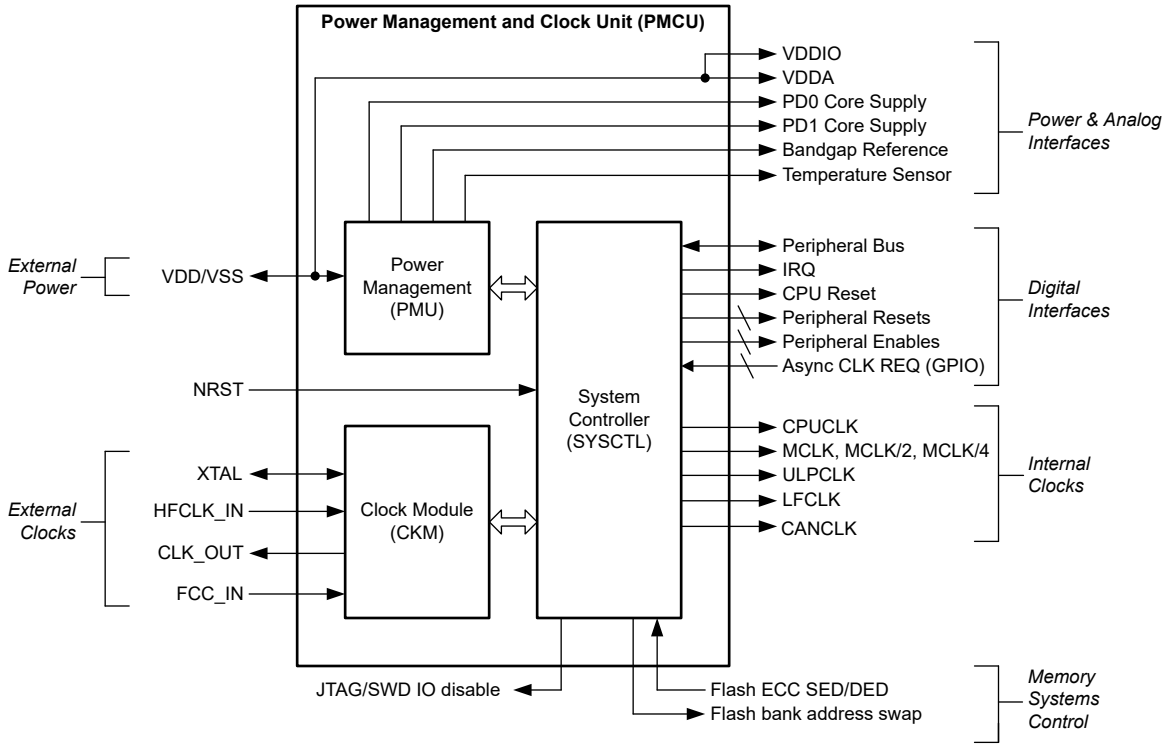


Figure 3-1. PMCU Top-Level Diagram

ADVANCE INFORMATION

Using this Guide

The PMU, CKM, and SYSCTL sections of this chapter describe the functionality provided by each submodule in detail.

The quick start section describes overall system level operation of the PMCU and how to configure the PMCU for different application scenarios.

3.1.1 Power Domains

Two core power domains are provided on the device: PD1 and PD0. PD1 is always powered in RUN and SLEEP modes and switched off in STOP and STANDBY modes. PD0 is always powered in RUN, SLEEP, STOP and STANDBY modes. PD1 and PD0 are both disabled in SHUTDOWN mode.

Table 3-1. Power Domains Summary

Power Mode	PD1	PD1 Supply	PD0	PD0 Supply	LDO Switch
RUN	ON	MAIN LDO	ON	MAIN LDO	CLOSED
SLEEP	ON (CPU clocks gated)	MAIN LDO	ON	MAIN LDO	CLOSED
STOP	OFF	STOP LDO (retention logic/memory)	ON	STOP LDO	OPEN
STANDBY	OFF	STOP LDO (retention logic/memory)	ON	STOP LDO	OPEN
SHUTDOWN	OFF	N/A	OFF	N/A	N/A

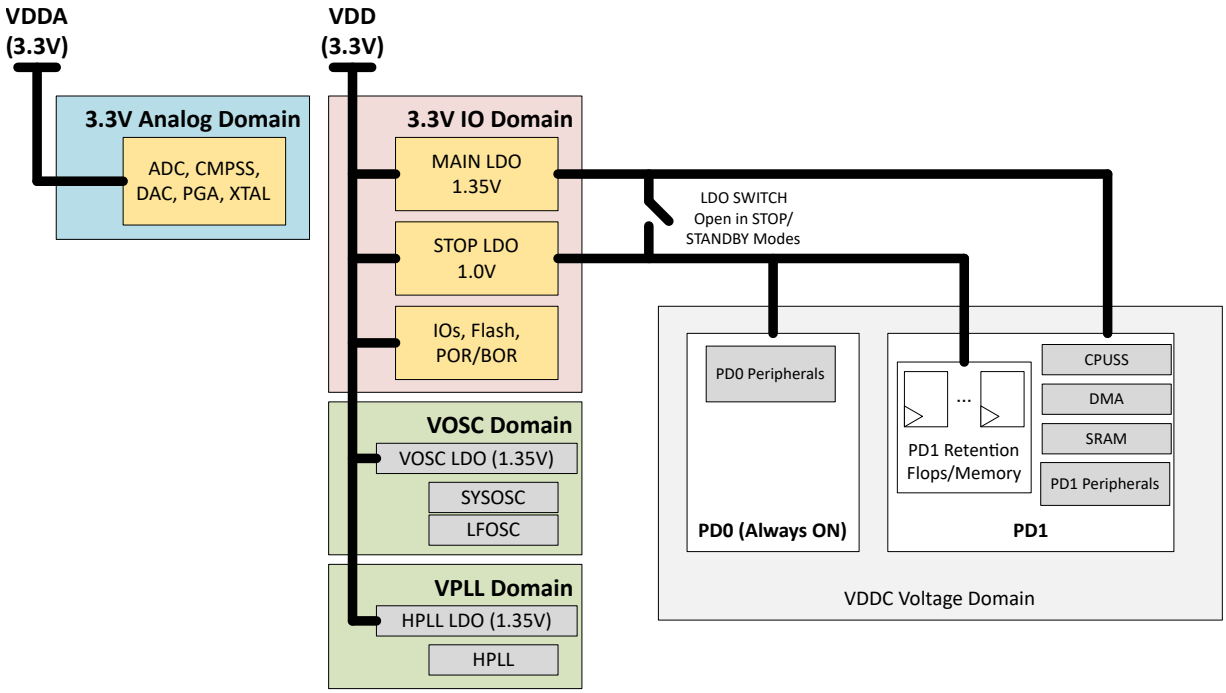


Figure 3-2. Power Hierarchy Block Diagram

- The PD1 domain includes the CPU subsystem, DMA, SRAM, and PD1 peripheral buses: Fast Peripheral Interconnect, Peripheral Interconnect, and CPU Interconnect which run from MCLK, MCLK/2, and MCLK/4, respectively.
- The PD0 domain includes the PD0 peripherals and PD0 bus segment (PD0 Interconnect) which runs at MCLK/4 in RUN and SLEEP mode, 4MHz in STOP mode, and 32kHz in STANDBY mode. This clock source, derived from MCLK/4 is referred to as ULPCLK. The PD0 domain is powered in all modes except SHUTDOWN and can be thought of as an "always-on" domain.
- The VDDC domain includes both PD1 and PD0

Note

For AM13E23x devices, MCLK=200MHz.

- The VDD domain refers to the single external power supply that provides power to the IO and analog peripherals.

Note

For AM13E23x devices, VDD=3.3V

The *Module Allocation and Instances* table describes which peripherals on a device are in PD1 and which are in PD0.

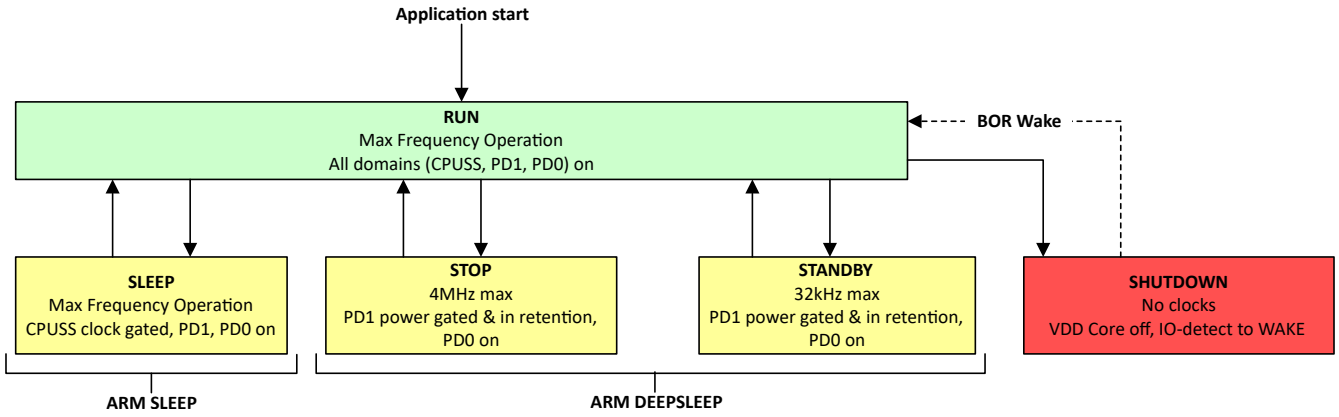
The *Power Supply* section describes the individual power supplies on the device.

ADVANCE INFORMATION

3.1.2 Operating Modes

Various operating modes (power modes) are provided to allow for optimization of the device power consumption based on application requirements. In order of decreasing power, the modes are: RUN, SLEEP, STOP, STANDBY and SHUTDOWN. The following figure shows the interaction between the modes.

Figure 3-3. Operating Power Modes



Note

For AM13E23x devices, the Max Frequency Operation is 200MHz (MCLK).

Supported Functionality by Operating Mode details what functions are available in each operating mode of the device. See the Operating Mode Selection section for information on how to configure the device for a particular operating mode.

Operating Mode Concept

The MCU implements a policy-based power and clock management scheme. Policies can be configured through application software to manage the clock in each operating mode to obtain the best balance of power and performance for a given application.

After the operating policy for each mode is configured, application software can enter and exit the various operating modes through simple register commands, and SYSCTL automatically manages all the necessary PMU states, oscillator and clock enable and disables.

A variety of hardware-triggered low-power mode suspension mechanisms also exist to enable on-demand access to a fast clock when requested by supported peripherals in STOP mode. All wakeups bring the device to RUN mode.

The policy-driven operating mode scheme enables application software to select the operating mode and corresponding policy that provide the lowest possible power consumption for background activities, bringing up a fast clock, or bringing the device to RUN (in the case of an IRQ) for burst handling.

3.1.2.1 RUN Mode

In RUN mode, the CPU is actively executing code and any peripheral can be enabled. In this mode, the MCLK and CPUCLK run from a fast clock source (SYSOSC, HFCLK, or SYSPLL). The VDDC domain is supplied by the on-chip MAIN LDO.

All functions are available at the highest frequency in this mode. If needed, software can be configured to a lower PLL frequency or lower system clock frequency to save dynamic power. For more information on configuring the PLL frequency, refer to the *System Phase-Locked Loop (SYSPLL)* Section.

3.1.2.2 SLEEP Mode

In SLEEP mode, the CPU is disabled (clock gated). Otherwise, the device configuration is the same as RUN.

3.1.2.3 STOP Mode

In STOP mode, the PD1 power domain is switched off and CPU, DMA, SRAM, FLASH, PLL, XTAL, and PD1 peripherals are disabled and kept in retention mode. The VDDC domain voltage is reduced to 10% as the active PD0 peripherals operate at 4MHz, supplied by SYSOSC appropriately divided to get the lower clock output.

During STOP mode operation, the SYSPLL/XTAL/HFCLKIN are switched off.

3.1.2.4 STANDBY Mode

In STANDBY mode, the PD1 power domain is switched off and CPU, DMA, SRAM, FLASH, PLL, XTAL, and PD1 peripherals are disabled and kept in retention mode. VDDC supply voltage is reduced to 10% and active PD0 peripherals operate at a maximum frequency of 32KHz, supplied by LFOSC. High-speed oscillators such as SYSPLL, XTAL, HFCLK_IN and SYSOSC are disabled. A GPIO toggle can wake the device from this mode.

There are 2 policy options for STANDBY mode: STANDBY0 and STANDBY1.

- **STANDBY0**: All PD0 peripherals receive the ULPCLK and LFCLK. A GPIO toggle can synchronously wakeup the device.
- **STANDBY1**: This is the lowest device power mode. PD0 peripherals (such as GPIO) can wake the system upon an external event through an asynchronous fast clock request, but PD0 peripherals are not actively clocked in this policy option.

3.1.2.5 SHUTDOWN Mode

In SHUTDOWN mode, core clocks are not available. The core regulator is completely disabled and all SRAM and register contents are lost, with the exception of the general-purpose memory in SYSCTL that can be used to store state information. The BOR and bandgap circuit are disabled.

Specific IO configuration settings are retained in the SHUTDOWN operating mode.

- Drive strength
- Pullup enable
- Pulldown enable
- Input enable
- Output enable
- Output state

The device can wake through a wake-up capable IO, a debug connection, or NRST.

SHUTDOWN mode has the lowest current consumption of any operating mode. Exiting SHUTDOWN mode triggers a BOR.

3.1.2.6 Supported Functionality by Operating Mode

The table below summarizes the behavior of different peripheral instances in each operating mode:

Table Legend

- **EN:** The function is enabled in the specified mode
- **DIS:** The function is disabled (either clock or power gated) in the specified mode, but the function's configuration is retained
- **RET:** The function's memory and configuration is retained.
- **OPT:** The function is optional in the specified mode, and remains enabled if configured to be enabled
- **NS:** The function is not automatically disabled in the specified mode, but its use is not supported
- **OFF:** The function is fully powered off in the specified mode, and no configuration is retained

Table 3-2. Power Modes Summary

		RUN	SLEEP	STOP	STANDBY0	STANDBY1	SHUTDOWN
Supply State	**VDD**	3.3V	3.3V	3.3V	3.3V	3.3V	3.3V
	VDDC	1.35V	1.35V	1.0V	1.0V	1.0V	0V
PMU	**POR Monitor**	EN	EN	EN	EN	EN	EN
	BOR Monitor	EN	EN	EN	EN	EN	OFF

ADVANCE INFORMATION

Table 3-2. Power Modes Summary (continued)

		RUN	SLEEP	STOP	STANDBY0	STANDBY1	SHUTDOWN
	Main LDO	EN	EN	OFF	OFF	OFF	OFF
	STOP LDO	EN	EN	EN	EN	EN	OFF
	VOSC LDO	EN	EN	EN	EN	EN	OFF
Oscillators	**SYSOSC**	EN	EN	EN	DIS	DIS	OFF
	LFOSC	EN	EN	EN	EN	EN	OFF
	XTAL	OPT	OPT	DIS	DIS	DIS	OFF
	HPLL	OPT	OPT	DIS	DIS	DIS	OFF
	SYSOSC Frequency	32MHz	32MHz	4MHz	OFF	OFF	OFF
Clocks	**MCLK**	200MHz	200MHz	DIS	DIS	DIS	OFF
	MCLK/2	100MHz	100MHz	DIS	DIS	DIS	OFF
	MCLK/4	50MHz	50MHz	DIS	DIS	DIS	OFF
	CPUCLK	200MHz	DIS	DIS	DIS	DIS	OFF
	LFCLK	32KHz	32KHz	32KHz	32KHz	32KHz	OFF
	ULPCLK	50MHz	50MHz	4MHz	32KHz	OFF	OFF
	CANCLK	20MHz	20MHz	DIS	DIS	DIS	OFF
	ADCCLK	200MHz	200MHz	DIS	DIS	DIS	OFF
Core Functions	**CPU**	EN	DIS	DIS	DIS	DIS	OFF
	DMA	EN	EN	DIS	DIS	DIS	OFF
	Flash	EN	EN	DIS	DIS	DIS	OFF
	SRAM	EN	EN	DIS + RET	DIS + RET	DIS + RET	OFF
Peripherals	**PD1 Peripherals**	OPT	OPT	DIS + RET	DIS + RET	DIS + RET	OFF
	PD0 Peripherals	OPT	OPT	EN	EN	EN	OFF
Analog	**ADC**	OPT	OPT	DIS	DIS	DIS	OFF
	CMPSS	OPT	OPT	OFF	OFF	OFF	OFF
	VREF	EN	EN	EN (Sampled)	EN (Sampled)	EN (Sampled)	OFF
IO Mux		EN	EN	EN + RET	EN + RET	EN + RET	OFF
IO Wakeup		N/A	EN	EN + RET	EN + RET	EN + RET	OFF
Wake Sources		N/A	• IRQ • IO • DMA Req • Debug Req • WWDT	• IO • Debug Req • WWDT	• IO • Debug Req • WWDT	• IO • Debug Req	• IO (subset)(1) • Debug Req

(1) Select WAKEUP GPIOs. Refer to the device-specific data sheet.

3.2 Quick Start Reference

The PMCU is designed to provide a simple, easy-to-use power management, clocking, and reset management functionality. This section describes the basic operating principles of the PMCU as well as tips and tricks for taking the default configuration out of reset and optimizing it for particular applications.

3.2.1 Increasing MCLK Precision

If an application requires high clocking accuracy for high-frequency peripherals, the best accuracy is achieved by using an external high-frequency crystal with the XTAL and sourcing the MCLK tree directly from HFCLK. By

ADVANCE INFORMATION

sourcing the MCLK directly from HFCLK, the bus clock to all peripherals in PD1 and PD0 will be the HFCLK. Crystal frequencies in the 10-25 MHz range are supported by the XTAL.

If precision clocking is needed for the CAN-FD controller and high CPU performance (high MCLK) is also required, it is also possible to source the CAN-FD controller (CANCLK) from HFCLK directly, asynchronous to MCLK, while running MCLK at maximum frequency using the PLL for the best computing performance.

3.2.2 Configuring MCLK for Maximum Speed

The best CPU compute performance is obtained by using the SYSPLL to generate the maximum clock frequency from either the SYSOSC reference or XTAL.

Running MCLK at the maximum clock frequency also provides the best possible timer resolution for TIMG, MCPWM, ECAP, and EQEP peripherals and the best throughput for ADCs in the PD1 domain.

3.2.3 High Speed Clock (SYSPLL, HFCLK) Handling in Low-Power Modes

The SYSPLL and HFCLK (XTAL, HFCLK_IN) high speed clock sources are not supported in the STOP and STANDBY operating modes. When a high-speed clock source (SYSPLL, HFCLK) is enabled, entering either the STOP mode or STANDBY mode will cause SYSCTL to automatically disable the SYSPLL and/or HFCLK before entering STOP or STANDBY mode. Upon exit from STOP or STANDBY mode to RUN mode, SYSCTL will automatically re-enable the SYSPLL and/or HFCLK if they were previously enabled before entering the low-power mode.

Before entering STOP or STANDBY for the first time after enabling the SYSPLL or HFCLK, and after waking up from STOP or STANDBY mode, application software must wait to enter STOP or STANDBY mode until any previously enabled high speed clock sources have completed startup. Application software must check the following before entering STOP or STANDBY mode:

- If the **SYSPLL was enabled,** then application software must wait for the SYSPLL to complete startup. When the SYSPLL startup is complete, the SYSPLLGOOD bit will be set in the CLKSTATUS register in SYSCTL. If the SYSPLL fails to start, the SYSPLLOFF bit will be set instead. The SYSPLLOFF bit indicates that the SYSPLL was dead at startup or was not previously enabled. Ensure that either SYSPLLGOOD or SYSPLLOFF is set before attempting to enter STOP or STANDBY.
- If the **HFCLK was enabled,** then application software must wait for the HFCLK to complete startup. When the HFCLK startup is complete, the HFCLKGOOD bit will be set in the CLKSTATUS register in SYSCTL. If the HFCLK fails to start, the HFCLKOFF bit will be set instead. The HFCLKOFF bit indicates that the HFCLK was dead at startup or was not previously enabled. Ensure that either HFCLKGOOD or HFCLKOFF is set before attempting to enter STOP or STANDBY.

In the event that the MCLK was configured to be sourced from HSCLK before entry to STOP or STANDBY, upon exit from STOP or STANDBY the MCLK will be sourced from SYSOSC initially and the CPU will be released to begin executing code at the SYSOSC frequency. SYSCTL will automatically restore the MCLK configuration to the previously selected high-speed clock when the high-speed clock has started and is ready for use. When MCLK switches back to the high-speed clock, SYSCTL will generate an HSCLK GOOD interrupt to alert the application that MCLK is again running from the high-speed clock.

3.3 Power Management (PMU)

The Power Management Unit (PMU) generates the regulated core supplies for the device and provides supervision of the external supply. It also contains a bandgap voltage reference used by the PMU and other analog peripherals. Refer to the device specific datasheet for the supported PMU features.

Key PMU features include:

- Support for the device across the device operating supply range
- **Main LDO:** Low-dropout linear voltage regulator to generate the internal core logic supply, with multiple operating modes for reducing device current in low-power modes (managed automatically by SYSCTL)
  - Also referred to as the **Core Regulator** or **Core LDO**
- **STOP LDO:** Low-dropout linear voltage regulator to power both PD0 active logic and PD1 retention logic/ memory in STOP/STANDBY modes
- Power-on reset (POR) for VDD supply
- Brownout reset (BOR) for VDD supply
- Bandgap voltage reference supporting the BOR, core regulator, and analog peripherals

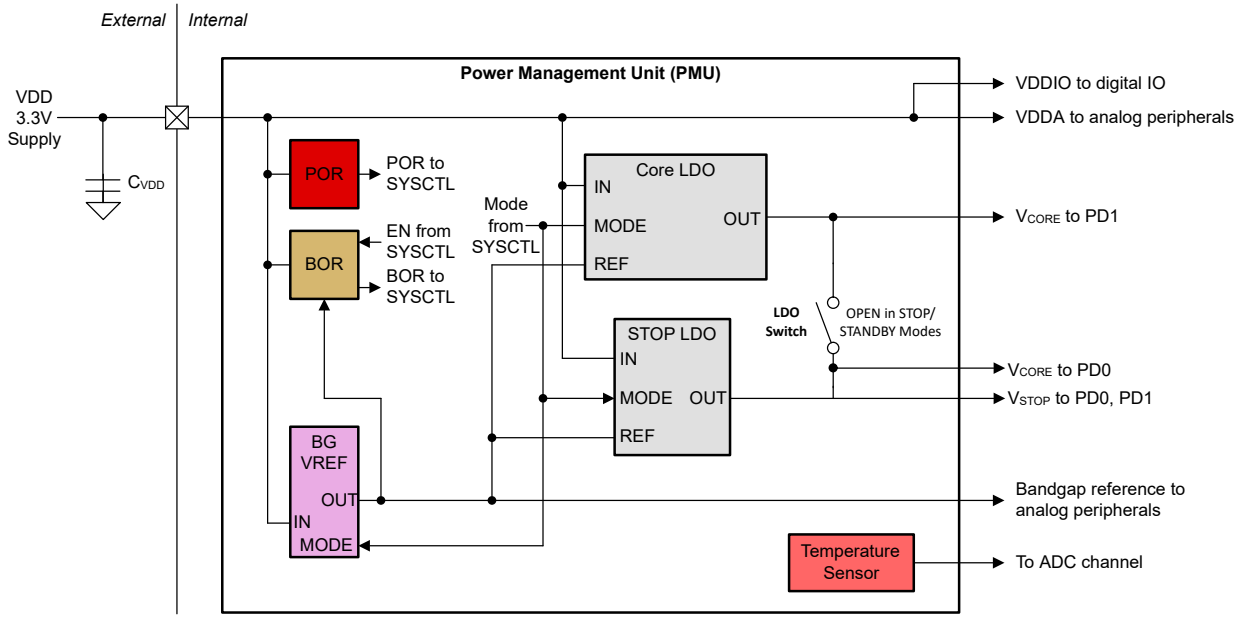


Figure 3-4. PMU Block Diagram

3.3.1 Power Supply

Power is supplied to the device through the VDD/VSS and VDDA/VSSA connections. A decoupling capacitor (C_{VDD}/C_{VDDA}) must be placed across all VDD/VSS and VDDA/VSSA supply pairs. See the device-specific data sheet for the correct value and tolerance of the decoupling capacitors.

Note

VDD and VDDA are to be connected at the board level. There is no separate supply monitor for VDDA, and it is expected that the same power supply being used for VDD is also connected to VDDA.

VDD is used directly to provide the IO supply (VDDIO). VDDA is used directly to provide the analog supply (VDDA).

ADVANCE INFORMATION

3.3.1.1 Main LDO

The PMU uses an on-chip, configurable, low-dropout linear voltage regulator to generate a 1.35V supply rail to power the device core, also referred to as the **core regulator**. The core regulator output (V_{CORE}) supplies power to the core logic, which includes the CPU, digital peripherals and the device memory.

The MAIN LDO is active in RUN and SLEEP modes. In all other power modes (STOP, STANDBY, SHUTDOWN), the regulator is off. SYSCTL automatically configures the regulator for the best power consumption based on the power mode that is currently active.

3.3.1.2 STOP LDO

The PMU uses an on-chip, configurable, low-dropout linear voltage regulator to generate a 1.0V supply rail to power the PD0 active logic and PD1 retention logic domains when the device is in STOP/STANDBY modes.

The STOP LDO is active in all power modes except for SHUTDOWN. In all other power modes (RUN, SLEEP, STOP, and STANDBY) the drive strength of the regulator is configured automatically to support the max load current of each mode. This reduces the quiescent current of the regulator when using low-power modes, improving low power performance. SYSCTL automatically configures the regulator for best power consumption based on the power mode which is currently active.

3.3.1.3 VOSC LDO

The VOSC LDO provides a fixed 1.35V to the SYSOSC, LFOSC and corresponding clock muxes. This LDO is on in all modes except SHUTDOWN.

3.3.1.4 HPLL LDO

This LDO provides a fixed 1.35V to the HPLL and will be switched OFF/ON with every PLL disable and enable cycle.

3.3.2 Supply Supervisors

The PMU provides two supply supervisor circuits:

- A power-on reset (POR) circuit to indicate that the external supply has reached sufficient voltage to start the on-chip bandgap reference and BOR circuit
- A brownout reset (BOR) circuit which verifies that the external supply is maintained at the specified voltage to support correct operation of the device

3.3.2.1 Power-on Reset (POR)

The power-on reset (POR) circuit assures a proper start-up of the device. During cold power-up, the device is held in a POR state until VDD passes the POR+ threshold. When VDD has passed POR+, the POR state is released and the bandgap reference and BOR monitor circuitare started. If VDD drops below the POR- level, then a POR- violation is asserted and the device is again held in a POR reset state.

The POR circuit does not indicate that VDD has reached a level high enough to support correct operation of the device. Rather, it is the first step in the boot process and is used to determine if the supply voltage is sufficient to power up the bandgap reference and BOR circuit, which are then used to determine if the supply has reached a level sufficient to for the device to run correctly.

The POR circuit is active in all power modes including SHUTDOWN and cannot be disabled.

3.3.2.2 Brownout Reset (BOR)

The brownout reset (BOR) supervisor monitors the external supply (VDD) and asserts or deasserts a BOR violation to SYSCTL. The primary responsibility of the BOR circuit is to make sure that the external supply is maintained high enough to enable correct operation of internal circuits, including the core regulator. The BOR threshold reference is derived from the internal bandgap circuit. The threshold is fixed and is always higher than the POR threshold. During cold start, after VDD passes the POR+ threshold, the bandgap reference and BOR circuit are started. The device is then held in a BOR state until VDD passes the BOR+ threshold. When VDD passes BOR+, the BOR supervisor releases the device to continue the boot process, and the PMU is started. A BOR- violation always generates a BOR- violation signal to SYSCTL, generating a BOR level reset. The BOR

supervisor is active in RUN, SLEEP, STOP, and STANDBY modes but is disabled automatically in SHUTDOWN mode.

The BOR threshold level is fixed, and is referred to as BOR2. Refer to the device-specific data sheet for BOR voltage level thresholds.

3.3.2.3 POR and BOR Behavior During Supply Changes

When the supply voltage (VDD) drops below POR-, the entire device state is cleared. Small variations in VDD that do not pass below the BOR- threshold do not cause a BOR- violation, and the device continues to RUN.

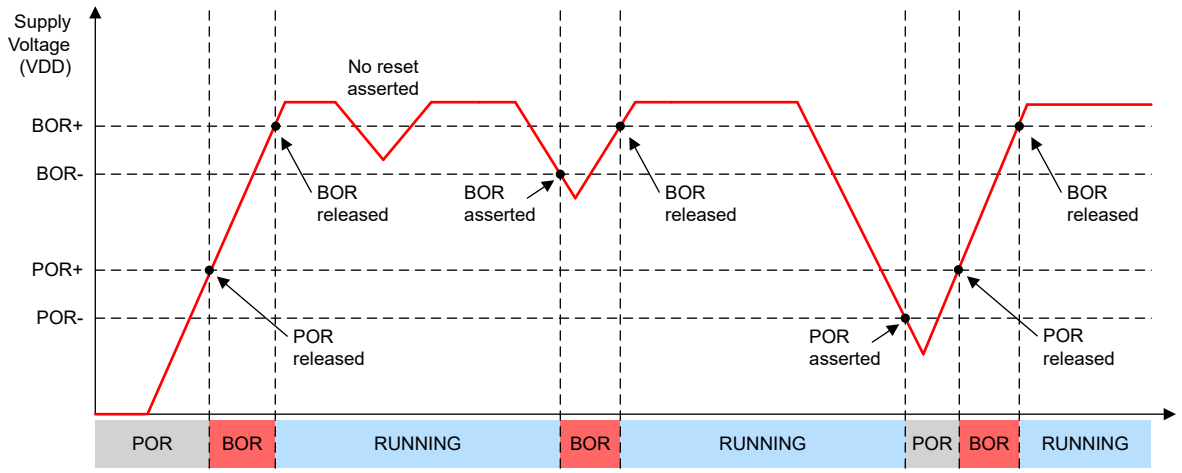


Figure 3-5. POR/BOR vs. Supply Voltage (VDD)

3.3.3 Bandgap Reference

The PMU provides a voltage reference that is stable across temperature and VDD supply voltage and is used by the device for internal functions, including:

- Deriving the brownout reset circuit thresholds
- Setting the output voltage for the core regulator
- Deriving the on-chip VREF levels for on-chip analog peripherals

The bandgap reference is enabled in RUN, SLEEP, STOP modes. This reference operates in a sampled mode in STANDBY to reduce power consumption and is disabled in SHUTDOWN mode. SYSCTL manages the bandgap state automatically; no user configuration is required.

3.3.4 Analog Supplies

The device has the following analog power supply pins for powering analog peripherals and the analog voltage references:

- **VDDA:** 3.3V analog supply used by all modules in ADCCIO (ADC, analog voltage references, CMPSS, internal temperature sensor)
- **VSSA:** Analog ground used by all modules in ADCCIO
- **VREFHI:** ADC0/1/2 reference HIGH input
- **VREFLO:** ADC0/1/2 reference LOW input

On all device packages, physical VREFHI and VREFLO pins are shared by all ADCs. Physical VDDA and VSSA pins supply the ADCs, comparators, PGAs, internal temperature sensor, and analog reference buffer (REFBUF).

3.3.4.1 Analog Reference Circuits

The analog reference circuits are supplied by a single bandgap circuit. The ADC voltage reference input is fixed to the VREFHI pin on the device. There is a single buffer that drives either the internal reference to the VREFHI pin or tri-states the reference buffer to high-impedance, which allows an external voltage source to drive the VREFHI pin. For both internal and external reference modes, VREFHI requires a decoupling capacitor, C_{VREF}. The value of C_{VREF} for internal reference mode is defined in the device-specific data sheet. The value of C_{VREF} for external reference mode is dependent on the external reference driver.

In internal reference mode, the ADCREFBUF drives all ADCs. In external reference mode, the VREFHI/VREFLO pads drive the reference. At any given time, **all ADCs** are in internal or external reference mode - i.e. there is no independent configuration of voltage reference for each ADC.

To put the ADCs in internal reference mode, the following condition must be met: {CIO.ANAREFSEL & (ADCEN0 || ADCEN1 || ADCEN2)}. At a register level, the ADC reference mode is selected by setting:

- ANAREFCTL.ANAREFSEL=0 (switch CLOSED) for internal reference mode
- ANAREFCTL.ANAREFSEL=1 (switch OPEN) for external reference mode

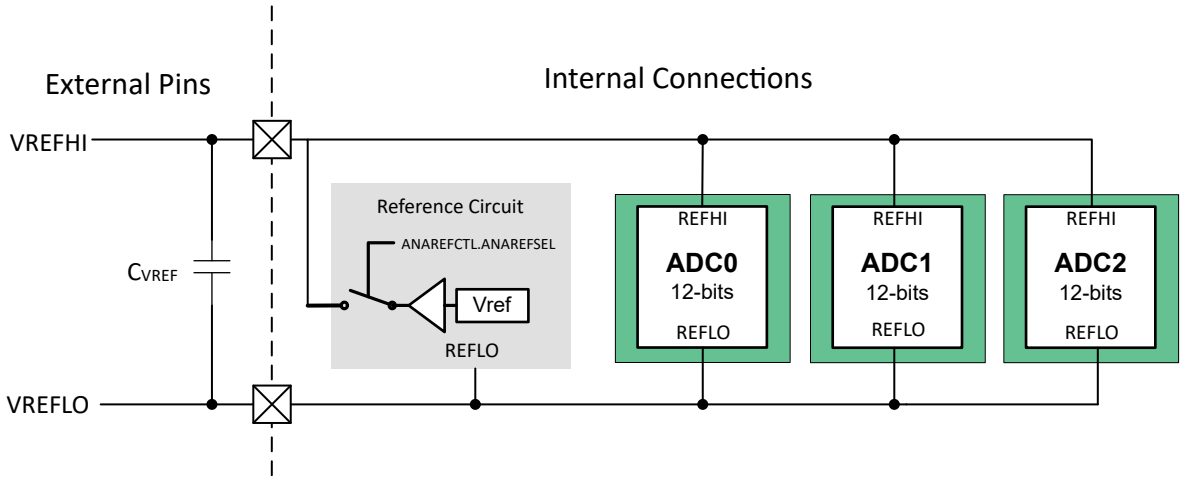


Figure 3-6. VREF to ADC Connections

Note

1. Because VREFHI and VREFLO pins are shared between all ADCs, the control for 2.5V or 3.3V internal reference mode will apply to all ADCs. The same logic applies for external reference mode, i.e. all ADCs will receive the same external voltage reference. For more information on the internal reference modes, refer to *Internal Reference Mode*.
2. The ADCENx corresponds to the ADCCTL1.ADCPWDNZ bit for each ADC
3. When all ADCs are disabled, the ADCREFBUF is disabled and any voltage applied to the VREFHI pin will discharge to VSSA.

For full details on the analog reference configuration, refer to Section 16.2.3 , *ADC - Voltage Reference*.

3.3.5 Internal Temperature Sensor

The PMU provides a temperature sensor which can be used to approximate the temperature of the device. The temperature sensor is connected internally to the following ADC mux channels:

Table 3-3. Temperature Sensor Connections

ADC Instance	ADC Mux Channel
ADC0	13
ADC2	13

The ADCs listed in the table above are used to perform temperature measurements. The temperature sensor outputs a voltage which has a linear relationship with temperature. The temperature coefficient (TS_C) is the slope of the temperature-voltage relationship (given in mV/C), and is given in the specifications section of the device-specific data sheet.

3.3.6 Peripheral Enable

All peripherals on a device, with the exception of infrastructure peripherals such as SYSCTL itself and the IOMUX, require two user actions to enable the peripheral to be ready for operation:

- Peripheral Write Enable
  - Provided by the peripheral write enable control register (PWREN) with a KEY and ENABLE field.
  - Before any other peripheral registers are configured by software, the peripheral itself must be enabled by writing the ENABLE bit together with the appropriate KEY value to the peripheral's PWREN register
  - When the PWREN.ENABLE bit is cleared, the peripheral's registers are not accessible for read/write operations.
- Peripheral Function Enable
  - Peripherals may have a functional enable bit in the corresponding register map to provide access to power or for operation

Note

After setting the ENABLE | KEY bits in the PWREN register to enable a peripheral, wait at least 4 ULPCLK clock cycles before accessing the rest of the peripheral's memory-mapped registers. The 4 cycles allow for the bus isolation signals at the peripheral's bus interface to update.

3.3.6.1 Automatic Peripheral Disable in Low Power Modes

Peripherals in power domain 1 (PD1) will be forced to a disabled state by SYSCTL upon entry into a STOP or STANDBY low-power mode. As such, these peripherals will not be available for use in STOP or STANDBY.

Most PD1 peripherals will retain their configuration settings after being automatically disabled, such that re-configuration is not required upon exit from STOP or STANDBY mode. See the peripheral-specific chapter in this guide for details on which peripheral registers are retained through STOP and STANDBY mode for PD1 peripherals.

If a PD1 peripheral was multiplexed to an IO pin (through the IOMUX) in an output configuration, the last valid output state (logic 0 or logic 1) from the peripheral to the IO is latched upon entry to STOP or STANDBY mode. This prevents external circuits from being disturbed by SYSCTL disabling a peripheral during low-power operation. Upon exit from STOP or STANDBY mode, the IO is again connected to the peripheral as the peripheral becomes re-enabled.

3.4 Clock Module (CKM)

The clock module contains the internal and external oscillators, the clock monitors, and the clock selection and control logic. A frequency clock counter is also provided for checking and/or calibrating the frequency of high-speed clocks against either the LFCLK or a reference period/pulse provided on an IO pin (FCC_IN).

3.4.1 Clock Tree

The following figure shows the top level clocking tree for AM13E230x family devices. This diagram shows the mapping between oscillators (sources) and clocks (destinations), as well as the SYSCTL register bit fields for the selection muxes.

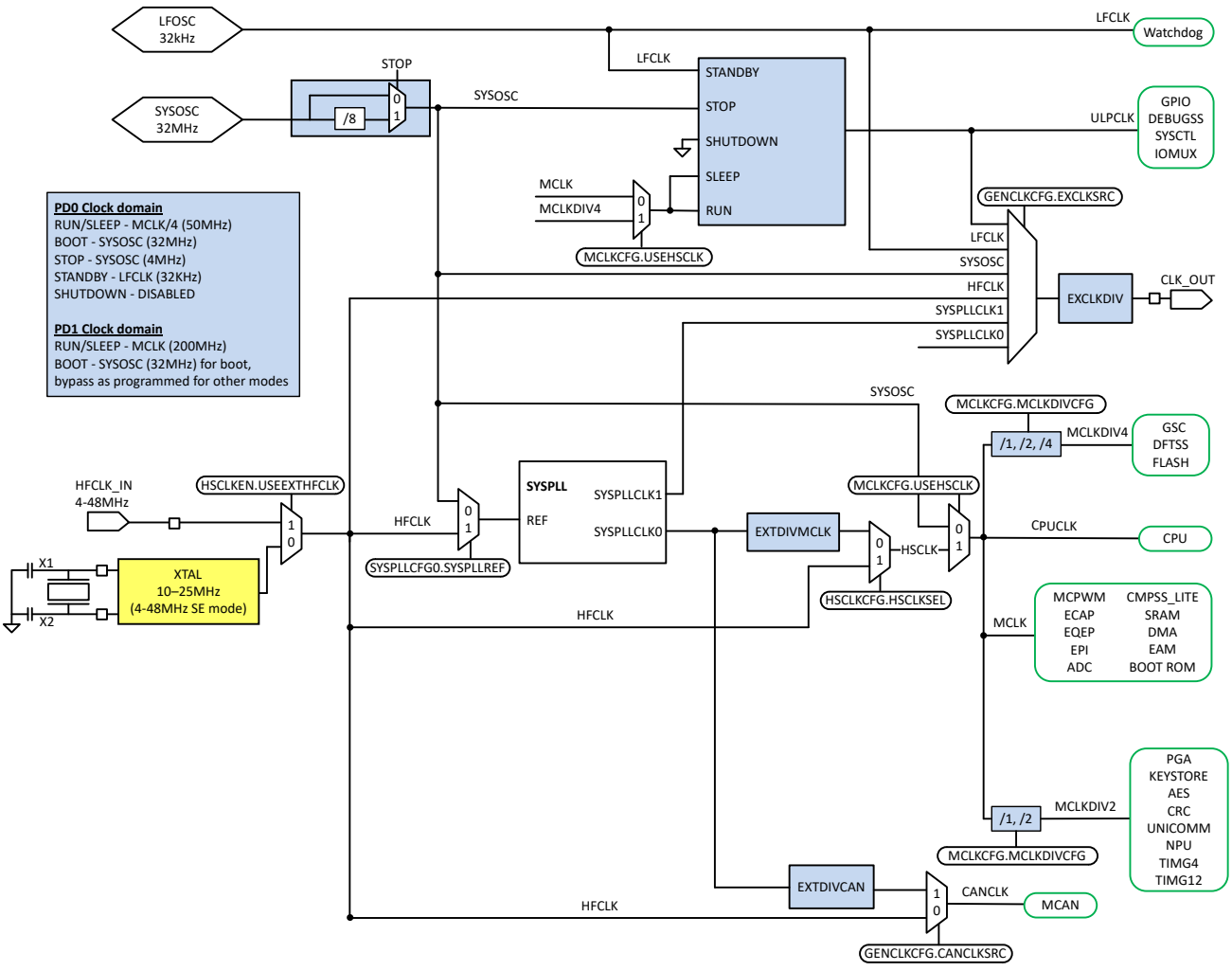


Figure 3-7. AM13E230x Top Level Clocking Diagram

There a few additional items to note about the AM13E230x clocking system:

1. LFCLK is a fixed-frequency 32kHz clock that can be selected by certain peripherals for ensuring a constant clock rate even when MCLK or ULPCLK changes source or rate. LFCLK is always synchronized to MCLK and ULPCLK.

ADVANCE INFORMATION

2. Semi-clock domain peripherals operate on MCLK divided by 2 (MCLKDIV2) clock. The MCLKDIV2 clock domain is synchronous to the MCLK domain and is provided through the semi-divider. This divider allows /2 or bypass with bit 26 of the MCLKCFG register (MCLKDIVCFG).
   a. Programming MCLKCFG[26] = '1' also forces MCLKCFG[24] = '1'. Therefore, writes to MCLKCFG[24] will be ignored if MCLKCFG[26] = '1'.
   b. MCLKDIV2 domain is used and configured when operating from HSCLK and not used when operating from SYSOSC (e.g. out of boot without PLL or out of low power mode).
3. Quarter-clock domain peripherals operate on MCLK divided by 4 (MCLKDIV4) clock. The MCLKDIV2 clock domain is synchronous to the MCLK domain and is provided through the quarter-divider. This divider allows /2, /4, or bypass with bits 24-25 of the MCLKCFG register (MCLKDIVCFG).

Table 3-4. Semi Clock (MCLKDIV2) Divider Select

MCLKCFG[26]	Divider Select
0	/1 (Bypass)
1	/2

Table 3-5. Quarter Clock (MCLKDIV4) Divider Select

MCLKCFG[25]	MCLKCFG[24]	Divider Select
0	0	/1 (Bypass)
0	1	/2
1	0	Reserved
1	1	/4

Valid MCLK Clock Combinations

Each of the MCLKDIV2 and MCLKDIV4 dividers can be configured through MCLKCFG[26:24] with divider selection as shown below. Possible frequency limits corresponding to clock divider values are listed in Table 3-6.

Table 3-6. Valid MCLK Divider Values

Frequency Range (MHz)	Option	MCLKCFG[26]	MCLKCFG[25]	MCLKCFG[24]	MCLK	MCLK/2	MCLK/4
100-200	1 (default)	1	1	1	/1	/2	/4
50-100	1 (default)	0	0	1	/1	/1	/2
	2	0	1	1	/1	/1	/4
	3	1	0	1	/1	/2	/2
	4	1	1	1	/1	/2	/4
<50	1 (default)	0	0	0	/1	/1	/1
	2	0	0	1	/1	/1	/2
	3	0	1	1	/1	/1	/4
	4	1	0	1	/1	/2	/2
	5	1	1	1	/1	/2	/4

3.4.2 Oscillators

Several internal and external oscillators are provided for generating low to high frequency clocks for use by the system. The CKM contains all the oscillators in the device and uses them to generate the system clocks. See the device specific data sheet for whether external oscillators are available.

Internal Oscillators

- LFOSC: low frequency oscillator (32kHz typical frequency)
- SYSOSC: system oscillator (32MHz frequency)

ADVANCE INFORMATION

- SYSPLL: system PLL (phase-locked loop) with programmable frequency (optional)

External Oscillators

- XTAL: crystal oscillator (10-25MHz frequency range)

In addition to XTAL, a high frequency digital clock input (HFCLK_IN) with a range of 4-48MHz can be provided to support cases where the XTAL is not used and a direct digital clock is supplied to the device. Note that XTAL also supports a single-ended differential mode.

3.4.2.1 Internal Low-Frequency Oscillator (LFOSC)

The low-frequency oscillator (LFOSC) is an on-chip low power oscillator which is factory trimmed to a frequency of 32.768kHz, providing a stable low frequency clock source.

LFOSC can provide higher accuracy when used over a reduced temperature range. See the device-specific data sheet for details.

The LFOSC is active by default after a BOOTRST, sourcing the LFCLK. The LFOSC startup monitor sets the LFOSCGOOD bit in the CLKSTATUS register when LFOSC is ready.

3.4.2.2 Internal System Oscillator (SYSOSC)

The system oscillator (SYSOSC) is an on-chip, accurate high frequency clock source (32MHz factory-trimmed). The SYSOSC provides a flexible high-speed clock source to the system in cases where the crystal oscillator (XTAL) is either not present or not used, or where fast wake-up from a low-power mode is required.

Key features of the SYSOSC include:

- High accuracy when using optional frequency correction loop (FCL) and reference resistor
  - The frequency correction loop supports correction through an internal resistor, depending on the device capabilities
- Fast start-up time from a low power state
- Capable of switching from base frequency to low frequency, or low frequency to base frequency, in one clock cycle with no functional interruption (gear shift)
  - Glitch-less divider and phase-aligned transition to minimize disturbance to peripherals
  - Fast settling to specified accuracy
  - SYSCTL can initiate seamless gear shift frequency switch in STOP mode with user-configurable FCL enablement to reduce SYSOSC current
- Two frequency options: 32MHz supporting RUN/SLEEP power modes and 4MHz in STOP mode
  - While in 32MHz mode, a divider giving a fixed 4MHz output can be multiplexed to when the mode switches

The SYSOSC is active at the device's base frequency (32MHz) by default after a brownout reset, sourcing MCLK.

3.4.2.2.1 SYSOSC Frequency and User Trims

SYSOSC comes factory trimmed for operation at the base frequency of 32MHz, with the option to divide down to 4MHz in STOP mode. The default frequency of operation is the base frequency. In STOP mode, the MCLK (and the ULPCLK) tree can run from SYSOSC with a 4MHz rate if the DISABLESTOP bit in the SYSOSCCFG register is low. When the low frequency is used, SYSCTL ensures that ULPCLK is always 4MHz even if SYSOSC is running at a higher frequency due to user configuration or due to an asynchronous request from a peripheral.

3.4.2.2.2 SYSOSC Frequency Correction Loop

The SYSOSC frequency accuracy can be improved through the use of the Frequency Correction Loop (FCL) feature. The FCL circuit uses an internal resistor to stabilize the SYSOSC frequency by providing a precise reference current for the SYSOSC. The overall frequency accuracy which is achievable depends on the operating temperature range together with the tolerance and temperature drift of the selected reference resistor.

3.4.2.2.2.1 SYSOSC FCL in Internal Resistor Mode

This section describes the procedure for selecting the internal resistor mode through the device SYSCTL registers. Enabling FCL in internal resistor mode can be beneficial to the user's application which require improved SYSOSC performance while not adding cost to the system for an external resistor.

Enabling FCL with Internal Resistor

To increase the SYSOSC accuracy with FCL, follow this procedure:

1. Enable FCL mode by setting the SETUSEFCL bit in the SYSOSCFCLCTL register.
2. When the FCL mode is enabled, software cannot disable the mode. A BOOTRST is required before a change to the FCL mode.

3.4.2.2.3 Disabling SYSOSC

It is not possible to disable SYSOSC when using a different high frequency clock to source MCLK (such as HFCLK or the PLL). This is due to SYSOSC being used by SYSCTL logic when MCLK is sourced from HFCLK or the PLL.

SYSOSC is always disabled automatically in STANDBY and SHUTDOWN modes. The DISABLE bit in the SYSOSCCFG register displays the status of SYSOSC enable/disable policy.

3.4.2.3 System Phase-Locked Loop (SYSPLL)

The system phase locked loop (SYSPLL) takes an input reference clock SYSPLLREF and scales the input frequency to produce user-specified high frequency clocks (SYSPLLCLK0 and SYSPLLCLK1) for use by the device. Specifically, the SYSPLL clock outputs can be used as sources to MCLK and CANCLK. Figure 3-8 shows the block diagram of the SYSPLL. The relation between SYSPLLCLK and SYSPLLREF is given by Equation 1.

$$f_{SYSPLLCLKx} = \frac{f_{SYSPLLREF} \times (QDIV)}{(PDIV) \times (RDIVCLKx)} \quad (1)$$

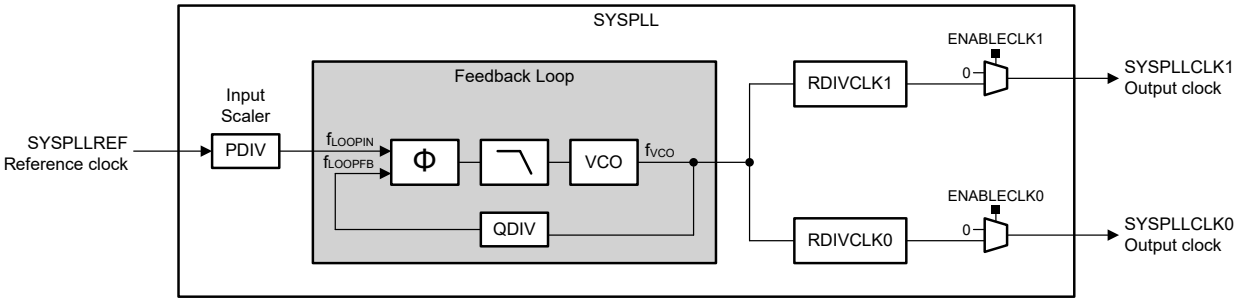


Figure 3-8. SYSPLL Diagram

3.4.2.3.1 Configuring SYSPLL Output Frequencies

The SYSPLL accepts an input reference clock from 4-48MHz. The available reference clocks include SYSOSC and HFCLK. The predivider PDIV scales the selected input reference clock ahead of the PLL feedback loop. PDIV can be selected as /1, /2, /4, or /8 by programming 0x0 to 0x3, respectively, into the PDIV field in the SYSPLLCFG1 register.

The effective divider can be calculated from the PDIV register setting as shown in Equation 2.

$$SYSPLLREF_{DIV} = 2^{PDIV} \quad (2)$$

ADVANCE INFORMATION

The PLL feedback loop sets the Voltage Controlled Oscillator (VCO) output equal to the divided input reference clock f_{LOOPIN} multiplied by the QDIV feedback divider. The QDIV divider is an integer divider with a valid range of /2 to /127. The desired QDIV divider is selected by programming 0x01 to 0x7E for /2 to /127, respectively, into the QDIV field of the SYSPLLCFG1 register. SYSPLLCFG1.QDIV=0x00 is an invalid configuration. The effective feedback divider can be calculated from the QDIV register setting as shown in Equation 3.

$$SYSPLLFB_{DIV} = QDIV + 1 \quad (3)$$

The output frequency of the VCO $f_{VCO}$is given in Equation 4.

$$f_{VCO} = f_{SYSPLLREF} \times SYSPLLFB_{DIV} / SYSPLLREF_{DIV} \quad (4)$$

The VCO output sources two separate SYSPLL outputs (SYSPLLCLK0 and SYSPLLCLK1). Each output has its own divider unit to enable generation of up to 2 different output frequencies for use by different modules in the device.

For SYSPLLCLK1 and SYSPLLCLK0, the output dividers can be set from /2 to /32 in steps of 2. To set the SYSPLLCLK0 or SYSPLLCLK1 output divider, program 0x0-0xF for /2 to /32, respectively, into the corresponding RDIVCLKx field in the SYSPLLCFG0 register. Equation 5 shows how to compute the effective SYSPLLCLK0 divider based on a given RDIVCLK0 setting, and Equation 6 shows how to compute the effective SYSPLLCLK1 divider based on a given RDIVCLK1 setting.

$$SYSPLLCLK0_{DIV} = 2 \times (RDIVCLK0+1) \quad (5)$$

$$SYSPLLCLK1_{DIV} = 2 \times (RDIVCLK1+1) \quad (6)$$

The SYSPLL output clock frequencies are thus set by the combination of f_{VCO} and the respective dividers:

$$f_{SYSPLLCLK0} = f_{VCO} / SYSPLLCLK0_{DIV} \quad (7)$$

$$f_{SYSPLLCLK1} = f_{VCO} / SYSPLLCLK1_{DIV} \quad (8)$$

Enabling and Disabling the SYSPLL

After configuration, enable the SYSPLL by setting the SYSPLLEN bit in the HSCLKEN register. Before enabling the SYSPLL, make sure that the SYSPLL is in a disabled state by verifying that the SYSPLLOFF bit in the CLKSTATUS register is set. After the SYSPLL is enabled, application software must not disable it until the SYSPLLGOOD bit or SYSPLLOFF bit is set in the CLKSTATUS register, indicating that the SYSPLL transitioned to a stable active state or a stable dead state. When the SYSPLL is enabled, the SYSPLL reference clock selection must not be changed.

Please refer to SYSPLL Usage Example (for CAN protocol) for details on how to configure and lock the SYSPLL.

Note

SYSOSC must be enabled and running at base frequency when the SYSPLL is enabled, even if HFCLK is used as the SYSPLL reference clock.

SYSPLL Usage Example (for CAN protocol)

To illustrate the above relationships, take as an example the following requirements:

- The high-frequency external clock (HFCLK) is used as the SYSPLL reference (operating at 20MHz in this example)
- The SYSPLL must be configured with an output frequency of 200MHz to source MCLK and CANCLK at 40MHz.

ADVANCE INFORMATION

To achieve this, the VCO can be configured for 400MHz through the use of PDIV and QDIV. Then, SYSPLLCLK0 can feed CANCLK and MCLK with an output divider of /2. Lastly, the external output divider EXTDIVCAN can be used to source CANCLK at 40MHz.

The steps below describe how to configure the CKM to use SYSPLL in this way:

1. Verify that the SYSPLL is disabled (SYSPLLOFF is set in CLKSTATUS)
2. Make sure that SYSOSC is running at base frequency (32MHz); this is a requirement for SYSPLL operation even though HFCLK is used as the SYSPLL reference clock instead of SYSOSC
3. Set HFCLK as the SYSPLL reference (make sure that the SYSPLLREF bit in the SYSPLLCFG0 register is cleared; this is the default state after reset)
4. Select a predivider PDIV to /2 (set SYSPLLCFG1.PDIV to 0x01), setting f_{LOOPIN} to 10MHz (20 divided by 2)
5. Load the PLL parameters into SYSPLLPARAM0 and SYSPLLPARAM1 to support f_{LOOPIN} of 10MHz
6. Set the feedback divider QDIV to 40 (set SYSPLLCFG1.QDIV to 0x27), giving f_{VCO}=400MHz (10MHz multiplied by 40)
7. Set the SYSPLL output dividers for SYSPLLCLK0 to /2 (set SYSPLLCFG0.RDIVCLK0 to 0x0) to get 200MHz at SYSPLLCLK0
8. Enable SYSPLLCLK0 outputs by setting the ENABLECLK0 bit in the SYSPLLCFG0 register
9. With HFCLK running at 20MHz frequency, enable the SYSPLL by setting SYSPLLEN in the HSCLKEN register
10. Wait for the SYSPLLGOOD indication by testing SYSPLLGOOD in the CLKSTATUS register
11. Select the SYSPLL as the HSCLK source by ensuring that the HSCLKSEL bit is cleared in the HSCLKCFG register (this is the default state)
12. Select the high-speed clock (HSCLK) as the source for MCLK by setting the USEHSCLK bit in the MCLKCFG register. This will switch MCLK from SYSOSC to HSCLK. MCLK is now running from SYSPLLCLK0 at 200MHz.
13. Enable the CANCLK divider (EXTDIVCAN) by setting GENCLKEN.CANEXTDIVEN and configure GENCLKEN.EXTDIVCAN to divide by 5 before sourcing CANCLK
14. Set the CANCLKSRC bit in the GENCLKCFG register to 0x01, applying SYSPLLCLK as CANCLK source

The SYSPLL divider values used in this example are summarized below for reference.

Table 3-7. SYSPLL Divider Example Settings

Parameter	Register	Bit Field	Bit Field Value	Actual Divider
Input reference clock divider	SYSPLLCFG1	PDIV	0x1	/2
VCO feedback loop divider	SYSPLLCFG1	QDIV	0x27	/40
Output clock 0 divider	SYSPLLCFG0	RDIVCLK0	0x0	/2
CAN clock divider	GENCLKEN	EXTDIVCAN	0x4	/5

Tuning Guidelines

In cases where there are multiple combinations of PDIV, QDIV, and RDIVCLKx that provide the desired output frequencies, consider these tuning guidelines to determine the best possible values for an application:

- Lower VCO frequencies (f_{VCO}) result in lower power consumption. Refer to the device data sheet for the allowable range of f_{VCO}.
- Higher feedback loop input frequencies (f_{LOOPIN}) have faster startup. For example, if an 200MHz output frequency is desired with f_{VCO}=200MHz, f_{VCO} = 200MHz can be derived from a SYSPLLREF clock 32MHz by setting PDIV to /8 and QDIV to /50. However, this gives slower startup because f_{LOOPIN} is <8MHz. The same result can be achieved by setting PDIV to /4 and QDIV to /25, but because f_{LOOPIN} is 8MHz in this case (32MHz divided by 4), the Table 3-8 for the higher input range can be used which give faster startup.

ADVANCE INFORMATION

3.4.2.3.2 Loading SYSPLL Lookup Parameters

Several tuning parameters must be configured in the SYSPLLPARAM0 and SYSPLLPARAM1 registers before using the SYSPLL. The values are determined by PLL feedback loop input clock frequency (f_{LOOPIN}).

SYSPLL supports four f_{LOOPIN} frequency ranges, given in Table 3-8. Each frequency range has a 64-bit lookup value in the FACTORY flash memory region that must be copied from flash into the SYSPLLPARAM0 and SYSPLLPARAM1 registers in SYSCTL before enabling the SYSPLL.

Table 3-8. SYSPLL Parameter Lookup

Lookup Address (SYSPLLPARAM0)	Lookup Address (SYSPLLPARAM1)
0x400B0128	0x400B012C

3.4.2.3.3 SYSPLL Startup Time

The PLL startup time depends on the PLL feedback loop input frequency f_{LOOPIN} and if the PLL is starting after previously running (for example, exiting a low-power mode) or starting for the first time after device boot.

Table 3-9 lists an example of how the PLL startup times can be affected by the loop input frequency f_{LOOPIN}. Note that f_{LOOPIN} is determined by the SYSPLLREF and user-configured PDIV and QDIV.

Note

These values are used as a relative example only. For specific device performance, please refer to the device datasheet.

Table 3-9. SYSPLL Startup Times (example)

f_{LOOPIN}	SYSPLL Startup Time (µs)	SYSPLL Startup Time on Exit From Low-Power Mode (µs)
32MHz ≤ FREQ	5	3
16MHz ≤ FREQ < 32MHz	15	12
8MHz ≤ FREQ < 16MHz	25	20
4MHz ≤ FREQ < 8MHz	40	35

3.4.2.4 External Crystal Oscillator (XTAL)

The external crystal oscillator (XTAL) can be used with standard crystals and resonators in the 10-25MHz range to generate a stable reference clock for the system. The XTAL can also operate in single-ended mode, with an external clock in the 4-48MHz range connected to the X1 pin. The XTAL can be used to clock the primary device clock tree (MCLK) directly, or it can be used as a precision reference to the on-chip PLL where higher frequencies can be generated.

To use the XTAL, a crystal or resonator must be populated between the X1 and X2 pins. Loading capacitors must be placed on both pins to circuit ground (VSS). The crystal load capacitors must be sized according to the specifications of the crystal being used. The IOMUX must be configured to enable XTAL functionality on the X1 and X2 pins.

A programmable XTAL startup time is provided with 64µs resolution. Program an appropriate startup time based on the desired crystal or resonator specifications into the XTALTIME field in the HFCLKCLKCFG register in SYSCTL before starting the XTAL.

Once configured properly, the XTAL is started by clearing the OSCOFF bit in the XTALCR register in SYSCTL. When the oscillator has started successfully, the HFCLK startup monitor will assert the HFCLKGOOD bit in the CLKSTATUS register in SYSCTL.

Note

SYSOSC must be enabled at base frequency when the XTAL is enabled.

ADVANCE INFORMATION

After clearing XTALCR.OSCOFF to enable the XTAL, application software must verify that either an HFCLKGOOD indication or an HFCLKOFF (off/dead) indication in the CLKSTATUS register was asserted by hardware. When disabling the XTAL, the XTAL must not be re-enabled again until the HFCLKOFF bit in the CLKSTATUS register is set by hardware.

To use XTAL as the PLL reference after receiving an HFCLKGOOD status, set the SYSPLLREF bit in the SYSPLLCFG0 register in SYSCTL. If XTAL is selected as a reference for the SYSPLL and the SYSPLL is enabled, then the SYSPLL must be disabled and the SYSPLLOFF bit in the CLKSTATUS register must be set before the XTAL can be disabled.

To use the XTAL directly as the MCLK source after receiving an HFCLKGOOD status, first set the HSCLKSEL bit in the HSCLKCFG register to select HFCLK as the high-speed clock source (rather than the system PLL output). Then, set the USEHSCLK bit in the MCLKCFG register to select the high-speed clock source as the MCLK source. Once USEHSCLK is set, HSCLKCFG must not change and the XTAL must not be disabled until the MCLK source is switched back to SYSOSC by clearing USEHSCLK and verifying that the HSCLKMUX bit in CLKSTATUS is cleared by hardware.

Crystal Oscillator Control and Configuration

External clock sources use the X1/X2 pins, which can either be used as an oscillator or as GPIOs. X1 is assigned to GPIO80 and X2 is assigned to GPIO81. These mode settings can be controlled through the XTALCR register. After power-up, the X1 and X2 pin functionality can be enabled by following the procedure in Using an External Crystal or Resonator.

Three types of external clock sources are supported:

- A single-ended 3.3V external clock. The clock signal must be connected to X1, with X2 available as a GPIO.
- An external crystal. The crystal must be connected across X1 and X2 with the load capacitors connected to VSS.
- An external resonator. The resonator must be connected across X1 and X2 with the ground connected to VSS.

Note

To ensure correct switching between single-ended and double-ended modes, the XTAL has to be powered down in between. That is, XTALCR.SE = 0 and XTALCR.OSCOFF = 1 has to be used as a transitory configuration state.

Table 3-10. XTAL Operating Modes

XTALCR.OSCOFF	XTALCR.SE	Operating Mode	X1 using GPIO80?	X2 using GPIO81?
0	0	Crystal Mode: Quartz crystal connected to X1/X2	No	No
0	1	Single-Ended Mode: External clock on X1	No	Yes
1	0	Oscillator off	Yes	Yes
1	1	Single-Ended Mode: External clock on X1	No	Yes

Using an External Crystal or Resonator

The X1 and X2 pins double as GPIO80 and GPIO81 respectively. At power-up, these pins are in GPIO mode and the on-chip crystal oscillator is powered off. The following procedure can be used to switch the pins to X1 and X2 mode and enable the oscillator:

1. Clear the XTALCR.OSCOFF bit
2. Wait for the crystal to power up (5-10ms is the typical wait time but this depends on the crystal that is being used)

3. Clear the X1 counter by writing a 1 to X1CNT.CLR and keep clearing until the X1 counter value in the X1CNT register is no longer saturated 2047 (0x7FF)
4. Wait for the X1 counter value in the X1CNT register to reach 2047 (0x7FF)
5. Repeat steps (3) and (4) three additional times
6. Select XTAL as the HFCLK source by writing a 0 to HSCLKEN.USEEXTHFCLK

Using an External Oscillator

The X1 and X2 pins double as GPIO80 and GPIO81 respectively. At power-up, these pins are in GPIO mode and the on-chip crystal oscillator is powered off. The following procedure can be used to switch the pins to X1 and X2 mode and enable the oscillator:

1. Clear the XTALCR.OSCOFF bit
2. Set the XTALCR.SE bit to enable single-ended mode
3. Clear the X1 counter by writing a 1 to X1CNT.CLR and keep clearing until the X1 counter value in the X1CNT register is no longer saturated 2047 (0x7FF)
4. Wait for the X1 counter value in the X1CNT register to reach 2047 (0x7FF)
5. Repeat steps (3) and (4) three additional times
6. Select XTAL as the HFCLK source by writing a 0 to HSCLKEN.USEEXTHFCLK

3.4.2.5 HFCLK_IN (Digital clock)

It is possible to bypass the XTAL circuit and bring in an external 4-48MHz digital clock signal into the device to use as the HFCLK source instead of XTAL. To configure HFCLK to use a digital clock input, first configure the IOMUX to enable the HFCLK_IN function on the appropriate pin. When IOMUX is configured correctly and the clock source is outputting a clock to HFCLK_IN, set the USEEXTHFCLK bit in the HSCLKEN register in SYSCTL.

Note

SYSOSC must be enabled at base frequency when the HFCLK_IN is enabled.

The HFCLK_IN can be used as the reference for SYSPLL. The SYSPLL must be off before changing the configuration of the SYSPLL reference clock source. The user can verify that SYSPLL is off by checking the SYSPLLOFF status bit in the CLKSTATUS register. The user must then select HFCLK_IN as the HFCLK source (by setting the USEEXTHFCLK bit in the HSCLKEN register) before setting the SYSPLLREF bit in the SYSPLLCFG0 register in SYSCTL. Finally, the SYSPLL can be enabled by using the SYSPLLEN bit in the HSCLKEN register.

To source MCLK from HFCLK_IN after selecting HFCLK_IN as the HFCLK source, first set the HSCLKSEL bit in the HSCLKCFG register to select HFCLK as the high-speed clock source. Then, set the USEHSCLK bit in the MCLKCFG register to select the high-speed clock source as the MCLK source. Once USEHSCLK is set, HSCLKCFG must not change and the HFCLK_IN must not be disabled until the MCLK source is switched back to SYSOSC by clearing USEHSCLK and verifying that the HSCLKMUX bit in CLKSTATUS was cleared by hardware.

HFCLK_IN is compatible with digital square wave CMOS clock inputs and should have a typical duty cycle of 50%.

Note

HFCLK_IN and XTAL are mutually exclusive and must not be enabled at the same time.

3.4.3 Clocks

The CKM takes oscillator outputs and generates a variety of functional clocks for use by the device. Refer to the device specific datasheet to determine the supported functional clocks.

Clocks

ADVANCE INFORMATION

- System Clocks
  - MCLK: Main system clock for PD1 peripherals and PD1 bus
  - CPUCLK: CPU clock, derived from MCLK is available only in RUN mode
  - ULPCLK: Main system clock for PD0 peripherals and PD0 bus, derived from MCLK
  - LFCLK: Fixed 32kHz clock, synchronized to MCLK/ULPCLK
  - HFCLK: High frequency external clock
  - HSCLK: High speed configurable clock for use by MCLK, sourced from SYSPLL or HFCLK
- Peripheral Specific Clocks
  - CANCLK: CAN-FD module functional clock
- External Clocks
  - CLK_OUT: External clock output with divider for pushing out a clock to external circuits

All clocks are disabled in SHUTDOWN mode.

3.4.3.1 MCLK (Main Clock) Tree

The MCLK is the main system clock and the root point of synchronization for all synchronized clocks (MCLK, CPUCLK, ULPCLK, and LFCLK). It is typically the highest speed clock in the system and supports operation up to 200MHz across the full temperature range of the device. The MCLK tree is the root source for the CPUCLK (in RUN mode), the PD1 high speed peripheral bus clock (in RUN and SLEEP modes), and the ULPCLK low power bus clock (in RUN, SLEEP, STOP, and STANDBY modes). In addition, the 32kHz LFCLK outputs are synchronized to MCLK.

The MCLK output to PD1 peripherals is enabled in RUN and SLEEP modes, and disabled in all other power modes. While the MCLK output to PD1 is disabled in STOP and STANDBY modes, the MCLK tree is still running to source ULPCLK and to provide synchronization for LFCLK.

The MCLK source is selected with a glitch free clock mux and can be changed dynamically at runtime by user software. It can also be changed automatically by hardware when entering STOP and STANDBY modes or during an asynchronous fast clock request.

The available sources for MCLK include:

- **HSCLK** (high-speed clock) at up to 200MHz which can be sourced by:
  - **SYSPLLCLK0** (used to reach the max MCLK speed of 200MHz)
  - **HFCLK** for applications where the main clock needs to be as accurate as possible
- **SYSOSC** at 32MHz
- **LFCLK** at 32kHz for applications where the entire system, including the CPU, runs at 32kHz with low peak operating current

Note

Set **SYSPLLCLK0** as the source of HSCLK before setting HSCLK as the clock source of MCLK, otherwise the device can be in an unpredictable state.

Using MCLK in RUN and SLEEP Mode

After boot, MCLK is sourced from SYSOSC by default. The decision of which oscillator to use to source MCLK is important because MCLK sets both the CPUCLK frequency and the bus clock frequency for PD1 peripherals. As a result, the accuracy and the clock speed of the oscillator selected for MCLK must be appropriate not only for the operation of the CPU but also for the operation of the PD1 peripherals that use the bus clock as their functional clock.

The clock source and frequency selection decisions made for MCLK also affect ULPCLK in RUN and SLEEP modes. See the ULPCLK section for more information on how MCLK and ULPCLK are related in RUN and SLEEP mode.

Using MCLK in STOP and STANDBY Mode

In STOP and STANDBY modes, the MCLK output to PD1 peripherals is disabled, but the ULPCLK, which is the bus clock for PD0 peripherals, is still active in STOP and optionally active in STANDBY. See the ULPCLK section for more information on how the MCLK source and ULPCLK are related in STOP and STANDBY mode.

MCLK Source Selection

Application software can change the MCLK source from SYSOSC to the SYSPLL or to the HFCLK (which is either XTAL or HFCLK_IN) by configuring the MCLKCFG.USEHSCLK and HSCLKCFG.HSCLKSEL register bits appropriately in SYSCTL. The following table gives the proper register bit configurations for selecting different clocks for MCLK in RUN and SLEEP modes.

Table 3-11. AM13E230x MCLK Source Selection in RUN and SLEEP Mode

Desired Source	MCLKCFG.USEHSCLK	HSCLKCFG.HSCLKSEL
SYSOSC	0	X
SYSPLLCLK0	1	0
HFCLK	1	1

To switch MCLK from SYSOSC to HSCLK:

1. Verify that MCLK is sourced from SYSOSC (CLKSTATUS.HSCLKMUX is cleared).
2. Enable the desired high speed sources (SYSPLL, XTAL, HFCLK_IN) according to their respective requirements.
3. Select the desired HSCLK source through the HSCLKCFG.HSCLKSEL control
4. Verify that CLKSTATUS.HSCLKGOOD is set, indicating that the selected HSCLK source is valid.
5. Set MCLKCFG.USEHSCLK to switch MCLK to HSCLK.

To switch MCLK from HSCLK to SYSOSC:

1. Verify that MCLK is sourced from HSCLK (CLKSTATUS.HSCLKMUX is set).
2. Clear MCLKCFG.USEHSCLK to switch MCLK to SYSOSC.
3. Wait for CLKSTATUS.HSCLKMUX to clear, indicating that MCLK is now sourced from SYSOSC.
4. If desired, disable any high-speed clock sources (SYSPLL or XTAL)

As shown in Table 3-11, the USELFCLK and USEHSCLK bits in MCLKCFG are mutually exclusive and must not be set at the same time.

Note

When MCLK is actively sourced by HSCLK (the HSCLKMUX bit in the CLKSTATUS register is set), the HSCLK source selection must not be changed (the HSCLKSEL bit in the HSCLKCFG register must not be changed). To change the HSCLK source, first switch MCLK to SYSOSC using the procedure given above, re-configure the HSCLK source, and then switch MCLK back to HSCLK.

3.4.3.2 CPUCLK (Processor Clock)

The processor clock (CPU clock) is always derived directly from MCLK and is active in RUN mode at the MCLK frequency. In all other power modes, CPUCLK is disabled.

3.4.3.3 ULPCLK (Low-Power Clock)

The ULPCLK is the bus clock for peripherals in the PD0 power domain. It supports operation up to 50 MHz and is derived directly from the MCLK tree through a clock divider (MCLK/4) which is enabled only when MCLK is sourced from a high-speed clock (SYSPLL, XTAL, or HFCLK_IN). The ULPCLK frequency is dependent on the MCLK dividers configuration and the selected power mode.

ULPCLK Behavior in RUN and SLEEP Modes

The PD0 power domain has a frequency limit of 50MHz in RUN and SLEEP modes. This frequency is derived from quarter domain i.e. MCLK/4 clock tree. As such, ULPCLK must be maintained to be ≤50MHz at all times. When MCLK is configured to run from SYSOSC, SYSCTL disables the MCLK/4 divider automatically and ULPCLK is sourced from MCLK as these clock sources are always ≤32 MHz.

However, when MCLK is configured to run from a high-speed clock (SYSPLL, XTAL, or HFCLK_IN), hardware shall switch back to MCLK/4 divider. The right configuration of this divider value is the responsibility of the application software to ensure that ULPCLK is ≤50 MHz in RUN and SLEEP modes by configuring it appropriately.

Note

MCLK/4 divider for the MCLK/4 domain is by default at /4. When MCLK < 100MHz and > 50MHz, then it can be configured to /2. When MCLK < 50MHz, then can be configured to /1. Application software shall ensure MCLK/2 and MCLK/4 dividers appropriately.

ULPCLK Behavior in STOP and STANDBY Modes

In STOP mode, the MCLK tree (and by extension, the ULPCLK) can run from SYSOSC with a 4MHz rate (if SYSOSCCFG.DISABLESTOP=0x0) or from LFCLK at 32kHz (if SYSOSCCFG.DISABLESTOP=0x1). When SYSOSC is used (SYSOSCCFG.DISABLESTOP=0x0), SYSCTL ensures that ULPCLK is always 4MHz even if SYSOSC is running at a higher frequency (due to user configuration or due to an asynchronous request from a peripheral).

In STANDBY mode, the ULPCLK runs from LFCLK to conserve power.

Table 3-12. AM13E230x ULPCLK by Operating Mode

Selected Power Mode	Configuration	Register Settings	ULPCLK Frequency
RUN or SLEEP (200MHz maximum)	MCLK source is SYSOSC	MCLKCFG.USEHSCLK=0x0, MCLKCFG.USELFCLK=0x0	ULPCLK is sourced from MCLK according to the MCLK configuration with $f_{ULPCLK} = f_{MCLK}$
	MCLK source is HSCLK (SYSPLL, XTAL, or HFCLK_IN)	MCLKCFG.USEHSCLK=0x1, MCLKCFG.USELFCLK=0x0	ULPCLK is sourced from MCLK according to the MCLK configuration with $f_{ULPCLK} = f_{MCLK}$/ Quarter Divider
STOP (4MHz maximum)	STOP with SYSOSC enabled	SYSOSCCFG.DISABLESTOP = 0x0	ULPCLK is sourced from SYSOSC with f_{ULPCLK} = 4MHz
STANDBY (32kHz maximum)	STANDBY with ULPCLK and LFCLK enabled	MCLKCFG.STOPCLKSTBY= 0x0	ULPCLK is sourced from LFCLK with $f_{ULPCLK} = f_{LFCLK}$ = 32kHz
SHUTDOWN (Off)	-	-	ULPCLK is off

3.4.3.4 LFCLK (Low-Frequency Clock)

LFCLK provides a continuous 32kHz clock to a variety of peripherals on the device. After a BOOTRST, LFCLK is sourced by the internal 32kHz oscillator (LFOSC).

LFCLK is active in RUN, SLEEP, STOP, and STANDBY power modes. It is possible to disable both ULPCLK and LFCLK together to most peripherals in STANDBY mode to achieve the lowest possible STANDBY mode power consumption (STANDBY1). To do so, set the STOPCLKSTBY bit in the MCLKCFG register in SYSCTL before entering STANDBY. In this state, a few times are the only clocked peripherals.

LFCLK is a synchronized clock. All LFCLK edges are synchronized to the main system clocks (MCLK and ULPCLK), meaning that the registers of peripherals clocked by LFCLK can be read or written to at any time without any special handling.

Note

When MCLK/ULPCLK are not sourced by LFCLK (for example, when they are sourced by SYSOSC) there is a 5 ULPCLK cycle synchronization delay between the low frequency clock source's clock edge and the corresponding LFCLK edge as seen by peripherals running from LFCLK.

When the MCLK/ULPCLK frequency is constant, this delay is constant and it does not add jitter to LFCLK. If the MCLK/ULPCLK frequency changes, the synchronization delay changes proportionally and this results in a small single-cycle LFCLK jitter at the MCLK/ULPCLK frequency transition point. This jitter changes the duty cycle of one LFCLK period, but there is no accumulation of error (there is never a change in the number of LFCLK periods, ensuring an accurate LFCLK time base for peripherals).

3.4.3.5 HFCLK (High-Frequency External Clock)

The high-frequency external clock (HFCLK) is the output of the high-frequency external clock selection mux, and can be selected as either the high-frequency oscillator (XTAL) output or the high-frequency digital clock input (HFCLK_IN).

The HFCLK can be used as the source for the following clocks:

- HSCLK source (to source MCLK)
- SYSPLL reference clock
- CANCLK source

The HFCLK is only available in RUN and SLEEP modes. HFCLK is automatically disabled by SYSCTL in all other modes, along with the XTAL itself.

3.4.3.6 HSCLK (High Speed Clock)

The high-speed clock (HSCLK) is the output of the high-speed clock selection mux, and can be selected to source MCLK by setting the USEHSCLK bit in the MCLKCFG register. The HSCLK is only a selection option for MCLK; it does not source any other functions. HSCLK can be configured to be sourced from the SYSPLL (SYSPLLCLK0) or the HFCLK (XTAL or HFCLK_IN). By default, the HSCLK is sourced from the SYSPLL. To change the HSCLK source to HFCLK, set the HSCLKSEL bit in the HSCLKCFG register.

The HSCLK is only available in RUN and SLEEP modes. It is automatically disabled by SYSCTL in all other modes, along with the SYSPLL and XTAL (if enabled).

SYSCTL will not switch MCLK to HSCLK, even if requested by software, if the HSCLK status check indicates that selected HSCLK source was not started correctly.

3.4.3.7 CANCLK (CAN-FD Functional Clock)

The CANCLK is a functional clock provided directly to the CAN-FD module from either the HFCLK (XTAL or HFCLK_IN) or the SYSPLL (SYSPLLCLK1). This clock is provided as a functional clock to the CAN-FD module asynchronous from the main clock (MCLK) for the highest possible accuracy. The CANCLK source is selected in SYSCTL by configuring the CANCLKSRC bit in the GENCLKCFG register. Additional CAN-FD clock configuration is provided within the CAN-FD peripheral itself (review the *CAN-FD* chapter for more detail).

3.4.3.8 External Clock Output (CLK_OUT)

A clock output unit is provided for sending digital clock signals from the device to external circuits or to the frequency clock counter. This feature is useful for clocking external circuitry such as an external ADC that does not have a clock source. The clock output unit has a flexible set of sources to select and includes a programmable divider.

Available clock sources for CLK_OUT:

- SYSPLLCLK0
- SYSPLLCLK1
- HFCLK

- SYSOSC
- ULPCLK
- LFCLK

The selected clock source can be divided by 1 (no divide), 2, 4, 6, 8, 10, 12, 14, or 16 before being output to the pin or to the frequency clock counter.

To use the clock output unit:

1. Configure IOMUX to select the CLK_OUT function on the device pin with CLK_OUT.
2. Select the desired clock source in the EXCLKSRC field of the GENCLKCFG register.
3. Set the desired clock divider, if necessary, in the EXCLKDIVVAL field of the GENCLKCFG register, and enable the divider by setting the EXCLKDIVEN bit. This must be done while EXCLKEN=0 (before CLK_OUT is enabled)
4. Enable the external clock output by setting the EXCLKEN bit in the GENCLKEN register.

Note

When the CLK_OUT source is selected as ULPCLK, the clock divider must be enabled (GENCLKCFG.EXCLKDIVEN must be set).

Note

When clearing the EXCLKEN bit to disable CLK_OUT, allow the clock source to run for 10 clock cycles to stabilize the EXCLKSRC mux.

Note

When disabling a clock source which is selected for CLK_OUT, it is recommended to disable the CLK_OUT function before disabling the clock source if it is important that CLK_OUT be logic low (0) when the clock source is disabled. If CLK_OUT is left enabled and the source for CLK_OUT is disabled, it is possible that CLK_OUT may stop in a logic high (1) state.

Note

When the CLK_OUT source is selected as SYSPLLCLK0 or SYSPLLCLK1, the SYSPLL output must be ≤48MHz. Further speed restrictions can exist depending on the IO capabilities of a specific device and pin; see the Digital IO specifications in the device-specific data sheet for details on supported IO speeds.

3.4.4 Clock Monitors

Several hardware clock monitors are provided to ensure that the CKM is functioning properly. Clock faults are processed through SYSCTL and result in either a brownout reset (in the event of a fatal fault) or a SYSCTL interrupt.

Table 3-13. AM13E230x CKM Clock Monitors

Clock Monitor	Description
MCLK	Asserts a fault if there is no MCLK activity for a period of 1-12 LFOSC cycles
LFOSC Startup	Indicates when LFOSC startup has completed and is available for use by peripherals
HFCLK Startup	Indicates if XTAL successfuly started within the specified startup time or if HFCLK_IN is stuck, before selecting HFCLK to source system functions
SYSPLL Startup	Indicates if SYSPLL successfully started, at which point the clock outputs from SYSPLL can be selected to source system functions
HSCLK Status	Indicates if the selected HSCLK source successfully started with a GOOD status or failed with a DEAD status

3.4.4.1 MCLK Monitor

A digital clock monitor can be used with MCLK. The MCLK monitor asserts an MCLK fault if there is no MCLK activity for a period of 1-12 LFCLK cycles. An MCLK fault is always considered fatal to the system and generates a BOOTRST.

The MCLK monitor can be enabled after the LFCLK is configured and running. To enable the MCLK monitor, set the MCLKDEADCHK bit in the MCLKCFG register in SYSCTL. When enabled, the MCLK monitor runs in all operating modes except for STANDBY1 and SHUTDOWN.

3.4.4.2 Startup Monitors

Clock startup monitors are provided for application software to check that that the LFOSC, XTAL/HFCLK_IN, SYSPLL, and HSCLK sources are alive before they are selected by software to be used to source a clock in the system. When a clock source has started successfully and is ready, a GOOD indication is given in the CLKSTATUS register in SYSCTL and an interrupt is generated. The startup monitors only provide a status indication when a related clock system configuration change is made. When an initial GOOD indication is given, the clock is not continuously monitored by the startup monitor. Continuous monitoring is provided for MCLK.

3.4.4.2.1 LFOSC Startup Monitor

The LFOSC is started automatically after a BOOTRST. Please refer to the device-specific data sheet for the LFOSC startup time. A startup monitor is provided to indicate to the application software when LFOSC startup has completed, at which time the LFCLK is available for use by peripherals. When LFSOSC startup has completed, the LFOSC startup monitor asserts the LFOSCGOOD bit in the CLKSTATUS register in SYSCTL and the LFOSCGOOD interrupt is asserted to alert the application.

3.4.4.2.2 HFCLK Startup Monitor

The XTAL takes time to start after being enabled. A startup monitor is provided to indicate to the application software if the XTAL has successfully started, at which point the HFCLK can be selected to source a variety of system functions. The HFCLK startup monitor also supports checking the HFCLK_IN digital clock input for a clock stuck fault.

To enable the HFCLK startup monitor, clear the HFCLKFLTCHK bit in the HFCLKCLKCFG register in SYSCTL (the default state is disabled).

When XTAL is started or the HFCLK_IN is selected as the HFCLK source, the HFCLKGOOD and HFCLKOFF bits in the CLKSTATUS register in SYSCTL are cleared.

In the case of XTAL being used, after the specified XTAL startup time has expired the XTAL status is tested. If the XTAL started successfully, the XTAL startup monitor asserts the HFCLKGOOD bit in the CLKSTATUS register and the HFCLKGOOD interrupt is also asserted. If the XTAL did not start within the specified startup time, the HFCLKOFF bit is set, indicating that the XTAL was dead at startup.

In the case of HFCLK_IN being used, after HFCLK_IN is selected a clock stuck check is performed. If the clock is alive, the HFCLKGOOD bit is set in the CLKSTATUS register and the HFCLKGOOD interrupt is also asserted. If the HFCLK_IN signal was stuck, the HFCLKOFF bit in CLKSTATUS register is set, indicating that the HFCLK_IN is dead.

If desired, checking of the HFCLK by the HFCLK startup monitor can be left disabled by keeping the HFCLKFLTCHK bit set in the HFCLKCLKCFG register in SYSCTL.

Note

The HFCLK must be in a stable state before attempting to enter STOP or STANDBY low power modes. Before entering STOP or STANDBY, make sure that either the HFCLKGOOD bit or the HFCLKOFF bit is set.

3.4.4.2.3 SYSPLL Startup Monitor

The SYSPLL takes time to start and settle after being enabled. A startup monitor is provided to indicate to the application software if the SYSPLL has successfully started, at which point the clock outputs from the SYSPLL can be selected to source a variety of system functions.

When the SYSPLL is started, the SYSPLLGOOD and SYSPLLOFF bits in the CLKSTATUS register in SYSCTL are cleared. After the startup/settling time has expired, the SYSPLL status is tested. If the SYSPLL started successfully, the SYSPLL startup monitor will assert the SYSPLLGOOD bit in the CLKSTATUS register and the SYSPLLGOOD interrupt will also be asserted. If the SYSPLL did not start within the specified time, the SYSPLLOFF bit will be set, indicating that the SYSPLL was dead at startup.

Note

The SYSPLL must be in a stable state before attempting to enter STOP or STANDBY low power modes. Before entering STOP or STANDBY, make sure that either the SYSPLLGOOD bit or the SYSPLLOFF bit is set.

3.4.4.2.4 HSCLK Status

The HSCLK is sourced from the HFCLK or the SYSPLL. The CLKSTATUS register in SYSCTL provides HSCLKGOOD and HSCLKDEAD indications, which indicate if the selected HSCLK source started successfully with a GOOD status or failed with a DEAD status, respectively. SYSCTL will not switch MCLK over to HSCLK, even if requested, if HSCLKGOOD is not set.

In addition, the HSCLKSOFF bit is provided in CLKSTATUS to indicate if all HSCLK sources (SYSPLL, HFCLK) are either off (disabled) or started with a DEAD status.

3.4.5 Frequency Clock Counter (FCC)

The frequency clock counter (FCC) enables flexible in-system testing and calibration of a variety of oscillators and clocks on the device. The FCC counts the number of clock periods seen on the selected source clock within a known fixed trigger period (derived from a secondary reference source) to provide an estimation of the frequency of the source clock.

Application software can use the FCC to measure the frequency of the following source oscillators and clocks (selected through the FCCSELCLK field in the GENCLKCFG register):

- MCLKDIV4
- SYSOSC
- HFCLK
- CLK_OUT
- SYSPLL (any of the two SYSPLL outputs)
- The external FCC input (FCC_IN)

The reference clock used to set the trigger time over which pulses of the source clock are counted is configurable (through the FCCTRIGSRC field in the GENCLKCFG register), and can be driven by:

- The external FCC input (FCC_IN)
- LFCLK

The trigger time period can be set in one of two ways (through the FCCLVLTRIG field in the GENCLKCFG register):

- Level triggered (one rising edge to one falling edge of the reference clock input)
- Rising-edge to rising-edge triggered, for a defined number of clock periods of the reference clock (selectable from 1 to 32 through the FCCTRIGCNT field in the GENCLKCFG register)

When the trigger source is selected as the external FCC input in level-triggered mode, a user-specified counting period can be set by applying a logic high pulse on the FCC_IN pin of the desired trigger length.

When the trigger source is selected as the LFCLK, using rising-edge to rising-edge triggering causes the FCC to capture the number of source clock pulses which occurred within 1 to 32 clock periods of the LFCLK (30.5µs).

The FCC counter is 22 bits and supports counting from 0 up to $2^{22} - 1 = 4{,}194{,}303$.

While the external FCC input (FCC_IN function) can be used as either the FCC clock source or the FCC trigger input, the signal cannot be used for both functions during the same FCC capture. FCC_IN must be configured as either the FCC clock source or the FCC trigger.

3.4.5.1 Using the FCC

Rising-Edge to Rising-Edge Triggered Mode with FCC_IN Trigger

The following steps describe how to use the FCC to count the number of source clock pulses within the trigger period set by the reference clock, with the FCC_IN pin being selected as the reference clock and the SYSOSC being selected as the source clock. This example would be useful for calibrating the SYSOSC frequency with respect to an accurate clock source provided to the FCC_IN pin externally.

1. Set the source clock to SYSOSC by configuring the FCCSELCLK field in the GENCLKCFG register.
2. Set the reference clock to FCC_IN by clearing the FCCTRIGSRC bit in the GENCLKCFG register.
3. Select rising-edge to rising-edge triggering by clearing the FCCLVLTRIG bit in the GENCLKCFG register.
4. Select the desired number of reference clock periods to count the source clock over in the FCCTRIGCNT field in the GENCLKCFG register.
5. Ensure that SYSOSC is enabled at the desired frequency, and that the external clock source connected to FCC_IN is constantly running before continuing.
6. Write the GO bit and KEY field to the FCCCMD register to start the FCC capture on the next trigger clock period.
7. Poll the FCCDONE status bit in the CLKSTATUS register. When the capture completes, FCCDONE will be set by hardware. FCCDONE is read-only and is automatically cleared by hardware when a new capture is started.
8. Extract the resulting count from the 22-bit DATA field in the FCC register.

Note

For an accurate count of FCC cycles, it is best to wait at least 6 pulses of FCC before setting FCCTRIGCNT=0x0

Rising-Edge to Rising-Edge Triggered Mode with LFCLK Trigger

The following steps describe how to use the FCC to count the number of source clock pulses within the reference clock period, with the LFCLK being selected as the reference clock and the SYSOSC being selected as the source clock. This example would be useful for calibrating the SYSOSC frequency with respect to an accurate 32.768kHz watch crystal.

1. Set the source clock to SYSOSC by configuring the FCCSELCLK field in the GENCLKCFG register.
2. Set the reference clock to LFCLK by setting the FCCTRIGSRC bit in the GENCLKCFG register.
3. Select rising-edge to rising-edge triggering by clearing the FCCLVLTRIG bit in the GENCLKCFG register.
4. Select the desired number of reference clock periods to count the source clock over in the FCCTRIGCNT field in the GENCLKCFG register.
5. Ensure that SYSOSC is enabled at the desired frequency, and that the LFCLK is running correctly before continuing.
6. Write the GO bit and KEY field to the FCCCMD register to start the FCC capture on the next trigger clock period.
7. Poll the FCCDONE status bit in the CLKSTATUS register. When the capture completes, FCCDONE will be set by hardware. FCCDONE is read-only and is automatically cleared by hardware when a new capture is started.

8. Extract the resulting count from the 22-bit DATA field in the FCC register. If SYSOSC was running at 32MHz and FCCTRIGCNT was set to '0' (one reference clock period), the result should be approximately 976 cycles counted within the single 32.768kHz period.
   a. To calibrate SYSOSC for 24MHz operation, the SYSOSC user trim must be adjusted until approximately 732 cycles are counted.
   b. To calibrate SYSOSC for 16MHz operation, the SYSOSC user trim must be adjusted until approximately 488 cycles are counted.

   In general, increasing the FCCTRIGCNT value increases the accuracy of the measurement, at the expense of longer measurement time.

Level Triggered Mode with FCC_IN Trigger and HFCLK_IN Clock

The following steps describe how to use the FCC to count the number of source clock pulses within one external reference pulse window, with HFCLK_IN being selected as the source clock. This example would be useful for measuring the frequency of an external clock source with respect to a fixed pulse width driven by an external signal.

1. Set the source clock to HFCLK by configuring the FCCSELCLK field in the GENCLKCFG register.
2. Set the trigger clock to the FCC_IN pin function by clearing the FCCTRIGSRC bit in the GENCLKCFG register.
3. Set level triggering by setting the FCCLVLTRIG bit in the GENCLKCFG register.
4. Ensure that IOMUX is configured for FCC_IN, that HFCLK is configured for HFCLK_IN, and that an external clock is sourcing HFCLK_IN.
5. Write the GO bit and KEY field to the FCCCMD register to start the FCC capture when FCC_IN goes logic high. Note that if FCC_IN is already logic high when GO is asserted, counting starts immediately. When using level mode, FCC_IN should be low when GO is set, and the trigger pulse should be sent to FCC_IN after GO is set.
6. Poll the FCCDONE status bit in the CLKSTATUS register. When the capture completes, FCCDONE will be set by hardware. FCCDONE is read-only and is automatically cleared by hardware when a new capture is started.
7. Extract the resulting count from the 22-bit DATA field in the FCC register.

3.4.5.2 FCC Frequency Computation and Accuracy

The frequency of the source clock can be computed after capture if the trigger time is known. The frequency is computed by dividing the number of source clock cycles captured by the trigger time. For example, if the trigger source was a 32.768kHz clock, the trigger mode was rising-edge to rising-edge, and the period count was 1, then the trigger time is one 32.768kHz clock period (30.5µs). If the captured source count were to come back as 122, the frequency of the source clock is computed as 122 divided by 30.5µs, giving a source clock frequency of approximately 3.99MHz.

$$f_{source} = FCC.DATA / ((GENCLKCFG.FCCTRIGCNT+1) / f_{ref}) \quad (9)$$

The FCC is accuracy is dependent on the trigger clock accuracy as well as the total number of clock cycles captured. The FCC intrinsic error is ≤2 source clock cycles per capture due to synchronization of the trigger to the source clock. Therefore, the impact of these two clock cycles is reduced as more cycles are counted (as the trigger time is increased and/or the source clock frequency is increased). Approximate intrinsic error of the FCC for various source clock frequencies captured against one 32.678kHz period (FCCTRIGCNT=0) and 32 clock periods (FCCTRIGCNT=31) are given in Table 3-14.

Table 3-14. FCC Error

Use Case (Source Clock Frequency)	FCC Trigger Time	FCC Count Result	FCC Count Uncertainty	Approximate FCC Intrinsic Uncertainty Error
4MHz source clock	30.5µs	122	2 cycles	1.6%

ADVANCE INFORMATION

Table 3-14. FCC Error (continued)

Use Case (Source Clock Frequency)	FCC Trigger Time	FCC Count Result	FCC Count Uncertainty	Approximate FCC Intrinsic Uncertainty Error	
	976.6µs	3906		0.05%	
32MHz source clock	30.5µs	976		0.20%	
	976.6µs	31250		0.01%	
80MHz source clock	30.5µs	2441		0.08%	
	200MHz source clock	30.5µs		6100	0.03%
976.6µs		195200		0.001%	

Note

When using the FCC_IN signal, it is recommended to have a fast slew rate of 10ns or less on the FCC_IN pin to minimize measurement uncertainty.

3.5 System Controller (SYSCTL)

The system controller (SYSCTL) contains all control logic for managing the configuration and state of the PMU and CKM analog circuitry. SYSCTL also provides reset management, NRST and JTAG/SWD pin mux control, flash bank swap control, and the ECC error handling for flash and SRAM.

All power, clock, and reset configuration is done through the SYSCTL memory-mapped register interface.

3.5.1 Resets and Device Initialization

The SYSCTL manages device reset levels and device initialization.

3.5.1.1 Reset Levels

The device has five reset levels:

1. Power-On Reset (POR)
2. Brownout Reset (BOR)
3. Boot Reset (BOOTRST)
4. System Reset (SYSRST)
5. CPU Reset (CPURST)

The general flow and relationship between reset levels is provided in the diagram below.

ADVANCE INFORMATION

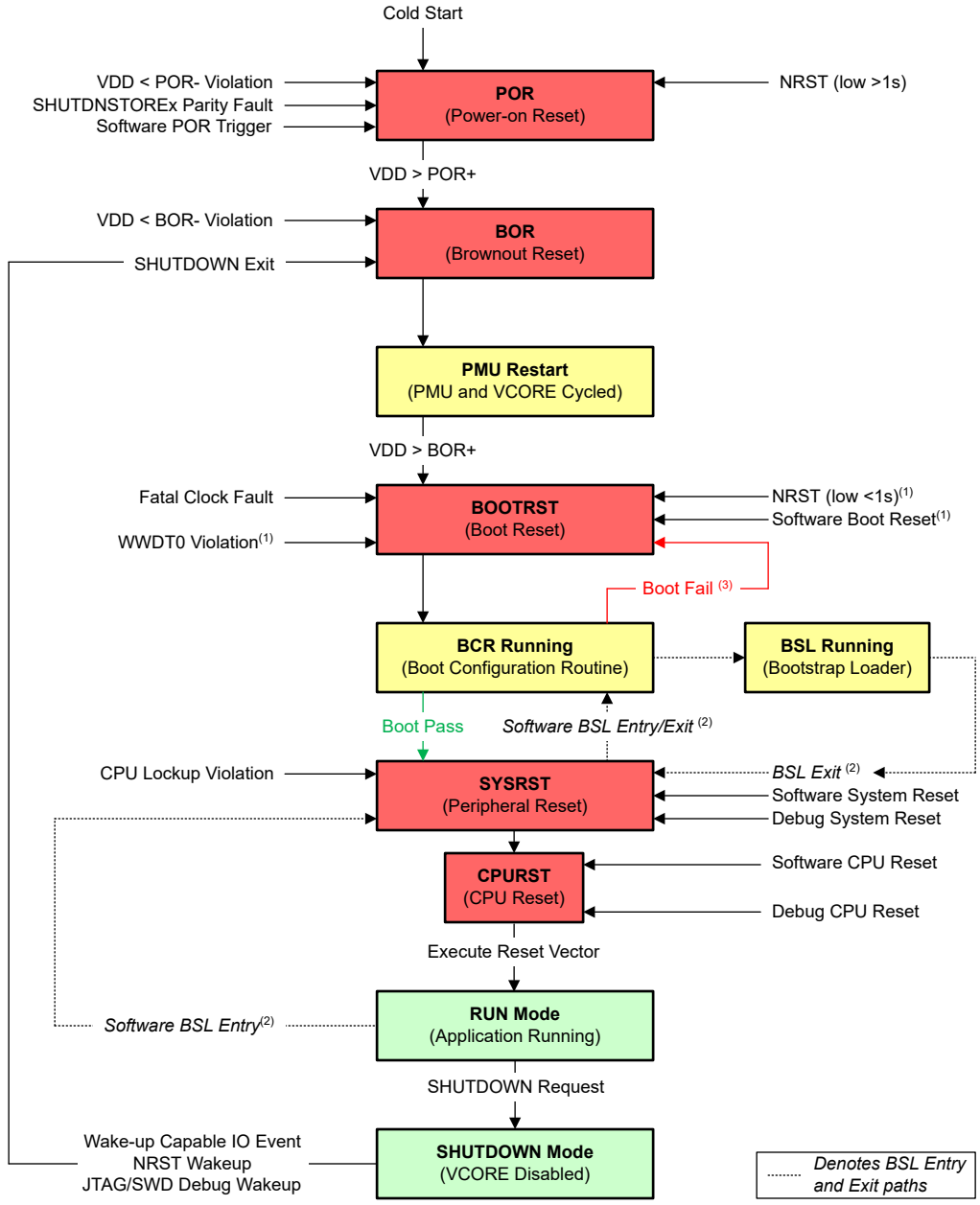


Figure 3-9. Device Reset Levels

(1)An NRST (low <1 s), software boot reset, or WWDT0 violation triggered BOOTRST runs the boot configuration routine.

(2)A software-triggered bootstrap loader (BSL) entry command first triggers a SYSRST, after which the boot configuration routine (BCR) is run to authenticate the BSL entry before starting the BSL. After the BSL execution concludes, a SYSRST is generated and the BCR executes again. When the BCR completes, a final SYSRST is issued and the application is started.

(3)If a boot fail occurs during execution of the boot configuration routine, a BOOTRST can be generated by SYSCTL to attempt the boot process again from the BOOTRST level.

ADVANCE INFORMATION

Note

SLEEP, STOP, and STANDBY operating modes are not shown in this diagram. These modes originate from and return to RUN mode unless an exception occurs which causes a reset level to be asserted or a mode to be suspended.

3.5.1.1.1 Power-on Reset (POR) Reset Level

A power-on reset (POR) is a complete device reset.

The following conditions generate a POR:

- The device powers up (cold start)
- A POR-supply monitor violation(VDD drops below the POR supply monitor negative-going threshold)
- A parity fault on PMU trim data or the shutdown memory
- Software triggers a POR through SYSCTL *(RESETLEVEL 0x03)*
- The NRST pin is held low for more than one second when in NRST mode

A POR always resets the shutdown memory, re-enables the NRST/JTAG/SWD pin functions (if disabled), and triggers a BOR.

3.5.1.1.2 Brownout Reset (BOR) Reset Level

A brownout reset (BOR) resets the device power management unit (PMU). All regulated core logic powered from VCORE are power cycled.

The following conditions generate a BOR:

- A POR.
- A BOR- Supply Monitor Violation (VDD drops below the BOR- supply monitor lower threshold voltage level).
- An exit from SHUTDOWN mode (through a wakeup-capable IO, NRST, or SWD) if SHUTDOWN mode is active.

The following are not reset by a BOR:

- The shutdown memory (SHUTDNSTOREx) if SHUTDOWN mode is present in the device
- The NRST state, if disabled by software.
- The JTAG/SWD state, if disabled by software.
- The latched IO pin state, when SHUTDOWN mode is active, and the cause of the BOR is an exit from SHUTDOWN mode (see SHUTDOWN mode handling).

A BOR always asserts a BOOTRST when VDD > BOR+ (brownout supply monitor upper threshold voltage level).

3.5.1.1.3 Boot Reset (BOOTRST) Reset Level

A boot reset (BOOTRST) triggers execution of the device boot configuration routine and resets the majority of the core logic, system memory (SRAM), and SYSOSC FCL mode (if enabled).

The following conditions generate a BOOTRST:

- A BOR.
- A fatal clock failure (see LFCLK Monitor and MCLK Monitor).
- A WWDT0 violation.
- Software triggers a BOOTRST through SYSCTL*(RESETLEVEL 0x01)*
- The NRST pin is held low for longer than the minimum reset pulse time but less than one second when in NRST mode. Refer to the device data sheet for reset timing requirements.
- A BOOTRST followed by a boot failure (re-attempt of a failed boot sequence).

The following are not reset by a BOOTRST:

- The shutdown memory (SHUTDNSTOREx) if SHUTDOWN mode is present.
- The NRST disable state, if disabled by software.
- The JTAG/SWD disable state, if disabled by software.

Following a BOOTRST, a SYSRST is always triggered upon successful completion of the boot configuration routine. If the boot configuration routine fails to complete successfully, a BOOTRST is again generated and the boot process is attempted again from the BOOTRST point. The boot process attempts to complete successfully up to 3 times, after which the device state locks until a BOR or POR reset occurs (see Section 3.5.1.8).

3.5.1.1.4 System Reset (SYSRST) Reset Level

A system reset clears the state of the CPU and all the peripherals, with the exceptions listed below.

The following conditions generate a SYSRST:

- A BOOTRST followed by a boot pass
- Software triggers a SYSRST with BSL entry through SYSCTL *(RESETLEVEL 0x02)*
- A bootstrap loader (BSL) exit (if present), which is always followed by execution of the boot configuration routine (BCR)
- A CPU lockup violation
- Software triggers a SYSRST through SYSCTL *(RESETLEVEL 0x00)*
- The debug subsystem triggers a system reset

The following are not reset by a SYSRST:

- The shutdown memory (SHUTDNSTOREx) (if present)
- The NRST state, if disabled by software
- The JTAG/SWD state, if disabled by software
- The SYSOSC frequency correction loop (FCL), if enabled by software

In most cases, the device is in RUN mode after a SYSRST, and the CPU executes the reset vector and begins execution of the application software.

There are exceptions to this:

- If the SYSRST was triggered with a BSL entry request, the BCR runs followed by the BSL.
- If the SYSRST was triggered due to a BSL exit request, the BCR runs, followed by an additional SYSRST, after which the application software starts.

3.5.1.1.5 CPU-only Reset (CPURST) Reset Level

A CPU-only reset clears the state of the CPU logic only. Peripheral states are not affected by a CPU reset. A CPU reset is only generated by software through the CPU AIRCR local register or by the debug subsystem.

3.5.1.2 Initial Conditions After Power-Up

After a POR, when the boot process completes, the initial device conditions are as follows:

- The NRST pin is always configured in NRST mode.
- JTAG IO are in JTAG mode. (TCK is pulled low, and TMS/TDI/TDO are pulled high.
- All other configurable I/O pins are high impedance (Hi-Z).
- Peripheral modules are disabled and reset as described in the related chapters of this manual.
- The device is in RUN operating mode.
- MCLK is sourced from the internal SYSOSC at base frequency (32MHz).
- LFCLK is sourced from the internal LFOSC
  - Note: LFOSC requires some time to start up before LFCLK can be used. Refer to the device data sheet for LFOSC timing requirements.
- SYSPLL is disabled.
- XTAL (Crystal Oscillator) is disabled.
- Any flash sectors configured to be write protected at boot are write protected.

3.5.1.3 NRST Pin

NRST is a dedicated (functional) pin and must be high for the device to boot successfully. This is achieved with an internal pullup resistor on the NRST pin.

After the device is started, a low pulse on NRST <1 second in duration triggers a BOOTRST. If a low pulse on NRST is held for >= 1 second, a POR is triggered.

Note

NRST Low Pulse **< 1 Second** = **BOOTRST** (Boot Reset)

Note

NRST Low Pulse **>= 1 Second** = **POR** (Power-on Reset)

Additionally, the NRST pin has an associated glitch filter circuit which prevents low pulses less than a certain duration from resetting the device. Refer to the device data sheet for the glitch filter and pull-up resistor specifications.

3.5.1.4 SWD/JTAG Pins

There are four JTAG/SWD (Serial Wire Debug) pins present on all devices: JTDI, JTCK_SWCLK, JTMS_SWDIO, and JTDO_SWO.

After a cold start, the pins are configured in JTAG/SWD mode to allow a debug connection to be established.

- JTDI pin is configured in the IOMUX and the **pull-up** is enabled.
- JTDO_SWD pin is configured in the IOMUX and the pull resistor is **disabled**.
- JTMS_SWDIO pin is configured in the IOMUX and the **pull-up** is enabled.
- JTCK_SWCLK pin is configured in the IOMUX and the **pull-down** is enabled.

The user is able to re-configure the JTAG/SWD pins as general purpose IO (GPIO) or alternative mux mode signal in software to enable use of these pins in an application when debug support is no longer required. To disable SWD functionality, set the DISABLE bit in the SWDCFG register in SYSCTL along with the KEY. Then configure IOMUX for the desired functionality.

After the JTAG/SWD pin functions are disabled, JTAG/SWD can only be re-enabled after a POR has occurred.

3.5.1.5 Generating Resets in Software

Software can generate a software POR, a software BOOTRST, a software SYSRST with bootstrap loader (BSL) entry, or a software SYSRST by issuing the appropriate command to SYSCTL. To issue a reset, first select the desired reset level in the RESETLEVEL register in SYSCTL. Then set the GO bit in the RESETCMD register along with the KEY value.

Table 3-15. Software Generated SYSCTL Reset Commands

LEVEL	Action
0x0	Software-triggered SYSRST
0x1	Software-triggerd BOOTRST
0x2	Software-triggered SYSRST with BSL entry
0x3	Software-triggered POR
0x4	Software-triggered CPURST

A CPU-only reset (CPURST) which does not reset the peripherals can also be triggered in software within the device CPU by setting the SYSRESETREQ bit in the AIRCR local CPU register. See the CPU Sub System chapter for more information.

Starting the BSL From Software

The software-triggered BSL entry *(RESETLEVEL 0x02)* is a special case of the SYSRST which provides a mechanism for the application software to start the ROM bootstrap loader (BSL). Direct access to the bootloader code is disabled during normal software execution in RUN mode. When application software commands a software-triggered BSL entry *(RESETLEVEL 0x02)*, a SYSRST is generated first, followed by execution of the

ADVANCE INFORMATION

boot configuration routine (for authentication), after which the BSL is started (if the device security policy has enabled the BSL execution).

Once the BSL has completed execution, a second SYSRST is issued and the BCR is executed. When the BCR completes, a final SYSRST is asserted to return control of the system back to the application software.

If the BSL is disabled by the user configuration, and a software-triggered BSL entry is invoked, the device issues a standard SYSRST instead and returns control of the system back to the application software.

Any system configuration which is not reset by a SYSRST is maintained through this entire process.

3.5.1.6 Reset Cause

After a device reset occurs, the lowest level reset cause which occurred during reset processing is captured in hardware so that application software can interrogate the reason for the reset and take any appropriate action when starting the application. The lowest level reset cause is encoded into a 5-bit field in the reset cause register in SYSCTL. The contents of the reset cause register are always cleared upon a read, and return zero after being read if no reset has occurred after the read. The reset cause encoding is provided in the table below.

Table 3-16. Reset Cause Encoding

Reset				Device Modules Reset									
Reset Level	Cause ID		Reset Cause	NRST/JTAG/SWD Disables	SHUTDN STOREx	Core Regulator	Debug Subsystem	SRAM	BCR/BSL Execution	IOMUX	DMA, FLASHCTL, SYSCTL	Peripherals	CPU
N/A	0x00	0	No reset since last read										
POR	0x01	1	VDD < POR- Violation	R	R	R	R	R	R	R	R	R	R
			PMU Trim Parity Fault										
			SHUTDNSTOREx Parity Fault										
	0x02	2	NRST Pin Reset (=>1sec)	R	R	R	R	R	R	R	R	R	R
	0x03	3	Software-triggered POR	R	R	R	R	R	R	R	R	R	R
BOR	0x04	4	VDD < BOR- Violation			R	R	R	R	R	R	R	R
	0x05	5	Wake from SHUTDOWN			R	R	R	R	R[1]	R	R	R
	0x06-0x8	6-8	Reserved										
BOOTRST	0x09	9	Fatal Clock Fault					R	R	R	R	R	R
	0x0A-0x0B	10-11	Reserved										
	0x0C	12	NRST Pin Reset (< 1sec)					R	R	R[2]	R	R	R
	0x0D	13	Software-triggered BOOTRST					R	R	R[2]	R	R	R
	0x0E	14	WWDT0 Violation					R	R	R[2]	R	R	R
	0x0F	15	Reserved										
SYSRST	0x10	16	BSL Exit						R	R[2]	R[3]	R	R
	0x11	17	BSL Entry						R	R[2]	R[3]	R	R
	0x12-0x13	18-19	Reserved										
	0x14	20	Uncorrectable Flash ECC Error							R[2]	R[3]	R	R
	0x15	21	CPU Lockup Violation							R[2]	R[3]	R	R
	0x16-0x19	22-25	Reserved										

Table 3-16. Reset Cause Encoding (continued)

Reset				Device Modules Reset									
Reset Level	Cause ID		Reset Cause	NRST/JTAG/SWD Disables	SHUTDN STOREx	Core Regulator	Debug Subsystem	SRAM	BCR/BSL Execution	IOMUX	DMA, FLASHCTL, SYSCTL	Peripherals	CPU
	0x1A	26	Debug-triggered SYSRST							R[2]	R[3]	R	R
	0x1B	27	Software-triggered SYSRST							R[2]	R[3]	R	R
CPURST	0x1C	28	Debug-triggered CPURST										R
	0x1D	29	Software-triggered CPURST										R
	0x1E-0x1F	30-31	Reserved										

If two reset causes occur simultaneously, the lowest cause reset ID value is prioritized and reported. For example, if a WWDT0 violation (cause 0x0E) occurs at the same time that a VDD < BOR- violation (cause 0x04) occurs, the reported reset cause is a BOR- violation (cause 0x04), as this is a lower level reset which clears additional aspects of the device state.

The reset cause encoding enables simple software handling during application startup. The reset cause value can be read by application software and tested to be within a certain value range to determine if the following occurred:

- **RESETCAUSE == 0x00:** No reset since last read
- **RESETCAUSE < 0x04:** The NRST/JTAG/SWD disable state is reset and must be reconfigured.
- **RESETCAUSE < 0x04:** The SHUTDNSTOREx memory is reset and must be reconfigured.
- **RESETCAUSE < 0x08:** The regulated VCORE domain and id power cycled. The SRAM is reinitialized and the contents are lost.
- **RESETCAUSE < 0x1C:** The peripherals are reset and must be reconfigured.

3.5.1.7 Peripheral Reset Control

Each peripheral on a device contains a reset control register (RSTCTL) and a status register (STAT).

The STAT register is a read-only register which contains a RESETSTKY bit, indicating if the peripheral was reset. This bit can be read by application software to determine if a peripheral was reset and needs to be re-configured. The RESETSTKY bit is cleared by writing the RESETSTKYCLR bit together with the KEY value to the RSTCTL register.

Application software can also force a reset of the peripheral by writing the RESETASSERT bit together with the KEY value to the RSTCTL register. This action resets the peripheral to the default state and sets the RESETSTKY bit in the STAT register.

Note

The RSTCTL register does not reset the FPUB and FSUB registers for a given peripheral. Use SYSRST or directly modify FPUB and FSUB directly of the peripheral to reset the publisher and subscriber event registers.

3.5.1.8 Boot Fail Handling

If a boot fails during execution of the boot configuration routine (BCR), SYSCTL asserts a BOOTRST to attempt another boot. A boot fail can be caused by the following:

- Boot configuration data integrity error

ADVANCE INFORMATION

- Device trim integrity error
- BCR timeout (BCR takes significantly longer than expected to complete for any other reason)

Up to three attempts to successfully boot the device are made by hardware. If the first, second, or third boot attempt is successful, the application starts normally. If the third attempt fails, then the boot process fails, no further boot attempts are made, and the application software is not started.

The purpose of the additional boot attempts is to allow the device to boot correctly if a transient (temporary) error was the cause of the boot fail. If three boot attempts are not successful, a steady-state error condition is likely present and the application is not started to prevent unexpected operation.

Note

If a device is locked due to three failed attempts to boot, and a BOR- violation occurs, a BOR and BOOTRST are still generated (by definition) and a single boot attempt is made. Under the same conditions, if power is completely removed from the device (triggering a POR- violation), then the device again attempts to boot up to three times.

3.5.2 Operating Mode Selection

The device operating mode is configured through the use of the following:

1. Policy bits in the SYSOSCCFG and MCLKCFG registers in SYSCTL (to control the behavior of SYSOSC in RUN, SLEEP, and STOP modes)
2. Policy bits in the PMODECFG register in SYSCTL (to set the deep sleep level of STOP, STANDBY or SHUTDOWN)
3. SLEEPDEEP policy bit in the SCR local CPU register (to select whether a WFI instruction triggers SLEEP mode or STOP/STANDBY/SHUTDOWN mode)
4. Use of the Arm® WFI (wait for interrupt) CPU instruction (to enter the configured SLEEP/STOP/STANDBY/SHUTDOWN state)

Before entering an operating mode where the CPU is disabled, make sure that the appropriate peripheral that can wake the CPU from sleep has been configured to generate a CPU interrupt on the desired event.

For a detailed description of the behavior of each operating mode, see the associated operating mode section.

Policy Bit Configuration

Table 3-17 defines how to configure the relevant policy bits for each operating mode. All values are indicated in binary format. A dash (-) indicates that the particular policy bit is a don't care for the specified operating mode.

Table 3-17. Operating Mode Policy Bit Configuration

Operating Mode Policy Control		RUN	SLEEP (1)	STOP	STANDBY		SHUTDOWN
Register	Bit	RUN0	SLEEP0	STOP0	STANDBY0	STANDBY1	
MCLKCFG	STOPCLKSTBY	-	-	-	0	1	1
PMODECFG	DSLEEP	-	-	00	01	01	10
SCR	SLEEPDEEP	0	0	1	1	1	1

(1) SLEEP mode behavior is always identical to RUN mode, except with the CPUCLK disabled. As such, the SLEEP behavior is determined by the configuration of RUN mode.

ADVANCE INFORMATION

Entering SLEEP Mode

Entering **SLEEP** mode disables the CPU, but otherwise maintains the same configuration as **RUN**. To enter **SLEEP** mode:

1. Configure the CPU for SLEEP by clearing the SLEEPDEEP bit in the local SCR register.
2. Enter selected mode by executing the WFI (wait for interrupt) CPU instruction.

Entering STOP or STANDBY Modes

To enter **STOP** or **STANDBY** mode:

1. Configure the PMODECFG register in SYSCTL to 0b00 (STOP) or 0b01 (STANDBY).
2. Configure the CPU for DEEP SLEEP by setting the SLEEPDEEP bit in the local SCR register.
3. Enter selected mode by executing a WFI (wait for interrupt) CPU instruction.

Entering SHUTDOWN Mode

To enter **SHUTDOWN** mode:

1. Configure the PMODECFG register in SYSCTL to 0b10 (SHUTDOWN).
2. Configure the CPU for DEEP SLEEP by setting the SLEEPDEEP bit in the local SCR register.
3. Enter selected mode by executing a WFI (wait for interrupt) CPU instruction.

3.5.3 Asynchronous Fast Clock Requests

The GPIO peripheral is configured to asynchronously assert a hardware request to the SYSCTL for a fast clock source, even if the device is operating in STOP or STANDBY mode. This mechanism for applications where the MCLK/ULPCLK tree is normally sourced from either LFCLK (at 32kHz) or SYSOSC, but a faster clock is temporarily needed to quickly handle a GPIO peripheral event.

Asynchronous fast clock requests are also useful for scenarios where the device is running in STANDBY1 mode. In STANDBY1 (when STOPCLKSTBY is set), the ULPCLK and LFCLK are disabled to all peripherals. To wake up the device from this state where the bus clock (ULPCLK) is disabled, a GPIO interrupt request forces an asynchronous fast clock request to wake the device to RUN mode.

Asynchronous fast clock requests temporarily provides the GPIO peripheral with bus clock (MCLK/ULPCLK), sourced from the SYSOSC, for the duration of the request.

Asynchronous Fast Clock Behavior

When configured, SYSCTL responds to a peripheral fast clock request in the following way:

1. If the device is currently in a STOP or STANDBY mode, the low power state is temporarily suspended to support running the bus clock (ULPCLK) at the SYSOSC base frequency.
2. If disabled, SYSOSC is forced to be enabled; if SYSOSC is already running but at a different frequency than base frequency, SYSOSC is forced to base frequency.
3. The MCLK/ULPCLK tree is forced to be sourced from SYSOSC at the base frequency; if the device is in RUN mode then the CPUCLK is also switches to the SYSOSC rate (the CPUCLK is always derived from MCLK).

After the configuration above is applied, values are held for the duration of time that the asynchronous request remains asserted plus approximately 1µs after the request is removed. The system then returns to the configuration which existed before the fast clock request, provided the CPU did not change the configuration during the request.

Asynchronous fast clock requests are ignored and results in no effect on the device configuration if any of the following are true:

- MCLK is already sourced from SYSOSC at base frequency
- Asynchronous fast clock requests are globally blocked by setting the BLOCKASYNCALL bit in the SYSOSCCFG register in SYSCTL

Peripheral Support

The GPIO peripheral provides support for generating an asynchronous fast clock request. The purpose, request source, and configuration requirements are given in #unique_123/unique_123_Connect_42_TABLE_SPY_TC3_54B.

Table 3-18. Peripheral Support for Asynchronous Fast Clock Requests

Peripheral	Purpose	Request Source	Configuration
GPIO	Fast CPU wake from GPIO event	GPIO activity	The GPIO generates an asynchronous fast clock request through the GPIO configuration registers. This is for applications where GPIO wake from STANDBY mode is desired, as the fast clock request results in the GPIO digital glitch filters running at SYSOSC base frequency. In addition to configuring the GPIO registers to request the fast clock, the BLOCKASYNCALL bit must be cleared in the SYSOSCCFG register to allow the request to propagate.

Fast CPU Event Handling

In addition to the GPIO fast clock request triggers, SYSCTL can be configured to generate an asynchronous fast clock request upon any IRQ to the CPU. This provides the lowest latency interrupt handling when the system is running at the LFCLK rate (32kHz), as the IRQ request propagates through the wake-up logic at the SYSOSC rate instead of the LFCLK rate (32kHz). When the FASTCPUEVENT bit is set in the SYSOSCCFG register in SYSCTL, any interrupt request to the CPU also generates a fast clock request.

Asynchronous Fast Clock Request Logic

The logic for asserting a fast clock request is given in the following figure.

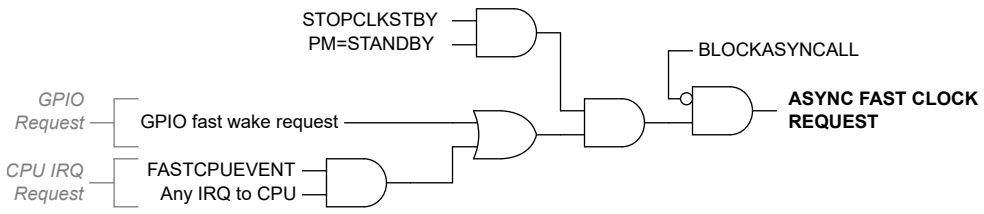


Figure 3-10. Asynchronous Fast Clock Request Logic

3.5.4 SRAM Write Protection

SRAM Write Protection Control for this device is implemented in the Global Security Controller (GSC). Please refer to the related chapter for more information.

3.5.5 Flash Wait States

Refer to the *Recommended Operation Conditions* section of the device specific data sheet to determine the max clock frequency supported for valid wait state configurations.

Additional details can be found in the FLASH chapter.

ADVANCE INFORMATION

3.5.6 Flash Bank Address Swap

Devices with multiple flash banks provide a mechanism for swapping the address space of the upper banks with the address space of the lower banks to enable firmware updates, where the firmware image has no knowledge of the containing memory bank, to be able to execute properly on the hardware.

To swap the addresses of the upper flash banks with the addresses of the lower flash banks, set the USEUPPER bit in the FLBANKSWAP register while also writing the KEY value.

Special care must be taken when swapping the flash bank address space. See the nonvolatile memory system chapter for additional considerations when using flash bank address swapping.

3.5.7 Shutdown Mode Handling

When the device is configured to enter SHUTDOWN mode, the core regulator is powered down and the device register contents and SRAM contents are lost. An exit from SHUTDOWN mode generates a BOR level reset. Two mechanisms are provided to preserve the device state when entering SHUTDOWN mode: IO latching and a small shutdown memory.

Shutdown IO State

The digital IO pin states (output low/high, pullup/pulldown, Hi-Z, drive configuration) are latched and retained upon entry to SHUTDOWN for pins supporting Wakeup functionality (refer to the device data sheet for the list of WAKEUP pins). After exiting SHUTDOWN mode, the IOs are held in the previous state until released by application software setting the RELEASE bit in the SHDNIOREL register along with the matching KEY value. When exiting SHUTDOWN mode, application software must first re-configure the IO to the proper state, then release the IO. To determine at startup if an exit from SHUTDOWN mode caused the reset, application software must read the RSTCAUSE register in SYSCTL.

Note

When exiting SHUTDOWN, **the JTAG and serial wire debug (SWD) pins also remain locked until application software sets the RELEASE bit**. As a result, a debug connection cannot establish when waking up from SHUTDOWN mode until the IO are released by application software.

Note

When exiting SHUTDOWN, the bootstrap loader (BSL) invoke pin must be held at a logic low level to prevent unintended entry into the BSL during exit from SHUTDOWN. An entry to the BSL during SHUTDOWN exit prevents the application code from starting. The BSL interfaces are not be available, and a SWD connection is possible since the IO states remain latched through SHUTDOWN exit until application software releases the IOs.

Shutdown Memory

To enable saving of application state information before entering SHUTDOWN mode, 4 bytes of shutdown memory are provided in SYSCTL. These memory locations are retained in SHUTDOWN mode and are readable by the application after exiting SHUTDOWN. To save data to the SHUTDOWN memory, write to the SHUTDNSTORE0-SHUTDNSTORE3 registers in SYSCTL.

3.5.8 Configuration Lockout

Configuration registers in SYSCTL can be locked out from writes to add a layer of robustness against unintended changes to the PMCU at runtime. To lock out the configuration registers from writes, set the ACTIVE bit in the WRITELOCK register in SYSCTL.

All SYSCTL registers are protected by the WRITELOCK functionality except for those listed below:

- WRITELOCK
- PMODECFG
- FCC, FCCCMD

- FLBANKSWAP
- RSTCAUSE (read-to-clear), RESETLEVEL, RESETCMD
- SHDNIOREL
- SHUTDNSTOREx

In addition to the overall SYSCTL configuration write lock feature, many SYSCTL registers also require a KEY value to be written in conjunction with the desire configuration data for the write to take effect.

3.5.9 System Status

The status of various aspects of the PMCU can be polled by software by reading the CLKSTATUS and SYSSTATUS registers in SYSCTL.

Checking Clock Status (CLKSTATUS)

The CLKSTATUS register in SYSCTL is a read-only register which indicates the current configuration and status of the clock module. Key status information provided in CLKSTATUS includes:

- The current SYSOSC frequency
- The current HSCLK selection
- The current MCLK selection
- The HFCLK and SYSPLL status
- The LFOSC status
- Indications that the HSCLK, HFCLK, and SYSPLL are disabled
- Error indications if a peripheral requested a clock and the clock cannot be generated

This status information is useful to validate that a requested clock change has completed successfully, or to check the true SYSOSC frequency in applications where SYSOSC can have asynchronous activation or frequency requests issued by peripherals.

Checking System Status (SYSSTATUS)

The SYSSTATUS register in SYSCTL is a read-only register which indicates flash ECC errors (SED and DED) along with other peripheral-specific status information. ECC error bits in SYSSTATUS are sticky (bits remain set when an ECC error occurs even if future reads do not have errors). These bits can be reset (cleared) by setting the ALLECC bit in the SYSSTATUSCLR register along with the KEY value.

3.5.10 Error Handling

The device includes several diagnostic mechanism to detect errors at runtime. The table below lists the error sources and corresponding handling mechanism.

Note

Not all devices support all diagnostic features. For example, some devices do not have ECC/parity on memories and some devices do not have dual watchdog timers. Always refer to the device-specific data sheet to understand which diagnostic features are available for a given device.

Table 3-19. Error Sources and Handling Mechanisms

Error Source	Error	Handling Mechanism
Flash	Non-correctable ECC Error	• For a CPU or DMA request, a FLASHDED nonmaskable interrupt is generated to the processor or a SYSRST is generated depending on configuration of the FLASHECCRSTDIS bit. • The FLASHDED sticky bit is set in the SYSSTATUS register in SYSCTL.
	Correctable ECC Error	• A FLASHSEC interrupt is also generated in SYSCTL. • The FLASHSEC sticky bit is set in the SYSSTATUS register in SYSCTL.

ADVANCE INFORMATION

Table 3-19. Error Sources and Handling Mechanisms (continued)

Error Source	Error	Handling Mechanism
SRAM	Parity error	• Nonmaskable interrupt is generated to the processor if from CPU request. • DMA data error interrupt is generated if from DMA request.
	Address error on CPU access	• A hard fault is generated in the CPU.
	Address error on DMA access	• A DMA address error interrupt is generated in the DMA controller.
	ECC error on CAN SRAM	• An interrupt is generated in the MCAN peripheral.
SHUTDNSTORExMemory	Parity error	• A POR is generated.
CKM	MCLK failure	• A BOOTRST is generated.
CPUSS	Memory protection unit violation	• A hard fault is generated in the CPU.
WWDT	WWDT0 violation	• A BOOTRST is generated or a nonmaskable interrupt is generated in the SYSCTL NMI registers depending on configuration of the WWDTLP0RSTDIS bit.
PMU	Trim parity error	• A POR is generated.
	POR0- supply error	• A POR is generated.
	BOR- supply error	• A BOR is generated.
CPUSS	Memory protection unit violation (if present)	• A hard fault is generated in the CPU.

Configurable NMI Triggers

Error sources can be configured to trigger either a nonmaskable interrupt (NMI) or a different handling mechanism. The SYSTEMCFG register in SYSCTL can be used to specify the desired error handling mechanism. For example, the WWDT0 can be configured to generate either a BOOTRST (default) or an NMI. Refer to the relevant SYSTEMCFG register for the available error handling options.

3.5.11 SYSCTL Events

The SYSCTL module contains two event publishers and no event subscribers. One event publisher manages SYSCTL interrupt requests (IRQs) to the CPU subsystem. The second publisher manages nonmaskable interrupts to the CPU subsystem for critical diagnostics.

The SYSCTL events are summarized in SYSCTL Events.

Table 3-20. SYSCTL Events

Event	Type	Source	Destination	Route	Configuration	Functionality
CPU interrupt	Publisher	SYSCTL	CPU Subsystem	Static route	SYSCTL interrupt registers	Fixed interrupt route from SYSCTL to CPU
CPU nonmaskable interrupt (NMI)	Publisher	SYSCTL	CPU Subsystem	Static route	NMI interrupt registers	Fixed interrupt route from SYSCTL to CPU

3.5.11.1 CPU Interrupt Events (CPU_INT)

The SYSCTL module provides interrupt sources which can be configured to source a CPU interrupt. In order of decreasing interrupt priority, the CPU interrupt events from the SYSCTL are given in Table 3-21.

ADVANCE INFORMATION

Table 3-21. SYSCTL CPU Interrupt Event Sources

Index (IIDX)	Name	Description
0 (Highest Priority)	NONE	No interrupt pending.
1	LFOSCGOOD	Indicates when LFOSC is prepare during startup, as LFOSC takes ≈1ms to start.
2	ANACLKERR	Indicates that an analog function is enabled and expected SYSOSC to be operating at a certain frequency, but SYSOSC is either not available or not operating at the required frequency.
3	FLASHSEC	Indicates that a flash memory one-bit correctable error is detected.
4	RESERVED	Reserved.
5	RESERVED	Reserved.
6	HFCLKGOOD	Indicates when the high frequency external clock (either the XTAL crystal oscillator or HFCLK_IN digital clock) are ready. This indication is useful when starting the clock system and waiting for HFCLK to be prepared before switching the MCLK source to HFCLK or before starting the SYSPLL with HFCLK as the SYSPLL reference.
7	SYSPLLGOOD	Indicates when the SYSPLL is ready and available for use. This indication is useful when starting the clock system and waiting for SYSPLL to be prepared before switching the MCLK source to SYSPLL.
8 (Lowest Priority	HSCLKGOOD	Indicates when the HSCLK (sourced by either HFCLK or a SYSPLL output) is ready. This indication is useful when waking up from STOP or STANDBY mode when HSCLK is configured as the MCLK source in RUN/SLEEP mode. When waking from STOP or STANDBY, MCLK operates from SYSOSC until the HSCLK is available, at which time SYSCTL automatically switches the MCLK source to the selected HSCLK source. This interrupt communicates that after wake-up from STOP/STANDBY, MCLK has properly switched to the correct HSCLK source and timing-sensitive peripherals sourced by MCLK can be used.

The CPU interrupt event configuration is managed with the SYSCTL IIDX, IMASK, RIS, MIS, ISET, and ICLR registers.

3.5.11.2 CPU Nonmaskable Interrupt (NMI) Events

The SYSCTL module provides nonmaskable interrupt sources which can be configured to source a nonmaskable interrupt event (NMI). In order of decreasing interrupt priority, the CPU events from SYSCTL are provided in the table below.

Table 3-22. SYSCTL CPU NMI Event Sources

Index (IIDX)	Name	Description
0 (Highest Priority)	NONE	No NMI pending.
1	RESERVED	Reserved.
2	WWDT0	A WWDT0 violation occurred.
3	SECURITY	Privileged/Unprivileged resource access ciolation (Flash/RAM/Peripherals) occurred.
4	FLASHDED	Indicates that a flash memory double-bit uncorrectable error is detected.
5	SRAMPAR	Indicates that an SRAM parity error is detected.
6	TMUROMPAR	Indicates that a TMU ROM parity error is detected.
7 (Lowest Priority)	SYSMEMACC	Indicates that a SYSMEM access error is detected.

The CPU nonmaskable interrupt event configuration is managed with the NMIIIDX, NMIRIS, NMIISET, and NMIICLR registers.

3.6 SYSCTL Registers

This Section describes the SYSCTL Registers.

3.6.1 SYSCTL Base Address Table

Table 3-23. SYSCTL Base Address Table

Bit Field Name		DriverLib Name	Base Address
Instance	Structure		
SysctlRegs	SYSCTL_REGS	SYSCTL	0x400A_F000

3.6.2 SYSCTL_REGS Registers

Table 3-24 lists the memory-mapped registers for the SYSCTL_REGS registers. All register offset addresses not listed in Table 3-24 should be considered as reserved locations and the register contents should not be modified.

Table 3-24. SYSCTL_REGS Registers

Offset	Acronym	Register Name	Section
800h	PWREN	IP Enable Register	Go
804h	RSTCTL	Power Control Register - Write Only Register, Always Read as 0	Go
814h	STAT	IP State Register - Read Only	Go
1020h	IIDX	SYSCTL interrupt index	Go
1028h	IMASK	SYSCTL interrupt mask	Go
1030h	RIS	SYSCTL raw interrupt status	Go
1038h	MIS	SYSCTL masked interrupt status	Go
1040h	ISET	SYSCTL interrupt set	Go
1048h	ICLR	SYSCTL interrupt clear	Go
1050h	NMIIIDX	NMI interrupt index	Go
1060h	NMIRIS	NMI raw interrupt status	Go
1070h	NMIISET	NMI interrupt set	Go
1078h	NMIICLR	NMI interrupt clear	Go
1100h	SYSOSCCFG	SYSOSC configuration	Go
1104h	MCLKCFG	Main clock (MCLK) configuration	Go
1108h	HSCLKEN	High-speed clock (HSCLK) source enable/disable	Go
110Ch	HSCLKCFG	High-speed clock (HSCLK) source selection	Go
1110h	HFCLKCLKCFG	High-frequency clock (HFCLK) configuration	Go
1120h	SYSPLLCFG0	SYSPLL reference and output configuration	Go
1124h	SYSPLLCFG1	SYSPLL reference and feedback divider	Go
1128h	SYSPLLPARAM0	SYSPLL PARAM0 (load from FACTORY region)	Go
112Ch	SYSPLLPARAM1	SYSPLL PARAM1 (load from FACTORY region)	Go
1130h	SYSPLLPARAM2	SYSPLL PARAM2 (load from FACTORY region)	Go
1134h	SYSPLLLDOCTL	SYSPLL LDO CTL (load from FACTORY region)	Go
1138h	SYSPLLLDOPROG	SYSPLL LDO VOUT PROG (load from FACTORY region)	Go
113Ch	GENCLKEN	General clock enable control	Go
1140h	GENCLKCFG	General clock configuration	Go
1144h	PMODECFG	Power mode configuration	Go
1148h	MLDOLPENCFG	LDO Configuration Control	Go
1150h	FCC	Frequency clock counter (FCC) count	Go
1154h	PMULDOSPARECTL	LDO Spare Control	Go
1158h	SYSCTL_ECO_REG1	Sysctl ECO Reg 1	Go
115Ch	SYSCTL_ECO_REG2	Sysctl ECO Reg 2	Go
1180h	SYSTEMCFG	System configuration	Go
1184h	SRAMCFG	System SRAM configuration	Go
1200h	WRITELOCK	SYSCTL register write lockout	Go
1204h	CLKSTATUS	Clock module (CKM) status	Go
1208h	SYSSTATUS	System status information	Go
1220h	RSTCAUSE	Reset cause	Go
1300h	RESETLEVEL	Reset level for application-triggered reset command	Go
1304h	RESETCMD	Execute an application-triggered reset command	Go

Table 3-24. SYSCTL_REGS Registers (continued)

Offset	Acronym	Register Name	Section
1310h	SYSOSCFCLCTL	SYSOSC frequency correction loop (FCL) ROSC enable	Go
131Ch	SHDNIOREL	SHUTDOWN IO release control	Go
1320h	EXRSTPIN	Disable the reset function of the NRST pin	Go
1324h	SYSSTATUSCLR	Clear sticky bits of SYSSTATUS	Go
1328h	SWDJCFG	Disable the SWD/JTAG function on the SWD/JTAG pins	Go
132Ch	FCCCMD	Frequency clock counter start capture	Go
1400h	SHUTDNSTORE0	Shutdown storage memory (byte 0)	Go
1404h	SHUTDNSTORE1	Shutdown storage memory (byte 1)	Go
1408h	SHUTDNSTORE2	Shutdown storage memory (byte 2)	Go
140Ch	SHUTDNSTORE3	Shutdown storage memory (byte 3)	Go
1410h	ADCSEQFRCGB	ADC Global Sequence Force	Go
1414h	ADCSEQFRCGBSEL	ADC Global Sequence Force Select	Go
1418h	M33SPARESOCLOCK1	M33C1 Spare SOC LOCK Reg 1	Go
141Ch	M33SPARESOCLOCK2	M33C1 Spare SOC LOCK Reg 2	Go
1420h	SYSCTL_READ_REG	Sysctl read only Reg	Go
1424h	PWREN_MCPERIPH	Register to control the power state	Go
1428h	RSTCTL_ASSERT_MCPERIPH	rstctl assert register to control reset assertion - Write Only Register, Always Read as 0	Go
142Ch	RSTCTL_CLEAR_MCPERIPH	rstctl clear register to control reset de-assertion - Write Only Register, Always Read as 0	Go
1430h	STAT_MCPERIPH	IP State Register - Read Only	Go
1434h	PWREN_SYSPERIPH	Register to control the power state	Go
1438h	RSTCTL_ASSERT_SYSPERIPH	rstcl assert Register - Write Only Register, Always Read as 0	Go
143Ch	RSTCTL_CLEAR_SYSPERIPH	rstctl clear register to control reset de-assertion - Write Only Register, Always Read as 0	Go
1440h	STAT_SYSPERIPH	IP State Register - Read Only	Go
1444h	CMPHPMXSEL	Bits to select one of the many sources on CompHP inputs. Refer to Pinmux diagram for details.	Go
144Ch	CMPLPMXSEL	Bits to select one of the many sources on CompLP inputs. Refer to Pinmux diagram for details.	Go
1450h	CMPHNMXSEL	Bits to select one of the many sources on CompHN inputs. Refer to Pinmux diagram for details.	Go
1454h	CMPLNMXSEL	Bits to select one of the many sources on CompLN inputs. Refer to Pinmux diagram for details.	Go
1458h	TSNSCFG	Temperature Sensor Config Register	Go
145Ch	TSNSSCTL	Temperature Sensor Control Register	Go
1460h	PGACONFIG	PGA Configuration Register	Go
1464h	REFCONFIGA	Reference Configuration Regsiter	Go
1468h	INTERNALTESTCTL	Internal Test Node Control Register	Go
146Ch	I2VCTL	I2V Logic Control	Go
1470h	ADCDACLOOPBACK	Not used in AM13	Go
1474h	XTALCR	XTAL Control Register	Go
1478h	XTALCR2	XTAL Control Register for pad init	Go
147Ch	X1CNT	x1cnt status register	Go
1480h	CMPSSCTL	CMPSS control register	Go
1484h	CMPSSDACBUFCONFIG	Config bits for CMPSS DAC buffer	Go

Table 3-24. SYSCTL_REGS Registers (continued)

Offset	Acronym	Register Name	Section
1488h	ANAREFCTL	Analog Reference Select	Go
148Ch	PERCLKCR	PWM Time Base Clock sync	Go
1490h	ADC_MMR_OVRD_CTL	ADC MMR Override control register for DFT: Control ADC enable override	Go
1494h	ADC_MMR_OVRD_VAL	ADC MMR Override value register for DFT : Value of ADC enable override	Go
1498h	VREGCONFIGDEBUG	VREG Configuration Debug Register	Go
149Ch	VREGCONFIGDFT	VREG Configuration DFT Register	Go
14A0h	AM13SPAREIREFENSOCLOCK	AM13 Spare IREFEN SOC LOCK Reg	Go
14A4h	AM13SPARESOCLOCK2	AM13 Spare SOC LOCK Reg 2	Go
14A8h	AM13SPARESOCLOCK3	AM13 Spare SOC LOCK Reg 3	Go
14ACh	AM13SPARESOCLOCK4	AM13 Spare SOC LOCK Reg 4	Go
2800h	PWREN	IP Enable Register	Go
2804h	RSTCTL	Power Control Register - Write Only Register, Always Read as 0	Go
2814h	STAT	IP State Register - Read Only	Go
4800h	PWREN	IP Enable Register	Go
4804h	RSTCTL	Power Control Register - Write Only Register, Always Read as 0	Go
4814h	STAT	IP State Register - Read Only	Go
000D0800h	PWREN	IP Enable Register	Go
000D0804h	RSTCTL	Power Control Register - Write Only Register, Always Read as 0	Go
000D0814h	STAT	IP State Register - Read Only	Go
000E8800h	PWREN	IP Enable Register	Go
000E8804h	RSTCTL	Power Control Register - Write Only Register, Always Read as 0	Go
000E8814h	STAT	IP State Register - Read Only	Go
000F0800h	PWREN	IP Enable Register	Go
000F0804h	RSTCTL	Power Control Register - Write Only Register, Always Read as 0	Go
000F0814h	STAT	IP State Register - Read Only	Go
000F2800h	PWREN	IP Enable Register	Go
000F2804h	RSTCTL	Power Control Register - Write Only Register, Always Read as 0	Go
000F2814h	STAT	IP State Register - Read Only	Go
000F4800h	PWREN	IP Enable Register	Go
000F4804h	RSTCTL	Power Control Register - Write Only Register, Always Read as 0	Go
000F4814h	STAT	IP State Register - Read Only	Go
000F6800h	PWREN	IP Enable Register	Go
000F6804h	RSTCTL	Power Control Register - Write Only Register, Always Read as 0	Go
000F6814h	STAT	IP State Register - Read Only	Go
00116800h	PWREN	IP Enable Register	Go
00116804h	RSTCTL	Power Control Register - Write Only Register, Always Read as 0	Go
00116814h	STAT	IP State Register - Read Only	Go

Table 3-24. SYSCTL_REGS Registers (continued)

Offset	Acronym	Register Name	Section
00180800h	PWREN	IP Enable Register	Go
00180804h	RSTCTL	Power Control Register - Write Only Register, Always Read as 0	Go
00180814h	STAT	IP State Register - Read Only	Go
00188800h	PWREN	IP Enable Register	Go
00188804h	RSTCTL	Power Control Register - Write Only Register, Always Read as 0	Go
00188814h	STAT	IP State Register - Read Only	Go
001B0800h	PWREN	IP Enable Register	Go
001B0804h	RSTCTL	Power Control Register - Write Only Register, Always Read as 0	Go
001B0814h	STAT	IP State Register - Read Only	Go
001B2800h	PWREN	IP Enable Register	Go
001B2804h	RSTCTL	Power Control Register - Write Only Register, Always Read as 0	Go
001B2814h	STAT	IP State Register - Read Only	Go
00630800h	PWREN	IP Enable Register	Go
00630804h	RSTCTL	Power Control Register - Write Only Register, Always Read as 0	Go
00630814h	STAT	IP State Register - Read Only	Go
00632800h	PWREN	IP Enable Register	Go
00632804h	RSTCTL	Power Control Register - Write Only Register, Always Read as 0	Go
00632814h	STAT	IP State Register - Read Only	Go
00634800h	PWREN	IP Enable Register	Go
00634804h	RSTCTL	Power Control Register - Write Only Register, Always Read as 0	Go
00634814h	STAT	IP State Register - Read Only	Go
00670800h	PWREN	IP Enable Register	Go
00670804h	RSTCTL	Power Control Register - Write Only Register, Always Read as 0	Go
00670814h	STAT	IP State Register - Read Only	Go
00672800h	PWREN	IP Enable Register	Go
00672804h	RSTCTL	Power Control Register - Write Only Register, Always Read as 0	Go
00672814h	STAT	IP State Register - Read Only	Go
00674800h	PWREN	IP Enable Register	Go
00674804h	RSTCTL	Power Control Register - Write Only Register, Always Read as 0	Go
00674814h	STAT	IP State Register - Read Only	Go

Complex bit access types are encoded to fit into small table cells. Table 3-25 shows the codes that are used for access types in this section.

Table 3-25. SYSCTL_REGS Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read

Table 3-25. SYSCTL_REGS Access Type Codes (continued)

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

1 PWREN Register (Offset = 800h) [Reset = 00XXXXXXh]

PWREN is shown in Figure 3-11 and described in Table 3-26.

Return to the Summary Table.

IP Enable Register

Figure 3-11. PWREN Register

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

Table 3-27. PWREN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	KEY	W	0h	KEY to Allow Enable State Change -- 0x26 26h = 0x26
23-1	RESERVED	R/W	XXXXh	
0	ENABLE	R/W	0h	IP Enable 0h = 0 1h = 1

ADVANCE INFORMATION

2 RSTCTL Register (Offset = 804h) [Reset = 00XXXXXXh]

RSTCTL is shown in Figure 3-12 and described in Table 3-27.

Return to the Summary Table.

Power Control Register - Write Only Register, Always Read as 0

Figure 3-12. RSTCTL Register

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

Table 3-29. RSTCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	KEY	W	0h	KEY to Allow Reset -- 0xb1 B1h = 0xb1
23-2	RESERVED	W	XXXXh	
1	RESETSTKYCLR	W	0h	Clear the RESET STICKY Bit 1h = 1
0	RESETASSERT	W	0h	Assert Reset to IP Domain. 1h = 1

3 STAT Register (Offset = 814h) [Reset = 0000XXXXh]

STAT is shown in Figure 3-13 and described in Table 3-28.

Return to the Summary Table.

IP State Register - Read Only

Figure 3-13. STAT Register

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

Table 3-31. STAT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-17	RESERVED	R	0h	
16	RESETSTKY	R	0h	IP has been Reset 0h = 0 1h = 1
15-0	RESERVED	R	XXXXh	

ADVANCE INFORMATION

4 IIDX Register (Offset = 1020h) [Reset = 00000000h]

IIDX is shown in Figure 3-14 and described in Table 3-29.

Return to the Summary Table.

SYSCTL interrupt index

Figure 3-14. IIDX Register

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

Table 3-33. IIDX Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R	0h	
2-0	STAT	R	0h	The SYSCTL interrupt index (IIDX) register generates a value corresponding to the highest priority pending interrupt source. This value may be used as an address offset for fast, deterministic handling in the interrupt service routine. A read of the IIDX register will clear the corresponding interrupt status in the RIS and MIS registers. 0h = No interrupt pending 1h = LFOSCGOOD interrupt pending 2h = 2 3h = 3 4h = 4 5h = 5 6h = 6 7h = 7

5 IMASK Register (Offset = 1028h) [Reset = 00000000h]

IMASK is shown in Figure 3-15 and described in Table 3-30.

Return to the Summary Table.

SYSCTL interrupt mask

Figure 3-15. IMASK Register

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

Table 3-35. IMASK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-7	RESERVED	R/W	0h	
6	HSCLKGOOD	R/W	0h	HSCLK GOOD 0h = 0 1h = 1
5	SYSPLLGOOD	R/W	0h	SYSPLL GOOD 0h = 0 1h = 1
4	HFCLKGOOD	R/W	0h	HFCLK GOOD 0h = 0 1h = 1
2	FLASHSEC	R/W	0h	Flash Single Error Correct 0h = 0 1h = 1
0	LFOSCGOOD	R/W	0h	Enable or disable the LFOSCGOOD interrupt. LFOSCGOOD indicates that the LFOSC has started successfully. 0h = Interrupt disabled 1h = Interrupt enabled

6 RIS Register (Offset = 1030h) [Reset = 00000000h]

RIS is shown in Figure 3-16 and described in Table 3-31.

Return to the Summary Table.

SYSCTL raw interrupt status

Figure 3-16. RIS Register

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

Table 3-37. RIS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-7	RESERVED	R	0h	
6	HSCLKGOOD	R	0h	HSCLK GOOD 0h = 0 1h = 1
5	SYSPLLGOOD	R	0h	SYSPLL GOOD 0h = 0 1h = 1
4	HFCLKGOOD	R	0h	HFCLK GOOD 0h = 0 1h = 1
2	FLASHSEC	R	0h	Flash Single Error Correct 0h = 0 1h = 1
0	LFOSCGOOD	R	0h	Raw status of the LFOSCGOOD interrupt. 0h = No interrupt pending 1h = Interrupt pending

7 MIS Register (Offset = 1038h) [Reset = 00000000h]

MIS is shown in Figure 3-17 and described in Table 3-32.

Return to the Summary Table.

SYSCTL masked interrupt status

Figure 3-17. MIS Register

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

Table 3-39. MIS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-7	RESERVED	R	0h	
6	HSCLKGOOD	R	0h	HSCLK GOOD 0h = 0 1h = 1
5	SYSPLLGOOD	R	0h	SYSPLL GOOD 0h = 0 1h = 1
4	HFCLKGOOD	R	0h	HFCLK GOOD 0h = 0 1h = 1
2	FLASHSEC	R	0h	Flash Single Error Correct 0h = 0 1h = 1
0	LFOSCGOOD	R	0h	Masked status of the LFOSCGOOD interrupt. 0h = No interrupt pending 1h = Interrupt pending

8 ISET Register (Offset = 1040h) [Reset = 00000000h]

ISET is shown in Figure 3-18 and described in Table 3-33.

Return to the Summary Table.

SYSCTL interrupt set

Figure 3-18. ISET Register

31	30	29	28	27	26	25	24
RESERVED							
W-0h							
23	22	21	20	19	18	17	16
RESERVED							
W-0h							
15	14	13	12	11	10	9	8
RESERVED							
W-0h							
7	6	5	4	3	2	1	0
RESERVED	HSCLKGOOD	SYSPLLGOOD	HFCLKGOOD		FLASHSEC		LFOSCGOOD
W-0h	W1S-0h	W1S-0h	W1S-0h		W1S-0h		W1S-0h

Table 3-41. ISET Register Field Descriptions

Bit	Field	Type	Reset	Description
31-7	RESERVED	W	0h	
6	HSCLKGOOD	W1S	0h	HSCLK GOOD 0h = 0 1h = 1
5	SYSPLLGOOD	W1S	0h	SYSPLL GOOD 0h = 0 1h = 1
4	HFCLKGOOD	W1S	0h	HFCLK GOOD 0h = 0 1h = 1
2	FLASHSEC	W1S	0h	Flash Single Error Correct 0h = 0 1h = 1
0	LFOSCGOOD	W1S	0h	Set the LFOSCGOOD interrupt. 0h = Writing 0h has no effect 1h = Set interrupt

9 ICLR Register (Offset = 1048h) [Reset = 00000000h]

ICLR is shown in Figure 3-19 and described in Table 3-34.

Return to the Summary Table.

SYSCTL interrupt clear

Figure 3-19. ICLR Register

31	30	29	28	27	26	25	24
RESERVED							
W-0h							
23	22	21	20	19	18	17	16
RESERVED							
W-0h							
15	14	13	12	11	10	9	8
RESERVED							
W-0h							
7	6	5	4	3	2	1	0
RESERVED	HSCLKGOOD	SYSPLLGOOD	HFCLKGOOD		FLASHSEC		LFOSCGOOD
W-0h	W1C-0h	W1C-0h	W1C-0h		W1C-0h		W1C-0h

Table 3-43. ICLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-7	RESERVED	W	0h	
6	HSCLKGOOD	W1C	0h	HSCLK GOOD 0h = 0 1h = 1
5	SYSPLLGOOD	W1C	0h	SYSPLL GOOD 0h = 0 1h = 1
4	HFCLKGOOD	W1C	0h	HFCLK GOOD 0h = 0 1h = 1
2	FLASHSEC	W1C	0h	Flash Single Error Correct 0h = 0 1h = 1
0	LFOSCGOOD	W1C	0h	Clear the LFOSCGOOD interrupt. 0h = Writing 0h has no effect 1h = Clear interrupt

10 NMIIIDX Register (Offset = 1050h) [Reset = 00000000h]

NMIIIDX is shown in Figure 3-20 and described in Table 3-35.

Return to the Summary Table.

NMI interrupt index

Figure 3-20. NMIIIDX Register

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

Table 3-45. NMIIIDX Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R	0h	
2-0	STAT	R	0h	The NMI interrupt index (NMIIIDX) register generates a value corresponding to the highest priority pending NMI source. This value may be used as an address offset for fast, deterministic handling in the NMI service routine. A read of the NMIIIDX register will clear the corresponding interrupt status in the NMIRIS register. 0h = No NMI pending 1h = Reserved 2h = 2 3h = 3 4h = 4 5h = 5 6h = 6 7h = 7

11 NMIRIS Register (Offset = 1060h) [Reset = 00000000h]

NMIRIS is shown in Figure 3-21 and described in Table 3-36.

Return to the Summary Table.

NMI raw interrupt status

Figure 3-21. NMIRIS Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED	SYSMEMACC	TMUROMPAR	SRAMPAR	FLASHDED	SECURITY	WWDT0	
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	

Table 3-47. NMIRIS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-7	RESERVED	R	0h	
6	SYSMEMACC	R	0h	SYSMEM Access error 0h = 0 1h = 1
5	TMUROMPAR	R	0h	TMU ROM Parity error 0h = 0 1h = 1
4	SRAMPAR	R	0h	SRAM Parity Error Detect 0h = 0 1h = 1
3	FLASHDED	R	0h	Flash Double Error Detect 0h = 0 1h = 1
2	SECURITY	R	0h	Security Fault 0h = 0 1h = 1
1	WWDT0	R	0h	Watch Dog 0 Fault 0h = 0 1h = 1

12 NMIISET Register (Offset = 1070h) [Reset = 00000000h]

NMIISET is shown in Figure 3-22 and described in Table 3-37.

Return to the Summary Table.

NMI interrupt set

Figure 3-22. NMIISET Register

31	30	29	28	27	26	25	24
RESERVED							
W-0h							
23	22	21	20	19	18	17	16
RESERVED							
W-0h							
15	14	13	12	11	10	9	8
RESERVED							
W-0h							
7	6	5	4	3	2	1	0
RESERVED	SYSMEMACC	TMUROMPAR	SRAMPAR	FLASHDED	SECURITY	WWDT0	
W-0h	W1S-0h	W1S-0h	W1S-0h	W1S-0h	W1S-0h	W1S-0h	

Table 3-49. NMIISET Register Field Descriptions

Bit	Field	Type	Reset	Description
31-7	RESERVED	W	0h	
6	SYSMEMACC	W1S	0h	SYSMEM Access error 0h = 0 1h = 1
5	TMUROMPAR	W1S	0h	TMU ROM Parity error 0h = 0 1h = 1
4	SRAMPAR	W1S	0h	SRAM Parity Error Detect 0h = 0 1h = 1
3	FLASHDED	W1S	0h	Flash Double Error Detect 0h = 0 1h = 1
2	SECURITY	W1S	0h	Security Fault 0h = 0 1h = 1
1	WWDT0	W1S	0h	Watch Dog 0 Fault 0h = 0 1h = 1

13 NMIICLR Register (Offset = 1078h) [Reset = 00000000h]

NMIICLR is shown in Figure 3-23 and described in Table 3-38.

Return to the Summary Table.

NMI interrupt clear

Figure 3-23. NMIICLR Register

31	30	29	28	27	26	25	24
RESERVED							
W-0h							
23	22	21	20	19	18	17	16
RESERVED							
W-0h							
15	14	13	12	11	10	9	8
RESERVED							
W-0h							
7	6	5	4	3	2	1	0
RESERVED	SYSMEMACC	TMUROMPAR	SRAMPAR	FLASHDED	SECURITY	WWDT0	
W-0h	W1C-0h	W1C-0h	W1C-0h	W1C-0h	W1C-0h	W1C-0h	

Table 3-51. NMIICLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-7	RESERVED	W	0h	
6	SYSMEMACC	W1C	0h	SYSMEM Access error 0h = 0 1h = 1
5	TMUROMPAR	W1C	0h	TMU ROM Parity error 0h = 0 1h = 1
4	SRAMPAR	W1C	0h	SRAM Parity Error Detect 0h = 0 1h = 1
3	FLASHDED	W1C	0h	Flash Double Error Detect 0h = 0 1h = 1
2	SECURITY	W1C	0h	Security Fault 0h = 0 1h = 1
1	WWDT0	W1C	0h	Watch Dog 0 Fault 0h = 0 1h = 1

14 SYSOSCCFG Register (Offset = 1100h) [Reset = 0002XXXXh]

SYSOSCCFG is shown in Figure 3-24 and described in Table 3-39.

Return to the Summary Table.

SYSOSC configuration

Figure 3-24. SYSOSCCFG Register

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

Table 3-53. SYSOSCCFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	RESERVED	R/W	0h	
17	FASTCPUEVENT	R/W	1h	if disabled CPU will not wakeup and continue in STANDBY 0h = An interrupt to the CPU will not assert a fast clock request 1h = An interrupt to the CPU will assert a fast clock request
16	BLOCKASYNCALL	R/W	0h	BLOCKASYNCALL may be used to mask block all asynchronous fast clock requests, preventing hardware from dynamically changing the active clock configuration when operating in a given mode. 0h = Asynchronous fast clock requests are controlled by the requesting peripheral 1h = All asynchronous fast clock requests are blocked
15-11	RESERVED	R/W	Xh	
10	DISABLE	R	0h	DISABLE sets the SYSOSC enable/disable policy. SYSOSC may be powered off in RUN, SLEEP, and STOP modes to reduce power consumption. When SYSOSC is disabled, MCLK and ULPCLK are sourced from LFCLK. 0h = Do not disable SYSOSC 1h = Disable SYSOSC immediately and source MCLK and ULPCLK from LFCLK
7-2	RESERVED	R/W	Xh	
1-0	FREQ	R/W	0h	Target operating frequency for the system oscillator (SYSOSC) 0h = Base frequency (32MHz) 1h = Low frequency (4MHz)

15 MCLKCFG Register (Offset = 1104h) [Reset = 07XXX2X0h]

MCLKCFG is shown in Figure 3-25 and described in Table 3-40.

Return to the Summary Table.

Main clock (MCLK) configuration

Figure 3-25. MCLKCFG Register

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

Table 3-55. MCLKCFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-27	RESERVED	R/W	0h	
26-24	MCLKDIVCFG	R/W	7h	MCLK Divider Configuration bits [1:0] are defined as MCLK4 is Bypass, MCLK2 is Bypass 0h = MCLK: No Divide, MCLK2: No Divide, MCLK4: No Divide 1h = MCLK: No Divide, MCLK2: No Divide, MCLK4: Divide MCLK by 2 3h = MCLK: No Divide, MCLK2: No Divide, MCLK4: Divide MCLK by 4 5h = MCLK: No Divide, MCLK2: Divide MCLK by 2, MCLK4: Divide MCLK by 2 7h = MCLK: No Divide, MCLK2: Divide MCLK by 2, MCLK4: MCLK by 4
23	RESERVED	R/W	Xh	
22	MCLKDEADCHK	R/W	0h	MCLKDEADCHK enables or disables the continuous MCLK dead check monitor. LFCLK must be running before MCLKDEADCHK is enabled. 0h = The MCLK dead check monitor is disabled 1h = The MCLK dead check monitor is enabled
21	STOPCLKSTBY	R/W	0h	STOPCLKSTBY sets the STANDBY mode policy (STANDBY0 or STANDBY1). When set, ULPCLK and LFCLK are disabled to all peripherals in STANDBY mode, with the exception of TIMG0 and TIMG1 which continue to run. Wake-up is only possible via an asynchronous fast clock request. 0h = ULPCLK/LFCLK runs to all PD0 peripherals in STANDBY mode 1h = ULPCLK/LFCLK is disabled to all peripherals in STANDBY mode except TIMG0 and TIMG1
20	USELFCLK	R	0h	LFCLK is not an MCLK sourcei in PD1, tied 0. 0h = MCLK will not use the low frequency clock (LFCLK) 1h = MCLK will use the low frequency clock (LFCLK)
19-17	RESERVED	R/W	Xh	

ADVANCE INFORMATION

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

Bit	Field	Type	Reset	Description
16	USEHSCLK	R/W	0h	USEHSCLK, together with USELFCLK, sets the MCLK source policy. Set USEHSCLK to use HSCLK (HFCLK or SYSPLL) as the MCLK source in RUN and SLEEP modes. 0h = MCLK will not use the high speed clock (HSCLK) 1h = MCLK will use the high speed clock (HSCLK) in RUN and SLEEP mode
15-13	RESERVED	R/W	Xh	
7-4	RESERVED	R/W	Xh	

16 HSCLKEN Register (Offset = 1108h) [Reset = 0000XXXXh]

HSCLKEN is shown in Figure 3-26 and described in Table 3-41.

Return to the Summary Table.

High-speed clock (HSCLK) source enable/disable

Figure 3-26. HSCLKEN Register

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

Table 3-57. HSCLKEN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-17	RESERVED	R/W	0h	
16	USEEXTHFCLK	R/W	0h	USEEXTHFCLK selects the HFCLK_IN digital clock input to be the source for HFCLK. 0h = Use XTAL as the HFCLK source 1h = Use the HFCLK_IN digital clock input as the HFCLK source
15-9	RESERVED	R/W	Xh	
8	SYSPLLEN	R/W	0h	SYSPLLEN enables or disables the system phase-lock loop (SYSPLL). 0h = Disable the SYSPLL 1h = Enable the SYSPLL
7-1	RESERVED	R/W	Xh	

17 HSCLKCFG Register (Offset = 110Ch) [Reset = 00000000h]

HSCLKCFG is shown in Figure 3-27 and described in Table 3-42.

Return to the Summary Table.

High-speed clock (HSCLK) source selection

Figure 3-27. HSCLKCFG Register

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

Table 3-59. HSCLKCFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R/W	0h	
0	HSCLKSEL	R/W	0h	HSCLKSEL selects the HSCLK source (SYSPLL or HFCLK). 0h = HSCLK is sourced from the SYSPLL 1h = HSCLK is sourced from the HFCLK

18 HFCLKCLKCFG Register (Offset = 1110h) [Reset = 1XXXXX00h]

HFCLKCLKCFG is shown in Figure 3-28 and described in Table 3-43.

Return to the Summary Table.

High-frequency clock (HFCLK) configuration

Figure 3-28. HFCLKCLKCFG Register

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

Table 3-61. HFCLKCLKCFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R/W	0h	
28	HFCLKFLTCHK	R/W	1h	HFCLKFLTCHK enables or disables the HFCLK startup monitor. 0h = HFCLK startup is not checked 1h = HFCLK startup is checked
27-14	RESERVED	R/W	XXXh	
11-8	RESERVED	R/W	Xh	
7-0	XTALTIME	R/W	0h	XTALTIME specifies the XTAL startup time in 64us resolution. If the HFCLK startup monitor is enabled (HFCLKFLTCHK), XTAL will be checked after this time expires. 0h = Minimum startup time (approximatly zero) FFh = Maximum startup time (approximatly 16.32ms)

19 SYSPLLCFG0 Register (Offset = 1120h) [Reset = 000000XXh]

SYSPLLCFG0 is shown in Figure 3-29 and described in Table 3-44.

Return to the Summary Table.

SYSPLL reference and output configuration

Figure 3-29. SYSPLLCFG0 Register

31	30	29	28	27	26	25	24
RESERVED							
R/W-0h							
23	22	21	20	19	18	17	16
RESERVED							
R/W-0h							
15	14	13	12	11	10	9	8
RDIVCLK1				RDIVCLK0			
R/W-0h				R/W-0h			
7	6	5	4	3	2	1	0
RESERVED		ENABLECLK1	ENABLECLK0	RESERVED			SYSPLLREF
R/W-Xh		R/W-1h	R/W-1h	R/W-Xh			R/W-0h

Table 3-63. SYSPLLCFG0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-20	RESERVED	R/W	0h	
15-12	RDIVCLK1	R/W	0h	RDIVCLK1 sets the final divider for the SYSPLLCLK1 output. 0h = SYSPLLCLK1 is divided by 2 1h = SYSPLLCLK1 is divided by 4 2h = SYSPLLCLK1 is divided by 6 3h = SYSPLLCLK1 is divided by 8 4h = SYSPLLCLK1 is divided by 10 5h = SYSPLLCLK1 is divided by 12 6h = SYSPLLCLK1 is divided by 14 7h = SYSPLLCLK1 is divided by 16 8h = SYSPLLCLK1 is divided by 18 9h = SYSPLLCLK1 is divided by 20 Ah = SYSPLLCLK1 is divided by 22 Bh = SYSPLLCLK1 is divided by 24 Ch = SYSPLLCLK1 is divided by 26 Dh = SYSPLLCLK1 is divided by 28 Eh = SYSPLLCLK1 is divided by 30 Fh = SYSPLLCLK1 is divided by 32
11-8	RDIVCLK0	R/W	0h	RDIVCLK0 sets the final divider for the SYSPLLCLK0 output. 0h = SYSPLLCLK0 is divided by 2 1h = SYSPLLCLK0 is divided by 4 2h = SYSPLLCLK0 is divided by 6 3h = SYSPLLCLK0 is divided by 8 4h = SYSPLLCLK0 is divided by 10 5h = SYSPLLCLK0 is divided by 12 6h = SYSPLLCLK0 is divided by 14 7h = SYSPLLCLK0 is divided by 16 8h = SYSPLLCLK0 is divided by 18 9h = SYSPLLCLK0 is divided by 20 Ah = SYSPLLCLK0 is divided by 22 Bh = SYSPLLCLK0 is divided by 24 Ch = SYSPLLCLK0 is divided by 26 Dh = SYSPLLCLK0 is divided by 28 Eh = SYSPLLCLK0 is divided by 30 Fh = SYSPLLCLK0 is divided by 32

Table 3-63. SYSPLLCFG0 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
7	RESERVED	R/W	Xh	
5	ENABLECLK1	R/W	1h	ENABLECLK1 enables or disables the SYSPLLCLK1 output. 0h = SYSPLLCLK1 is disabled 1h = SYSPLLCLK1 is enabled
4	ENABLECLK0	R/W	1h	ENABLECLK0 enables or disables the SYSPLLCLK0 output. 0h = SYSPLLCLK0 is disabled 1h = SYSPLLCLK0 is enabled
3-2	RESERVED	R/W	Xh	
0	SYSPLLREF	R/W	0h	SYSPLLREF selects the system PLL (SYSPLL) reference clock source. 0h = SYSPLL reference is SYSOSC 1h = SYSPLL reference is HFCLK

20 SYSPLLCFG1 Register (Offset = 1124h) [Reset = 000023XXh]

SYSPLLCFG1 is shown in Figure 3-30 and described in Table 3-45.

Return to the Summary Table.

SYSPLL reference and feedback divider

Figure 3-30. SYSPLLCFG1 Register

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

Table 3-65. SYSPLLCFG1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-15	RESERVED	R/W	0h	
14-8	QDIV	R/W	23h	QDIV selects the SYSPLL feedback path divider. 0h = Divide-by-one is not a valid QDIV option 1h = Feedback path is divided by 2 7Eh = Feedback path is divided by 127 (0x7E)
7-2	RESERVED	R/W	Xh	
1-0	PDIV	R/W	0h	PDIV selects the SYSPLL reference clock prescale divider. 0h = SYSPLLREF is not divided 1h = SYSPLLREF is divided by 2 2h = SYSPLLREF is divided by 4 3h = SYSPLLREF is divided by 8

21 SYSPLLPARAM0 Register (Offset = 1128h) [Reset = X1XAX0X0h]

SYSPLLPARAM0 is shown in Figure 3-31 and described in Table 3-46.

Return to the Summary Table.

SYSPLL PARAM0 (load from FACTORY region)

Figure 3-31. SYSPLLPARAM0 Register

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

Table 3-67. SYSPLLPARAM0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	CAPBOVERRIDE	R/W	1h	CAPBOVERRIDE controls the override for Cap B 0h = Cap B override disabled 1h = Cap B override enabled
30-29	RESERVED	R/W	Xh	
28-24	CAPBVAL	R/W	1h	Override value for Cap B
23-22	RESERVED	R/W	Xh	
21-16	CPCURRENT	R/W	Ah	Charge pump current
15	RESERVED	R/W	Xh	
14-8	STARTTIMELP	R/W	0h	Startup time from low power mode exit to locked clock, in 1us resolution
7	RESERVED	R/W	Xh	
6-0	STARTTIME	R/W	0h	Startup time from enable to locked clock, in 1us resolution

22 SYSPLLPARAM1 Register (Offset = 112Ch) [Reset = 0FXX01XFh]

SYSPLLPARAM1 is shown in Figure 3-32 and described in Table 3-47.

Return to the Summary Table.

SYSPLL PARAM1 (load from FACTORY region)

Figure 3-32. SYSPLLPARAM1 Register

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

Table 3-69. SYSPLLPARAM1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	LPFRESC	R/W	Fh	Loop filter Res C
23-18	RESERVED	R/W	Xh	
17-8	LPFRESA	R/W	1h	Loop filter Res A
7-5	RESERVED	R/W	Xh	
4-0	LPFCAPA	R/W	Fh	Loop filter Cap A

23 SYSPLLPARAM2 Register (Offset = 1130h) [Reset = 0000000Xh]

SYSPLLPARAM2 is shown in Figure 3-33 and described in Table 3-48.

Return to the Summary Table.

SYSPLL PARAM2 (load from FACTORY region)

Figure 3-33. SYSPLLPARAM2 Register

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

Table 3-71. SYSPLLPARAM2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R/W	0h	
3	RNGFIXVCOIBIASCFG	R/W	1h	0 value for Temperature Compensation R addition
2	RESERVED	R/W	Xh	
1-0	LPFCAPC	R/W	0h	Loop filter Cap C

ADVANCE INFORMATION

24 SYSPLLLDOCTL Register (Offset = 1134h) [Reset = 00000000h]

SYSPLLLDOCTL is shown in Figure 3-34 and described in Table 3-49.

Return to the Summary Table.

SYSPLL LDO CTL (load from FACTORY region)

Figure 3-34. SYSPLLLDOCTL Register

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

Table 3-73. SYSPLLLDOCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R/W	0h	
15-0	LDOCTLLOWV	R/W	0h	LDO Configurability

25 SYSPLLLDOPROG Register (Offset = 1138h) [Reset = 00000004h]

SYSPLLLDOPROG is shown in Figure 3-35 and described in Table 3-50.

Return to the Summary Table.

SYSPLL LDO VOUT PROG (load from FACTORY region)

Figure 3-35. SYSPLLLDOPROG Register

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

Table 3-75. SYSPLLLDOPROG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R/W	0h	
2-0	LDOVOUTPROGLOWV	R/W	4h	HPLL LDO Vout Prog

26 GENCLKEN Register (Offset = 113Ch) [Reset = 000000XXh]

GENCLKEN is shown in Figure 3-36 and described in Table 3-51.

Return to the Summary Table.

General clock enable control

Figure 3-36. GENCLKEN Register

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

Table 3-77. GENCLKEN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-19	RESERVED	R/W	0h	
15	CANEXTDIVEN	R/W	0h	CANEXTDIVEN enables or disables the divider function of the PLL Source to CAN. 0h = CLock divider is disabled (passthrough, EXTDIVCAN is not applied) 1h = Clock divider is enabled (EXTDIVCAN is applied)
14-12	EXTDIVCAN	R/W	0h	EXTDIVCAN selects the divider value for the divider for the PLL Source to CAN. 0h = CLK_OUT source is divided by 2 1h = CLK_OUT source is divided by 4 2h = CLK_OUT source is divided by 6 3h = CLK_OUT source is divided by 8 4h = CLK_OUT source is divided by 10 5h = CLK_OUT source is divided by 12 6h = CLK_OUT source is divided by 14 7h = CLK_OUT source is divided by 16
11	MCLKEXTDIVEN	R/W	0h	MCLKEXTDIVEN enables or disables the divider function of the PLL Source to MCLK. 0h = CLock divider is disabled (passthrough, EXTDIVMCLK is not applied) 1h = Clock divider is enabled (EXTDIVMCLK is applied)
10-8	EXTDIVMCLK	R/W	0h	EXTDIVMCLK selects the divider value for the divider for the PLL Source MCLK. 0h = CLK_OUT source is divided by 2 1h = CLK_OUT source is divided by 4 2h = CLK_OUT source is divided by 6 3h = CLK_OUT source is divided by 8 4h = CLK_OUT source is divided by 10 5h = CLK_OUT source is divided by 12 6h = CLK_OUT source is divided by 14 7h = CLK_OUT source is divided by 16
7-5	RESERVED	R/W	Xh	
3-1	RESERVED	R/W	Xh	

Table 3-77. GENCLKEN Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
0	EXCLKEN	R/W	0h	EXCLKEN enables the CLK_OUT external clock output block. 0h = CLK_OUT block is disabled 1h = CLK_OUT block is enabled

27 GENCLKCFG Register (Offset = 1140h) [Reset = 00000X0Xh]

GENCLKCFG is shown in Figure 3-37 and described in Table 3-52.

Return to the Summary Table.

General clock configuration

Figure 3-37. GENCLKCFG Register

31	30	29	28	27	26	25	24
RESERVED			FCCTRIGCNT				
R/W-0h			R/W-0h				
23	22	21	20	19	18	17	16
		FCCLVLTRIG	FCCTRIGSRC	FCCSELCLK			
		R/W-0h	R/W-0h	R/W-0h			
15	14	13	12	11	10	9	8
				RESERVED			CANCLKSRC
				R/W-Xh			R/W-0h
7	6	5	4	3	2	1	0
EXCLKDIVEN	EXCLKDIVVAL			RESERVED	EXCLKSRC		
R/W-0h	R/W-0h			R/W-Xh	R/W-0h		

Table 3-79. GENCLKCFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	RESERVED	R/W	0h	
28-24	FCCTRIGCNT	R/W	0h	FCCTRIGCNT specifies the number of trigger clock periods in the trigger window. FCCTRIGCNT=0h (one trigger clock period) up to 1Fh (32 trigger clock periods) may be specified.
21	FCCLVLTRIG	R/W	0h	FCCLVLTRIG selects the frequency clock counter (FCC) trigger mode. 0h = Rising edge to rising edge triggered 1h = Level triggered
20	FCCTRIGSRC	R/W	0h	FCCTRIGSRC selects the frequency clock counter (FCC) trigger source. 0h = FCC trigger is the external pin 1h = FCC trigger is the LFCLK
19-16	FCCSELCLK	R/W	0h	FCCSELCLK selectes the frequency clock counter (FCC) clock source. 0h = FCC clock is MCLK/4 1h = FCC clock is SYSOSC 2h = FCC clock is HFCLK 3h = FCC clock is the CLK_OUT selection 4h = FCC clock is SYSPLLCLK0 5h = FCC clock is SYSPLLCLK1 6h = Reserved 7h = FCC clock is the FCCIN external input
11-10	RESERVED	R/W	Xh	
8	CANCLKSRC	R/W	0h	CANCLKSRC selects the CANCLK source. 0h = CANCLK source is HFCLK 1h = CANCLK source is SYSPLLCLK
7	EXCLKDIVEN	R/W	0h	EXCLKDIVEN enables or disables the divider function of the CLK_OUT external clock output block. 0h = CLock divider is disabled (passthrough, EXCLKDIVVAL is not applied) 1h = Clock divider is enabled (EXCLKDIVVAL is applied)

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

Bit	Field	Type	Reset	Description
6-4	EXCLKDIVVAL	R/W	0h	EXCLKDIVVAL selects the divider value for the divider in the CLK_OUT external clock output block. 0h = CLK_OUT source is divided by 2 1h = CLK_OUT source is divided by 4 2h = CLK_OUT source is divided by 6 3h = CLK_OUT source is divided by 8 4h = CLK_OUT source is divided by 10 5h = CLK_OUT source is divided by 12 6h = CLK_OUT source is divided by 14 7h = CLK_OUT source is divided by 16
3	RESERVED	R/W	Xh	
2-0	EXCLKSRC	R/W	0h	EXCLKSRC selects the source for the CLK_OUT external clock output block. ULPCLK and MFPCLK require the CLK_OUT divider (EXCLKDIVEN) to be enabled 0h = CLK_OUT is SYSOSC 1h = CLK_OUT is ULPCLK (EXCLKDIVEN must be enabled) 2h = CLK_OUT is LFCLK 3h = CLK_OUT is MFPCLK (EXCLKDIVEN must be enabled) 4h = CLK_OUT is HFCLK 5h = CLK_OUT is SYSPLLCLK1 (SYSPLLCLK1 must be <=48MHz)

28 PMODECFG Register (Offset = 1144h) [Reset = 00000000h]

PMODECFG is shown in Figure 3-38 and described in Table 3-53.

Return to the Summary Table.

Power mode configuration

Figure 3-38. PMODECFG Register

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

Table 3-81. PMODECFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R/W	0h	
1-0	DSLEEP	R/W	0h	DSLEEP selects the operating mode to enter upon a DEEPSLEEP request from the CPU. 0h = STOP mode is entered 1h = STANDBY mode is entered 2h = SHUTDOWN mode is entered 3h = Reserved

29 MLDOLPENCFG Register (Offset = 1148h) [Reset = 000000XXh]

MLDOLPENCFG is shown in Figure 3-39 and described in Table 3-54.

Return to the Summary Table.

LDO Configuration Control

Figure 3-39. MLDOLPENCFG Register

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

Table 3-83. MLDOLPENCFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	RESERVED	R/W	0h	
7-1	RESERVED	R/W	Xh	
0	CVLODIS	R/W	0h	Control to disable lowering the core voltage for STOP and STANDBY 0h = Lower Core Voltage for STOP and STANDBY mode 1h = Do Not Lower Core Voltage for STOP and STANDBY mode to provide faster wakeup

ADVANCE INFORMATION

30 FCC Register (Offset = 1150h) [Reset = 00000000h]

FCC is shown in Figure 3-40 and described in Table 3-55.

Return to the Summary Table.

Frequency clock counter (FCC) count

Figure 3-40. FCC Register

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

Table 3-85. FCC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-22	RESERVED	R	0h	
21-0	DATA	R	0h	Frequency clock counter (FCC) count value.

31 PMULDOSPARECTL Register (Offset = 1154h) [Reset = 00000000h]

PMULDOSPARECTL is shown in Figure 3-41 and described in Table 3-56.

Return to the Summary Table.

LDO Spare Control

Figure 3-41. PMULDOSPARECTL Register

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

Table 3-87. PMULDOSPARECTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R/W	0h	

ADVANCE INFORMATION

32 SYSCTL_ECO_REG1 Register (Offset = 1158h) [Reset = 00000000h]

SYSCTL_ECO_REG1 is shown in Figure 3-42 and described in Table 3-57.

Return to the Summary Table.

Sysctl ECO Reg 1

Figure 3-42. SYSCTL_ECO_REG1 Register

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

Table 3-89. SYSCTL_ECO_REG1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ecoreg	R/W	0h	ECO Reg 1 for M33

33 SYSCTL_ECO_REG2 Register (Offset = 115Ch) [Reset = 00000000h]

SYSCTL_ECO_REG2 is shown in Figure 3-43 and described in Table 3-58.

Return to the Summary Table.

Sysctl ECO Reg 2

Figure 3-43. SYSCTL_ECO_REG2 Register

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

Table 3-91. SYSCTL_ECO_REG2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ecoreg	R/W	0h	ECO Reg 2 for M33

ADVANCE INFORMATION

34 SYSTEMCFG Register (Offset = 1180h) [Reset = 00XXXXXXh]

SYSTEMCFG is shown in Figure 3-44 and described in Table 3-59.

Return to the Summary Table.

System configuration

Figure 3-44. SYSTEMCFG Register

31	30	29	28	27	26	25	24
KEY							
W-0h							
23	22	21	20	19	18	17	16
RESERVED							
R/W-XXXXh							
15	14	13	12	11	10	9	8
RESERVED							
R/W-XXXXh							
7	6	5	4	3	2	1	0
RESERVED					FLASHECCRSTDIS	RESERVED	WWDTLP0RSTDIS
R/W-XXXXh					R/W-1h	R/W-Xh	R/W-0h

Table 3-93. SYSTEMCFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	KEY	W	0h	The key value of 1Bh (27) must be written to KEY together with contents to be updated. Reads as 0 1Bh = Issue write
23-3	RESERVED	R/W	XXXXh	
2	FLASHECCRSTDIS	R/W	1h	FLASHECCRSTDIS specifies whether a flash ECC double error detect (DED) will trigger a SYSRST or an NMI. 0h = Flash ECC DED will trigger a SYSRST 1h = Flash ECC DED will trigger a NMI
1	RESERVED	R/W	Xh	
0	WWDTLP0RSTDIS	R/W	0h	WWDTLP0RSTDIS specifies whether a WWDT Error Event will trigger a BOOTRST or an NMI. 0h = WWDTLP0 Error Event will trigger a BOOTRST 1h = WWDTLP0 Error Event will trigger an NMI

35 SRAMCFG Register (Offset = 1184h) [Reset = 00X0XXX0h]

SRAMCFG is shown in Figure 3-45 and described in Table 3-60.

Return to the Summary Table.

System SRAM configuration

Figure 3-45. SRAMCFG Register

31	30	29	28	27	26	25	24
KEY							
W-0h							
23	22	21	20	19	18	17	16
RESERVED				BANKINITDIS3	BANKINITDIS2	BANKINITDIS1	BANKINITDIS0
R/W-Xh				R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED							
R/W-XXXh							
7	6	5	4	3	2	1	0
RESERVED							
R/W-XXXh							

Table 3-95. SRAMCFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	KEY	W	0h	The key value of B5h (181) must be written to KEY together with contents to be updated. Reads as 0 B5h = Issue write
23-20	RESERVED	R/W	Xh	
19	BANKINITDIS3	R/W	0h	SRAM BANK3 Initialization 0h = SRAM BANK3 will Initialize when transitioning from OFF to ON 1h = SRAM BANK3 will NOT Initialize when transitioning from OFF to ON
18	BANKINITDIS2	R/W	0h	SRAM BANK2 Initialization 0h = SRAM BANK2 will Initialize when transitioning from OFF to ON 1h = SRAM BANK2 will NOT Initialize when transitioning from OFF to ON
17	BANKINITDIS1	R/W	0h	SRAM BANK1 Initialization 0h = SRAM BANK1 will Initialize when transitioning from OFF to ON 1h = SRAM BANK1 will NOT Initialize when transitioning from OFF to ON
16	BANKINITDIS0	R/W	0h	SRAM BANK0 Initialization 0h = SRAM BANK0 will Initialize when transitioning from OFF to ON 1h = SRAM BANK0 will NOT Initialize when transitioning from OFF to ON
15-4	RESERVED	R/W	XXXh	

36 WRITELOCK Register (Offset = 1200h) [Reset = 00000000h]

WRITELOCK is shown in Figure 3-46 and described in Table 3-61.

Return to the Summary Table.

SYSCTL register write lockout

Figure 3-46. WRITELOCK Register

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

Table 3-97. WRITELOCK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R/W	0h	
0	ACTIVE	R/W	0h	ACTIVE controls whether critical SYSCTL registers are write protected or not. 0h = Allow writes to lockable registers 1h = Disallow writes to lockable registers

37 CLKSTATUS Register (Offset = 1204h) [Reset = XXXXXXXXh]

CLKSTATUS is shown in Figure 3-47 and described in Table 3-62.

Return to the Summary Table.

Clock module (CKM) status

Figure 3-47. CLKSTATUS Register

31	30	29	28	27	26	25	24
	RESERVED	SYSPLLBLKUPD	HFCLKBLKUPD	RESERVED		FCCDONE	FCLMODE
	R-Xh	R-0h	R-0h	R-Xh		R-0h	R-0h
23	22	21	20	19	18	17	16
RESERVED		HSCLKGOOD	HSCLKDEAD	RESERVED			CURHSCLKSEL
R-Xh		R-0h	R-0h	R-Xh			R-0h
15	14	13	12	11	10	9	8
RESERVED	SYSPLLOFF	HFCLKOFF	HSCLKSOFF	LFOSCGOOD	RESERVED	SYSPLLGOOD	HFCLKGOOD
R-Xh	R-1h	R-1h	R-1h	R-0h	R-Xh	R-0h	R-0h
7	6	5	4	3	2	1	0
		RESERVED	HSCLKMUX	RESERVED		SYSOSCFREQ	
		R-Xh	R-0h	R-Xh		R-0h	

Table 3-99. CLKSTATUS Register Field Descriptions

Bit	Field	Type	Reset	Description
30	RESERVED	R	Xh	
29	SYSPLLBLKUPD	R	0h	SYSPLLBLKUPD indicates when writes to SYSPLLCFG0/1 and SYSPLLPARAM0/1 are blocked. 0h = writes to SYSPLLCFG0/1 and SYSPLLPARAM0/1 are allowed 1h = writes to SYSPLLCFG0/1 and SYSPLLPARAM0/1 are blocked
28	HFCLKBLKUPD	R	0h	HFCLKBLKUPD indicates when writes to the HFCLKCLKCFG register are blocked. 0h = Writes to HFCLKCLKCFG are allowed 1h = Writes to HFCLKCLKCFG are blocked
27-26	RESERVED	R	Xh	
25	FCCDONE	R	0h	FCCDONE indicates when a frequency clock counter capture is complete. 0h = FCC capture is not done 1h = FCC capture is done
24	FCLMODE	R	0h	FCLMODE indicates if the SYSOSC frequency correction loop (FCL) is enabled. 0h = SYSOSC FCL is disabled 1h = SYSOSC FCL is enabled
23-22	RESERVED	R	Xh	
21	HSCLKGOOD	R	0h	HSCLKGOOD is set by hardware if the selected clock source for HSCLK started successfully. 0h = The HSCLK source did not start correctly 1h = The HSCLK source started correctly
20	HSCLKDEAD	R	0h	HSCLKDEAD is set by hardware if the selected source for HSCLK was started but did not start successfully. 0h = The HSCLK source was not started or started correctly 1h = The HSCLK source did not start correctly
19-18	RESERVED	R	Xh	

Table 3-99. CLKSTATUS Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	CURHSCLKSEL	R	0h	CURHSCLKSEL indicates the current clock source for HSCLK. 0h = HSCLK is currently sourced from the SYSPLL 1h = HSCLK is currently sourced from the HFCLK
15	RESERVED	R	Xh	
14	SYSPLLOFF	R	1h	SYSPLLOFF indicates if the SYSPLL is disabled or was dead at startup. When the SYSPLL is started, SYSPLLOFF is cleared by hardware. Following startup of the SYSPLL, if the SYSPLL startup monitor determines that the SYSPLL was not started correctly, SYSPLLOFF is set. 0h = SYSPLL started correctly and is enabled 1h = SYSPLL is disabled or was dead startup
13	HFCLKOFF	R	1h	HFCLKOFF indicates if the HFCLK is disabled or was dead at startup. When the HFCLK is started, HFCLKOFF is cleared by hardware. Following startup of the HFCLK, if the HFCLK startup monitor determines that the HFCLK was not started correctly, HFCLKOFF is set. 0h = HFCLK started correctly and is enabled 1h = HFCLK is disabled or was dead at startup
12	HSCLKSOFF	R	1h	HSCLKSOFF is set when the high speed clock sources (SYSPLL, HFCLK) are disabled or dead. It is the logical AND of HFCLKOFF and SYSPLLOFF. 0h = SYSPLL, HFCLK, or both were started correctly and remain enabled 1h = SYSPLL and HFCLK are both either off or dead
11	LFOSCGOOD	R	0h	LFOSCGOOD indicates when the LFOSC startup has completed and the LFOSC is ready for use. 0h = LFOSC is not ready 1h = LFOSC is ready
10	RESERVED	R	Xh	
9	SYSPLLGOOD	R	0h	SYSPLLGOOD indicates if the SYSPLL started correctly. When the SYSPLL is started, SYSPLLGOOD is cleared by hardware. After the startup settling time has expired, the SYSPLL status is tested. If the SYSPLL started successfully the SYSPLLGOOD bit is set, else it is left cleared. 0h = SYSPLL did not start correctly 1h = SYSPLL started correctly
8	HFCLKGOOD	R	0h	HFCLKGOOD indicates that the HFCLK started correctly. When the XTAL is started or HFCLK_IN is selected as the HFCLK source, this bit will be set by hardware if a valid HFCLK is detected, and cleared if HFCLK is not operating within the expected range. 0h = HFCLK did not start correctly 1h = HFCLK started correctly
5	RESERVED	R	Xh	
4	HSCLKMUX	R	0h	HSCLKMUX indicates if MCLK is currently sourced from the high-speed clock (HSCLK). 0h = MCLK is not sourced from HSCLK 1h = MCLK is sourced from HSCLK
3-2	RESERVED	R	Xh	
1-0	SYSOSCFREQ	R	0h	SYSOSCFREQ indicates the current SYSOSC operating frequency. 0h = SYSOSC is at base frequency (32MHz) 1h = SYSOSC is at low frequency (4MHz)

38 SYSSTATUS Register (Offset = 1208h) [Reset = XXX00XX0h]

SYSSTATUS is shown in Figure 3-48 and described in Table 3-63.

Return to the Summary Table.

System status information

Figure 3-48. SYSSTATUS Register

31	30	29	28	27	26	25	24
REBOOTATTEMPTS		RESERVED					
R-0h		R-XXh					
23	22	21	20	19	18	17	16
RESERVED				SRAMBANK3READY	SRAMBANK2READY	SRAMBANK1READY	SRAMBANK0READY
R-XXh				R-0h	R-0h	R-0h	R-0h
15	14	13	12	11	10	9	8
	SHDNIOLOCK	SWDJCFGDIS	EXTRSTPINDIS	RESERVED			MCAN0READY
	R-0h	R-0h	R-0h	R-Xh			R-0h
7	6	5	4	3	2	1	0
RESERVED	PMUIREFGOOD					FLASHSEC	FLASHDED
R-Xh	R-0h					R-0h	R-0h

Table 3-101. SYSSTATUS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	REBOOTATTEMPTS	R	0h	REBOOTATTEMPTS indicates the number of boot attempts taken before the user application starts.
29-20	RESERVED	R	XXh	
19	SRAMBANK3READY	R	0h	SRAM BANK3 READY STATE 0h = SRAM BANK3 is NOT READY for access 1h = SRAM BANK3 is READY for access
18	SRAMBANK2READY	R	0h	SRAM BANK2 READY STATE 0h = SRAM BANK2 is NOT READY for access 1h = SRAM BANK2 is READY for access
17	SRAMBANK1READY	R	0h	SRAM BANK1 READY STATE 0h = SRAM BANK1 is NOT READY for access 1h = SRAM BANK1 is READY for access
16	SRAMBANK0READY	R	0h	SRAM BANK0 READY STATE 0h = SRAM BANK0 is NOT READY for access 1h = SRAM BANK0 is READY for access
14	SHDNIOLOCK	R	0h	SHDNIOLOCK indicates when IO is locked due to SHUTDOWN 0h = IO IS NOT Locked due to SHUTDOWN 1h = IO IS Locked due to SHUTDOWN
13	SWDJCFGDIS	R	0h	SWDJCFGDIS indicates when user has disabled the use of SWD/JTAG Port 0h = SWD/JTAG Port Enabled 1h = SWD/JTAG Port Disabled
12	EXTRSTPINDIS	R	0h	EXTRSTPINDIS indicates when user has disabled the use of external reset pin 0h = External Reset Pin Enabled 1h = External Reset Pin Disabled
11-9	RESERVED	R	Xh	

Table 3-101. SYSSTATUS Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
8	MCAN0READY	R	0h	MCAN0READY indicates when the MCAN0 peripheral is ready. 0h = MCAN0 is not ready 1h = MCAN0 is ready
7	RESERVED	R	Xh	
6	PMUIREFGOOD	R	0h	PMUIREFGOOD is set by hardware when the PMU current reference is ready. 0h = IREF is not ready 1h = IREF is ready
1	FLASHSEC	R	0h	FLASHSEC indicates if a flash ECC single bit error was detected and corrected (SEC). 0h = No flash ECC single bit error detected 1h = Flash ECC single bit error was detected and corrected
0	FLASHDED	R	0h	FLASHDED indicates if a flash ECC double bit error was detected (DED). 0h = No flash ECC double bit error detected 1h = Flash ECC double bit error detected

39 RSTCAUSE Register (Offset = 1220h) [Reset = 00000000h]

RSTCAUSE is shown in Figure 3-49 and described in Table 3-64.

Return to the Summary Table.

Reset cause

Figure 3-49. RSTCAUSE Register

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

Table 3-103. RSTCAUSE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-5	RESERVED	R	0h	
4-0	ID	RC	0h	ID is a read-to-clear field which indicates the lowest level reset cause since the last read. 0h = No reset since last read 1h = POR- violation, SHUTDNSTOREx or PMU trim parity fault 2h = NRST triggered POR (>1s hold) 3h = Software triggered POR 4h = BOR0- violation 5h = SHUTDOWN mode exit 8h = Non-PMU trim parity fault 9h = Fatal clock failure Ch = NRST triggered BOOTRST (<1s hold) Dh = Software triggered BOOTRST Eh = WWDT0 violation 10h = BSL exit 11h = BSL entry 14h = Flash uncorrectable ECC error 15h = CPULOCK violation 1Ah = Debug triggered SYSRST 1Bh = Software triggered SYSRST 1Ch = Debug triggered CPURST 1Dh = Software triggered CPURST

40 RESETLEVEL Register (Offset = 1300h) [Reset = 00000000h]

RESETLEVEL is shown in Figure 3-50 and described in Table 3-65.

Return to the Summary Table.

Reset level for application-triggered reset command

Figure 3-50. RESETLEVEL Register

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

Table 3-105. RESETLEVEL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R/W	0h	
2-0	LEVEL	R/W	0h	LEVEL is used to specify the type of reset to be issued when RESETCMD is set to generate a software triggered reset. 0h = Issue a SYSRST (CPU plus peripherals only) 1h = Issue a BOOTRST (CPU, peripherals, and boot configuration routine) 2h = Issue a SYSRST and enter the boot strap loader (BSL) 3h = Issue a power-on reset (POR) 4h = Issue a SYSRST and exit the boot strap loader (BSL)

41 RESETCMD Register (Offset = 1304h) [Reset = 00XXXXXXh]

RESETCMD is shown in Figure 3-51 and described in Table 3-66.

Return to the Summary Table.

Execute an application-triggered reset command

Figure 3-51. RESETCMD Register

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

Table 3-107. RESETCMD Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	KEY	W	0h	The key value of E4h (228) must be written to KEY together with GO to trigger the reset. E4h = Issue reset
23-1	RESERVED	W	XXXXh	
0	GO	W	0h	Execute the reset specified in RESETLEVEL.LEVEL. Must be written together with the KEY. 1h = Issue reset

42 SYSOSCFCLCTL Register (Offset = 1310h) [Reset = 00XXXXXXh]

SYSOSCFCLCTL is shown in Figure 3-52 and described in Table 3-67.

Return to the Summary Table.

SYSOSC frequency correction loop (FCL) ROSC enable

Figure 3-52. SYSOSCFCLCTL Register

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

Table 3-109. SYSOSCFCLCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	KEY	W	0h	The key value of 2Ah (42) must be written to KEY together with SETUSEFCL to enable the FCL. 2Ah = Issue Command
23-2	RESERVED	W	XXXXh	
0	SETUSEFCL	W	0h	Set SETUSEFCL to enable the frequency correction loop in SYSOSC. Once enabled, this state is locked until the next BOOTRST. 1h = Enable the SYSOSC FCL

43 SHDNIOREL Register (Offset = 131Ch) [Reset = 00XXXXXXh]

SHDNIOREL is shown in Figure 3-53 and described in Table 3-68.

Return to the Summary Table.

SHUTDOWN IO release control

Figure 3-53. SHDNIOREL Register

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

Table 3-111. SHDNIOREL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	KEY	W	0h	The key value 91h must be written to KEY together with RELEASE to set RELEASE. 91h = Issue command
23-1	RESERVED	W	XXXXh	
0	RELEASE	W	0h	Set RELEASE to release the IO after a SHUTDOWN mode exit. 1h = Release IO

ADVANCE INFORMATION

44 EXRSTPIN Register (Offset = 1320h) [Reset = 00XXXXXXh]

EXRSTPIN is shown in Figure 3-54 and described in Table 3-69.

Return to the Summary Table.

Disable the reset function of the NRST pin

Figure 3-54. EXRSTPIN Register

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

Table 3-113. EXRSTPIN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	KEY	W	0h	The key value 1Eh must be written together with DISABLE to disable the reset function. 1Eh = Issue command
23-1	RESERVED	W	XXXXh	
0	DISABLE	W	0h	Set DISABLE to disable the reset function of the NRST pin. Once set, this configuration is locked until the next POR. 0h = Reset function of NRST pin is enabled 1h = Reset function of NRST pin is disabled

ADVANCE INFORMATION

45 SYSSTATUSCLR Register (Offset = 1324h) [Reset = 00XXXXXXh]

SYSSTATUSCLR is shown in Figure 3-55 and described in Table 3-70.

Return to the Summary Table.

Clear sticky bits of SYSSTATUS

Figure 3-55. SYSSTATUSCLR Register

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

Table 3-115. SYSSTATUSCLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	KEY	W	0h	The key value CEh (206) must be written to KEY together with ALLECC to clear the ECC state. CEh = Issue command
23-1	RESERVED	W	XXXXh	
0	ALLECC	W	0h	Set ALLECC to clear all ECC related SYSSTATUS indicators. 1h = Clear ECC error state

46 SWDJCFG Register (Offset = 1328h) [Reset = 00XXXXXXh]

SWDJCFG is shown in Figure 3-56 and described in Table 3-71.

Return to the Summary Table.

Disable the SWD/JTAG function on the SWD/JTAG pins

Figure 3-56. SWDJCFG Register

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

Table 3-117. SWDJCFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	KEY	W	0h	The key value 62h (98) must be written to KEY together with DISBALE to disable the SWD/JTAG functions. 62h = Issue command
23-1	RESERVED	W	XXXXh	
0	DISABLE	W	0h	Set DISABLE to disable the SWD/JTAG function on SWD/JTAG pins, allowing the SWD/JTAG pins to be used as GPIO. 1h = Disable SWD/JTAG function on SWD/JTAG pins

47 FCCCMD Register (Offset = 132Ch) [Reset = 00XXXXXXh]

FCCCMD is shown in Figure 3-57 and described in Table 3-72.

Return to the Summary Table.

Frequency clock counter start capture

Figure 3-57. FCCCMD Register

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

Table 3-119. FCCCMD Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	KEY	W	0h	The key value 0Eh (14) must be written with GO to start a capture. 0Eh = Issue command
23-1	RESERVED	W	XXXXh	
0	GO	W	0h	Set GO to start a capture with the frequency clock counter (FCC). 1h = 1

48 SHUTDNSTORE0 Register (Offset = 1400h) [Reset = 00000000h]

SHUTDNSTORE0 is shown in Figure 3-58 and described in Table 3-73.

Return to the Summary Table.

Shutdown storage memory (byte 0)

Figure 3-58. SHUTDNSTORE0 Register

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

Table 3-121. SHUTDNSTORE0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-10	RESERVED	R/W	0h	
9	RESERVED	R	0h	Reserved
8	RESERVED	R/W	0h	Reserved
7-0	DATA	R/W	0h	Shutdown storage byte 0

49 SHUTDNSTORE1 Register (Offset = 1404h) [Reset = 00000000h]

SHUTDNSTORE1 is shown in Figure 3-59 and described in Table 3-74.

Return to the Summary Table.

Shutdown storage memory (byte 1)

Figure 3-59. SHUTDNSTORE1 Register

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

Table 3-123. SHUTDNSTORE1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-10	RESERVED	R/W	0h	
9	RESERVED	R	0h	Reserved
8	RESERVED	R/W	0h	Reserved
7-0	DATA	R/W	0h	Shutdown storage byte 1

50 SHUTDNSTORE2 Register (Offset = 1408h) [Reset = 00000000h]

SHUTDNSTORE2 is shown in Figure 3-60 and described in Table 3-75.

Return to the Summary Table.

Shutdown storage memory (byte 2)

Figure 3-60. SHUTDNSTORE2 Register

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

Table 3-125. SHUTDNSTORE2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-10	RESERVED	R/W	0h	
9	RESERVED	R	0h	Reserved
8	RESERVED	R/W	0h	Reserved
7-0	DATA	R/W	0h	Shutdown storage byte 2

51 SHUTDNSTORE3 Register (Offset = 140Ch) [Reset = 00000000h]

SHUTDNSTORE3 is shown in Figure 3-61 and described in Table 3-76.

Return to the Summary Table.

Shutdown storage memory (byte 3)

Figure 3-61. SHUTDNSTORE3 Register

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

Table 3-127. SHUTDNSTORE3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-10	RESERVED	R/W	0h	
9	RESERVED	R	0h	Reserved
8	RESERVED	R/W	0h	Reserved
7-0	DATA	R/W	0h	Shutdown storage byte 3

52 ADCSEQFRCGB Register (Offset = 1410h) [Reset = 00000000h]

ADCSEQFRCGB is shown in Figure 3-62 and described in Table 3-77.

Return to the Summary Table.

ADC Global Sequence Force

Figure 3-62. ADCSEQFRCGB Register

31	30	29	28	27	26	25	24
RESERVED							
R/W-0h							
23	22	21	20	19	18	17	16
RESERVED							
R/W-0h							
15	14	13	12	11	10	9	8
RESERVED							
R/W-0h							
7	6	5	4	3	2	1	0
RESERVED				SEQ3	SEQ2	SEQ1	SEQ0
R/W-0h				R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 3-129. ADCSEQFRCGB Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R/W	0h	
3	SEQ3	R/W	0h	Generate synchronous SW trigger for SEQ3
2	SEQ2	R/W	0h	Generate synchronous SW trigger for SEQ2
1	SEQ1	R/W	0h	Generate synchronous SW trigger for SEQ1
0	SEQ0	R/W	0h	Generate synchronous SW trigger for SEQ0

53 ADCSEQFRCGBSEL Register (Offset = 1414h) [Reset = 00000000h]

ADCSEQFRCGBSEL is shown in Figure 3-63 and described in Table 3-78.

Return to the Summary Table.

ADC Global Sequence Force Select

Figure 3-63. ADCSEQFRCGBSEL Register

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

Table 3-131. ADCSEQFRCGBSEL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R/W	0h	
2	ADCC	R/W	0h	Generate synchronous SW trigger for ADCC
1	ADCB	R/W	0h	Generate synchronous SW trigger for ADCB
0	ADCA	R/W	0h	Generate synchronous SW trigger for ADCA

54 M33SPARESOCLOCK1 Register (Offset = 1418h) [Reset = 00000000h]

M33SPARESOCLOCK1 is shown in Figure 3-64 and described in Table 3-79.

Return to the Summary Table.

M33C1 Spare SOC LOCK Reg 1

Figure 3-64. M33SPARESOCLOCK1 Register

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

Table 3-133. M33SPARESOCLOCK1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	SPARE	R/W	0h	Spare SOC LOCK Register 1

ADVANCE INFORMATION

55 M33SPARESOCLOCK2 Register (Offset = 141Ch) [Reset = 00000000h]

M33SPARESOCLOCK2 is shown in Figure 3-65 and described in Table 3-80.

Return to the Summary Table.

M33C1 Spare SOC LOCK Reg 2

Figure 3-65. M33SPARESOCLOCK2 Register

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

Table 3-135. M33SPARESOCLOCK2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	SPARE	R/W	0h	Spare SOC LOCK Register 2

ADVANCE INFORMATION

56 SYSCTL_READ_REG Register (Offset = 1420h) [Reset = 00000000h]

SYSCTL_READ_REG is shown in Figure 3-66 and described in Table 3-81.

Return to the Summary Table.

Sysctl read only Reg

Figure 3-66. SYSCTL_READ_REG Register

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

Table 3-137. SYSCTL_READ_REG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ecoreg	R/W	0h	Read only register

57 PWREN_MCPERIPH Register (Offset = 1424h) [Reset = 00XXX000h]

PWREN_MCPERIPH is shown in Figure 3-67 and described in Table 3-82.

Return to the Summary Table.

Register to control the power state

Figure 3-67. PWREN_MCPERIPH Register

31	30	29	28	27	26	25	24
KEY							
W-0h							
23	22	21	20	19	18	17	16
RESERVED							
R/W-XXh							
15	14	13	12	11	10	9	8
RESERVED	xbar	cmpss3	cmpss2	cmpss1	cmpss0	pwm4	pwm3
R/W-XXh	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
pwm2	pwm1	pwm0	ecap1	ecap0	eqep2	eqep1	eqep0
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 3-139. PWREN_MCPERIPH Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	KEY	W	0h	KEY to Allow Enable State Change -- 0x26 26h = 0x26
23-15	RESERVED	R/W	XXh	
14	xbar	R/W	0h	Enable the power to IP, KEY must be set to 26h to write to this bit 0h = Disable Power 1h = Enable Power
13	cmpss3	R/W	0h	Enable the power to IP, KEY must be set to 26h to write to this bit 0h = Disable Power 1h = Enable Power
12	cmpss2	R/W	0h	Enable the power to IP, KEY must be set to 26h to write to this bit 0h = Disable Power 1h = Enable Power
11	cmpss1	R/W	0h	Enable the power to IP, KEY must be set to 26h to write to this bit 0h = Disable Power 1h = Enable Power
10	cmpss0	R/W	0h	Enable the power to IP, KEY must be set to 26h to write to this bit 0h = Disable Power 1h = Enable Power
9	pwm4	R/W	0h	Enable the power to IP, KEY must be set to 26h to write to this bit 0h = Disable Power 1h = Enable Power
8	pwm3	R/W	0h	Enable the power to IP, KEY must be set to 26h to write to this bit 0h = Disable Power 1h = Enable Power
7	pwm2	R/W	0h	Enable the power to IP, KEY must be set to 26h to write to this bit 0h = Disable Power 1h = Enable Power
6	pwm1	R/W	0h	Enable the power to IP, KEY must be set to 26h to write to this bit 0h = Disable Power 1h = Enable Power

Table 3-139. PWREN_MCPERIPH Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
5	pwm0	R/W	0h	Enable the power to IP, KEY must be set to 26h to write to this bit 0h = Disable Power 1h = Enable Power
4	ecap1	R/W	0h	Enable the power to IP, KEY must be set to 26h to write to this bit 0h = Disable Power 1h = Enable Power
3	ecap0	R/W	0h	Enable the power to IP, KEY must be set to 26h to write to this bit 0h = Disable Power 1h = Enable Power
2	eqep2	R/W	0h	Enable the power to IP, KEY must be set to 26h to write to this bit 0h = Disable Power 1h = Enable Power
1	eqep1	R/W	0h	Enable the power to IP, KEY must be set to 26h to write to this bit 0h = Disable Power 1h = Enable Power
0	eqep0	R/W	0h	Enable the power to IP, KEY must be set to 26h to write to this bit 0h = Disable Power 1h = Enable Power

58 RSTCTL_ASSERT_MCPERIPH Register (Offset = 1428h) [Reset = 00XXX000h]

RSTCTL_ASSERT_MCPERIPH is shown in Figure 3-68 and described in Table 3-83.

Return to the Summary Table.

rstctl assert register to control reset assertion - Write Only Register, Always Read as 0

Figure 3-68. RSTCTL_ASSERT_MCPERIPH Register

31	30	29	28	27	26	25	24
KEY							
W-0h							
23	22	21	20	19	18	17	16
RESERVED							
W-XXh							
15	14	13	12	11	10	9	8
RESERVED	RESETASSERT_xbar	RESETASSERT_cmpss3	RESETASSERT_cmpss2	RESETASSERT_cmpss1	RESETASSERT_cmpss0	RESETASSERT_pwm4	RESETASSERT_pwm3
W-XXh	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h
7	6	5	4	3	2	1	0
RESETASSERT_pwm2	RESETASSERT_pwm1	RESETASSERT_pwm0	RESETASSERT_ecap1	RESETASSERT_ecap0	RESETASSERT_eqep2	RESETASSERT_eqep1	RESETASSERT_eqep0
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h

Table 3-141. RSTCTL_ASSERT_MCPERIPH Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	KEY	W	0h	KEY to Allow Enable State Change -- 0xb1 B1h = 0xb1
23-15	RESERVED	W	XXh	
14	RESETASSERT_xbar	W	0h	assert reset to the peripheral, KEY must be set to B1h to write to this bit 0h = Writing 0 has no effect 1h = Assert reset
13	RESETASSERT_cmpss3	W	0h	assert reset to the peripheral, KEY must be set to B1h to write to this bit 0h = Writing 0 has no effect 1h = Assert reset
12	RESETASSERT_cmpss2	W	0h	assert reset to the peripheral, KEY must be set to B1h to write to this bit 0h = Writing 0 has no effect 1h = Assert reset
11	RESETASSERT_cmpss1	W	0h	assert reset to the peripheral, KEY must be set to B1h to write to this bit 0h = Writing 0 has no effect 1h = Assert reset
10	RESETASSERT_cmpss0	W	0h	assert reset to the peripheral, KEY must be set to B1h to write to this bit 0h = Writing 0 has no effect 1h = Assert reset
9	RESETASSERT_pwm4	W	0h	assert reset to the peripheral, KEY must be set to B1h to write to this bit 0h = Writing 0 has no effect 1h = Assert reset
8	RESETASSERT_pwm3	W	0h	assert reset to the peripheral, KEY must be set to B1h to write to this bit 0h = Writing 0 has no effect 1h = Assert reset

Table 3-141. RSTCTL_ASSERT_MCPERIPH Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
7	RESETASSERT_pwm2	W	0h	assert reset to the peripheral, KEY must be set to B1h to write to this bit 0h = Writing 0 has no effect 1h = Assert reset
6	RESETASSERT_pwm1	W	0h	assert reset to the peripheral, KEY must be set to B1h to write to this bit 0h = Writing 0 has no effect 1h = Assert reset
5	RESETASSERT_pwm0	W	0h	assert reset to the peripheral, KEY must be set to B1h to write to this bit 0h = Writing 0 has no effect 1h = Assert reset
4	RESETASSERT_ecap1	W	0h	assert reset to the peripheral, KEY must be set to B1h to write to this bit 0h = Writing 0 has no effect 1h = Assert reset
3	RESETASSERT_ecap0	W	0h	assert reset to the peripheral, KEY must be set to B1h to write to this bit 0h = Writing 0 has no effect 1h = Assert reset
2	RESETASSERT_eqep2	W	0h	assert reset to the peripheral, KEY must be set to B1h to write to this bit 0h = Writing 0 has no effect 1h = Assert reset
1	RESETASSERT_eqep1	W	0h	assert reset to the peripheral, KEY must be set to B1h to write to this bit 0h = Writing 0 has no effect 1h = Assert reset
0	RESETASSERT_eqep0	W	0h	assert reset to the peripheral, KEY must be set to B1h to write to this bit 0h = Writing 0 has no effect 1h = Assert reset

59 RSTCTL_CLEAR_MCPERIPH Register (Offset = 142Ch) [Reset = 00XXX000h]

RSTCTL_CLEAR_MCPERIPH is shown in Figure 3-69 and described in Table 3-84.

Return to the Summary Table.

rstctl clear register to control reset de-assertion - Write Only Register, Always Read as 0

Figure 3-69. RSTCTL_CLEAR_MCPERIPH Register

31	30	29	28	27	26	25	24
KEY							
W-0h							
23	22	21	20	19	18	17	16
RESERVED							
W-XXh							
15	14	13	12	11	10	9	8
RESERVED	RESETSTKYCLR_xbar	RESETSTKYCLR_cmpss3	RESETSTKYCLR_cmpss2	RESETSTKYCLR_cmpss1	RESETSTKYCLR_cmpss0	RESETSTKYCLR_pwm4	RESETSTKYCLR_pwm3
W-XXh	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h
7	6	5	4	3	2	1	0
RESETSTKYCLR_pwm2	RESETSTKYCLR_pwm1	RESETSTKYCLR_pwm0	RESETSTKYCLR_ecap1	RESETSTKYCLR_ecap0	RESETSTKYCLR_eqep2	RESETSTKYCLR_eqep1	RESETSTKYCLR_eqep0
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h

Table 3-143. RSTCTL_CLEAR_MCPERIPH Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	KEY	W	0h	KEY to Allow Enable State Change -- 0xb1 B1h = 0xb1
23-15	RESERVED	W	XXh	
14	RESETSTKYCLR_xbar	W	0h	Clear the RESETSTKY bit in the STAT register, KEY must be set to B1h to write to this bit 1h = Writing 0 has no effect
13	RESETSTKYCLR_cmpss3	W	0h	Clear the RESETSTKY bit in the STAT register, KEY must be set to B1h to write to this bit 1h = Writing 0 has no effect
12	RESETSTKYCLR_cmpss2	W	0h	Clear the RESETSTKY bit in the STAT register, KEY must be set to B1h to write to this bit 1h = Writing 0 has no effect
11	RESETSTKYCLR_cmpss1	W	0h	Clear the RESETSTKY bit in the STAT register, KEY must be set to B1h to write to this bit 1h = Writing 0 has no effect
10	RESETSTKYCLR_cmpss0	W	0h	Clear the RESETSTKY bit in the STAT register, KEY must be set to B1h to write to this bit 1h = Writing 0 has no effect
9	RESETSTKYCLR_pwm4	W	0h	Clear the RESETSTKY bit in the STAT register, KEY must be set to B1h to write to this bit 1h = Writing 0 has no effect
8	RESETSTKYCLR_pwm3	W	0h	Clear the RESETSTKY bit in the STAT register, KEY must be set to B1h to write to this bit 1h = Writing 0 has no effect
7	RESETSTKYCLR_pwm2	W	0h	Clear the RESETSTKY bit in the STAT register, KEY must be set to B1h to write to this bit 1h = Writing 0 has no effect
6	RESETSTKYCLR_pwm1	W	0h	Clear the RESETSTKY bit in the STAT register, KEY must be set to B1h to write to this bit 1h = Writing 0 has no effect

Table 3-143. RSTCTL_CLEAR_MCPERIPH Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
5	RESETSTKYCLR_pwm0	W	0h	Clear the RESETSTKY bit in the STAT register, KEY must be set to B1h to write to this bit 1h = Writing 0 has no effect
4	RESETSTKYCLR_ecap1	W	0h	Clear the RESETSTKY bit in the STAT register, KEY must be set to B1h to write to this bit 1h = Writing 0 has no effect
3	RESETSTKYCLR_ecap0	W	0h	Clear the RESETSTKY bit in the STAT register, KEY must be set to B1h to write to this bit 1h = Writing 0 has no effect
2	RESETSTKYCLR_eqep2	W	0h	Clear the RESETSTKY bit in the STAT register, KEY must be set to B1h to write to this bit 1h = Writing 0 has no effect
1	RESETSTKYCLR_eqep1	W	0h	Clear the RESETSTKY bit in the STAT register, KEY must be set to B1h to write to this bit 1h = Writing 0 has no effect
0	RESETSTKYCLR_eqep0	W	0h	Clear the RESETSTKY bit in the STAT register, KEY must be set to B1h to write to this bit 1h = Writing 0 has no effect

60 STAT_MCPERIPH Register (Offset = 1430h) [Reset = 00000000h]

STAT_MCPERIPH is shown in Figure 3-70 and described in Table 3-85.

Return to the Summary Table.

IP State Register - Read Only

Figure 3-70. STAT_MCPERIPH Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED	xbar	cmpss3	cmpss2	cmpss1	cmpss0	pwm4	pwm3
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
7	6	5	4	3	2	1	0
pwm2	pwm1	pwm0	ecap1	ecap0	eqep2	eqep1	eqep0
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

Table 3-145. STAT_MCPERIPH Register Field Descriptions

Bit	Field	Type	Reset	Description
31-15	RESERVED	R	0h	
14	xbar	R	0h	This bit indicates, if the peripheral was reset, since this bit was cleared by RESETSTKYCLR in the RSTCTL register 0h = The peripheral has not been reset since this bit was last cleared by RESETSTKYCLR in the RSTCTL register 1h = The peripheral was reset since the last bit clear
13	cmpss3	R	0h	This bit indicates, if the peripheral was reset, since this bit was cleared by RESETSTKYCLR in the RSTCTL register 0h = The peripheral has not been reset since this bit was last cleared by RESETSTKYCLR in the RSTCTL register 1h = The peripheral was reset since the last bit clear
12	cmpss2	R	0h	This bit indicates, if the peripheral was reset, since this bit was cleared by RESETSTKYCLR in the RSTCTL register 0h = The peripheral has not been reset since this bit was last cleared by RESETSTKYCLR in the RSTCTL register 1h = The peripheral was reset since the last bit clear
11	cmpss1	R	0h	This bit indicates, if the peripheral was reset, since this bit was cleared by RESETSTKYCLR in the RSTCTL register 0h = The peripheral has not been reset since this bit was last cleared by RESETSTKYCLR in the RSTCTL register 1h = The peripheral was reset since the last bit clear
10	cmpss0	R	0h	This bit indicates, if the peripheral was reset, since this bit was cleared by RESETSTKYCLR in the RSTCTL register 0h = The peripheral has not been reset since this bit was last cleared by RESETSTKYCLR in the RSTCTL register 1h = The peripheral was reset since the last bit clear
9	pwm4	R	0h	This bit indicates, if the peripheral was reset, since this bit was cleared by RESETSTKYCLR in the RSTCTL register 0h = The peripheral has not been reset since this bit was last cleared by RESETSTKYCLR in the RSTCTL register 1h = The peripheral was reset since the last bit clear

Table 3-145. STAT_MCPERIPH Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
8	pwm3	R	0h	This bit indicates, if the peripheral was reset, since this bit was cleared by RESETSTKYCLR in the RSTCTL register 0h = The peripheral has not been reset since this bit was last cleared by RESETSTKYCLR in the RSTCTL register 1h = The peripheral was reset since the last bit clear
7	pwm2	R	0h	This bit indicates, if the peripheral was reset, since this bit was cleared by RESETSTKYCLR in the RSTCTL register 0h = The peripheral has not been reset since this bit was last cleared by RESETSTKYCLR in the RSTCTL register 1h = The peripheral was reset since the last bit clear
6	pwm1	R	0h	This bit indicates, if the peripheral was reset, since this bit was cleared by RESETSTKYCLR in the RSTCTL register 0h = The peripheral has not been reset since this bit was last cleared by RESETSTKYCLR in the RSTCTL register 1h = The peripheral was reset since the last bit clear
5	pwm0	R	0h	This bit indicates, if the peripheral was reset, since this bit was cleared by RESETSTKYCLR in the RSTCTL register 0h = The peripheral has not been reset since this bit was last cleared by RESETSTKYCLR in the RSTCTL register 1h = The peripheral was reset since the last bit clear
4	ecap1	R	0h	This bit indicates, if the peripheral was reset, since this bit was cleared by RESETSTKYCLR in the RSTCTL register 0h = The peripheral has not been reset since this bit was last cleared by RESETSTKYCLR in the RSTCTL register 1h = The peripheral was reset since the last bit clear
3	ecap0	R	0h	This bit indicates, if the peripheral was reset, since this bit was cleared by RESETSTKYCLR in the RSTCTL register 0h = The peripheral has not been reset since this bit was last cleared by RESETSTKYCLR in the RSTCTL register 1h = The peripheral was reset since the last bit clear
2	eqep2	R	0h	This bit indicates, if the peripheral was reset, since this bit was cleared by RESETSTKYCLR in the RSTCTL register 0h = The peripheral has not been reset since this bit was last cleared by RESETSTKYCLR in the RSTCTL register 1h = The peripheral was reset since the last bit clear
1	eqep1	R	0h	This bit indicates, if the peripheral was reset, since this bit was cleared by RESETSTKYCLR in the RSTCTL register 0h = The peripheral has not been reset since this bit was last cleared by RESETSTKYCLR in the RSTCTL register 1h = The peripheral was reset since the last bit clear
0	eqep0	R	0h	This bit indicates, if the peripheral was reset, since this bit was cleared by RESETSTKYCLR in the RSTCTL register 0h = The peripheral has not been reset since this bit was last cleared by RESETSTKYCLR in the RSTCTL register 1h = The peripheral was reset since the last bit clear

61 PWREN_SYSPERIPH Register (Offset = 1434h) [Reset = 00XXXXX0h]

PWREN_SYSPERIPH is shown in Figure 3-71 and described in Table 3-86.

Return to the Summary Table.

Register to control the power state

Figure 3-71. PWREN_SYSPERIPH Register

31	30	29	28	27	26	25	24
KEY							
W-0h							
23	22	21	20	19	18	17	16
RESERVED							
R/W-XXXXh							
15	14	13	12	11	10	9	8
RESERVED							
R/W-XXXXh							
7	6	5	4	3	2	1	0
RESERVED			epi	pga2	pga1	pga0	tinie
R/W-XXXXh			R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 3-147. PWREN_SYSPERIPH Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	KEY	W	0h	KEY to Allow Enable State Change -- 0x26 26h = 0x26
23-5	RESERVED	R/W	XXXXh	
4	epi	R/W	0h	Enable the power to IP, KEY must be set to 26h to write to this bit 0h = Disable Power 1h = Enable Power
3	pga2	R/W	0h	Enable the power to IP, KEY must be set to 26h to write to this bit 0h = Disable Power 1h = Enable Power
2	pga1	R/W	0h	Enable the power to IP, KEY must be set to 26h to write to this bit 0h = Disable Power 1h = Enable Power
1	pga0	R/W	0h	Enable the power to IP, KEY must be set to 26h to write to this bit 0h = Disable Power 1h = Enable Power
0	tinie	R/W	0h	Enable the power to IP, KEY must be set to 26h to write to this bit 0h = Disable Power 1h = Enable Power

62 RSTCTL_ASSERT_SYSPERIPH Register (Offset = 1438h) [Reset = 00XXXXX0h]

RSTCTL_ASSERT_SYSPERIPH is shown in Figure 3-72 and described in Table 3-87.

Return to the Summary Table.

rstcl assert Register - Write Only Register, Always Read as 0

Figure 3-72. RSTCTL_ASSERT_SYSPERIPH Register

31	30	29	28	27	26	25	24
KEY							
W-0h							
23	22	21	20	19	18	17	16
RESERVED							
W-XXXXh							
15	14	13	12	11	10	9	8
RESERVED							
W-XXXXh							
7	6	5	4	3	2	1	0
RESERVED			RESETASSERT_epi	RESETASSERT_pga2	RESETASSERT_pga1	RESETASSERT_pga0	RESETASSERT_tinie
W-XXXXh			W-0h	W-0h	W-0h	W-0h	W-0h

Table 3-149. RSTCTL_ASSERT_SYSPERIPH Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	KEY	W	0h	KEY to Allow Enable State Change -- 0xb1 B1h = 0xb1
23-5	RESERVED	W	XXXXh	
4	RESETASSERT_epi	W	0h	assert reset to the peripheral, KEY must be set to B1h to write to this bit 0h = Writing 0 has no effect 1h = Assert reset
3	RESETASSERT_pga2	W	0h	assert reset to the peripheral, KEY must be set to B1h to write to this bit 0h = Writing 0 has no effect 1h = Assert reset
2	RESETASSERT_pga1	W	0h	assert reset to the peripheral, KEY must be set to B1h to write to this bit 0h = Writing 0 has no effect 1h = Assert reset
1	RESETASSERT_pga0	W	0h	assert reset to the peripheral, KEY must be set to B1h to write to this bit 0h = Writing 0 has no effect 1h = Assert reset
0	RESETASSERT_tinie	W	0h	assert reset to the peripheral, KEY must be set to B1h to write to this bit 0h = Writing 0 has no effect 1h = Assert reset

63 RSTCTL_CLEAR_SYSPERIPH Register (Offset = 143Ch) [Reset = 00XXXXX0h]

RSTCTL_CLEAR_SYSPERIPH is shown in Figure 3-73 and described in Table 3-88.

Return to the Summary Table.

rstctl clear register to control reset de-assertion - Write Only Register, Always Read as 0

Figure 3-73. RSTCTL_CLEAR_SYSPERIPH Register

31	30	29	28	27	26	25	24
KEY							
W-0h							
23	22	21	20	19	18	17	16
RESERVED							
W-XXXXh							
15	14	13	12	11	10	9	8
RESERVED							
W-XXXXh							
7	6	5	4	3	2	1	0
RESERVED			RESETSTKYCLR_epi	RESETSTKYCLR_pga2	RESETSTKYCLR_pga1	RESETSTKYCLR_pga0	RESETSTKYCLR_tinie
W-XXXXh			W-0h	W-0h	W-0h	W-0h	W-0h

Table 3-151. RSTCTL_CLEAR_SYSPERIPH Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	KEY	W	0h	KEY to Allow Enable State Change -- 0xb1 B1h = 0xb1
23-5	RESERVED	W	XXXXh	
4	RESETSTKYCLR_epi	W	0h	Clear the RESETSTKY bit in the STAT register, KEY must be set to B1h to write to this bit 1h = Writing 0 has no effect
3	RESETSTKYCLR_pga2	W	0h	Clear the RESETSTKY bit in the STAT register, KEY must be set to B1h to write to this bit 1h = Writing 0 has no effect
2	RESETSTKYCLR_pga1	W	0h	Clear the RESETSTKY bit in the STAT register, KEY must be set to B1h to write to this bit 1h = Writing 0 has no effect
1	RESETSTKYCLR_pga0	W	0h	Clear the RESETSTKY bit in the STAT register, KEY must be set to B1h to write to this bit 1h = Writing 0 has no effect
0	RESETSTKYCLR_tinie	W	0h	Clear the RESETSTKY bit in the STAT register, KEY must be set to B1h to write to this bit 1h = Writing 0 has no effect

64 STAT_SYSPERIPH Register (Offset = 1440h) [Reset = 00000000h]

STAT_SYSPERIPH is shown in Figure 3-74 and described in Table 3-89.

Return to the Summary Table.

IP State Register - Read Only

Figure 3-74. STAT_SYSPERIPH Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED			epi	pga2	pga1	pga0	tinie
R-0h			R-0h	R-0h	R-0h	R-0h	R-0h

Table 3-153. STAT_SYSPERIPH Register Field Descriptions

Bit	Field	Type	Reset	Description
31-5	RESERVED	R	0h	
4	epi	R	0h	This bit indicates, if the peripheral was reset, since this bit was cleared by RESETSTKYCLR in the RSTCTL register 0h = The peripheral has not been reset since this bit was last cleared by RESETSTKYCLR in the RSTCTL register 1h = The peripheral was reset since the last bit clear
3	pga2	R	0h	This bit indicates, if the peripheral was reset, since this bit was cleared by RESETSTKYCLR in the RSTCTL register 0h = The peripheral has not been reset since this bit was last cleared by RESETSTKYCLR in the RSTCTL register 1h = The peripheral was reset since the last bit clear
2	pga1	R	0h	This bit indicates, if the peripheral was reset, since this bit was cleared by RESETSTKYCLR in the RSTCTL register 0h = The peripheral has not been reset since this bit was last cleared by RESETSTKYCLR in the RSTCTL register 1h = The peripheral was reset since the last bit clear
1	pga0	R	0h	This bit indicates, if the peripheral was reset, since this bit was cleared by RESETSTKYCLR in the RSTCTL register 0h = The peripheral has not been reset since this bit was last cleared by RESETSTKYCLR in the RSTCTL register 1h = The peripheral was reset since the last bit clear
0	tinie	R	0h	This bit indicates, if the peripheral was reset, since this bit was cleared by RESETSTKYCLR in the RSTCTL register 0h = The peripheral has not been reset since this bit was last cleared by RESETSTKYCLR in the RSTCTL register 1h = The peripheral was reset since the last bit clear

65 CMPHPMXSEL Register (Offset = 1444h) [Reset = 00000000h]

CMPHPMXSEL is shown in Figure 3-75 and described in Table 3-90.

Return to the Summary Table.

Bits to select one of the many sources on CompHP inputs. Refer to Pinmux diagram for details.

Figure 3-75. CMPHPMXSEL Register

31	30	29	28	27	26	25	24
RESERVED							
R/W-0h							
23	22	21	20	19	18	17	16
RESERVED							
R/W-0h							
15	14	13	12	11	10	9	8
RESERVED				CMP3HPMXSEL			CMP2HPMXSEL
R/W-0h				R/W-0h			R/W-0h
7	6	5	4	3	2	1	0
CMP2HPMXSEL		CMP1HPMXSEL			CMP0HPMXSEL		
R/W-0h		R/W-0h			R/W-0h		

Table 3-155. CMPHPMXSEL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-12	RESERVED	R/W	0h	
11-9	CMP3HPMXSEL	R/W	0h	CMP3HPMXSEL bits
8-6	CMP2HPMXSEL	R/W	0h	CMP2HPMXSEL bits
5-3	CMP1HPMXSEL	R/W	0h	CMP1HPMXSEL bits
2-0	CMP0HPMXSEL	R/W	0h	CMP0HPMXSEL bits

66 CMPLPMXSEL Register (Offset = 144Ch) [Reset = 00000000h]

CMPLPMXSEL is shown in Figure 3-76 and described in Table 3-91.

Return to the Summary Table.

Bits to select one of the many sources on CompLP inputs. Refer to Pinmux diagram for details.

Figure 3-76. CMPLPMXSEL Register

31	30	29	28	27	26	25	24
RESERVED							
R/W-0h							
23	22	21	20	19	18	17	16
RESERVED							
R/W-0h							
15	14	13	12	11	10	9	8
RESERVED				CMP3LPMXSEL			CMP2LPMXSEL
R/W-0h				R/W-0h			R/W-0h
7	6	5	4	3	2	1	0
CMP2LPMXSEL		CMP1LPMXSEL			CMP0LPMXSEL		
R/W-0h		R/W-0h			R/W-0h		

Table 3-157. CMPLPMXSEL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-12	RESERVED	R/W	0h	
11-9	CMP3LPMXSEL	R/W	0h	CMP3LPMXSEL bits
8-6	CMP2LPMXSEL	R/W	0h	CMP2LPMXSEL bits
5-3	CMP1LPMXSEL	R/W	0h	CMP1LPMXSEL bits
2-0	CMP0LPMXSEL	R/W	0h	CMP0LPMXSEL bits

67 CMPHNMXSEL Register (Offset = 1450h) [Reset = 00000000h]

CMPHNMXSEL is shown in Figure 3-77 and described in Table 3-92.

Return to the Summary Table.

Bits to select one of the many sources on CompHN inputs. Refer to Pinmux diagram for details.

Figure 3-77. CMPHNMXSEL Register

31	30	29	28	27	26	25	24
RESERVED							
R/W-0h							
23	22	21	20	19	18	17	16
RESERVED							
R/W-0h							
15	14	13	12	11	10	9	8
RESERVED							
R/W-0h							
7	6	5	4	3	2	1	0
RESERVED				CMP3HNMXSEL	CMP2HNMXSEL	CMP1HNMXSEL	CMP0HNMXSEL
R/W-0h				R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 3-159. CMPHNMXSEL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R/W	0h	
3	CMP3HNMXSEL	R/W	0h	CMP3HNMXSEL bits
2	CMP2HNMXSEL	R/W	0h	CMP2HNMXSEL bits
1	CMP1HNMXSEL	R/W	0h	CMP1HNMXSEL bits
0	CMP0HNMXSEL	R/W	0h	CMP0HNMXSEL bits

68 CMPLNMXSEL Register (Offset = 1454h) [Reset = 00000000h]

CMPLNMXSEL is shown in Figure 3-78 and described in Table 3-93.

Return to the Summary Table.

Bits to select one of the many sources on CompLN inputs. Refer to Pinmux diagram for details.

Figure 3-78. CMPLNMXSEL Register

31	30	29	28	27	26	25	24
RESERVED							
R/W-0h							
23	22	21	20	19	18	17	16
RESERVED							
R/W-0h							
15	14	13	12	11	10	9	8
RESERVED							
R/W-0h							
7	6	5	4	3	2	1	0
RESERVED				CMP3LNMXSEL	CMP2LNMXSEL	CMP1LNMXSEL	CMP0LNMXSEL
R/W-0h				R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 3-161. CMPLNMXSEL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R/W	0h	
3	CMP3LNMXSEL	R/W	0h	CMP3LNMXSEL bits
2	CMP2LNMXSEL	R/W	0h	CMP2LNMXSEL bits
1	CMP1LNMXSEL	R/W	0h	CMP1LNMXSEL bits
0	CMP0LNMXSEL	R/W	0h	CMP0LNMXSEL bits

69 TSNSCFG Register (Offset = 1458h) [Reset = 00000000h]

TSNSCFG is shown in Figure 3-79 and described in Table 3-94.

Return to the Summary Table.

Temperature Sensor Config Register

Figure 3-79. TSNSCFG Register

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

Table 3-163. TSNSCFG Register Field Descriptions

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

70 TSNSSCTL Register (Offset = 145Ch) [Reset = 00000000h]

TSNSSCTL is shown in Figure 3-80 and described in Table 3-95.

Return to the Summary Table.

Temperature Sensor Control Register

Figure 3-80. TSNSSCTL Register

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

Table 3-165. TSNSSCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R/W	0h	
0	ENABLE	R/W	0h	Temperature Sensor Enable

ADVANCE INFORMATION

71 PGACONFIG Register (Offset = 1460h) [Reset = 00000000h]

PGACONFIG is shown in Figure 3-81 and described in Table 3-96.

Return to the Summary Table.

PGA Configuration Register

Figure 3-81. PGACONFIG Register

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

Table 3-167. PGACONFIG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-20	RESERVED	R/W	0h	
19-18	RESERVED	R/W	0h	Reserved
17-14	RESERVED	R/W	0h	Reserved
13	RESERVED	R/W	0h	Reserved
12-7	RESERVED	R/W	0h	Reserved
6	RESERVED	R/W	0h	Reserved
5	RESERVED	R/W	0h	Reserved
4	RESERVED	R/W	0h	Reserved
3	RESERVED	R/W	0h	Reserved
2	RESERVED	R/W	0h	Reserved
1	RESERVED	R/W	0h	Reserved
0	RESERVED	R/W	0h	Reserved

72 REFCONFIGA Register (Offset = 1464h) [Reset = 00000000h]

REFCONFIGA is shown in Figure 3-82 and described in Table 3-97.

Return to the Summary Table.

Reference Configuration Regsiter

Figure 3-82. REFCONFIGA Register

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

Table 3-169. REFCONFIGA Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	Reserved
30	RESERVED	R/W	0h	Reserved
29-28	RESERVED	R/W	0h	Reserved
27	RESERVED	R/W	0h	Reserved
26-21	RESERVED	R/W	0h	Reserved
20	RESERVED	R/W	0h	Reserved
19	RESERVED	R/W	0h	Reserved
18-15	RESERVED	R/W	0h	Reserved
14-12	RESERVED	R/W	0h	Reserved
11-5	RESERVED	R/W	0h	Reserved
4	RESERVED	R/W	0h	Reserved
3	RESERVED	R/W	0h	Reserved
2	RESERVED	R/W	0h	Reserved
1	RESERVED	R/W	0h	Reserved
0	RESERVED	R/W	0h	Reserved

73 INTERNALTESTCTL Register (Offset = 1468h) [Reset = 00000000h]

INTERNALTESTCTL is shown in Figure 3-83 and described in Table 3-98.

Return to the Summary Table.

Internal Test Node Control Register

Figure 3-83. INTERNALTESTCTL Register

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

Table 3-171. INTERNALTESTCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	RESERVED	R/W	0h	
8-6	RESERVED	R/W	0h	Reserved

ADVANCE INFORMATION

Table 3-171. INTERNALTESTCTL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
5-0	TESTSEL	R/W	0h	Test Select 1h = VDDCORE 2h = VDDA 3h = VSSA 4h = VREFLOAC 5h = CDAC1H 6h = CDAC1L 7h = CDAC2H 8h = CDAC2H 9h = CDAC2H Ah = CDAC2H Bh = CDAC2H Ch = CDAC2H 1Dh = ENZ_CALIB_GAIN_3P3V will be made low. ADCA and ADCC will be in gain calibration mode, and 0.9xVREFHIAB pin voltage will be sampled by both ADCs through internal test-mux output 1Eh = CMPSS1 VDDA sense on TESTANA0,VSSA sense on TESTANA1 1Fh = ADCA VDDA sense on TESTANA0,VSSA sense on TESTANA1 20h = COMP DAC BUFFER VDDA sense on TESTANA0,VSSA sense on TESTANA1 21h = PGA1 VDDA sense on TESTANA0,VSSA sense on TESTANA1 22h = ADCCIO_TESTANA0_INT 23h = PMM/HPLL/INTOSC TESTANA0_INT 24h = ADCCIO_TESTANA1_INT 25h = PMM/HPLL/INTOSC TESTANA1_INT 26h = Enable resistor for I2V conversion. The same control enables the sampling of voltage across a resistor by ADC. R=2.5k, 38: R=10k, 39: R=35k 29h = USB_TESTANA0_INT 2Ah = USB_TESTANA1_INT 2Bh = USB_TESTANA0_INT & USB_TESTANA1_INT 2Ch = VSS 2Dh = Bring FLT3 & TESTPAD3 of flash on TESTANA1 2Eh = Enable resistor for I2V conversion. The same control enables the sampling of voltage across a resistor by ADC. R=2.5k, 47: R=10k, 48: R=35k 31h = VREFLOAC

74 I2VCTL Register (Offset = 146Ch) [Reset = 00000000h]

I2VCTL is shown in Figure 3-84 and described in Table 3-99.

Return to the Summary Table.

I2V Logic Control

Figure 3-84. I2VCTL Register

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

Table 3-173. I2VCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R/W	0h	
7-0	RESERVED	R/W	0h	Reserved

ADVANCE INFORMATION

75 ADCDACLOOPBACK Register (Offset = 1470h) [Reset = 00000000h]

ADCDACLOOPBACK is shown in Figure 3-85 and described in Table 3-100.

Return to the Summary Table.

Not used in AM13

Figure 3-85. ADCDACLOOPBACK Register

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

Table 3-175. ADCDACLOOPBACK Register Field Descriptions

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

76 XTALCR Register (Offset = 1474h) [Reset = 00000001h]

XTALCR is shown in Figure 3-86 and described in Table 3-101.

Return to the Summary Table.

XTAL Control Register

Figure 3-86. XTALCR Register

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

Table 3-177. XTALCR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R/W	0h	
1	SE	R/W	0h	XTAL Oscilator in Single-Ended 0h = XTAL oscillator in Crystal mode 1h = XTAL oscilator in single-ended mode (through X1)
0	OSCOFF	R/W	1h	This bit if 1, powers-down the XTAL oscillator macro and hence doesnt let X2 to be driven by the XTAL oscillator. If a crystal is connected to X1/X2, user needs to first clear this bit, wait for the oscillator to power up (using X1CNT) and then only switch the clock source to X1/X2 0h = XTAL Oscillator powered-up using X1/X2 1h = XTAL Oscillator powered-down

ADVANCE INFORMATION

77 XTALCR2 Register (Offset = 1478h) [Reset = 0000XXXXh]

XTALCR2 is shown in Figure 3-87 and described in Table 3-102.

Return to the Summary Table.

XTAL Control Register for pad init

Figure 3-87. XTALCR2 Register

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

Table 3-179. XTALCR2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	FKEEPXI	R/W	0h	This field when written 0xface allows to hold the force value on XI as programmed on XIF. 0xface: Force on XI is continued as per XIF value regardless of XOSC ON/OFF state only in XTAL mode. In Single ended mode this field has no impact. Any other value: Force on XI is removed with enabling of XOSC as per FEN function.
15-3	RESERVED	R/W	XXXh	
2	FEN	R/W	0h	XOSC pads initialisation enable. Configures XTAL oscillator pad initilisation. This register has effect only when XOSC is OFF (no SE , no XTAL mode). If this register is set during XOSC off state (XOSCOFF=1 & SE=0) then upon change of these controls this bit gets reset and rearmed 0h = XOSC pads are not driven through GPIO connection. 1h = XOSC pads are driven through connected GPIO as per XIF & XOF values.
1	XOF	R/W	1h	XO Initial value deposited before XOSC start. Polarity selection to initialise XO/X2 pad of the XOSC before start-up. This value shall be deposited on the pad before XOSC started (XOSCOFF=1). If FEN=0 or XOSC is in XTAL or SE mode then this value will not be applied to the pad.
0	XIF	R/W	1h	XI Initial value deposited before XOSC start. Polarity selection to initialise XI/X1 pad of the XOSC before start-up. This value shall be deposited on the pad before XOSC started (XOSCOFF=1). If FEN=0 or XOSC is in XTAL or SE mode then this value will not be applied to the pad.

ADVANCE INFORMATION

78 X1CNT Register (Offset = 147Ch) [Reset = 0000XX00h]

X1CNT is shown in Figure 3-88 and described in Table 3-103.

Return to the Summary Table.

x1cnt status register

Figure 3-88. X1CNT Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															CLR
R/W-0h															W-0h
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED					X1CNT										
R/W-Xh					R-0h										

Table 3-181. X1CNT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-17	RESERVED	R/W	0h	
16	CLR	W	0h	X1 Counter clear: A write of 1 to this bit field clears the X1CNT and makes it count from 0x0 again (provided X1 clock is ticking). Writes of 0 are ignore to this bit field
15-11	RESERVED	R/W	Xh	
10-0	X1CNT	R	0h	This counter increments on every X1 CLOCKs positive-edge. Once it reaches the values of 0x7ff, it freezes. Before switching from SYSOSC/PLL clock to X1, application must check this counter and make sure that it has saturated. This will ensure that the Crystal connected to X1/X2 is oscillating

79 CMPSSCTL Register (Offset = 1480h) [Reset = XXXXXXXXh]

CMPSSCTL is shown in Figure 3-89 and described in Table 3-104.

Return to the Summary Table.

CMPSS control register

Figure 3-89. CMPSSCTL Register

31	30	29	28	27	26	25	24
CMPSSCTLEN	RESERVED						
R/W-0h	R/W-000XXXXXh						
23	22	21	20	19	18	17	16
RESERVED							
R/W-000XXXXXh							
15	14	13	12	11	10	9	8
RESERVED							
R/W-000XXXXXh							
7	6	5	4	3	2	1	0
RESERVED						CMP3LDACOUTEN	CMP2LDACOUTEN
R/W-000XXXXXh						R/W-0h	R/W-0h

Table 3-183. CMPSSCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31	CMPSSCTLEN	R/W	0h	Enable the CMPSSCTL Register
30-2	RESERVED	R/W	XXXXh	
1	CMP3LDACOUTEN	R/W	0h	Enable general purpose DAC functionality for CMPSS3 COMPDACL
0	CMP2LDACOUTEN	R/W	0h	Enable general purpose DAC functionality for CMPSS2 COMPDACL

80 CMPSSDACBUFCONFIG Register (Offset = 1484h) [Reset = 00000000h]

CMPSSDACBUFCONFIG is shown in Figure 3-90 and described in Table 3-105.

Return to the Summary Table.

Config bits for CMPSS DAC buffer

Figure 3-90. CMPSSDACBUFCONFIG Register

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

Table 3-185. CMPSSDACBUFCONFIG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R/W	0h	
15-8	RESERVED	R/W	0h	Reserved
7-0	RESERVED	R/W	0h	Reserved

ADVANCE INFORMATION

81 ANAREFCTL Register (Offset = 1488h) [Reset = 000000XXh]

ANAREFCTL is shown in Figure 3-91 and described in Table 3-106.

Return to the Summary Table.

Analog Reference Select

Figure 3-91. ANAREFCTL Register

31	30	29	28	27	26	25	24
RESERVED							
R/W-0h							
23	22	21	20	19	18	17	16
RESERVED							
R/W-0h							
15	14	13	12	11	10	9	8
RESERVED							ANAREF2P5SEL
R/W-0h							R/W-0h
7	6	5	4	3	2	1	0
RESERVED					RESERVED	RESERVED	ANAREFSEL
R/W-Xh					R/W-0h	R/W-0h	R/W-1h

Table 3-187. ANAREFCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	RESERVED	R/W	0h	
8	ANAREF2P5SEL	R/W	0h	Analog reference 2.5V source select. In internal reference mode, this bit selects which voltage the internal reference buffer drives onto the VREFHI pin. The buffer can drive either 1.65V onto the pin, resulting in a reference range of 0 to 3.3V, or the buffer can drive 2.5V onto the pin, resulting in a reference range of 0 to 2.5V. If switching between these two modes, the user must allow adequate time for the external capacitor to charge to the new voltage before using the ADC or buffered DAC. 0 - Internal 1.65V reference mode (3.3V reference range) 1 - Internal 2.5V reference mode (2.5V reference range)
7-3	RESERVED	R/W	Xh	
2	RESERVED	R/W	0h	Reserved
1	RESERVED	R/W	0h	Reserved
0	ANAREFSEL	R/W	1h	Analog reference mode select. This bit selects whether the VREFHI pin uses internal reference mode (the device drives a voltage onto the VREFHI pin) or external reference mode (the system is expected to drive a voltage into the VREFHI pin). 0 - Internal reference mode 1 - External reference mode

82 PERCLKCR Register (Offset = 148Ch) [Reset = 00000000h]

PERCLKCR is shown in Figure 3-92 and described in Table 3-107.

Return to the Summary Table.

PWM Time Base Clock sync

Figure 3-92. PERCLKCR Register

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

Table 3-189. PERCLKCR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R/W	0h	
0	TBCLKSYNC	R/W	0h	PWM Time Base Clock sync: When set PWM time bases of all the PWM modules belonging to the same CPU-Subsystem (as partitioned using their CPUSEL bits) start counting

83 ADC_MMR_OVRD_CTL Register (Offset = 1490h) [Reset = 00000000h]

ADC_MMR_OVRD_CTL is shown in Figure 3-93 and described in Table 3-108.

Return to the Summary Table.

ADC MMR Override control register for DFT: Control ADC enable override

Figure 3-93. ADC_MMR_OVRD_CTL Register

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

Table 3-191. ADC_MMR_OVRD_CTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R/W	0h	
2	RESERVED	R/W	0h	Reserved
1	RESERVED	R/W	0h	Reserved
0	RESERVED	R/W	0h	Reserved

84 ADC_MMR_OVRD_VAL Register (Offset = 1494h) [Reset = 00000000h]

ADC_MMR_OVRD_VAL is shown in Figure 3-94 and described in Table 3-109.

Return to the Summary Table.

ADC MMR Override value register for DFT : Value of ADC enable override

Figure 3-94. ADC_MMR_OVRD_VAL Register

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

Table 3-193. ADC_MMR_OVRD_VAL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R/W	0h	
2	RESERVED	R/W	0h	Reserved
1	RESERVED	R/W	0h	Reserved
0	RESERVED	R/W	0h	Reserved

ADVANCE INFORMATION

85 VREGCONFIGDEBUG Register (Offset = 1498h) [Reset = 00X0X0X0h]

VREGCONFIGDEBUG is shown in Figure 3-95 and described in Table 3-110.

Return to the Summary Table.

VREG Configuration Debug Register

Figure 3-95. VREGCONFIGDEBUG Register

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

Table 3-195. VREGCONFIGDEBUG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	RESERVED	R/W	0h	
27-26	RESERVED	R/W	0h	Reserved
25-24	RESERVED	R/W	0h	Reserved
23-22	RESERVED	R/W	Xh	
21	RESERVED	R/W	0h	Reserved
20	RESERVED	R/W	0h	Reserved
19	RESERVED	R/W	0h	Reserved
18	RESERVED	R/W	0h	Reserved
17	RESERVED	R/W	0h	Reserved
16	RESERVED	R/W	0h	Reserved
15	RESERVED	R/W	Xh	
14-7	RESERVED	R/W	0h	Reserved
6-5	RESERVED	R/W	0h	Reserved
4	RESERVED	R/W	Xh	
3-2	RESERVED	R/W	0h	Reserved
1-0	RESERVED	R/W	0h	Reserved

86 VREGCONFIGDFT Register (Offset = 149Ch) [Reset = 00000000h]

VREGCONFIGDFT is shown in Figure 3-96 and described in Table 3-111.

Return to the Summary Table.

VREG Configuration DFT Register

Figure 3-96. VREGCONFIGDFT Register

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

Table 3-197. VREGCONFIGDFT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-12	RESERVED	R/W	0h	
11-4	RESERVED	R/W	0h	Reserved
3	RESERVED	R/W	0h	Reserved
2	RESERVED	R/W	0h	Reserved
1	RESERVED	R/W	0h	Reserved
0	RESERVED	R/W	0h	Reserved

87 AM13SPAREIREFENSOCLOCK Register (Offset = 14A0h) [Reset = 00000000h]

AM13SPAREIREFENSOCLOCK is shown in Figure 3-97 and described in Table 3-112.

Return to the Summary Table.

AM13 Spare IREFEN SOC LOCK Reg

Figure 3-97. AM13SPAREIREFENSOCLOCK Register

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

Table 3-199. AM13SPAREIREFENSOCLOCK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-6	RESERVED	R/W	0h	
5-0	SPARE	R/W	0h	Spare IREFEN SOC LOCK Register

88 AM13SPARESOCLOCK2 Register (Offset = 14A4h) [Reset = 00000001h]

AM13SPARESOCLOCK2 is shown in Figure 3-98 and described in Table 3-113.

Return to the Summary Table.

AM13 Spare SOC LOCK Reg 2

Figure 3-98. AM13SPARESOCLOCK2 Register

31	30	29	28	27	26	25	24
SPARE							
R/W-0h							
23	22	21	20	19	18	17	16
SPARE							
R/W-0h							
15	14	13	12	11	10	9	8
SPARE							
R/W-0h							
7	6	5	4	3	2	1	0
SPARE							SRAM3_STATIC_MUX_SEL
R/W-0h							R/W-1h

Table 3-201. AM13SPARESOCLOCK2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	SPARE	R/W	0h	Spare SOC LOCK Register 2
0	SRAM3_STATIC_MUX_SEL	R/W	1h	SRAM3 static mux select between CBUS and SBUS 0h = SBUS path is selected 1h = CBUS path is selected

ADVANCE INFORMATION

89 AM13SPARESOCLOCK3 Register (Offset = 14A8h) [Reset = 00000000h]

AM13SPARESOCLOCK3 is shown in Figure 3-99 and described in Table 3-114.

Return to the Summary Table.

AM13 Spare SOC LOCK Reg 3

Figure 3-99. AM13SPARESOCLOCK3 Register

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

Table 3-203. AM13SPARESOCLOCK3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	SPARE	R/W	0h	Spare SOC LOCK Register 3

90 AM13SPARESOCLOCK4 Register (Offset = 14ACh) [Reset = 00000000h]

AM13SPARESOCLOCK4 is shown in Figure 3-100 and described in Table 3-115.

Return to the Summary Table.

AM13 Spare SOC LOCK Reg 4

Figure 3-100. AM13SPARESOCLOCK4 Register

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

Table 3-205. AM13SPARESOCLOCK4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	SPARE	R/W	0h	Spare SOC LOCK Register 4

91 PWREN Register (Offset = 2800h) [Reset = 00XXXXXXh]

PWREN is shown in Figure 3-101 and described in Table 3-116.

Return to the Summary Table.

IP Enable Register

Figure 3-101. PWREN Register

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

Table 3-207. PWREN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	KEY	W	0h	KEY to Allow Enable State Change -- 0x26 26h = 0x26
23-1	RESERVED	R/W	XXXXh	
0	ENABLE	R/W	0h	IP Enable 0h = 0 1h = 1

ADVANCE INFORMATION

92 RSTCTL Register (Offset = 2804h) [Reset = 00XXXXXXh]

RSTCTL is shown in Figure 3-102 and described in Table 3-117.

Return to the Summary Table.

Power Control Register - Write Only Register, Always Read as 0

Figure 3-102. RSTCTL Register

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

Table 3-209. RSTCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	KEY	W	0h	KEY to Allow Reset -- 0xb1 B1h = 0xb1
23-2	RESERVED	W	XXXXh	
1	RESETSTKYCLR	W	0h	Clear the RESET STICKY Bit 1h = 1
0	RESETASSERT	W	0h	Assert Reset to IP Domain. 1h = 1

ADVANCE INFORMATION

93 STAT Register (Offset = 2814h) [Reset = 0000XXXXh]

STAT is shown in Figure 3-103 and described in Table 3-118.

Return to the Summary Table.

IP State Register - Read Only

Figure 3-103. STAT Register

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

Table 3-211. STAT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-17	RESERVED	R	0h	
16	RESETSTKY	R	0h	IP has been Reset 0h = 0 1h = 1
15-0	RESERVED	R	XXXXh	

ADVANCE INFORMATION

94 PWREN Register (Offset = 4800h) [Reset = 00XXXXXXh]

PWREN is shown in Figure 3-104 and described in Table 3-119.

Return to the Summary Table.

IP Enable Register

Figure 3-104. PWREN Register

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

Table 3-213. PWREN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	KEY	W	0h	KEY to Allow Enable State Change -- 0x26 26h = 0x26
23-1	RESERVED	R/W	XXXXh	
0	ENABLE	R/W	0h	IP Enable 0h = 0 1h = 1

ADVANCE INFORMATION

95 RSTCTL Register (Offset = 4804h) [Reset = 00XXXXXXh]

RSTCTL is shown in Figure 3-105 and described in Table 3-120.

Return to the Summary Table.

Power Control Register - Write Only Register, Always Read as 0

Figure 3-105. RSTCTL Register

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

Table 3-215. RSTCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	KEY	W	0h	KEY to Allow Reset -- 0xb1 B1h = 0xb1
23-2	RESERVED	W	XXXXh	
1	RESETSTKYCLR	W	0h	Clear the RESET STICKY Bit 1h = 1
0	RESETASSERT	W	0h	Assert Reset to IP Domain. 1h = 1

ADVANCE INFORMATION

96 STAT Register (Offset = 4814h) [Reset = 0000XXXXh]

STAT is shown in Figure 3-106 and described in Table 3-121.

Return to the Summary Table.

IP State Register - Read Only

Figure 3-106. STAT Register

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

Table 3-217. STAT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-17	RESERVED	R	0h	
16	RESETSTKY	R	0h	IP has been Reset 0h = 0 1h = 1
15-0	RESERVED	R	XXXXh	

ADVANCE INFORMATION

97 PWREN Register (Offset = 000D0800h) [Reset = 00XXXXXXh]

PWREN is shown in Figure 3-107 and described in Table 3-122.

Return to the Summary Table.

IP Enable Register

Figure 3-107. PWREN Register

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

Table 3-219. PWREN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	KEY	W	0h	KEY to Allow Enable State Change -- 0x26 26h = 0x26
23-1	RESERVED	R/W	XXXXh	
0	ENABLE	R/W	0h	IP Enable 0h = 0 1h = 1

ADVANCE INFORMATION

98 RSTCTL Register (Offset = 000D0804h) [Reset = 00XXXXXXh]

RSTCTL is shown in Figure 3-108 and described in Table 3-123.

Return to the Summary Table.

Power Control Register - Write Only Register, Always Read as 0

Figure 3-108. RSTCTL Register

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

Table 3-221. RSTCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	KEY	W	0h	KEY to Allow Reset -- 0xb1 B1h = 0xb1
23-2	RESERVED	W	XXXXh	
1	RESETSTKYCLR	W	0h	Clear the RESET STICKY Bit 1h = 1
0	RESETASSERT	W	0h	Assert Reset to IP Domain. 1h = 1

ADVANCE INFORMATION

99 STAT Register (Offset = 000D0814h) [Reset = 0000XXXXh]

STAT is shown in Figure 3-109 and described in Table 3-124.

Return to the Summary Table.

IP State Register - Read Only

Figure 3-109. STAT Register

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

Table 3-223. STAT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-17	RESERVED	R	0h	
16	RESETSTKY	R	0h	IP has been Reset 0h = 0 1h = 1
15-0	RESERVED	R	XXXXh	

ADVANCE INFORMATION

100 PWREN Register (Offset = 000E8800h) [Reset = 00XXXXXXh]

PWREN is shown in Figure 3-110 and described in Table 3-125.

Return to the Summary Table.

IP Enable Register

Figure 3-110. PWREN Register

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

Table 3-225. PWREN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	KEY	W	0h	KEY to Allow Enable State Change -- 0x26 26h = 0x26
23-1	RESERVED	R/W	XXXXh	
0	ENABLE	R/W	0h	IP Enable 0h = 0 1h = 1

ADVANCE INFORMATION

101 RSTCTL Register (Offset = 000E8804h) [Reset = 00XXXXXXh]

RSTCTL is shown in Figure 3-111 and described in Table 3-126.

Return to the Summary Table.

Power Control Register - Write Only Register, Always Read as 0

Figure 3-111. RSTCTL Register

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

Table 3-227. RSTCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	KEY	W	0h	KEY to Allow Reset -- 0xb1 B1h = 0xb1
23-2	RESERVED	W	XXXXh	
1	RESETSTKYCLR	W	0h	Clear the RESET STICKY Bit 1h = 1
0	RESETASSERT	W	0h	Assert Reset to IP Domain. 1h = 1

102 STAT Register (Offset = 000E8814h) [Reset = 0000XXXXh]

STAT is shown in Figure 3-112 and described in Table 3-127.

Return to the Summary Table.

IP State Register - Read Only

Figure 3-112. STAT Register

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

Table 3-229. STAT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-17	RESERVED	R	0h	
16	RESETSTKY	R	0h	IP has been Reset 0h = 0 1h = 1
15-0	RESERVED	R	XXXXh	

ADVANCE INFORMATION

103 PWREN Register (Offset = 000F0800h) [Reset = 00XXXXXXh]

PWREN is shown in Figure 3-113 and described in Table 3-128.

Return to the Summary Table.

IP Enable Register

Figure 3-113. PWREN Register

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

Table 3-231. PWREN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	KEY	W	0h	KEY to Allow Enable State Change -- 0x26 26h = 0x26
23-1	RESERVED	R/W	XXXXh	
0	ENABLE	R/W	0h	IP Enable 0h = 0 1h = 1

ADVANCE INFORMATION

104 RSTCTL Register (Offset = 000F0804h) [Reset = 00XXXXXXh]

RSTCTL is shown in Figure 3-114 and described in Table 3-129.

Return to the Summary Table.

Power Control Register - Write Only Register, Always Read as 0

Figure 3-114. RSTCTL Register

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

Table 3-233. RSTCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	KEY	W	0h	KEY to Allow Reset -- 0xb1 B1h = 0xb1
23-2	RESERVED	W	XXXXh	
1	RESETSTKYCLR	W	0h	Clear the RESET STICKY Bit 1h = 1
0	RESETASSERT	W	0h	Assert Reset to IP Domain. 1h = 1

ADVANCE INFORMATION

105 STAT Register (Offset = 000F0814h) [Reset = 0000XXXXh]

STAT is shown in Figure 3-115 and described in Table 3-130.

Return to the Summary Table.

IP State Register - Read Only

Figure 3-115. STAT Register

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

Table 3-235. STAT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-17	RESERVED	R	0h	
16	RESETSTKY	R	0h	IP has been Reset 0h = 0 1h = 1
15-0	RESERVED	R	XXXXh	

ADVANCE INFORMATION

106 PWREN Register (Offset = 000F2800h) [Reset = 00XXXXXXh]

PWREN is shown in Figure 3-116 and described in Table 3-131.

Return to the Summary Table.

IP Enable Register

Figure 3-116. PWREN Register

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

Table 3-237. PWREN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	KEY	W	0h	KEY to Allow Enable State Change -- 0x26 26h = 0x26
23-1	RESERVED	R/W	XXXXh	
0	ENABLE	R/W	0h	IP Enable 0h = 0 1h = 1

ADVANCE INFORMATION

107 RSTCTL Register (Offset = 000F2804h) [Reset = 00XXXXXXh]

RSTCTL is shown in Figure 3-117 and described in Table 3-132.

Return to the Summary Table.

Power Control Register - Write Only Register, Always Read as 0

Figure 3-117. RSTCTL Register

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

Table 3-239. RSTCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	KEY	W	0h	KEY to Allow Reset -- 0xb1 B1h = 0xb1
23-2	RESERVED	W	XXXXh	
1	RESETSTKYCLR	W	0h	Clear the RESET STICKY Bit 1h = 1
0	RESETASSERT	W	0h	Assert Reset to IP Domain. 1h = 1

108 STAT Register (Offset = 000F2814h) [Reset = 0000XXXXh]

STAT is shown in Figure 3-118 and described in Table 3-133.

Return to the Summary Table.

IP State Register - Read Only

Figure 3-118. STAT Register

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

Table 3-241. STAT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-17	RESERVED	R	0h	
16	RESETSTKY	R	0h	IP has been Reset 0h = 0 1h = 1
15-0	RESERVED	R	XXXXh	

ADVANCE INFORMATION

109 PWREN Register (Offset = 000F4800h) [Reset = 00XXXXXXh]

PWREN is shown in Figure 3-119 and described in Table 3-134.

Return to the Summary Table.

IP Enable Register

Figure 3-119. PWREN Register

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

Table 3-243. PWREN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	KEY	W	0h	KEY to Allow Enable State Change -- 0x26 26h = 0x26
23-1	RESERVED	R/W	XXXXh	
0	ENABLE	R/W	0h	IP Enable 0h = 0 1h = 1

ADVANCE INFORMATION

110 RSTCTL Register (Offset = 000F4804h) [Reset = 00XXXXXXh]

RSTCTL is shown in Figure 3-120 and described in Table 3-135.

Return to the Summary Table.

Power Control Register - Write Only Register, Always Read as 0

Figure 3-120. RSTCTL Register

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

Table 3-245. RSTCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	KEY	W	0h	KEY to Allow Reset -- 0xb1 B1h = 0xb1
23-2	RESERVED	W	XXXXh	
1	RESETSTKYCLR	W	0h	Clear the RESET STICKY Bit 1h = 1
0	RESETASSERT	W	0h	Assert Reset to IP Domain. 1h = 1

ADVANCE INFORMATION

111 STAT Register (Offset = 000F4814h) [Reset = 0000XXXXh]

STAT is shown in Figure 3-121 and described in Table 3-136.

Return to the Summary Table.

IP State Register - Read Only

Figure 3-121. STAT Register

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

Table 3-247. STAT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-17	RESERVED	R	0h	
16	RESETSTKY	R	0h	IP has been Reset 0h = 0 1h = 1
15-0	RESERVED	R	XXXXh	

ADVANCE INFORMATION

112 PWREN Register (Offset = 000F6800h) [Reset = 00XXXXXXh]

PWREN is shown in Figure 3-122 and described in Table 3-137.

Return to the Summary Table.

IP Enable Register

Figure 3-122. PWREN Register

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

Table 3-249. PWREN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	KEY	W	0h	KEY to Allow Enable State Change -- 0x26 26h = 0x26
23-1	RESERVED	R/W	XXXXh	
0	ENABLE	R/W	0h	IP Enable 0h = 0 1h = 1

ADVANCE INFORMATION

113 RSTCTL Register (Offset = 000F6804h) [Reset = 00XXXXXXh]

RSTCTL is shown in Figure 3-123 and described in Table 3-138.

Return to the Summary Table.

Power Control Register - Write Only Register, Always Read as 0

Figure 3-123. RSTCTL Register

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

Table 3-251. RSTCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	KEY	W	0h	KEY to Allow Reset -- 0xb1 B1h = 0xb1
23-2	RESERVED	W	XXXXh	
1	RESETSTKYCLR	W	0h	Clear the RESET STICKY Bit 1h = 1
0	RESETASSERT	W	0h	Assert Reset to IP Domain. 1h = 1

ADVANCE INFORMATION

114 STAT Register (Offset = 000F6814h) [Reset = 0000XXXXh]

STAT is shown in Figure 3-124 and described in Table 3-139.

Return to the Summary Table.

IP State Register - Read Only

Figure 3-124. STAT Register

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

Table 3-253. STAT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-17	RESERVED	R	0h	
16	RESETSTKY	R	0h	IP has been Reset 0h = 0 1h = 1
15-0	RESERVED	R	XXXXh	

ADVANCE INFORMATION

115 PWREN Register (Offset = 00116800h) [Reset = 00XXXXXXh]

PWREN is shown in Figure 3-125 and described in Table 3-140.

Return to the Summary Table.

IP Enable Register

Figure 3-125. PWREN Register

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

Table 3-255. PWREN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	KEY	W	0h	KEY to Allow Enable State Change -- 0x26 26h = 0x26
23-1	RESERVED	R/W	XXXXh	
0	ENABLE	R/W	0h	IP Enable 0h = 0 1h = 1

ADVANCE INFORMATION

116 RSTCTL Register (Offset = 00116804h) [Reset = 00XXXXXXh]

RSTCTL is shown in Figure 3-126 and described in Table 3-141.

Return to the Summary Table.

Power Control Register - Write Only Register, Always Read as 0

Figure 3-126. RSTCTL Register

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

Table 3-257. RSTCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	KEY	W	0h	KEY to Allow Reset -- 0xb1 B1h = 0xb1
23-2	RESERVED	W	XXXXh	
1	RESETSTKYCLR	W	0h	Clear the RESET STICKY Bit 1h = 1
0	RESETASSERT	W	0h	Assert Reset to IP Domain. 1h = 1

ADVANCE INFORMATION

117 STAT Register (Offset = 00116814h) [Reset = 0000XXXXh]

STAT is shown in Figure 3-127 and described in Table 3-142.

Return to the Summary Table.

IP State Register - Read Only

Figure 3-127. STAT Register

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

Table 3-259. STAT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-17	RESERVED	R	0h	
16	RESETSTKY	R	0h	IP has been Reset 0h = 0 1h = 1
15-0	RESERVED	R	XXXXh	

118 PWREN Register (Offset = 00180800h) [Reset = 00XXXXXXh]

PWREN is shown in Figure 3-128 and described in Table 3-143.

Return to the Summary Table.

IP Enable Register

Figure 3-128. PWREN Register

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

Table 3-261. PWREN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	KEY	W	0h	KEY to Allow Enable State Change -- 0x26 26h = 0x26
23-1	RESERVED	R/W	XXXXh	
0	ENABLE	R/W	0h	IP Enable 0h = 0 1h = 1

ADVANCE INFORMATION

119 RSTCTL Register (Offset = 00180804h) [Reset = 00XXXXXXh]

RSTCTL is shown in Figure 3-129 and described in Table 3-144.

Return to the Summary Table.

Power Control Register - Write Only Register, Always Read as 0

Figure 3-129. RSTCTL Register

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

Table 3-263. RSTCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	KEY	W	0h	KEY to Allow Reset -- 0xb1 B1h = 0xb1
23-2	RESERVED	W	XXXXh	
1	RESETSTKYCLR	W	0h	Clear the RESET STICKY Bit 1h = 1
0	RESETASSERT	W	0h	Assert Reset to IP Domain. 1h = 1

120 STAT Register (Offset = 00180814h) [Reset = 0000XXXXh]

STAT is shown in Figure 3-130 and described in Table 3-145.

Return to the Summary Table.

IP State Register - Read Only

Figure 3-130. STAT Register

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

Table 3-265. STAT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-17	RESERVED	R	0h	
16	RESETSTKY	R	0h	IP has been Reset 0h = 0 1h = 1
15-0	RESERVED	R	XXXXh	

ADVANCE INFORMATION

121 PWREN Register (Offset = 00188800h) [Reset = 00XXXXXXh]

PWREN is shown in Figure 3-131 and described in Table 3-146.

Return to the Summary Table.

IP Enable Register

Figure 3-131. PWREN Register

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

Table 3-267. PWREN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	KEY	W	0h	KEY to Allow Enable State Change -- 0x26 26h = 0x26
23-1	RESERVED	R/W	XXXXh	
0	ENABLE	R/W	0h	IP Enable 0h = 0 1h = 1

ADVANCE INFORMATION

122 RSTCTL Register (Offset = 00188804h) [Reset = 00XXXXXXh]

RSTCTL is shown in Figure 3-132 and described in Table 3-147.

Return to the Summary Table.

Power Control Register - Write Only Register, Always Read as 0

Figure 3-132. RSTCTL Register

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

Table 3-269. RSTCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	KEY	W	0h	KEY to Allow Reset -- 0xb1 B1h = 0xb1
23-2	RESERVED	W	XXXXh	
1	RESETSTKYCLR	W	0h	Clear the RESET STICKY Bit 1h = 1
0	RESETASSERT	W	0h	Assert Reset to IP Domain. 1h = 1

123 STAT Register (Offset = 00188814h) [Reset = 0000XXXXh]

STAT is shown in Figure 3-133 and described in Table 3-148.

Return to the Summary Table.

IP State Register - Read Only

Figure 3-133. STAT Register

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

Table 3-271. STAT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-17	RESERVED	R	0h	
16	RESETSTKY	R	0h	IP has been Reset 0h = 0 1h = 1
15-0	RESERVED	R	XXXXh	

ADVANCE INFORMATION

124 PWREN Register (Offset = 001B0800h) [Reset = 00XXXXXXh]

PWREN is shown in Figure 3-134 and described in Table 3-149.

Return to the Summary Table.

IP Enable Register

Figure 3-134. PWREN Register

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

Table 3-273. PWREN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	KEY	W	0h	KEY to Allow Enable State Change -- 0x26 26h = 0x26
23-1	RESERVED	R/W	XXXXh	
0	ENABLE	R/W	0h	IP Enable 0h = 0 1h = 1

ADVANCE INFORMATION

125 RSTCTL Register (Offset = 001B0804h) [Reset = 00XXXXXXh]

RSTCTL is shown in Figure 3-135 and described in Table 3-150.

Return to the Summary Table.

Power Control Register - Write Only Register, Always Read as 0

Figure 3-135. RSTCTL Register

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

Table 3-275. RSTCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	KEY	W	0h	KEY to Allow Reset -- 0xb1 B1h = 0xb1
23-2	RESERVED	W	XXXXh	
1	RESETSTKYCLR	W	0h	Clear the RESET STICKY Bit 1h = 1
0	RESETASSERT	W	0h	Assert Reset to IP Domain. 1h = 1

126 STAT Register (Offset = 001B0814h) [Reset = 0000XXXXh]

STAT is shown in Figure 3-136 and described in Table 3-151.

Return to the Summary Table.

IP State Register - Read Only

Figure 3-136. STAT Register

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

Table 3-277. STAT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-17	RESERVED	R	0h	
16	RESETSTKY	R	0h	IP has been Reset 0h = 0 1h = 1
15-0	RESERVED	R	XXXXh	

127 PWREN Register (Offset = 001B2800h) [Reset = 00XXXXXXh]

PWREN is shown in Figure 3-137 and described in Table 3-152.

Return to the Summary Table.

IP Enable Register

Figure 3-137. PWREN Register

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

Table 3-279. PWREN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	KEY	W	0h	KEY to Allow Enable State Change -- 0x26 26h = 0x26
23-1	RESERVED	R/W	XXXXh	
0	ENABLE	R/W	1h	IP Enable 0h = 0 1h = 1

ADVANCE INFORMATION

128 RSTCTL Register (Offset = 001B2804h) [Reset = 00XXXXXXh]

RSTCTL is shown in Figure 3-138 and described in Table 3-153.

Return to the Summary Table.

Power Control Register - Write Only Register, Always Read as 0

Figure 3-138. RSTCTL Register

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

Table 3-281. RSTCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	KEY	W	0h	KEY to Allow Reset -- 0xb1 B1h = 0xb1
23-2	RESERVED	W	XXXXh	
1	RESETSTKYCLR	W	0h	Clear the RESET STICKY Bit 1h = 1
0	RESETASSERT	W	0h	Assert Reset to IP Domain. 1h = 1

129 STAT Register (Offset = 001B2814h) [Reset = 0000XXXXh]

STAT is shown in Figure 3-139 and described in Table 3-154.

Return to the Summary Table.

IP State Register - Read Only

Figure 3-139. STAT Register

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

Table 3-283. STAT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-17	RESERVED	R	0h	
16	RESETSTKY	R	0h	IP has been Reset 0h = 0 1h = 1
15-0	RESERVED	R	XXXXh	

ADVANCE INFORMATION

130 PWREN Register (Offset = 00630800h) [Reset = 00XXXXXXh]

PWREN is shown in Figure 3-140 and described in Table 3-155.

Return to the Summary Table.

IP Enable Register

Figure 3-140. PWREN Register

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

Table 3-285. PWREN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	KEY	W	0h	KEY to Allow Enable State Change -- 0x26 26h = 0x26
23-1	RESERVED	R/W	XXXXh	
0	ENABLE	R/W	0h	IP Enable 0h = 0 1h = 1

ADVANCE INFORMATION

131 RSTCTL Register (Offset = 00630804h) [Reset = 00XXXXXXh]

RSTCTL is shown in Figure 3-141 and described in Table 3-156.

Return to the Summary Table.

Power Control Register - Write Only Register, Always Read as 0

Figure 3-141. RSTCTL Register

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

Table 3-287. RSTCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	KEY	W	0h	KEY to Allow Reset -- 0xb1 B1h = 0xb1
23-2	RESERVED	W	XXXXh	
1	RESETSTKYCLR	W	0h	Clear the RESET STICKY Bit 1h = 1
0	RESETASSERT	W	0h	Assert Reset to IP Domain. 1h = 1

132 STAT Register (Offset = 00630814h) [Reset = 0000XXXXh]

STAT is shown in Figure 3-142 and described in Table 3-157.

Return to the Summary Table.

IP State Register - Read Only

Figure 3-142. STAT Register

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

Table 3-289. STAT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-17	RESERVED	R	0h	
16	RESETSTKY	R	0h	IP has been Reset 0h = 0 1h = 1
15-0	RESERVED	R	XXXXh	

ADVANCE INFORMATION

133 PWREN Register (Offset = 00632800h) [Reset = 00XXXXXXh]

PWREN is shown in Figure 3-143 and described in Table 3-158.

Return to the Summary Table.

IP Enable Register

Figure 3-143. PWREN Register

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

Table 3-291. PWREN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	KEY	W	0h	KEY to Allow Enable State Change -- 0x26 26h = 0x26
23-1	RESERVED	R/W	XXXXh	
0	ENABLE	R/W	0h	IP Enable 0h = 0 1h = 1

ADVANCE INFORMATION

134 RSTCTL Register (Offset = 00632804h) [Reset = 00XXXXXXh]

RSTCTL is shown in Figure 3-144 and described in Table 3-159.

Return to the Summary Table.

Power Control Register - Write Only Register, Always Read as 0

Figure 3-144. RSTCTL Register

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

Table 3-293. RSTCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	KEY	W	0h	KEY to Allow Reset -- 0xb1 B1h = 0xb1
23-2	RESERVED	W	XXXXh	
1	RESETSTKYCLR	W	0h	Clear the RESET STICKY Bit 1h = 1
0	RESETASSERT	W	0h	Assert Reset to IP Domain. 1h = 1

135 STAT Register (Offset = 00632814h) [Reset = 0000XXXXh]

STAT is shown in Figure 3-145 and described in Table 3-160.

Return to the Summary Table.

IP State Register - Read Only

Figure 3-145. STAT Register

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

Table 3-295. STAT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-17	RESERVED	R	0h	
16	RESETSTKY	R	0h	IP has been Reset 0h = 0 1h = 1
15-0	RESERVED	R	XXXXh	

ADVANCE INFORMATION

136 PWREN Register (Offset = 00634800h) [Reset = 00XXXXXXh]

PWREN is shown in Figure 3-146 and described in Table 3-161.

Return to the Summary Table.

IP Enable Register

Figure 3-146. PWREN Register

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

Table 3-297. PWREN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	KEY	W	0h	KEY to Allow Enable State Change -- 0x26 26h = 0x26
23-1	RESERVED	R/W	XXXXh	
0	ENABLE	R/W	0h	IP Enable 0h = 0 1h = 1

ADVANCE INFORMATION

137 RSTCTL Register (Offset = 00634804h) [Reset = 00XXXXXXh]

RSTCTL is shown in Figure 3-147 and described in Table 3-162.

Return to the Summary Table.

Power Control Register - Write Only Register, Always Read as 0

Figure 3-147. RSTCTL Register

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

Table 3-299. RSTCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	KEY	W	0h	KEY to Allow Reset -- 0xb1 B1h = 0xb1
23-2	RESERVED	W	XXXXh	
1	RESETSTKYCLR	W	0h	Clear the RESET STICKY Bit 1h = 1
0	RESETASSERT	W	0h	Assert Reset to IP Domain. 1h = 1

138 STAT Register (Offset = 00634814h) [Reset = 0000XXXXh]

STAT is shown in Figure 3-148 and described in Table 3-163.

Return to the Summary Table.

IP State Register - Read Only

Figure 3-148. STAT Register

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

Table 3-301. STAT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-17	RESERVED	R	0h	
16	RESETSTKY	R	0h	IP has been Reset 0h = 0 1h = 1
15-0	RESERVED	R	XXXXh	

ADVANCE INFORMATION

139 PWREN Register (Offset = 00670800h) [Reset = 00XXXXXXh]

PWREN is shown in Figure 3-149 and described in Table 3-164.

Return to the Summary Table.

IP Enable Register

Figure 3-149. PWREN Register

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

Table 3-303. PWREN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	KEY	W	0h	KEY to Allow Enable State Change -- 0x26 26h = 0x26
23-1	RESERVED	R/W	XXXXh	
0	ENABLE	R/W	0h	IP Enable 0h = 0 1h = 1

ADVANCE INFORMATION

140 RSTCTL Register (Offset = 00670804h) [Reset = 00XXXXXXh]

RSTCTL is shown in Figure 3-150 and described in Table 3-165.

Return to the Summary Table.

Power Control Register - Write Only Register, Always Read as 0

Figure 3-150. RSTCTL Register

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

Table 3-305. RSTCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	KEY	W	0h	KEY to Allow Reset -- 0xb1 B1h = 0xb1
23-2	RESERVED	W	XXXXh	
1	RESETSTKYCLR	W	0h	Clear the RESET STICKY Bit 1h = 1
0	RESETASSERT	W	0h	Assert Reset to IP Domain. 1h = 1

141 STAT Register (Offset = 00670814h) [Reset = 0000XXXXh]

STAT is shown in Figure 3-151 and described in Table 3-166.

Return to the Summary Table.

IP State Register - Read Only

Figure 3-151. STAT Register

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

Table 3-307. STAT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-17	RESERVED	R	0h	
16	RESETSTKY	R	0h	IP has been Reset 0h = 0 1h = 1
15-0	RESERVED	R	XXXXh	

142 PWREN Register (Offset = 00672800h) [Reset = 00XXXXXXh]

PWREN is shown in Figure 3-152 and described in Table 3-167.

Return to the Summary Table.

IP Enable Register

Figure 3-152. PWREN Register

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

Table 3-309. PWREN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	KEY	W	0h	KEY to Allow Enable State Change -- 0x26 26h = 0x26
23-1	RESERVED	R/W	XXXXh	
0	ENABLE	R/W	0h	IP Enable 0h = 0 1h = 1

ADVANCE INFORMATION

143 RSTCTL Register (Offset = 00672804h) [Reset = 00XXXXXXh]

RSTCTL is shown in Figure 3-153 and described in Table 3-168.

Return to the Summary Table.

Power Control Register - Write Only Register, Always Read as 0

Figure 3-153. RSTCTL Register

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

Table 3-311. RSTCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	KEY	W	0h	KEY to Allow Reset -- 0xb1 B1h = 0xb1
23-2	RESERVED	W	XXXXh	
1	RESETSTKYCLR	W	0h	Clear the RESET STICKY Bit 1h = 1
0	RESETASSERT	W	0h	Assert Reset to IP Domain. 1h = 1

144 STAT Register (Offset = 00672814h) [Reset = 0000XXXXh]

STAT is shown in Figure 3-154 and described in Table 3-169.

Return to the Summary Table.

IP State Register - Read Only

Figure 3-154. STAT Register

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

Table 3-313. STAT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-17	RESERVED	R	0h	
16	RESETSTKY	R	0h	IP has been Reset 0h = 0 1h = 1
15-0	RESERVED	R	XXXXh	

ADVANCE INFORMATION

145 PWREN Register (Offset = 00674800h) [Reset = 00XXXXXXh]

PWREN is shown in Figure 3-155 and described in Table 3-170.

Return to the Summary Table.

IP Enable Register

Figure 3-155. PWREN Register

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

Table 3-315. PWREN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	KEY	W	0h	KEY to Allow Enable State Change -- 0x26 26h = 0x26
23-1	RESERVED	R/W	XXXXh	
0	ENABLE	R/W	0h	IP Enable 0h = 0 1h = 1

ADVANCE INFORMATION

146 RSTCTL Register (Offset = 00674804h) [Reset = 00XXXXXXh]

RSTCTL is shown in Figure 3-156 and described in Table 3-171.

Return to the Summary Table.

Power Control Register - Write Only Register, Always Read as 0

Figure 3-156. RSTCTL Register

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

Table 3-317. RSTCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	KEY	W	0h	KEY to Allow Reset -- 0xb1 B1h = 0xb1
23-2	RESERVED	W	XXXXh	
1	RESETSTKYCLR	W	0h	Clear the RESET STICKY Bit 1h = 1
0	RESETASSERT	W	0h	Assert Reset to IP Domain. 1h = 1

147 STAT Register (Offset = 00674814h) [Reset = 0000XXXXh]

STAT is shown in Figure 3-157 and described in Table 3-172.

Return to the Summary Table.

IP State Register - Read Only

Figure 3-157. STAT Register

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

Table 3-319. STAT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-17	RESERVED	R	0h	
16	RESETSTKY	R	0h	IP has been Reset 0h = 0 1h = 1
15-0	RESERVED	R	XXXXh	

Chapter 4

Central Processing Unit (CPU)

The CPU subsystem (CPUSS) includes the Arm Cortex-M33 processor and the interrupt logic.

ADVANCE INFORMATION

4.1 Overview

The AM13E230x CPU subsystem (CPUSS) contains the central processing unit (CPU) along with associated supporting logic and read-only memory (ROM). The functional blocks that comprise the CPUSS include:

- The Arm Cortex-M33 32-bit CPU and internal peripherals
- The CPU bus splitter and router
- The interrupt management logic and DEEPSLEEP entry and exit logic
- The CPU debug interface to the Debug Subsystem (DEBUGSS)
- The Read-Only Memory (ROM) used for the BCR and BSL

The top level architecture of the CPUSS is shown in Figure 4-1.

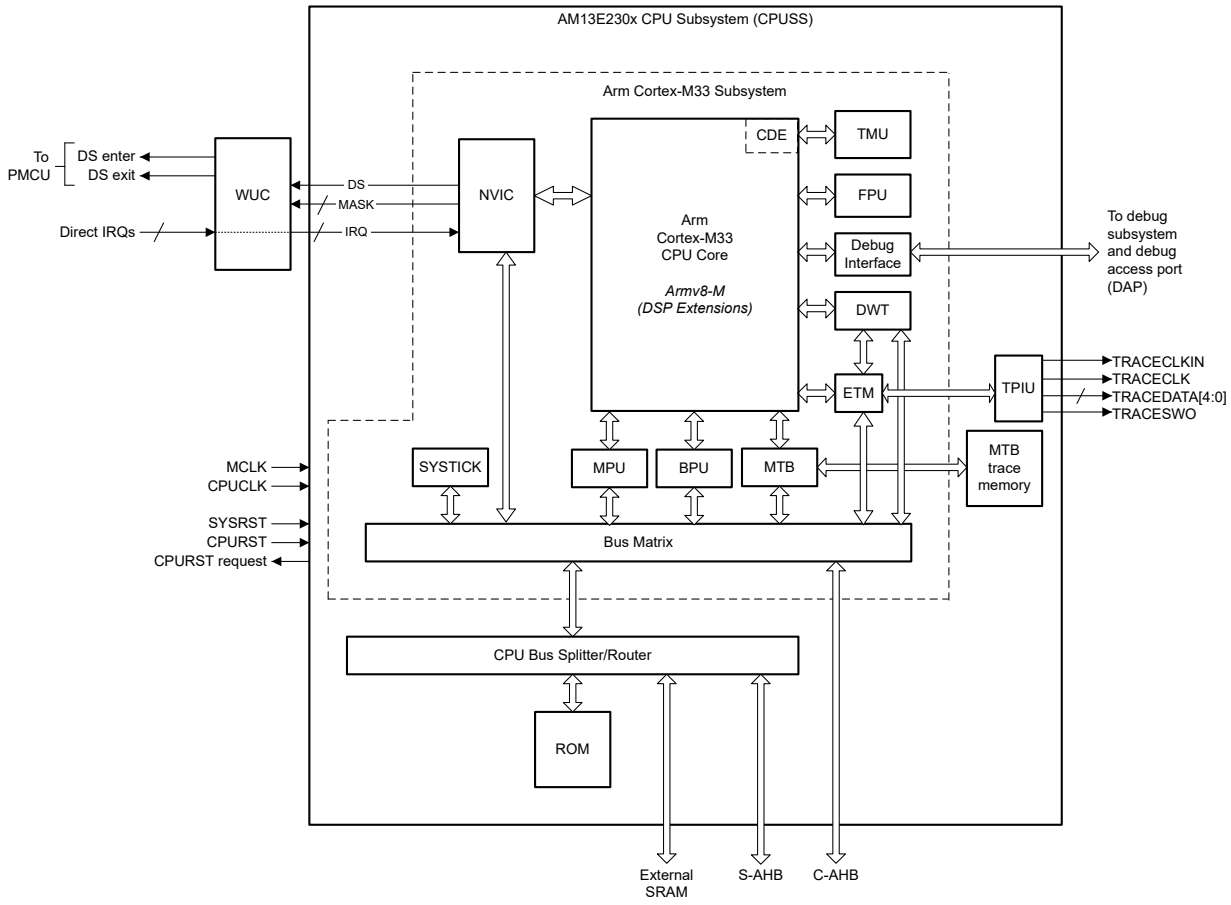


Figure 4-1. AM13E230x CPUSS Top Level Diagram

ADVANCE INFORMATION

4.2 CPU

The CPU subsystem (CPUSS) includes the Arm Cortex-M33 processor and the interrupt logic.

4.2.1 Arm Cortex-M33 CPU

The CPUSS contains an energy-efficient Arm Cortex-M33 CPU implementing the Armv8-M instruction set architecture with support for CPU clock speeds up to 200MHz. The Cortex-M33 is a Harvard architecture style 32-bit processor with a 3-stage ultra-low power pipeline.

The Cortex-M33 implementation on AM13E230x devices has the following features:

- Up to 200MHz execution frequency
- Little-endian (least significant byte at lowest byte address location)
- Support for 32-bit word instruction fetches
- Single cycle 32×32 multiply instruction
- Integrated Memory Protection Unit (MPU)
- Integrated Floating Point Unit (FPU)
- Custom Datapath Extension Trigonometric Math Unit (CDE TMU)
- DSP extensions
- User and privileged execution modes
- Integrated 24-bit system tick timer (SYSTICK)
- Eight hardware breakpoints and four hardware watchpoints for debug
- Micro Trace Buffer (MTB)
- Embedded Trace Macrocell (ETM) with timestamp counter
- Reset-all-registers support
- Vector table offset support

The Cortex-M33 architecture enables excellent code density, deterministic interrupt handling, and upwards compatibility with other processor architectures in the Arm Cortex-M family.

A general overview of the Arm Cortex-M33 is given in this section to provide a basic understanding of the features of the processor. For detailed information on developing with the Arm Cortex-M33 processor, refer to the *Arm Cortex-M33 Devices Generic User's Guide*.

4.2.2 CPU Register File

The Arm Cortex-M33 processor instructions operate on registers in the CPU register file. The processor contains a register file consisting of 16 32-bit wide standard registers and 5 special registers as shown in Figure 4-2.

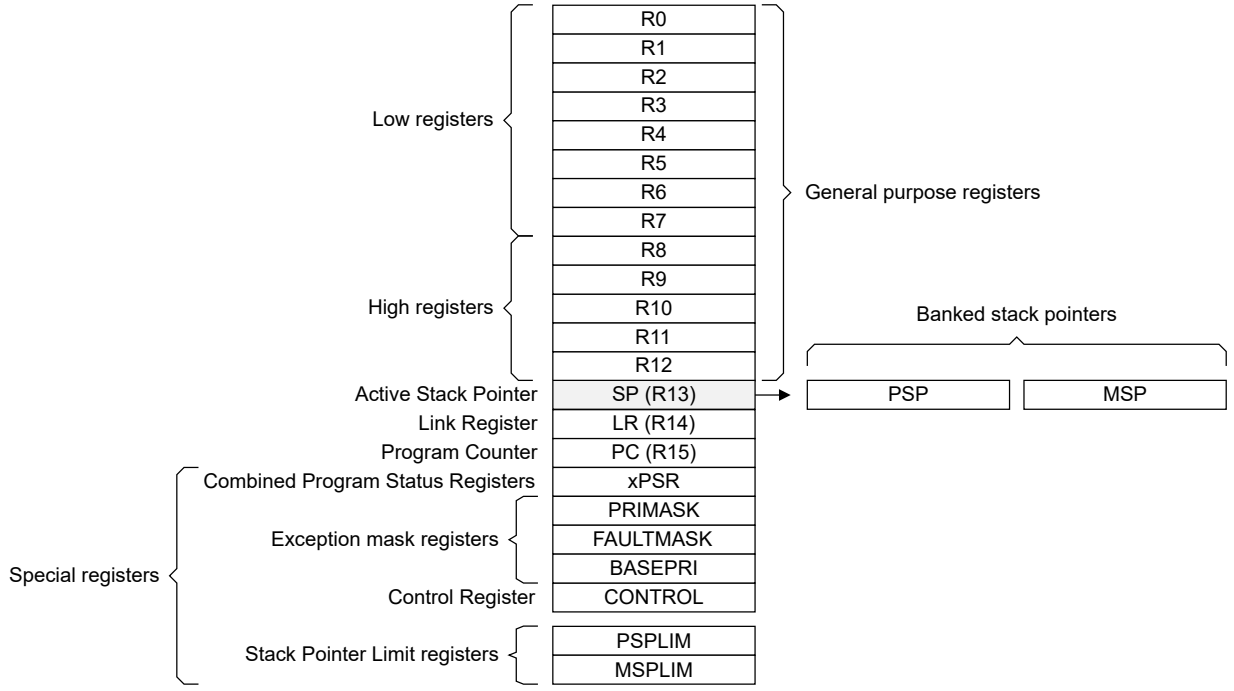


Figure 4-2. CPU Core Registers

General Purpose Registers (R0-R12)

The processor provides 13 general purpose registers, R0-R12, for operating on data. Registers R0 to R7 (low registers) are accessible by all instructions which specify a general purpose register. Registers R8 to R12 (high registers) are not accessible by 16-bit instructions but are accessible by any 32-bit instructions which specify a general purpose register.

Stack Pointer Register (R13)

The stack pointer is contained in R13, and can contain the Main Stack Pointer (MSP) or the Process Stack Pointer (PSP). When the processor is running in handler mode, the MSP is always used. When the processor is running in thread mode, the MSP or PSP can be used depending on the configuration of the SPSEL bit in the CONTROL register.

After a CPURST, the processor automatically and unconditionally fetches the default stack pointer from the first address of main flash (0x0000.0000) as the main stack pointer (MSP).

The MSPLIM and PSPLIM are registers that define the lower limit for the corresponding stack. The processor raises an exception on most instructions that attempt to update the stack pointer below the defined limit.

Link Register (R14)

The Link Register (LR) is register R14. It stores the return information for subroutines, function calls, and exceptions. On reset, the processor sets the LR value to 0xFFFFFFFF. It is accessible in privileged and unprivileged mode.

Program Counter Register (R15)

The program counter register (R15) contains the address of the next instruction to be executed. The PC is accessible in privileged and unprivileged mode.

After a CPURST, the processor automatically and unconditionally fetches the default PC from the second word of main flash (0x0000.0004).

Special Registers

Special registers include the program status register (xPSR), the interrupt mask register (PRIMASK), fault mask register (FAULTMASK), base priority mask register (BASEPRI), and the control register (CONTROL). Special registers are typically accessed by using the CPS, MRS, and MSR system instructions.

- **Program Status Register (xPSR):** The xPSR is a combination of the application status (APSR), interrupt status (IPSR), and execution status (EPSR) registers. Application software can access the PSR with MRS and MSR instructions, accessing either the complete PSR or a combination of one or more registers, with some restrictions. The PSR registers can be accessed with MRS and MSR instructions using the mnemonics given in #unique_143/unique_143_Connect_42_TABLE_I1S_XWQ_GPB .
  - The **Application Program Status Register (APSR)** contains the N, Z, C, V, Q, and GE flags which are used by the processor to evaluate conditional branch instructions. These bits are located in BIT31, BIT30, BIT29, BIT28, BIT27, and BIT16 to BIT19 of the PSR, respectively.
  - The **Interrupt Program Status Register (IPSR)** reports the exception number of the current interrupt service routine when in handler mode. In thread mode, this register reads as all zero. Attempts to write to the IPSR are ignored. The exception number field is presented in from BIT0 to BIT5 of the PSR.
  - The **Execution Program Status Register (EPSR)** contains the Thumb state bit and the execution state bits for the If-Then (IT) instruction, and Interruptible-Continuable Instruction (ICI) field for an interrupted load multiple or store multiple instruction. Attempts to read the EPSR directly through application software using the `MRS` instruction always return zero. Attempts to write the EPSR using the `MSR` instruction in application software are ignored.
- **Interrupt Mask Register (PRIMASK):** BIT0 of the PRIMASK register (PM) can be used to mask all interrupts to the processor which have configurable priority (see Section 4.3). This can be thought of as a global peripheral interrupt mask control. The processor ignores unprivileged writes to PRIMASK. Clearing PM to 0 enables interrupts. Setting PM to 1 disables interrupts. The CPS instruction can be used to change the PM bit value in the PRIMASK register.
- **Fault Mask Register (FAULTMASK):** The Fault mask register prevents activation of all exceptions with configurable priority and some exceptions with fixed priority. The priority it is boosted to changes based on the value of BIT14 (PRIS) and BIT13 (BFHFNMINS) bit of the AIRCR register in the SCB.
- **Base Priority Mask Register (BASEPRI)** The base priority mask register can be used to change the priority level that is required for exception preemption. BIT0 to BIT7 of BASEPRI are used for configure what the boosted priority will be. When the current execution priority is boosted to a particular value, all exceptions with a lower priority are masked. Writing 0 to BASEPRI disables base priority boosting.
- **Control Register (CONTROL):** The CONTROL register can be used to define whether the code executing in thread mode is privileged or unprivileged by clearing or setting the nPRIV bit (BIT0), respectively. It can also be used to select the active stack pointer used in R13 as either the Main Stack Pointer (MSP) or Process Stack Pointer (PSP) by clearing or setting the SPSEL bit (BIT1), respectively. The FPCA bit (BIT2) of the CONTROL register also indicates whether the FPU state is active. A CPURST clears the CONTROL register to zero. The processor ignores unprivileged writes to the CONTROL register. The SPSEL stack pointer selection bit is updated by the processor automatically when entering and returning from exceptions.

Note

For more information on the core registers, please refer to the Arm Cortex-M33 Devices Generic User Guide.

Note

Software must implement an ISB barrier instruction after writing to CONTROL to ensure that any changes take effect before the next application instruction is executed by the processor.

Table 4-1. Program Status Register (xPSR) Access Mnemonics

Mnemonic	Subregisters Included
xPSR	APSR, IPSR, EPSR
APSR	APSR
IPSR	IPSR
EPSR	EPSR
IAPSR	IPSR and APSR
EAPSR	EPSR and APSR
IEPSR	IPSR and EPSR

4.2.3 Stack Behavior

The Arm Cortex-M33 processor implements a full descending stack protocol. The stack pointer register (SP) always indicates the location of the last stacked data. When new data is added to the call stack, the SP value is decremented and the data is written to the location indicated by the SP after being decremented.

The Arm Cortex-M33 supports managing two independent stacks with two pointers: the main stack (MSP) and the process stack (PSP).

4.2.4 Execution Modes and Privilege Levels

The processor supports two primary modes of execution:

- **Thread mode** (for executing application software)
- **Handler mode** (for handling processor exceptions and peripheral interrupts)

By default, the processor is in thread mode out of reset. If an exception is issued to the processor, the processor handles the exception in handler mode and return to thread mode after handler execution is complete. Code running in thread mode can be configured as being privileged or unprivileged, based on the configuration of the CONTROL register. Code running in handler mode always executes as privileged.

Note

Handler mode always uses the MSP, so the processor ignores explicit writes to the active stack pointer bit of the CONTROL register when in Handler mode. The exception entry and return mechanisms automatically update the CONTROL register based on the EXC_RETURN value.

By default, thread mode uses the MSP. To switch the stack pointer that is used in thread mode to the PSP, either:

1. Use the `MSR` instruction to set the CONTROL.SPSEL bit to 1.
2. Perform an exception return to thread mode with the appropriate EXC_RETURN value.

In general, code which executes as privileged has complete control of the processor configuration, including control of the MPU, Systick, NVIC, and SCB. Only privileged code can change the privilege level for code running in thread mode.

Code that is executing in thread mode in an unprivileged state cannot access the previously mentioned resources (MPU, Systick, NVIC, and SCB).

4.2.5 Address Space and Supported Data Sizes

AM13E230x devices implement a flat memory map with a 32-bit byte-addressable address space. Byte addresses are unsigned numbers ranging from zero to 2^{32}-1.

Address Space

The processor sees the address space as containing 2^{30} 32-bit words, with each word being word-aligned (4-byte aligned). Pointers are always 32-bits, and stack operations (for example, push, pop) increment the stack pointer by 4 addresses (4 bytes). Address calculations by the processor wrap around if they overflow or underflow the 32-bit memory space.

Instruction fetches by the processor are always 16-bit half-word aligned.

Data reads by the processor must be naturally aligned (for example, words must be word aligned, half words must be half-word aligned, etc.).

Supported Data Sizes

The processor supports the following data types:

- 32-bit words
- 16-bit halfwords
- 8-bit bytes
- 32-bit single-precision floating point numbers
- 64-bit double-precision floating point numbers

Signed and unsigned data is supported, and signed data is stored in CPU registers in 32-bit two's complement format. The Armv8-M instruction set does not provide native instructions supporting operations on 64-bit double-word data.

Load operations from memory to a CPU register can be signed or unsigned when the data size is less than 32 bits. When loading unsigned half-word or byte data to a CPU register, the value is zero-extended to 32 bits automatically. When loading signed half-word or byte data to a CPU register, the value is sign-extended to 32 bits automatically.

Stores from CPU registers to memory are sign agnostic.

All instruction and data accesses use little endian byte order.

4.3 Interrupts and Exceptions

Peripheral interrupt exceptions and system exceptions temporarily pause the processor's normal execution flow so that the processor can be used to handle an event.

The following can cause interruption of normal execution flow:

- **Reset**: The exception model treats reset as a special form of exception. When either power-on or warm reset is asserted, the operation of the processor stops, potentially at any point in an instruction. When reset is deasserted, execution restarts from the address provided by the reset entry in the vector table. Execution restarts as privileged execution in Thread mode.
- **Non-Maskable Interrupt (NMI)**: A Non-Maskable Interrupt (NMI) can be signaled by a peripheral or triggered by software. This is the highest priority exception other than reset. It is permanently enabled and has a fixed priority of -2. NMIs cannot be masked or preempted by any exception other than Reset.
- **HardFault**: A HardFault is an exception that occurs because of an error during exception processing, or because an exception cannot be managed by any other exception mechanism. HardFaults have a fixed priority of -1, meaning they have higher priority than any exception with configurable priority.
- **MemManage**: A MemManage fault is an exception that occurs because of a memory protection violation, compared to the MPU or the fixed memory protection constraints, for both instruction and data memory transactions. This fault is always used to abort instruction accesses to "Execute Never" (XN) memory regions.
- **BusFault**: A BusFault is an exception that occurs because of a memory-related fault for an instruction or data memory transaction. This might be from an error that is detected on a bus in the memory system.
- **UsageFault**: A UsageFault is an exception that occurs because of a fault related to instruction execution. This includes: an undefined instruction, an illegal unaligned access, invalid state on instruction execution,

or an error on exception return. Software can configure the core to report a UsageFault when: there is an unaligned address on word and halfword memory access or division by zero.
- **SVCall**: A Supervisor Call (SVC) is an exception that is triggered by the SVC instruction.
- **DebugMonitor**: A DebugMonitor exception. If halting debug is disabled and the debug monitor is enabled, a debug event causes a DebugMonitor exception when the group priority of the DebugMonitor exception is greater than the current execution priority.
- **PendSV**: PendSV is an asynchronous request for system-level service.
- **SysTick**: A SysTick exception is an exception the system timer generates when it reaches zero. Software can also be configured to generate a SysTick exception.
- **Interrupt (IRQ)**: An interrupt, or IRQ, is an exception signaled by a peripheral, or generated by a software request. All interrupts are asynchronous to instruction execution. In the system, peripherals use interrupts to communicate with the processor.

Properties of the different exception types are detailed in Table 4-4.

Exception States

Each exception source to the processor will be in one of the below states at any given point in time:
- **Inactive** (not active, not pending)
- **Pending** (waiting to be serviced by the processor)
- **Active** (actively being serviced by the processor but has not completed)
- **Active and pending** (actively being serviced by the processor and there is a pending exception from the same source)

Exception Prioritization, Entry, and Exit

Exceptions are prioritized by the processor together with the Nested Vectored Interrupt Controller (NVIC). Each exception has either a fixed priority (Reset, NMI, HardFault) or a configurable priority (MemManage, BusFault, UsageFault, DebugMonitor, SVCall, PendSV, SysTick, peripheral IRQs). Exceptions with configurable priority can be disabled by application software running in privileged mode. Exceptions with fixed priority cannot be disabled.

The processor exception model supports preemption, tail-chaining, and late-arrival features to boost exception handling performance:

- In the **preemption** case, if an exception of higher priority is pending when an exception of lower priority is executing, the higher priority exception will preempt the ongoing handler servicing the lower priority exception.
- In the **tail-chaining** case, if a valid-for-entry exception is pending at the time of completion of an exception handler, then the application context is not restored from the stack and control is given immediately to the pending exception.
- In the **late-arriving** case, if a higher priority exception occurs during entry to a lower priority exception, the higher priority exception will be serviced first after the processor state is saved to the stack. Once the higher priority exception handling is complete, the lower priority exception (which is still pending) is serviced based on the tail-chaining procedure.

An exception entry is issued if and when all of the following are true:

- An exception is in a **pending** state
- The priority of the pending exception is higher than the limit set by the exception mask register (PRIMASK)
- The processor is currently in either thread mode (not servicing an exception) or the newly pending exception is higher priority than the exception which is currently being serviced (resulting in a preemption)

Processor exceptions are vectored. When an exception occurs, the current processor state is pushed onto the stack which was active at the time of the event, and execution is vectored to the entry point address in the vector table corresponding to the exception which is to be processed.

If the exception is tail-chained to a previous handler which has completed, then there is no need to push any state to the stack and the interrupt service routine can be vectored to immediately. Likewise, if the exception is higher priority than a previous exception which started entry but did not complete entry, then there is no need to save the context again (late arrival).

Upon completion of an exception handler, if there is no exception pending which needs to be handled then the processor will pop the state from the stack and restore the processor to the previous state which it was in when the exception occurred.

4.3.1 Peripheral Interrupts (IRQs)

Peripheral interrupt functionality is managed by several components on the device:

- The Nested Vectored Interrupt Controller (NVIC)
- The Wake-Up Controller (WUC)

AM13E230x devices include an Arm Nested Vectored Interrupt Controller (NVIC) with the Cortex-M33 CPU for managing peripheral interrupts. The NVIC operation is tightly integrated with the processor and supports 48 native peripheral interrupt sources.

The Wake-Up Controller (WUC) determines if the PD1 power domain (containing the processor) needs to be powered up to service a peripheral interrupt if the PD1 domain is powered down in STOP or STANDBY mode.

4.3.1.1 Nested Vectored Interrupt Controller (NVIC)

The Nested Vectored Interrupt Controller (NVIC) is an industry-standard Arm component which interfaces peripheral interrupts (which are external to the processor) into the CPU. The NVIC supports connection of up to 48 native peripheral interrupt sources. For information on which peripheral is being mapped, please refer to .

Table 4-2. NVIC IRQ Mapping

NVIC IRQ mapping	Interrupt Source
0	PMCU
1	Debug
2	Flash
3	WWDT0
4	GPIO0
5	GPIO1
6	GPIO2
7	GPIO3
8	ADC0_SEQ0
9	ADC0_SEQ1
10	ADC0_SEQ2
11	ADC0_SEQ3
12	ADC0_DCOMP
13	ADC1_SEQ0
14	ADC1_SEQ1
15	ADC1_SEQ2
16	ADC1_SEQ3
17	ADC1_DCOMP
18	ADC2_SEQ0

Table 4-2. NVIC IRQ Mapping (continued)

NVIC IRQ mapping	Interrupt Source
19	ADC2_SEQ1
20	ADC2_SEQ2
21	ADC2_SEQ3
22	ADC2_DCOMP
23	MCAN
24	PWM0_INT
25	PWM1_INT
26	PWM2_INT
27	PWM3_INT
28	PWM4_INT
29	ECAP0
30	ECAP1
31	EQEP0
32	EQEP1
33	EQEP2
34	UC0
35	UC1
36	UC2
37	UC3
38	UC4
39	UC5
40	DMA
41	TINIE_LITE
42	EPI
43	AES.INT_EVENT0
44	TIMG4_0
45	TIMG12_0
46	TMU_LUF_INT
47	TMU_LVF_INT
48-63	Reserved (Tie low)

The NVIC is configured through memory-mapped registers in the system private peripheral bus (PPB) region. See *Arm Cortex-M33 NVIC Registers* for the list of NVIC registers. The software development kit (SDK) provided with the devices supports the standard Arm Cortex Microcontroller Software Interface Standard (CMSIS) register access definitions for the NVIC. Application software must use 32-bit aligned, word-size transactions when accessing any NVIC register.

In addition to interfacing peripheral interrupts to the processor, the NVIC also supports programmable priority for each interrupt.

Enabling and Disabling Interrupts

Peripheral interrupt enables can be read, set, and cleared through the interrupt set-enable (ISER) and interrupt clear-enable (ICER) registers in the NVIC. Each ICER and ISER register supports 32 different interrupts. The interrupts are mapped to the ISER and ICER registers with interrupt zero in the BIT0 position (LSB) and interrupt 31 in the BIT31 position (MSB) of each register. For example, SETENA[m] in NVIC_ISERn, allows interrupt 32n + m to be accessed.

To enable an interrupt, set the corresponding enable bit in the ISER register. Writing a '0' to ISER has no effect. It is possible to read the ISER or ICER registers to determine which interrupts are enabled. Upon a read, a '1' indicates that an interrupt is enabled; a '0' indicates disabled. To disable an interrupt, set the corresponding clear enable bit in the ICER register. Writing a '0' to ICER has no effect.

Note

In addition to enabling a peripheral interrupt at the NVIC, it is generally necessary to also configure the interrupt configuration of the corresponding peripheral as well. Most peripherals have multiple interrupt sources, which are merged together in the peripheral to source a single NVIC interrupt. Masking of individual peripheral interrupts is done within the peripheral's interrupt management registers.

In the event that an interrupt is disabled in the NVIC, if the interrupt is asserted by the corresponding peripheral then the NVIC interrupt will go to a pending state but the processor is not interrupted. If an interrupt is disabled when in an active state (when a handler is running) it will remain active until the exception handler returns or a reset occurs, but no further activations will happen.

Note

If a peripheral asserts an interrupt to the NVIC, but that peripheral's interrupt in the NVIC is disabled, the device may remain in a higher power mode than expected as the wake-up controller (WUC) is holding an event for the processor. To prevent this situation, ensure peripheral interrupts are masked at the peripheral directly, versus only masking interrupts at the NVIC.

Setting and Clearing Pending Interrupt Status

Pending interrupt status can be read, set, and cleared through the interrupt set-pending (ISPR) and interrupt clear-pending (ICPR) registers in the NVIC. Each ISPR and ICPR register supports 32 interrupts, with interrupt zero in the BIT0 position (LSB) and interrupt 31 in the BIT31 position (MSB) of each register. For example, SETPEND[m] in NVIC_ISPRn, allows interrupt 32n + m to be accessed.

To read if an interrupt is pending, read either the ISPR or the ICPR. Upon a read, a '1' indicates that an interrupt is pending; a '0' indicates not pending. To set an interrupt to a pending state through software, set the corresponding bit in the ISPR register. Writing a '0' to ISPR has no effect. To clear an interrupt pending state, set the corresponding bit in the ICPR register. Writing a '0' to ICPR has no effect. Note that if a peripheral interrupt condition is still present, the pending state will be set again by hardware even if it is cleared.

Active State of Interrupts

The IABR registers in the NVIC indicate the active state of each interrupt. Each IABR register supports 32 interrupts, with interrupt 0 in the BIT0 position (LSB) and interrupt 31 in the BIT31 position (MSB) of each register. For example, ACTIVE[m] in NVIC_IABRn, indicates the active state for interrupt 32n+m.

To read if an interrupt is active, read the IABR register. Upon a read, a '1' indicates that an interrupt is active; a '0' indicates not active.

Software Triggered Interrupts

The STIR register in the NVIC is used to generate an interrupt from software. The INTID field of the STIR register designates the interrupt ID of the interrupt to trigger, in the range of 0-48. For example, a INTID value

of 0x03 specified IRQ3. When the USERSETMPEND bit in the CCR register is set to '1', unpriveileged software can access the STIR register. Only privileged software can enable unprivileged access.

Setting Interrupt Priority

Interrupts on the NVIC have programmable priority. There are 16 priority levels possible. Priority is set by programming the IPR registers in the NVIC. Each priority field is 8 bits in length, and the priority for 4 interrupts is configured per 32-bit IPR register. AM13E230x only implements the most significant 4 bits of each 8-bit priority field (giving the 16 priority levels). Lower priority values have higher priority. System exceptions (Reset, NMI, and HardFault) have fixed priorities of -4, -2, and -1, respectively. As such, these exceptions always have higher priority than peripheral interrupts. Peripheral interrupt priorities are programmable as 0, 16, 32, 48, 64, …, 240 with 0 being highest priority and 240 being lowest priority.

If the processor is currently handling an exception, it can only be preempted by a higher priority exception. In the event that there are multiple exceptions in a pending state which all have the same priority level assigned, the exception with the lowest exception number is taken first.

Note

Application software must not change the priority of an interrupt while the corresponding interrupt is either active (being handled) or enabled. Doing so can result in unpredictable behavior.

Table 4-3. Arm Cortex-M33 NVIC Registers

Address	Register	CMSIS	Description
0xE000.E100 - 0xE000.E13C	NVIC_ISER0 - NVIC_ISER15	NVIC->ISER[0] - NVIC->ISER[15]	Interrupt set-enable register
0xE000.E180 - 0xE000.E1BC	NVIC_ICER0 - NVIC_ICER15	NVIC->ICER[0] - NVIC->ICER[15]	Interrupt clear-enable register
0xE000.E200 - 0xE000.E23C	NVIC_ISPR0 - NVIC_ISPR15	NVIC->ISPR[0] - NVIC->ISPR[15]	Interrupt set-pending register
0xE000.E280 - 0xE000.E2BC	NVIC_ICPR0 - NVIC_ICPR15	NVIC->ICPR[0] - NVIC->ICPR[15]	Interrupt clear-pending register
0xE000.E300 - 0xE000.E33C	NVIC_IABR0 - NVIC_IABR15	NVIC->IABR[0] - NVIC->IABR[15]	Interrupt active bit register
0xE000.E400 - 0xE000.E5DC	NVIC_IPR0 - NVIC_IPR119	NVIC->IPR[0] - NVIC->IPR[119]	Interrupt priority register (0 - 3) - Interrupt priority register (115 - 119)
0xE000EF00	STIR	NVIC->STIR	Software trigger interrupt register

4.3.1.2 Wake Up Controller (WUC)

The Wake Up Controller (WUC) is responsible for monitoring for assertion of interrupts when the processor is powered down in the STOP or STANDBY operating mode. In these modes, the entire PD1 power domain is power gated, and as such, the processor and NVIC are not available to check for interrupts. The WUC retains a copy of which peripheral interrupt sources to the NVIC were enabled when the processor entered STOP or STANDBY mode. In the event that an enabled interrupt is issued, the WUC will handshake with the PMCU to bring the device out of STOP or STANDBY mode so that the CPU can service the interrupt. The WUC will capture the interrupt state and present it to the NVIC and processor when the processor is brought up, such that the processor will see the interrupt even if the raw interrupt status of the peripheral is removed before the processor finishes powering up to service the interrupt.

The WUC requires no configuration by application software upon entry to or exit from low-power modes, and operation is transparent to application software.

4.3.2 Interrupt and Exception Table

The Arm Cortex-M33 interrupt vector table is 500 words long (2000 bytes). The complete platform interrupt and exception table with vector table addresses is given in Table 4-4.

ADVANCE INFORMATION

See Table 4-2 for a complete list of which interrupts a particular device supports along with a more detail interrupt vector table describing which NVIC number is used for what peripheral.

Table 4-4. AM13E230x Platform Processor Interrupt and Exception Table

Exception Number	NVIC Number(1)	Priority Group	Secure Vector	Vector Table Address	Vector Description(2)
-	-	-	-	0x0000.0000	Stack pointer
1	-	-4	Reset	0x0000.0004	Reset vector
2	-14	-2	NMI	0x0000.0008	NMI handler
3	-13	-1	HardFault	0x0000.000C	Hard fault handler
4	-12	Configurable	MemManage	0x0000.0010	Memory Protection Violation
5	-11	Configurable	BusFault	0x0000.0014	Memory-related fault
6	-10	Configurable	UsageFault	0x0000.0018	UsageFault
7	-	-	Reserved	0x0000.001C	-
8	-	-	Reserved	0x0000.0020	-
9	-	-	Reserved	0x0000.0024	-
10	-	-	Reserved	0x0000.0028	-
11	-5	Configurable	SVCall	0x0000.002C	Supervisor call handler
12	-4	Configurable	DebugMonitor	0x0000.0030	DebugMonitor exception
13	-	-	Reserved	0x0000.0034	-
14	-2	Configurable	PendSV	0x0000.0038	Pended supervisor handler
15	-1	Configurable	SysTick	0x0000.003C	SysTick handler
16	0	Configurable	Device Interrupt 0	0x0000.0040	Handler for Device Interrupt 0
17	1	Configurable	Device Interrupt 1	0x0000.0044	Handler for Device Interrupt 1
18	2	Configurable	Device Interrupt 2	0x0000.0048	Handler for Device Interrupt 2
...	...	...	...	...	...
63	47	Configurable	Device Interrupt 47	0x0000.00FC	Handler for Device Interrupt 47

(1) The NVIC number also indicates the relative interrupt priority if multiple NVIC interrupts have the same group priority. However, an interrupt does not preempt an active handler for another interrupt with the same group priority, even if the interrupt has a higher (numerically lower) NVIC position. For preemption to occur, the new interrupt must be configured to a higher priority group (numerically lower).

(2) For more information on the exceptions, please see Arm Cortex-M33 Generic User Guide

ADVANCE INFORMATION

Non-Maskable Interrupt (NMI)

The CPU implements a nonmaskable interrupt, which handles critical interrupts which must be serviced immediately by the processor. The NMI interrupt sources are managed by SYSCTL. See the corresponding NMI information in the SYSCTL section of the PMCU chapter.

4.3.3 Processor Lockup Scenario

There are several exception conditions which can cause the processor to enter a lockup state. On AM13E230x devices, a processor lockup is considered a fatal fault which always triggers a SYSRST to clear the lockup condition and restart the system.

A lockup state is entered by the processor if an SVC (supervisor call) or fault condition occurs while the processor is handling an exception with priority of -1 or higher (numerically lower). Such a fault is considered by the processor to be unexpected under normal operating conditions.

The following examples are conditions that can trigger a lockup state in the processor:

- The processor cannot fetch the stack pointer or reset vector at reset

- The processor cannot fetch the NMI vector
- The processor cannot fetch the hard fault vector
- A memory fault occurs when the processor is already handling an exception with priority of -1 or -2 (hard fault or NMI)
- A supervisor call (SVC) occurs when the processor is already handling an exception with priority of -1 or -2 (hard fault or NMI)
- A usage fault or undefined instruction is fetched when the processor is already handling an exception with priority of -1 or -2 (hard fault or NMI)
- A BKPT instruction is executed when the processor is already handling an exception with priority of -1 or -2 (hard fault or NMI)

4.4 CPU Peripherals

The Arm Cortex-M33 includes tightly coupled peripherals for system timing, memory protection, floating point computation, and security

4.4.1 System Control Block (SCB)

The System Control Block (SCB) provides system implementation information and system control functionality, as well as configuration, control, and reporting of processor exceptions.

The SCB is configured through memory-mapped registers in the system private peripheral bus (PPB) region. See Table 4-5 for the list of SCB registers. The software development kit (SDK) provided with the devices supports the standard Arm Cortex Microcontroller Software Interface Standard (CMSIS) register access definitions for the SCB. Application software must use 32-bit aligned, word-size transactions when accessing any SCB register.

Table 4-5. Arm Cortex-M33 System Control Block Registers

Address	Register	CMSIS	Description
0xE000.ED00	CPUID	SCB->CPUID	Read-only register indicating the processor part number, version, and implementation information
0xE000.ED04	ICSR	SCB->ICSR	Provides a set-pending bit for the non-maskable interrupt exception, and set-pending and clear-pending bits for the PendSV and SysTick exceptions
0xE000.ED08	VTOR	SCB->VTOR	Used to specify the vector table offset from 0x0000.0000
0xE000.ED0C	AIRCR	SCB->AIRCR	Used to issue a CPU reset request (SYSRESETREQ)
0xE000.ED10	SCR	SCB->SCR	System control register, used to control low-power mode behavior
0xE000.ED14	CCR	SCB->CCR	Read-only register indicating behavior of the processor
0xE000.ED18	SHPR1	SCB->SHP[1]	Used to configure priority level of Usagefault, BusFault, and MemManage system handlers
0xE000.ED1C	SHPR2	SCB->SHP[2]	Used to configure the priority level of the SVCall system handler
0xE000.ED20	SHPR3	SCB->SHP[3]	Used to configure the priority level of the SysTick and PendSV system handlers
0xE000.ED24	SHCSR	SCB->SHCSR	Enables the system handlers. Indicates the pending status of the BusFault, MemManage fault, and SVC exceptions, and indicates the active status of the system handlers.
0xE000.ED28	CFSR	SCB->CFSR	Used to indicate cause of MemManage fault, BusFault, and Usage Fault
0xE000.ED2C	HFSR	SCB->HFSR	Provides information about events that active the HardFault handler
0xE000.ED34	MMFAR	SCB->MMFAR	Contains the address of the MemManage fault location
0xE000.ED38	BFAR	SCB->BFAR	Contains the address of the BusFault location

For detailed information on the SCB register configuration, see the SCB section of the Arm Cortex-M33 Devices Generic User Guide.

4.4.2 System Tick Timer (SysTick)

The system tick timer (SysTick) is a tightly-coupled timer clocked by MCLK, which can be used for system time keeping. The SysTick is available in RUN and SLEEP modes but is not available for use in STOP, STANDBY, or SHUTDOWN modes.

The SysTick is configured through memory-mapped registers in the system private peripheral bus (PPB) region. See *SysTick Registers* for the list of SysTick configuration registers. The software development kit (SDK) provided with the devices supports the standard Arm Cortex Microcontroller Software Interface Standard (CMSIS) register access definitions for the SysTick.

The SysTick timer can be used in several different ways, including:

- As an RTOS timer which fires at a programmable rate (for example, 100Hz) and invokes a SysTick routine
- A high-speed alarm timer
- A simple counter used to measure time to completion and time used in an application
- A timeout counter to check that a routine has not timed out within a specified period

The SysTick timer is a simple 24-bit down counter, which counts down from its reload value to zero. Upon reaching zero, SysTick will reload the value programmed into the reload value register (SYST_RVR) on the next clock cycle, and then again begin counting down to zero. Writing a value of zero to the SYST_RVR disables the counter on the next wrap.

A SysTick event is generated when the SysTick counter reaches zero, and which point the COUNTFLAG status flag will be set. Reading the SYST_CSR register will clear the COUNTFLAG status bit. Writing to the current value register (SYST_CVR) clears the register and the COUNTFLAG status but does not generate an interrupt to the CPU. Reading SYST_CVR returns the counter value at time of access.

Application software must only use 32-bit word-aligned word accesses to the SysTick registers. To initialize the SysTick, follow the steps below:

1. Program the desired reload value (example: to generate a flag every 1000 MCLK cycles, program 999) to SYST_RVR
2. Clear the current value by writing to the SYST_CVR register
3. Program the SYST_CSR register to enable SysTick

Note

The SysTick counter does not decrement when the CPU is halted for debug.

Table 4-6. SysTick Registers

Address	Register	CMSIS	Description
0xE002.E014	SYST_CSR	SysTick->CTRL	Control and status register used to enable/disable SysTick and the SysTick exception, and to check the COUNTFLAG status
0xE002.E014	SYST_RVR	SysTick->LOAD	Reload register used to program the counter reload value to set the SysTick interval in MCLK cycles
0xE002.E018	SYST_CVR	SysTick->VAL	Returns the current value of the SysTick counter; a write clears the counter to zero and clears COUNTFLAG in SYST_CSR
0xE002.E01C	SYST_CALIB	SysTick->CALIB	Indicates the SysTick calibration value

4.4.3 Memory Protection Unit (MPU)

The Memory Protection Unit (MPU) can be used to check all memory accesses made by the processor against a set of access permission policies which can be defined by the programmer. When used together with the privileged/unprivileged execution modes of the Arm Cortex-M33, the MPU supports limiting access to certain memory locations to privileged code only. If unprivileged code accesses a nonrestricted region, execution continues as if the MPU was not present. However, if unprivileged code issues an access to a restricted region, a HardFault is generated if MemMange fault is not enabled in the processor. It is also possible to restrict access to both privileged and unprivileged code (no access possible through the processor).

The MPU is configured through memory-mapped registers in the system private peripheral bus (PPB) region. See *MPU Registers* for the list of MPU configuration registers. The software development kit (SDK) provided with the devices supports the standard Arm Cortex Microcontroller Software Interface Standard (CMSIS) register access definitions for the MPU.

Since the MPU checks all memory accesses from the processor (including accesses to flash, SRAM, and peripherals), it is well suited for improving reliability and robustness in threaded applications involving an RTOS. The MPU can be used to restrict the memory access of individual threads, including establishing stack boundaries and limiting access to specific peripherals. Key state data used by the RTOS can be protected from modification by unprivileged threads.

Note

The MPU only monitors access to the memory map which originate from the processor. The MPU does not restrict the access of the DMA controller.

Note

Protecting flash memory regions with the MPU does not protect the flash memory from being read or modified by the flash controller. This is because the flash controller can take control of the flash memory banks directly for read verify, program, and erase operations. Dedicated write protection functions are provided within the flash controller, and these mechanisms are for use when write protecting the flash memory. Software can configure the MPU to restrict access to the flash controller registers, thus preventing the CPU from configuring or initiating a flash command directly.

The MPU provides a mechanism to partition the device memory map into 8 regions (numbered 0-7) plus a default background region. Each region can be configured with access permissions and memory attributes, and regions can be configured to overlap if desired. In the case of overlapping regions, a memory access to a location existing in multiple regions is subjected to the attributes of the region with the highest number.

When the MPU is not enabled (ENABLE bit is cleared in the MPU_CTRL register), the device uses the default memory map and the CPU has access to the memory map as if the MPU was not present.

When the MPU is enabled for use, access to the vector table and the system control space are always permitted, but access to any other location depends on the following:

- The region configurations
- Whether the access was privileged or unprivileged
- The privileged software default memory map access control configuration (PRIVDEFENA)

Because the Arm Cortex-M33 is a single-bus CPU architecture, there is no delineation between an instruction fetch and a data access by the MPU. Instructions and data are treated the same.

Note

If the MPU is enabled (ENABLE bit is set), then PRIVDEFENA must be set to give privileged software access to the memory map, or at least one memory region must be configured and enabled for the CPU to proceed without a hard fault. If PRIVDEFENA is cleared, no regions are defined and enabled, and the MPU is enabled, then the entire memory map is restricted, and any access generates a fault.

Table 4-7. MPU Registers

Address	Register	CMSIS	Description
0xE002.ED90	MPU_TYPE	MPU->TYPE	Type register indicating that if the MPU is present, and if so, how many regions supported
0xE002.ED94	MPU_CTRL	MPU->CTRL	MPU control register for enabling and configuring the MPU
0xE002.ED98	MPU_RNR	MPU->RNR	Region select register for using MPU_RBAR and MPU_RLAR
0xE002.ED9C	MPU_RBAR	MPU->RBAR	Defines the base address of the MPU region selected by the MPU_RNR
0xE002.EDA0	MPU_RLAR	MPU->RLAR	The MPU_RLAR defines the limit address of the MPU region selected by the MPU_RNR

For detailed information on the MPU register configuration, see the MPU section of the Arm Cortex-M33 Devices Generic User Guide.

4.4.4 Floating Point Unit (FPU)

The Cortex®-M33 Floating-Point Unit (FPU) implements the FPv5 floating-point extensions.The FPU fully supports single-precision add, subtract, multiply, divide, multiply and accumulate, and square root operations. It also provides conversions between fixed-point and floating-point data formats, and floating-point constant instructions.

The FPU provides floating-point computation functionality that is compliant with the ANSI/IEEE Std 754-2008, IEEE Standard for Binary Floating-Point Arithmetic, referred to as the IEEE 754 standard.

The FPU contains 32 single-precision extension registers, which users can also access as 16 doubleword registers for load, store, and move operations.

Note

The FPU is not connected to the NVIC. This requires the programmer to check the FPSCR register to see the status of the FPU, which is accessible by instruction.

Table 4-8. FPU Registers

Address	Register	CMSIS	Description
0xE002.EF34	FPCCR	FPU->FPCCR	Used to control different attributes of the FPU
0xE002.EF38	FPCAR	FPU->FPCAR	Holds the location of upopulated floating-point register space
0xE002.EF3C	FPDSCR	FPU->FPDSCR	Holds the default values for the floating-point status control data

For detailed information on the FPU, see the FPU section of the Arm Cortex-M33 Devices Generic User Guide.

4.4.5 Digital Signal Processing Extension

The Arm Cortex-M33 allows specific instructions to be used for digital signal processing (DSP). This gives programmers the ability to use the DSP instructions given in the Armv8-m instruction set. For more information on these instructions please see Armv8-M Architecture Reference Manual.

4.4.6 Custom Datapath Extension TMU

The Trigonometric Math Unit (TMU) is a floating-point hardware accelerator that uses the Custom Datapath Extension (CDE) interface of the Arm® Cortex-M33 to enable faster trigonometric math function functions. The trigonometric math functions are supported as an enhanced instruction set on the M33 core that utilizes the Floating Point Unit (FPU) registers for the operands and the data path implemented on the CDE interface. For more information on these instructions, please refer to Chapter 5 .

4.5 Read-Only Memory (ROM)

The CPUSS contains a 48KB read-only memory which contains the executable code for the boot configuration routine (BCR) and bootstrap loader (BSL). The ROM is active after a BOOTRST, or after a SYSRST with BSL entry/exit. The ROM is disabled automatically when the application is started and is not accessible by application software.

Chapter 5

Trigonometric Math Unit (TMU)

This chapter describes the trigonometric math unit (TMU) .

ADVANCE INFORMATION

5.1 Introduction

The Trigonometric Math Unit (TMU) is a floating-point hardware accelerator that uses the Custom Datapath Extension (CDE) interface of the Arm® Cortex-M33 core to reduce the effective MIPS of the M33 core by enabling faster trigonometric math function calculations. The trigonometric math functions are supported as an enhanced instruction set on the M33 core that utilizes the M33 core's Floating Point Unit (FPU) registers for the operands and the data path implemented on the CDE interface.

In many sensing and control algorithms, common math operations such as Sine, Cosine, Arctangent, Division, and Square Root take many cyces to complete in a software-only implementation. The TMU hardware accelerator significantly reduces cycle times, and thus increases the achievable control frequencies of a system by adding custom instructions specifcally for these operations to the M33 CPU.

5.2 Features

- TMU functions implemented as native Cortex-M33 instructions
- Utilizes Custom Datapath Extension (CDE) interface of the M33 CPU
- M33 FPU registers used to store operands, data path is implemented outside of FPU
- C29 TMU32 operations supported:
  - SINPUF32
  - COSPUF32
  - ATANPUF32
  - DIVF32
  - QUADF32
  - SQRTF32
  - IEXP2F32
  - LOG2F32
  - ISQRTF32
- RSCTFLT instruction to read/write Latched Underflow (LUF) and Latched Overflow (LVF) flags
- External interface for LUF/LVF interrupts and ROM parity error
- Safety enabled through parity protection for ROM
- No pipelining of back to back instructions inside the CDE TMU - M33 CPU expected to issue new instruction after the result is given for the previous instruction
- Interruptible multicycle instructions

The MCU SDK enables TMU usage in software. Refer to the SDK for examples using the TMU.

5.3 Functional Operation

5.3.1 Supported TMU Instructions

The following mathematical operations are supported on the M33 CDE TMU accelerator:

Table 5-1. M33 CDE TMU Instructions

Instruction	Description	# of Inputs	# of Outputs	Equivalent Operation	# of M33 Cycles
SINPUF32	Returns the SINE of input value	1 : Sm	1: Sd	sin(Sm * 2pi rad) Sm = -1.0 to 1.0	5
COSPUF32	Returns the COSINE of input value	1 : Sm	1 : Sd	cos(Sm * 2pi rad) Sm = -1.0 to 1.0	5
ATANPUF32	Returns the ARCTAN of input value	1 : Sm	1: Sd	arctan(Sm) rad/2pi Sm = -1.0 to 1.0 Sd = -0.125 to 0.125	5

Table 5-1. M33 CDE TMU Instructions (continued)

Instruction	Description	# of Inputs	# of Outputs	Equivalent Operation	# of M33 Cycles
DIVF32	Returns the DIV value of two input values	2 : Sn = X, Sm = Y	1: Sd = (X / Y)	Sn / Sm	6
QUADF32	Returns the quadrant value of the (X,Y) input values	2 : Sn = X, Sm = Y	1 : Sd	Quadrant(X,Y)	1
SQRTF32	Returns the Square Root of input value	1 : Sm	1 : Sd	sqrt(Sm)	6
IEXP2F32	Returns the Inverse Exponent of input value. **Note** There is loss of accuracy with increased values of the exponent. Refer to the software example test cases in the SDK for more details.	1 : Sm	1 : Sd	$1 / (2^{Sm})$	5
LOG2F32	Returns the Base-2 Logarithm of input value	1 : Sm	1 : Sd	$Log_2(Sm)$	6
ISQRTF32	Returns the Inverse Square Root of input value	1 : Sm	1 : Sd	1 / sqrt(Sm)	6
RSCTFLG	Read/Set/Clear the LUF and/or LVF flags that are maintained inside the TMU CDE **Note** For a **flag read only** instruction, keep Rm[1:0]=00 **Note** The default value of the LUV and LVF flags at reset are 0.	2: Rn : 2-bit write value (Rn[0] - LUF, Rn[1]-LVF) Rm : 2-bit mask (Rm[0] - LUF, Rm[1] - LVF)	1 : Rd 2-bit read value (Rd[0] - LUF, Rd[1] - LVF)	**Operation** • LUF = (LUF & ~Rm[0]) | (Rn[0] & Rm[0]) • LVF = (LVF & ~RM[1]) | (Rn[1] & Rm[1]) • Rd[0] = LUF • Rd[1] = LVF **Examples** • Read both flags • Set both flags • Clear both flags • Set LUF only • Set LVF only • Clear LUF only • Clear LVF only	1

- Sd: Destination Register (floating-point)
- Sn: Source Register (floating-point)
- Sm: Source Register (floating-point)
- Rd: Destination Register (general-purpose)
- Rn: Source register (general-purpose)
- Rm: Source register (general-purpose)

Chapter 6

TinyEngine™ NPU

This chapter describes the features and operation of the TinyEngine™ NPU, used to improve the efficiency of machine learning inferencing.

6.1 Introduction

The TinyEngine™ NPU can support intelligent inferencing running pre-trained models. Capable of 800–1600MOPS(Mega Operations Per Second) with example model support for ARC fault detection or Motor Fault detection, the TinyEngine™ NPUprovides up to 10x Neural Network (NN) inferencing cycle improvement versus a software only based implementation. Load and train models with tools from TI: Model Composer GUI or TI's command-line Modelmaker tool for an advanced set of capabilities. Both of these options automatically generate source code for the ARM® Cortex® -M33 , eliminating the need to manually write code.

Figure 6-1shows the toolchain and steps to add TinyEngine™ NPU support to a project, starting with importing or using existing models from TI, training the models, generating the associated software libraries, and integrating into an existing Code Composer Studio™ IDE project.

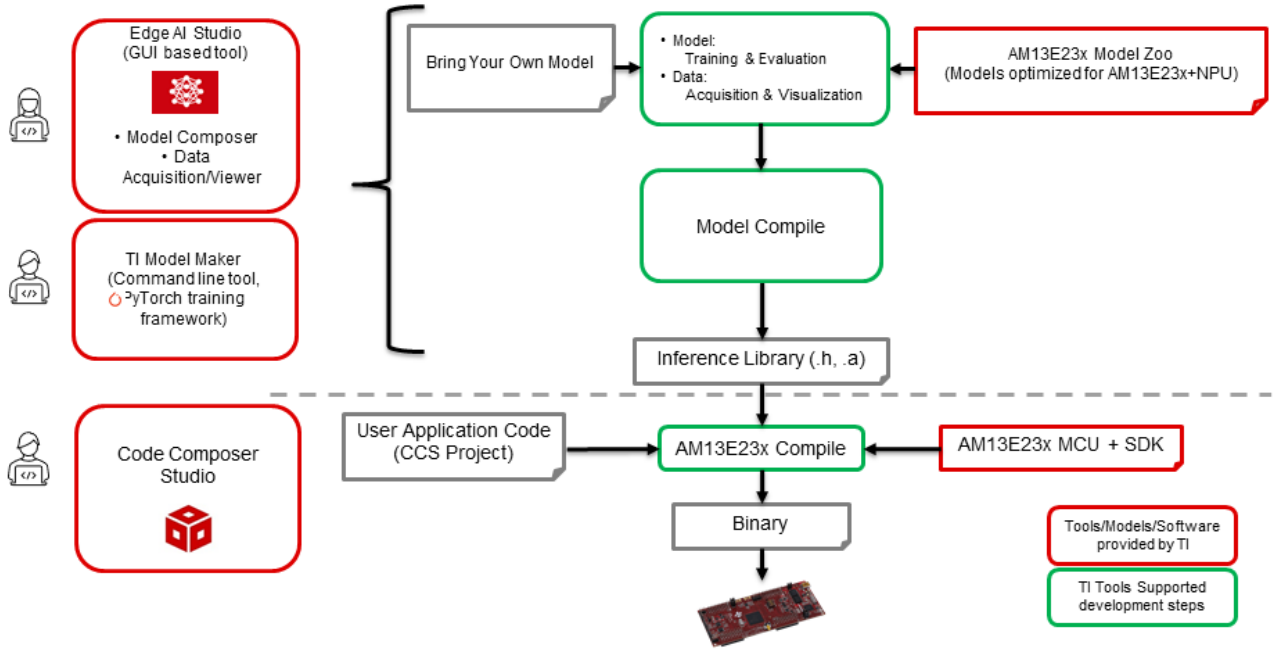


Figure 6-1. TinyEngine™ NPUDevelopment Flow

6.1.1 TinyEngine™ NPU Related Collateral

Getting Started Materials

- Design more efficient, compact motor systems with the DRV7308 GaN IPM (Video)

ADVANCE INFORMATION

The read-only memory (ROM) is used to configure the device before starting the user application.

7.1 ROM Overview

The ROM contains the executable code for the boot configuration routine (BCR) and bootstrap loader (BSL). The ROM includes:

- Lifecycle Detection of the device
- Configuring the SoC Security as per the NON-Main flash data
- Loads TI trim data
- Authentication of the Application in Flash
- Logs boot errors in all life-cycle states
- ROM Services
  - Authentication of service requests through DSSM
  - Debug Services
  - Key Services
  - RMA Services
  - BCR Existing Services

The ROM is active after a BOOTRST, or after a SYSRST with BSL entry/exit. The ROM is disabled automatically when the application is started and is not accessible by application software.

AM13E230x devices provide a 48KB ROM that includes the implementation for BCR and BSL.

The DSSM Commands for the AM13E230x device are detailed in Section 34.1.2.4.2 .

7.2 Memory Map

Table 7-1 details the high-level overview of different Flash, Data, Engineering, and TRIM Sectors and their associated protections.

Table 7-1. ROM Memory Map

Memory Usage	Memory Size	Start Address	Permission	Purpose
ROM	48KB	TI reserved	Immutable	BCR & BSL code
BANK0 NM0	2KB	0x60100000	Immutable and BCR has Write/Erase/Read permission	User keys
BANK0 NM1	2KB	0x60100800	Reprogrammable	BCR & BSL configuration
BANK1 NM0	2KB	0x60101000	BCR has Write/Erase/Read permission	Unused by ROM
BANK1 NM1	2KB	0x60101800	Reprogrammable	BCR & BSL configuration (Duplicate of BANK0 NM1)

7.3 Boot Configuration Routine (BCR)

The boot configuration routine is the first firmware to run on the device after a BOOTRST. The BCR manages the following at boot time:

- Configuring the debug interface security policy

- Optionally executing a mass erase
- Optionally executing a factory reset
- Configuring the flash memory static write protection policy
- Optionally verifying the integrity of some or all of the application firmware (with a 32-bit CRC)
- Optionally starting the bootstrap loader (BSL)

7.3.1 SWD Mass Erase and Factory Reset Commands

The BCR provides mass erase and factory reset functionality through commands sent to the device over SWD from a debug probe using the debug subsystem mailbox (DSSM). When the device is not configured for SWD disabled, the mass erase and factory reset commands can be individually configured to be enabled with a unique 128-bit password, or disabled. By default, both commands are enabled.

The SWD mass erase and factory reset DSSM commands supersede any static write protection policies. For example, if SWD factory reset is configured to be enabled or enabled with password, the BCR configuration data can be reset even if it is statically write protected.

SWD Mass Erase

A SWD mass erase is an erase of the MAIN flash regions only, which typically includes the user application. The BCR and BSL policies stored in the NONMAIN flash region are not affected by a mass erase. A mass erase is useful for erasing all application code and data while leaving the device configuration itself intact.

To set the mass erase command mode and password, configure the BOOTCFG3.MASSERASECMDACCESS field and the PWDMASSERASE password fields in the NONMAIN memory.

SWD Factory Reset

A SWD factory reset is an erase of the MAIN flash regions followed by a reset of the NONMAIN flash region to default values. Such an erase is useful for completely resetting the BCR and BSL device boot policies while also erasing the application code and data.

To set the factory reset command mode and password, configure the BOOTCFG3.FACTORYRESETCMDACCESS field and the PWDFACTORYRESET password fields in the NONMAIN memory.

7.3.2 Fast Boot

The execution time of the BCR can be reduced by enabling fast boot mode. Fast boot mode, when enabled, speeds up the boot process using the following methods:

- Limiting the BSL entry methods. When fast boot mode is enabled, only the SYSCTL register invoke method and the DSSM invoke method can be used to enter the bootloader. The other BSL invoke conditions are not tested (for example, pin based invoke).
- Bypassing the application CRC check (even if the application CRC check is enabled).

To enable fast boot mode, set the fast boot mode to enabled in the BOOTCFG2.FASTBOOTMODE field in the configuration memory.

7.4 Bootstrap Loader (BSL)

The bootstrap loader (BSL) provides a method to program and/or verify the device memory through a standard UART, I2C, or MCAN interface. Key features of the BSL which are accessible through the serial interface include:

- Programming and erase of flash memory
- Ability to return a firmware version number through a pointer to the main flash
- Ability to specify a hardware invoke GPIO
- Ability to enable code/data read-out (disabled by default)
- Ability to return a 32-bit CRC of a code/data region (2KB minimum region size) to verify programming
- Access is always protected with a 256-bit password

- Configurable security alert handling for resisting brute force attacks
- Configurable UART, I2C and MCAN pins

For a complete description of the BSL features, see the *AM13E230x Bootstrap Loader User's Guide*.

The BSL can be enabled or disabled by setting the bootloader mode to enabled or disabled in the BOOTCFG2.BSLMODE field in the NONMAIN flash memory. When the BSL is disabled, the bootloader cannot be entered through any invocation mechanism and the device will boot from the application code in the flash.

7.4.1 Application Version

The BSL supports returning an application version number through the BSL serial interface. This allows the BSL host to interrogate the firmware version without being able to read the firmware. The version field is 32 bits in length. To link the application version command to a version number programmed in the main flash, program the address of the version number in the APP_REV_POINTER field in configuration memory. The version data is only returned if the address specified in APP_REV_POINTER corresponds to a valid flash memory address.

7.4.2 GPIO Invoke

The bootloader supports hardware invoking after a BOOTRST through the use of a GPIO. The BSL configuration in the configuration memory contains the pad, pin, and polarity definition for the GPIO invocation. Devices come configured from TI for a specific GPIO and polarity, but software can change this default by modifying the GPIO pin configuration in the BSL configuration in configuration memory.

To specify the polarity of the BSL_invoke pin, configure the BSLCONFIG0.BSLIVK_LVL field in the configuration memory.

To specify the device pin to be used for BSL_invoke, configure the following fields in the configuration memory:

- Store the IOMUX PINCMx index into the BSLCONFIG0.BSLIVK_PAD_NUM field
- Store the GPIO port (A/B/C/D) into the BSLCONFIG0.BSLIVK_GPIOPORT field
- Store the GPIO pin (0-31) into the BSLCONFIG0.BSLIVK_GPIOPIN field

See the device specific data sheet to determine the default BSL invoke GPIO.

7.4.3 BSL Triggered Mass Erase and Factory Reset

It is possible to send a mass erase or factory reset command to the BSL. The commands work in a similar way as the SWD mass erase and factory reset commands, with several key exceptions.

BSL Mass Erase

A mass erase command sent to the BSL will erase the MAIN flash memory.
Any MAIN flash memory sectors which are configured to be statically write protected (via the BANK0_WRITE_ERASE_PROTECTION_A, BANK0_WRITE_ERASE_PROTECTION_B, BANK1_WRITE_ERASE_PROTECTION_A and BANK1_WRITE_ERASE_PROTECTION_B fields in the configuration memory) will not be erased. The device configuration memory is never erased by a mass erase.

BSL Factory Reset

A factory reset command sent to the BSL will first perform a BSL mass erase to erase the main flash memory (excluding any sectors which are statically write protected). Then, it will additionally erase the device configuration memory.

A BSL factory reset command is only accepted if the following are true:

1. The configuration memory itself is not currently configured to be statically write protected (BANK0_NM_USER_CONFIG field in the configuration memory is set to not protected)
2. The factory reset command is not configured to be disabled (BOOTCFG4.FACTORY_RESET_MODE field in the configuration memory is not set to disabled)

ADVANCE INFORMATION

The BSL host must program valid configuration data back into the configuration memory (via BSL commands) before terminating the BSL session, or the device may enter an unrecoverable state.

Note

If the configuration memory is left unprogrammed after a BSL factory reset, the device will assume a maximally restrictive state on the next reset cycle, any application code in MAIN flash will not be started, and it will not be possible to access the device via any means. To prevent lockout, always ensure that valid configuration data is programmed into configuration memory following a BSL factory reset.

7.5 Lifecycle Management

AM13E230x devices support lifecycle management to manage the device process from the TI factory to customer development, to production, to end-of-life. The Lifecycle Management controls which features of the device are enabled at each stage of the product's life.

Lifecycle management allows customers to increase the level of security from open to completely closed in a controlled manner, and prevent on-device information from getting leaked by converting the device to a specific state when the device is at its end-of-life or in failure analysis.

The device ROM code supports the AM13E230x lifecycle with two states that represent the operation state of the device:

- HS (High Security) - fully operational unit
  - The HS lifecycle state is broken down into multiple **Device Sub-Type** states that represent the specific configuration of the operational device.
- BAD/DCOM - decomissioned or faulty unit

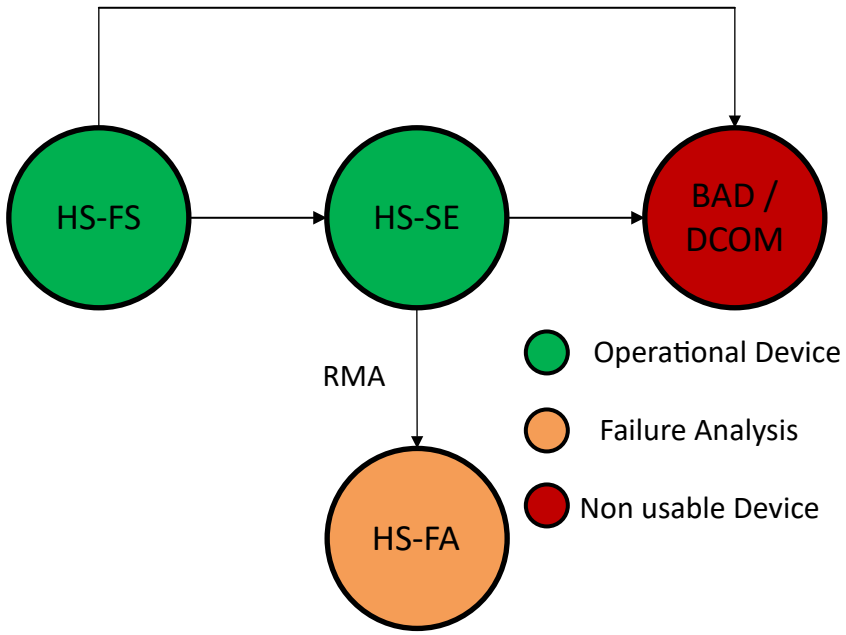


Figure 7-1. AM13E230x Lifecycle Transitions

7.5.1 Device Sub-Type

HS (High Security) lifecycle state is divided into the following **Device Sub-Type** states. The bootcode will read the corresponding security sector to determine the device sub-types for High Security.

- HS-FS (High Security-Field Securable): The devices are shipped from TI in this state. In the HS-FS state, the device can be used with or without security features. If security features are not required, then the device will remain in this state for the rest of the device's lifecycle (**unless the device enters a BAD state or a failure analysis is performed)**.
- HS-SE (High Security-Security Enforced): This is the state the device moves to once the signed CSC (Customer Secure Code) or application has been programmed and verified by Secure Boot.

In addition to the two above Operational Device states, the device enters HS-FA (High Security-Failure Analysis) state when failure analysis is enabled on the device.

7.5.2 Lifecycle Transitions

During the life of the device, the state is changed depending on the current state of the device and the security settings applied.

The following sequence transitions a device from HS-FS to HS-SE:

1. If debug port is not locked, application/CSC can be programmed via debug port
2. Alternatively, invoke the BSL to program the application over supported BSL interfaces (UART/I2C/MCAN)
3. Program the signed CSC or application over BSL
4. Once ROM validates the CSC or application, the device is marked as HS-SE

The following sequence transitions a device from HS-SE to HS-FA:

1. DSSM command (DSSM_COMMAND_RMA) shall be given to the device by the user
2. If Combined Certificate is received and validated, then ROM marks the device as HS-FA
3. Device will go into HS-FA Authenticated flow

To convert a device to DCOM/BAD, the user can send the DSSM command DSSM_COMMAND_CONVERT_BAD_DEVICE.

Note

Customer Return Flow

Customers need to transition the device to HS-FA state from HS-SE before sending the device back to TI for failure analysis. Customer has the option to secure the different assets (e.g. code, keys, etc.) by programming / erasing them. Customer should be aware that erasing the content will restrict / limit the debug of some of the failures (e.g. flash data retention failure cannot be debugged once the content is erased).

7.6 Boot and Startup Sequence

Figure 7-2 illustrates the boot and startup sequence in security enabled applications. At BOOTRST, TI Bootcode starts executing. According to the device's lifecycle status, the device will execute the corresponding process as regular boot, secure boot, or retest flow.

The Bootloader (BSL) can be invoked by Bootcode when a valid bootloader invocation condition is detected. The Bootloader provides a method to program and verify the device Flash memory through a standard serial interface (see Section 7.4).

During the Bootcode execution, the Debug Subsystem Mailbox (DSSM) enables a debug probe to pass messages to the target device through the Serial Wire Debug (SWD) interface, and for the target device to return data to the debug probe. In Bootcode, the DSSM helps with the transmission of commands to the device during boot, including lifecycle state change requests, authenticating the debug probe for password-protected debug, mass erase, and factory reset operations.

After a successful boot, the Bootcode issues BOOTDONE. At this point in the sequence, SYSCTL issues a SYSRST to the device to trigger execution from flash memory. Depending on the boot configuration record, this leads to either the start of the main application (if CSC does **not** exist in this configuration) or to the start of the CSC (if CSC is configured).

CSC (Customer Secure Code) executes from flash memory to further configure security elements. Note that CSC is **customer-owned** secure software. This two-step secure boot concept enables application-specific policies to be implemented and managed by CSC while the standard security enablers are implemented by Bootcode. When CSC has completed execution, the system undergoes a second SYSRST followed by the start of the main aplication.

In this security model, both TI Bootcode and CSC are considered *trusted*. The main application is considered *untrusted*. When execution jumps to the main application, all security elements are expected to have been configured, including restrictions on the debug port, memory accesses, and key management.

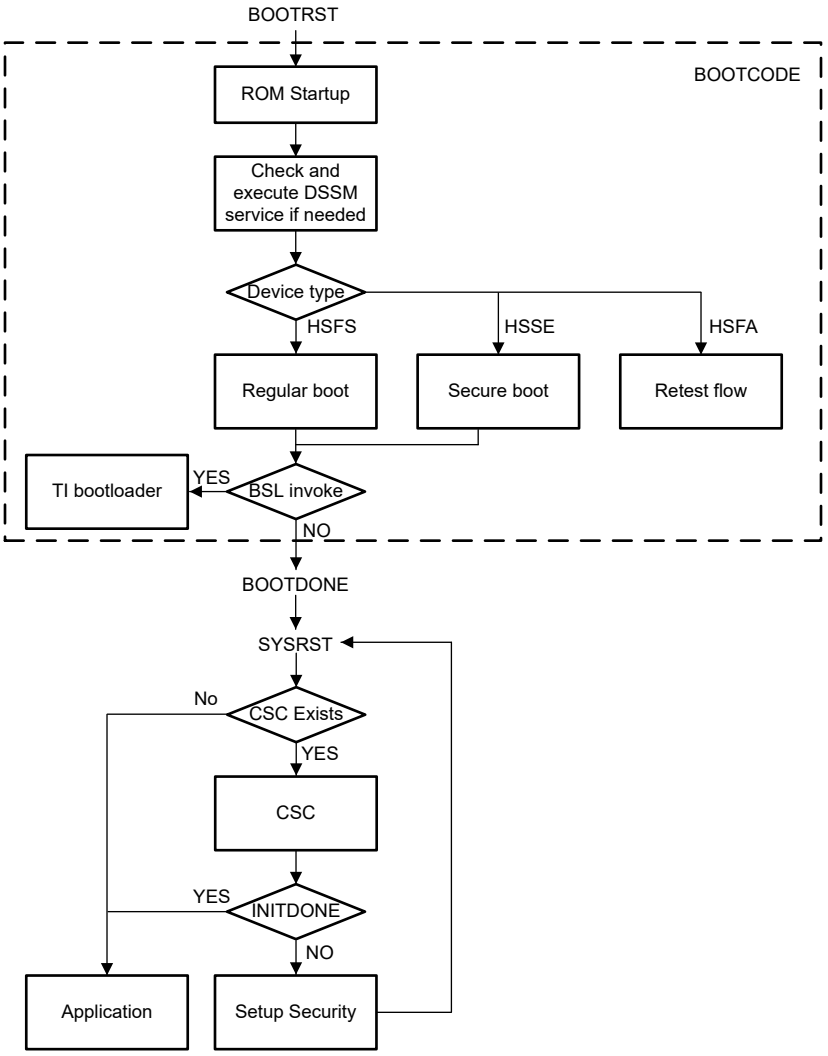


Figure 7-2. AM13E230x Secure Boot and Startup Sequence

Note

The AM13E230x security architecture provides a clean separation between policies and mechanisms. It is possible to build applications where no CSC is implemented and the main application begins immediately following BOOTRST.

The secure exection flow is the path where CSC_EXISTS = YES. In this case, it may be observed that after BOOTRST, two SYSRSTs will be issues before the main application is launched.

After the first SYSRST, the CSC executes. It configures security and issues INITDONE = NO. At this point, the security configuration is locked and enforced. A second SYSRST is issued, and restarts CSC execution. At the second SYSRST, INITDONE = YES and the main application is launched.

If CSC is not enabled, then only one SYSRST is issued and the application launches following the reset.

7.6.1 Secure Boot

AM13E230x devices support an immutable secure boot in ROM leveraging the device's lifecycle management. The immutable secure boot establishes a chain of trust that allows the execution of authenticated and authorized firmware on the device, utilizing the device's lifecycle management transitions: from a provisioning state (HS-FS) to a permanently secured operational state (HS-SE).

Secure boot provides mechanisms to validate image integrity through 128-bit CMAC, and verify key integrity via BCH encoding. ROM secure boot supports the following features:

- Verification of the integrity of the firmware through 128-bit CMAC
- Anti-rollback to prevent downgrade to older firmware that can be a potentail security vulberibility
- Key revocation to decomission compromised keys - support for up to 4 keys
- TI proprietary MCUBoot trailer processing for security metadata

AM13E230x devices support up to 4 keys. Secure boot uses the first key for verification. The rest of the keys are utilized if the first key is revoked.

Secure boot is performed only when the device is in HS-SE state. In HS-SE state, secure boot is performed during one of the following conditions:

- BOOTRST
- Power-on reset
- Wake up from Shutdown

7.6.2 Customer Secure Code (CSC)

If the user wants to add additional features or functionality to the immutable secure boot on AM13E230x devices, then a Customer Secure Code (CSC) (also referred to as the secondary bootloader) can be implemented in the MAIN flash of the device. The ROM provides the following features to support the CSC:

- Blank swap decision
- Initialization of the secure key store

When the CSC issues INITDONE (by writing to SYSCTL.SECCFG.INITDONE MMR) then SYSCTL issues the second SYSRST (as outlined in Section 7.6). The device again starts execution from 0x0 mapped to flash, and the CSC executes a second time. After the second CSC execution, the CSC will find that INITDONE has already been issued previously (from after the first CSC execution), determined by reading the SYSCTL.SECCFG.SECSTATUS.INITDONE bit. The main application is then called.

The customer owned secure code can leverage the Global Security Controller (GSC) to implement additional security capabilities such as:

- Flash read-execute protection
- Flash IP protection
- SRAM write-execute mutual exclusion

The CSC specifies which bank holds the more recent authenticated application image. If that bank is physical bank 0 (same bank as where CSC is executing from), then bank 1 is read-write only and does not have execute privilege. If the correct application image is determined to be on physical bank 1, then the CSC must issue a bank-swap request.

The CSC indicates the end of CSC execution by writing to the SYSCTL.SECCFG.INITDONE register with a PASS value (0x1) along with a KEY value of 0x9d. Successfully writing to the INITDONE register results in a second SYSRST operation during which the bank-swap takes effect.

Chapter 8

Global Security Controller (GSC)

The Global Security Controller (GSC) peripheral is a system module that can configure the security firewalls for privileged and non-privileged context information to access on-chip memory and peripherals. After configuration, the GSC detects out of context transactions to the Flash, SRAM and peripherals, and generates interrupts when the privileged context is violated.

ADVANCE INFORMATION

8.1 GSC Introduction

The Global Security Controller (GSC) configures the privilege attribute for resources on the device. The GSC consists of multiple blocks that verifies the device's access properties are observed the same by all bus initiators. In essence, the GSC provides firewall access based on the priviliege property of the application code and initiators.

8.1.1 GSC Features

- Configures privileged and unprivileged attribute for memory and peripherals
- Write and erase protection for the Flash
- Direct monitoring of initiator access to memory and peripherals to generate interrupts for context violations
- Logs violation information in the EAM for the initiator and the destination address
- Hide protection control for the Flash in 2 KB sector size granularity

8.2 GSC Operation

The GSC comprises of three important sub-blocks: the Peripheral Protection Controller (PPC), the SRAM Protection Controller (SPC) and the Flash Protection Controller (FPC). A block diagram of the GSC is shown in *GSC Functional Block Diagram*.

8.2.1 Functional Block Diagram

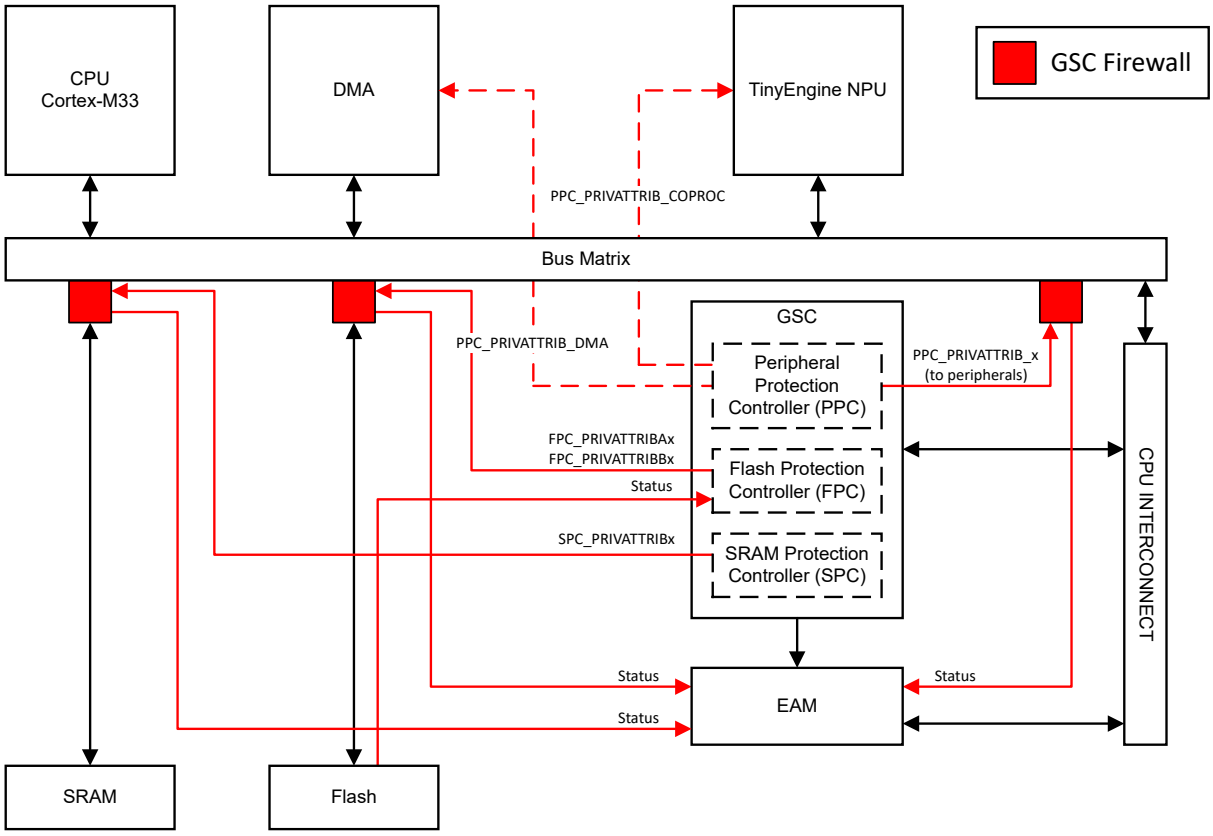


Figure 8-1. GSC Functional Block Diagram

8.2.2 Peripheral Protection Controller

The Peripheral Protection Controller (PPC) configures the firewall for peripheral access. The PPC_PRIVATTRIB_x register for a given peripheral configures the firewall for privilege or non-privilege access. Please refer to Table 8-1 for the peripheral to PPC_PRIVATTRIB register mapping.

Table 8-1. PPC Priviliged Attribute Register Mapping

Peripheral	PPC Register
ADC	PPC_PRIVATTRIB_ADC
PWM, ECAP, EQEP, TIMG4, TIMG12	PPC_PRIVATTRIB_TIMER
UNICOMM	PPC_PRIVATTRIB_UNICOMM
GPIO	PPC_PRIVATTRIB_GPIO
MCAN	PPC_PRIVATTRIB_MCAN
CMPSS	PPC_PRIVATTRIB_CMPSS
WWDT	PPC_PRIVATTRIB_WDT
CRC	PPC_PRIVATTRIB_CRC
AES	PPC_PRIVATTRIB_AES
IOMUX	PPC_PRIVATTRIB_IOMUX
DEBUGSS	PPC_PRIVATTRIB_DEBUGSS
KEYSTORECTL	PPC_PRIVATTRIB_KEYSTORECTL
EAM	PPC_PRIVATTRIB_EAM
WUC	PPC_PRIVATTRIB_WUC
FLASH	PPC_PRIVATTRIB_FRIREGS
MEMCFG	PPC_PRIVATTRIB_MEMCFG
SYSCTL	PPC_PRIVATTRIB_SYSCTL
DMA	PPC_PRIVATTRIB_DMA
GSC	PPC_PRIVATTRIB_GSC
TINIE NPU	PPC_PRIVATTRIB_COPROC
EPI	PPC_PRIVATTRIB_EPI
PGA	PPC_PRIVATTRIB_PGA
XBAR	PPC_PRIVATTRIB_XBAR

Note

The CPU thread privilege mode is configured by the CONTROL register. Please refer to Section 4.2.2 for more details.

When a peripheral is configured as a privileged peripheral then only an initiator which is operating in the privileged context can access the peripheral configuration, status or data registers. Any initiator access not in the privilege context will generate a Non-Maskable Interrupt (NMI) that is logged in the EAM module.

In addition to generating a NMI, any write access that shall be ignored and a read access shall return all zeroes to the corresponding initiator. This prevents unpriviliged code or initiator from accessing or modifying data in the peripheral space at run time.

Certain peripheral firewall are configured with the privilege attribute enabled (CRC, DEBUGSS, SYSCTL, DMA, and GSC). This allows the application code to leverage a default privileged setting of the device to implement bootloaders or authentication code.

8.2.2.1 DMA Security

The DMA is uniquely handled within the GSC framework as it is both an initiator and target. When the DMA privilege attribute is configured (PPC.PPC_PRIVATTRIB_DMA), then both the initiator and the target port for configuration/status are set to privileged.

The power-on reset value for the DMA is always privileged. Only the CPU in privileged mode can configure the DMA0 control and channel registers. Once configured, all write and read access generated by DMA0 initiator ports appear as privileged to the source and destination peripherals/memories. This ensures that after the device is powered on and the CPU begins code execution, the DMA0 is in a privileged context for moving data between peripherals and memory and does not require any additional configuration.

Note

The GSC is also privileged at power-on reset. Any non-privileged application code cannot directly modify the DMA property. This must be handled within privileged application code.

8.2.2.2 TinyEngine NPU Security

The TinyEngine NPU is also handled differently within the GSC framework as it is both an initiator and target. When the NPU's privilege attribute is configured (PPC_PRIVATTRIB_COPROC.PRIV_TINIE), then both the initiator and the target port are set to privileged.

Once configured, all transactions generated by NPU's initiator ports appear as privileged to the SRAM. Only the CPU in privileged mode can configure the NPU.

8.2.3 SRAM Protection Controller

The SRAM Protection Controller (SPC) configures the firewall for the SRAM and monitors the transactions from the initiators. Any violation of the security property set for the SRAM results in a NMI being triggered and the violation logged in the EAM. Similar to the PPC sub-block, the SPC provides the SPC_PRIVATTRIB[x] register to configure the firewall for privilege or non-privilege access.

To manage the SRAM firewall efficiently, the SRAM is partitioned into 16 KB pages. Each SPC_PRIVATTRIB[x] register controls four 16 KB pages. Further granularity is provided by partitioning each page into variable-sized chunks. These chunks are represented within the page control field of the SPC_PRIVATTRIB[x] register in sets of 8-bits as shown in Figure 8-2.

As mentioned above, each page is comprised of seven chunks of variable size. As a result, out of the 8-bits available for a page, only 7-bits are used with the most significant bit being reserved. The chunks in a SRAM page are comprised of the following sizes:

- 3 chunks of 4 KB
- 1 chunk of 2 KB
- 1 chunk of 1 KB
- 2 chunks of 512 B

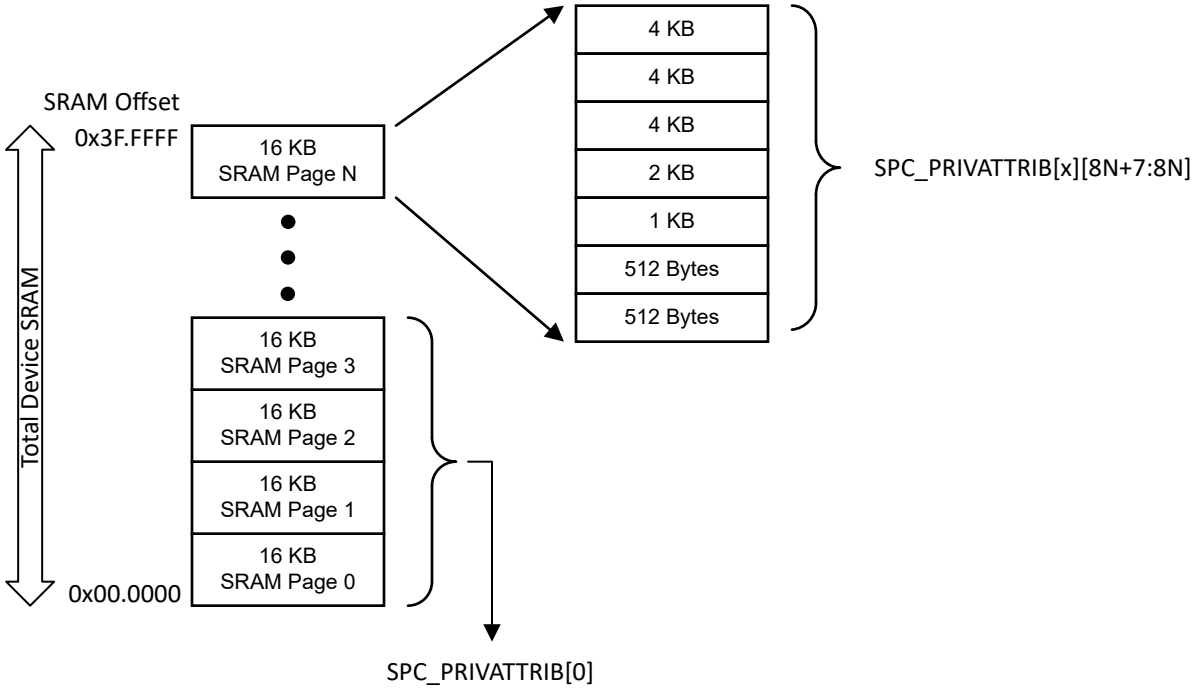


Figure 8-2. SRAM Page and Chunk Mapping

8.2.3.1 SRAM Page Use Model

Each chunk within a 16 KB SRAM page can be configured independently with the privilege attributes to maxmize SRAM usage for different security aspects of the application code.

For example, in Figure 8-3, each chunk within the 16 KB SRAM page is provisioned such that:

- 9 KB of SRAM can be accessed by a unprivileged initiator
- 7 KB of SRAM can be accessed by a privileged initiator

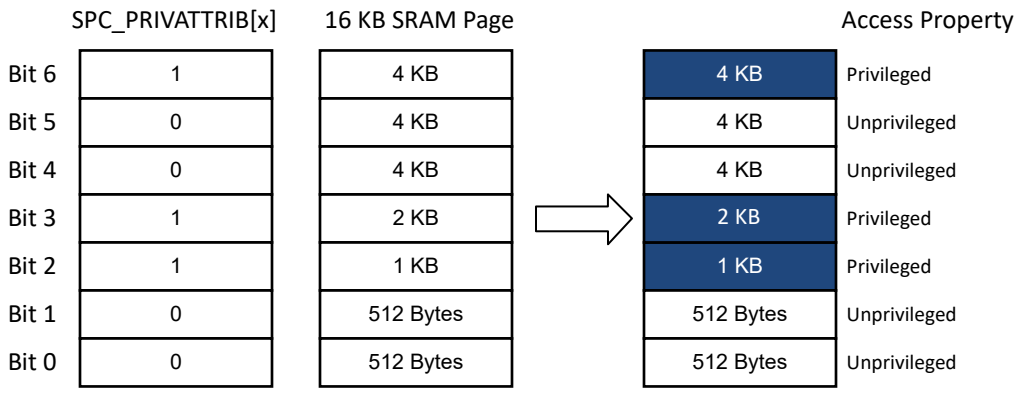


Figure 8-3. SRAM Page Use Example

ADVANCE INFORMATION

The SPC allows that privileged regions within the SRAM do not need to be continguous. Additionally, by default the SPC is configurable by a privileged initiator, which allows the privileged context to dynamically allocate memory properties. This is useful when SRAM has to be allocated for a critical application code from a common memory pool.

Note

Before changing the SRAM chunk memory property, it is the privileged applications responsibility to ensure that all initiators are properly configured and any data in SRAM is zero initialized.

8.2.4 Flash Protection Controller

The Flash Protection Controller (FPC) configures the firewall for the program and data flash, and monitors the transactions from the initiators. Any violation of the security property set for the flash results in a NMI being triggered and the violation logged in the EAM.

Similar to the PPC and SPC sub-blocks, the FPC provides the flash privilege access firewall registers FPC_PRIVATTRIBA[x] for the first 32 sectors of each main flash bank and FPC_PRIVATTRIBB[x] for the remaining sectors in each flash bank.

In addition, the FPC provides flash write/erase protection registers FPC_WEPROTA[x] and FPC_WEPROTB[x]. The FPC_WEPROTA[x] register provides write/erase protection for the first 32 sectors in each of the MAIN flash bank at 2 KB sector size. The FPC_WEPROTB[x] register provides write/erase protection for remaining flash sectors with each bit mapping to 8 sectors of 2 KB size (for a total of 16 KB size).

Note

Given AM13E230x devices have two main flash banks, security property registers with the suffix '0' control flash bank 0 and suffix '1' control flash bank 1. For example, FPC_WEPROTA0 controls bank 0 and FPC_WEPROTA1 controls bank 1.

To manage the flash firewalls efficiently, the FPC uses the sector information for each flash bank. Each flash bank consists of multiple 2 KB flash sectors. An illustration of the 512 KB flash on AM13E230x devices is shown in Figure 8-4.

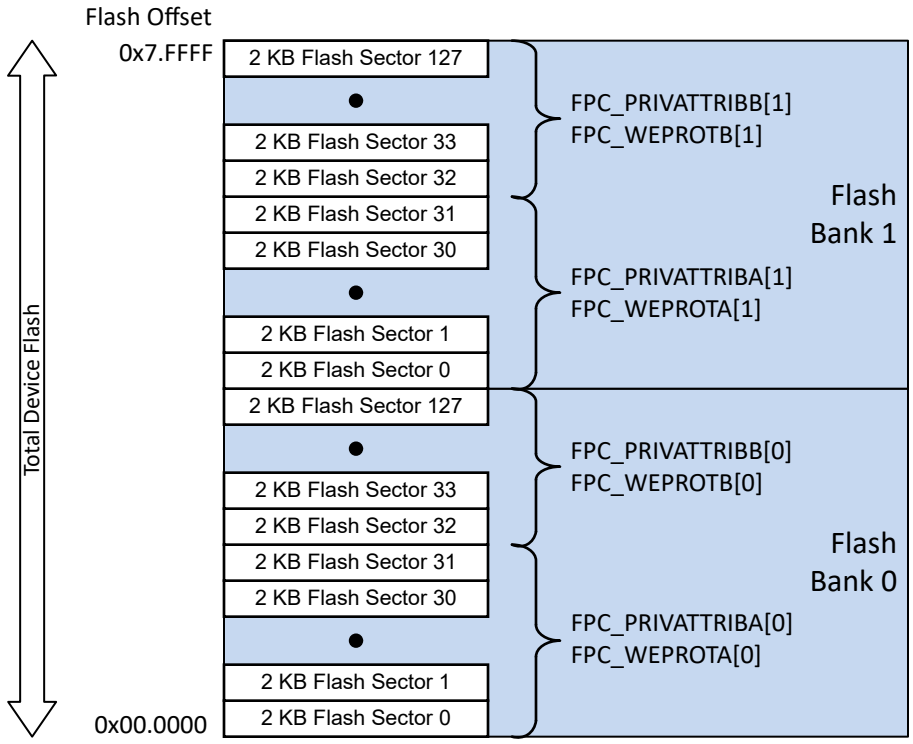


Figure 8-4. Flash Sector Mapping

8.2.4.1 Flash Bank Security Implementation

As mentioned earlier, the firewall control for the flash banks is at a 2KB sector level. To maintain maximum flexibility and reasonable ease of use, each privilege attribute register configures firewalls slightly differently:

- **FPC_PRIVATTRIBA[x], FPC_WEPROTA[x]**: Each bit is mapped to individual 2KB sector, starting at sector 0 up to sector 31. All 32-bits of the register are used for the mapping with bit 0 mapped to sector 0 and bit 31 mapped to sector 31.
- **FPC_PRIVATTRIBB[x], FPC_WEPROTB[x]**: Each bit is mapped to a set of eight sectors (total of 16 KB), starting at sector 32 up to sector 127.
  - With a 512 KB flash size, there are 96 sectors left after removing the first 32 sectors from each flash bank, so these registers use 12 bits. The first 4-bits of these registers are reserved. Bit 4 is mapped to sectors 32 to 39 and bit 15 is mapped to sector 120 to sector 127.

Note

The "x" in the above register names must be replaced by "0" for bank 0 and "1" for bank 1, for the dual-bank configuration on AM13E230x devices.

The Table 8-2 shows some example of the register mapping for 1 MB flash.

Table 8-2. FPC Firewall Register Decoding for 512 KB Flash

Flash Privilege Access Register	Flash Write-Erase Protect Register	Flash Sector (2KB)
FPC_PRIVATTRIBAx[0]	FPC_WEPROTAx[0]	0
FPC_PRIVATTRIBAx[1]	FPC_WEPROTAx[1]	1
FPC_PRIVATTRIBAx[2]	FPC_WEPROTAx[2]	2
FPC_PRIVATTRIBAx[3]	FPC_WEPROTAx[3]	3

ADVANCE INFORMATION

Table 8-2. FPC Firewall Register Decoding for 512 KB Flash (continued)

Flash Privilege Access Register	Flash Write-Erase Protect Register	Flash Sector (2KB)
FPC_PRIVATTRIBAx[4]	FPC_WEPROTAx[4]	4
FPC_PRIVATTRIBAx[5]	FPC_WEPROTAx[5]	5
FPC_PRIVATTRIBAx[6]	FPC_WEPROTAx[6]	6
FPC_PRIVATTRIBAx[7]	FPC_WEPROTAx[7]	7
FPC_PRIVATTRIBAx[8]	FPC_WEPROTAx[8]	8
FPC_PRIVATTRIBAx[9]	FPC_WEPROTAx[9]	9
FPC_PRIVATTRIBAx[10]	FPC_WEPROTAx[10]	10
FPC_PRIVATTRIBAx[11]	FPC_WEPROTAx[11]	11
FPC_PRIVATTRIBAx[12]	FPC_WEPROTAx[12]	12
FPC_PRIVATTRIBAx[13]	FPC_WEPROTAx[13]	13
FPC_PRIVATTRIBAx[14]	FPC_WEPROTAx[14]	14
FPC_PRIVATTRIBAx[15]	FPC_WEPROTAx[15]	15
FPC_PRIVATTRIBAx[16]	FPC_WEPROTAx[16]	16
FPC_PRIVATTRIBAx[17]	FPC_WEPROTAx[17]	17
FPC_PRIVATTRIBAx[18]	FPC_WEPROTAx[18]	18
FPC_PRIVATTRIBAx[19]	FPC_WEPROTAx[19]	19
FPC_PRIVATTRIBAx[20]	FPC_WEPROTAx[20]	20
FPC_PRIVATTRIBAx[21]	FPC_WEPROTAx[21]	21
FPC_PRIVATTRIBAx[22]	FPC_WEPROTAx[22]	22
FPC_PRIVATTRIBAx[23]	FPC_WEPROTAx[23]	23
FPC_PRIVATTRIBAx[24]	FPC_WEPROTAx[24]	24
FPC_PRIVATTRIBAx[25]	FPC_WEPROTAx[25]	25
FPC_PRIVATTRIBAx[26]	FPC_WEPROTAx[26]	26
FPC_PRIVATTRIBAx[27]	FPC_WEPROTAx[27]	27
FPC_PRIVATTRIBAx[28]	FPC_WEPROTAx[28]	28
FPC_PRIVATTRIBAx[29]	FPC_WEPROTAx[29]	29
FPC_PRIVATTRIBAx[30]	FPC_WEPROTAx[30]	30
FPC_PRIVATTRIBAx[31]	FPC_WEPROTAx[31]	31
FPC_PRIVATTRIBBx[4]	FPC_WEPROTBx[4]	32 - 39
FPC_PRIVATTRIBBx[5]	FPC_WEPROTBx[5]	40 - 47
FPC_PRIVATTRIBBx[6]	FPC_WEPROTBx[6]	48 - 55
FPC_PRIVATTRIBBx[7]	FPC_WEPROTBx[7]	56 - 63
FPC_PRIVATTRIBBx[8]	FPC_WEPROTBx[8]	64 - 71
FPC_PRIVATTRIBBx[9]	FPC_WEPROTBx[9]	72 - 79
FPC_PRIVATTRIBBx[10]	FPC_WEPROTBx[10]	80 - 87
FPC_PRIVATTRIBBx[11]	FPC_WEPROTBx[11]	88 - 95
FPC_PRIVATTRIBBx[12]	FPC_WEPROTBx[12]	96 - 103

Table 8-2. FPC Firewall Register Decoding for 512 KB Flash (continued)

Flash Privilege Access Register	Flash Write-Erase Protect Register	Flash Sector (2KB)
FPC_PRIVATTRIBBx[13]	FPC_WEPROTBx[13]	104 - 111
FPC_PRIVATTRIBBx[14]	FPC_WEPROTBx[14]	112 - 119
FPC_PRIVATTRIBBx[15]	FPC_WEPROTBx[15]	120 - 127

8.2.4.2 Flash Hide Protection

The FPC is also responsible for managing the Hide Protection (HDP) function. The HDP function allows an application to execute a section of code one time, and prevent subsequent read or execute access to the code. As an example, this is useful feature especially when running a second stage authentication code allowing main application to be validated before execution and prevent access to any stored symmetric or private keys.

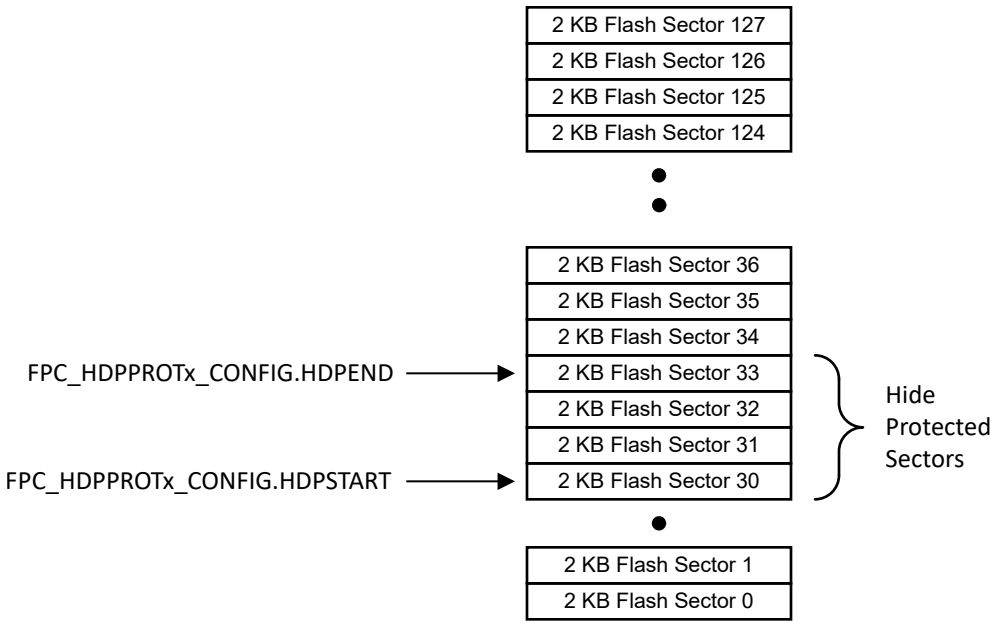


Figure 8-5. Flash Hide Protection Memory View

As shown in Figure 8-5, the HDPSTART and HDPEND fields in the FPC_HDPPROTx_CONFIG register define the start and end sector address which are to be hide protected.

Note

The "x" in the FPC_HDPPROTx_CONFIG must be replace by "0" for flash bank 0 and "1" for flash bank 1, for the dual-bank configuration on AM13E230x devices. In other words, each flash bank has a corresponding hide protection configuration register.

The Table 8-3 shows the valid configuration of values that define the sectors that can be hide protected.

Table 8-3. HDPSTART and HDPEND Rule

RULE	COMMENT
HDPSTART = HDPEND	Hide protection for exactly 1 sector identified by HDPSTART
HDPSTART < HDPEND	Hide protection for sectors from HDPSTART to HDPEND (inclusive)

ADVANCE INFORMATION

Table 8-3. HDPSTART and HDPEND Rule (continued)

RULE	COMMENT
HDPSTART > HDPEND	Invalid configuration defined, no hide protection

In addition to start and end sector address for HDP, there are two additional registers. The FPC_HDPPROT_CONTROL is used to validate the corresponding configuration register to be active. The FPC_HDPEN_CONTROL is to trigger the mechanism for hide protection. As shown in Figure 8-6, once the start sector, end sector and valid control bits are configured correctly, the HDP is ready. When CPU executes the HDP region code, the HDP is armed. Any function call or jump outside of the HDP automatically enables the mechanism, preventing the application code from re-entering the HDP region. An access, execute or read, generates an NMI.

Note

When executing code in HDP region followed by branch to SRAM, the SRAM code can access HDP region. Once all SRAM code execution is completed, the SRAM code must branch back and exit the HDP region to continue execution in flash (out of the HDP region) for the HDP mechanism to enable. If application does not want SRAM to access HDP region, then the code must implement a trampoline function to enable HDP before branching to SRAM code.

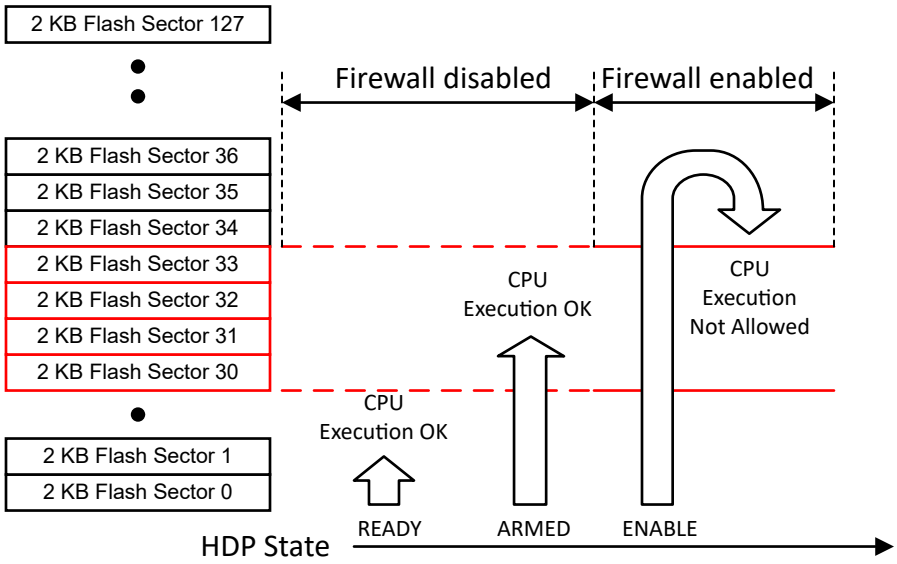


Figure 8-6. HDP Execution Flow

All the HDP registers are writeable once and can be set by the application code. A system reset does not allow the application to re-configure, therefore a BOOTRST is required.

8.2.5 Strict Privilege Context Protection

The firewall configurations in GSC prevents unprivileged initiators from accessing privileged peripherals and memory. However, an privileged initiator could still access unprivileged peripherals and memory. The attribute violation configuration registers (xPC_ATTRIBVIOL_CONFIG) for PPC, SPC and FPC can be configured to prevent the latter case.

Table 8-4. Privilege Attribute Violation Configuration Registers

Sub-Module	Attribute Violation Configuration Registers
PPC	PPC_ATTRIBVIOLP_CONFIG

Table 8-4. Privilege Attribute Violation Configuration Registers (continued)

Sub-Module	Attribute Violation Configuration Registers
SPC	SPC_ATTRIBVIOLP_CONFIG
FPC	FPC_ATTRIBVIOLP_CONFIG

These registers are typically used when an application wants to avoid memory overflow or accidental peripheral access outside of the context. As shown in Table 8-5, on a violating access, the GSC generates a NMI for the CPU. A violating write has no effect and a read will return all zero.

Table 8-5. Attribute Violation Rule Table

xPC_ATTRIBVIOL_CONFIG	Bus Transaction	Privileged Target	Unprivileged Target
PRIVVIOL = 0	Privileged Initiator	Allowed	Allowed
	Unprivileged Initator	NMI	Allowed
PRIVVIOL = 1	Privileged Initiator	Allowed	NMI
	Unprivileged Initator	NMI	Allowed

Note

Careful design of the application (with an NMI handler) must be taken to avoid a lock-up scenario.

8.2.6 GSC Configuration Lock

The GSC also has the capability of locking and committing the security setting on a per region basis using the GSC_LOCK register. There are total of 6 regions which can be individually locked or committed.

- Attribute Violation Configuration
- PPC Attribute
- SPC Attribute
- FPC Attribute
- Vector Table Offset Register (VTOR)
- DMA trigger selection

When locked, the corresponding register in the region cannot be modified. To update a region register, the KEY of 0xA551 has to be written along with the modified register bit to unlock the register.

When commit control is applied using the GSC_COMMIT register, then the GSC_LOCK register cannot be updated even if the correct KEY is used. The only mechanism to update a commit is with a BOOTRST, before the application can make changes.

8.3 GSC Registers

This Section describes the GSC Registers.

8.3.1 GSC Base Address Table

Table 8-6. GSC Base Address Table

Bit Field Name		DriverLib Name	Base Address
Instance	Structure		
GscRegs	GSC_REGS	GSC	0x4004_6000

ADVANCE INFORMATION

8.3.2 GSC_LITE_INTERNAL_REGS Registers

Table 8-7 lists the memory-mapped registers for the GSC_LITE_INTERNAL_REGS registers. All register offset addresses not listed in Table 8-7 should be considered as reserved locations and the register contents should not be modified.

Table 8-7. GSC_LITE_INTERNAL_REGS Registers

Offset	Acronym	Register Name	Write Protection	Section
1004h	SPC_ATTRIBVIOLP_CONFIG	SRAM Privilege Access Attribute Violation Configuration Register		Go
1014h	PPC_ATTRIBVIOLP_CONFIG	Peripheral Privilege Access Attribute Violation Configuration Register		Go
1024h	FPC_ATTRIBVIOLP_CONFIG	Flash Privilege Access Attribute Violation Configuration Register		Go
1040h	FPC_HDPPROT0_CONFIG	Flash Hide Protection for Bank0 Configuration Register		Go
1044h	FPC_HDPPROT1_CONFIG	Flash Hide Protection for Bank1 Configuration Register		Go
104Ch	FPC_HDPPROT_CONTROL	Flash Hide Protection Control Register		Go
1050h	FPC_HDPEN_CONTROL	Flash Hide Protection Enable Control Register		Go
1104h	PPC_PRIVATTRIB_ADC	Peripheral Privilege Access ADC		Go
110Ch	PPC_PRIVATTRIB_TIMER	Peripheral Privilege Access Timer		Go
1114h	PPC_PRIVATTRIB_UNICOMM	Peripheral Privilege Access UniComm		Go
111Ch	PPC_PRIVATTRIB_GPIO	Peripheral Privilege Access GPIO		Go
1124h	PPC_PRIVATTRIB_MCAN	Peripheral Privilege Access CAN		Go
1134h	PPC_PRIVATTRIB_CMPSS	Peripheral Privilege Access CMPSS		Go
113Ch	PPC_PRIVATTRIB_WDT	Peripheral Privilege Access Watchdog		Go
114Ch	PPC_PRIVATTRIB_CRC	Peripheral Privilege Access CRC		Go
1154h	PPC_PRIVATTRIB_AES	Peripheral Privilege Access AES		Go
116Ch	PPC_PRIVATTRIB_IOMUX	Peripheral Privilege Access IOMUX		Go
1184h	PPC_PRIVATTRIB_DEBUGSS	Peripheral Privilege Access DEBUGSS		Go
1194h	PPC_PRIVATTRIB_KEYSTORECTL	Peripheral Privilege Access KEYSTORECTL		Go
119Ch	PPC_PRIVATTRIB_EAM	Peripheral Privilege Access EAM		Go
11ACh	PPC_PRIVATTRIB_FRIREGS	Peripheral Privilege Access FRIREGS		Go
11B4h	PPC_PRIVATTRIB_MEMCFG	Peripheral Privilege Access MEMCFG		Go
120Ch	PPC_PRIVATTRIB_SYSCTL	Peripheral Privilege Access System Control		Go
1214h	PPC_PRIVATTRIB_DMA	Peripheral Privilege Access DMA		Go
121Ch	PPC_PRIVATTRIB_GSC	Peripheral Privilege Access GSC		Go
1224h	PPC_PRIVATTRIB_COPROC	Peripheral Privilege Access Co-processor		Go
122Ch	PPC_PRIVATTRIB_EPI	Peripheral Privilege Access EPI		Go
1234h	PPC_PRIVATTRIB_PGA	Peripheral Privilege Access PGA		Go
123Ch	PPC_PRIVATTRIB_XBAR	Peripheral Privilege Access XBAR		Go
12C0h	SPC_PRIVATTRIB0	SRAM Privilege Attribute-0 Register		Go
12C4h	SPC_PRIVATTRIB1	SRAM Privilege Attribute-1 Register		Go
1400h	FPC_PRIVATTRIBA0	Flash Privilege Attribute Register A0		Go
1404h	FPC_PRIVATTRIBA1	Flash Privilege Attribute Register A1		Go
1408h	FPC_PRIVATTRIBB0	Flash Privilege Attribute Register B0		Go
140Ch	FPC_PRIVATTRIBB1	Flash Privilege Attribute Register B1		Go
1600h	FPC_WEPROTA0	Flash Write Protect Attribute Register A0		Go
1604h	FPC_WEPROTA1	Flash Write Protect Attribute Register A1		Go

Table 8-7. GSC_LITE_INTERNAL_REGS Registers (continued)

Offset	Acronym	Register Name	Write Protection	Section
1608h	FPC_WEPROTB0	Flash Write Protect Attribute Register B0		Go
160Ch	FPC_WEPROTB1	Flash Write Protect Attribute Register B1		Go
1800h	FPC_FLSEMREQ	Flash semaphore request register		Go
1804h	FPC_FLSEMCLR	Flash semaphore release register		Go
1808h	FPC_FLSEMSTAT	Flash semaphore statys registers		Go
1B84h	VTOR_NS	Non - Secure Vector Table Offset Register		Go
1D80h	GSC_LOCK	GSC Lock configuration register		Go
1D84h	GSC_COMMIT	GSC commit configuration register		Go
1FFCh	GSC_REVISION	GSC Revision register		Go

Complex bit access types are encoded to fit into small table cells. Table 8-8 shows the codes that are used for access types in this section.

Table 8-8. GSC_LITE_INTERNAL_REGS Access Type Codes

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

1 SPC_ATTRIBVIOLP_CONFIG Register (Offset = 1004h) [Reset = 00000000h]

SPC_ATTRIBVIOLP_CONFIG is shown in Figure 8-7 and described in Table 8-9.

Return to the Summary Table.

SRAM Privilege Access Attribute Violation Configuration Register

Figure 8-7. SPC_ATTRIBVIOLP_CONFIG Register

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

Table 8-10. SPC_ATTRIBVIOLP_CONFIG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	Reserved
0	PRIVVIOL	R/W	0h	Controls the behavior of privilege access to non-privilege SRAM 0: Access is allowed 1: Access generates bus-fault and NMI Reset type: XRSn

ADVANCE INFORMATION

2 PPC_ATTRIBVIOLP_CONFIG Register (Offset = 1014h) [Reset = 00000000h]

PPC_ATTRIBVIOLP_CONFIG is shown in Figure 8-8 and described in Table 8-10.

Return to the Summary Table.

Peripheral Privilege Access Attribute Violation Configuration Register

Figure 8-8. PPC_ATTRIBVIOLP_CONFIG Register

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

Table 8-12. PPC_ATTRIBVIOLP_CONFIG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	Reserved
0	PRIVVIOL	R/W	0h	Controls the behavior of privilege access to non-privilege peripherals 0: Access is allowed 1: Access generates bus-fault and NMI Reset type: XRSn

3 FPC_ATTRIBVIOLP_CONFIG Register (Offset = 1024h) [Reset = 00000000h]

FPC_ATTRIBVIOLP_CONFIG is shown in Figure 8-9 and described in Table 8-11.

Return to the Summary Table.

Flash Privilege Access Attribute Violation Configuration Register

Figure 8-9. FPC_ATTRIBVIOLP_CONFIG Register

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

Table 8-14. FPC_ATTRIBVIOLP_CONFIG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	Reserved
0	PRIVVIOL	R/W	0h	Controls the behavior of privilege access to non-privilege Flash 0: Access is allowed 1: Access generates bus-fault and NMI Reset type: XRSn

4 FPC_HDPPROT0_CONFIG Register (Offset = 1040h) [Reset = 00FF0000h]

FPC_HDPPROT0_CONFIG is shown in Figure 8-10 and described in Table 8-12.

Return to the Summary Table.

Flash Hide Protection for Bank0 Configuration Register

Figure 8-10. FPC_HDPPROT0_CONFIG Register

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

Table 8-16. FPC_HDPPROT0_CONFIG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	RESERVED	R	0h	Reserved
23-16	HDPSTART	WOnce	FFh	Sector number where hide protection starts Reset type: Rsn
15-8	RESERVED	R	0h	Reserved
7-0	HDPEND	WOnce	0h	Sector number where hide protection ends Reset type: Rsn

5 FPC_HDPPROT1_CONFIG Register (Offset = 1044h) [Reset = 00FF0000h]

FPC_HDPPROT1_CONFIG is shown in Figure 8-11 and described in Table 8-13.

Return to the Summary Table.

Flash Hide Protection for Bank1 Configuration Register

Figure 8-11. FPC_HDPPROT1_CONFIG Register

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

Table 8-18. FPC_HDPPROT1_CONFIG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	RESERVED	R	0h	Reserved
23-16	HDPSTART	WOnce	FFh	Sector number where hide protection starts Reset type: Rsn
15-8	RESERVED	R	0h	Reserved
7-0	HDPEND	WOnce	0h	Sector number where hide protection ends Reset type: Rsn

6 FPC_HDPPROT_CONTROL Register (Offset = 104Ch) [Reset = 00000000h]

FPC_HDPPROT_CONTROL is shown in Figure 8-12 and described in Table 8-14.

Return to the Summary Table.

Flash Hide Protection Control Register

Figure 8-12. FPC_HDPPROT_CONTROL Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED						HDPPROT1ACCDIS	HDPPROT0ACCDIS
R-0h						R/W1S-0h	R/W1S-0h

Table 8-20. FPC_HDPPROT_CONTROL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R	0h	Reserved
1	HDPPROT1ACCDIS	R/W1S	0h	Enables the FPC_HDPPROT1_CONFIG to take effect. 0: HDP region access is enabled 1: HDP region access is disabled Reset type: Rsn
0	HDPPROT0ACCDIS	R/W1S	0h	Enables the FPC_HDPPROT0_CONFIG to take effect. 0: HDP region access is enabled 1: HDP region access is disabled Reset type: Rsn

7 FPC_HDPEN_CONTROL Register (Offset = 1050h) [Reset = 00000000h]

FPC_HDPEN_CONTROL is shown in Figure 8-13 and described in Table 8-15.

Return to the Summary Table.

Flash Hide Protection Enable Control Register

Figure 8-13. FPC_HDPEN_CONTROL Register

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

Table 8-22. FPC_HDPEN_CONTROL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	Reserved
0	HDPEN	R/W1S	0h	HDP mode enable. This bit is set in the OTP during provisioning. 0: HDP mode is not used 1: HDP mode is enabled Reset type: Rsn

ADVANCE INFORMATION

8 PPC_PRIVATTRIB_ADC Register (Offset = 1104h) [Reset = 00000000h]

PPC_PRIVATTRIB_ADC is shown in Figure 8-14 and described in Table 8-16.

Return to the Summary Table.

Peripheral Privilege Access ADC

Figure 8-14. PPC_PRIVATTRIB_ADC Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED					PRIV_ADC2	PRIV_ADC1	PRIV_ADC0
R-0h					R/W-0h	R/W-0h	R/W-0h

Table 8-24. PPC_PRIVATTRIB_ADC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R	0h	Reserved
2	PRIV_ADC2	R/W	0h	Privilege access enabled to ADC2 0: Non-Privilege access is enabled 1: Privilege access is enabled Reset type: XRSn
1	PRIV_ADC1	R/W	0h	Privilege access enabled to ADC1 0: Non-Privilege access is enabled 1: Privilege access is enabled Reset type: XRSn
0	PRIV_ADC0	R/W	0h	Privilege access enabled to ADC0 0: Non-Privilege access is enabled 1: Privilege access is enabled Reset type: XRSn

9 PPC_PRIVATTRIB_TIMER Register (Offset = 110Ch) [Reset = 00000000h]

PPC_PRIVATTRIB_TIMER is shown in Figure 8-15 and described in Table 8-17.

Return to the Summary Table.

Peripheral Privilege Access Timer

Figure 8-15. PPC_PRIVATTRIB_TIMER Register

31	30	29	28	27	26	25	24
RESERVED						PRIV_TIMG12_0	PRIV_TIMG4_0
R-0h						R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
RESERVED					PRIV_EQEP2	PRIV_EQEP1	PRIV_EQEP0
R-0h					R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED						PRIV_ECAP1	PRIV_ECAP0
R-0h						R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
RESERVED			PRIV_PWM4	PRIV_PWM3	PRIV_PWM2	PRIV_PWM1	PRIV_PWM0
R-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 8-26. PPC_PRIVATTRIB_TIMER Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	RESERVED	R	0h	Reserved
25	PRIV_TIMG12_0	R/W	0h	Privilege access enabled to TIMG12_0 0: Non-Privilege access is enabled 1: Privilege access is enabled Reset type: XRSn
24	PRIV_TIMG4_0	R/W	0h	Privilege access enabled to TIMG4_0 0: Non-Privilege access is enabled 1: Privilege access is enabled Reset type: XRSn
23-19	RESERVED	R	0h	Reserved
18	PRIV_EQEP2	R/W	0h	Privilege access enabled to EQEP2 0: Non-Privilege access is enabled 1: Privilege access is enabled Reset type: XRSn
17	PRIV_EQEP1	R/W	0h	Privilege access enabled to EQEP1 0: Non-Privilege access is enabled 1: Privilege access is enabled Reset type: XRSn
16	PRIV_EQEP0	R/W	0h	Privilege access enabled to EQEP0 0: Non-Privilege access is enabled 1: Privilege access is enabled Reset type: XRSn
15-10	RESERVED	R	0h	Reserved

Table 8-26. PPC_PRIVATTRIB_TIMER Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
9	PRIV_ECAP1	R/W	0h	Privilege access enabled to ECAP1 0: Non-Privilege access is enabled 1: Privilege access is enabled Reset type: XRSn
8	PRIV_ECAP0	R/W	0h	Privilege access enabled to ECAP0 0: Non-Privilege access is enabled 1: Privilege access is enabled Reset type: XRSn
7-5	RESERVED	R	0h	Reserved
4	PRIV_PWM4	R/W	0h	Privilege access enabled to PWM4 0: Non-Privilege access is enabled 1: Privilege access is enabled Reset type: XRSn
3	PRIV_PWM3	R/W	0h	Privilege access enabled to PWM3 0: Non-Privilege access is enabled 1: Privilege access is enabled Reset type: XRSn
2	PRIV_PWM2	R/W	0h	Privilege access enabled to PWM2 0: Non-Privilege access is enabled 1: Privilege access is enabled Reset type: XRSn
1	PRIV_PWM1	R/W	0h	Privilege access enabled to PWM1 0: Non-Privilege access is enabled 1: Privilege access is enabled Reset type: XRSn
0	PRIV_PWM0	R/W	0h	Privilege access enabled to PWM0 0: Non-Privilege access is enabled 1: Privilege access is enabled Reset type: XRSn

10 PPC_PRIVATTRIB_UNICOMM Register (Offset = 1114h) [Reset = 00000000h]

PPC_PRIVATTRIB_UNICOMM is shown in Figure 8-16 and described in Table 8-18.

Return to the Summary Table.

Peripheral Privilege Access UniComm

Figure 8-16. PPC_PRIVATTRIB_UNICOMM Register

31	30	29	28	27	26	25	24
RESERVED						PRIV_S1_COMMON	PRIV_S0_COMMON
R-0h						R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED		PRIV_S1U5	PRIV_S1U4	PRIV_S1U3	PRIV_S0U2	PRIV_S0U1	PRIV_S0U0
R-0h		R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 8-28. PPC_PRIVATTRIB_UNICOMM Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	RESERVED	R	0h	Reserved
25	PRIV_S1_COMMON	R/W	0h	Privilege access enabled to UniComm S1 common region 0: Non-Privilege access is enabled 1: Privilege access is enabled Reset type: XRSn
24	PRIV_S0_COMMON	R/W	0h	Privilege access enabled to UniComm S0 common region 0: Non-Privilege access is enabled 1: Privilege access is enabled Reset type: XRSn
23-6	RESERVED	R	0h	Reserved
5	PRIV_S1U5	R/W	0h	Privilege access enabled to UniComm S1U5 0: Non-Privilege access is enabled 1: Privilege access is enabled Reset type: XRSn
4	PRIV_S1U4	R/W	0h	Privilege access enabled to UniComm S1U4 0: Non-Privilege access is enabled 1: Privilege access is enabled Reset type: XRSn
3	PRIV_S1U3	R/W	0h	Privilege access enabled to UniComm S1U3 0: Non-Privilege access is enabled 1: Privilege access is enabled Reset type: XRSn
2	PRIV_S0U2	R/W	0h	Privilege access enabled to UniComm S0U2 0: Non-Privilege access is enabled 1: Privilege access is enabled Reset type: XRSn

Table 8-28. PPC_PRIVATTRIB_UNICOMM Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
1	PRIV_S0U1	R/W	0h	Privilege access enabled to UniComm S0U1 0: Non-Privilege access is enabled 1: Privilege access is enabled Reset type: XRSn
0	PRIV_S0U0	R/W	0h	Privilege access enabled to UniComm S0U0 0: Non-Privilege access is enabled 1: Privilege access is enabled Reset type: XRSn

11 PPC_PRIVATTRIB_GPIO Register (Offset = 111Ch) [Reset = 00000000h]

PPC_PRIVATTRIB_GPIO is shown in Figure 8-17 and described in Table 8-19.

Return to the Summary Table.

Peripheral Privilege Access GPIO

Figure 8-17. PPC_PRIVATTRIB_GPIO Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED				PRIV_GPIO3	PRIV_GPIO2	PRIV_GPIO1	PRIV_GPIO0
R-0h				R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 8-30. PPC_PRIVATTRIB_GPIO Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R	0h	Reserved
3	PRIV_GPIO3	R/W	0h	Privilege access enabled to GPIO3 0: Non-Privilege access is enabled 1: Privilege access is enabled Reset type: XRSn
2	PRIV_GPIO2	R/W	0h	Privilege access enabled to GPIO2 0: Non-Privilege access is enabled 1: Privilege access is enabled Reset type: XRSn
1	PRIV_GPIO1	R/W	0h	Privilege access enabled to GPIO1 0: Non-Privilege access is enabled 1: Privilege access is enabled Reset type: XRSn
0	PRIV_GPIO0	R/W	0h	Privilege access enabled to GPIO0 0: Non-Privilege access is enabled 1: Privilege access is enabled Reset type: XRSn

12 PPC_PRIVATTRIB_MCAN Register (Offset = 1124h) [Reset = 00000000h]

PPC_PRIVATTRIB_MCAN is shown in Figure 8-18 and described in Table 8-20.

Return to the Summary Table.

Peripheral Privilege Access CAN

Figure 8-18. PPC_PRIVATTRIB_MCAN Register

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

Table 8-32. PPC_PRIVATTRIB_MCAN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	Reserved
0	PRIV_MCAN	R/W	0h	Privilege access enabled to MCAN 0: Non-Privilege access is enabled 1: Privilege access is enabled Reset type: XRSn

13 PPC_PRIVATTRIB_CMPSS Register (Offset = 1134h) [Reset = 00000000h]

PPC_PRIVATTRIB_CMPSS is shown in Figure 8-19 and described in Table 8-21.

Return to the Summary Table.

Peripheral Privilege Access CMPSS

Figure 8-19. PPC_PRIVATTRIB_CMPSS Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED				PRIV_CMPSS3	PRIV_CMPSS2	PRIV_CMPSS1	PRIV_CMPSS0
R-0h				R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 8-34. PPC_PRIVATTRIB_CMPSS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R	0h	Reserved
3	PRIV_CMPSS3	R/W	0h	Privilege access enabled to CMPSS3 0: Non-Privilege access is enabled 1: Privilege access is enabled Reset type: XRSn
2	PRIV_CMPSS2	R/W	0h	Privilege access enabled to CMPSS2 0: Non-Privilege access is enabled 1: Privilege access is enabled Reset type: XRSn
1	PRIV_CMPSS1	R/W	0h	Privilege access enabled to CMPSS1 0: Non-Privilege access is enabled 1: Privilege access is enabled Reset type: XRSn
0	PRIV_CMPSS0	R/W	0h	Privilege access enabled to CMPSS0 0: Non-Privilege access is enabled 1: Privilege access is enabled Reset type: XRSn

14 PPC_PRIVATTRIB_WDT Register (Offset = 113Ch) [Reset = 00000000h]

PPC_PRIVATTRIB_WDT is shown in Figure 8-20 and described in Table 8-22.

Return to the Summary Table.

Peripheral Privilege Access Watchdog

Figure 8-20. PPC_PRIVATTRIB_WDT Register

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

Table 8-36. PPC_PRIVATTRIB_WDT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	Reserved
0	PRIV_WWDT	R/W	0h	Privilege access enabled to Windowed Watchdog 0: Non-Privilege access is enabled 1: Privilege access is enabled Reset type: XRSn

15 PPC_PRIVATTRIB_CRC Register (Offset = 114Ch) [Reset = 00000001h]

PPC_PRIVATTRIB_CRC is shown in Figure 8-21 and described in Table 8-23.

Return to the Summary Table.

Peripheral Privilege Access CRC

Figure 8-21. PPC_PRIVATTRIB_CRC Register

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

Table 8-38. PPC_PRIVATTRIB_CRC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	Reserved
0	PRIV_CRC	R/W	1h	Privilege access enabled to CRC 0: Non-Privilege access is enabled 1: Privilege access is enabled Reset type: XRSn

16 PPC_PRIVATTRIB_AES Register (Offset = 1154h) [Reset = 00000000h]

PPC_PRIVATTRIB_AES is shown in Figure 8-22 and described in Table 8-24.

Return to the Summary Table.

Peripheral Privilege Access AES

Figure 8-22. PPC_PRIVATTRIB_AES Register

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

Table 8-40. PPC_PRIVATTRIB_AES Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	Reserved
0	PRIV_AES	R/W	0h	Privilege access enabled to AES 0: Non-Privilege access is enabled 1: Privilege access is enabled Reset type: XRSn

17 PPC_PRIVATTRIB_IOMUX Register (Offset = 116Ch) [Reset = 00000000h]

PPC_PRIVATTRIB_IOMUX is shown in Figure 8-23 and described in Table 8-25.

Return to the Summary Table.

Peripheral Privilege Access IOMUX

Figure 8-23. PPC_PRIVATTRIB_IOMUX Register

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

Table 8-42. PPC_PRIVATTRIB_IOMUX Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	Reserved
0	PRIV_IOMUX	R/W	0h	Privilege access enabled to IOMUX 0: Non-Privilege access is enabled 1: Privilege access is enabled Reset type: XRSn

18 PPC_PRIVATTRIB_DEBUGSS Register (Offset = 1184h) [Reset = 00000001h]

PPC_PRIVATTRIB_DEBUGSS is shown in Figure 8-24 and described in Table 8-26.

Return to the Summary Table.

Peripheral Privilege Access DEBUGSS

Figure 8-24. PPC_PRIVATTRIB_DEBUGSS Register

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

Table 8-44. PPC_PRIVATTRIB_DEBUGSS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	Reserved
0	PRIV_DEBUGSS	R/W	1h	Privilege access enabled to DEBUGSS 0: Non-Privilege access is enabled 1: Privilege access is enabled Reset type: XRSn

19 PPC_PRIVATTRIB_KEYSTORECTL Register (Offset = 1194h) [Reset = 00000000h]

PPC_PRIVATTRIB_KEYSTORECTL is shown in Figure 8-25 and described in Table 8-27.

Return to the Summary Table.

Peripheral Privilege Access KEYSTORECTL

Figure 8-25. PPC_PRIVATTRIB_KEYSTORECTL Register

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

Table 8-46. PPC_PRIVATTRIB_KEYSTORECTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	Reserved
0	PRIV_KEYSTORECTL	R/W	0h	Privilege access enabled to KEYSTORECTL 0: Non-Privilege access is enabled 1: Privilege access is enabled Reset type: XRSn

ADVANCE INFORMATION

20 PPC_PRIVATTRIB_EAM Register (Offset = 119Ch) [Reset = 00000000h]

PPC_PRIVATTRIB_EAM is shown in Figure 8-26 and described in Table 8-28.

Return to the Summary Table.

Peripheral Privilege Access EAM

Figure 8-26. PPC_PRIVATTRIB_EAM Register

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

Table 8-48. PPC_PRIVATTRIB_EAM Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	Reserved
0	PRIV_EAM	R/W	0h	Privilege access enabled to EAM 0: Non-Privilege access is enabled 1: Privilege access is enabled Reset type: XRSn

21 PPC_PRIVATTRIB_FRIREGS Register (Offset = 11ACh) [Reset = 00000000h]

PPC_PRIVATTRIB_FRIREGS is shown in Figure 8-27 and described in Table 8-29.

Return to the Summary Table.

Peripheral Privilege Access FRIREGS

Figure 8-27. PPC_PRIVATTRIB_FRIREGS Register

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

Table 8-50. PPC_PRIVATTRIB_FRIREGS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	Reserved
0	PRIV_FRIREGS	R/W	0h	Privilege access enabled to FRIREGS 0: Non-Privilege access is enabled 1: Privilege access is enabled Reset type: XRSn

22 PPC_PRIVATTRIB_MEMCFG Register (Offset = 11B4h) [Reset = 00000000h]

PPC_PRIVATTRIB_MEMCFG is shown in Figure 8-28 and described in Table 8-30.

Return to the Summary Table.

Peripheral Privilege Access MEMCFG

Figure 8-28. PPC_PRIVATTRIB_MEMCFG Register

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

Table 8-52. PPC_PRIVATTRIB_MEMCFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	Reserved
0	PRIV_MEMCFG	R/W	0h	Privilege access enabled to MEMCFG 0: Non-Privilege access is enabled 1: Privilege access is enabled Reset type: XRSn

23 PPC_PRIVATTRIB_SYSCTL Register (Offset = 120Ch) [Reset = 00000001h]

PPC_PRIVATTRIB_SYSCTL is shown in Figure 8-29 and described in Table 8-31.

Return to the Summary Table.

Peripheral Privilege Access System Control

Figure 8-29. PPC_PRIVATTRIB_SYSCTL Register

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

Table 8-54. PPC_PRIVATTRIB_SYSCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	Reserved
0	PRIV_SYSCTL	R/W	1h	Privilege access enabled to SYSCTL 0: Non-Privilege access is enabled 1: Privilege access is enabled Reset type: XRSn

24 PPC_PRIVATTRIB_DMA Register (Offset = 1214h) [Reset = 00000001h]

PPC_PRIVATTRIB_DMA is shown in Figure 8-30 and described in Table 8-32.

Return to the Summary Table.

Peripheral Privilege Access DMA

Figure 8-30. PPC_PRIVATTRIB_DMA Register

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

Table 8-56. PPC_PRIVATTRIB_DMA Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	Reserved
0	PRIV_DMA	R/W	1h	Privilege access enabled to DMA 0: Non-Privilege access is enabled 1: Privilege access is enabled Reset type: XRSn

25 PPC_PRIVATTRIB_GSC Register (Offset = 121Ch) [Reset = 00000001h]

PPC_PRIVATTRIB_GSC is shown in Figure 8-31 and described in Table 8-33.

Return to the Summary Table.

Peripheral Privilege Access GSC

Figure 8-31. PPC_PRIVATTRIB_GSC Register

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

Table 8-58. PPC_PRIVATTRIB_GSC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	Reserved
0	PRIV_GSC	R/W	1h	Privilege access enabled to GSC 0: Non-Privilege access is enabled 1: Privilege access is enabled Reset type: XRSn

26 PPC_PRIVATTRIB_COPROC Register (Offset = 1224h) [Reset = 00000000h]

PPC_PRIVATTRIB_COPROC is shown in Figure 8-32 and described in Table 8-34.

Return to the Summary Table.

Peripheral Privilege Access Co-processor

Figure 8-32. PPC_PRIVATTRIB_COPROC Register

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

Table 8-60. PPC_PRIVATTRIB_COPROC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	Reserved
0	PRIV_TINIE	R/W	0h	Privilege access enabled to TINIE 0: Non-Privilege access is enabled 1: Privilege access is enabled Reset type: XRSn

27 PPC_PRIVATTRIB_EPI Register (Offset = 122Ch) [Reset = 00000000h]

PPC_PRIVATTRIB_EPI is shown in Figure 8-33 and described in Table 8-35.

Return to the Summary Table.

Peripheral Privilege Access EPI

Figure 8-33. PPC_PRIVATTRIB_EPI Register

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

Table 8-62. PPC_PRIVATTRIB_EPI Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	Reserved
0	PRIV_EPI	R/W	0h	Privilege access enabled to EPI 0: Non-Privilege access is enabled 1: Privilege access is enabled Reset type: XRSn

28 PPC_PRIVATTRIB_PGA Register (Offset = 1234h) [Reset = 00000000h]

PPC_PRIVATTRIB_PGA is shown in Figure 8-34 and described in Table 8-36.

Return to the Summary Table.

Peripheral Privilege Access PGA

Figure 8-34. PPC_PRIVATTRIB_PGA Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED					PRIV_PGA2	PRIV_PGA1	PRIV_PGA0
R-0h					R/W-0h	R/W-0h	R/W-0h

Table 8-64. PPC_PRIVATTRIB_PGA Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R	0h	Reserved
2	PRIV_PGA2	R/W	0h	Privilege access enabled to PGA2 0: Non-Privilege access is enabled 1: Privilege access is enabled Reset type: XRSn
1	PRIV_PGA1	R/W	0h	Privilege access enabled to PGA1 0: Non-Privilege access is enabled 1: Privilege access is enabled Reset type: XRSn
0	PRIV_PGA0	R/W	0h	Privilege access enabled to PGA0 0: Non-Privilege access is enabled 1: Privilege access is enabled Reset type: XRSn

29 PPC_PRIVATTRIB_XBAR Register (Offset = 123Ch) [Reset = 00000000h]

PPC_PRIVATTRIB_XBAR is shown in Figure 8-35 and described in Table 8-37.

Return to the Summary Table.

Peripheral Privilege Access XBAR

Figure 8-35. PPC_PRIVATTRIB_XBAR Register

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

Table 8-66. PPC_PRIVATTRIB_XBAR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R	0h	Reserved
2	PRIV_OUTPUT_XBAR_FLAGS	R/W	0h	Privilege access enabled to OUTPUT_XBAR_FLAGS 0: Non-Privilege access is enabled 1: Privilege access is enabled Reset type: XRSn
1	PRIV_XBAR_INPUT_FLAGS	R/W	0h	Privilege access enabled to XBAR_INPUT_FLAGS 0: Non-Privilege access is enabled 1: Privilege access is enabled Reset type: XRSn
0	PRIV_ALL_XBAR_CONFIG	R/W	0h	Privilege access enabled to ALL_XBAR_CONFIG including SYNC_SOC_REGS 0: Non-Privilege access is enabled 1: Privilege access is enabled Reset type: XRSn

30 SPC_PRIVATTRIB0 Register (Offset = 12C0h) [Reset = 7F7F7F7Fh]

SPC_PRIVATTRIB0 is shown in Figure 8-36 and described in Table 8-38.

Return to the Summary Table.

SRAM Privilege Attribute-0 Register

Figure 8-36. SPC_PRIVATTRIB0 Register

31	30	29	28	27	26	25	24
RESERVED	C3B2_4KB	C3B1_4KB	C3B0_4KB	C3B0_2KB	C3B0_1KB	C3B1_512B	C3B0_512B
R-0h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h
23	22	21	20	19	18	17	16
RESERVED	C2B2_4KB	C2B1_4KB	C2B0_4KB	C2B0_2KB	C2B0_1KB	C2B1_512B	C2B0_512B
R-0h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h
15	14	13	12	11	10	9	8
RESERVED	C1B2_4KB	C1B1_4KB	C1B0_4KB	C1B0_2KB	C1B0_1KB	C1B1_512B	C1B0_512B
R-0h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h
7	6	5	4	3	2	1	0
RESERVED	C0B2_4KB	C0B1_4KB	C0B0_4KB	C0B0_2KB	C0B0_1KB	C0B1_512B	C0B0_512B
R-0h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h

Table 8-68. SPC_PRIVATTRIB0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R	0h	Reserved
30	C3B2_4KB	R/W	1h	privilege access enabled to 3rd 4KB in chunk-3 0: Non-privilege access is enabled 1: privilege access is enabled Reset type: XRSn
29	C3B1_4KB	R/W	1h	privilege access enabled to 2nd 4KB in chunk-3 0: Non-privilege access is enabled 1: privilege access is enabled Reset type: XRSn
28	C3B0_4KB	R/W	1h	privilege access enabled to 1st 4KB in chunk-3 0: Non-privilege access is enabled 1: privilege access is enabled Reset type: XRSn
27	C3B0_2KB	R/W	1h	privilege access enabled to 2KB in chunk-3 0: Non-privilege access is enabled 1: privilege access is enabled Reset type: XRSn
26	C3B0_1KB	R/W	1h	privilege access enabled to 1KB in chunk-3 0: Non-privilege access is enabled 1: privilege access is enabled Reset type: XRSn
25	C3B1_512B	R/W	1h	privilege access enabled to 2nd 512 byte in chunk-3 0: Non-privilege access is enabled 1: privilege access is enabled Reset type: XRSn

Table 8-68. SPC_PRIVATTRIB0 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
24	C3B0_512B	R/W	1h	privilege access enabled to 1st 512 byte in chunk-3 0: Non-privilege access is enabled 1: privilege access is enabled Reset type: XRSn
23	RESERVED	R	0h	Reserved
22	C2B2_4KB	R/W	1h	privilege access enabled to 3rd 4KB in chunk-2 0: Non-privilege access is enabled 1: privilege access is enabled Reset type: XRSn
21	C2B1_4KB	R/W	1h	privilege access enabled to 2nd 4KB in chunk-2 0: Non-privilege access is enabled 1: privilege access is enabled Reset type: XRSn
20	C2B0_4KB	R/W	1h	privilege access enabled to 1st 4KB in chunk-2 0: Non-privilege access is enabled 1: privilege access is enabled Reset type: XRSn
19	C2B0_2KB	R/W	1h	privilege access enabled to 2KB in chunk-2 0: Non-privilege access is enabled 1: privilege access is enabled Reset type: XRSn
18	C2B0_1KB	R/W	1h	privilege access enabled to 1KB in chunk-2 0: Non-privilege access is enabled 1: privilege access is enabled Reset type: XRSn
17	C2B1_512B	R/W	1h	privilege access enabled to 2nd 512 byte in chunk-2 0: Non-privilege access is enabled 1: privilege access is enabled Reset type: XRSn
16	C2B0_512B	R/W	1h	privilege access enabled to 1st 512 byte in chunk-2 0: Non-privilege access is enabled 1: privilege access is enabled Reset type: XRSn
15	RESERVED	R	0h	Reserved
14	C1B2_4KB	R/W	1h	privilege access enabled to 3rd 4KB in chunk-1 0: Non-privilege access is enabled 1: privilege access is enabled Reset type: XRSn
13	C1B1_4KB	R/W	1h	privilege access enabled to 2nd 4KB in chunk-1 0: Non-privilege access is enabled 1: privilege access is enabled Reset type: XRSn
12	C1B0_4KB	R/W	1h	privilege access enabled to 1st 4KB in chunk-1 0: Non-privilege access is enabled 1: privilege access is enabled Reset type: XRSn

Table 8-68. SPC_PRIVATTRIB0 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
11	C1B0_2KB	R/W	1h	privilege access enabled to 2KB in chunk-1 0: Non-privilege access is enabled 1: privilege access is enabled Reset type: XRSn
10	C1B0_1KB	R/W	1h	privilege access enabled to 1KB in chunk-1 0: Non-privilege access is enabled 1: privilege access is enabled Reset type: XRSn
9	C1B1_512B	R/W	1h	privilege access enabled to 2nd 512 byte in chunk-1 0: Non-privilege access is enabled 1: privilege access is enabled Reset type: XRSn
8	C1B0_512B	R/W	1h	privilege access enabled to 1st 512 byte in chunk-1 0: Non-privilege access is enabled 1: privilege access is enabled Reset type: XRSn
7	RESERVED	R	0h	Reserved
6	C0B2_4KB	R/W	1h	privilege access enabled to 3rd 4KB in chunk-0 0: Non-privilege access is enabled 1: privilege access is enabled Reset type: XRSn
5	C0B1_4KB	R/W	1h	privilege access enabled to 2nd 4KB in chunk-0 0: Non-privilege access is enabled 1: privilege access is enabled Reset type: XRSn
4	C0B0_4KB	R/W	1h	privilege access enabled to 1st 4KB in chunk-0 0: Non-privilege access is enabled 1: privilege access is enabled Reset type: XRSn
3	C0B0_2KB	R/W	1h	privilege access enabled to 2KB in chunk-0 0: Non-privilege access is enabled 1: privilege access is enabled Reset type: XRSn
2	C0B0_1KB	R/W	1h	privilege access enabled to 1KB in chunk-0 0: Non-privilege access is enabled 1: privilege access is enabled Reset type: XRSn
1	C0B1_512B	R/W	1h	privilege access enabled to 2nd 512 byte in chunk-0 0: Non-privilege access is enabled 1: privilege access is enabled Reset type: XRSn
0	C0B0_512B	R/W	1h	privilege access enabled to 1st 512 byte in chunk-0 0: Non-privilege access is enabled 1: privilege access is enabled Reset type: XRSn

31 SPC_PRIVATTRIB1 Register (Offset = 12C4h) [Reset = 7F7F7F7Fh]

SPC_PRIVATTRIB1 is shown in Figure 8-37 and described in Table 8-39.

Return to the Summary Table.

SRAM Privilege Attribute-1 Register

Figure 8-37. SPC_PRIVATTRIB1 Register

31	30	29	28	27	26	25	24
RESERVED	C3B2_4KB	C3B1_4KB	C3B0_4KB	C3B0_2KB	C3B0_1KB	C3B1_512B	C3B0_512B
R-0h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h
23	22	21	20	19	18	17	16
RESERVED	C2B2_4KB	C2B1_4KB	C2B0_4KB	C2B0_2KB	C2B0_1KB	C2B1_512B	C2B0_512B
R-0h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h
15	14	13	12	11	10	9	8
RESERVED	C1B2_4KB	C1B1_4KB	C1B0_4KB	C1B0_2KB	C1B0_1KB	C1B1_512B	C1B0_512B
R-0h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h
7	6	5	4	3	2	1	0
RESERVED	C0B2_4KB	C0B1_4KB	C0B0_4KB	C0B0_2KB	C0B0_1KB	C0B1_512B	C0B0_512B
R-0h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h

Table 8-70. SPC_PRIVATTRIB1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R	0h	Reserved
30	C3B2_4KB	R/W	1h	privilege access enabled to 3rd 4KB in chunk-3 0: Non-privilege access is enabled 1: privilege access is enabled Reset type: XRSn
29	C3B1_4KB	R/W	1h	privilege access enabled to 2nd 4KB in chunk-3 0: Non-privilege access is enabled 1: privilege access is enabled Reset type: XRSn
28	C3B0_4KB	R/W	1h	privilege access enabled to 1st 4KB in chunk-3 0: Non-privilege access is enabled 1: privilege access is enabled Reset type: XRSn
27	C3B0_2KB	R/W	1h	privilege access enabled to 2KB in chunk-3 0: Non-privilege access is enabled 1: privilege access is enabled Reset type: XRSn
26	C3B0_1KB	R/W	1h	privilege access enabled to 1KB in chunk-3 0: Non-privilege access is enabled 1: privilege access is enabled Reset type: XRSn
25	C3B1_512B	R/W	1h	privilege access enabled to 2nd 512 byte in chunk-3 0: Non-privilege access is enabled 1: privilege access is enabled Reset type: XRSn

Table 8-70. SPC_PRIVATTRIB1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
24	C3B0_512B	R/W	1h	privilege access enabled to 1st 512 byte in chunk-3 0: Non-privilege access is enabled 1: privilege access is enabled Reset type: XRSn
23	RESERVED	R	0h	Reserved
22	C2B2_4KB	R/W	1h	privilege access enabled to 3rd 4KB in chunk-2 0: Non-privilege access is enabled 1: privilege access is enabled Reset type: XRSn
21	C2B1_4KB	R/W	1h	privilege access enabled to 2nd 4KB in chunk-2 0: Non-privilege access is enabled 1: privilege access is enabled Reset type: XRSn
20	C2B0_4KB	R/W	1h	privilege access enabled to 1st 4KB in chunk-2 0: Non-privilege access is enabled 1: privilege access is enabled Reset type: XRSn
19	C2B0_2KB	R/W	1h	privilege access enabled to 2KB in chunk-2 0: Non-privilege access is enabled 1: privilege access is enabled Reset type: XRSn
18	C2B0_1KB	R/W	1h	privilege access enabled to 1KB in chunk-2 0: Non-privilege access is enabled 1: privilege access is enabled Reset type: XRSn
17	C2B1_512B	R/W	1h	privilege access enabled to 2nd 512 byte in chunk-2 0: Non-privilege access is enabled 1: privilege access is enabled Reset type: XRSn
16	C2B0_512B	R/W	1h	privilege access enabled to 1st 512 byte in chunk-2 0: Non-privilege access is enabled 1: privilege access is enabled Reset type: XRSn
15	RESERVED	R	0h	Reserved
14	C1B2_4KB	R/W	1h	privilege access enabled to 3rd 4KB in chunk-1 0: Non-privilege access is enabled 1: privilege access is enabled Reset type: XRSn
13	C1B1_4KB	R/W	1h	privilege access enabled to 2nd 4KB in chunk-1 0: Non-privilege access is enabled 1: privilege access is enabled Reset type: XRSn
12	C1B0_4KB	R/W	1h	privilege access enabled to 1st 4KB in chunk-1 0: Non-privilege access is enabled 1: privilege access is enabled Reset type: XRSn

Table 8-70. SPC_PRIVATTRIB1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
11	C1B0_2KB	R/W	1h	privilege access enabled to 2KB in chunk-1 0: Non-privilege access is enabled 1: privilege access is enabled Reset type: XRSn
10	C1B0_1KB	R/W	1h	privilege access enabled to 1KB in chunk-1 0: Non-privilege access is enabled 1: privilege access is enabled Reset type: XRSn
9	C1B1_512B	R/W	1h	privilege access enabled to 2nd 512 byte in chunk-1 0: Non-privilege access is enabled 1: privilege access is enabled Reset type: XRSn
8	C1B0_512B	R/W	1h	privilege access enabled to 1st 512 byte in chunk-1 0: Non-privilege access is enabled 1: privilege access is enabled Reset type: XRSn
7	RESERVED	R	0h	Reserved
6	C0B2_4KB	R/W	1h	privilege access enabled to 3rd 4KB in chunk-0 0: Non-privilege access is enabled 1: privilege access is enabled Reset type: XRSn
5	C0B1_4KB	R/W	1h	privilege access enabled to 2nd 4KB in chunk-0 0: Non-privilege access is enabled 1: privilege access is enabled Reset type: XRSn
4	C0B0_4KB	R/W	1h	privilege access enabled to 1st 4KB in chunk-0 0: Non-privilege access is enabled 1: privilege access is enabled Reset type: XRSn
3	C0B0_2KB	R/W	1h	privilege access enabled to 2KB in chunk-0 0: Non-privilege access is enabled 1: privilege access is enabled Reset type: XRSn
2	C0B0_1KB	R/W	1h	privilege access enabled to 1KB in chunk-0 0: Non-privilege access is enabled 1: privilege access is enabled Reset type: XRSn
1	C0B1_512B	R/W	1h	privilege access enabled to 2nd 512 byte in chunk-0 0: Non-privilege access is enabled 1: privilege access is enabled Reset type: XRSn
0	C0B0_512B	R/W	1h	privilege access enabled to 1st 512 byte in chunk-0 0: Non-privilege access is enabled 1: privilege access is enabled Reset type: XRSn

32 FPC_PRIVATTRIBA0 Register (Offset = 1400h) [Reset = FFFFFFFFh]

FPC_PRIVATTRIBA0 is shown in Figure 8-38 and described in Table 8-40.

Return to the Summary Table.

Flash Privilege Attribute Register A0

Figure 8-38. FPC_PRIVATTRIBA0 Register

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

Table 8-72. FPC_PRIVATTRIBA0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	PRIV_PAGE	R/W	FFFFFFFFh	Privilege access enable for first 32 sectors of flash with each bit representing 1 sector of 2KB 0: Non-Privilege access is enabled 1: Privilege access is enabled Reset type: XRSn

33 FPC_PRIVATTRIBA1 Register (Offset = 1404h) [Reset = FFFFFFFFh]

FPC_PRIVATTRIBA1 is shown in Figure 8-39 and described in Table 8-41.

Return to the Summary Table.

Flash Privilege Attribute Register A1

Figure 8-39. FPC_PRIVATTRIBA1 Register

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

Table 8-74. FPC_PRIVATTRIBA1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	PRIV_PAGE	R/W	FFFFFFFFh	Privilege access enable for first 32 sectors of flash with each bit representing 1 sector of 2KB 0: Non-Privilege access is enabled 1: Privilege access is enabled Reset type: XRSn

34 FPC_PRIVATTRIBB0 Register (Offset = 1408h) [Reset = 0000FFF0h]

FPC_PRIVATTRIBB0 is shown in Figure 8-40 and described in Table 8-42.

Return to the Summary Table.

Flash Privilege Attribute Register B0

Figure 8-40. FPC_PRIVATTRIBB0 Register

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

Table 8-76. FPC_PRIVATTRIBB0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15-4	PRIV_PAGE	R/W	FFFh	Privilege access enable for 64KB-256KB flash with each bit representing 8 sectors of 2KB 0: Non-privilege access is enabled 1: privilege access is enabled Reset type: XRSn
3-0	RESERVED	R	0h	Reserved

35 FPC_PRIVATTRIBB1 Register (Offset = 140Ch) [Reset = 0000FFF0h]

FPC_PRIVATTRIBB1 is shown in Figure 8-41 and described in Table 8-43.

Return to the Summary Table.

Flash Privilege Attribute Register B1

Figure 8-41. FPC_PRIVATTRIBB1 Register

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

Table 8-78. FPC_PRIVATTRIBB1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15-4	PRIV_PAGE	R/W	FFFh	Privilege access enable for 64KB-256KB flash with each bit representing 8 sectors of 2KB 0: Non-privilege access is enabled 1: privilege access is enabled Reset type: XRSn
3-0	RESERVED	R	0h	Reserved

36 FPC_WEPROTA0 Register (Offset = 1600h) [Reset = 00000000h]

FPC_WEPROTA0 is shown in Figure 8-42 and described in Table 8-44.

Return to the Summary Table.

Flash Write Protect Attribute Register A0

Figure 8-42. FPC_WEPROTA0 Register

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

Table 8-80. FPC_WEPROTA0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	WEPROT_PAGE	R/W1S	0h	Write protection enable for first 32 sectors of flash with each bit representing 1 sector of 2KB 0: Page is not erase/program protected 1: Page is erase/program protected Reset type: XRSn

37 FPC_WEPROTA1 Register (Offset = 1604h) [Reset = 00000000h]

FPC_WEPROTA1 is shown in Figure 8-43 and described in Table 8-45.

Return to the Summary Table.

Flash Write Protect Attribute Register A1

Figure 8-43. FPC_WEPROTA1 Register

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

Table 8-82. FPC_WEPROTA1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	WEPROT_PAGE	R/W1S	0h	Write protection enable for first 32 sectors of flash with each bit representing 1 sector of 2KB 0: Page is not erase/program protected 1: Page is erase/program protected Reset type: XRSn

38 FPC_WEPROTB0 Register (Offset = 1608h) [Reset = 00000000h]

FPC_WEPROTB0 is shown in Figure 8-44 and described in Table 8-46.

Return to the Summary Table.

Flash Write Protect Attribute Register B0

Figure 8-44. FPC_WEPROTB0 Register

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

Table 8-84. FPC_WEPROTB0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15-4	WEPROT_PAGE	R/W1S	0h	Write Protection enable for 64KB-256KB flash with each bit representing 8 sectors of 2KB 0: Page is not erase/program protected 1: Page is erase/program protected Reset type: XRSn
3-0	RESERVED	R	0h	Reserved

39 FPC_WEPROTB1 Register (Offset = 160Ch) [Reset = 00000000h]

FPC_WEPROTB1 is shown in Figure 8-45 and described in Table 8-47.

Return to the Summary Table.

Flash Write Protect Attribute Register B1

Figure 8-45. FPC_WEPROTB1 Register

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

Table 8-86. FPC_WEPROTB1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15-4	WEPROT_PAGE	R/W1S	0h	Write Protection enable for 64KB-256KB flash with each bit representing 8 sectors of 2KB 0: Page is not erase/program protected 1: Page is erase/program protected Reset type: XRSn
3-0	RESERVED	R	0h	Reserved

40 FPC_FLSEMREQ Register (Offset = 1800h) [Reset = 00000000h]

FPC_FLSEMREQ is shown in Figure 8-46 and described in Table 8-48.

Return to the Summary Table.

Flash semaphore request register

Figure 8-46. FPC_FLSEMREQ Register

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

Table 8-88. FPC_FLSEMREQ Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	Reserved
0	REQ	R-0/W1S	0h	If the FLSEMSTAT.ASSIGNED bit is cleared, writing a '1' to this bit causes: 1) Load the Privilege attribute of the code performing the write into the FLSEMSTAT.PRIV bit field, and 2) Set the FLSEMSTAT.ASSIGNED bit. If the FLSEMSTAT.ASSIGNED is already set when a write to this bit occurs, the write will be ignored. If the above conditions are not met during the write, the write will be ignored with no error indicator. It is advised that the writing code perform a read of the FLSEMSTAT register to ensure it was set after writing to this bit. Reset type: XRSn

ADVANCE INFORMATION

41 FPC_FLSEMCLR Register (Offset = 1804h) [Reset = 00000000h]

FPC_FLSEMCLR is shown in Figure 8-47 and described in Table 8-49.

Return to the Summary Table.

Flash semaphore release register

Figure 8-47. FPC_FLSEMCLR Register

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

Table 8-90. FPC_FLSEMCLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	Reserved
0	CLR	R-0/W1S	0h	If the following conditions are met, a write of '1' to this bit causes the FLSEMSTAT register to its reset state: 1) The FLSEMSTAT.PRIV of the code performing the write is 0x1 (i.e. privilege code can force relinquishing of the semaphore) or 2) FLSEMSTAT.ASSIGNED bit is set, and FLSEMSTAT.PRIV matches privilege attribute code performing the write. If the above conditions are not met during the write, the write will be ignored with no error indicated. It is advised that the writing code perform a read of the FLSEMSTAT register to ensure it was cleared after writing to this bit. Reset type: XRSn

42 FPC_FLSEMSTAT Register (Offset = 1808h) [Reset = 00000000h]

FPC_FLSEMSTAT is shown in Figure 8-48 and described in Table 8-50.

Return to the Summary Table.

Flash semaphore statys registers

Figure 8-48. FPC_FLSEMSTAT Register

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

Table 8-92. FPC_FLSEMSTAT Register Field Descriptions

Bit	Field	Type	Reset	Description
31	ASSIGNED	R	0h	PRIV field is valid when this bit is set. 0 : PRIV is unassigned and flash controller is not accessible 1 : PRIV value is valid and the flash controller is currently assigned to the respective code Reset type: XRSn
30	MATCH	R	0h	On a read, this bit will reflect whether the reader's PRIV matches the ownership of the flash controller. 0 : Code performing read does not own the flash controller semaphore 1 : Code performing read owns the flash controller semphore. This avoids code from being required to know whether the current code segment owns the flash controller configuration. Reset type: XRSn
29-17	RESERVED	R	0h	Reserved
16	DBGACC	R	0h	Defines the flash controller semaphore owners debug access indication 0x0 : functional access 0x1 :Debug access Note : This bit is only an indication that the flash semaphore was last grabbed by the funtional/debug master. The other privileged master is still allowed to clear and grab the semaphore anytime. Also, if it was previously grabbed by a privileged master, the other privileged master can still access the flash controller. Reset type: XRSn
15-9	RESERVED	R	0h	Reserved
8	PRIV	R	0h	Defines the flash controller semaphore owners privilage attribute 0x0 : non-privilege 0x1 :prvilege Reset type: XRSn

Table 8-92. FPC_FLSEMSTAT Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
7-1	RESERVED	R	0h	Reserved
0	RESERVED	R	0h	Reserved

43 VTOR_NS Register (Offset = 1B84h) [Reset = 01000001h]

VTOR_NS is shown in Figure 8-49 and described in Table 8-51.

Return to the Summary Table.

Non - Secure Vector Table Offset Register

Figure 8-49. VTOR_NS Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VTOR_NS																															
R/W-01000001h																															

Table 8-94. VTOR_NS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	VTOR_NS	R/W	01000001h	value of INITNSVTOR port at CPUSS boundary Reset type: Rsn

44 GSC_LOCK Register (Offset = 1D80h) [Reset = 00000000h]

GSC_LOCK is shown in Figure 8-50 and described in Table 8-52.

Return to the Summary Table.

GSC Lock configuration register

Figure 8-50. GSC_LOCK Register

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

Table 8-96. GSC_LOCK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	KEY	R-0/W	0h	Write 0xA551 to update the register Reset type: XRSn
15-8	RESERVED	R-0	0h	Reserved
7	RESERVED	R/W	0h	Reserved
6	RESERVED	R-0	0h	Reserved
5	VTOR	R/W	0h	Configuration impacts registers VTOR_* 0: Register configuration is not locked. 1: Register configuration is locked. Once register is locked - reads give the MMR data and writes are ignored Reset type: Rsn
4	RESERVED	R-0	0h	Reserved
3	FPPC_AATRIB	R/W	0h	Configuration impacts registers FPC_PRIVATTRIB_* 0: Register configuration is not locked. 1: Register configuration is locked. Once register is locked - reads give the MMR data and writes are ignored Reset type: XRSn
2	SPC_AATRIB	R/W	0h	Configuration impacts registers SPC_PRIVATTRIB_* 0: Register configuration is not locked. 1: Register configuration is locked. Once register is locked - reads give the MMR data and writes are ignored Reset type: XRSn

Table 8-96. GSC_LOCK Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
1	PPC_AATRIB	R/W	0h	Configuration impacts registers PPC_PRIVATTRIB_* 0: Register configuration is not locked. 1: Register configuration is locked. Once register is locked - reads give the MMR data and writes are ignored Reset type: XRSn
0	ATTRIBVIOL_CONFIG	R/W	0h	Configuration impacts registers SPC_ATTRIBVIOLP_CONFIG, PPC_ATTRIBVIOLP_CONFIG, FPC_ATTRIBVIOLP_CONFIG 0: Register configuration is not locked. 1: Register configuration is locked. Once register is locked - reads give the MMR data and writes are ignored Reset type: XRSn

45 GSC_COMMIT Register (Offset = 1D84h) [Reset = 00000000h]

GSC_COMMIT is shown in Figure 8-51 and described in Table 8-53.

Return to the Summary Table.

GSC commit configuration register

Figure 8-51. GSC_COMMIT Register

31	30	29	28	27	26	25	24
KEY							
R-0/W-0h							
23	22	21	20	19	18	17	16
KEY							
R-0/W-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED	RESERVED	VTOR	RESERVED	FPPC_AATRIB	SPC_AATRIB	PPC_AATRIB	ATTRIBVIOL_CONFIG
R/W1S-0h	R-0-0h	R/W1S-0h	R-0-0h	R/W1S-0h	R/W1S-0h	R/W1S-0h	R/W1S-0h

Table 8-98. GSC_COMMIT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	KEY	R-0/W	0h	Write 0xA442 to update the register Reset type: XRSn
15-8	RESERVED	R-0	0h	Reserved
7	RESERVED	R/W1S	0h	Reserved
6	RESERVED	R-0	0h	Reserved
5	VTOR	R/W1S	0h	0: Register lock configuration is not committed. 1: Register configuration is committed. Commit configuration blocks the writes going to the lock register. Once configuration is committed, only reset can change the configuration. Reset type: Rsn
4	RESERVED	R-0	0h	Reserved
3	FPPC_AATRIB	R/W1S	0h	0: Register lock configuration is not committed. 1: Register configuration is committed. Commit configuration blocks the writes going to the lock register. Once configuration is committed, only reset can change the configuration. Reset type: XRSn
2	SPC_AATRIB	R/W1S	0h	0: Register lock configuration is not committed. 1: Register configuration is committed. Commit configuration blocks the writes going to the lock register. Once configuration is committed, only reset can change the configuration. Reset type: XRSn

Table 8-98. GSC_COMMIT Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
1	PPC_AATRIB	R/W1S	0h	0: Register lock configuration is not committed. 1: Register configuration is committed. Commit configuration blocks the writes going to the lock register. Once configuration is committed, only reset can change the configuration. Reset type: XRSn
0	ATTRIBVIOL_CONFIG	R/W1S	0h	0: Register lock configuration is not committed. 1: Register configuration is committed. Commit configuration blocks the writes going to the lock register. Once configuration is committed, only reset can change the configuration. Reset type: XRSn

46 GSC_REVISION Register (Offset = 1FFCh) [Reset = 00000000h]

GSC_REVISION is shown in Figure 8-52 and described in Table 8-54.

Return to the Summary Table.

GSC Revision register

Figure 8-52. GSC_REVISION Register

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

Table 8-100. GSC_REVISION Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15-8	MAJREV	R	0h	This hardcoded field defines the major revision of the IP. Reset type: XRSn
7-0	MINREV	R	0h	This hardcoded field defines the minor revision of the IP. Reset type: XRSn

Chapter 9

Direct Memory Access (DMA)

The direct memory access (DMA) module transfers data from one address to another, without CPU intervention. This chapter describes the operation of the DMA module.

ADVANCE INFORMATION

9.1 DMA Overview

The DMA transfers data from a source address to a destination address without CPU intervention across a set of independently configurable channels. It does so by receiving an interrupt, or trigger through hardware or software to initiate a data transfer. The DMA can be used to read data from the ADC result registers, transfer data to or from memory blocks, and transfer data to or from various peripherals.

AM13E230x devices support 12 DMA channels separated into 6 BASIC and 6 FULL_FEATURE channels. Each channel can be configured for various data sizes, transfer modes, addressing modes, and triggers.

Using the DMA controller can increase the throughput of peripheral modules. It can also reduce system power consumption by allowing the CPU to remain in a low-power mode, without having to awaken to move data to or from a peripheral.

DMA controller features include:

- Up to 12 independent transfer channels
- Byte (8-bit), half-word (16-bit), word (32-bit), long-word (64-bit), and long-long-word (128-bit) or mixed byte and word transfer capability
- Single or block transfer modes
- Seven flexible addressing modes
- Transfer counter block size supports up to 64k transfers of any data type
- Configurable DMA transfer trigger selection
- Configurable DMA channel priorities
- DMA trigger support in various operating modes

The DMA controller block diagram is shown in Figure 9-1.

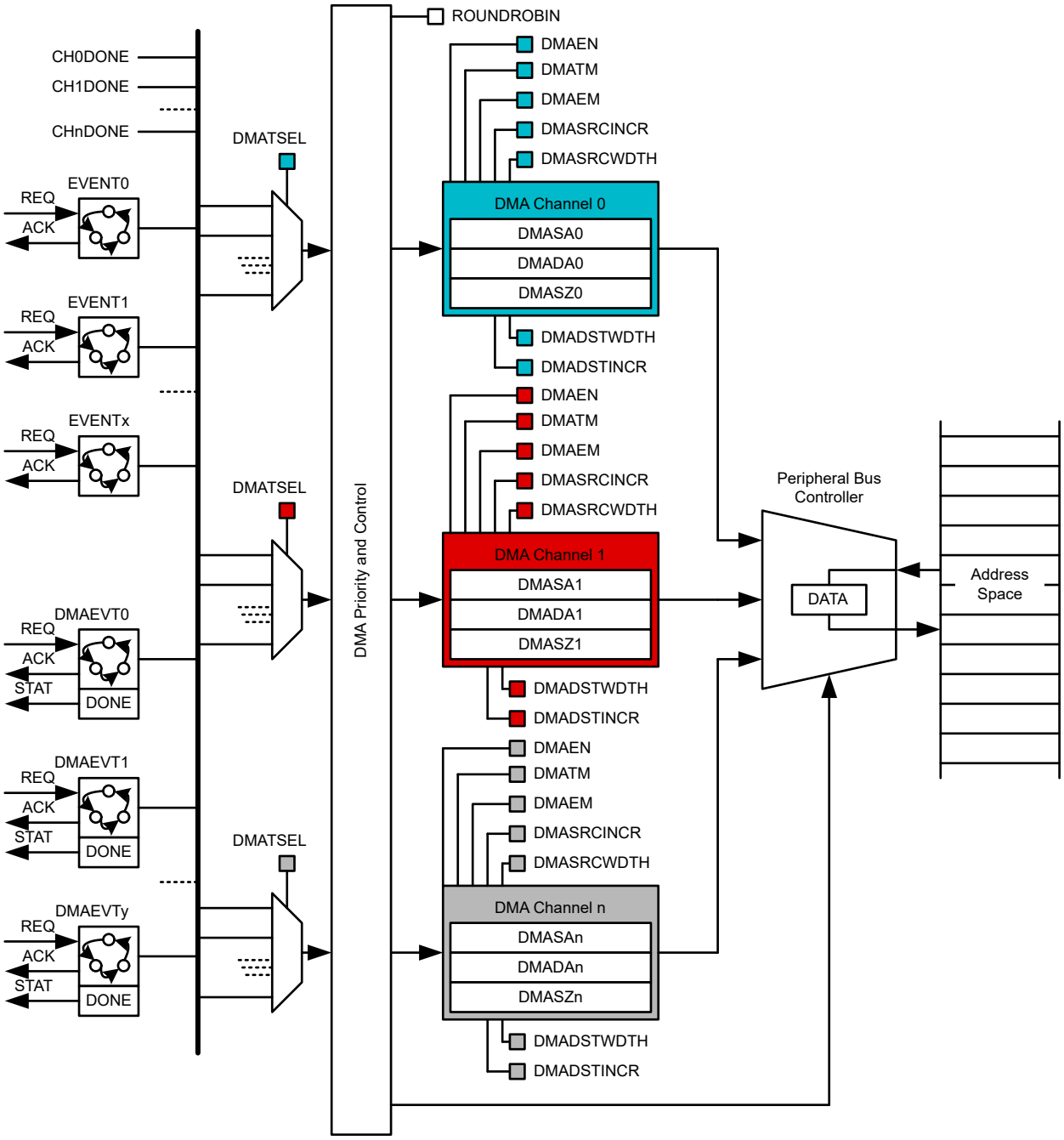


Figure 9-1. DMA Block Diagram

9.2 DMA Operation

For each DMA channel, the DMA controller utilizes a source address pointer (DMASA), a destination address pointer (DMADA), and a transfer counter (DMASZ). After the DMA channel is configured, the channel must be enabled by setting the DMA_CTL.DMAEN bit to begin transfers. With each transfer, the DMA reads a data element from the source and writes the element at the destination. After the transfer, the source and the destination addresses are updated based on the source increment (DMASRCINCR) and destination increment (DMADSTINCR) registers, and the DMASZ is decremented. The DMA loops through this cycle until the DMASZ reaches zero, at which point the DMA channel generates a raw interrupt status (RIS) flag.

The full setup and operation of the DMA is discussed in the following sections.

9.2.1 Channel Types

The DMA module has **six** channels supporting basic (BASIC) features and **six** channels with full-featured (FULL) support. BASIC channels support only single or block transfers, and FULL channels additionally support repeated single transfers, repeated block transfers, and features such as early interrupt request generation and extended table, fill, and gather modes.

The highest priority DMA channels, (DMA0 - DMA5) are FULL channels, and the remaining lower priority channels are BASIC channels (DMA6 - DMA11).

Table 9-1 shows the features supported in the available basic and full-feature DMA channel types.

Table 9-1. Feature Comparison of BASIC and FULL Channels

DMA Feature	Full-Feature	Basic Channel
Single/Block transfer mode	✓	✓
Repeated transfer mode	✓	–
Block burst mode	✓	✓
4 Addressing modes	✓	✓
Stride mode	✓	✓
Extended modes	✓	–
Early IRQ notification	✓	–
Auto enable	✓	✓
Configurable Trigger Selection	✓	✓
Cascading channel support	✓	✓

9.2.2 Channel Priorities

The default DMA channel priorities are DMA0 through DMA15. If two or three triggers happen simultaneously or are pending, the channel with the highest priority completes its transfer (single or block transfer) first, then the second priority channel, then the third priority channel. Transfers in progress are not halted if a higher-priority channel is triggered. The higher-priority channel waits until the transfer in progress completes before starting.

The DMA channel priorities are configurable with the ROUNDROBIN bit of the DMAPRIO register. When the ROUNDROBIN bit is set, the channel that completes a transfer becomes the lowest priority. The order of the priority of the channels always stays the same, DMA0-DMA1-DMA2, for example, for three channels. When the ROUNDROBIN bit is cleared, the channel priority returns to the default priority.

Table 9-2. Round-Robin DMA Priority Example Using 3 DMA Channels

Current DMA Priority	Transfer Occurs	New DMA Priority
DMA0 - DMA1 - DMA2	DMA1	DMA2 - DMA0 - DMA1
DMA2 - DMA0 - DMA1	DMA2	DMA0 - DMA1 - DMA2
DMA2 - DMA0 - DMA1	DMA0	DMA1 - DMA2 - DMA0

9.2.3 Initiating DMA Transfers

To begin a transfer, the DMA controller requires a trigger. Each DMA channel is independently configured for a specific trigger source within the correlating DMA trigger control registers.

Note

The trigger control registers must be modified only when the DMA channel is disabled, otherwise unpredictable DMA triggers can occur.

The DMA can be configured for either an external trigger or an internal channel trigger via the DMATINT bits of the DMA Trigger Select register (DMATCTL). The DMATSEL bits then configure the exact trigger. See *AM13E230x DMA Triggers* for the list of external triggers available, along with the respective DMATSEL values.

When configuring for an internal channel trigger, the DMATSEL bits must be set to the desired DMA channel index.

Note

The recommended software initialization sequence is to configure the DMA channel trigger source before the triggering peripheral is enabled to avoid triggering the DMA before the channel is fully configured. For DMA channels configured with DMA_TRIG_RX trigger sources, clear the RX FIFO of the UNICOMM before enabling or re-enabling the DMA channel.

When an enabled DMA channel receives a trigger from it's trigger source, the DMA controller begins execution of the data transfer. For both trigger types, completion of activity occurs when a DMA channel's size (DMASZ) counter reaches zero.

9.2.3.1 DMA - DMA Trigger Source Options

Table 9-3. DMA Trigger Source Options

Select Index	Trigger Source	Trigger Type Mapping
0	DMA_SOFTWARE_TRIGGER (Tie low)	Reserved
1	ADC0_DMA_TRIG1	DMA_FIX_TRIG_SEL[0]
2	ADC0_DMA_TRIG2	DMA_FIX_TRIG_SEL[1]
3	ADC0_DMA_TRIG3	DMA_FIX_TRIG_SEL[2]
4	ADC0_DMA_TRIG4	DMA_FIX_TRIG_SEL[3]
5	ADC1_DMA_TRIG1	DMA_FIX_TRIG_SEL[4]
6	ADC1_DMA_TRIG2	DMA_FIX_TRIG_SEL[5]
7	ADC1_DMA_TRIG3	DMA_FIX_TRIG_SEL[6]
8	ADC1_DMA_TRIG4	DMA_FIX_TRIG_SEL[7]
9	ADC2_DMA_TRIG1	DMA_FIX_TRIG_SEL[8]
10	ADC2_DMA_TRIG2	DMA_FIX_TRIG_SEL[9]
11	ADC2_DMA_TRIG3	DMA_FIX_TRIG_SEL[10]
12	ADC2_DMA_TRIG4	DMA_FIX_TRIG_SEL[11]
13	PWM0_SOCA	DMA_FIX_TRIG_SEL[12]
14	PWM0_SOCB	DMA_FIX_TRIG_SEL[13]
15	PWM0_SOCC	DMA_FIX_TRIG_SEL[14]
16	PWM0_SOCD	DMA_FIX_TRIG_SEL[15]
17	PWM1_SOCA	DMA_FIX_TRIG_SEL[16]
18	PWM1_SOCB	DMA_FIX_TRIG_SEL[17]
19	PWM1_SOCC	DMA_FIX_TRIG_SEL[18]
20	PWM1_SOCD	DMA_FIX_TRIG_SEL[19]
21	PWM2_SOCA	DMA_FIX_TRIG_SEL[20]
22	PWM2_SOCB	DMA_FIX_TRIG_SEL[21]
23	PWM2_SOCC	DMA_FIX_TRIG_SEL[22]
24	PWM2_SOCD	DMA_FIX_TRIG_SEL[23]
25	PWM3_SOCA	DMA_FIX_TRIG_SEL[24]
26	PWM3_SOCB	DMA_FIX_TRIG_SEL[25]
27	PWM3_SOCC	DMA_FIX_TRIG_SEL[26]
28	PWM3_SOCD	DMA_FIX_TRIG_SEL[27]
29	PWM4_SOCA	DMA_FIX_TRIG_SEL[28]
30	PWM4_SOCB	DMA_FIX_TRIG_SEL[29]
31	PWM4_SOCC	DMA_FIX_TRIG_SEL[30]
32	PWM4_SOCD	DMA_FIX_TRIG_SEL[31]

Table 9-3. DMA Trigger Source Options (continued)

Select Index	Trigger Source	Trigger Type Mapping
33	Reserved (Tie low)	DMA_FIX_TRIG_SEL[32]
34	TIMG4_0.GEN_EVENT1	DMA_FIX_TRIG_SEL[33]
35	Reserved (Tie low)	DMA_FIX_TRIG_SEL[34]
36	TIMG12_0.GEN_EVENT1	DMA_FIX_TRIG_SEL[35]
37	AES.INT_EVENT1	DMA_FIX_TRIG_SEL[36]
38	AES.INT_EVENT2	DMA_FIX_TRIG_SEL[37]
39	ECAP1DMA	DMA_FIX_TRIG_SEL[38]
40	ECAP2DMA	DMA_FIX_TRIG_SEL[39]
41	GPIO0.GEN_EVENT0	DMA_FIX_TRIG_SEL[40]
42	GPIO0.GEN_EVENT1	DMA_FIX_TRIG_SEL[41]
43	GPIO1.GEN_EVENT0	DMA_FIX_TRIG_SEL[42]
44	GPIO1.GEN_EVENT1	DMA_FIX_TRIG_SEL[43]
45	GPIO2.GEN_EVENT0	DMA_FIX_TRIG_SEL[44]
46	GPIO2.GEN_EVENT1	DMA_FIX_TRIG_SEL[45]
47	GPIO3.GEN_EVENT0	DMA_FIX_TRIG_SEL[46]
48	Reserved (Tie low)	DMA_FIX_TRIG_SEL[47]
49	Reserved (Tie low)	DMA_FIX_TRIG_SEL[48]
50	Reserved (Tie low)	DMA_FIX_TRIG_SEL[49]
51	Reserved (Tie low)	DMA_FIX_TRIG_SEL[50]
52	TINIE_LITE	DMA_FIX_TRIG_SEL[51]
53	UC0_TX	DMA_TRIG_SEL[0]
54	UC0_RX	DMA_TRIG_SEL[1]
55	UC1_TX	DMA_TRIG_SEL[2]
56	UC1_RX	DMA_TRIG_SEL[3]
57	UC2_TX	DMA_TRIG_SEL[4]
58	UC2_RX	DMA_TRIG_SEL[5]
59	UC3_TX	DMA_TRIG_SEL[6]
60	UC3_RX	DMA_TRIG_SEL[7]
61	UC4_TX	DMA_TRIG_SEL[8]
62	UC4_RX	DMA_TRIG_SEL[9]
63	UC5_TX	DMA_TRIG_SEL[10]
64	UC5_RX	DMA_TRIG_SEL[11]

9.2.3.2 Cascading DMA Channels

DMA channels can be internally triggered upon the completion of activity on another DMA channel to support cascading. This is beneficial for applications where data can be retrieved, transferred, and/or error-checked without an interrupt or event configuration.

For example, if UART data is received and transmitted to SRAM through DMA channel 0 and DMATSEL is set to UART RX, then DMA channel 1 can be internally triggered when the UART is finished receiving data. If DMA channel 1 is configured to transmit the data from SRAM to CRC, then the DMA transfer will trigger once the UART data is received. In this case, the DMA channels are cascaded from Channel 0 to Channel 1.

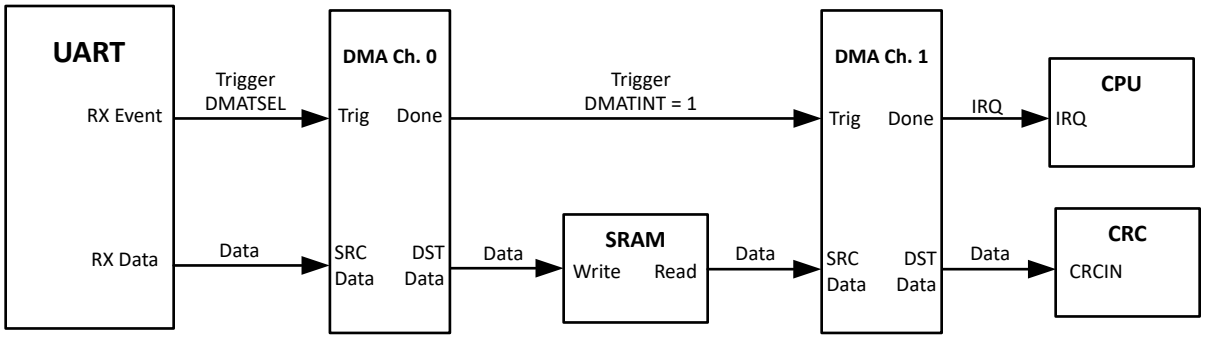


Figure 9-2. DMA Cascading Channels

9.2.4 Transfer Modes

The DMA has four transfer modes that are selected by the transfer mode (DMATM) bits in the DMA channel control registers. These transfer modes are listed in Table 9-4. The transfer mode of each channel is individually configurable. For example, channel 0 can be configured in repeated block transfer mode, while channel 1 is configured for block transfer mode, and channel 2 operates in single transfer mode. The transfer mode (single, block, repeat) is configured independently from the addressing mode (decrement, fixed, increment, stride). Any addressing mode can be used with any transfer mode.

The five types of data that can be transferred are selectable by the destination width (DMADSTWDTH) and source width (DMASRCWDTH) control bits. The source and destination locations can be either byte (8-bit), half-word (16-bit), word (32-bit), long-word (64-bit), or long-long-word (128-bit) data. Transfers can be byte to byte, half-word to half-word, word to word, long-word to long-word, long-long-word to long-long-word or any combination of the five. When transferring a wider bit width source to a shorter bit width destination, only the lower bits of the destination data transfers. When transferring a shorter bit width source to a wider bit width destination, the upper bytes of the destination data is cleared when the transfer occurs. There is no packing or unpacking support by combining several source byte transfers to one single destination word or the reverse.

Table 9-4. DMA Transfer Modes

DMATM	Transfer Mode	Description	Channel Type
0	Single transfer	Each transfer requires a trigger. DMAEN is automatically cleared when DMASZ transfers have been made.	Basic or Full-Feature
1	Block transfer	A complete block of individual single transfers is transferred with one trigger. DMASZ is equal to the block size (in amount of single transfers) and decrements after each single transfer of data. DMAEN is automatically cleared at the end of the block transfer (when DMASZ = 0).	Basic or Full-Feature
2	Repeated single transfer	Each transfer requires a trigger. The DAMSA, DMADA and DMASZ registers are reloaded to the original value when the DMASZ counted down to zero. DMAEN remains enabled.	Full-feature
3	Repeated block transfer	A complete block is transferred with one trigger and continuous transferring. The DAMSA, DMADA and DMASZ registers are reloaded to the original value when the DMASZ counted down to zero. DMAEN remains enabled.	Full-feature

9.2.4.1 Single Transfer

In single transfer mode (DMATM = 0), each byte, half-word, word, long-word, or long-long-word transfer requires a separate trigger.

The DMASZ register defines the number of transfers to be made. The DMADSTINCR and DMASRCINCR bits select if the destination address and the source address's are incremented or decremented after each transfer. If DMASZ = 0, no transfers occur. The DMASA and DMADA registers are incremented or decremented after each transfer. DMASZ is decremented after each transfer. The DMADSTWDTH indicates whether the destination address increments or decrements by 1, 2, 4, 8, or 16 with each transfer cycle. The same is true for

the DMASRCWDTH and the source address respectively. When the DMASZ register decrements to zero, the RIS.DMACHx flag is set.

The DMAEN bit is cleared automatically when DMASZ decrements to zero and must be set again for another transfer to occur.

9.2.4.2 Block Transfer

In block transfer mode (DMATM = 1), a transfer of a complete block of data occurs after one trigger.

The DMASZ register defines the size of the block in the number of single transfers of the specified data type, and the DMADSTINCR and DMASRCINCR bits select if the destination address and the source address are incremented or decremented after each transfer of the block. The DMA is done transferring the block once DMASZ = 0, which sets the corresponding RIS.DMACHx flag. The DMADSTWDTH indicates whether the destination address increments or decrements by 1, 2, 4, 8 or 16 with each transfer cycle. The same is true for the DMASRCWDTH and the source address respectively. The DMASZ register is decremented after each transfer of one data type and shows the number of single transfers remaining.

The DMAEN bit is cleared after the completion of the block transfer and must be set again before another block transfer can be triggered. After a block transfer has started, any other trigger signal that occurs during the block transfer is ignored until the block transfer is complete.

9.2.4.3 Repeated Single Transfer

Repeated single transfer mode is only available in FULL feature DMA channels.

In repeated single transfer mode (DMATM = 2), the DMA controller remains enabled with DMAEN = 1, and a transfer occurs every time a trigger occurs.

The DMASA, DMADA, and DMASZ registers are copied into internal hidden registers. The values of DMASA and DMADA are incremented or decremented after each transfer. The DMASZ register is decremented after each transfer. The DMADSTWDTH indicates whether the destination address increments or decrements by 1, 2, 4, 8 or 16 with each transfer cycle. The same is true for the DMASRCWDTH and the source address respectively. When the DMASZ register decrements to zero, the intialized value is reloaded from an internal register and the corresponding RIS.DMACHx flag is set. The DMA channel remains enabled and waits for another trigger before starting the next transfer.

Note

When using repeated single transfer mode, the DMA does not support pausing and continuing a transfer by disabling a channel (to pause) and then re-enabling the channel (to continue).

9.2.4.4 Repeated Block Transfer

Repeated block transfer mode is only available in FULL feature DMA channels.

In repeated block transfer mode (DMATM = 3), the DMAEN bit remains set after completion of the block transfer. The next trigger after the completion of a repeated block transfer starts another block transfer.

The DMASA, DMADA, and DMASZ registers are copied into internal hidden registers. The temporary values of DMASA and DMADA are incremented or decremented after each transfer in the block. The DMADSTWDTH indicates whether the destination address increments or decrements by 1, 2, 4, 8 or 16 with each transfer cycle. The same is true for the DMASRCWDTH and the source address respectively. The DMASZ register is decremented after each transfer of the block and shows the number of transfers remaining in the block. When the DMASZ register decrements to zero, the initialized value is reloaded from an internal register and the corresponding RIS.DMACHx flag is set. The DMA channel remains enabled and waits for another trigger before starting the next transfer.

9.2.4.5 Burst Block Mode

By default, once a block transfer begins, the DMA does not service any other incoming triggers until the transfer-in-progress is complete. In the case of larger block transfers, this can cause a delay in servicing higher

priority channels. As such, the DMA module supports a burst block mode for suspending an active channel after a configurable number of transfers to service other pending channels. The burst block size is configurable by setting DMAPRIO.BURSTSZ to 8, 16, 32, or an infinite number of transfers (essentially the full block size). When using Burst Block mode, the DMA will pause an in-progress block transfer after the configured burst block size and check for pending triggers. If a higher priority channel is pending after the burst block, the DMA will execute the higher priority channel and resume on the suspended channel once the higher priority channel is complete. If no other channel is pending, the priority logic assigns the control back to the original block transfer for the next burst.

9.2.5 Pausing DMA Transfers

A DMA block transfer in progress can be paused by clearing the DMAEN bit. The DMA stops after the completion of the ongoing transfer cycle and all the channel registers stay in the current state during a channel pause. The block transfer can be continued as originally configured after setting the DMAEN bit again and resending the trigger. Please note that an additional trigger is necessary for halted transfer to resume.

Note

A single transfer in progress cannot be interrupted.

9.2.6 DMA Auto-enable

All DMA channels can be set to automatically enable based on a register write to DMASA, DMADA, or DMASZ. The automatic DMA channel enable source is configured by the DMAAUTOEN bits of the DMA channel control register.

This allows application software to only have to modify one register when updating a DMA channel configuration as opposed to modifying DMASA, DMADA, or DMASZ and then also re-enabling the DMA channel by setting DMACTL.DMAEN

9.2.7 Addressing Modes

The DMA module has seven addressing modes: 4 basic addressing modes available to all channels and 3 extended modes available on full-feature channels. The addressing mode for each DMA channel is independently configurable. For example, channel 0 can transfer between two fixed addresses, while channel 1 transfers between two blocks of addresses.

The 4 basic addressing modes are:

1. Fixed address to fixed address
2. Fixed address to block of addresses
3. Block of addresses to fixed address
4. Block of addresses to block of addresses

The 3 extended modes are:

1. Fill data to block of addresses
2. Data table to specific address
3. Gather data from address table to fixed address or block of addresses

The configuration and operation of the addressing modes are detailed in the following sections.

9.2.7.1 Basic Addressing Modes

The basic addressing modes available on channel types are configured with the source increment (DMASRCINCR) and destination increment (DMADSTINCR) control bits within the respective channel control registers. The DMASRCINCR bits select if the source address is incremented, decremented, or unchanged after each transfer. The DMADSTINCR bits select if the destination address is incremented, decremented, or unchanged after each transfer. The basic addressing modes are showcased in DMA Basic Addressing Modes. The configuration settings for the basic addressing modes are in DMA Basic Addressing Mode Configuration.

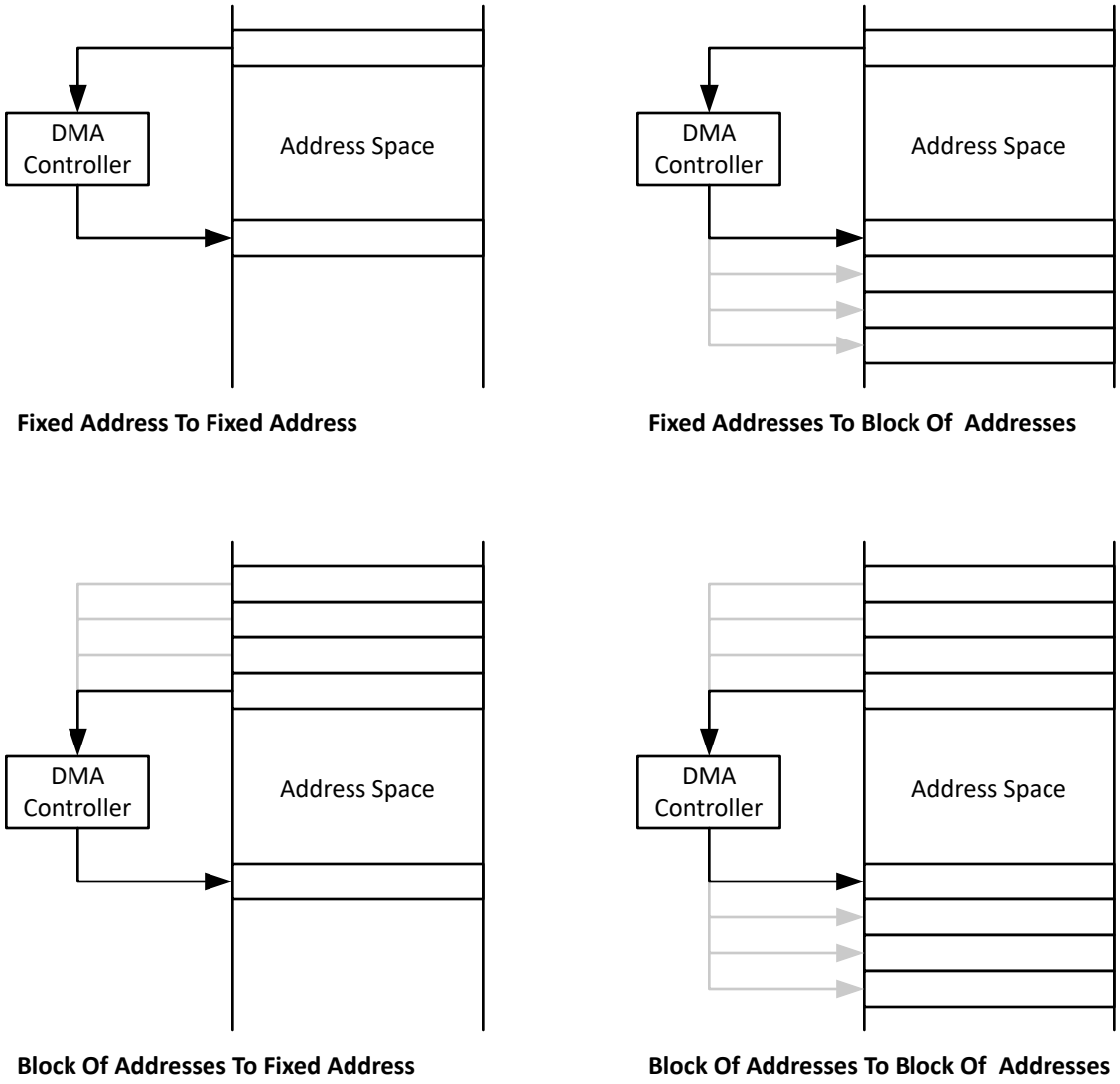


Figure 9-3. DMA Basic Addressing Modes

Table 9-5. DMA Basic Addressing Mode Configuration

Addressing Mode	DMASRCINCR	DMADSTINCR	Recommended Usage	Example
Fixed address to fixed address	Unchanged	Unchanged	Peripheral to peripheral	Transfer ADCRESULT0 register value to UNICOMM-UART TXDATA register
Fixed address to block of addresses	Unchanged	Increment/Decrement	Peripheral to memory	UNICOMM Receive operation - Transfer RXDATA register value to a buffer in memory
Block of addresses to fixed address	Increment/Decrement	Unchanged	Memory to peripheral	UNICOMM Transmit operation - Transfer buffer in memory to TXDATA register

Table 9-5. DMA Basic Addressing Mode Configuration (continued)

Addressing Mode	DMASRCINCR	DMADSTINCR	Recommended Usage	Example
Block of addresses to block of addresses	Increment/Decrement	Increment/Decrement	Memory to memory	Transfer memory buffer contents to another memory buffer

9.2.7.2 Stride Mode

To make data organization easier, all channels support a “stride” mode where the DMA source and destination addresses can be incremented to a higher value (rather than +1) after a transfer. This is helpful for re-organizing the order of data between the source and destination.

To support incremental strides, set the DMADSTINCR and/or DMASRCINCR to STRIDE_n, where n is the number of destination and/or source increments. The real address increment values for the source and destination addresses are based on the definitions of DMASRCWDTH and DMADSTWDTH respectively. The formulas for calculating the incremental strides are:

- DMASRCINCR = (STRIDE_n * DMASRCWDTH)
- DMADSTINCR = (STRIDE_n * DMADSTWDTH)

Note

Stride mode is increment only.

9.2.7.3 Extended Modes

FULL channels have 3 extended modes selected by the DMAEM bits as listed in Table 9-6. A graphical overview of the extended modes can be seen in Figure 9-4. Each channel is individually configurable for a specific extended mode. For example, channel 0 can be configured in table mode while channel 1 is configured in fill mode. These extended modes are each designed with a specific use case in mind, the recommendation is to configure the DMASA, DMADA, DMASRCINCR, DMADSTINCR, DMASRCWDTH, and DMADSTWDTH registers/fields as described in the below sections for each mode so the intended operation to occurs.

Table 9-6. DMA Extended Modes

DMAEM	Extended Mode	Description
0	Normal mode	Operation is defined by the transfer mode (DMATM) and basic addressing mode configurations.
1	Gather mode	Used to read data from an address table and copy to configurable address
2	Fill mode	Used to fill predefined data patterns into memory
3	Table mode	Used to help configure a table of peripheral control registers

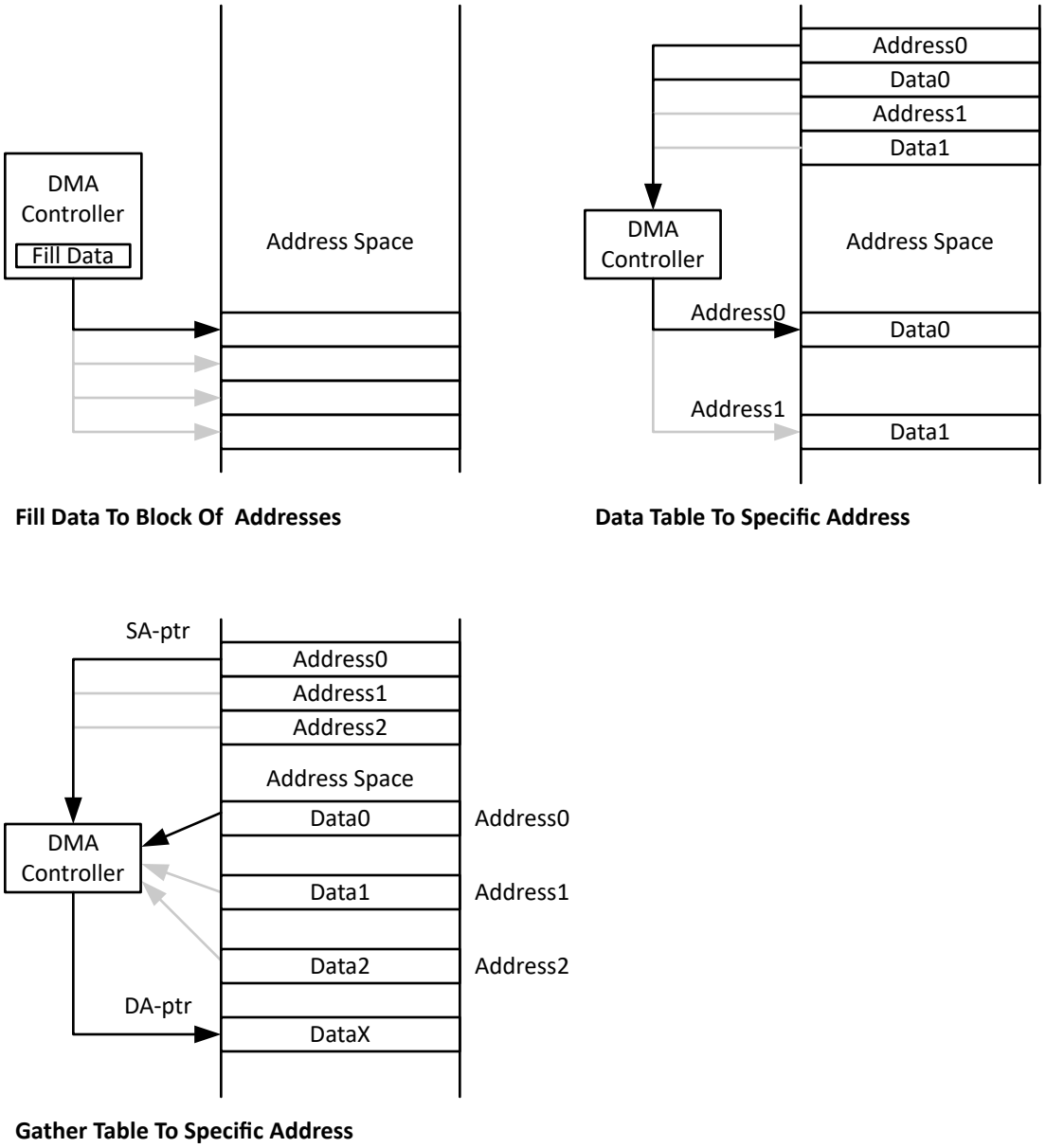


Figure 9-4. DMA Extended Modes Diagram

9.2.7.3.1 Fill Mode

In fill mode (DMAEM = 2h), the DMA module takes a predefined FILL value/pattern and writes this to a user defined segment of memory. The DMATM register setting is ignored and the "block transfer" mode is used automatically. The intended use case for this mode is to initialize or re-initialize a memory buffer with values (for example all zeros or incrementing data) using the DMA, without the need for a software loop doing each write individually via the CPU.

In other modes, the 32-bit value programmed to the DMASA register is used as a pointer to a memory location, whereas in FILL mode, the value programmed to the DMASA register is used directly as the FILL data. The DMASRCINCR bit field is used to indicate whether the FILL data remains constant or is incremented/ decremented with every write cycle. This feature allows for filling a memory block with a sequential pattern. The DMASRCWDTH bit field indicates the magnitude to increment of the FILL mode data. Refer to Table 9-7 for how to use DMASRCWDTH in fill mode.

The benefit of doing this operation using FILL mode (DMAEN=2) rather than normal mode with block transfer configured (DMAEN=2, DMATM=1), is that FILL mode has an improved performance due to the fact that the transfer of each individual element in normal mode requires the DMA to read and write, whereas in FILL mode, the fill data is stored inside the DMASA register, so each element only requires a write operation.

Table 9-7. DMASRCWDTH in Fill Mode

DMASRCWDTH	FILL Mode Data Increment Value
0	±1
1	±2
2	±4
3	±8

The channel destination registers and bit fields DMADA, DMADSTINCR, and DMADSTWDTH all behave as expected and influence where and how in memory the FILL pattern is written.

9.2.7.3.2 Table Mode

In table mode (DMAEM = 3h), the DMA controller executes two 32-bit reads from the source and one 32-bit write to a determined destination. Table mode allows the DMA to parse through a table of addresses and data to configure peripheral memory mapped registers in a single block transfer. This feature can be leveraged to interpret a table of addresses and data and use the DMA to efficiently program that data to the associated addresses without CPU intervention. For example, to configure a peripheral to a default setting after an interrupt or initialize/re-initialize a peripheral's settings after entering a low power mode.

The DMASRCWDTH bit field must be set to "3" (64-bit mode) and the DMADSTWDTH bit field must be set to "2" (32-bit mode). This is to make sure that the DMA controller reads the table correctly (two 32-bit reads) and writes the table data correctly (one 32-bit write). The DMASRCINCR bit field can be set to either increment or decrement the source address after transfers. DMASZ is set to represent the number of entries in the table and DMATM must be set for block transfer mode. The DMADA and DMADSTINCR are ignored in table mode and can be treated as "don't care" values as the destination write is based on the information in the table.

For table mode to work properly, the DMASA register must be programmed with the start address of the table, which must be aligned to 64-bit data (DMASA[2:0] = "000b"). The address stored in the table needs to be on the lower word of a 64-bit data (ADDR[2:0] = "000b") while the data needs to be on the upper word of a 64-bit data (ADDR[2:0] = "100b"). Table 9-8 is an example of a table in memory compatible with the DMA table mode.

Table 9-8. Example of an Incremental Table Compatible with DMA Table Mode

Table Address	Table Data
0x0000	Address 0
0x0004	Data 0
0x0008	Address 1
0x000C	Data 1

The DMA programs the addresses in the order laid out by the table. User must be sure the order of the addresses/data pairs in the table complies with the requirements of the peripheral. For example, if the peripheral requires the register containing an enable bit be written after all other register configurations, make sure this is the last address and data pair in the table.

9.2.7.3.3 Gather Mode

In gather mode (DMAEM = 1h), the DMA controller executes two 32-bit reads and one 32-bit write based on an address table located at the source. Different from table mode, where the table consists of an Address followed by the Data to be written in a 64-bit block, in gather mode the initial table contains a list of source addresses to be read from. The DMA controller reads an address, then reads the data stored at the read address, and then writes the data read to a memory block at a specified destination. The intended use case for this feature is to read all the registers of a specific peripheral and save the state to a buffer in memory. This feature is especially

useful for CRC calculations, memory mapped register validation checks, or to save the register context before entering low power mode.

DMASA points to the beginning of a table containing addresses of data to gather. DMADA points to the memory location used to write (or "gather") the gathered data to. The DMASRCWDTH bit field and the DMADSTWDTH bit fields must be set to "2" (32-bit mode). This is to make sure that the DMA controller reads the gather table correctly (one 32-bit read) and writes the gathered data correctly (one 32-bit write). The DMASRCINCR bit field and DMADSTINCR bit fieldcan be set to either increment or decrement the source address after transfers. DMASZ is set to represent the number of entries in the table and DMATM must be set for block transfer mode.

The basic steps that the DMA controller takes in gather mode are as follows:

1. The DMA controller reads the data stored at the source address (DMASA).
2. The data read at the source address is copied into a temporary address register.
3. The DMA controller reads the gather data from the address stored in the temporary address register.
4. The DMA controller then writes the gathered data to the destination address (DMADA).
5. The DMASA, DMADA, and DMASZ registers are incremented or decremented as configured.

9.2.8 DMA Controller Interrupts

Each DMA channel has its own raw interrupt status (RIS) flag. Each RIS flag is set in any mode when the corresponding DMASZ register counts to zero. If the corresponding interrupt mask (MASK) and RIS bits are set, an interrupt request is generated.

All RIS flags are prioritized, with DMA0 being the highest, and combined to source a single interrupt vector. The highest-priority enabled interrupt generates a number in the interrupt index (IIDX) register. This number can be evaluated or added to the program counter (PC) to automatically enter the appropriate software routine.

Any access, read or write, of the IIDX register automatically resets the highest pending interrupt flag. If another interrupt flag is set, another interrupt is immediately generated after servicing the initial interrupt. For example, assume that DMA0 has the highest priority. If the DMA0-RIS and DMA2-RIS flags are set when the interrupt service routine accesses the IIDX register, DMA0-RIS is reset automatically. After the interrupt service routine is executed, the DMA2-RIS generates another interrupt which can then be serviced.

9.2.8.1 Using DMA with System Interrupts

System interrupt service routines are interrupted by DMA transfers. If an interrupt service routine or other routine must execute with no interruptions, the DMA controller should be disabled before executing the routine.

9.2.9 DMA Trigger Event Status

The DMA controller supports dedicated DMA events. See the *DMA Trigger* section of the Event Manager for details on the DMA event trigger protocol. The idea is that the DMA can inform the event triggering peripheral about the status of the assigned DMA channel. This will allow the triggering peripheral to issue an interrupt itself after the completion of a repeated transfer, instead of the DMA issuing an interrupt event. The advantage is that the DMA interrupt service routine does not need to keep track of the assigned function of the channel and instead leaves that to the triggering peripheral's own interrupt service routine.

The DMA status sent to the triggering peripheral will reflect the value of the DMASZ register. If the last DMA transfer resulted in a size decrement to zero, the DMA will return the status of 1, indicating the end of the transfer. Otherwise the status will be 0.

Additionally, for FULL channels, the DMA module can generate an early interrupt request (IRQ) to the CPU to indicate that a transfer will complete within a configurable number of transfers (1, 2, 4, 8, 32, 64, half-DMASZ).

An early IRQ event is enabled by setting the DMAPREIRQ bits of the channel control register to the desired number of transfers. When the DMA has reached the number of transfers, the corresponding DMA channel’s PREIRQ interrupt is set.

Early DMA interrupt generation is useful to:

- Reduce the interrupt latency in timing-critical applications where it would be beneficial to let the DMA preemptively generate the IRQ before the DMA transfer is complete
- Serve as a “progress notification” when scheduling other tasks for the CPU to complete
- Transfer weights of a neural network layer and notify the CPU to complete software configuration writes to the IP
- Implement a ping-pong buffer (by setting DMAPREIRQ to half)

9.2.10 DMA Operating Mode Support

The DMA supports triggered transfers in RUN mode, as well as in the SLEEP, STOP, and STANDBY low-power modes. Refer to the following sections for more details.

9.2.10.1 Transfer in RUN Mode

In RUN mode the system is fully operational. The CPU and all other peripherals and resources are available, therefore there is no restriction on the DMA functionality in RUN mode.

9.2.10.2 Transfer in SLEEP Mode

In SLEEP mode only the CPU is halted. All other peripherals and resources are available as when in RUN mode, therefore there is no restriction on the DMA functionality in SLEEP mode. All peripherals that can trigger a DMA transfer in RUN mode will also be able to trigger a DMA transfer in SLEEP mode.

9.2.10.3 Transfer in STOP Mode

In STOP mode the CPU is halted and the ULPCLK is limited to 4MHz operation. PD1 peripherals are disabled and in retention mode when applicable. Only PD0 peripherals are functional and therefore only PD0 peripherals are able to trigger a DMA transfer in STOP mode. The DMA is located in PD1, so although the register settings are retained, the DMA is not functional during STOP mode. If a DMA channel is triggered during this mode, the transfer does not take place until the device leaves STOP mode.

9.2.10.4 Transfers in STANDBY Mode

In STANDBY mode the CPU is halted and the ULPCLK is limited to 32kHz operation. PD1 peripherals are disabled and in retention mode when applicable. Only PD0 peripherals are functional and therefore only PD0 peripherals are able to trigger a DMA transfer in STANDBY mode. The DMA is located in PD1, so although the register settings are retained, the DMA is not functional during STOP mode. If a DMA channel is triggered during this mode, the transfer does not take place until the device leaves STOP mode.

9.2.11 DMA Address and Data Errors

The DMA has the ability to flag address or data errors. Source or destination address errors can come from accessing a protected or nonexisting memory range. If an address error occurs, the interrupt index IIDX[j].STAT flags a DMA address error (11h). Address error interrupts can be masked, set, and cleared using the ADDERR bit.

Note

The DMA does not perform range checking. If the DMA transfer occurs over a protected memory range, the destination data reports zeros (0h) for each byte of the DMA transaction that overlaps the protected or nonexisting memory range.

Data errors can occur in SRAM or flash if there is an ECC or parity error. If a data error occurs, the interrupt index IIDX[j].STAT flags a DMA data error (12h). Data error interrupts can be masked, set, and cleared using the DATERR bit.

Note

The DMA does not automatically disable if an address or data error occurs.

This Section describes the DMA Registers.

10.1 DMA Base Address Table

Table 10-1. DMA Base Address Table

Bit Field Name		DriverLib Name	Base Address
Instance	Structure		
Dma0Regs	DMA_REGS	DMA0	0x4002_0000

10.2 DMA_REGS Registers

Table 10-2 lists the memory-mapped registers for the DMA_REGS registers. All register offset addresses not listed in Table 10-2 should be considered as reserved locations and the register contents should not be modified.

Table 10-2. DMA_REGS Registers

Offset	Acronym	Register Name	Section
400h	FSUB_0	Subscriber Port 0	Go
404h	FSUB_1	Subscriber Port 1	Go
444h	FPUB_1	Publisher Port 0	Go
480h	CPU_CONNECT_0	CPU Connect	Go
1018h	PDBGCTL	Peripheral Debug Control	Go
1020h	IIDX	Interrupt index	Go
1028h	IMASK	Interrupt mask	Go
1030h	RIS	Raw interrupt status	Go
1038h	MIS	Masked interrupt status	Go
1040h	ISET	Interrupt set	Go
1048h	ICLR	Interrupt clear	Go
1050h	IIDX	Interrupt index	Go
1058h	IMASK	Interrupt mask	Go
1060h	RIS	Raw interrupt status	Go
1068h	MIS	Masked interrupt status	Go
1070h	ISET	Interrupt set	Go
1078h	ICLR	Interrupt clear	Go
10E0h	EVT_MODE	Event Mode	Go
10FCh	DESC	Module Description	Go
1100h	DMAPRIO	DMA Channel Priority Control	Go
1110h + formula	DMATCTL_j	DMA Trigger Select	Go
1200h + formula	DMACTL_j	DMA Channel Control	Go
1204h + formula	DMASA_j	DMA Channel Source Address	Go
1208h + formula	DMADA_j	DMA Channel Destination Address	Go
120Ch + formula	DMASZ_j	DMA Channel Size	Go

Complex bit access types are encoded to fit into small table cells. Table 10-3 shows the codes that are used for access types in this section.

Table 10-3. DMA_REGS Access Type Codes

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

Table 10-3. DMA_REGS Access Type Codes (continued)

Access Type	Code	Description
i,j,k,l,m,n		When these variables are used in a register name, an offset, or an address, they refer to the value of a register array where the register is part of a group of repeating registers. The register groups form a hierarchical structure and the array is represented with a formula.
y		When this variable is used in a register name, an offset, or an address it refers to the value of a register array.

1 FSUB_0 Register (Offset = 400h) [Reset = 00000000h]

FSUB_0 is shown in Figure 10-1 and described in Table 10-4.

Return to the Summary Table.

Subscriber port

Figure 10-1. FSUB_0 Register

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

Table 10-5. FSUB_0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R/W	0h	
7-0	CHANID	R/W	0h	0 = disconnected. 1-255 = connected to channelID = CHANID. 0h = A value of 0 specifies that the event is not connected FFh = Consult your device datasheet as the actual allowed maximum may be less than 255.

2 FSUB_1 Register (Offset = 404h) [Reset = 00000000h]

FSUB_1 is shown in Figure 10-2 and described in Table 10-5.

Return to the Summary Table.

Subscriber port

Figure 10-2. FSUB_1 Register

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

Table 10-7. FSUB_1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R/W	0h	
7-0	CHANID	R/W	0h	0 = disconnected. 1-255 = connected to channelID = CHANID. 0h = A value of 0 specifies that the event is not connected FFh = Consult your device datasheet as the actual allowed maximum may be less than 255.

3 FPUB_1 Register (Offset = 444h) [Reset = 00000000h]

FPUB_1 is shown in Figure 10-3 and described in Table 10-6.

Return to the Summary Table.

Publisher port

Figure 10-3. FPUB_1 Register

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

Table 10-9. FPUB_1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R/W	0h	
7-0	CHANID	R/W	0h	0 = disconnected. 1-255 = connected to channelID = CHANID. 0h = A value of 0 specifies that the event is not connected FFh = Consult your device datasheet as the actual allowed maximum may be less than 255.

4 CPU_CONNECT_0 Register (Offset = 480h) [Reset = 00000000h]

CPU_CONNECT_0 is shown in Figure 10-4 and described in Table 10-7.

Return to the Summary Table.

Directly connect peripheral publisher port to application processor

Figure 10-4. CPU_CONNECT_0 Register

31	30	29	28	27	26	25	24
RESERVED							
R/W-0h							
23	22	21	20	19	18	17	16
RESERVED							
R/W-0h							
15	14	13	12	11	10	9	8
RESERVED							
R/W-0h							
7	6	5	4	3	2	1	0
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	CPUSS0_CONN	NWPW_CONN
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 10-11. CPU_CONNECT_0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R/W	0h	
7	RESERVED	R/W	0h	Reserved
6	RESERVED	R/W	0h	Reserved
5	RESERVED	R/W	0h	Reserved
4	RESERVED	R/W	0h	Reserved
3	RESERVED	R/W	0h	Reserved
2	RESERVED	R/W	0h	Reserved
1	CPUSS0_CONN	R/W	0h	CPUSS0 connect bit. 0h = The CPU is not connected. 1h = The CPU is connected.
0	NWPW_CONN	R/W	0h	NWPW connect bit. 0h = The NWPW is not connected. 1h = The NWPW is connected.

5 PDBGCTL Register (Offset = 1018h) [Reset = 00000000h]

PDBGCTL is shown in Figure 10-5 and described in Table 10-8.

Return to the Summary Table.

This register can be used by the software developer to control the behavior of the peripheral relative to the 'Core Halted' input

Figure 10-5. PDBGCTL Register

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

Table 10-13. PDBGCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R/W	0h	
1	SOFT	R/W	1h	Soft halt boundary control. This function is only available, if [FREE] is set to 'STOP' 0h = The peripheral will halt immediately, even if the resultant state will result in corruption if the system is restarted 1h = The peripheral blocks the debug freeze until it has reached a boundary where it can resume without corruption
0	FREE	R/W	1h	Free run control 0h = The peripheral freezes functionality while the Core Halted input is asserted and resumes when it is deasserted. 1h = The peripheral ignores the state of the Core Halted input

6 IIDX Register (Offset = 1020h) [Reset = 00000000h]

IIDX is shown in Figure 10-6 and described in Table 10-9.

Return to the Summary Table.

This register provides the highest priority enabled interrupt index. Value 0x00 means no event pending. Interrupt 1 is the highest priority, IIDX next highest, 4, 8, . . . IIDX^31 is the least priority. That is, the least bit position that is set to 1 denotes the highest priority pending interrupt. The priority order is fixed. However, users can implement their own prioritization schemes using other registers that expose the full set of interrupts that have occurred.
On each read, only one interrupt is indicated. On a read, the current interrupt (highest priority) is automatically cleared by the hardware and the corresponding interrupt flag in RIS [RIS] and MIS [MIS] are cleared as well. After a read from the CPU (not from the debug interface), the register is updated with the next highest priority interrupt, if none are pending, then it should display 0x0.
Note: The number of DMACH is device dependent. Please consult the datasheet of the specific device to map which channel number is implemented.

Figure 10-6. IIDX Register

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

Table 10-15. IIDX Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	
7-0	STAT	R	0h	Interrupt index status 00h = No bit is set means there is no pending interrupt request 01h = DMA Channel 0 size counter reached zero (DMASZ=0). 02h = DMA Channel 1 size counter reached zero (DMASZ=0). 03h = DMA Channel 2 size counter reached zero (DMASZ=0). 04h = DMA Channel 3 size counter reached zero (DMASZ=0). 05h = DMA Channel 4 size counter reached zero (DMASZ=0). 06h = DMA Channel 5 size counter reached zero (DMASZ=0). 07h = DMA Channel 6 size counter reached zero (DMASZ=0). 08h = DMA Channel 7 size counter reached zero (DMASZ=0). 09h = DMA Channel 8 size counter reached zero (DMASZ=0). 0Ah = DMA Channel 9 size counter reached zero (DMASZ=0). 0Bh = DMA Channel 10 size counter reached zero (DMASZ=0). 0Ch = DMA Channel 11 size counter reached zero (DMASZ=0). 0Dh = DMA Channel 12 size counter reached zero (DMASZ=0). 0Eh = DMA Channel 13 size counter reached zero (DMASZ=0). 0Fh = DMA Channel 14 size counter reached zero (DMASZ=0). 10h = DMA Channel 15 size counter reached zero (DMASZ=0). 11h = PRE-IRQ event for DMA Channel 0. 12h = PRE-IRQ event for DMA Channel 1. 13h = PRE-IRQ event for DMA Channel 2. 14h = PRE-IRQ event for DMA Channel 3. 15h = PRE-IRQ event for DMA Channel 4. 16h = PRE-IRQ event for DMA Channel 5. 17h = PRE-IRQ event for DMA Channel 6. 18h = PRE-IRQ event for DMA Channel 7. 19h = DMA address error, SRC address not reachable. 1Ah = DMA data error, SRC data might be corrupted (PAR or ECC error).

7 IMASK Register (Offset = 1028h) [Reset = 00000000h]

IMASK is shown in Figure 10-7 and described in Table 10-10.

Return to the Summary Table.

Interrupt Mask. If a bit is set, then the corresponding interrupt is un-masked. Un-masking the interrupt causes the raw interrupt to be visible in IIDX [IIDX], as well as MIS [MIS].
Note: The number of DMACH is device dependent. Please consult the datasheet of the specific device to map which channel number is implemented.

Figure 10-7. IMASK Register

31	30	29	28	27	26	25	24
RESERVED						DATAERR	ADDRERR
R/W-0h						R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
RESERVED	RESERVED	PREIRQCH5	PREIRQCH4	PREIRQCH3	PREIRQCH2	PREIRQCH1	PREIRQCH0
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
DMACH15	DMACH14	DMACH13	DMACH12	DMACH11	DMACH10	DMACH9	DMACH8
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
DMACH7	DMACH6	DMACH5	DMACH4	DMACH3	DMACH2	DMACH1	DMACH0
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 10-17. IMASK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	RESERVED	R/W	0h	
25	DATAERR	R/W	0h	DMA data error, SRC data might be corrupted (PAR or ECC error). 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
24	ADDRERR	R/W	0h	DMA address error, SRC address not reachable. 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
23	RESERVED	R/W	0h	Reserved
22	RESERVED	R/W	0h	Reserved
21	PREIRQCH5	R/W	0h	Pre-IRQ for Channel 5. Size counter reached Pre-IRQ threshold. 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
20	PREIRQCH4	R/W	0h	Pre-IRQ for Channel 4. Size counter reached Pre-IRQ threshold. 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
19	PREIRQCH3	R/W	0h	Pre-IRQ for Channel 3. Size counter reached Pre-IRQ threshold. 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
18	PREIRQCH2	R/W	0h	Pre-IRQ for Channel 2. Size counter reached Pre-IRQ threshold. 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
17	PREIRQCH1	R/W	0h	Pre-IRQ for Channel 1. Size counter reached Pre-IRQ threshold. 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
16	PREIRQCH0	R/W	0h	Pre-IRQ for Channel 0. Size counter reached Pre-IRQ threshold. 0h = Clear interrupt mask bit 1h = Set interrupt mask bit

Table 10-17. IMASK Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
15	DMACH15	R/W	0h	DMA Channel 15 interrupt signal. Size counter reached zero (DMASZ=0). 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
14	DMACH14	R/W	0h	DMA Channel 14 interrupt signal. Size counter reached zero (DMASZ=0). 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
13	DMACH13	R/W	0h	DMA Channel 13 interrupt signal. Size counter reached zero (DMASZ=0). 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
12	DMACH12	R/W	0h	DMA Channel 12 interrupt signal. Size counter reached zero (DMASZ=0). 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
11	DMACH11	R/W	0h	DMA Channel 11 interrupt signal. Size counter reached zero (DMASZ=0). 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
10	DMACH10	R/W	0h	DMA Channel 10 interrupt signal. Size counter reached zero (DMASZ=0). 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
9	DMACH9	R/W	0h	DMA Channel 9 interrupt signal. Size counter reached zero (DMASZ=0). 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
8	DMACH8	R/W	0h	DMA Channel 8 interrupt signal. Size counter reached zero (DMASZ=0). 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
7	DMACH7	R/W	0h	DMA Channel 7 interrupt signal. Size counter reached zero (DMASZ=0). 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
6	DMACH6	R/W	0h	DMA Channel 6 interrupt signal. Size counter reached zero (DMASZ=0). 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
5	DMACH5	R/W	0h	DMA Channel 5 interrupt signal. Size counter reached zero (DMASZ=0). 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
4	DMACH4	R/W	0h	DMA Channel 4 interrupt signal. Size counter reached zero (DMASZ=0). 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
3	DMACH3	R/W	0h	DMA Channel 3 interrupt signal. Size counter reached zero (DMASZ=0). 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
2	DMACH2	R/W	0h	DMA Channel 2 interrupt signal. Size counter reached zero (DMASZ=0). 0h = Clear interrupt mask bit 1h = Set interrupt mask bit

Table 10-17. IMASK Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
1	DMACH1	R/W	0h	DMA Channel 1 interrupt signal. Size counter reached zero (DMASZ=0). 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
0	DMACH0	R/W	0h	DMA Channel 0 interrupt signal. Size counter reached zero (DMASZ=0). 0h = Clear interrupt mask bit 1h = Set interrupt mask bit

8 RIS Register (Offset = 1030h) [Reset = 00000000h]

RIS is shown in Figure 10-8 and described in Table 10-11.

Return to the Summary Table.

Raw interrupt status. Reflects all pending interrupts, regardless of masking. The RIS register allows the user to implement a poll scheme. A flag set in this register can be cleared by writing 1 to the ICLR [ICLR] register bit even if the corresponding IMASK [IMASK] bit is not enabled.
Note: The number of DMACH is device dependent. Please consult the datasheet of the specific device to map which channel number is implemented.

Figure 10-8. RIS Register

31	30	29	28	27	26	25	24
RESERVED						DATAERR	ADDRERR
R-0h						R-0h	R-0h
23	22	21	20	19	18	17	16
RESERVED	RESERVED	PREIRQCH5	PREIRQCH4	PREIRQCH3	PREIRQCH2	PREIRQCH1	PREIRQCH0
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
15	14	13	12	11	10	9	8
DMACH15	DMACH14	DMACH13	DMACH12	DMACH11	DMACH10	DMACH9	DMACH8
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
7	6	5	4	3	2	1	0
DMACH7	DMACH6	DMACH5	DMACH4	DMACH3	DMACH2	DMACH1	DMACH0
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

Table 10-19. RIS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	RESERVED	R	0h	
25	DATAERR	R	0h	DMA data error, SRC data might be corrupted (PAR or ECC error). 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
24	ADDRERR	R	0h	DMA address error, SRC address not reachable. 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
23	RESERVED	R	0h	Reserved
22	RESERVED	R	0h	Reserved
21	PREIRQCH5	R	0h	Pre-IRQ for Channel 5. Size counter reached Pre-IRQ threshold. 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
20	PREIRQCH4	R	0h	Pre-IRQ for Channel 4. Size counter reached Pre-IRQ threshold. 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
19	PREIRQCH3	R	0h	Pre-IRQ for Channel 3. Size counter reached Pre-IRQ threshold. 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
18	PREIRQCH2	R	0h	Pre-IRQ for Channel 2. Size counter reached Pre-IRQ threshold. 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
17	PREIRQCH1	R	0h	Pre-IRQ for Channel 1. Size counter reached Pre-IRQ threshold. 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
16	PREIRQCH0	R	0h	Pre-IRQ for Channel 0. Size counter reached Pre-IRQ threshold. 0h = Clear interrupt mask bit 1h = Set interrupt mask bit

Table 10-19. RIS Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
15	DMACH15	R	0h	DMA Channel 15 interrupt signals that size counter reached zero (DMASZ=0). 0h = Interrupt did not occur 1h = Interrupt occurred
14	DMACH14	R	0h	DMA Channel 14 interrupt signals that size counter reached zero (DMASZ=0). 0h = Interrupt did not occur 1h = Interrupt occurred
13	DMACH13	R	0h	DMA Channel 13 interrupt signals that size counter reached zero (DMASZ=0). 0h = Interrupt did not occur 1h = Interrupt occurred
12	DMACH12	R	0h	DMA Channel 12 interrupt signals that size counter reached zero (DMASZ=0). 0h = Interrupt did not occur 1h = Interrupt occurred
11	DMACH11	R	0h	DMA Channel 11 interrupt signals that size counter reached zero (DMASZ=0). 0h = Interrupt did not occur 1h = Interrupt occurred
10	DMACH10	R	0h	DMA Channel 10 interrupt signals that size counter reached zero (DMASZ=0). 0h = Interrupt did not occur 1h = Interrupt occurred
9	DMACH9	R	0h	DMA Channel 9 interrupt signals that size counter reached zero (DMASZ=0). 0h = Interrupt did not occur 1h = Interrupt occurred
8	DMACH8	R	0h	DMA Channel 8 interrupt signals that size counter reached zero (DMASZ=0). 0h = Interrupt did not occur 1h = Interrupt occurred
7	DMACH7	R	0h	DMA Channel 7 interrupt signals that size counter reached zero (DMASZ=0). 0h = Interrupt did not occur 1h = Interrupt occurred
6	DMACH6	R	0h	DMA Channel 6 interrupt signals that size counter reached zero (DMASZ=0). 0h = Interrupt did not occur 1h = Interrupt occurred
5	DMACH5	R	0h	DMA Channel 5 interrupt signals that size counter reached zero (DMASZ=0). 0h = Interrupt did not occur 1h = Interrupt occurred
4	DMACH4	R	0h	DMA Channel 4 interrupt signals that size counter reached zero (DMASZ=0). 0h = Interrupt did not occur 1h = Interrupt occurred
3	DMACH3	R	0h	DMA Channel 3 interrupt signals that size counter reached zero (DMASZ=0). 0h = Interrupt did not occur 1h = Interrupt occurred
2	DMACH2	R	0h	DMA Channel 2 interrupt signals that size counter reached zero (DMASZ=0). 0h = Interrupt did not occur 1h = Interrupt occurred

Table 10-19. RIS Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
1	DMACH1	R	0h	DMA Channel 1 interrupt signals that size counter reached zero (DMASZ=0). 0h = Interrupt did not occur 1h = Interrupt occurred
0	DMACH0	R	0h	DMA Channel 0 interrupt signals that size counter reached zero (DMASZ=0). 0h = Interrupt did not occur 1h = Interrupt occurred

9 MIS Register (Offset = 1038h) [Reset = 00000000h]

MIS is shown in Figure 10-9 and described in Table 10-12.

Return to the Summary Table.

Masked interrupt status. This is an AND of the IMASK [IMASK] and RIS [RIS] registers.
Note: The number of DMACH is device dependent. Please consult the datasheet of the specific device to map which channel number is implemented.

Figure 10-9. MIS Register

31	30	29	28	27	26	25	24
RESERVED						DATAERR	ADDRERR
R-0h						R-0h	R-0h
23	22	21	20	19	18	17	16
RESERVED	RESERVED	PREIRQCH5	PREIRQCH4	PREIRQCH3	PREIRQCH2	PREIRQCH1	PREIRQCH0
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
15	14	13	12	11	10	9	8
DMACH15	DMACH14	DMACH13	DMACH12	DMACH11	DMACH10	DMACH9	DMACH8
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
7	6	5	4	3	2	1	0
DMACH7	DMACH6	DMACH5	DMACH4	DMACH3	DMACH2	DMACH1	DMACH0
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

Table 10-21. MIS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	RESERVED	R	0h	
25	DATAERR	R	0h	DMA data error, SRC data might be corrupted (PAR or ECC error). 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
24	ADDRERR	R	0h	DMA address error, SRC address not reachable. 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
23	RESERVED	R	0h	Reserved
22	RESERVED	R	0h	Reserved
21	PREIRQCH5	R	0h	Pre-IRQ for Channel 5. Size counter reached Pre-IRQ threshold. 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
20	PREIRQCH4	R	0h	Pre-IRQ for Channel 4. Size counter reached Pre-IRQ threshold. 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
19	PREIRQCH3	R	0h	Pre-IRQ for Channel 3. Size counter reached Pre-IRQ threshold. 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
18	PREIRQCH2	R	0h	Pre-IRQ for Channel 2. Size counter reached Pre-IRQ threshold. 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
17	PREIRQCH1	R	0h	Pre-IRQ for Channel 1. Size counter reached Pre-IRQ threshold. 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
16	PREIRQCH0	R	0h	Pre-IRQ for Channel 0. Size counter reached Pre-IRQ threshold. 0h = Clear interrupt mask bit 1h = Set interrupt mask bit

Table 10-21. MIS Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
15	DMACH15	R	0h	DMA Channel 15 interrupt signals that size counter reached zero (DMASZ=0). 0h = Interrupt did not occur or is masked out 1h = Interrupt occurred
14	DMACH14	R	0h	DMA Channel 14 interrupt signals that size counter reached zero (DMASZ=0). 0h = Interrupt did not occur or is masked out 1h = Interrupt occurred
13	DMACH13	R	0h	DMA Channel 13 interrupt signals that size counter reached zero (DMASZ=0). 0h = Interrupt did not occur or is masked out 1h = Interrupt occurred
12	DMACH12	R	0h	DMA Channel 12 interrupt signals that size counter reached zero (DMASZ=0). 0h = Interrupt did not occur or is masked out 1h = Interrupt occurred
11	DMACH11	R	0h	DMA Channel 11 interrupt signals that size counter reached zero (DMASZ=0). 0h = Interrupt did not occur or is masked out 1h = Interrupt occurred
10	DMACH10	R	0h	DMA Channel 10 interrupt signals that size counter reached zero (DMASZ=0). 0h = Interrupt did not occur or is masked out 1h = Interrupt occurred
9	DMACH9	R	0h	DMA Channel 9 interrupt signals that size counter reached zero (DMASZ=0). 0h = Interrupt did not occur or is masked out 1h = Interrupt occurred
8	DMACH8	R	0h	DMA Channel 8 interrupt signals that size counter reached zero (DMASZ=0). 0h = Interrupt did not occur or is masked out 1h = Interrupt occurred
7	DMACH7	R	0h	DMA Channel 7 interrupt signals that size counter reached zero (DMASZ=0). 0h = Interrupt did not occur or is masked out 1h = Interrupt occurred
6	DMACH6	R	0h	DMA Channel 6 interrupt signals that size counter reached zero (DMASZ=0). 0h = Interrupt did not occur or is masked out 1h = Interrupt occurred
5	DMACH5	R	0h	DMA Channel 5 interrupt signals that size counter reached zero (DMASZ=0). 0h = Interrupt did not occur or is masked out 1h = Interrupt occurred
4	DMACH4	R	0h	DMA Channel 4 interrupt signals that size counter reached zero (DMASZ=0). 0h = Interrupt did not occur or is masked out 1h = Interrupt occurred
3	DMACH3	R	0h	DMA Channel 3 interrupt signals that size counter reached zero (DMASZ=0). 0h = Interrupt did not occur or is masked out 1h = Interrupt occurred
2	DMACH2	R	0h	DMA Channel 2 interrupt signals that size counter reached zero (DMASZ=0). 0h = Interrupt did not occur or is masked out 1h = Interrupt occurred

Table 10-21. MIS Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
1	DMACH1	R	0h	DMA Channel 1 interrupt signals that size counter reached zero (DMASZ=0). 0h = Interrupt did not occur or is masked out 1h = Interrupt occurred
0	DMACH0	R	0h	DMA Channel 0 interrupt signals that size counter reached zero (DMASZ=0). 0h = Interrupt did not occur or is masked out 1h = Interrupt occurred

10 ISET Register (Offset = 1040h) [Reset = 00000000h]

ISET is shown in Figure 10-10 and described in Table 10-13.

Return to the Summary Table.

Interrupt set. Allows interrupts to be set by software (useful in diagnostics and safety checks). Writing a 1 to a bit in ISET will set the event and therefore the related RIS [RIS] bit also gets set. If the interrupt is enabled through the mask, then the corresponding MIS [MIS] bit is also set.
Note: The number of DMACH is device dependent. Please consult the datasheet of the specific device to map which channel number is implemented.

Figure 10-10. ISET Register

31	30	29	28	27	26	25	24
RESERVED						DATAERR	ADDRERR
W-0h						W-0h	W-0h
23	22	21	20	19	18	17	16
RESERVED	RESERVED	PREIRQCH5	PREIRQCH4	PREIRQCH3	PREIRQCH2	PREIRQCH1	PREIRQCH0
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h
15	14	13	12	11	10	9	8
DMACH15	DMACH14	DMACH13	DMACH12	DMACH11	DMACH10	DMACH9	DMACH8
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h
7	6	5	4	3	2	1	0
DMACH7	DMACH6	DMACH5	DMACH4	DMACH3	DMACH2	DMACH1	DMACH0
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h

Table 10-23. ISET Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	RESERVED	W	0h	
25	DATAERR	W	0h	DMA data error, SRC data might be corrupted (PAR or ECC error). 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
24	ADDRERR	W	0h	DMA address error, SRC address not reachable. 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
23	RESERVED	W	0h	Reserved
22	RESERVED	W	0h	Reserved
21	PREIRQCH5	W	0h	Pre-IRQ for Channel 5. Size counter reached Pre-IRQ threshold. 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
20	PREIRQCH4	W	0h	Pre-IRQ for Channel 4. Size counter reached Pre-IRQ threshold. 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
19	PREIRQCH3	W	0h	Pre-IRQ for Channel 3. Size counter reached Pre-IRQ threshold. 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
18	PREIRQCH2	W	0h	Pre-IRQ for Channel 2. Size counter reached Pre-IRQ threshold. 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
17	PREIRQCH1	W	0h	Pre-IRQ for Channel 1. Size counter reached Pre-IRQ threshold. 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
16	PREIRQCH0	W	0h	Pre-IRQ for Channel 0. Size counter reached Pre-IRQ threshold. 0h = Clear interrupt mask bit 1h = Set interrupt mask bit

Table 10-23. ISET Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
15	DMACH15	W	0h	DMA Channel 0 interrupt signals that size counter reached zero (DMASZ=0). 0h = Writing 0 has no effect 1h = Set interrupt
14	DMACH14	W	0h	DMA Channel 14 interrupt signals that size counter reached zero (DMASZ=0). 0h = Writing 0 has no effect 1h = Set interrupt
13	DMACH13	W	0h	DMA Channel 13 interrupt signals that size counter reached zero (DMASZ=0). 0h = Writing 0 has no effect 1h = Set interrupt
12	DMACH12	W	0h	DMA Channel 12 interrupt signals that size counter reached zero (DMASZ=0). 0h = Writing 0 has no effect 1h = Set interrupt
11	DMACH11	W	0h	DMA Channel 11 interrupt signals that size counter reached zero (DMASZ=0). 0h = Writing 0 has no effect 1h = Set interrupt
10	DMACH10	W	0h	DMA Channel 0 interrupt signals that size counter reached zero (DMASZ=0). 0h = Writing 0 has no effect 1h = Set interrupt
9	DMACH9	W	0h	DMA Channel 9 interrupt signals that size counter reached zero (DMASZ=0). 0h = Writing 0 has no effect 1h = Set interrupt
8	DMACH8	W	0h	DMA Channel 8 interrupt signals that size counter reached zero (DMASZ=0). 0h = Writing 0 has no effect 1h = Set interrupt
7	DMACH7	W	0h	DMA Channel 7 interrupt signals that size counter reached zero (DMASZ=0). 0h = Writing 0 has no effect 1h = Set interrupt
6	DMACH6	W	0h	DMA Channel 6 interrupt signals that size counter reached zero (DMASZ=0). 0h = Writing 0 has no effect 1h = Set interrupt
5	DMACH5	W	0h	DMA Channel 5 interrupt signals that size counter reached zero (DMASZ=0). 0h = Writing 0 has no effect 1h = Set interrupt
4	DMACH4	W	0h	DMA Channel 4 interrupt signals that size counter reached zero (DMASZ=0). 0h = Writing 0 has no effect 1h = Set interrupt
3	DMACH3	W	0h	DMA Channel 3 interrupt signals that size counter reached zero (DMASZ=0). 0h = Writing 0 has no effect 1h = Set interrupt
2	DMACH2	W	0h	DMA Channel 2 interrupt signals that size counter reached zero (DMASZ=0). 0h = Writing 0 has no effect 1h = Set interrupt

Table 10-23. ISET Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
1	DMACH1	W	0h	DMA Channel 1 interrupt signals that size counter reached zero (DMASZ=0). 0h = Writing 0 has no effect 1h = Set interrupt
0	DMACH0	W	0h	DMA Channel 0 interrupt signals that size counter reached zero (DMASZ=0). 0h = Writing 0 has no effect 1h = Set interrupt

11 ICLR Register (Offset = 1048h) [Reset = 00000000h]

ICLR is shown in Figure 10-11 and described in Table 10-14.

Return to the Summary Table.

Interrupt clear. Write a 1 to clear corresponding Interrupt.
Note: The number of DMACH is device dependent. Please consult the datasheet of the specific device to map which channel number is implemented.

Figure 10-11. ICLR Register

31	30	29	28	27	26	25	24
RESERVED						DATAERR	ADDRERR
W-0h						W-0h	W-0h
23	22	21	20	19	18	17	16
RESERVED	RESERVED	PREIRQCH5	PREIRQCH4	PREIRQCH3	PREIRQCH2	PREIRQCH1	PREIRQCH0
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h
15	14	13	12	11	10	9	8
DMACH15	DMACH14	DMACH13	DMACH12	DMACH11	DMACH10	DMACH9	DMACH8
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h
7	6	5	4	3	2	1	0
DMACH7	DMACH6	DMACH5	DMACH4	DMACH3	DMACH2	DMACH1	DMACH0
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h

Table 10-25. ICLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	RESERVED	W	0h	
25	DATAERR	W	0h	DMA data error, SRC data might be corrupted (PAR or ECC error). 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
24	ADDRERR	W	0h	DMA address error, SRC address not reachable. 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
23	RESERVED	W	0h	Reserved
22	RESERVED	W	0h	Reserved
21	PREIRQCH5	W	0h	Pre-IRQ for Channel 5. Size counter reached Pre-IRQ threshold. 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
20	PREIRQCH4	W	0h	Pre-IRQ for Channel 4. Size counter reached Pre-IRQ threshold. 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
19	PREIRQCH3	W	0h	Pre-IRQ for Channel 3. Size counter reached Pre-IRQ threshold. 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
18	PREIRQCH2	W	0h	Pre-IRQ for Channel 2. Size counter reached Pre-IRQ threshold. 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
17	PREIRQCH1	W	0h	Pre-IRQ for Channel 1. Size counter reached Pre-IRQ threshold. 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
16	PREIRQCH0	W	0h	Pre-IRQ for Channel 0. Size counter reached Pre-IRQ threshold. 0h = Clear interrupt mask bit 1h = Set interrupt mask bit

Table 10-25. ICLR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
15	DMACH15	W	0h	DMA Channel 15 interrupt signals that size counter reached zero (DMASZ=0). 0h = Writing 0 has no effect 1h = Clear interrupt
14	DMACH14	W	0h	DMA Channel 14 interrupt signals that size counter reached zero (DMASZ=0). 0h = Writing 0 has no effect 1h = Clear interrupt
13	DMACH13	W	0h	DMA Channel 13 interrupt signals that size counter reached zero (DMASZ=0). 0h = Writing 0 has no effect 1h = Clear interrupt
12	DMACH12	W	0h	DMA Channel 12 interrupt signals that size counter reached zero (DMASZ=0). 0h = Writing 0 has no effect 1h = Clear interrupt
11	DMACH11	W	0h	DMA Channel 11 interrupt signals that size counter reached zero (DMASZ=0). 0h = Writing 0 has no effect 1h = Clear interrupt
10	DMACH10	W	0h	DMA Channel 10 interrupt signals that size counter reached zero (DMASZ=0). 0h = Writing 0 has no effect 1h = Clear interrupt
9	DMACH9	W	0h	DMA Channel 9 interrupt signals that size counter reached zero (DMASZ=0). 0h = Writing 0 has no effect 1h = Clear interrupt
8	DMACH8	W	0h	DMA Channel 8 interrupt signals that size counter reached zero (DMASZ=0). 0h = Writing 0 has no effect 1h = Clear interrupt
7	DMACH7	W	0h	DMA Channel 7 interrupt signals that size counter reached zero (DMASZ=0). 0h = Writing 0 has no effect 1h = Clear interrupt
6	DMACH6	W	0h	DMA Channel 6 interrupt signals that size counter reached zero (DMASZ=0). 0h = Writing 0 has no effect 1h = Clear interrupt
5	DMACH5	W	0h	DMA Channel 5 interrupt signals that size counter reached zero (DMASZ=0). 0h = Writing 0 has no effect 1h = Clear interrupt
4	DMACH4	W	0h	DMA Channel 4 interrupt signals that size counter reached zero (DMASZ=0). 0h = Writing 0 has no effect 1h = Clear interrupt
3	DMACH3	W	0h	DMA Channel 3 interrupt signals that size counter reached zero (DMASZ=0). 0h = Writing 0 has no effect 1h = Clear interrupt
2	DMACH2	W	0h	DMA Channel 2 interrupt signals that size counter reached zero (DMASZ=0). 0h = Writing 0 has no effect 1h = Clear interrupt

Table 10-25. ICLR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
1	DMACH1	W	0h	DMA Channel 1 interrupt signals that size counter reached zero (DMASZ=0). 0h = Writing 0 has no effect 1h = Clear interrupt
0	DMACH0	W	0h	DMA Channel 0 interrupt signals that size counter reached zero (DMASZ=0). 0h = Writing 0 has no effect 1h = Clear interrupt

12 IIDX Register (Offset = 1050h) [Reset = 00000000h]

IIDX is shown in Figure 10-12 and described in Table 10-15.

Return to the Summary Table.

This register provides the highest priority enabled interrupt index. Value 0x00 means no event pending. Interrupt 1 is the highest priority, IIDX next highest, 4, 8, . . . IIDX^31 is the least priority. That is, the least bit position that is set to 1 denotes the highest priority pending interrupt. The priority order is fixed. However, users can implement their own prioritization schemes using other registers that expose the full set of interrupts that have occurred.
On each read, only one interrupt is indicated. On a read, the current interrupt (highest priority) is automatically cleared by the hardware and the corresponding interrupt flag in RIS [RIS] and MIS [MIS] are cleared as well. After a read from the CPU (not from the debug interface), the register is updated with the next highest priority interrupt, if none are pending, then it should display 0x0.
Note: The number of DMACH is device dependent. Please consult the datasheet of the specific device to map which channel number is implemented.

Figure 10-12. IIDX Register

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

Table 10-27. IIDX Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	
7-0	STAT	R	0h	Interrupt index status 00h = No bit is set means there is no pending interrupt request 01h = DMA Channel 0 size counter reached zero (DMASZ=0). 02h = DMA Channel 1 size counter reached zero (DMASZ=0). 03h = DMA Channel 2 size counter reached zero (DMASZ=0). 04h = DMA Channel 3 size counter reached zero (DMASZ=0). 05h = DMA Channel 4 size counter reached zero (DMASZ=0). 06h = DMA Channel 5 size counter reached zero (DMASZ=0). 07h = DMA Channel 6 size counter reached zero (DMASZ=0). 08h = DMA Channel 7 size counter reached zero (DMASZ=0). 09h = DMA Channel 8 size counter reached zero (DMASZ=0). 0Ah = DMA Channel 9 size counter reached zero (DMASZ=0). 0Bh = DMA Channel 10 size counter reached zero (DMASZ=0). 0Ch = DMA Channel 11 size counter reached zero (DMASZ=0). 0Dh = DMA Channel 12 size counter reached zero (DMASZ=0). 0Eh = DMA Channel 13 size counter reached zero (DMASZ=0). 0Fh = DMA Channel 14 size counter reached zero (DMASZ=0). 10h = DMA Channel 15 size counter reached zero (DMASZ=0). 11h = PRE-IRQ event for DMA Channel 0. 12h = PRE-IRQ event for DMA Channel 1. 13h = PRE-IRQ event for DMA Channel 2. 14h = PRE-IRQ event for DMA Channel 3. 15h = PRE-IRQ event for DMA Channel 4. 16h = PRE-IRQ event for DMA Channel 5. 17h = PRE-IRQ event for DMA Channel 6. 18h = PRE-IRQ event for DMA Channel 7. 19h = DMA address error, SRC address not reachable. 1Ah = DMA data error, SRC data might be corrupted (PAR or ECC error).

13 IMASK Register (Offset = 1058h) [Reset = 00000000h]

IMASK is shown in Figure 10-13 and described in Table 10-16.

Return to the Summary Table.

Interrupt Mask. If a bit is set, then the corresponding interrupt is un-masked. Un-masking the interrupt causes the raw interrupt to be visible in IIDX [IIDX], as well as MIS [MIS].
Note: The number of DMACH is device dependent. Please consult the datasheet of the specific device to map which channel number is implemented.

Figure 10-13. IMASK Register

31	30	29	28	27	26	25	24
RESERVED						DATAERR	ADDRERR
R/W-0h						R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
RESERVED	RESERVED	PREIRQCH5	PREIRQCH4	PREIRQCH3	PREIRQCH2	PREIRQCH1	PREIRQCH0
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
DMACH15	DMACH14	DMACH13	DMACH12	DMACH11	DMACH10	DMACH9	DMACH8
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
DMACH7	DMACH6	DMACH5	DMACH4	DMACH3	DMACH2	DMACH1	DMACH0
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 10-29. IMASK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	RESERVED	R/W	0h	
25	DATAERR	R/W	0h	DMA data error, SRC data might be corrupted (PAR or ECC error). 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
24	ADDRERR	R/W	0h	DMA address error, SRC address not reachable. 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
23	RESERVED	R/W	0h	Reserved
22	RESERVED	R/W	0h	Reserved
21	PREIRQCH5	R/W	0h	Pre-IRQ for Channel 5. Size counter reached Pre-IRQ threshold. 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
20	PREIRQCH4	R/W	0h	Pre-IRQ for Channel 4. Size counter reached Pre-IRQ threshold. 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
19	PREIRQCH3	R/W	0h	Pre-IRQ for Channel 3. Size counter reached Pre-IRQ threshold. 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
18	PREIRQCH2	R/W	0h	Pre-IRQ for Channel 2. Size counter reached Pre-IRQ threshold. 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
17	PREIRQCH1	R/W	0h	Pre-IRQ for Channel 1. Size counter reached Pre-IRQ threshold. 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
16	PREIRQCH0	R/W	0h	Pre-IRQ for Channel 0. Size counter reached Pre-IRQ threshold. 0h = Clear interrupt mask bit 1h = Set interrupt mask bit

Table 10-29. IMASK Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
15	DMACH15	R/W	0h	DMA Channel 15 interrupt signal. Size counter reached zero (DMASZ=0). 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
14	DMACH14	R/W	0h	DMA Channel 14 interrupt signal. Size counter reached zero (DMASZ=0). 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
13	DMACH13	R/W	0h	DMA Channel 13 interrupt signal. Size counter reached zero (DMASZ=0). 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
12	DMACH12	R/W	0h	DMA Channel 12 interrupt signal. Size counter reached zero (DMASZ=0). 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
11	DMACH11	R/W	0h	DMA Channel 11 interrupt signal. Size counter reached zero (DMASZ=0). 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
10	DMACH10	R/W	0h	DMA Channel 10 interrupt signal. Size counter reached zero (DMASZ=0). 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
9	DMACH9	R/W	0h	DMA Channel 9 interrupt signal. Size counter reached zero (DMASZ=0). 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
8	DMACH8	R/W	0h	DMA Channel 8 interrupt signal. Size counter reached zero (DMASZ=0). 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
7	DMACH7	R/W	0h	DMA Channel 7 interrupt signal. Size counter reached zero (DMASZ=0). 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
6	DMACH6	R/W	0h	DMA Channel 6 interrupt signal. Size counter reached zero (DMASZ=0). 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
5	DMACH5	R/W	0h	DMA Channel 5 interrupt signal. Size counter reached zero (DMASZ=0). 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
4	DMACH4	R/W	0h	DMA Channel 4 interrupt signal. Size counter reached zero (DMASZ=0). 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
3	DMACH3	R/W	0h	DMA Channel 3 interrupt signal. Size counter reached zero (DMASZ=0). 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
2	DMACH2	R/W	0h	DMA Channel 2 interrupt signal. Size counter reached zero (DMASZ=0). 0h = Clear interrupt mask bit 1h = Set interrupt mask bit

Table 10-29. IMASK Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
1	DMACH1	R/W	0h	DMA Channel 1 interrupt signal. Size counter reached zero (DMASZ=0). 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
0	DMACH0	R/W	0h	DMA Channel 0 interrupt signal. Size counter reached zero (DMASZ=0). 0h = Clear interrupt mask bit 1h = Set interrupt mask bit

14 RIS Register (Offset = 1060h) [Reset = 00000000h]

RIS is shown in Figure 10-14 and described in Table 10-17.

Return to the Summary Table.

Raw interrupt status. Reflects all pending interrupts, regardless of masking. The RIS register allows the user to implement a poll scheme. A flag set in this register can be cleared by writing 1 to the ICLR [ICLR] register bit even if the corresponding IMASK [IMASK] bit is not enabled.
Note: The number of DMACH is device dependent. Please consult the datasheet of the specific device to map which channel number is implemented.

Figure 10-14. RIS Register

31	30	29	28	27	26	25	24
RESERVED						DATAERR	ADDRERR
R-0h						R-0h	R-0h
23	22	21	20	19	18	17	16
RESERVED	RESERVED	PREIRQCH5	PREIRQCH4	PREIRQCH3	PREIRQCH2	PREIRQCH1	PREIRQCH0
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
15	14	13	12	11	10	9	8
DMACH15	DMACH14	DMACH13	DMACH12	DMACH11	DMACH10	DMACH9	DMACH8
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
7	6	5	4	3	2	1	0
DMACH7	DMACH6	DMACH5	DMACH4	DMACH3	DMACH2	DMACH1	DMACH0
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

Table 10-31. RIS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	RESERVED	R	0h	
25	DATAERR	R	0h	DMA data error, SRC data might be corrupted (PAR or ECC error). 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
24	ADDRERR	R	0h	DMA address error, SRC address not reachable. 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
23	RESERVED	R	0h	Reserved
22	RESERVED	R	0h	Reserved
21	PREIRQCH5	R	0h	Pre-IRQ for Channel 5. Size counter reached Pre-IRQ threshold. 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
20	PREIRQCH4	R	0h	Pre-IRQ for Channel 4. Size counter reached Pre-IRQ threshold. 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
19	PREIRQCH3	R	0h	Pre-IRQ for Channel 3. Size counter reached Pre-IRQ threshold. 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
18	PREIRQCH2	R	0h	Pre-IRQ for Channel 2. Size counter reached Pre-IRQ threshold. 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
17	PREIRQCH1	R	0h	Pre-IRQ for Channel 1. Size counter reached Pre-IRQ threshold. 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
16	PREIRQCH0	R	0h	Pre-IRQ for Channel 0. Size counter reached Pre-IRQ threshold. 0h = Clear interrupt mask bit 1h = Set interrupt mask bit

Table 10-31. RIS Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
15	DMACH15	R	0h	DMA Channel 15 interrupt signals that size counter reached zero (DMASZ=0). 0h = Interrupt did not occur 1h = Interrupt occurred
14	DMACH14	R	0h	DMA Channel 14 interrupt signals that size counter reached zero (DMASZ=0). 0h = Interrupt did not occur 1h = Interrupt occurred
13	DMACH13	R	0h	DMA Channel 13 interrupt signals that size counter reached zero (DMASZ=0). 0h = Interrupt did not occur 1h = Interrupt occurred
12	DMACH12	R	0h	DMA Channel 12 interrupt signals that size counter reached zero (DMASZ=0). 0h = Interrupt did not occur 1h = Interrupt occurred
11	DMACH11	R	0h	DMA Channel 11 interrupt signals that size counter reached zero (DMASZ=0). 0h = Interrupt did not occur 1h = Interrupt occurred
10	DMACH10	R	0h	DMA Channel 10 interrupt signals that size counter reached zero (DMASZ=0). 0h = Interrupt did not occur 1h = Interrupt occurred
9	DMACH9	R	0h	DMA Channel 9 interrupt signals that size counter reached zero (DMASZ=0). 0h = Interrupt did not occur 1h = Interrupt occurred
8	DMACH8	R	0h	DMA Channel 8 interrupt signals that size counter reached zero (DMASZ=0). 0h = Interrupt did not occur 1h = Interrupt occurred
7	DMACH7	R	0h	DMA Channel 7 interrupt signals that size counter reached zero (DMASZ=0). 0h = Interrupt did not occur 1h = Interrupt occurred
6	DMACH6	R	0h	DMA Channel 6 interrupt signals that size counter reached zero (DMASZ=0). 0h = Interrupt did not occur 1h = Interrupt occurred
5	DMACH5	R	0h	DMA Channel 5 interrupt signals that size counter reached zero (DMASZ=0). 0h = Interrupt did not occur 1h = Interrupt occurred
4	DMACH4	R	0h	DMA Channel 4 interrupt signals that size counter reached zero (DMASZ=0). 0h = Interrupt did not occur 1h = Interrupt occurred
3	DMACH3	R	0h	DMA Channel 3 interrupt signals that size counter reached zero (DMASZ=0). 0h = Interrupt did not occur 1h = Interrupt occurred
2	DMACH2	R	0h	DMA Channel 2 interrupt signals that size counter reached zero (DMASZ=0). 0h = Interrupt did not occur 1h = Interrupt occurred

Table 10-31. RIS Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
1	DMACH1	R	0h	DMA Channel 1 interrupt signals that size counter reached zero (DMASZ=0). 0h = Interrupt did not occur 1h = Interrupt occurred
0	DMACH0	R	0h	DMA Channel 0 interrupt signals that size counter reached zero (DMASZ=0). 0h = Interrupt did not occur 1h = Interrupt occurred

15 MIS Register (Offset = 1068h) [Reset = 00000000h]

MIS is shown in Figure 10-15 and described in Table 10-18.

Return to the Summary Table.

Masked interrupt status. This is an AND of the IMASK [IMASK] and RIS [RIS] registers.
Note: The number of DMACH is device dependent. Please consult the datasheet of the specific device to map which channel number is implemented.

Figure 10-15. MIS Register

31	30	29	28	27	26	25	24
RESERVED						DATAERR	ADDRERR
R-0h						R-0h	R-0h
23	22	21	20	19	18	17	16
RESERVED	RESERVED	PREIRQCH5	PREIRQCH4	PREIRQCH3	PREIRQCH2	PREIRQCH1	PREIRQCH0
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
15	14	13	12	11	10	9	8
DMACH15	DMACH14	DMACH13	DMACH12	DMACH11	DMACH10	DMACH9	DMACH8
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
7	6	5	4	3	2	1	0
DMACH7	DMACH6	DMACH5	DMACH4	DMACH3	DMACH2	DMACH1	DMACH0
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

Table 10-33. MIS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	RESERVED	R	0h	
25	DATAERR	R	0h	DMA data error, SRC data might be corrupted (PAR or ECC error). 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
24	ADDRERR	R	0h	DMA address error, SRC address not reachable. 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
23	RESERVED	R	0h	Reserved
22	RESERVED	R	0h	Reserved
21	PREIRQCH5	R	0h	Pre-IRQ for Channel 5. Size counter reached Pre-IRQ threshold. 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
20	PREIRQCH4	R	0h	Pre-IRQ for Channel 4. Size counter reached Pre-IRQ threshold. 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
19	PREIRQCH3	R	0h	Pre-IRQ for Channel 3. Size counter reached Pre-IRQ threshold. 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
18	PREIRQCH2	R	0h	Pre-IRQ for Channel 2. Size counter reached Pre-IRQ threshold. 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
17	PREIRQCH1	R	0h	Pre-IRQ for Channel 1. Size counter reached Pre-IRQ threshold. 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
16	PREIRQCH0	R	0h	Pre-IRQ for Channel 0. Size counter reached Pre-IRQ threshold. 0h = Clear interrupt mask bit 1h = Set interrupt mask bit

Table 10-33. MIS Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
15	DMACH15	R	0h	DMA Channel 15 interrupt signals that size counter reached zero (DMASZ=0). 0h = Interrupt did not occur or is masked out 1h = Interrupt occurred
14	DMACH14	R	0h	DMA Channel 14 interrupt signals that size counter reached zero (DMASZ=0). 0h = Interrupt did not occur or is masked out 1h = Interrupt occurred
13	DMACH13	R	0h	DMA Channel 13 interrupt signals that size counter reached zero (DMASZ=0). 0h = Interrupt did not occur or is masked out 1h = Interrupt occurred
12	DMACH12	R	0h	DMA Channel 12 interrupt signals that size counter reached zero (DMASZ=0). 0h = Interrupt did not occur or is masked out 1h = Interrupt occurred
11	DMACH11	R	0h	DMA Channel 11 interrupt signals that size counter reached zero (DMASZ=0). 0h = Interrupt did not occur or is masked out 1h = Interrupt occurred
10	DMACH10	R	0h	DMA Channel 10 interrupt signals that size counter reached zero (DMASZ=0). 0h = Interrupt did not occur or is masked out 1h = Interrupt occurred
9	DMACH9	R	0h	DMA Channel 9 interrupt signals that size counter reached zero (DMASZ=0). 0h = Interrupt did not occur or is masked out 1h = Interrupt occurred
8	DMACH8	R	0h	DMA Channel 8 interrupt signals that size counter reached zero (DMASZ=0). 0h = Interrupt did not occur or is masked out 1h = Interrupt occurred
7	DMACH7	R	0h	DMA Channel 7 interrupt signals that size counter reached zero (DMASZ=0). 0h = Interrupt did not occur or is masked out 1h = Interrupt occurred
6	DMACH6	R	0h	DMA Channel 6 interrupt signals that size counter reached zero (DMASZ=0). 0h = Interrupt did not occur or is masked out 1h = Interrupt occurred
5	DMACH5	R	0h	DMA Channel 5 interrupt signals that size counter reached zero (DMASZ=0). 0h = Interrupt did not occur or is masked out 1h = Interrupt occurred
4	DMACH4	R	0h	DMA Channel 4 interrupt signals that size counter reached zero (DMASZ=0). 0h = Interrupt did not occur or is masked out 1h = Interrupt occurred
3	DMACH3	R	0h	DMA Channel 3 interrupt signals that size counter reached zero (DMASZ=0). 0h = Interrupt did not occur or is masked out 1h = Interrupt occurred
2	DMACH2	R	0h	DMA Channel 2 interrupt signals that size counter reached zero (DMASZ=0). 0h = Interrupt did not occur or is masked out 1h = Interrupt occurred

Table 10-33. MIS Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
1	DMACH1	R	0h	DMA Channel 1 interrupt signals that size counter reached zero (DMASZ=0). 0h = Interrupt did not occur or is masked out 1h = Interrupt occurred
0	DMACH0	R	0h	DMA Channel 0 interrupt signals that size counter reached zero (DMASZ=0). 0h = Interrupt did not occur or is masked out 1h = Interrupt occurred

16 ISET Register (Offset = 1070h) [Reset = 00000000h]

ISET is shown in Figure 10-16 and described in Table 10-19.

Return to the Summary Table.

Interrupt set. Allows interrupts to be set by software (useful in diagnostics and safety checks). Writing a 1 to a bit in ISET will set the event and therefore the related RIS [RIS] bit also gets set. If the interrupt is enabled through the mask, then the corresponding MIS [MIS] bit is also set.
Note: The number of DMACH is device dependent. Please consult the datasheet of the specific device to map which channel number is implemented.

Figure 10-16. ISET Register

31	30	29	28	27	26	25	24
RESERVED						DATAERR	ADDRERR
W-0h						W-0h	W-0h
23	22	21	20	19	18	17	16
RESERVED	RESERVED	PREIRQCH5	PREIRQCH4	PREIRQCH3	PREIRQCH2	PREIRQCH1	PREIRQCH0
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h
15	14	13	12	11	10	9	8
DMACH15	DMACH14	DMACH13	DMACH12	DMACH11	DMACH10	DMACH9	DMACH8
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h
7	6	5	4	3	2	1	0
DMACH7	DMACH6	DMACH5	DMACH4	DMACH3	DMACH2	DMACH1	DMACH0
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h

Table 10-35. ISET Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	RESERVED	W	0h	
25	DATAERR	W	0h	DMA data error, SRC data might be corrupted (PAR or ECC error). 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
24	ADDRERR	W	0h	DMA address error, SRC address not reachable. 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
23	RESERVED	W	0h	Reserved
22	RESERVED	W	0h	Reserved
21	PREIRQCH5	W	0h	Pre-IRQ for Channel 5. Size counter reached Pre-IRQ threshold. 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
20	PREIRQCH4	W	0h	Pre-IRQ for Channel 4. Size counter reached Pre-IRQ threshold. 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
19	PREIRQCH3	W	0h	Pre-IRQ for Channel 3. Size counter reached Pre-IRQ threshold. 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
18	PREIRQCH2	W	0h	Pre-IRQ for Channel 2. Size counter reached Pre-IRQ threshold. 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
17	PREIRQCH1	W	0h	Pre-IRQ for Channel 1. Size counter reached Pre-IRQ threshold. 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
16	PREIRQCH0	W	0h	Pre-IRQ for Channel 0. Size counter reached Pre-IRQ threshold. 0h = Clear interrupt mask bit 1h = Set interrupt mask bit

Table 10-35. ISET Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
15	DMACH15	W	0h	DMA Channel 0 interrupt signals that size counter reached zero (DMASZ=0). 0h = Writing 0 has no effect 1h = Set interrupt
14	DMACH14	W	0h	DMA Channel 14 interrupt signals that size counter reached zero (DMASZ=0). 0h = Writing 0 has no effect 1h = Set interrupt
13	DMACH13	W	0h	DMA Channel 13 interrupt signals that size counter reached zero (DMASZ=0). 0h = Writing 0 has no effect 1h = Set interrupt
12	DMACH12	W	0h	DMA Channel 12 interrupt signals that size counter reached zero (DMASZ=0). 0h = Writing 0 has no effect 1h = Set interrupt
11	DMACH11	W	0h	DMA Channel 11 interrupt signals that size counter reached zero (DMASZ=0). 0h = Writing 0 has no effect 1h = Set interrupt
10	DMACH10	W	0h	DMA Channel 0 interrupt signals that size counter reached zero (DMASZ=0). 0h = Writing 0 has no effect 1h = Set interrupt
9	DMACH9	W	0h	DMA Channel 9 interrupt signals that size counter reached zero (DMASZ=0). 0h = Writing 0 has no effect 1h = Set interrupt
8	DMACH8	W	0h	DMA Channel 8 interrupt signals that size counter reached zero (DMASZ=0). 0h = Writing 0 has no effect 1h = Set interrupt
7	DMACH7	W	0h	DMA Channel 7 interrupt signals that size counter reached zero (DMASZ=0). 0h = Writing 0 has no effect 1h = Set interrupt
6	DMACH6	W	0h	DMA Channel 6 interrupt signals that size counter reached zero (DMASZ=0). 0h = Writing 0 has no effect 1h = Set interrupt
5	DMACH5	W	0h	DMA Channel 5 interrupt signals that size counter reached zero (DMASZ=0). 0h = Writing 0 has no effect 1h = Set interrupt
4	DMACH4	W	0h	DMA Channel 4 interrupt signals that size counter reached zero (DMASZ=0). 0h = Writing 0 has no effect 1h = Set interrupt
3	DMACH3	W	0h	DMA Channel 3 interrupt signals that size counter reached zero (DMASZ=0). 0h = Writing 0 has no effect 1h = Set interrupt
2	DMACH2	W	0h	DMA Channel 2 interrupt signals that size counter reached zero (DMASZ=0). 0h = Writing 0 has no effect 1h = Set interrupt

Table 10-35. ISET Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
1	DMACH1	W	0h	DMA Channel 1 interrupt signals that size counter reached zero (DMASZ=0). 0h = Writing 0 has no effect 1h = Set interrupt
0	DMACH0	W	0h	DMA Channel 0 interrupt signals that size counter reached zero (DMASZ=0). 0h = Writing 0 has no effect 1h = Set interrupt

17 ICLR Register (Offset = 1078h) [Reset = 00000000h]

ICLR is shown in Figure 10-17 and described in Table 10-20.

Return to the Summary Table.

Interrupt clear. Write a 1 to clear corresponding Interrupt.
Note: The number of DMACH is device dependent. Please consult the datasheet of the specific device to map which channel number is implemented.

Figure 10-17. ICLR Register

31	30	29	28	27	26	25	24
RESERVED						DATAERR	ADDRERR
W-0h						W-0h	W-0h
23	22	21	20	19	18	17	16
RESERVED	RESERVED	PREIRQCH5	PREIRQCH4	PREIRQCH3	PREIRQCH2	PREIRQCH1	PREIRQCH0
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h
15	14	13	12	11	10	9	8
DMACH15	DMACH14	DMACH13	DMACH12	DMACH11	DMACH10	DMACH9	DMACH8
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h
7	6	5	4	3	2	1	0
DMACH7	DMACH6	DMACH5	DMACH4	DMACH3	DMACH2	DMACH1	DMACH0
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h

Table 10-37. ICLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	RESERVED	W	0h	
25	DATAERR	W	0h	DMA data error, SRC data might be corrupted (PAR or ECC error). 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
24	ADDRERR	W	0h	DMA address error, SRC address not reachable. 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
23	RESERVED	W	0h	Reserved
22	RESERVED	W	0h	Reserved
21	PREIRQCH5	W	0h	Pre-IRQ for Channel 5. Size counter reached Pre-IRQ threshold. 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
20	PREIRQCH4	W	0h	Pre-IRQ for Channel 4. Size counter reached Pre-IRQ threshold. 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
19	PREIRQCH3	W	0h	Pre-IRQ for Channel 3. Size counter reached Pre-IRQ threshold. 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
18	PREIRQCH2	W	0h	Pre-IRQ for Channel 2. Size counter reached Pre-IRQ threshold. 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
17	PREIRQCH1	W	0h	Pre-IRQ for Channel 1. Size counter reached Pre-IRQ threshold. 0h = Clear interrupt mask bit 1h = Set interrupt mask bit
16	PREIRQCH0	W	0h	Pre-IRQ for Channel 0. Size counter reached Pre-IRQ threshold. 0h = Clear interrupt mask bit 1h = Set interrupt mask bit

Table 10-37. ICLR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
15	DMACH15	W	0h	DMA Channel 15 interrupt signals that size counter reached zero (DMASZ=0). 0h = Writing 0 has no effect 1h = Clear interrupt
14	DMACH14	W	0h	DMA Channel 14 interrupt signals that size counter reached zero (DMASZ=0). 0h = Writing 0 has no effect 1h = Clear interrupt
13	DMACH13	W	0h	DMA Channel 13 interrupt signals that size counter reached zero (DMASZ=0). 0h = Writing 0 has no effect 1h = Clear interrupt
12	DMACH12	W	0h	DMA Channel 12 interrupt signals that size counter reached zero (DMASZ=0). 0h = Writing 0 has no effect 1h = Clear interrupt
11	DMACH11	W	0h	DMA Channel 11 interrupt signals that size counter reached zero (DMASZ=0). 0h = Writing 0 has no effect 1h = Clear interrupt
10	DMACH10	W	0h	DMA Channel 10 interrupt signals that size counter reached zero (DMASZ=0). 0h = Writing 0 has no effect 1h = Clear interrupt
9	DMACH9	W	0h	DMA Channel 9 interrupt signals that size counter reached zero (DMASZ=0). 0h = Writing 0 has no effect 1h = Clear interrupt
8	DMACH8	W	0h	DMA Channel 8 interrupt signals that size counter reached zero (DMASZ=0). 0h = Writing 0 has no effect 1h = Clear interrupt
7	DMACH7	W	0h	DMA Channel 7 interrupt signals that size counter reached zero (DMASZ=0). 0h = Writing 0 has no effect 1h = Clear interrupt
6	DMACH6	W	0h	DMA Channel 6 interrupt signals that size counter reached zero (DMASZ=0). 0h = Writing 0 has no effect 1h = Clear interrupt
5	DMACH5	W	0h	DMA Channel 5 interrupt signals that size counter reached zero (DMASZ=0). 0h = Writing 0 has no effect 1h = Clear interrupt
4	DMACH4	W	0h	DMA Channel 4 interrupt signals that size counter reached zero (DMASZ=0). 0h = Writing 0 has no effect 1h = Clear interrupt
3	DMACH3	W	0h	DMA Channel 3 interrupt signals that size counter reached zero (DMASZ=0). 0h = Writing 0 has no effect 1h = Clear interrupt
2	DMACH2	W	0h	DMA Channel 2 interrupt signals that size counter reached zero (DMASZ=0). 0h = Writing 0 has no effect 1h = Clear interrupt

Table 10-37. ICLR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
1	DMACH1	W	0h	DMA Channel 1 interrupt signals that size counter reached zero (DMASZ=0). 0h = Writing 0 has no effect 1h = Clear interrupt
0	DMACH0	W	0h	DMA Channel 0 interrupt signals that size counter reached zero (DMASZ=0). 0h = Writing 0 has no effect 1h = Clear interrupt

18 EVT_MODE Register (Offset = 10E0h) [Reset = 00000000h]

EVT_MODE is shown in Figure 10-18 and described in Table 10-21.

Return to the Summary Table.

Event mode register. It is used to select whether each line is disabled, in software mode (software clears the RIS) or in hardware mode (hardware clears the RIS)

Figure 10-18. EVT_MODE Register

31	30	29	28	27	26	25	24
RESERVED							
R/W-0h							
23	22	21	20	19	18	17	16
RESERVED							
R/W-0h							
15	14	13	12	11	10	9	8
RESERVED							
R/W-0h							
7	6	5	4	3	2	1	0
RESERVED				EVT1_CFG		INT0_CFG	
R/W-0h				R-0h		R-0h	

Table 10-39. EVT_MODE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R/W	0h	
3-2	EVT1_CFG	R	2h	Event line mode select for event corresponding to generic event GEN_EVENT 0h = The interrupt or event line is disabled. 1h = The interrupt or event line is in software mode. Software must clear the RIS. 2h = The interrupt or event line is in hardware mode. The hardware (another module) clears automatically the associated RIS flag.
1-0	INT0_CFG	R	1h	Event line mode select for event corresponding to interrupt event CPU_INT 0h = The interrupt or event line is disabled. 1h = The interrupt or event line is in software mode. Software must clear the RIS. 2h = The interrupt or event line is in hardware mode. The hardware (another module) clears automatically the associated RIS flag.

19 DESC Register (Offset = 10FCh) [Reset = 00000000h]

DESC is shown in Figure 10-19 and described in Table 10-22.

Return to the Summary Table.

This register identifies the peripheral and its exact version.

Figure 10-19. DESC Register

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

Table 10-41. DESC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	MODULEID	R	2511h	Module identification contains a unique peripheral identification number. The assignments are maintained in a central database for all of the platform modules to ensure uniqueness. 0h = Smallest value FFFFh = Highest possible value
15-12	FEATUREVER	R	Fh	Feature Set for the DMA: number of DMA channel minus one (e.g. 0->1ch, 2->3ch, 15->16ch). 0h = Smallest value (1 channel) Fh = Highest value (16 channel)
11-8	RESERVED	R	0h	
7-4	MAJREV	R	0h	Major rev of the IP 0h = Smallest value Fh = Highest possible value
3-0	MINREV	R	0h	Minor rev of the IP 0h = Smallest value Fh = Highest possible value

20 DMAPRIO Register (Offset = 1100h) [Reset = 00000000h]

DMAPRIO is shown in Figure 10-20 and described in Table 10-23.

Return to the Summary Table.

DMA Channel Priority Control

Figure 10-20. DMAPRIO Register

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

Table 10-43. DMAPRIO Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	RESERVED	R/W	0h	
17-16	BURSTSZ	R/W	0h	Define the burst size of a block transfer, before the priority is re-evaluated 0h = There is no burst size, the whole block transfer is completed on one transfer without interruption 1h = The burst size is 8, after 9 transfers the block transfer is interrupted and the priority is reevaluated 2h = The burst size is 16, after 17 transfers the block transfer is interrupted and the priority is reevaluated 3h = The burst size is 32, after 33 transfers the block transfer is interrupted and the priority is reevaluated
15-1	RESERVED	R/W	0h	
0	ROUNDROBIN	R/W	0h	Round robin. This bit enables the round-robin DMA channel priorities. 0h = Roundrobin priority disabled, DMA channel priority is fixed: DMA0-DMA1-DMA2-...-DMA16 1h = Roundrobin priority enabled, DMA channel priority changes with each transfer

21 DMATCTL_j Register (Offset = 1110h + formula) [Reset = 00000000h]

DMATCTL_j is shown in Figure 10-21 and described in Table 10-24.

Return to the Summary Table.

DMA Trigger Control

Offset = 1110h + (j * 4h); where j = 0h to Fh

Figure 10-21. DMATCTL_j Register

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

Table 10-45. DMATCTL_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R/W	0h	
7	DMATINT	R/W	0h	DMA Trigger by Internal Channel 0h = DMATSEL will define external trigger select as transfer trigger. 1h = DMATSEL will define internal channel as transfer trigger select. 0-> Channel0-done, 1-> Channel1-done, ...
6-0	DMATSEL	R/W	0h	DMA Trigger Select Note: Reference the datasheet of the device to see the specific trigger mapping. 00h = Software trigger request 7Fh = Highest possible value

22 DMACTL_j Register (Offset = 1200h + formula) [Reset = 00000000h]

DMACTL_j is shown in Figure 10-22 and described in Table 10-25.

Return to the Summary Table.

DMA Channel Control

Offset = 1200h + (j * 10h); where j = 0h to Fh

Figure 10-22. DMACTL_j Register

31	30	29	28	27	26	25	24
RESERVED		DMATM		RESERVED		DMAEM	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	
23	22	21	20	19	18	17	16
DMADSTINCR				DMASRCINCR			
R/W-0h				R/W-0h			
15	14	13	12	11	10	9	8
RESERVED	DMADSTWDTH			RESERVED	DMASRCWDTH		
R/W-0h	R/W-0h			R/W-0h	R/W-0h		
7	6	5	4	3	2	1	0
RESERVED	DMAPREIRQ			DMAAUTOEN		DMAEN	DMAREQ
R/W-0h	R/W-0h			R/W-0h		R/W-0h	R/W-0h

Table 10-47. DMACTL_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	RESERVED	R/W	0h	
29-28	DMATM	R/W	0h	DMA transfer mode register Note: The repeat-single (2h) and repeat-block (3h) transfer are only available in a FULL-channel configuration. Please consult the datasheet of the specific device to map which channel number has FULL or BASIC capability. In a BASIC channel configuration only the values for single (0h) and block (1h) transfer can be set. 0h = Single transfer. Each transfers requires a new trigger. When the DMASZ counts down to zero an event can be generated and the DMAEN is cleared. 1h = Block transfer. Each trigger transfers the complete block defined in DMASZ. After the transfer is complete an event can be generated and the DMAEN is cleared. 2h = Repeated single transfer. Each transfers requires a new trigger. When the DMASZ counts down to zero an event can be generated. After the last transfer the DMASA, DMADA, DAMSZ registers are restored to its initial value and the DMAEN stays enabled. 3h = Repeated block transfer. Each trigger transfers the complete block defined in DMASZ. After the last transfer the DMASA, DMADA, DAMSZ registers are restored to its initial value and the DMAEN stays enabled.
27-26	RESERVED	R/W	0h	

Table 10-47. DMACTL_j Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
25-24	DMAEM	R/W	0h	DMA extended mode Note: The extended transfer modes are only available in a FULL-channel configuration. Please consult the datasheet of the specific device to map which channel number has FULL or BASIC capability. In a BASIC channel configuration this register is a read-only register and reads 0x0. 0h = Normal mode is related to transfers from SRC to DST 1h = Gather mode will read a data from an address table located at SA, and the data is transfered to the DA 2h = Fill mode will copy the SA register content as data to DA 3h = Table mode will read an address and data value from SA and write the data to address
23-20	DMADSTINCR	R/W	0h	DMA destination increment. This bit selects automatic incrementing or decrementing of the destination address DMADA for each transfer. The amount of change to the DMADA is based on the definitin in the DMADSTWDTH. For example an increment of 1 (+1) on a WORD transfer will increment the DMADA by 4. 0h = Address is unchanged (+0) 2h = Decremented by 1 (-1 * DMADSTWDTH) 3h = Incremented by 1 (+1 * DMADSTWDTH) 8h = Stride size 2 (+2 * DMADSTWDTH) 9h = Stride size 3 (+3 * DMADSTWDTH) Ah = Stride size 4 (+4 * DMADSTWDTH) Bh = Stride size 5 (+5 * DMADSTWDTH) Ch = Stride size 6 (+6 * DMADSTWDTH) Dh = Stride size 7 (+7 * DMADSTWDTH) Eh = Stride size 8 (+8 * DMADSTWDTH) Fh = Stride size 9 (+9 * DMADSTWDTH)
19-16	DMASRCINCR	R/W	0h	DMA source increment. This bit selects automatic incrementing or decrementing of the source address DMASA for each transfer. The amount of change to the DMASA is based on the definitin in the DMASRCWDTH. For example an increment of 1 (+1) on a WORD transfer will increment the DMASA by 4. 0h = Address is unchanged (+0) 2h = Decremented by 1 (-1 * DMASRCWDTH) 3h = Incremented by 1 (+1 * DMASRCWDTH) 8h = Stride size 2 (+2 * DMASRCWDTH) 9h = Stride size 3 (+3 * DMASRCWDTH) Ah = Stride size 4 (+4 * DMASRCWDTH) Bh = Stride size 5 (+5 * DMASRCWDTH) Ch = Stride size 6 (+6 * DMASRCWDTH) Dh = Stride size 7 (+7 * DMASRCWDTH) Eh = Stride size 8 (+8 * DMASRCWDTH) Fh = Stride size 9 (+9 * DMASRCWDTH)
15	RESERVED	R/W	0h	
14-12	DMADSTWDTH	R/W	0h	DMA destination width. This bit selects the destination as a byte, half word, word, long word or long-long word. 0h = Destination data width is BYTE (8-bit) 1h = Destination data width is HALF-WORD (16-bit) 2h = Destination data width is WORD (32-bit) 3h = Destination data width is LONG-WORD (64-bit) 4h = Source data width is LONG-LONG-WORD (128-bit)
11	RESERVED	R/W	0h	
10-8	DMASRCWDTH	R/W	0h	DMA source width. This bit selects the source data width as a byte, half word, word, long word or long-long word. 0h = Source data width is BYTE (8-bit) 1h = Source data width is HALF-WORD (16-bit) 2h = Source data width is WORD (32-bit) 3h = Source data width is LONG-WORD (64-bit) 4h = Source data width is LONG-LONG-WORD (128-bit)
7	RESERVED	R/W	0h	

Table 10-47. DMACTL_j Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
6-4	DMAPREIRQ	R/W	0h	Enable an early IRQ event. This can help software to react quicker to and DMA done event or allows some additional configuration before the channel is complete. Note: This register is only available in a FULL-channel configuration. Please consult the datasheet of the specific device to map which channel number has FULL or BASIC capability. In a BASIC configuration this register is a read only value and always reads as 0x0. 0h = Pre-IRQ event disabled. 1h = Issure Pre-IRQ event when DMASZ=1 2h = Issure Pre-IRQ event when DMASZ=2 3h = Issure Pre-IRQ event when DMASZ=4 4h = Issure Pre-IRQ event when DMASZ=8 5h = Issure Pre-IRQ event when DMASZ=32 6h = Issure Pre-IRQ event when DMASZ=64 7h = Issure Pre-IRQ event when DMASZ reached the half size point of the original transfer size
3-2	DMAAUTOEN	R/W	0h	Automatic DMA channel enable on DMASA, DMADA, DMASZ register write. If channel is configured as SW trigger (DMATCTL=0), the AUTOEN will set the DMAEN and DMAREQ. If channel is configured as HW trigger (DMACTL!=0), the AUTOEN will only set the DMAEN. 0h = No automatic DMA enable 1h = Automatic DMA enable on DMASA register write. 2h = Automatic DMA enable on DMADA register write. 3h = Automatic DMA enable on DMASZ register write.
1	DMAEN	R/W	0h	DMA enable 0h = DMA channel disabled 1h = DMA channel enabled
0	DMAREQ	R/W	0h	DMA request. Software-controlled DMA start. DMAREQ is reset automatically. 0h = Default read value 1h = DMA transfer request (start DMA)

23 DMASA_j Register (Offset = 1204h + formula) [Reset = 00000000h]

DMASA_j is shown in Figure 10-23 and described in Table 10-26.

Return to the Summary Table.

DMA Channel Source Address

Offset = 1204h + (j * 10h); where j = 0h to Fh

Figure 10-23. DMASA_j Register

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

Table 10-49. DMASA_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ADDR	R/W	0h	DMA Channel Source Address 0h = Smallest value FFFFFFFFh = Highest possible value

24 DMADA_j Register (Offset = 1208h + formula) [Reset = 00000000h]

DMADA_j is shown in Figure 10-24 and described in Table 10-27.

Return to the Summary Table.

DMA Channel Destination Address

Offset = 1208h + (j * 10h); where j = 0h to Fh

Figure 10-24. DMADA_j Register

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

Table 10-51. DMADA_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ADDR	R/W	0h	DMA Channel Destination Address 0h = Smallest value FFFFFFFFh = Highest possible value

25 DMASZ_j Register (Offset = 120Ch + formula) [Reset = 00000000h]

DMASZ_j is shown in Figure 10-25 and described in Table 10-28.

Return to the Summary Table.

DMA Channel Size

Offset = 120Ch + (j * 10h); where j = 0h to Fh

Figure 10-25. DMASZ_j Register

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

Table 10-53. DMASZ_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R/W	0h	
15-0	SIZE	R/W	0h	DMA Channel Size in number of transfers 0h = Smallest value FFFFh = Highest possible value

ADVANCE INFORMATION

Chapter 11
Flash Module

This chapter describes the nonvolatile flash memory module of the device.

11.1 Flash (NVM)

The Flash module provides nonvolatile memory (NVM) for storing executable code and data.

ADVANCE INFORMATION

11.1.1 Introduction to Flash and OTP Memory

Flash is an electrically erasable/programmable nonvolatile memory that can be programmed and erased many times to ease code development. Flash memory can be used primarily as a program memory for the core, and secondarily as static data memory. This section describes the proper sequence to configure the wait states and operating mode of Flash. This section also includes information on one-time programmable (OTP) Flash, how to improve Flash performance by enabling the Flash cache mode, and the SECDED safety feature. The nonvolatile memory system provides in-system programmable flash memory for storing executable code and data. This chapter describes the functionality provided by the nonvolatile memory system.

11.1.1.1 Flash Features

Key features of the Flash nonvolatile memory system include:

- Two Flash Banks up to 256KB each (refer to the device data sheet for flash sizes)
- In-circuit program and erase supported across the entire supply voltage range
- Flash commands for Program, Erase, Mode Change, Read-verify, and Blank-verify functions
- Internal programming voltage generation
- 128-bit flash word size (144-bit when ECC is present)
- Static write protection (latched at boot and held until Power on Reset)
- Dynamic write protection (configurable at runtime)
- Sector (1KB) and Bank (up to 256KB) Erase
- Automatic hardware pre-verification to extend flash bank longevity
- Automatic hardware post-verification of program/erase
- ECC Single Error Correction Dual Error Detection (SECDED) protection
- Bank address swap mode (for OTA dual-image firmware updates)
- Program register cache for time efficient programming (2, 4, or 8 flash words)

11.1.1.2 System Components

The nonvolatile memory system consists of three components (listed below):

- One or more flash memory banks (for storing code and data)
- The flash controller (for managing all program/erase operations on the flash banks)
- The read interface (for interfacing the flash banks to the CPU and peripheral bus)

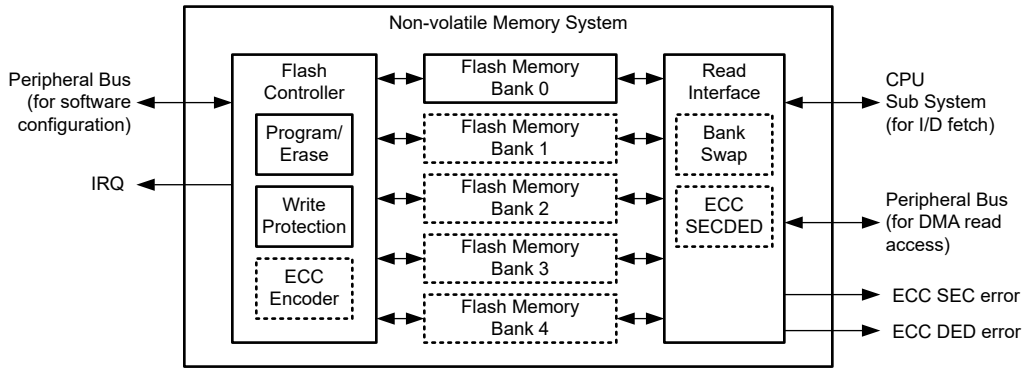


Figure 11-1. Non-volatile Memory System Components

Note

AM13E23x devices have 2 flash memory banks.

11.1.1.3 Terminology

Key flash bank terms are defined in this section to be used as a reference for the rest of this chapter.

ADVANCE INFORMATION

Table 11-1. NVM System Terminology

Term	Definition	Size
Flash word	Basic data size for program and read operations on the flash memory (also the read bus width to the system)	128 data bits (144 bits with ECC)
Word line	Group of flash words within a sector, with maximum program operation limit before sector erase	16 flash words (128 data bytes, optionally 16 ECC bytes)
Sector	Group of word lines that are erased together (minimum erase resolution of the flash memory)	8 word lines (1024 data bytes, optionally 128 ECC bytes)
Bank	Group of sectors that can be mass erased in one operation. Only one read, program, erase, or verify operation can run concurrently on a given bank.	Variable
Region	Logical assignment of a region of flash memory from a bank.	Variable

11.1.2 Flash Memory Bank Organization

The flash memory is used for storing application code and data, the device boot configuration, and parameters which are preprogrammed by TI from the factory. The flash memory is organized into one or more banks, and the memory in each bank is further mapped into one or more logical memory regions and assigned system address space for use by the application.

11.1.2.1 Banks

The nonvolatile memory system provides support for up to 5 flash memory banks (enumerated as BANK0 through BANK4). The number of flash banks present is device dependent. AM13E23x devices implement 2 flash memory banks (BANK0 and BANK1).

On AM13E23x devices, a program/erase operation on a bank will also stall read requests issued to the bank which is executing the program/erase operation, but it will not stall read requests issued to any other bank. As such, the presence of multiple banks enables application cases such as:

- Dual-image firmware updates (an application can execute code out of one flash bank while a second image is programmed to a second symmetrical flash bank without stalling the application execution)
- EEPROM emulation (an application can execute code out of one flash bank while a second flash bank is used for writing data without stalling the application execution)

11.1.2.2 Flash Memory Regions

The memory within each bank is mapped to one or more logical regions based upon the functions that the memory in each bank supports. There are four regions: FACTORY, NONMAIN (Configuration NVM), MAIN (Flash Memory), and DATA.

Table 11-2. Flash Memory Regions

Flash Memory Region	Region Contents	Executable	Used by	Programmed by
FACTORY	Device ID and other parameters	No	Boot ROM Application	TI Only (Read-Only)
NONMAIN (Configuration NVM)	Device boot configuration (BCR and BSL)	No	Boot ROM	TI, User
MAIN (Flash Memory)	Application code and data	Yes	Application	User
DATA	Data, or EEPROM emulation	No	Application	User

Devices with one bank implement the FACTORY, NONMAIN, and MAIN regions on BANK0 (the only bank present), and the data region is not available. Devices with multiple banks also implement FACTORY, NONMAIN, and MAIN regions on BANK0, but include additional banks (BANK1 through BANKx) that can implement MAIN or DATA regions.

All devices include a memory-mapped FACTORY region which provides read-only data describing the capabilities of a device as well as any factory-provided trim information for use by application software.

11.1.2.3 Addressing

The flash memory regions are assigned to address space in the system memory map.

The NONMAIN, DATA, and FACTORY regions are assigned to the peripheral address space (0x6000_0000) as the memory does not contain any executable code. The CPU can not fetch executable instructions from this region.

The MAIN region is assigned to both the code address space (0x0000_0000) and the peripheral address space (0x6000_0000). Instruction and data fetches are recommended to always be done through the code address space as this gives the best performance. The CPU can not fetch executable instructions from this region.

Error Correction Code (ECC)

For error correction code (ECC) support, the ECC codes for all memory regions are also assigned to an address space and software can read the ECC codes as data for diagnostic purposes. It is also possible to read the contents of any of the memory regions without ECC correction applied.

11.1.2.3.1 Flash Memory Map

The following table lists the system address space assignments. These assignments are consistent for all devices.

Table 11-3. Flash Region Memory Map

Region	Read Type	ECC Behavior	Base Address
NONMAIN	Data read	Corrected	0x6010.0000
		ECC code	0x6026.0000
MAIN	Instruction fetch or data read	Corrected	0x0000.0000
	Data read	ECC code	0x6020.0000
DATA	Data read	Corrected	0x6011.8000
		ECC code	0x6028.0000
FACTORY	Data read	Corrected	0x6011.0000
		ECC code	0x6027.0000

NONMAIN, DATA, and FACTORY data reads are processed through the peripheral bus and peripheral address space only. MAIN regions can be accessed through either the CPU bus matrix or through the peripheral bus, depending on whether code address space or peripheral address space is used. The code address space is recommended for CPU accesses (instruction fetches or data reads), as these accesses do not cross the peripheral bus and thus do not compete with the DMA for control of the peripheral bus. See the bus architecture section for a detailed description of the bus interconnect.

For ECC, an access to an ECC code address returns the 16-bit ECC value for the entire 128-bit flash word that was accessed.

11.1.3 Flash Controller

The flash controller manages all program, erase, and verification operations performed on the nonvolatile memory system. Memory-mapped registers exist in the peripheral region of the device memory map which must be configured by software to perform operations on the flash memory.

TI provides software abstraction for the flash controller as a part of the DriverLib layer of the software development kit (SDK). TI recommends using the DriverLib abstraction layer when operating on the flash memory with software, but this is not a mandatory requirement. To use the DriverLib software abstraction layer to perform operations on the flash memory, review the software development kit (SDK) documentation provided separately from this document. To directly operate on the flash memory with using low level register accesses to the flash controller, review the remainder of this section in detail.

Note

The flash control registers (FLASH_CTRL_REGS and NVMNW_REGS) are not always be configured to default values after a reset. This can occur if the boot configuration routine (BCR) or boot strap loader (BSL) performs an operation on the flash memory during boot. When configuring the flash control registers for an operation, verify that all registers which are relevant for the operation are correctly configured.

11.1.3.1 Overview of Flash Controller Commands

Operations on the flash memory are executed by configuring the command type (CMDTYPE) and command control (CMDCTL) registers for the desired command, along with any other registers which must be configured for a particular command, and writing 0x01 to the command execute (CMDEXEC) register to initiate the command.

When 0x01 is set in CMDEXEC, the commanded operation begins executing. While an operation is executing, most configuration registers are blocked for writes until the operation completes. Some registers (for example, mask registers) can change state under hardware control while the operation runs to completion. The flash controller indicates completion of the commanded operation by setting the CMDDONE bit in the command status (STATCMD) register. The flash controller also sources an interrupt vector to the CPU subsystem to indicate a “DONE” status when an operation has completed.

The software sequence of setting the CMDEXEC bit and waiting for the CMDDONE response must be executed from either the device SRAM or from a different flash bank from the bank that is being operated on, as the flash controller takes control of the flash bank undergoing the operation. Reads to the flash bank that is being operated on while the flash controller is executing the command are not predictable.

The flash controller provides five basic commands for operating on the flash memory, specified in the COMMAND field of the CMDTYPE register. These commands are described in Table 11-4.

Table 11-4. Flash Controller Commands

Command	Description
NOOP	No operation (default setting).
PROGRAM	Selects a program operation on the flash memory.
ERASE	Selects an erase operation on the flash memory.
READVERIFY	Selects a standalone read verify operation.

There are two additional commands: Mode Change for only changing the bank and pump modes (which are otherwise set automatically by hardware) and no other operation; and Clear Status.

Note

TI recommends calling the clear status command (CMDTYPE = 0x5) before calling any flash operation.

11.1.3.2 Command Diagnostics

The flash controller updates several software-readable registers to communicate information about an initiated operation.

11.1.3.2.1 Command Status

The STATCMD register is a read-only register which provides diagnostic information about an operation which is been initiated or completed. The CMDINPROGRESS bit indicates that an operation is currently ongoing. The CMDDONE bit indicates that an operation has completed. These bits can be polled by software to determine the state of the flash controller during operations.

In addition, the STATCMD register has the following bitfields for additional diagnostic information:

- CMDPASS: Command pass bit to indicate if a command passed (valid when CMDDONE is 1).
- FAILWEPROT: Indicates the command failed due to Write/Erase protect sector violation.
- FAILVERIFY: Indicates the command failed due to verify error.
- FAILILLADDR: Indicates the command failed due to the use of an illegal address.
- FAILMODE: Indicates the command failed because a bank is set to a mode other than READ. Program and Erase commands cannot be initiated unless all banks are in READ mode.
- FAILINVDATA: Indicates that the PROGRAM command failed because a program attempted to store a 0 value as a 1.
- FAILMISC: This is an extra bit that indicates the command failed due to error outside of what is currently defined.

11.1.3.2.2 Address Translation

The address status (STATADDR) register is a read-only register which can be read to determine the current bank ID, region ID, and bank address which the flash controller is pointing to. These values can increment during execution of certain commands, in which case the value present after the completion of a command indicates the last address touched by the flash controller.

11.1.3.2.3 Pulse Counts

The pulse count status (STATPCNT) register is a read-only register which can be read to determine the current pulse count applied during a program or erase operation.

11.1.3.3 NOOP Command

When not using the flash controller, it is best to set the COMMAND field to NOOP to prevent any unintentional operations on the flash memory in the event that the CMDEXEC bit is unintentionally set. Executing a NOOP command has no effect on the flash memory.

11.1.3.4 PROGRAM Command

The program command is used to write (program) the flash memory. Specifically, the purpose of a PROGRAM operation is to configure the flash bits in one or more flash words from the nondeterministic erased state to the deterministic programmed state. Once a byte is programmed using the PROGRAM command, the byte can not be re-programmed unless the sector is erased using the ERASE command.

All devices support single flash word programming of 128 data bits (plus 16 ECC bits) at a time, with control to limit the scope of a program operation to specific bytes within a 128-bit flash word.

AM13E23x devices additionally have support for a multi-word programming mode where 2 or 4 flash words can be written with a single commanded operation. Multi-word programming, when available, significantly speeds up programming when multiple words need to be programmed (for example, during production programming or firmware updates).

11.1.3.4.1 Program Bit Masking Behavior

The flash controller provides a program verification mechanism to extend the lifetime of the flash bank. During program operations, the command data (CMDDATAx) registers are used by the flash controller as a programming bit mask to indicate which specific bits in the flash word require program pulses. As a result, data which is loaded into the CMDDATAx registers before starting the program operation will be lost from the CMDDATAx registers during and/or upon completion of the program operation. If the same data is to be programmed again, the CMDDATAx registers must be re-loaded by software with the correct data values to be programmed.

11.1.3.4.2 Target Data Alignment

There are two options for loading data to be programmed: direct mode or indexed mode. The programmer must select the mode that is most suitable to the application requirements.

Additional alignment rules apply when loading data into the CMDDATAx and CMDECCx registers on devices that support multiword programming, even if the multiword programming feature is not used and only single word programming is not used.

- 1-word program operations must have CMDADDR (the target system address) aligned to a 0b000 boundary (for example, the 3 LSBs in CMDADDR must be zero).
- 2-word program operations must have CMDADDR (the target system address) aligned to a 0b0000 boundary (for example, the 4 LSBs in CMDADDR must be zero).
- 4-word program operations must have CMDADDR (the target system address) aligned to a 0b0.0000 boundary (for example, the 5 LSBs in CMDADDR must be zero).

Direct Data Load

To configure a program operation with direct data loading, the target data is loaded into the appropriate CMDDATAx registers based on the number of flash words supported by the device, the target address alignment, and the target data size. For example, if a 4-word program operation is to be initiated on a device supporting 4-word programming, the CMDDATA0-CMDDATA7 registers are populated with the target data. If ECC is being specified directly (rather than automatically calculated by the flash controller) then the appropriate CMDECCx registers also needs to be populated with the ECC values for each data word being programmed.

Indexed Data Load

Rather than buffering data into all the CMDDATAx registers individually, it is possible to use only the CMDDATA0-CMDDATA1 registers in combination with an index register (CMDDATAINDEX) to indicate the flash word offset of the data being loaded. In this way, the index can be adjusted for each word loaded, and each target data word can be written to the same 128-bit space (CMDDATA0-CMDDATA1). When an index is applied, the loaded data is mapped by the hardware into the appropriate CMDDATAx register. For example, if a 4-word program operation is to be initiated on a device supporting 4-word programming, the CMDDATA1:0 registers are loaded 4 times with the target data, with CMDDATAINDEX being incremented by one before each new word is loaded into CMDDATA1:0.

Alignment Rules

The alignment rules for each device configuration (2 or 4 words) is given in the tables below with guidance on how target data must be placed with the CMDDATAx, CMDECCx, and CMDDATAINDEX registers for 1, 2, or 4 word programming operations.

Table 11-5. Data Load Alignment for Programming of 2 Flash Words

Direct Load Registers	Indexed Load Index	1 Word Aligned to 0b000	2 Words Aligned to 0b0000	4 Words Aligned to 0b0.0000
CMDDATA1:0 / CMDECC0	CMDINDEX = 0	Target data word 0 if the target address ends in 0b0000	Target data word 0	Not supported
CMDDATA3:2 / CMDECC1	CMDINDEX = 1	Target data word 0 if the target address ends in 0b1000	Target data word 1	

Table 11-6. Data Load Alignment for Programming of 4 Flash Words

Direct Load Registers	Indexed Load Index	1 Word Aligned to 0b000	2 Words Aligned to 0b0000	4 Words Aligned to 0b0.0000
CMDDATA1:0 / CMDECC0	CMDINDEX= 0	Target data word 0 if the target address ends in 0b0.0000	Target data word 0 if the target address ends in 0b0.0000	Target data word 0
CMDDATA3:2 / CMDECC1	CMDINDEX= 1	Target data word 0 if the target address ends in 0b0.1000	Target data word 1 if the target address ends in 0b0.0000	Target data word 1
CMDDATA5:4 / CMDECC2	CMDINDEX= 2	Target data word 0 if the target address ends in 0b1.0000	Target data word 0 if the target address ends in 0b1.0000	Target data word 2
CMDDATA7:6 / CMDECC3	CMDINDEX= 3	Target data word 0 if the target address ends in 0b1.1000	Target data word 1 if the target address ends in 0b1.0000	Target data word 3

11.1.3.4.3 Executing a PROGRAM Operation

To program the flash memory:

1. Select the command in the CMDTYPE register:

ADVANCE INFORMATION

a. Set the COMMAND field in the CMDTYPE register to PROGRAM.
b. Set the SIZE field in the CMDTYPE register to the desired size (1, 2, or 4 flash words). The hardware does not check for invalid configuration of the SIZE field; software must verify that the selection option is supported by the device. Note that SECTOR and BANK sizes are not valid sizes for PROGRAM operations. These sizes only apply to erase operations.

2. Configure the program command in the CMDCTL register:
   a. On devices with ECC, the flash controller by default generates the needed ECC bits from the data during the PROGRAM operation. Optionally, software can override the hardware ECC code generation and manually provide the ECC code to be programmed by setting the ECCGENOVR bit in CMDCTL register.
3. Select the target programming address in the CMDADDR and CMDBYTEN register:
   a. Load the target system address into the CMDADDR register to indicate the base address from which programming starts. The target address must be a flash word address (128-bit aligned). The flash controller translates the system address into the applicable flash region, bank ID, and bank address. If desired, after the operation completes the flash region, bank ID, and bank address can be read from the STATADDR register. In a multi-word program, STATADDR indicates the bank ID and the final address programmed.
   b. If subword programming (programming of less than the full 128 or 144 bit flash word) is desired, configure the CMDBYTEN register to set the bytes within the addressed flash word which are to be programmed. Each bit in CMDBYTEN corresponds to a byte in the addressed flash word to be programmed, including the ECC byte. For example, programming of the ECC code can be masked by clearing bit 8 in CMDBYTEN while programming the data bytes of the flash word. Note that there is a maximum number of program operations allowed per word line before a sector erase must be applied (see the device specific data sheet for the maximum).
4. Load the data to program into the CMDDATAx registers:
   a. For a single flash word programming operation (128 or 144 bits depending on the presence of ECC), load the data into the CMDDATAx registers consistent with the alignment requirements (for devices which only support single-word programming, CMDDATA0 and CMDDATA1 are always used regardless of the target address).
   b. For multi-word programming (if available and selected), load data into the CMDDATAx registers consistent with the alignment rules and the size of the multi-word program operation specified in step 1.
   c. If ECCGENOVR in the CMDCTL register is set above (disabling hardware ECC code generation), then write the ECC data in the CMDDATAECC0 register (for single word programming) and optionally additional CMDDATAECCx registers as applicable for multi-word programming.
   d. Note that the CMDDATA registers are used as bit masking registers by the flash controller during the program operation; after the operation completes, the values written to these registers are overwritten by the flash controller.
5. Validate the write protection scheme is configured to allow writes to the target addresses (see the write protection section of this guide for additional information on configuring write protection).
6. Execute the program operation by writing 0x1 to the CMDEXEC register.
7. Monitor for completion of the program operation:
   a. The STATCMD register can be polled to determine the status of the program operation. The CMDINPROGRESS bit is set by hardware as soon as the command is initiated. The CMDDONE bit is set when the operation terminates.
   b. When CMDDONE is set, the CMDPASS bit is reset or set at the same time to indicate whether the operation completed successfully or failed. If a program attempted to load on a protected region, the FAILWEPROT or FAILILLADDR bit is asserted depending on dynamic or static write protection. If the program operation cannot be completed successfully within the maximum program pulse count limit FAILVERIFY is asserted. See the device-specific data sheet for maximum program times.
8. After completion of a program operation, the flash controller configures several settings:
   a. All dynamic write protection registers are set to a protected state (to protect against inadvertent programming)

b. All data registers are set to 1.
c. All program byte enables are cleared to 0.

9. Following programming of the flash memory, there can be stale data in the processor's cache logic. Before reading programmed locations, TI recommends to first flush the cache in the CPU subsystem.

Note

Direct user configuration is not required. The TI Flash API takes care of these steps for the user. This information is provided for debug/reference.

11.1.3.5 ERASE Command

The erase command is used to erase individual sectors of flash memory (for MAIN, NONMAIN, or DATA regions) or a complete bank of flash memory (for MAIN regions only). From this erased state, bits can later be programmed to a ‘0’ state or a '1' state as desired using the PROGRAM command. It is not possible to erase with a resolution lower than one sector (1KB) with sector alignment. For devices with multiple banks, a bank erase must be executed on all banks to erase the entire MAIN region on the device.

11.1.3.5.1 Erase Sector Masking Behavior

The flash controller provides an erase verification mechanism to extend the lifetime of the flash bank. The CMDWEPROTx registers are used as an erase mask and are manipulated by the flash controller during the execution of the erase operation. At the end of all erase operations, the CMDWEPROTx registers are set to a fully protected state to prevent against inadvertent programming and must be re-configured before attempting another program or erase operation.

11.1.3.5.2 Executing an ERASE Operation

To erase a sector or bank of the flash memory, follow these steps:

1. Select the command in the CMDTYPE register:
   a. Set the COMMAND field in the CMDTYPE register to ERASE.
   b. Set the SIZE field in the CMDTYPE register to SECTOR or BANK. Sizes other than SECTOR or BANK (for example, ONEWORD) are not supported by the ERASE command. It is the responsibility of software to check the configuration before issuing an ERASE command.
2. Select the target erase address in the CMDADDR register:
   a. Store the target system address into the CMDADDR register to indicate the base address of the sector or bank to be erased. When performing a bank erase with write protection enabled (such that only unprotected sectors are erased), ensure that the address written to CMDADDR is in an unprotected sector. The flash controller will translate the system address into the applicable flash region, bank ID, and bank address. If desired, after the operation completes the flash region, bank ID, and bank address can be read from the STATADDR register.
3. Ensure the write protection scheme is configured to allow writes to the target sectors (see the write protection section of this guide for additional information on configuring write protection).
4. Execute the erase operation by writing 0x1 to the CMDEXEC register.
5. Monitor for completion of the erase operation:
   a. The STATCMD register can be polled to determine the status of the erase operation. The CMDINPROGRESS bit will be set by hardware as soon as the command is initiated. The CMDDONE bit will be set when the operation terminates. When CMDDONE is set, the CMDPASS bit will be reset or set at the same time to indicate whether the operation completed successfully or failed.
   b. If an erase was attempted on a protected region, the FAILWEPROT bit is asserted.
   c. If the erase operation cannot be completed successfully within the maximum erase pulse count limit, FAILVERIFY will be asserted.
6. After completion of the erase operation, the flash controller will configure several settings to protect against inadvertent programming:

ADVANCE INFORMATION

a. All dynamic write protection registers are set to a protected state.

11.1.3.6 READVERIFY Command

The read verify command can be used to read a flash location and compare the data to data which is preloaded into the CMDDATA registers of the flash controller. The command can be applied to a single flash word, multiple flash words (if the device supports multi-word programming), an entire sector, or an entire bank. When performing a read verify on an entire sector or bank, the data in CMDDATAx will be re-used.

11.1.3.6.1 Executing a READVERIFY Operation

To execute a read verify command, follow these steps:

1. Select the command in the CMDTYPE register:
   a. Set the COMMAND field in the CMDTYPE register to READVERIFY.
   b. Set the SIZE field in the CMDTYPE register to the desired size.
2. Configure the read verify command in the CMDCTL register:
   a. If the desire is to manually provide ECC bits along with the data, set the ECCGENOVR bit in the CMDCTL register. If ECCGENOVR is cleared, the flash controller will generate ECC bits for comparison based on the provided compare data.
3. Select the target address to verify on the CMDADDR register:
   a. Load the target system address into the CMDADDR register to indicate the base address to be verified. The flash controller will translate the system address into the applicable flash region, bank ID, and bank address. If desired, after the operation completes the flash region, bank ID, and bank address can be read from the STATADDR register.
4. Load the data to verify into the CMDDATAx registers:
   a. For single word verification, write the data to be verified to the CMDDATA0 and CMDDATA1 registers. For multi-word verification, if available on the target device, write the data to be verified to the appropriate CMDDATAx registers beyond CMDDATA0 and CMDDATA1.
5. Configure the byte enable settings in the CMDBYTEN register:
   a. Any CMDBYTEN bit set to logic “0” will mask the associated data byte from being compared during the execution of the READVERIFY command. This can be used to verify data less than one flash word (less than 64 bits).
6. Execute the read verify operation by writing 0x1 to the CMDEXEC register.
7. Monitor for completion of the read verify operation:
   a. The STATCMD register can be polled to determine the status of the erase operation. The CMDINPROGRESS bit will be set by hardware as soon as the command is initiated. The CMDDONE bit will be set when the operation terminates. When CMDDONE is set, the CMDPASS bit will be reset or set at the same time to indicate whether the read verification passed or failed.
   b. The FAILVERIFY bit in STATCMD will be set if any data read from the flash did not match the expected data loaded in CMDDATAx.
8. After completion of the read verify operation, the flash controller will configure several settings:
   a. All dynamic write protection registers are set to a protected state (to protect against inadvertent programming).
   b. All data registers are set to ‘1’s.
   c. All program byte enables are cleared to ‘0’s.

11.1.3.7 Overriding the System Address With a Bank ID, Region ID, and Bank Address

Normally, flash controller commands are targeted to a specific flash location by loading a system memory map address into the CMDADDR register. This is the recommended way to specify the target address for a PROGRAM, ERASE, or READVERIFY command. In this mode of operation, the flash controller automatically

translates the system address into the corresponding bank ID, region ID, and bank address which are used to execute the command on the flash memory. Application software does not need to specify these items individually; only the system address is needed.

However, in some circumstances direct specification of the target flash bank, region, and address in the specified bank/region is needed. For example, if the desire is to erase a complete bank when doing a mass erase operation on the device, application software actually does not need to have any knowledge of the system address of the bank to be erased- the software only needs to specify the bank ID and region ID to the flash controller to erase the bank.

To use the flash controller to execute a command in address override mode, set the ADDRXLATEOVR bit in the CMDCTL register, and specify the bank ID, region, and bank address before executing the command. To return to system addressed mode, clear ADDRXLATEOVR. ADDRXLATEOVR is cleared by default (supporting system address operation).

Example Case - Bank Erase with ADDRXLATEOVR

To erase the MAIN region of BANK0 by specifying the bank ID and region instead of the system address, follow the steps in Section 11.1.3.5.2 , but replace step 2 with the alternate steps given below:

1. Set the ADDRXLATEOVR bit in CMDCTL to enable address translation override mode
2. Specify BANK0 by setting the BANKSEL field to 0x1 in the CMDCTL register
3. Specify the MAIN region by setting the REGIONSEL field to 0x1 in the CMDCTL register
4. Set the CMDADDR register to 0x0000.0000

11.1.3.8 FLASHCTL Events

The flash controller contains one event publisher (CPU_INT) to manage FLASHCTL interrupt requests (IRQs) to the CPU subsystem through a static event route.

Table 11-7 summarizes the FLASHCTL events.

Table 11-7. FLASHCTL Events

Event	Type	Source	Destination	Route	Configuration	Functionality
CPU Interrupt Event	Publisher	FLASHCTL	CPU Subsystem	Static route	CPU_INT registers	Fixed interrupt route from FLASHCTL to CPU

11.1.4 Write Protection

The flash controller provides two write protection mechanisms (one static, one dynamic) which are applied in parallel (logical OR) to protect user-specified sectors during any attempted program or erase operation. If a program or erase operation is issued to a write protected flash sector, the operation terminates with a FAILWEPROT error reported in the STATCMD register.

Note

This is the lowest priority of Write Protection, after Global Security Controller configurations of 2 layers (Life-cycle HW Blocking + WDPROC Blocking). Refer to the GSC Chapter for additional details.

11.1.4.1 Write Protection Resolution

The write protection resolution for both the static and dynamic write protection mechanisms is dependent on the flash bank and memory region which is being protected.

Flash Bank	Memory Region	Write Protection Resolution
0	NONMAIN	512B (Entire Region)
	MAIN	1KB (1 sector) for first 32KB (32 sectors) 8KB (8 sectors) for remaining sectors

ADVANCE INFORMATION

Flash Bank	Memory Region	Write Protection Resolution
1	NONMAIN	512B (Entire Region)
	MAIN	8KB (8 sectors)
	DATA	1KB (1 sector)

11.1.4.2 Static Write Protection

The static write protection scheme is configured and latched during device boot by the immutable ROM boot code before the main application in flash can execute. After a flash sector is configured to be write protected through the static write protection mechanism, software cannot remove the protection at runtime without a reboot. Thus, sectors protected by static write protection can be thought of as immutable after boot. This type of protection is useful for protecting a custom bootloader in a dual-image application, or for extending the secure root of trust from the ROM boot code into a portion of the main flash region to enable a secure boot manager with locked public keys.

The static write protection scheme is configured by programming the appropriate bits in the NONMAIN flash region, which is read by the boot code before the main application starts. The NONMAIN flash sector can be statically write protected, resulting in a system in which the static write protection scheme is fully permanent and cannot be modified. If the NONMAIN sector is configured to be statically protected, along with any other flash sectors, all statically protected flash sectors can be functionally viewed as read only memory which cannot be updated by any means.

Static write protection can be configured for all sectors of flash memory across all banks present on a device. Program or erase static write protected area will cause illegal address failure, and the FAILILLADDR bit in STATCMD will be set. Instructions for configuring the static write protection, along with the rest of the NONMAIN region, are given in *Boot Configuration*.

11.1.4.3 Dynamic Write Protection

The dynamic write protection scheme is intended to be configured at runtime by software. This scheme provides a simple way for application software to specify sectors to protect from modification by any program or erase operations that are issued with the flash controller. Unlike the static write protection mechanism, the dynamic mechanism is not lockable and thus does not provide any level of data security.

There are two primary uses for dynamic write protection. First, it provides an extra level of robustness against unintentional program or erase of specified sectors in applications that involve in-system programming for either firmware updates or EEPROM emulation. Second, it provides a way to simplify a situation where a bank erase is desired but a small number of sectors should not be erased when a bank erase command is issued. One example would be an application running on a single-bank device in which most of the MAIN region sectors are used to store the executable image, but a few sectors are used to store device-specific data that should not be erased during a firmware update. In that case, the sectors containing the device-specific data can be protected with dynamic write protection, and a bank erase command can be issued to erase all other sectors. This has the benefit that the majority of the MAIN region can be erased with a single command (a bank erase) rather than individual sector-by-sector commands, which would have a longer overall erase time and use more energy.

The dynamic write protection scheme is configured by setting up the CMDWEPROTx registers in the flash controller. The CMDWEPROTx registers cover one flash bank at a time. This means that these registers must be configured with knowledge of which flash bank an attempted program or erase operation will be applied to. Note that the CMDWEPROTx registers are reset to a protected state at the end of all program and erase operations. These registers must be re-configured by software before a new operation is initiated. Program or erase dynamic write protected area will cause write/erase protction failure, and the FAILWEPROT bit in STATCMD will be set.

11.1.4.3.1 Configuring Protection for the MAIN Region

The CMDWEPROTA register is used to configure the dynamic write protection for the first 32 sectors (32KB) of the MAIN region in BANK0. Each bit corresponds to one sector of the MAIN region, starting from the beginning of the MAIN region. CMDWEPROTA is only applicable to operations on the lower 32 sectors of the MAIN region of BANK0 (sectors 0-31). It is not used during program/erase operations applied to other sectors.

The CMDWEPROTB register is also used to configure the dynamic write protection for the MAIN region. There are two modes in which CMDWEPROTB is applied, depending on whether the program/erase operation is being applied to BANK0 or BANK1. In the case of a program/erase operation on BANK0, CMDWEPROTB protects sectors 32-255 in 8-sector increments (1 bit per 8KB), starting from BIT4. BIT0-BIT3 in CMDWEPROTB are ignored. The lower 32 sectors (sectors 0-31 of BANK0) are protected by CMDWEPROTA. In the case of a program/erase operation on BANK1, CMDWEPROTB protects sectors 0-255 in 8-sector increments (1 bit per 8 sectors), starting from BIT0 in CMDWEPROTB.

11.1.4.3.2 Configuring Protection for the NONMAIN Region

The CMDWEPROTNM register protects the NONMAIN region from program and erase. One protection bit is provided per sector.

11.1.5 Flash Read Interface

The Flash subsystem includes one Flash Read Interface modules (FRI-1). The Flash read interface modules provide a means for the various initiators on the system to perform read operations on Flash memory. These initiators include:

- **M33 CPU**: The CPU has one 128-bit instruction fetch bus, and two 64-bit data buses.
- **DMA**: The DMA module can perform 64-bit data reads of Flash memory.
- **Debug Interface**: A debugger connected to M33 CPU can perform data read accesses to Flash memory locations.

The Flash read interface is addressable through one or more Flash read ports. Because of bank interleaving and support for bank swapping during firmware updates, Flash code banks do not have a fixed mapping to CPU address space. Rather, each read port has a fixed address range, but can map to a different Flash region depending on the current system configuration. The Flash bank access router performs the background translation, routing read access requests to the intended bank or banks as directed by the Global Security Controller (GSC). The are two types of Flash read operations that can be performed: program instruction fetches for CPU execution, and data accesses.

FRI-1: CPU1 Program

FRI-1 interfaces with the instruction fetch bus of the M33 CPU, and is primarily used for executing program code on the CPU. Additionally, FRI-1 is connected to all data read initiators in the system. This read interface has four read ports, numbered RP0 to RP3. These read ports are mapped to code banks depending on the configuration of BANKMODE and BANKMAP.CPU1SWAP registers in the SSU. These settings are described in subsequent sections.

FRI-3: Firmware Update Region

FRI-3 provides an address range to be used for programming and verifying an updated firmware image into the second half of Flash memory, while the current application continues to execute from the first half through FRI-1 and FRI-2 (if present). When the firmware update is complete, the Flash memory ranges can then be swapped using BANKMAP.CPUxSWAP, and the newly programmed firmware now executes from FRI-1 and FRI-2 (if present). This supports Firmware-Over-The-Air (FOTA) and Live Firmware Update (LFU) firmware upgrade mechanisms. FRI-3 does not have an instruction fetch interface—only data reads are possible from all system initiators. FRI-3 has two read ports, numbered RP0 to RP1, which are mapped to available Flash code memory.

FRI-4: Data Bank

FRI-4 is exclusively used for the data bank (FLC1.B4), and does not have an instruction fetch interface. Thus, application code cannot be executed from the data bank; only read accesses are possible from all system initiators. FRI-4 has one read port (RP0) which has a fixed mapping to the data Flash bank, and does not change based on the BANKMODE or BANKMAP.CPUxSWAP settings.

11.1.5.1 Bank Modes and Swapping

The BANKMODE register in the SSU_GEN_REGS register range configures the system Flash bank mode setting. These modes determine the allocation of Flash memory to primary CPUs in the system, and whether the swapping feature is enabled for firmware updates. The BANKMODE register is loaded from the BANKMGMT sector of the active code bank pair in FLC1 during device boot. The active BANKMGMT sector is determined based on the values of the BANK_STATUS and BANK_UPDATE fields; afterward, the winning BANKMGMT sector is used to configure the system BANKMODE.

The available bank mode options are described in Table 11-8.

Table 11-8. CPU Bank Modes

BANKMODE	Flash Mapping	Swap Enabled	1-CPU Devices
0	All program Flash mapped to CPU1	No	Available
1		Yes	Available
2	Reserved	N/A	N/A
3		N/A	N/A

The FLBANKSWAP.USEUPPER register bitfield in SYSCTL is used to determine the current active A/B swap setting for firmware updates using FOTA or LFU.

For specific mappings of Flash address ranges to read port addresses refer to the device data sheet.

11.1.5.2 Flash Wait States

When a CPU performs a read access to a Flash memory address, data is returned after (RWAIT + 1) MCLK cycles.

For an access to the SECCFG or BANKMGMT Flash regions, data is returned after 10 MCLK cycles.

RWAIT defines the number of random access wait states, and is configured using the RWAIT field in the FRDCNTL register. At reset, RWAIT defaults to a value of 2. RWAIT can be reconfigured to a lower value when the CPU clock frequency is low enough to accommodate the Flash access time. For a table of supported RWAIT values versus CPU clock frequency ranges, refer to the device data sheet.

For a given system clock frequency, configure RWAIT using the following formula:

$$\text{RWAIT} = \text{ceiling}\left(\frac{\text{MCLK}}{\text{FCLK}} - 1\right) \quad (10)$$

where MCLK is the system operating frequency, and FCLK is the Flash clock frequency.

FCLK must be ≤ FCLK_{max}, the allowed maximum Flash clock frequency defined in the device data sheet.

If RWAIT results in a fractional value when calculated using the above formula, round up RWAIT to the nearest integer.

Note

Zero wait state reads are only supported when MCLK <= 50MHz.

11.1.5.3 Buffer and Cache Mechanisms

Each Flash read interface includes mechanisms designed to maximize performance for read operations. These mechanisms include:

- A program code block cache (MAIN Region Block Cache for both Data and Code Access)
- A line buffer for data read operations (Only for NONMAIN/ENGR Regions, not MAIN region)

These mechanisms are disabled by default at device boot, and must be configured to enable them before application execution commences.

11.1.5.4 Flash Read Arbitration

Each Flash read interface contains multiple levels of arbitration to determine in which order concurrent read requests from multiple initiators are serviced.

1. The first level of arbitration is a fixed priority arbiter between CPU read buses. Data Read Bus 1 (DRB1) always gets a higher priority than Data Read Bus 2 (DRB2). This arbitration level enables zero-wait-state switching between the two data buses belonging to the same CPU. The actual data read bus used for a specific read request is automatically generated by the compiler.
2. The second level of arbitration is the L1 data read pipeline. This level arbitrates between data read requests generated by multiple initiators in the system. The initiators are selected using a round-robin pointer. The round-robin selection order for FRI-1, FRI-3, and FRI-4 is as follows:
   a. CPU1
   b. DMA
   c. Data Pre-read
   d. DEBUGSS

   Switching between initiators incurs a one-cycle pipeline delay, to allow for the round-robin pointer change.
3. For FRI-1, there is a third level of arbitration between fetch and data requests (L2). This arbiter is also a round-robin arbiter like the second level, with selection using the following order:
   a. Instruction fetch (CPU1 for FRI-1)
   b. L1 data read pipeline output

   Changes to the L2 round-robin pointer incur a one-cycle pipeline delay, similar to the L1 arbiter.

11.1.5.5 Error Correction Code (ECC) Protection

The device features an end-to-end safety architecture, with ECC logic. When a Flash data read or instruction fetch operation is performed, ECC check bits are sent along with the data bits onto the CPU program or data buses. The CPU automatically checks incoming instructions and data using the ECC bits, correcting single-bit data errors, and generating a fault for double-bit (uncorrectable) errors. These errors generate a regular interrupt for a single-bit error, or a non-masking interrupt (NMI) for a double-bit error.

Each 8-bit ECC code is computed based on the requested data address and 64 bits of data. For each 128-bit Flash word, there are two ECC codes (8 for the upper 64 bits and 8 for the lower 64 bits), for a total of 16 ECC bits. ECC bits can be independently read through separate FRI read port memory regions as described in the device data sheet.

For more information on ECC protections, see the Section 3.5 chapter.

Note

Read accesses from other initiators such as the debugger or DMA do not have ECC protection. This means flash memory reads from these initiators do not have error detection and correction enabled.

11.1.5.6 Procedure to Change Flash Read Interface Registers

While configuring a Flash Read Interface (FRI), no accesses to any Flash memory that is covered by the Flash Read Interface can be in progress. This includes instructions still in the CPU pipeline, data reads, and instruction prefetch operations. To be sure that no access takes place during the configuration change, follow the procedure shown below for any code that modifies the Flash Read Interface registers.

1. Start executing the application code from RAM or Flash.
2. Branch to or call the Flash configuration code (that writes the FRI registers) in RAM. This is required to properly flush the CPU pipeline before the configuration change. Any function that changes the FRI configuration must reside in RAM, and cannot execute from Flash memory using the same FRI.
3. Execute the Flash configuration code to configure the FRI registers (FRDCNTL, FRIx_INTF_CTRL, and so on).

4. At the end of the Flash configuration code execution, wait nine cycles to allow the write instructions to propagate through the CPU pipeline. This must be done before the return-from-function call is made.
5. Return to the calling function residing in Flash or RAM, and continue execution.

Note

Flash read interfaces that do not include instruction fetch capability, such as FRI-3 and FRI-4, can be configured by code executing directly from Flash, provided that no data access to the covered Flash banks is being made. In such cases, verify that no current data access to the target Flash banks is being performed by placing the FRI configuration code in a separate function. Be sure to wait nine cycles to flush the CPU pipeline before branching back to regular application execution.

11.1.6 Read Interface

The read interface provides the read path to the CPU subsystem (for instruction/data fetch), the read path to the peripheral bus (for use by the DMA controller or CPU), main bank address swapping, and detection and reporting of ECC SEC or DED errors.

11.1.6.1 Bank Address Swapping

Devices that contain more than one bank support swapping of the MAIN regions of the banks within the address space. This mechanism enables two versions of application firmware to be programmed into the device without the firmware needing to know which physical bank it exists in.

Table 11-9 gives an example of the mapping before and after a bank swap is requested for a device with a 512KB main flash split across 2 banks (256KB each).

Table 11-9. Bank Address Swap Translation

Bank and Region	Address Space Before Swap	Address Space After Swap
BANK0 MAIN	0x0000.0000 – 0x0003.FFFF	0x0004.0000 – 0x0007.FFFF
BANK1 MAIN	0x0004.0000 – 0x0007.FFFF	0x0000.0000 – 0x0003.FFFF

After a device reset, the MAIN region of the lower bank is always mapped to the lowest address space. The application software is responsible for determining if a bank address swap is to be applied. The bank address swap control is contained within the SYSCTL module; see the SYSCTL chapter for register and bit definitions. During an address swap, the application software must meet the following constraints:

1. The software must disable interrupts before issuing the bank swap command.
2. The software must poll the bank swap status after issuing the bank swap command before proceeding with execution.
3. If the swap command and poll status routines are executing from flash, they must exist at the exact same location in both banks such that execution resumes from where it left off after the bank swap. This restriction does not apply if the bank swap and status polling is done from SRAM and not from flash memory.

11.1.6.2 ECC Error Handling

The read interface detects and corrects single bit errors in a 64-bit flash word (SEC) and detects dual-bit errors in a 64-bit flash word (DED). ECC checks are ignored for an “all-1’s” or “all-0’s” scenario.

11.1.6.2.1 Single bit (correctable) errors

Single bit errors are corrected automatically before the requested data is returned to either the CPU subsystem or the peripheral bus (DMA). Errors occurring as a result of a debug access are not reported.

11.1.6.2.2 Dual bit (uncorrectable) errors

Dual bit errors are detectable but not correctable and are indicated to SYSCTL. SYSCTL can be configured to generate either a nonmaskable interrupt (NMI) or a reset, as uncorrectable errors can be fatal. Errors occurring as a result of a debug access are not reported.

11.2 FLASH Registers

This Section describes the FLASH Registers.

11.2.1 FLASH Base Address Table

Table 11-10. FLASH Base Address Table

Bit Field Name		DriverLib Name	Base Address
Instance	Structure		
FlashRegs	FLASH_CTRL_REGS	FLASH	0x4002_8000
NvmnwRegs	NVMNW_REGS	NVMNW	0x4004_2000

11.2.2 FLASH_CTRL_REGS Registers

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

Table 11-11. FLASH_CTRL_REGS Registers

Offset	Acronym	Register Name	Write Protection	Section
1000h	FRDCNTL	Flash Read Control Register		Go
100Ch	FRD_INTF_CTRL	Flash Read Interface Control Register		Go
1010h	DTB_MUXSEL	Flash Read Interface DTB Mux select		Go
1100h	ECC_ENABLE	ECC Enable		Go
1104h	FECC_CTRL	ECC Control		Go

Complex bit access types are encoded to fit into small table cells. Table 11-12 shows the codes that are used for access types in this section.

Table 11-12. FLASH_CTRL_REGS Access Type Codes

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

1 FRDCNTL Register (Offset = 1000h) [Reset = 02000201h]

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

Return to the Summary Table.

Flash Read Control Register

Figure 11-2. FRDCNTL Register

31	30	29	28	27	26	25	24
RESERVED				RESERVED			
R-0h				R/W-2h			
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED				RWAIT			
R-0h				R/W-2h			
7	6	5	4	3	2	1	0
RESERVED							WS0_MODE
R-0h							R/W-1h

Table 11-14. FRDCNTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	RESERVED	R	0h	Reserved
27-24	RESERVED	R/W	2h	Reserved
23-12	RESERVED	R	0h	Reserved
11-8	RWAIT	R/W	2h	Random read waitstate These bits indicate how many waitstates are added to a flash read/fetch access. The RWAIT value can be set anywhere from 0 to 0xF. For a flash access, data is returned in RWAIT+1 MCLK cycles. Note: The required wait states for each MCLK frequency can be found in the device data manual. Reset type: SYSRST
7-1	RESERVED	R	0h	Reserved
0	WS0_MODE	R/W	1h	When set, waitstate of the flash will be forced to 0 Waitstate. When this bit is set, RWAIT and TRIMENGRRWAIT values will be ignored. Reset type: SYSRST

ADVANCE INFORMATION

2 FRD_INTF_CTRL Register (Offset = 100Ch) [Reset = 00000000h]

FRD_INTF_CTRL is shown in Figure 11-3 and described in Table 11-14.

Return to the Summary Table.

Flash Read Interface Control Register

Figure 11-3. FRD_INTF_CTRL Register

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

Table 11-16. FRD_INTF_CTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15-3	RESERVED	R	0h	Reserved
2	CODE_CACHE_EN	R/W	0h	Code cache enable. 0 A value of 0 disables the Code cache. 1 A value of 1 enables the Code cache. Reset type: SYSRST
1	DATA_CACHE_EN	R/W	0h	Data cache enable. 0 A value of 0 disables the data cache. 1 A value of 1 enables the data cache. Reset type: SYSRST
0	RESERVED	R/W	0h	Reserved

ADVANCE INFORMATION

3 DTB_MUXSEL Register (Offset = 1010h) [Reset = 00000000h]

DTB_MUXSEL is shown in Figure 11-4 and described in Table 11-15.

Return to the Summary Table.

Flash Read Interface DTB Mux select

Figure 11-4. DTB_MUXSEL Register

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

Table 11-18. DTB_MUXSEL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R	0h	Reserved
1-0	DTB_MUX_SEL_FRI	R/W	0h	DTB Mux Select for Flash read interface signals to be exported out on DTB bus. Based on the MUXSELECT value, corresponding banks' DTB signals will be exported out. 00 - Bank0 01 - Bank1 10 - Bank2 11 - Reserved Reset type: SYSRST

ADVANCE INFORMATION

4 ECC_ENABLE Register (Offset = 1100h) [Reset = 0000000Ah]

ECC_ENABLE is shown in Figure 11-5 and described in Table 11-16.

Return to the Summary Table.

ECC Enable

Figure 11-5. ECC_ENABLE Register

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

Table 11-20. ECC_ENABLE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15-4	RESERVED	R	0h	Reserved
3-0	ENABLE	R/W	Ah	ECC enable. A value of 0xA would enable ECC. Any other value would disable ECC. Reset type: SYSRST

ADVANCE INFORMATION

5 FECC_CTRL Register (Offset = 1104h) [Reset = 00000000h]

FECC_CTRL is shown in Figure 11-6 and described in Table 11-17.

Return to the Summary Table.

ECC Control

Figure 11-6. FECC_CTRL Register

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

Table 11-22. FECC_CTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15-2	RESERVED	R	0h	Reserved
1-0	ECC_TEST_EN	R/W	0h	ECC test mode enable. 00 ECC test mode disabled 01 ECC test mode enabled, one of the 64 data bits is flipped and fed to the redundant ECC logic (on both ECC logic low and ECC logc high blocks). 11 ECC test mode enabled, Two of the 64 data bits are flipped and fed to the redundant ECC logic (on both ECC logic low and ECC logc high blocks). 10 Reserved Reset type: SYSRST

ADVANCE INFORMATION

11.2.3 NVMNW_REGS Registers

Table 11-18 lists the memory-mapped registers for the NVMNW_REGS registers. All register offset addresses not listed in Table 11-18 should be considered as reserved locations and the register contents should not be modified.

Table 11-23. NVMNW_REGS Registers

Offset	Acronym	Register Name	Section
1020h	IIDX	Interrupt Index Register	Go
1028h	IMASK	Interrupt Mask Register	Go
1030h	RIS	Raw Interrupt Status Register	Go
1038h	MIS	Masked Interrupt Status Register	Go
1040h	ISET	Interrupt Set Register	Go
1048h	ICLR	Interrupt Clear Register	Go
1100h	CMDEXEC	Command Execute Register	Go
1104h	CMDTYPE	Command Type Register	Go
1108h	CMDCTL	Command Control Register	Go
1120h	CMDADDR	Command Address Register	Go
1124h	CMDBYTEN	Command Program Byte Enable Register	Go
112Ch	CMDDATAINDEX	Command Data Index Register	Go
1130h	CMDDATA0	Command Data Register 0	Go
1134h	CMDDATA1	Command Data Register 1	Go
1138h	CMDDATA2	Command Data Register 2	Go
113Ch	CMDDATA3	Command Data Register Bits 127:96	Go
1140h	CMDDATA4	Command Data Register 4	Go
1144h	CMDDATA5	Command Data Register 5	Go
1148h	CMDDATA6	Command Data Register 6	Go
114Ch	CMDDATA7	Command Data Register 7	Go
1150h	CMDDATA8	Command Data Register 8	Go
1154h	CMDDATA9	Command Data Register 9	Go
1158h	CMDDATA10	Command Data Register 10	Go
115Ch	CMDDATA11	Command Data Register 11	Go
1160h	CMDDATA12	Command Data Register 12	Go
1164h	CMDDATA13	Command Data Register 13	Go
1168h	CMDDATA14	Command Data Register 14	Go
116Ch	CMDDATA15	Command Data Register 15	Go
11B0h	CMDDATAECC0	Command Data Register ECC 0	Go
11B4h	CMDDATAECC1	Command Data Register ECC 1	Go
11B8h	CMDDATAECC2	Command Data Register ECC 2	Go
11BCh	CMDDATAECC3	Command Data Register ECC 3	Go
11D0h	CMDWEPROTA	Command Write Erase Protect A Register	Go
11D4h	CMDWEPROTB	Command Write Erase Protect B Register	Go
1210h	CMDWEPROTNM	Command Write Erase Protect Non-Main Register	Go
13D0h	STATCMD	Command Status Register	Go
13D8h	STATPCNT	Pulse Count Status Register	Go

Complex bit access types are encoded to fit into small table cells. Table 11-19 shows the codes that are used for access types in this section.

Table 11-24. NVMNW_REGS Access Type Codes

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

1 IIDX Register (Offset = 1020h) [Reset = 00000000h]

IIDX is shown in Figure 11-7 and described in Table 11-20.

Return to the Summary Table.

Interrupt Index Register:
The IIDX register provides the highest priority enabled interrupt index.
PSD compliant register.
Note that it is not recommended to use this register if the system clock is running at a slower clock frequency than the flash wrapper clock. If this is the case, then reading this register may fail to update the RIS register correctly. The MIS register can be read directly, and a write to ICLR can be used to clear interrupts when this clock relationship is present.

Figure 11-7. IIDX Register

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

Table 11-26. IIDX Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	Reserved
0	STAT	R	0h	Indicates which interrupt has fired. 0x0 means no event pending. The priority order is fixed. On each read, only one interrupt is indicated. On a read, the current interrupt (highest priority) is automatically cleared by the hardware and the corresponding interrupt flags in the RIS and MIS are cleared as well. After a read from the CPU (not from the debug interface), the register must be updated with the next highest priority interrupt. 0h (R/W) = No Interrupt Pending 1h (R/W) = DONE Interrupt Pending

2 IMASK Register (Offset = 1028h) [Reset = 00000000h]

IMASK is shown in Figure 11-8 and described in Table 11-21.

Return to the Summary Table.

Interrupt Mask Register: The IMASK register holds the current interrupt mask settings. Masked interrupts are read in the MIS register. PSD compliant register.

Figure 11-8. IMASK Register

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

Table 11-28. IMASK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	Reserved
0	DONE	R/W	0h	Interrupt mask for DONE: 0: Interrupt is disabled in MIS register 1: Interrupt is enabled in MIS register 0h (R/W) = Interrupt is masked out 1h (R/W) = Interrupt will request an interrupt service routine and corresponding bit in [IPSTANDARD.MIS] will be set

3 RIS Register (Offset = 1030h) [Reset = 00000000h]

RIS is shown in Figure 11-9 and described in Table 11-22.

Return to the Summary Table.

Raw Interrupt Status Register: The RIS register reflects all pending interrupts, regardless of masking. The RIS register allows the user to implement a poll scheme. A flag set in this register can be cleared by writing a 1 to the ICLR register bit even if the corresponding IMASK bit is not enabled. A flag can be set by software by writing a 1 to the ISET register. Reading the IIDX register will also clear the corresponding bit in RIS. PSD compliant register.

Figure 11-9. RIS Register

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

Table 11-30. RIS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	Reserved
0	DONE	R	0h	Flash wrapper operation completed. This interrupt bit is set by firmware or the corresponding bit in the ISET register. It is cleared by the corresponding bit in in the ICLR register or reading the IIDX register when this interrupt is the highest priority. 0h (R/W) = Interrupt did not occur 1h (R/W) = Interrupt occurred

ADVANCE INFORMATION

4 MIS Register (Offset = 1038h) [Reset = 00000000h]

MIS is shown in Figure 11-10 and described in Table 11-23.

Return to the Summary Table.

Masked Interrupt Status Register:
The MIS register is a bit-wise AND of the contents of the IMASK and RIS registers. This is kept mainly for ARM compatibility, and has limited use since the highest priority interrupt index is returned through the IIDX register.
PSD
compliant register.

Figure 11-10. MIS Register

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

Table 11-32. MIS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	Reserved
0	DONE	R	0h	Flash wrapper operation completed. This masked interrupt bit reflects the bitwise AND of the corresponding RIS and IMASK bits. 0h (R/W) = Masked interrupt did not occur 1h (R/W) = Masked interrupt occurred

5 ISET Register (Offset = 1040h) [Reset = 00000000h]

ISET is shown in Figure 11-11 and described in Table 11-24.

Return to the Summary Table.

Interrupt Set Register: The ISET register allows software to write a 1 to set corresponding interrupt. Safety: This meets a safety requirement to allow software diagnostics to trigger interrupts. PSD compliant register.

Figure 11-11. ISET Register

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

Table 11-34. ISET Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	W	0h	Reserved
0	DONE	W	0h	0: No effect 1: Set the DONE interrupt in the RIS register 0h (R/W) = Writing a 0 has no effect 1h (R/W) = Set [IPSTANDARD.RIS] bit

6 ICLR Register (Offset = 1048h) [Reset = 00000000h]

ICLR is shown in Figure 11-12 and described in Table 11-25.

Return to the Summary Table.

Interrupt Clear Register. The ICLR register allows allows software to write a 1 to clear corresponding interrupt. PSD compliant register.

Figure 11-12. ICLR Register

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

Table 11-36. ICLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	W	0h	Reserved
0	DONE	W	0h	0: No effect 1: Clear the DONE interrupt in the RIS register 0h (R/W) = Writing a 0 has no effect 1h (R/W) = Clear [IPSTANDARD.RIS] bit

ADVANCE INFORMATION

7 CMDEXEC Register (Offset = 1100h) [Reset = 00000000h]

CMDEXEC is shown in Figure 11-13 and described in Table 11-26.

Return to the Summary Table.

Command Execute Register:
Initiates execution of the command specified in the CMDTYPE register.
This register is blocked for writes after being written to 1 and prior to
STATCMD.DONE being set by the flash wrapper hardware.
flash wrapper hardware clears this register after the processing of the command
has completed.

Figure 11-13. CMDEXEC Register

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

Table 11-38. CMDEXEC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	Reserved
0	VAL	R/W	0h	Command Execute value Initiates execution of the command specified in the CMDTYPE register. 0h (R/W) = Command will not execute or is not executing in flash wrapper 1h (R/W) = Command will execute or is executing in flash wrapper

8 CMDTYPE Register (Offset = 1104h) [Reset = 00000000h]

CMDTYPE is shown in Figure 11-14 and described in Table 11-27.

Return to the Summary Table.

Command Type Register
This register specifies the type of command to be executed by the flash wrapper hardware.
This register is blocked for writes after CMDEXEC is written to a 1 and prior to STATCMD.DONE being set by the hardware to indicate that command execution has completed.

Figure 11-14. CMDTYPE Register

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

Table 11-40. CMDTYPE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-7	RESERVED	R	0h	Reserved
6-4	SIZE	R/W	0h	Command size 0h (R/W) = Operate on 1 flash word 1h (R/W) = Operate on 2 flash words 2h (R/W) = Operate on 4 flash words 3h (R/W) = Operate on 8 flash words 4h (R/W) = Operate on a flash sector 5h (R/W) = Operate on an entire flash bank
3	RESERVED	R	0h	Reserved
2-0	COMMAND	R/W	0h	Command type 0h (R/W) = No Operation 1h (R/W) = Program 2h (R/W) = Erase 3h (R/W) = ReadVerify - Perform a standalone ReadVerify operation. 4h (R/W) = Mode Change - Perform a mode change only, no other operation. 5h (R/W) = Clear Status - Clear status bits in FW_SMSTAT only. 6h (R/W) = Blank Verify - Check whether a flash word is in the erased state. This command may only be used with CMDTYPE.SIZE = ONEWORD

9 CMDCTL Register (Offset = 1108h) [Reset = 00000000h]

CMDCTL is shown in Figure 11-15 and described in Table 11-28.

Return to the Summary Table.

Command Control Register This register configures specific capabilities of the state machine for related to the execution of a command. This register is blocked for writes after CMDEXEC is written to a 1 and prior to STATCMD.DONE being set by the hardware to indicate that command execution has completed.

Figure 11-15. CMDCTL Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED		DATAVEREN	SSERASEDIS	ERASEMASKDIS	PROGMASKDIS	ECCGENOVR	ADDRXLATEOVR
R-0h		R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
POSTVEREN	PREVEREN	RESERVED	REGIONSEL				RESERVED
R/W-0h	R/W-0h	R-0h	R/W-0h				R/W-0h
7	6	5	4	3	2	1	0
RESERVED			BANKSEL	MODESEL			
R/W-0h			R/W-0h	R/W-0h			

Table 11-42. CMDCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-22	RESERVED	R	0h	Reserved
21	DATAVEREN	R/W	0h	Enable invalid data verify. This checks for 0->1 transitions in the memory when a program operation is initiated. If such a transition is found, the program will fail with an error without executing the program. 0h (R/W) = Disable 1h (R/W) = Enable
20	SSERASEDIS	R/W	0h	Disable Stair-Step Erase. If set, the default VHV trim voltage setting will be used for all erase pulses. By default, this bit is reset, meaning that the VHV voltage will be stepped during successive erase pulses. The step count, step voltage, begin and end voltages are all hard-wired. 0h (R/W) = Enable 1h (R/W) = Disable
19	ERASEMASKDIS	R/W	0h	Disable use of erase mask for erase Bit masking will not be used during erase verify. If one or more sectors fail the verify either before (prever) or after (postver) the operation, then all specified flash sectors will receive subsequent erase pulse. 0h (R/W) = Enable 1h (R/W) = Disable
18	PROGMASKDIS	R/W	0h	Disable use of program mask for programming. Bit masking will not be used during program verify. If one or more bits fail the verify either before (prever) or after (postver) the operation, then all specified flash entries will receive subsequent program pulse. 0h (R/W) = Enable 1h (R/W) = Disable

ADVANCE INFORMATION

Table 11-42. CMDCTL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
17	ECCGENOVR	R/W	0h	Override hardware generation of ECC data for program. Use data written to CMDDATAECC*. 0h (R/W) = Do not override 1h (R/W) = Override
16	ADDRXLATEOVR	R/W	0h	Override hardware address translation of address in CMDADDR from a system address to a bank address and bank ID. Use data written to CMDADDR directly as the bank address. Use the value written to CMDCTL.BANKSEL directly as the bank ID. Use the value written to CMDCTL.REGIONSEL directly as the region ID. 0h (R/W) = Do not override 1h (R/W) = Override
15	POSTVEREN	R/W	1h	Enable verify after program or erase 0h (R/W) = Disable 1h (R/W) = Enable
14	PREVEREN	R/W	1h	Enable verify before program or erase. For program, bits already programmed to the requested value will be masked. For erase, sectors already erased will be masked. 0h (R/W) = Disable 1h (R/W) = Enable
13	RESERVED	R	0h	Reserved
12-9	REGIONSEL	R/W	0h	Bank Region A specific region ID can be written to this field to indicate to which region an operation is to be applied if CMDCTL.ADDRXLATEOVR is set. 1h (R/W) = Main Region 2h (R/W) = Non-Main Region 4h (R/W) = Trim Region 8h (R/W) = Engr Region
8-5	RESERVED	R/W	0h	
4	BANKSEL	R/W	0h	Bank Select A specific Bank ID can be written to this field to indicate to which bank an operation is to be applied if CMDCTL.ADDRXLATEOVR is set. 1h (R/W) = Bank 0 2h (R/W) = Bank 1 4h (R/W) = Bank 2 8h (R/W) = Bank 3 10h (R/W) = Bank 4
3-0	MODESEL	R/W	0h	Mode This field is only used for the Mode Change command type. Otherwise, bank and pump modes are set automaticlly through the NW hardware. 0h = Read Mode 2h = Read Margin 0 Mode 4h = Read Margin 1 Mode 6h = Read Margin 0B Mode 7h = Read Margin 1B Mode 9h = Program Verify Mode Ah = Program Single Word Bh = Erase Verify Mode Ch = Erase Sector Eh = Program Multiple Word Fh = Erase Bank

10 CMDADDR Register (Offset = 1120h) [Reset = 00000000h]

CMDADDR is shown in Figure 11-16 and described in Table 11-29.

Return to the Summary Table.

Command Address Register:
This register forms the target address of a command. The use cases are as follows:
1) For single-word program, this address indicates the flash bank word to be programmed.
2) For multi-word program, this address indicates the first flash bank address for the program. The address will be incremented for further words.
3) For sector erase, this address indicates the sector to be erased.
4) For bank erase, the address indicates the bank to be erased.
Note the address written to this register will be submitted for translation to the flash wrapper address translation interface, and the translated address will be used to access the bank. However, if the CMDCTL.ADDRXLATEOVR bit is set, then the address written to this register will be used directly as the bank address.
This register is blocked for writes after a 1 is written to the CMDEXEC register and prior to STATCMD.DONE being set by the flash wrapper hardware.

Figure 11-16. CMDADDR Register

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

Table 11-44. CMDADDR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	VAL	R/W	0h	Address value 0h = Minimum value of [VAL] FFFFFFFFh = Maximum value of [VAL]

11 CMDBYTEN Register (Offset = 1124h) [Reset = 00000000h]

CMDBYTEN is shown in Figure 11-17 and described in Table 11-30.

Return to the Summary Table.

Command Program Byte Enable Register:
This register forms a per-byte enable for programming data. For data bytes to be programmed, a 1 must be written to the corresponding bit in this register. Normally, all bits are written to 1, allowing program of full flash words. However, leaving some bits 0 allows programming of 8-bit, 16-bit, 32-bit or 64-bit portions of a flash word.
During verify, data bytes read from the flash will not be checked if the corresponding CMDBYTEN bit is 0.
ECC data bytes are protected by the 1-2 MSB bits in this register, depending on the presence of ECC and the flash word data width.
This register is blocked for writes after a 1 is written to the CMDEXEC register and prior to STATCMD.DONE being set by the flash wrapper hardware.
This register is written to all 0 after the completion of all flash wrapper commands.

Figure 11-17. CMDBYTEN Register

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

Table 11-46. CMDBYTEN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	RESERVED	R	0h	Reserved
17-8	RESERVED	R/W	0h	
7-0	VAL	R/W	0h	Command Byte Enable value. A 1-bit per flash word byte value is placed in this register. 0h = Minimum value of [VAL] 0003FFFFh = Maximum value of [VAL]

12 CMDDATAINDEX Register (Offset = 112Ch) [Reset = 00000000h]

CMDDATAINDEX is shown in Figure 11-18 and described in Table 11-31.

Return to the Summary Table.

Command Program Data Index Register:
When multiple data registers are available for multi-word program, this register can be written with an index which points to one of the data registers. When a write to CMDDATA* is done, the data will be written to the physical data register indexed by the value in this register.
Up to 8 data registers can be present, so this register can be written with 0x0 to 0x7. If less than 8 data registers are present, successive MSB bits of this register are ignored when indexing the CMDDATA* registers.
This register is blocked for writes after a 1 is written to the CMDEXEC register and prior to STATCMD.DONE being set by the flash wrapper hardware.

Figure 11-18. CMDDATAINDEX Register

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

Table 11-48. CMDDATAINDEX Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R	0h	Reserved
2-0	VAL	R/W	0h	Data register index 0h = Minimum value of [VAL] 7h = Maximum value of [VAL]

ADVANCE INFORMATION

13 CMDDATA0 Register (Offset = 1130h) [Reset = 00000000h]

CMDDATA0 is shown in Figure 11-19 and described in Table 11-32.

Return to the Summary Table.

Command Data Register 0
This register forms the data for a command.
For DATAWIDTH == 128: This register represents bits 31:0 of flash word data register 0.
For DATAWIDTH == 64: This register represents bits 31:0 of flash word data register 0.
This register is blocked for writes after a 1 is written to the CMDEXEC register and prior to STATCMD.DONE being set by the flash wrapper hardware.
This register is used to aggregate masking for bits that do not require additional program pulses during program operations, and will be written to all 1 after the completion of all flash wrapper commands.
Use cases for the CMDDATA* registers are as follows:
1) Program - These registers contain the data to be programmed.
2) Erase - These registers are not used.

Figure 11-19. CMDDATA0 Register

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

Table 11-50. CMDDATA0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	VAL	R/W	FFFFFFFFh	A 32-bit data value is placed in this field. 0h = Minimum value of [VAL] FFFFFFFFh = Maximum value of [VAL]

14 CMDDATA1 Register (Offset = 1134h) [Reset = 00000000h]

CMDDATA1 is shown in Figure 11-20 and described in Table 11-33.

Return to the Summary Table.

Command Data Register 1
This register forms the data for a command.
For DATAWIDTH == 128: This register represents bits 63:32 of flash word data register 0.
For DATAWIDTH == 64: This register represents bits 63:32 of flash word data register 0.
This register is blocked for writes after a 1 is written to the CMDEXEC
register and prior to CMDSTAT.DONE being set by the flash wrapper
hardware.
This register is used to aggregate masking for bits that do not
require additional program pulses during program operations, and will be
written to all 1 after the completion of all flash wrapper commands.
Use cases for the CMDDATA* registers are as follows:
1) Program - These registers contain the data to be programmed.
2) Erase - These registers are not used.

Figure 11-20. CMDDATA1 Register

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

Table 11-52. CMDDATA1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	VAL	R/W	FFFFFFFFh	A 32-bit data value is placed in this field. 0h = Minimum value of [VAL] FFFFFFFFh = Maximum value of [VAL]

15 CMDDATA2 Register (Offset = 1138h) [Reset = 00000000h]

CMDDATA2 is shown in Figure 11-21 and described in Table 11-34.

Return to the Summary Table.

Command Data Register 2
This register forms the data for a command.
For DATAWIDTH == 128: This register represents bits 95:64 of flash word data register 0.
For DATAWIDTH == 64: This register represents bits 31:0 of flash word data register 1.
This register is blocked for writes after a 1 is written to the CMDEXEC register and prior to STATCMD.DONE being set by the flash wrapper hardware.
This register is used to aggregate masking for bits that do not require additional program pulses during program operations, and will be written to all 1 after the completion of all flash wrapper commands.
Use cases for the CMDDATA* registers are as follows:
1) Program - These registers contain the data to be programmed.
2) Erase - These registers are not used.

Figure 11-21. CMDDATA2 Register

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

Table 11-54. CMDDATA2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	VAL	R/W	FFFFFFFFh	A 32-bit data value is placed in this field. 0h = Minimum value of [VAL] FFFFFFFFh = Maximum value of [VAL]

16 CMDDATA3 Register (Offset = 113Ch) [Reset = 00000000h]

CMDDATA3 is shown in Figure 11-22 and described in Table 11-35.

Return to the Summary Table.

Command Data Register 3
This register forms the data for a command.
For DATAWIDTH == 128: This register represents bits 127:96 of flash word data register 0.
For DATAWIDTH == 64: This register represents bits 63:32 of flash word data register 1.
This register is blocked for writes after a 1 is written to the CMDEXEC
register and prior to STATCMD.DONE being set by the flash wrapper
hardware.
This register is used to aggregate masking for bits that do not
require additional program pulses during program operations, and will be
written to all 1 after the completion of all flash wrapper commands.
Use cases for the CMDDATA* registers are as follows:
1) Program - These registers contain the data to be programmed.
2) Erase - These registers are not used.

Figure 11-22. CMDDATA3 Register

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

Table 11-56. CMDDATA3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	VAL	R/W	FFFFFFFFh	A 32-bit data value is placed in this field. 0h = Minimum value of [VAL] FFFFFFFFh = Maximum value of [VAL]

17 CMDDATA4 Register (Offset = 1140h) [Reset = 00000000h]

CMDDATA4 is shown in Figure 11-23 and described in Table 11-36.

Return to the Summary Table.

Command Data Register 4
This register forms the data for a command.
For DATAWIDTH == 128: This register represents bits 31:0 of flash word data register 1.
For DATAWIDTH == 64: This register represents bits 31:0 of flash word data register 2.
This register is blocked for writes after a 1 is written to the CMDEXEC register and prior to STATCMD.DONE being set by the flash wrapper hardware.
This register is used to aggregate masking for bits that do not require additional program pulses during program operations, and will be written to all 1 after the completion of all flash wrapper commands.
Use cases for the CMDDATA* registers are as follows:
1) Program - These registers contain the data to be programmed.
2) Erase - These registers are not used.

Figure 11-23. CMDDATA4 Register

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

Table 11-58. CMDDATA4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	VAL	R/W	FFFFFFFFh	A 32-bit data value is placed in this field. T 0h = Minimum value of [VAL] FFFFFFFFh = Maximum value of [VAL]

18 CMDDATA5 Register (Offset = 1144h) [Reset = 00000000h]

CMDDATA5 is shown in Figure 11-24 and described in Table 11-37.

Return to the Summary Table.

Command Data Register 5
This register forms the data for a command.
For DATAWIDTH == 128: This register represents bits 63:32 of flash word data register 1.
For DATAWIDTH == 64: This register represents bits 63:32 of flash word data register 2.
This register is blocked for writes after a 1 is written to the CMDEXEC
register and prior to STATCMD.DONE being set by the flash wrapper
hardware.
This register is used to aggregate masking for bits that do not
require additional program pulses during program operations, and will be
written to all 1 after the completion of all flash wrapper commands.
Use cases for the CMDDATA* registers are as follows:
1) Program - These registers contain the data to be programmed.
2) Erase - These registers are not used.

Figure 11-24. CMDDATA5 Register

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

Table 11-60. CMDDATA5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	VAL	R/W	FFFFFFFFh	A 32-bit data value is placed in this field. 0h = Minimum value of [VAL] FFFFFFFFh = Maximum value of [VAL]

19 CMDDATA6 Register (Offset = 1148h) [Reset = 00000000h]

CMDDATA6 is shown in Figure 11-25 and described in Table 11-38.

Return to the Summary Table.

Command Data Register 6
This register forms the data for a command.
For DATAWIDTH == 128: This register represents bits 95:64 of flash word data register 1.
For DATAWIDTH == 64: This register represents bits 31:0 of flash word data register 3.
This register is blocked for writes after a 1 is written to the CMDEXEC
register and prior to STATCMD.DONE being set by the flash wrapper
hardware.
This register is used to aggregate masking for bits that do not
require additional program pulses during program operations, and will be
written to all 1 after the completion of all flash wrapper commands.
Use cases for the CMDDATA* registers are as follows:
1) Program - These registers contain the data to be programmed.
2) Erase - These registers are not used.

Figure 11-25. CMDDATA6 Register

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

Table 11-62. CMDDATA6 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	VAL	R/W	FFFFFFFFh	A 32-bit data value is placed in this field. 0h = Minimum value of [VAL] FFFFFFFFh = Maximum value of [VAL]

20 CMDDATA7 Register (Offset = 114Ch) [Reset = 00000000h]

CMDDATA7 is shown in Figure 11-26 and described in Table 11-39.

Return to the Summary Table.

Command Data Register 7
This register forms the data for a command.
For DATAWIDTH == 128: This register represents bits 127:96 of flash word data register 1.
For DATAWIDTH == 64: This register represents bits 63:32 of flash word data register 3.
This register is blocked for writes after a 1 is written to the CMDEXEC register and prior to STATCMD.DONE being set by the flash wrapper hardware.
This register is used to aggregate masking for bits that do not require additional program pulses during program operations, and will be written to all 1 after the completion of all flash wrapper commands.
Use cases for the CMDDATA* registers are as follows:
1) Program - These registers contain the data to be programmed.
2) Erase - These registers are not used.

Figure 11-26. CMDDATA7 Register

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

Table 11-64. CMDDATA7 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	VAL	R/W	FFFFFFFFh	A 32-bit data value is placed in this field. 0h = Minimum value of [VAL] FFFFFFFFh = Maximum value of [VAL]

21 CMDDATA8 Register (Offset = 1150h) [Reset = 00000000h]

CMDDATA8 is shown in Figure 11-27 and described in Table 11-40.

Return to the Summary Table.

Command Data Register 8
This register forms the data for a command.
For DATAWIDTH == 128: This register represents bits 31:0 of flash word data register 2.
For DATAWIDTH == 64: This register represents bits 31:0 of flash word data register 4.
This register is blocked for writes after a 1 is written to the CMDEXEC
register and prior to STATCMD.DONE being set by the flash wrapper
hardware.
This register is used to aggregate masking for bits that do not
require additional program pulses during program operations, and will be
written to all 1 after the completion of all flash wrapper commands.
Use cases for the CMDDATA* registers are as follows:
1) Program - These registers contain the data to be programmed.
2) Erase - These registers are not used.

Figure 11-27. CMDDATA8 Register

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

Table 11-66. CMDDATA8 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	VAL	R/W	FFFFFFFFh	A 32-bit data value is placed in this field. 0h = Minimum value of [VAL] FFFFFFFFh = Maximum value of [VAL]

22 CMDDATA9 Register (Offset = 1154h) [Reset = 00000000h]

CMDDATA9 is shown in Figure 11-28 and described in Table 11-41.

Return to the Summary Table.

Command Data Register 9
This register forms the data for a command.
For DATAWIDTH == 128: This register represents bits 63:32 of flash word data register 2.
For DATAWIDTH == 64: This register represents bits 63:32 of flash word data register 4.
This register is blocked for writes after a 1 is written to the CMDEXEC
register and prior to STATCMD.DONE being set by the flash wrapper
hardware.
This register is used to aggregate masking for bits that do not
require additional program pulses during program operations, and will be
written to all 1 after the completion of all flash wrapper commands.
Use cases for the CMDDATA* registers are as follows:
1) Program - These registers contain the data to be programmed.
2) Erase - These registers are not used.

Figure 11-28. CMDDATA9 Register

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

Table 11-68. CMDDATA9 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	VAL	R/W	FFFFFFFFh	A 32-bit data value is placed in this field. 0h = Minimum value of [VAL] FFFFFFFFh = Maximum value of [VAL]

23 CMDDATA10 Register (Offset = 1158h) [Reset = 00000000h]

CMDDATA10 is shown in Figure 11-29 and described in Table 11-42.

Return to the Summary Table.

Command Data Register 10
This register forms the data for a command.
For DATAWIDTH == 128: This register represents bits 95:64 of flash word data register 2.
For DATAWIDTH == 64: This register represents bits 31:0 of flash word data register 5.
This register is blocked for writes after a 1 is written to the CMDEXEC
register and prior to STATCMD.DONE being set by the flash wrapper
hardware.
This register is used to aggregate masking for bits that do not
require additional program pulses during program operations, and will be
written to all 1 after the completion of all flash wrapper commands.
Use cases for the CMDDATA* registers are as follows:
1) Program - These registers contain the data to be programmed.
2) Erase - These registers are not used.

Figure 11-29. CMDDATA10 Register

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

Table 11-70. CMDDATA10 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	VAL	R/W	FFFFFFFFh	A 32-bit data value is placed in this field. 0h = Minimum value of [VAL] FFFFFFFFh = Maximum value of [VAL]

24 CMDDATA11 Register (Offset = 115Ch) [Reset = 00000000h]

CMDDATA11 is shown in Figure 11-30 and described in Table 11-43.

Return to the Summary Table.

Command Data Register 11
This register forms the data for a command.
For DATAWIDTH == 128: This register represents bits 127:96 of flash word data register 2.
For DATAWIDTH == 64: This register represents bits 63:32 of flash word data register 5.
This register is blocked for writes after a 1 is written to the CMDEXEC
register and prior to STATCMD.DONE being set by the flash wrapper
hardware.
This register is used to aggregate masking for bits that do not
require additional program pulses during program operations, and will be
written to all 1 after the completion of all flash wrapper commands.
Use cases for the CMDDATA* registers are as follows:
1) Program - These registers contain the data to be programmed.
2) Erase - These registers are not used.

Figure 11-30. CMDDATA11 Register

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

Table 11-72. CMDDATA11 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	VAL	R/W	FFFFFFFFh	A 32-bit data value is placed in this field. 0h = Minimum value of [VAL] FFFFFFFFh = Maximum value of [VAL]

25 CMDDATA12 Register (Offset = 1160h) [Reset = 00000000h]

CMDDATA12 is shown in Figure 11-31 and described in Table 11-44.

Return to the Summary Table.

Command Data Register 12
This register forms the data for a command.
For DATAWIDTH == 128: This register represents bits 31:0 of flash word data register 3.
For DATAWIDTH == 64: This register represents bits 31:0 of flash word data register 6.
This register is blocked for writes after a 1 is written to the CMDEXEC register and prior to STATCMD.DONE being set by the flash wrapper hardware.
This register is used to aggregate masking for bits that do not require additional program pulses during program operations, and will be written to all 1 after the completion of all flash wrapper commands.
Use cases for the CMDDATA* registers are as follows:
1) Program - These registers contain the data to be programmed.
2) Erase - These registers are not used.

Figure 11-31. CMDDATA12 Register

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

Table 11-74. CMDDATA12 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	VAL	R/W	FFFFFFFFh	A 32-bit data value is placed in this field. 0h = Minimum value of [VAL] FFFFFFFFh = Maximum value of [VAL]

26 CMDDATA13 Register (Offset = 1164h) [Reset = 00000000h]

CMDDATA13 is shown in Figure 11-32 and described in Table 11-45.

Return to the Summary Table.

Command Data Register 13
This register forms the data for a command.
For DATAWIDTH == 128: This register represents bits 63:32 of flash word data register 3.
For DATAWIDTH == 64: This register represents bits 63:32 of flash word data register 6.
This register is blocked for writes after a 1 is written to the CMDEXEC
register and prior to STATCMD.DONE being set by the flash wrapper
hardware.
This register is used to aggregate masking for bits that do not
require additional program pulses during program operations, and will be
written to all 1 after the completion of all flash wrapper commands.
Use cases for the CMDDATA* registers are as follows:
1) Program - These registers contain the data to be programmed.
2) Erase - These registers are not used.

Figure 11-32. CMDDATA13 Register

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

Table 11-76. CMDDATA13 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	VAL	R/W	FFFFFFFFh	A 32-bit data value is placed in this field. 0h = Minimum value of [VAL] FFFFFFFFh = Maximum value of [VAL]

27 CMDDATA14 Register (Offset = 1168h) [Reset = 00000000h]

CMDDATA14 is shown in Figure 11-33 and described in Table 11-46.

Return to the Summary Table.

Command Data Register 14
This register forms the data for a command.
For DATAWIDTH == 128: This register represents bits 95:64 of flash word data register 3.
For DATAWIDTH == 64: This register represents bits 31:0 of flash word data register 7.
This register is blocked for writes after a 1 is written to the CMDEXEC
register and prior to STATCMD.DONE being set by the flash wrapper
hardware.
This register is used to aggregate masking for bits that do not
require additional program pulses during program operations, and will be
written to all 1 after the completion of all flash wrapper commands.
Use cases for the CMDDATA* registers are as follows:
1) Program - These registers contain the data to be programmed.
2) Erase - These registers are not used.

Figure 11-33. CMDDATA14 Register

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

Table 11-78. CMDDATA14 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	VAL	R/W	FFFFFFFFh	A 32-bit data value is placed in this field. 0h = Minimum value of [VAL] FFFFFFFFh = Maximum value of [VAL]

28 CMDDATA15 Register (Offset = 116Ch) [Reset = 00000000h]

CMDDATA15 is shown in Figure 11-34 and described in Table 11-47.

Return to the Summary Table.

Command Data Register 15
This register forms the data for a command.
For DATAWIDTH == 128: This register represents bits 127:96 of flash word data register 3.
For DATAWIDTH == 64: This register represents bits 63:32 of flash word data register 7.
This register is blocked for writes after a 1 is written to the CMDEXEC
register and prior to STATCMD.DONE being set by the flash wrapper
hardware.
This register is used to aggregate masking for bits that do not
require additional program pulses during program operations, and will be
written to all 1 after the completion of all flash wrapper commands.
Use cases for the CMDDATA* registers are as follows:
1) Program - These registers contain the data to be programmed.
2) Erase - These registers are not used.

Figure 11-34. CMDDATA15 Register

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

Table 11-80. CMDDATA15 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	VAL	R/W	FFFFFFFFh	A 32-bit data value is placed in this field. 0h = Minimum value of [VAL] FFFFFFFFh = Maximum value of [VAL]

29 CMDDATAECC0 Register (Offset = 11B0h) [Reset = 00000000h]

CMDDATAECC0 is shown in Figure 11-35 and described in Table 11-48.

Return to the Summary Table.

Command Data Register 0
This register forms the ECC portion of the data for a command. This ECC data in this register covers flash data register 0.
The hardware ECC generation in flash wrapper can be overridden and ECC data developed elsewhere can be used. ECC data is placed in this register.
This register is blocked for writes after a 1 is written to the CMDEXEC register and prior to STATCMD.DONE being set by the flash wrapper hardware.
Use cases for the CMDDATA* registers are as follows:
1) Program - These registers contain the data to be programmed.
2) Erase - These registers are not used.
This register is used to aggregate masking for bits that do not require additional program pulses during program operations, and will be written to all 1 after the completion of all flash wrapper commands.

Figure 11-35. CMDDATAECC0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																VAL1								VAL0							
R-0h																R/W-0h								R/W-0h							

Table 11-82. CMDDATAECC0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15-8	VAL1	R/W	FFh	ECC data for bits 127:64 of the data is placed here. 0h = Minimum value FFh = Maximum value
7-0	VAL0	R/W	FFh	ECC data for bits 63:0 of the data is placed here. 0h = Minimum value FFh = Maximum value

30 CMDDATAECC1 Register (Offset = 11B4h) [Reset = 00000000h]

CMDDATAECC1 is shown in Figure 11-36 and described in Table 11-49.

Return to the Summary Table.

Command Data Register 1
This register forms the ECC portion of the data for a command. This ECC data in this register covers flash data register 0.
The hardware ECC generation in flash wrapper can be overridden and ECC data developed elsewhere can be used. ECC data is placed in this register.
This register is blocked for writes after a 1 is written to the CMDEXEC register and prior to STATCMD.DONE being set by the flash wrapper hardware.
Use cases for the CMDDATA* registers are as follows:
1) Program - These registers contain the data to be programmed.
2) Erase - These registers are not used.
This register is used to aggregate masking for bits that do not require additional program pulses during program operations, and will be written to all 1 after the completion of all flash wrapper commands.

Figure 11-36. CMDDATAECC1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																VAL1								VAL0							
R-0h																R/W-0h								R/W-0h							

Table 11-84. CMDDATAECC1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15-8	VAL1	R/W	FFh	ECC data for bits 127:64 of the data is placed here. 0h = Minimum value FFh = Maximum value
7-0	VAL0	R/W	FFh	ECC data for bits 63:0 of the data is placed here. 0h = Minimum value FFh = Maximum value

31 CMDDATAECC2 Register (Offset = 11B8h) [Reset = 00000000h]

CMDDATAECC2 is shown in Figure 11-37 and described in Table 11-50.

Return to the Summary Table.

Command Data Register 2
This register forms the ECC portion of the data for a command. This ECC data in this register covers flash data register 2.
The hardware ECC generation in flash wrapper can be overridden and ECC data developed elsewhere can be used. ECC data is placed in this register.
This register is blocked for writes after a 1 is written to the CMDEXEC register and prior to STATCMD.DONE being set by the flash wrapper hardware.
Use cases for the CMDDATA* registers are as follows:
1) Program - These registers contain the data to be programmed.
2) Erase - These registers are not used.
This register is used to aggregate masking for bits that do not require additional program pulses during program operations, and will be written to all 1 after the completion of all flash wrapper commands.

Figure 11-37. CMDDATAECC2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																VAL1								VAL0							
R-0h																R/W-0h								R/W-0h							

Table 11-86. CMDDATAECC2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15-8	VAL1	R/W	FFh	ECC data for bits 127:64 of the data is placed here. 0h = Minimum value FFh = Maximum value
7-0	VAL0	R/W	FFh	ECC data for bits 63:0 of the data is placed here. 0h = Minimum value FFh = Maximum value

32 CMDDATAECC3 Register (Offset = 11BCh) [Reset = 00000000h]

CMDDATAECC3 is shown in Figure 11-38 and described in Table 11-51.

Return to the Summary Table.

Command Data Register 3
This register forms the ECC portion of the data for a command. This ECC data in this register covers flash data register 3.
The hardware ECC generation in flash wrapper can be overridden and ECC data developed elsewhere can be used. ECC data is placed in this register.
This register is blocked for writes after a 1 is written to the CMDEXEC register and prior to STATCMD.DONE being set by the flash wrapper hardware.
Use cases for the CMDDATA* registers are as follows:
1) Program - These registers contain the data to be programmed.
2) Erase - These registers are not used.
This register is used to aggregate masking for bits that do not require additional program pulses during program operations, and will be written to all 1 after the completion of all flash wrapper commands.

Figure 11-38. CMDDATAECC3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																VAL1								VAL0							
R-0h																R/W-0h								R/W-0h							

Table 11-88. CMDDATAECC3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15-8	VAL1	R/W	FFh	ECC data for bits 127:64 of the data is placed here. 0h = Minimum value FFh = Maximum value
7-0	VAL0	R/W	FFh	ECC data for bits 63:0 of the data is placed here. 0h = Minimum value FFh = Maximum value

33 CMDWEPROTA Register (Offset = 11D0h) [Reset = 00000000h]

CMDWEPROTA is shown in Figure 11-39 and described in Table 11-52.

Return to the Summary Table.

Command WriteErase Protect A Register
This register allows the first 32 sectors of the main region to be protected from program or erase, with 1 bit protecting each sector. If the main region size is smaller than 32 sectors, then this register provides protection for the whole region.
This register is blocked for writes after a 1 is written to the CMDEXEC register and prior to STATCMD.DONE being set by the flash wrapper hardware.
In addition, this register is used to aggregate masking for sectors that do not require additional erase pulses during bank erase operations, and will be written to all 1 after the completion of all flash wrapper commands.

Figure 11-39. CMDWEPROTA Register

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

Table 11-90. CMDWEPROTA Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	VAL	R/W	FFFFFFFFh	Each bit protects 1 sector. bit [0]: When 1, sector 0 of the flash memory will be protected from program and erase. bit [1]: When 1, sector 1 of the flash memory will be protected from program and erase. : : bit [31]: When 1, sector 31 of the flash memory will be protected from program and erase. 0h = Minimum value of [VAL] FFFFFFFFh = Maximum value of [VAL]

ADVANCE INFORMATION

34 CMDWEPROTB Register (Offset = 11D4h) [Reset = 00000000h]

CMDWEPROTB is shown in Figure 11-40 and described in Table 11-53.

Return to the Summary Table.

Command WriteErase Protect B Register
This register allows main region sectors to be protected from program and erase. Each bit corresponds to a group of 8 sectors.
There are multiple cases for how these protect bits are applied:
1. Single-bank system, CMDWEPROTA register present:
The first 32 sectors are protected via the CMDWEPROTA register. Thus, the protection given by the bits in CMDWEPROTB begin with sector 32.
2. Single-bank system, CMDWEPROTA register not present:
The protection given by the bits in CMDWEPROTB begin with sector 0.
3. Multi-bank system, CMDWEPROTA register present - Bank 0:
The first 32 sectors of bank 0 are protected via the CMDWEPROTA register. Thus, only bits 4 and above of CMDWEPROTB would be applicable to bank 0. The protection of bit 4 and above would begin at sector 32. Bits 3:0 of WEPROTB are ignored for bank 0.
4. Multi-bank system, CMDWEPROTA register present, Banks 1-N:
For banks other than bank 0 in a multi-bank system, CMDWEPROTA has no effect, so the bits in CMDWEPROTB will protect these banks starting from sector 0.
5. Multi-bank system, CMDWEPROTA register not present:
The bits in CMDWEPROTB will protect any of the banks starting from sector 0.
This register is blocked for writes after a 1 is written to the CMDEXEC register and prior to STATCMD.DONE being set by the flash wrapper hardware.
In addition, this register is used to aggregate masking for sectors that do not require additional erase pulses during bank erase operations, and will be written to all 1 after the completion of all flash wrapper commands.

Figure 11-40. CMDWEPROTB Register

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

Table 11-92. CMDWEPROTB Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	VAL	R/W	FFFFFFFFh	Each bit protects a group of 8 sectors. When a bit is 1, the associated 8 sectors in the flash will be protected from program and erase. A maximum of 256 sectors can be protected with this register. 0h = Minimum value of [VAL] FFFFFFFFh = Maximum value of [VAL]

35 CMDWEPROTNM Register (Offset = 1210h) [Reset = 00000000h]

CMDWEPROTNM is shown in Figure 11-41 and described in Table 11-54.

Return to the Summary Table.

Command WriteErase Protect Non-Main
Register
This register allows non-main region region sectors to be protected from program and erase. Each bit corresponds to 1 sector.
This register is blocked for writes after a 1 is written to the CMDEXEC register and prior to STATCMD.DONE being set by the flash wrapper hardware.
In addition, this register is used to aggregate masking for sectors that do not require additional erase pulses during bank erase operations, and will be written to all 1 after the completion of all flash wrapper commands.

Figure 11-41. CMDWEPROTNM Register

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

Table 11-94. CMDWEPROTNM Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	VAL	R/W	FFFFFFFFh	Each bit protects 1 sector. bit [0]: When 1, sector 0 of the non-main region will be protected from program and erase. bit [1]: When 1, sector 1 of the non-main region will be protected from program and erase. : : bit [31]: When 1, sector 31 of the non-main will be protected from program and erase. 0h = Minimum value of [VAL] FFFFFFFFh = Maximum value of [VAL]

36 STATCMD Register (Offset = 13D0h) [Reset = 00000000h]

STATCMD is shown in Figure 11-42 and described in Table 11-55.

Return to the Summary Table.

Command Status Register This register contains status regarding completion and errors of command execution.

Figure 11-42. STATCMD Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED			FAILMISC	RESERVED			FAILINVDATA
R-0h			R-0h	R-0h			R-0h
7	6	5	4	3	2	1	0
FAILMODE	FAILILLADDR	FAILVERIFY	FAILWEPROT	RESERVED	CMDINPROGRESS	CMDPASS	CMDDONE
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

Table 11-96. STATCMD Register Field Descriptions

Bit	Field	Type	Reset	Description
31-13	RESERVED	R	0h	Reserved
12	FAILMISC	R	0h	Command failed due to error other than write/erase protect violation or verify error. This is an extra bit in case a new failure mechanism is added which requires a status bit. 0h = No Fail 1h = Fail
11-9	RESERVED	R	0h	Reserved
8	FAILINVDATA	R	0h	Program command failed because an attempt was made to program a stored 0 value to a 1. 0h = No Fail 1h = Fail
7	FAILMODE	R	0h	Command failed because a bank has been set to a mode other than READ. Program and Erase commands cannot be initiated unless all banks are in READ mode. 0h = No Fail 1h = Fail
6	FAILILLADDR	R	0h	Command failed due to the use of an illegal address 0h = No Fail 1h = Fail
5	FAILVERIFY	R	0h	Command failed due to verify error 0h = No Fail 1h = Fail
4	FAILWEPROT	R	0h	Command failed due to Write/Erase Protect Sector Violation 0h = No Fail 1h = Fail
3	RESERVED	R	0h	Reserved
2	CMDINPROGRESS	R	0h	Command In Progress 0h = Complete 1h = In Progress

ADVANCE INFORMATION

Table 11-96. STATCMD Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
1	CMDPASS	R	0h	Command Pass - valid when CMD_DONE field is 1 0h = Fail 1h = Pass
0	CMDDONE	R	0h	Command Done 0h = Not Done 1h = Done

37 STATPCNT Register (Offset = 13D8h) [Reset = 00000000h]

STATPCNT is shown in Figure 11-43 and described in Table 11-56.

Return to the Summary Table.

Current Pulse Count Register: Read only register giving read access to the state machine current pulse count value for program/erase operations.

Figure 11-43. STATPCNT Register

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

Table 11-98. STATPCNT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-12	RESERVED	R	0h	Reserved
11-0	PULSECNT	R	0h	Current Pulse Counter Value 0h = Minimum value FFFh = Maximum value

ADVANCE INFORMATION

12.1 EAM

The Error Aggregator Module (EAM) provides a log of errors related to memory access and Error Correction Code (ECC) failures in the device.

12.1.1 EAM Introduction

The EAM module's purpose is to aggregate safety and security errors from multiple sources. The EAM then stores these errors in the Memory Mapped Registers (MMR's). The types of errors are split into three groups: safety errors, security errors, and access errors. Each group has their own set of fixed priority which is defined in the Section 12.1.2 .

The top-level block diagram for the EAM is shown in Figure 12-1

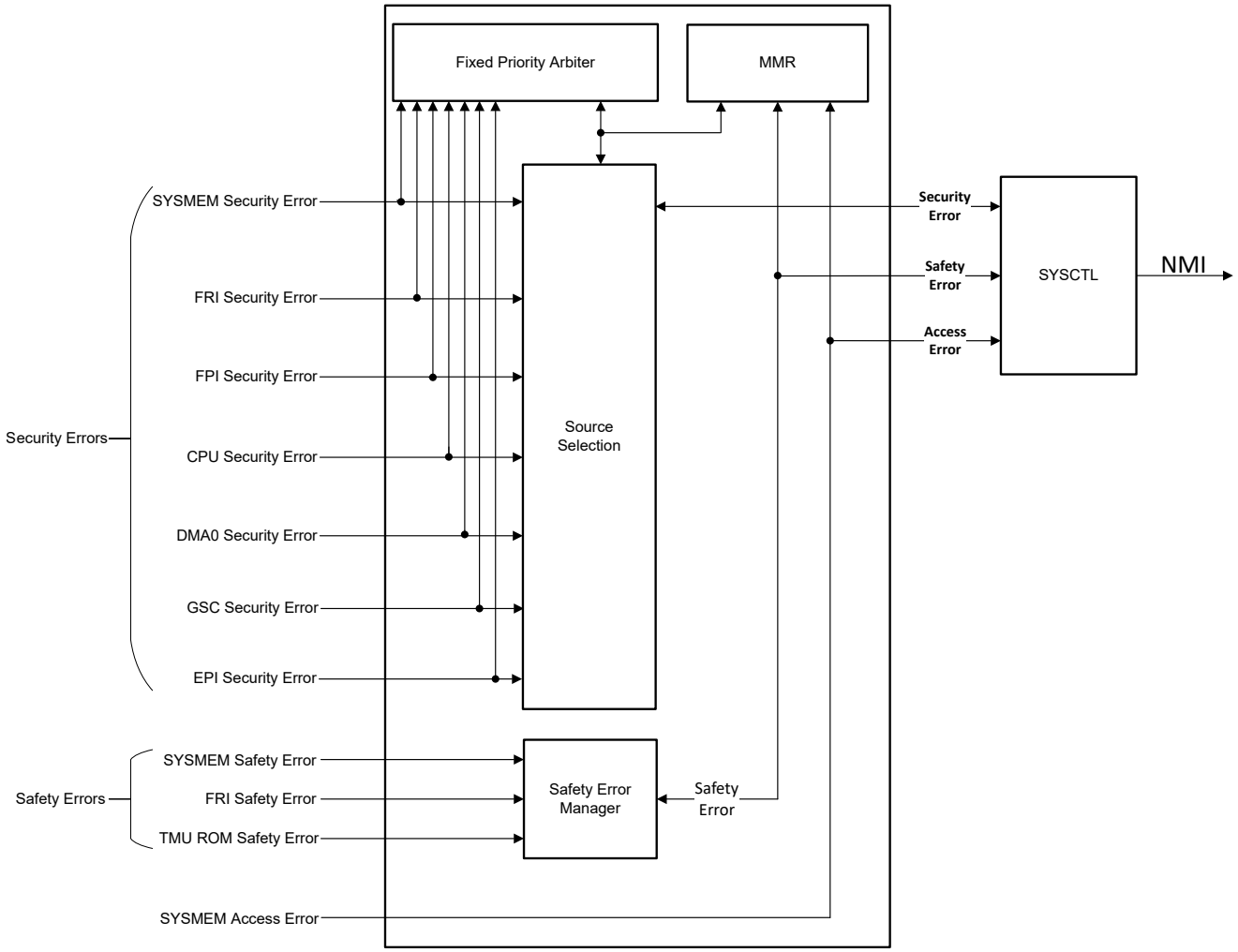


Figure 12-1. Top-Level EAM Block Diagram

12.1.2 EAM Operation

Each aggregator stores the errors from various sources with their own fixed priority. The fixed priority is determined by the fixed priority arbiter. The fixed priority table uses 1 as the highest priority and the largest number being the lowest priority. See the corresponding error aggregator on details on the corresponding fixed priority arbiter. Information on which address the error occurred, corresponding error type and error flag are stored in the respective register.

When an error occurs the following steps are taken:

1. Error source is selected based on the fixed priority arbiter
2. Error flag bit, address, and id are latched into respective registers corresponding to the selected error
3. Selected error will be forwarded to SYSCTL generating an NMI or interrupt
4. Wait for software to write to the clear register

5. After the clear register is written to the error type, flag, and address will also be cleared

During step 4 the corresponding clear register must be written to by software for the next error to properly propagate. To see the corresponding register to clear please see *EAM Registers*.

12.1.2.1 Security Error Aggregator

The security error aggregator is used to handle the security errors generated from SRAM (SYSMEM), Flash Read Interface (FRI), Flash Program Interface (FPI), CPU, DMA0, and the Global Security Controller (GSC). These modules all generate different types of security errors that are propagated to the security error aggregator. After receiving these errors, the security error aggregator generates an security error non-maskable interrupt (NMI). Detailed information about the NMI can be seen in the SYSCTL Events section.

Note

AM13E230x devices do not have ARM TrustZone technology enabled. All security errors are due to privilege policy violations.

When one of the security errors occur, the security fixed priority arbiter selects the error source based on the Table 12-1. After the security fixed priority arbiter selects an error, the corresponding error is stored in the SECURITY_ERR_FLAG register. Then the error is forwarded to SYSCTL and causes a security error NMI. This transaction can be seen in the Figure 12-2.

Table 12-1. Security Error Aggregator's Fixed Priority table

Priority	Source
1 (Highest Priority)	MEM0
2	MEM1
3	MEM2
4	MEM3
5	FRI
6	FPI
7	CPU
8	DMA0
9	Reserved
10	GSC
11	EPI

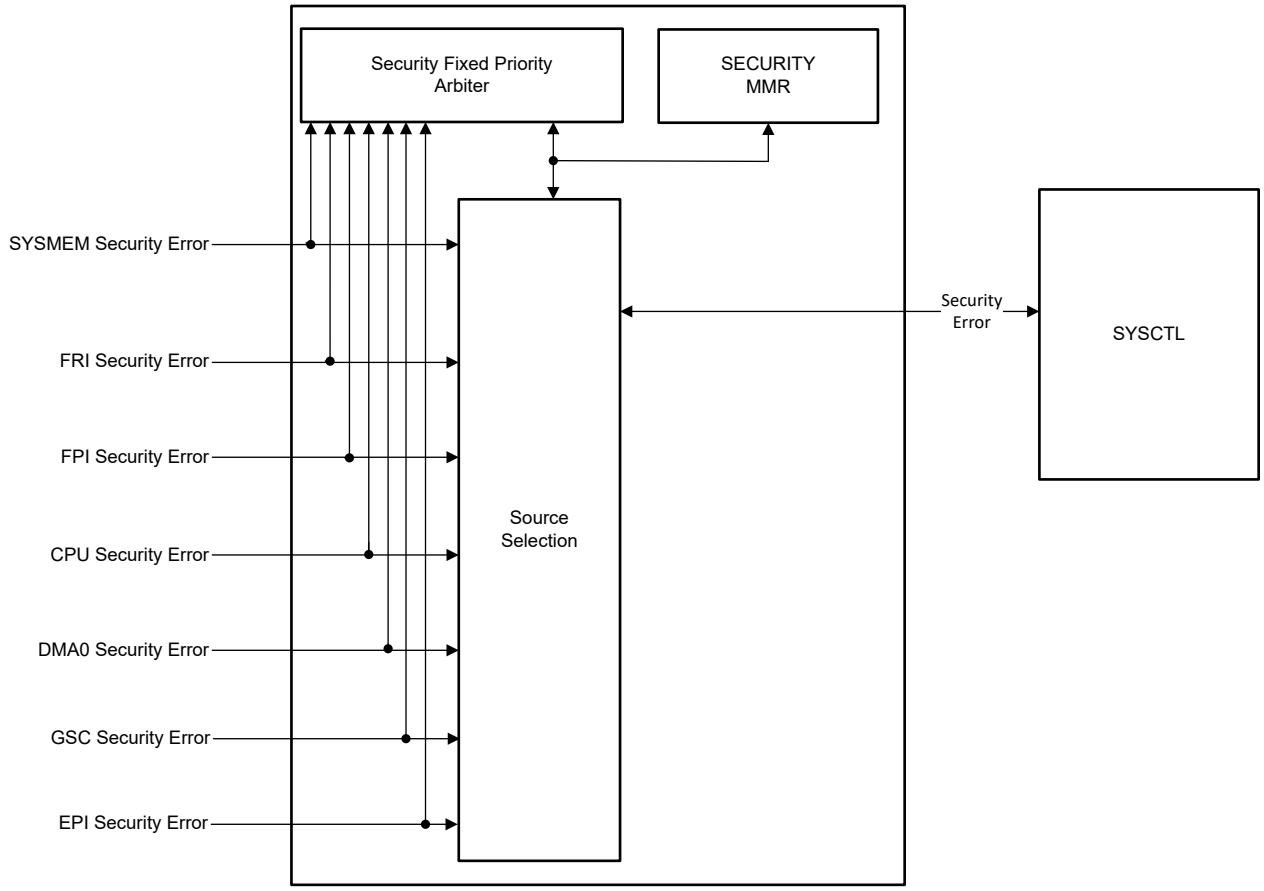


Figure 12-2. Security Error Aggregator Block Diagram

To properly understand the error that has occurred, information about the error can be found in the SECURITY_ERR_MSTID and SECURITY_ERR_ADDR registers. To clear the error and propagate the next error, the SECURITY_ERR_CLR register must be written to. Please see the *EAM Registers* section for a more detailed description on these registers.

12.1.2.2 Safety Error Aggregator

The safety error aggregator is the other error aggregator in the EAM. This error aggregator handles multiple safety errors caused by ECC or parity failures.When the safety error aggregator receives an ECC or parity error the module causes a NMI or an interrupt depending on the error. Information on the different types of errors can be seen in the SYSCTL Events section. These ECC or parity errors can occur from the Flash Read Interface (FRI) or the SRAM memory (SYSMEM). Information on the ECC or parity error can be seen in the ECC Error Handling section of the flash chapter and information on the SRAM errors can be seen in the SRAM region section of the Architecture chapter. The key difference is that the safety error aggregator can generate NMIs as well as interrupts to handle the different types of ECC errors. See Table 12-2 for details on which interrupts are generated from the corresponding error.

Table 12-2. Safety Error Aggregator NMI and Interrupt table

Error	Resulting Interrupt
Flash Read ECC SEC	FLASHSEC Interrupt
Flash Read ECC DED	FLASHDED NMI
SRAM MEM0 - SRAM MEM3 Parity error	SRAMPARITY NMI

ADVANCE INFORMATION

Due to the single bit errors (SEC) and dual bit errors (DED) generating different types of interrupts there are 4 sets of MMR's for this error aggregator: FRI_SEC, FRI_DED, SYSMEM_PARITY, and TMU_PARITY. Each one of these groups of MMR's has an individual set of MSTID, ADDR, FLAG, and CLR registers. To see detailed descriptions on these registers please see *EAM Registers*.

For a visual representation of the safety error aggregator module, please see Figure 12-3 to find the configuration of the source selectors, fixed priority arbiters, and segregation module. The fixed priority arbiter is only used for SYSMEM errors and as such the SYSMEM_PARITY uses the same priority level. The fixed priority arbiter is not needed for the flash read interface as there is only one type of flash error from the SEC and DED interfaces. To see the different priority levels please see table Table 12-3.

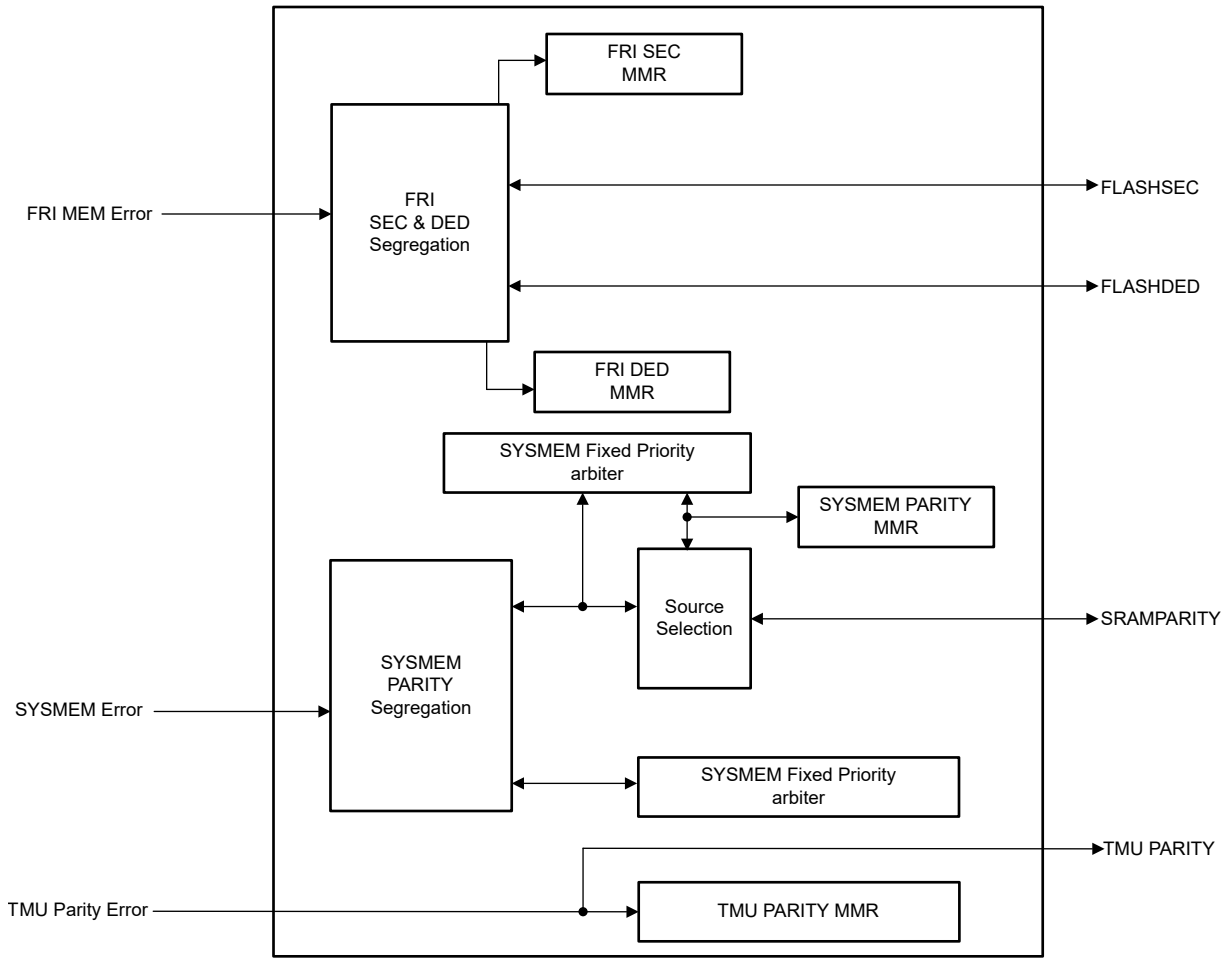


Figure 12-3. Safety Error Aggregator Block Diagram

Table 12-3. Safety Error Aggregator's Fixed Priority table

Priority	Source
1 (Highest Priority)	MEM0
2	MEM1
3	MEM2
4	MEM3

ADVANCE INFORMATION

12.1.2.3 SYSMEM Access Error

The third type of error handled by the EAM is a SYSMEM Access Error. This error exists due to the AM13E230x device design which dual maps SRAM3 (MEM3) to both the C-AHB (CPU bus) and S-AHB (System bus). Refer to the Architecture chapter for more information on the AM13E230x data buses.

There are two situations in which an access error is triggered:

1. The SRAM3_STATIC_MUX is configured for C-AHB access (default, SRAM3_STATIC_MUX_SEL = 1h) and S-AHB access is attempted.
2. The SRAM3_STATIC_MUX mux is configured for S-AHB access (SRAM3_STATIC_MUX_SEL = 0h) and C-AHB access is attempted.

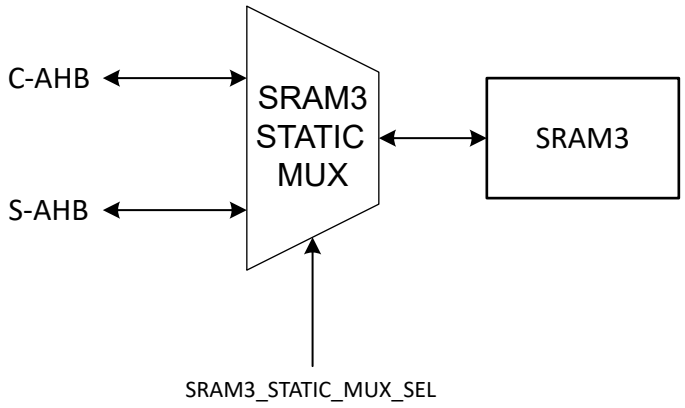


Figure 12-4. SYSMEM Access Error

12.2 EAM Registers

This Section describes the EAM Registers.

12.2.1 EAM Base Address Table

Table 12-4. EAM Base Address Table

Bit Field Name		DriverLib Name	Base Address
Instance	Structure		
EamRegs	EAM_REGS	EAM	0x4002_C000

12.2.2 EAM_REGS Registers

Table 12-5 lists the memory-mapped registers for the EAM_REGS registers. All register offset addresses not listed in Table 12-5 should be considered as reserved locations and the register contents should not be modified.

Table 12-5. EAM_REGS Registers

Offset	Acronym	Register Name	Write Protection	Section
1000h	REVISION	IP revision id register		Go
1010h	SECURITY_ERR_FLAG	Security error flag		Go
1014h	SECURITY_ERR_CLR	Security error clear		Go
1018h	SECURITY_ERR_MSTID	Security error master id		Go
101Ch	SECURITY_ERR_ADDR	Security error address		Go
1030h	FRI_SEC_FLAG	FRI SEC flag		Go
1034h	FRI_SEC_CLR	FRI SEC clear		Go
1038h	FRI_SEC_ADDR	FRI SEC address		Go
103Ch	FRI_SEC_MSTID	FRI SEC master id		Go
1050h	FRI_DED_FLAG	FRI DED flag		Go
1054h	FRI_DED_CLR	FRI DED clear		Go
1058h	FRI_DED_ADDR	FRI DED address		Go
105Ch	FRI_DED_MSTID	FRI DED master id		Go
10B0h	SYSMEM_PAR_FLAG	SYSMEM DED flag		Go
10B4h	SYSMEM_PAR_CLR	SYSMEM DED clear		Go
10B8h	SYSMEM_PAR_ADDR	SYSMEM DED address		Go
10BCh	SYSMEM_PAR_MSTID	SYSMEM DED master id		Go
10D0h	TMUROM_PAR_FLAG	SYSMEM DED flag		Go
10D4h	TMUROM_PAR_CLR	SYSMEM DED clear		Go
10D8h	TMUROM_PAR_ADDR	SYSMEM DED address		Go
10DCh	TMUROM_PAR_TYPE	SYSMEM DED master id		Go
10F0h	SYSMEM_ACC_FLAG	SYSMEM access error flag		Go
10F4h	SYSMEM_ACC_CLR	SYSMEM access error clear		Go
10F8h	SYSMEM_ACC_ADDR	SYSMEM access error address		Go
10FCh	SYSMEM_ACC_TYPE	SYSMEM access error type		Go

Complex bit access types are encoded to fit into small table cells. Table 12-6 shows the codes that are used for access types in this section.

Table 12-6. EAM_REGS Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
R-0	R -0	Read Returns 0s
Write Type		
W1C	W 1C	Write 1 to clear
Reset or Default Value		
-n		Value after reset or the default value
Register Array Variables		

Table 12-6. EAM_REGS Access Type Codes (continued)

Access Type	Code	Description
i,j,k,l,m,n		When these variables are used in a register name, an offset, or an address, they refer to the value of a register array where the register is part of a group of repeating registers. The register groups form a hierarchical structure and the array is represented with a formula.
y		When this variable is used in a register name, an offset, or an address it refers to the value of a register array.

1 REVISION Register (Offset = 1000h) [Reset = 00000000h]

REVISION is shown in Figure 12-5 and described in Table 12-7.

Return to the Summary Table.

IP revision id register

Figure 12-5. REVISION Register

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

Table 12-8. REVISION Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R-0	0h	Reserved
0	REVISION	R	0h	Revision register Reset type: PORESETn

ADVANCE INFORMATION

2 SECURITY_ERR_FLAG Register (Offset = 1010h) [Reset = 00000000h]

SECURITY_ERR_FLAG is shown in Figure 12-6 and described in Table 12-8.

Return to the Summary Table.

Security error flag

Figure 12-6. SECURITY_ERR_FLAG Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED				FLSEM_ACCESS_ERROR	FLC_MMR_ACCESS_ERROR	FPI_ILLSIZE	FPI_ILLCMD
R-0-0h				R-0h	R-0h	R-0h	R-0h
7	6	5	4	3	2	1	0
FPI_ILLMODECH	FPI_ILLRDVER	FPI_ILLERASE	FPI_ILLPROG	FPI_ILLADDR	HDP	PRIV	RESERVED
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

Table 12-10. SECURITY_ERR_FLAG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-12	RESERVED	R-0	0h	Reserved
11	FLSEM_ACCESS_ERROR	R	0h	FLSEM ACCESS ERROR flag Reset type: PORESETn
10	FLC_MMR_ACCESS_ERROR	R	0h	FLC MMR ACCESS ERROR flag Reset type: PORESETn
9	FPI_ILLSIZE	R	0h	FPI ILLSIZE flag Reset type: PORESETn
8	FPI_ILLCMD	R	0h	FPI ILLCMD flag Reset type: PORESETn
7	FPI_ILLMODECH	R	0h	FPI ILLMODECH flag Reset type: PORESETn
6	FPI_ILLRDVER	R	0h	FPI ILLRDVER flag Reset type: PORESETn
5	FPI_ILLERASE	R	0h	FPI ILLERASE flag Reset type: PORESETn
4	FPI_ILLPROG	R	0h	FPI ILLPROG flag Reset type: PORESETn
3	FPI_ILLADDR	R	0h	FPI ILLADDR flag Reset type: PORESETn

Table 12-10. SECURITY_ERR_FLAG Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
2	HDP	R	0h	Hide protection error flag Reset type: PORESETn
1	PRIV	R	0h	Privilege error flag Reset type: PORESETn
0	RESERVED	R	0h	Reserved

3 SECURITY_ERR_CLR Register (Offset = 1014h) [Reset = 00000000h]

SECURITY_ERR_CLR is shown in Figure 12-7 and described in Table 12-9.

Return to the Summary Table.

Security error clear

Figure 12-7. SECURITY_ERR_CLR Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED				FLSEM_ACCESS_ERROR	FLC_MMR_ACCESS_ERROR	FPI_ILLSIZE	FPI_ILLCMD
R-0-0h				R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h
7	6	5	4	3	2	1	0
FPI_ILLMODECH	FPI_ILLRDVER	FPI_ILLERASE	FPI_ILLPROG	FPI_ILLADDR	HDP	PRIV	RESERVED
R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h

Table 12-12. SECURITY_ERR_CLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-12	RESERVED	R-0	0h	Reserved
11	FLSEM_ACCESS_ERROR	R-0/W1C	0h	writing '1' will clear SECURITY_ERR_FLAG[FLSEM_ACCESS_ERROR] register. Reset type: PORESETn
10	FLC_MMR_ACCESS_ERROR	R-0/W1C	0h	writing '1' will clear SECURITY_ERR_FLAG[FLC_MMR_ACCESS_ERROR] register. Reset type: PORESETn
9	FPI_ILLSIZE	R-0/W1C	0h	writing '1' will clear SECURITY_ERR_FLAG[FPI_ILLSIZE] register. Reset type: PORESETn
8	FPI_ILLCMD	R-0/W1C	0h	writing '1' will clear SECURITY_ERR_FLAG[FPI_ILLCMD] register. Reset type: PORESETn
7	FPI_ILLMODECH	R-0/W1C	0h	writing '1' will clear SECURITY_ERR_FLAG[FPI_ILLMODECH] register. Reset type: PORESETn
6	FPI_ILLRDVER	R-0/W1C	0h	writing '1' will clear SECURITY_ERR_FLAG[FPI_ILLRDVER] register. Reset type: PORESETn
5	FPI_ILLERASE	R-0/W1C	0h	writing '1' will clear SECURITY_ERR_FLAG[FPI_ILLERASE] register. Reset type: PORESETn
4	FPI_ILLPROG	R-0/W1C	0h	writing '1' will clear SECURITY_ERR_FLAG[FPI_ILLPROG] register. Reset type: PORESETn

Table 12-12. SECURITY_ERR_CLR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3	FPI_ILLADDR	R-0/W1C	0h	writing '1' will clear SECURITY_ERR_FLAG[FPI_ILLADDR] register. Reset type: PORESETn
2	HDP	R-0/W1C	0h	writing '1' will clear SECURITY_ERR_FLAG[HDP] register. Reset type: PORESETn
1	PRIV	R-0/W1C	0h	writing '1' will clear SECURITY_ERR_FLAG[PRIV] register. Reset type: PORESETn
0	RESERVED	R-0/W1C	0h	Reserved

4 SECURITY_ERR_MSTID Register (Offset = 1018h) [Reset = 00000000h]

SECURITY_ERR_MSTID is shown in Figure 12-8 and described in Table 12-10.

Return to the Summary Table.

Security error master id

Figure 12-8. SECURITY_ERR_MSTID Register

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

Table 12-14. SECURITY_ERR_MSTID Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R-0	0h	Reserved
7-0	MSTID	R	0h	Security error master ID. This field will be cleared along with the corresponding flag clear. Reset type: PORESETn

5 SECURITY_ERR_ADDR Register (Offset = 101Ch) [Reset = 00000000h]

SECURITY_ERR_ADDR is shown in Figure 12-9 and described in Table 12-11.

Return to the Summary Table.

Security error address

Figure 12-9. SECURITY_ERR_ADDR Register

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

Table 12-16. SECURITY_ERR_ADDR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ADDRESS	R	0h	Security error address. This field will be cleared along with the corresponding flag clear. Reset type: PORESETn

6 FRI_SEC_FLAG Register (Offset = 1030h) [Reset = 00000000h]

FRI_SEC_FLAG is shown in Figure 12-10 and described in Table 12-12.

Return to the Summary Table.

FRI single error correction flag

Figure 12-10. FRI_SEC_FLAG Register

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

Table 12-18. FRI_SEC_FLAG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R-0	0h	Reserved
0	SEC	R	0h	Single error correction error Reset type: PORESETn

ADVANCE INFORMATION

7 FRI_SEC_CLR Register (Offset = 1034h) [Reset = 00000000h]

FRI_SEC_CLR is shown in Figure 12-11 and described in Table 12-13.

Return to the Summary Table.

FRI single error correction clear

Figure 12-11. FRI_SEC_CLR Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
R-0-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED															SEC
R-0-0h															R-0/W1C-0h

Table 12-20. FRI_SEC_CLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R-0	0h	Reserved
0	SEC	R-0/W1C	0h	writing '1' will clear FRI_SEC_FLAG[SEC] register. Reset type: PORESETn

ADVANCE INFORMATION

8 FRI_SEC_ADDR Register (Offset = 1038h) [Reset = 00000000h]

FRI_SEC_ADDR is shown in Figure 12-12 and described in Table 12-14.

Return to the Summary Table.

FRI single error correction address

Figure 12-12. FRI_SEC_ADDR Register

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

Table 12-22. FRI_SEC_ADDR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ADDRESS	R	0h	Single error correction error address. This field will be cleared along with the corresponding flag clear. Reset type: PORESETn

ADVANCE INFORMATION

9 FRI_SEC_MSTID Register (Offset = 103Ch) [Reset = 00000000h]

FRI_SEC_MSTID is shown in Figure 12-13 and described in Table 12-15.

Return to the Summary Table.

FRI single error correction master id

Figure 12-13. FRI_SEC_MSTID Register

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

Table 12-24. FRI_SEC_MSTID Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R-0	0h	Reserved
7-0	MSTID	R	0h	Single error correction error master ID. This field will be cleared along with the corresponding flag clear. Reset type: PORESETn

10 FRI_DED_FLAG Register (Offset = 1050h) [Reset = 00000000h]

FRI_DED_FLAG is shown in Figure 12-14 and described in Table 12-16.

Return to the Summary Table.

FRI double error detection flag

Figure 12-14. FRI_DED_FLAG Register

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

Table 12-26. FRI_DED_FLAG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R-0	0h	Reserved
1	DIAG_DED	R	0h	diagnostic Double error detect error Reset type: PORESETn
0	DED	R	0h	Double error detect error Reset type: PORESETn

11 FRI_DED_CLR Register (Offset = 1054h) [Reset = 00000000h]

FRI_DED_CLR is shown in Figure 12-15 and described in Table 12-17.

Return to the Summary Table.

FRI double error detection clear

Figure 12-15. FRI_DED_CLR Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED						DIAG_DED	SEC
R-0-0h						R-0/W1C-0h	R-0/W1C-0h

Table 12-28. FRI_DED_CLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R-0	0h	Reserved
1	DIAG_DED	R-0/W1C	0h	writing '1' will clear FRI_DED_FLAG[DIAG_DED] register. Reset type: PORESETn
0	SEC	R-0/W1C	0h	writing '1' will clear FRI_DED_FLAG[DED] register. Reset type: PORESETn

ADVANCE INFORMATION

12 FRI_DED_ADDR Register (Offset = 1058h) [Reset = 00000000h]

FRI_DED_ADDR is shown in Figure 12-16 and described in Table 12-18.

Return to the Summary Table.

FRI double error detection address

Figure 12-16. FRI_DED_ADDR Register

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

Table 12-30. FRI_DED_ADDR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ADDRESS	R	0h	Single error correction error address. This field will be cleared along with the corresponding flag clear. Reset type: PORESETn

ADVANCE INFORMATION

13 FRI_DED_MSTID Register (Offset = 105Ch) [Reset = 00000000h]

FRI_DED_MSTID is shown in Figure 12-17 and described in Table 12-19.

Return to the Summary Table.

FRI double error detection master id

Figure 12-17. FRI_DED_MSTID Register

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

Table 12-32. FRI_DED_MSTID Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R-0	0h	Reserved
7-0	MSTID	R	0h	Single error correction error master ID. This field will be cleared along with the corresponding flag clear. Reset type: PORESETn

14 SYSMEM_PAR_FLAG Register (Offset = 10B0h) [Reset = 00000000h]

SYSMEM_PAR_FLAG is shown in Figure 12-18 and described in Table 12-20.

Return to the Summary Table.

SYSMEM parity error flag

Figure 12-18. SYSMEM_PAR_FLAG Register

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

Table 12-34. SYSMEM_PAR_FLAG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R-0	0h	Reserved
1	WR_PAR	R	0h	Parity error during write Reset type: PORESETn
0	RD_PAR	R	0h	Parity error during read Reset type: PORESETn

ADVANCE INFORMATION

15 SYSMEM_PAR_CLR Register (Offset = 10B4h) [Reset = 00000000h]

SYSMEM_PAR_CLR is shown in Figure 12-19 and described in Table 12-21.

Return to the Summary Table.

SYSMEM parity error clear

Figure 12-19. SYSMEM_PAR_CLR Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED						WR_PAR	RD_PAR
R-0-0h						R-0/W1C-0h	R-0/W1C-0h

Table 12-36. SYSMEM_PAR_CLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R-0	0h	Reserved
1	WR_PAR	R-0/W1C	0h	writing '1' will clear SYSMEM_PAR_FLAG[WR_PAR] register. Reset type: PORESETn
0	RD_PAR	R-0/W1C	0h	writing '1' will clear SYSMEM_PAR_FLAG[RD_PAR] register. Reset type: PORESETn

ADVANCE INFORMATION

16 SYSMEM_PAR_ADDR Register (Offset = 10B8h) [Reset = 00000000h]

SYSMEM_PAR_ADDR is shown in Figure 12-20 and described in Table 12-22.

Return to the Summary Table.

SYSMEM parity error address

Figure 12-20. SYSMEM_PAR_ADDR Register

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

Table 12-38. SYSMEM_PAR_ADDR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ADDRESS	R	0h	Parity error address. This field will be cleared along with the corresponding flag clear. Reset type: PORESETn

17 SYSMEM_PAR_MSTID Register (Offset = 10BCh) [Reset = 00000000h]

SYSMEM_PAR_MSTID is shown in Figure 12-21 and described in Table 12-23.

Return to the Summary Table.

SYSMEM parity error master id

Figure 12-21. SYSMEM_PAR_MSTID Register

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

Table 12-40. SYSMEM_PAR_MSTID Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R-0	0h	Reserved
7-0	MSTID	R	0h	Parity error master ID. This field will be cleared along with the corresponding flag clear. Reset type: PORESETn

18 TMUROM_PAR_FLAG Register (Offset = 10D0h) [Reset = 00000000h]

TMUROM_PAR_FLAG is shown in Figure 12-22 and described in Table 12-24.

Return to the Summary Table.

TMUROM parity error flag

Figure 12-22. TMUROM_PAR_FLAG Register

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

Table 12-42. TMUROM_PAR_FLAG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R-0	0h	Reserved
0	PAR	R	0h	Parity error Reset type: PORESETn

19 TMUROM_PAR_CLR Register (Offset = 10D4h) [Reset = 00000000h]

TMUROM_PAR_CLR is shown in Figure 12-23 and described in Table 12-25.

Return to the Summary Table.

TMUROM parity error clear

Figure 12-23. TMUROM_PAR_CLR Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
R-0-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED															PAR
R-0-0h															R-0/W1C-0h

Table 12-44. TMUROM_PAR_CLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R-0	0h	Reserved
0	PAR	R-0/W1C	0h	writing '1' will clear TMUROM_PAR_FLAG[PAR] register. Reset type: PORESETn

20 TMUROM_PAR_ADDR Register (Offset = 10D8h) [Reset = 00000000h]

TMUROM_PAR_ADDR is shown in Figure 12-24 and described in Table 12-26.

Return to the Summary Table.

TMUROM parity error address

Figure 12-24. TMUROM_PAR_ADDR Register

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

Table 12-46. TMUROM_PAR_ADDR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-15	RESERVED	R-0	0h	Reserved
14-0	ADDRESS	R	0h	Parity error address. This field will be cleared along with the corresponding flag clear. Reset type: PORESETn

21 TMUROM_PAR_TYPE Register (Offset = 10DCh) [Reset = 00000000h]

TMUROM_PAR_TYPE is shown in Figure 12-25 and described in Table 12-27.

Return to the Summary Table.

TMUROM parity error type

Figure 12-25. TMUROM_PAR_TYPE Register

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

Table 12-48. TMUROM_PAR_TYPE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R-0	0h	Reserved
2-0	TYPE	R	0h	Parity error type. This field will be cleared along with the corresponding flag clear. [0] - Y0i ROM table read has parity error [1] - S1i ROM table read has parity error [2] - S2i ROM table read has parity error Reset type: PORESETn

ADVANCE INFORMATION

22 SYSMEM_ACC_FLAG Register (Offset = 10F0h) [Reset = 00000000h]

SYSMEM_ACC_FLAG is shown in Figure 12-26 and described in Table 12-28.

Return to the Summary Table.

SYSMEM access error flag

Figure 12-26. SYSMEM_ACC_FLAG Register

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

Table 12-50. SYSMEM_ACC_FLAG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	
0	SYSMEM access error	R	0h	

23 SYSMEM_ACC_CLR Register (Offset = 10F4h) [Reset = 00000000h]

SYSMEM_ACC_CLR is shown in Figure 12-27 and described in Table 12-29.

Return to the Summary Table.

SYSMEM access error clear

Figure 12-27. SYSMEM_ACC_CLR Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
R-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED															ERR
R-0h															R/W1C-0h

Table 12-52. SYSMEM_ACC_CLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	Reserved
0	ERR	R/W1C	0h	writing '1' will clear SYSMEM_ACC_FLAG[ERR] register.

ADVANCE INFORMATION

24 SYSMEM_ACC_ADDR Register (Offset = 10F8h) [Reset = 00000000h]

SYSMEM_ACC_ADDR is shown in Figure 12-28 and described in Table 12-30.

Return to the Summary Table.

SYSMEM access error address

Figure 12-28. SYSMEM_ACC_ADDR Register

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

Table 12-54. SYSMEM_ACC_ADDR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ADDRESS	R	0h	Access error address. This field will be cleared along with the corresponding flag clear.

25 SYSMEM_ACC_TYPE Register (Offset = 10FCh) [Reset = 00000000h]

SYSMEM_ACC_TYPE is shown in Figure 12-29 and described in Table 12-31.

Return to the Summary Table.

SYSMEM access error type

Figure 12-29. SYSMEM_ACC_TYPE Register

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

Table 12-56. SYSMEM_ACC_TYPE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-5	RESERVED	R	0h	Reserved
4-0	TYPE	R	0h	Access error type. This field will be cleared along with the corresponding flag clear. [4] - CBUS Access Error [3] - SBUS Access Error [2] - Write Error [1] - Read Error [0] - Fetch Error

Chapter 13
Events

The event manager provides the peripheral-to-ADC, peripheral-to-DMA, and peripheral-to-CPU (IRQ) event connections.

13.1 Events Overview

The event manager transfers digital events from one entity (for example, a peripheral) to another (for example, the CPU, DMA, or ADC). The event manager implements event transfer through a defined set of event generators (publishers) and receivers (subscribers) which are interconnected through an event fabric containing a variety of fixed (static) routes.

Events which are transferred by the event manager include:

- CPU Interrupts - Peripheral event transferred to the CPU as an interrupt request (IRQ)
  - Example: Timer TIMx interrupt is sent to the CPU
- DMA Triggers - Peripheral event transferred to the DMA as a DMA trigger
  - Example: UNICOMM-UART data receive trigger to DMA to request a DMA transfer
- ADC SOC Trigger - Peripheral event transferred to the ADC to directly trigger an action in hardware
  - Example: Timer TIMx peripheral generates a periodic event to the ADC, and the ADC uses the event to trigger a start-of-conversion (SOC).

13.1.1 Event Publisher

An event publisher is the source of an event that is propagated on the event fabric. Peripherals are event publishers for three different event types: CPU interrupts, DMA triggers and ADC SOC events configured on the device. Publisher behavior configured in peripheral registers. Some peripherals use the standard Event Management Registers, some use a unique set of registers, and some have fixed settings that are not user configurable. See Table 13-1 for all the event publishers and the event registers present in each peripheral.

Table 13-1. Event Registers Per Peripheral

Peripheral Publisher	Standard/IP-specific/Fixed Condition	Event Registers
ADC	IP-specific	ADCINTSEL, ADCDMAINTSEL, ADCRAWINTFLG, ADCINTFLG, ADCINTFLGFRC, ADCINTFLGCLR
MCPWM	IP-specific	INTEN, INTFLAG, INTCLR, INTFRC
GPIO	Standard	IIDX, IMASK, RIS, MIS, ISET, ICLR
DMA	Standard	IIDX, IMASK, RIS, MIS, ISET, ICLR
TIM (timers)	Standard	IIDX, IMASK, RIS, MIS, ISET, ICLR
ECAP	IP-specific	ECINT, ECFLG, ECCLR, ECFRC
EQEP	IP-specific	QEINT, QFLG, QCLR, QFRC
MCAN	Standard	IIDX, IMASK, RIS, MIS, ISET, ICLR
WWDT	Standard	IIDX, IMASK, RIS, MIS, ISET, ICLR
TINIE_LITE	Fixed Condition	N/A
Flash	Standard	IIDX, IMASK, RIS, MIS, ISET, ICLR
TMU	Fixed Condition	N/A
EPI	IP-specific	EPIM, EPIRIS, EPIMIS, EPIEISC
AES	Standard	IIDX, IMASK, RIS, MIS, ISET, ICLR

13.1.1.1 Standard Event Registers

The standard event management register set contains 6 registers: RIS, IMASK, MIS, ISET, ICLR, and IIDX.

ADVANCE INFORMATION

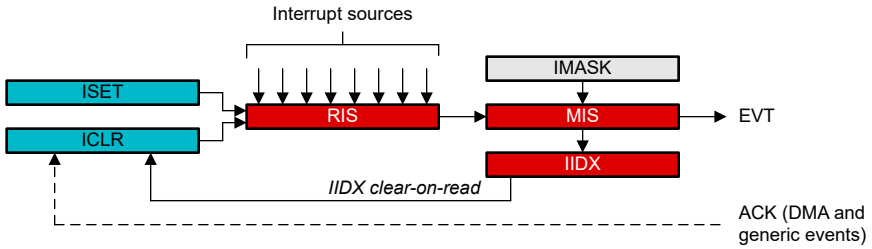


Figure 13-1. Event Management Register Relationship

The peripheral generating the event contains one or more interrupt source signals which connect to the raw interrupt status (RIS) register. Software can poll RIS at any time to check the raw interrupt status. Software can also clear pending interrupts in the RIS register by writing to the corresponding bit position in the ICLR register. The RIS and IMASK registers are combined through a bit-wise AND function in the MIS register (masked interrupt status). To unmask an interrupt, set the corresponding bit in the IMASK register. Once unmasked, a pending interrupt is flagged in both the RIS and MIS registers, and an event is generated. The IIDX register also be updates with the index of the highest priority pending interrupt.

For a CPU interrupt (CPU_INT) with a Section 13.1.4.1 , a read of the IIDX register clears the highest priority pending interrupt in the RIS and MIS registers and return the index of the highest priority pending interrupt to application software.

For a DMA trigger event (DMA_TRIGx, DMA_TRIG_TX, DMA_TRIG_RX) with a Section 13.1.4.2 , the hardware four-way handshake sends an ACK signal to the ICLR mechanism, which clears the pending interrupt in the RIS and MIS registers.

For an ADC conversion event (ADC_SOCx) with a Section 13.1.4.3 , the hardware four-way handshake sends an ACK signal to the ICLR mechanism, which clears the pending interrupt in the RIS and MIS registers.

Table 13-2. Standardized Event Management Registers

Register	Description	R/W	Functionality
RIS	Raw interrupt status	R	Indicates the current pending interrupt status, with one bit provided per interrupt condition. Writing to ICLR clears the corresponding bit in the RIS register if the interrupt condition is no longer present.
IMASK	Interrupt mask	RW	Used by application software to configure which interrupt conditions propagate into an event, with one bit provided per interrupt condition.
MIS	Masked interrupt status	R	Indicates the current pending masked interrupt status to software and hardware, with one bit provided per interrupt condition. MIS is the bit-wise AND of the RIS and IMASK registers. Writing to ICLR clears the corresponding bit in the RIS register if the interrupt condition is no longer present. If RIS is cleared, the corresponding bit in the MIS register is also automatically cleared.
ISET	Software interrupt set control	W	Used by application software to force an interrupt condition for diagnostics. Writing to ISET sets the corresponding bit in the RIS register. If the interrupt condition is enabled in IMASK, the corresponding bit in the MIS register is also set. Writing a '1' to a bit location in ISET sets the respective interrupt status.
ICLR	Software interrupt clear control	W	Used by application software to clear a pending interrupt status in RIS. Writing a '1' to a bit location in ICLR clears the respective interrupt status. If an interrupt is enabled in IMASK, the corresponding bit location in MIS is also cleared automatically when RIS clears. If the interrupt condition is still present, clearing the status has no effect and the RIS remains set.
IIDX	Pending interrupt index	R	Used by application software to read the highest priority pending interrupt while simultaneously clearing the highest priority interrupt status in RIS and MIS. A read of IIDX returns 0 if no unmasked interrupts are pending (MIS==0), else an index value is returned indicating the highest priority pending interrupt.

13.1.2 Event Subscriber

Event subscribers include the CPU, the DMA, and the ADC (see Section 13.1.3). Event subscribers are modules that take a predefined action upon specific events which are published to the event fabric by an Event Publisher.

13.1.3 Event Routing Map

The event capabilities of each peripheral type are shown in AM13E230x Event Routing Map.

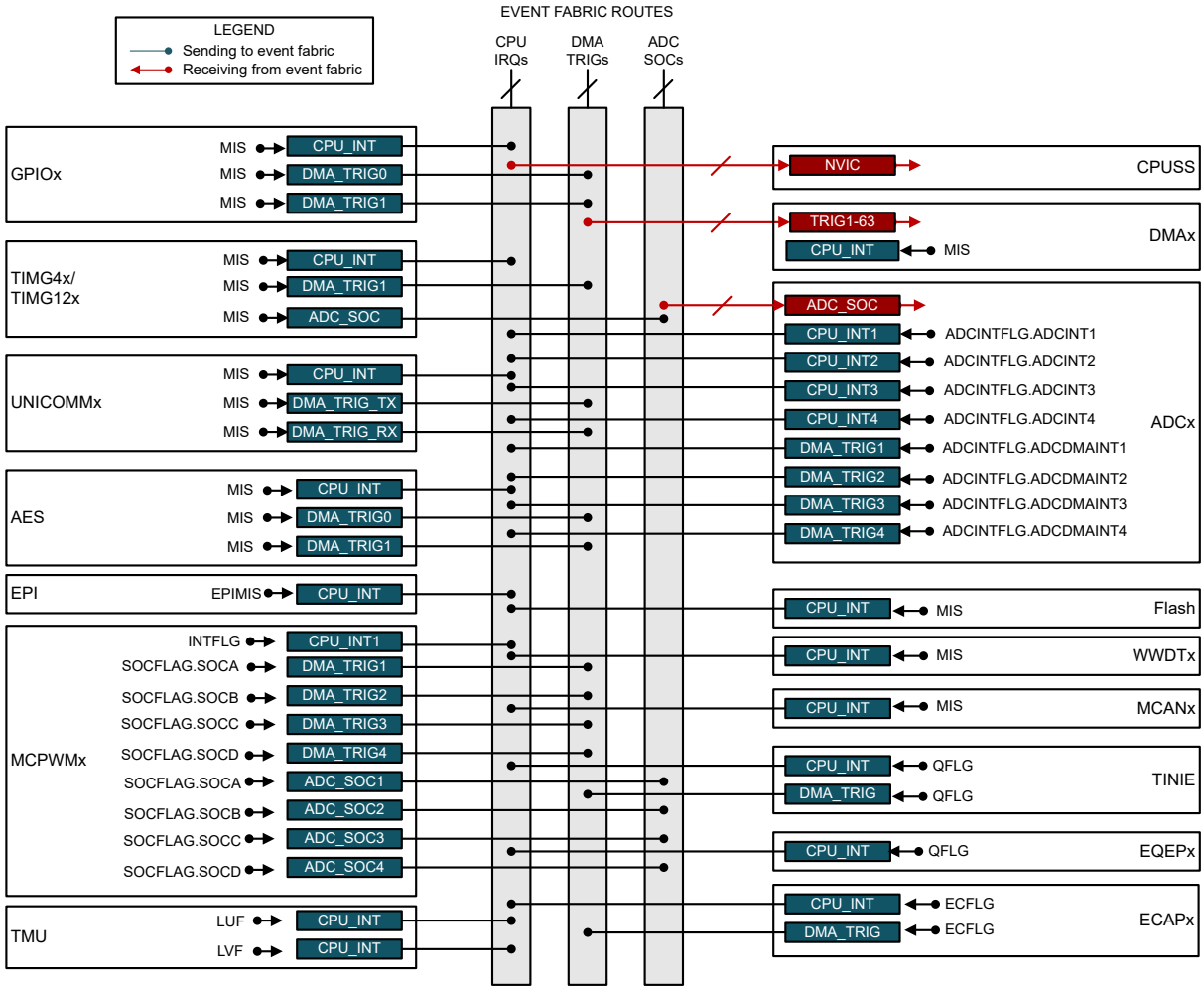


Figure 13-2. AM13E23x Event Routing Map

13.1.4 Event Fabric Routing

There are three different types of fixed routes through the event fabric that connect a publisher to a subscriber, depending on which peripheral is the subscriber: CPU Interrupt Events, DMA Trigger Events, and ADC SOC Events.

13.1.4.1 CPU Interrupt Event Route (CPU_INT)

A CPU interrupt event route is a fixed, point-to-point connection between a peripheral event publisher and the CPU subsystem, used to propagate CPU interrupts.

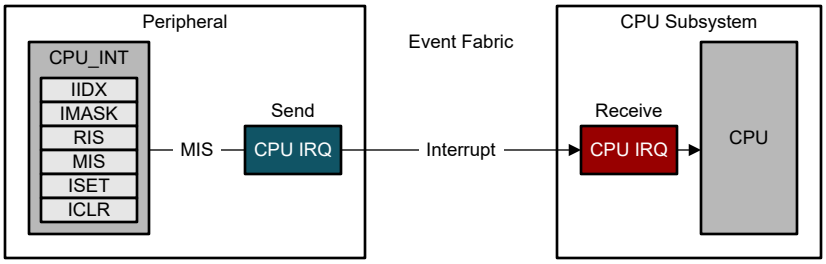


Figure 13-3. CPU Interrupts

For each peripheral that is capable of generating a CPU interrupt, a fixed route is provided from the peripheral to the CPU.

Standard Event Registers CPU_INTs

In the case of peripheral publishers that have the standard Event Management Registers, the route goes from the masked interrupt status (MIS) register to the CPU subsystem's interrupt management logic. If software does not clear the interrupt request by reading IIDX or writing to the bit in ICLR, the request will remain pending to the CPU subsystem. See *Responding to CPU Interrupts in Application Software* for guidance on setting and clearing interrupt status with the event management registers.

IP-Specific Event Registers

For CPU_INTs on publishing peripherals that implement a unique register set for managing CPU interrupts, there is also unique logic used to configure each trigger condition. Each ADC instance, for example, has five unique interrupt lines to the CPU, where only one condition can be used to trigger each path. Each MCPWM peripheral however, has only one interrupt path to the CPU, but multiple conditions can be OR'ed together for the same IRQ. See each peripheral-specific section of this document for guidance on how to configure interrupts for each of these peripherals.

Fixed Condition

For a few CPU interrupts, there is a fixed condition coming from the publishing peripheral. These IRQ's, when enabled, always trigger on the same peripheral condition. See each peripheral-specific section of this document for an explanation of the interrupt conditionals for each of these peripherals.

13.1.4.2 DMA Trigger Event Route (DMA_TRIG)

DMA_TRIGx paths rely on the DMA controller IRQ to indicate to the CPU that the DMA transfer is complete. For DMA_TRIG_RX and DMA_TRIG_TX paths, used to connect the UNICOMM modules and the DMA, there is an additional side-band signal that passes the DMA done condition back to the UNICOMM peripheral to indicate when a DMA activity has run to completion. This forgoes the need for the DMA IRQ to indicate the completion of transfer. An example of a DMA trigger route is shown in Figure 13-4.

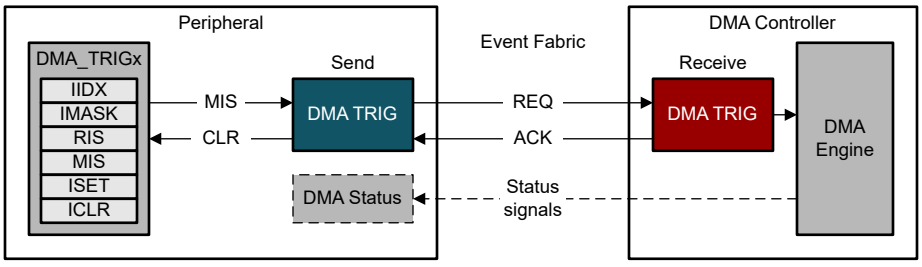


Figure 13-4. DMA Route

For each peripheral that is capable of generating a DMA trigger, a fixed route is provided from the peripheral to the DMA.

Standard Event Registers DMA_TRIGs

ADVANCE INFORMATION

In the case of peripheral publishers that have the standard Event Management Registers, the route goes from the masked interrupt status (MIS) register to the DMA's trigger management logic. When a trigger is received by the DMA, the DMA acknowledges the request and the peripheral clears the request. The DMA also acknowledges the cleared request, after which a new request can be asserted by the peripheral.

The DMA route can also contain status signals (for specific peripherals) to indicate to the triggering peripheral that a DMA transfer sequence has completed. For example, the DMA can be set up to transfer *N* number of bytes from an SRAM buffer into the UNICOMM-UART TXDATA register based on the DMA_TRIG_TX trigger. Upon each trigger from the UART, the DMA will acknowledge that the transfer was successful. On the *N*th byte, the DMA will send a complete status signal to the UART, which the UART can use to propagate a transfer completion interrupt to the CPU.

IP-Specific Event Registers DMA_TRIGs

For publishing peripherals that implement a unique register set for managing DMA triggers, there is also unique logic used to configure each trigger condition. See the peripheral-specific section of this document for guidance on how to configure DMA triggers in this case.

13.1.4.3 ADC Start Of Conversion Event Route (ADC_SOC)

An ADC SOC event route is a fixed, point-to-point connection between a peripheral event publisher and the ADC peripheral, used to trigger start of conversions on the ADC module.

Standard Event Registers DMA_TRIGs

In the case of peripheral publishers that have the standard Event Management Registers, the route goes from the masked interrupt status (MIS) register to the ADC's start of conversion logic. When a conversion trigger is received by the ADC, the ADC acknowledges the request and the peripheral clears the request. The ADC also acknowledges the cleared request, after which a new request can be asserted by the peripheral.

IP-Specific Event Registers DMA_TRIGs

For publishing peripherals that implement a unique register set for managing ADC conversions, there is also unique logic used to configure each trigger condition. See the peripheral-specific section of this document for guidance on how to configure ADC trigger converions in this case.

13.1.5 Event Propagation Latency

Event route channels for peripheral subscribers with the standard event registers implement a four-way hardware handshake between the publishing entity and the subscribing entity. This handshake requires four ULPCLK cycles to complete:

1. Request from publisher to subscriber
2. Acknowledge from subscriber to publisher
3. De-assert of request from publisher to subscriber
4. Acknowledge of de-assert from subscriber to publisher

If the publishing peripheral sends two requests and the first request has not cleared the handshake, the second request is dropped.

Note

For IP-specific and Fixed publisher paths, there is no such associated latency.

14.1 IOMUX

The IOMUX controls the configuration of all device pins with digital input-output (IO) functions, including: digital function selection, inversion control, drive strength (if applicable), the pullup or pulldown resistor (if applicable), and wake-up configuration (if applicable on certain IOs for wakeup from SHUTDOWN mode).

14.1.1 IOMUX Overview

The IOMUX manages the configuration of the digital IO pins through the Pin Control Management registers (PINCMx). Key functions configured by IOMUX include:

- Selection of which peripheral is multiplexed to each digital IO pin
- Digital Input Path Configuration
  - Input Path Enable/Disable
  - Input Signal Logic Inversion Control (Control bit shared with Output Logic Inversion)
- Digital Output Path Configuration
  - Drive Strength Control
  - Output Connection Enable/Disable
  - Output Signal Logic Inversion Control (Control bit shared with Input Logic Inversion)
- Wakeup Configuration (for Wakeup from SHUTDOWN mode on specified pins per table below)
  - Read wake up source from the WAKESTAT bit from the PINCMx register
  - Wake up compare level configuration to wake up on either a logic-low or logic-high
- Pull-up and Pull-down Resistor Control

14.1.1.1 IO Types and Analog Sharing

The IOMUX manages the selection of which peripheral function is to be used on a digital IO. IOMUX also provides the controls for the output driver, input path, and the wakeup logic for wakeup from SHUTDOWN mode.

Digital IO Types

There are several digital IO types which can be included on a given device. Each digital IO type supports different features. Digital IO Features by IO Type lists the features which are included with each IO type. See the device-specific data sheet for which IO type is used on a given package pin.

Table 14-1. Digital IO Features by IO Type

IO Structure	Inversion Control	Drive Strength Control	Pullup Resistor	Pulldown Resistor	Wakeup Logic
Standard-drive (SDIO)	Y		Y	Y	
Standard-drive with wake (SDIO)	Y		Y	Y	Y
High-drive (HDIO)	Y	Y	Y	Y	
High-speed (HSIO)	Y	Y	Y	Y	

Digital IO Shared with Analog Functions

Certain pins on a device are digital only and do not have any analog functions connected to the pin. Other pins can have one or more analog functions connected to the pin in addition to the digital IO functions. Analog functions are never selected within the IOMUX; these are always configured within of the respective analog peripheral. Analog peripherals have no knowledge of, or interaction with, the IOMUX.

In general, when analog functionality is used on a pin which also has digital functions, the IOMUX configuration for that pin must be left in the default (high-Z) state so as to not interfere with the proper operation of the analog function. However, it is possible to have the IOMUX active on a pin when an analog peripheral is also interacting with the pin, provided that the application software verifies there is not a conflict between the functions. For example, it is possible to have the pullup or pulldown resistor on an IO enabled at the same time that the ADC is running a conversion on the same IO. However, an invalid configuration of enabling the output driver on an IO at the same time that an analog peripheral is driving the IO (for example, a CMPSSx_DACL or PGAx output). This creates an IO conflict.

Application software is responsible for making sure that the IOMUX settings do not conflict with any analog peripheral functions which can be enabled on a shared pad.

IO Slice

The mixed-signal IO pin slice diagram for a full featured IO pin is shown in Superset IO Slice. Not all pins have analog functions, wake-up logic, drive strength control, and pullup or pulldown resistors available. See the Pin Configuration and Functions table within the device-specific data sheet for detailed information on what features and peripherals are supported on a specific pin. An example for a single pin is shown below in IOMUX Pin Example.

Table 14-2. IOMUX Pin Example

PINCMx	Pin Name	Analog [MuxMode]	Digital [MuxMode]	IO Structure
IOMUX_PA2	PA2	Unconnected	Unconnected	High-Drive
		ANALOG_AIN8 [ANALOG]	GPIO02 [1]	
		A0_3 [A0]	MCPWM4_2A [2]	
		CMP1_HN0 [A1]	MCPWM3_1B [3]	
		PGA0_OUT [A2]	MCPWM3_3A [4]	
			MCPWM4_3A [5]	
			MCPWM4_2B [6]	
			UC1_TX_SDA_PICO [7]	
			MCPWM3_1A [8]	
			UC0_TX_SDA_PICO [11]	
			OUTPUTXBAR5 [14]	

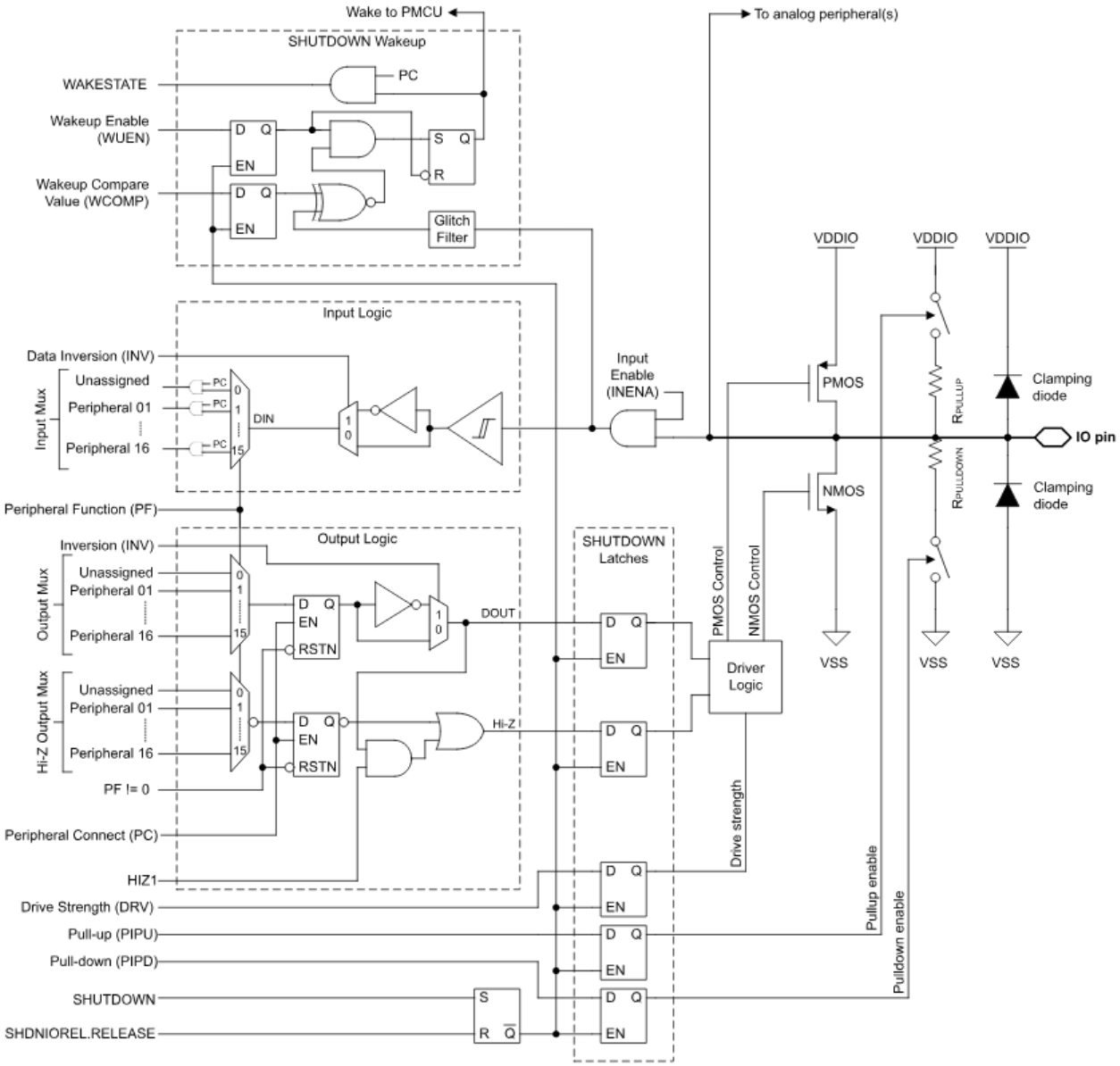


Figure 14-1. Superset IO Slice

Default IOMUX State

The initial state of the IOMUX pin slice for all digital IO after a BOOTRST is as follows:

- The digital IO is in a high-impedance state
- The peripheral function selection field (PINCMx.PF) is cleared such that no peripheral function is selected
- The digital IO output is disabled by clearing the peripheral connect (PINCMx.PC) bit
- The input enable (PINCMx.INENAx) bit is cleared (disabled)
- The inversion logic (PINCMx.INV) is disabled
- The Hi-Z output high (PINCMx.HIZ1) mode is disabled
- The pullup/pulldown (PINCMx.PIPU and PINCMx.PIPD) resistors are disabled
- The drive strength control (PINCMx.DRV) is reset
- The wakeup logic (PINCMx.WUEN) is disabled

Note

The JTAG/serial wire debug (JTAG/SWD) pins are a specific exception to the above default state. The debug pins are configured in SWD mode by default, and can be switched to an alternate setting after start-up.

14.1.2 IOMUX Operation

Each digital IO on a device has a dedicated 32-bit PINCM register in the IOMUX peripheral register space which is used to configure the digital functions of the respective IO. See the device specific data sheet for determining the PINCM register index which corresponds to the IO to be configured.

14.1.2.1 Peripheral Function (PF) Assignment

When setting up the initial IOMUX configuration for an IO after a BOOTRST, application software can select which digital peripheral from the supported options is to be connect to an IO by writing the appropriate peripheral select value to the PF field while simultaneously setting the PC and INENA bits in the PINCMx register corresponding to the targeted pin. The IOMUX configurations for a given peripheral must be set before the peripheral connected to the IOs has been initialized for operation.

To change the peripheral function selection for a digital IO at runtime after a peripheral function has already been configured for that IO, the following procedure must be followed:

1. Disable the currently connected peripheral function
2. Clear the PC bit (input/output connect bit) and INENA (input connect bit) in the corresponding PINCMx register
3. Write 0x0 to the PF field in the PINCMx to clear the logic in the data path
4. Select the new peripheral function by writing the peripheral function ID to the PF register
5. Set the PC and INENA bits in the PINCMx register to connect the newly selected peripheral
6. Enable the newly selected peripheral for operation

At runtime, the INENA bit can be used to mask the input from the IO to the peripheral, if desired. When INENA is cleared, a connected digital peripheral sees a logic low (0) from the IO, regardless of the external state of the IO. If an IO supports wakeup from SHUTDOWN mode, the INENA bit also controls propagation of the IO state to the SHUTDOWN mode wakeup logic.

If a peripheral is assigned to an IO, but the peripheral is in a disabled state, the last valid output conditions (output logic level and Hi-Z state) are latched in the IOMUX output logic. When the peripheral is enabled, the IOMUX releases the latched state to allow the (now enabled) peripheral's output state to propagate to the IO. The PMCU indicates to the IOMUX when a peripheral is entering a disabled state via the IORET signal, which is combined with the PC signal via a logic OR to control the output state latches. This mechanism handles preservation of the last valid output state of peripherals in power domain 1 (PD1) when entering STOP or STANDBY mode, as PD1 peripherals are always temporarily disabled upon entry to STOP/STANDBY, and re-enabled upon exit from STOP/STANDBY modes.

When no peripheral function is selected (PF==0) the output latches are put into a reset state, causing the output NMOS and PMOS to be disabled (leaving the IO pin in a Hi-Z state with the exception of any enabled pullup/pulldown resistors). Note that the pullup/pulldown resistors are never controlled by either a connected peripheral or the peripheral muxing logic. These resistors are only controlled by the IOMUX control bits.

14.1.2.2 Logic High to Hi-Z Conversion

The IOMUX supports translating an output high signal from a connected peripheral into a Hi-Z output state at the IO pin. This functionality is particularly useful for open-drain digital input/output applications. When this functionality is enabled, the IO pin state as a function of the peripheral output is as shown in Table 14-3.

Table 14-3. Logic High to Hi-Z Truth Table

Connected Peripheral Output	IO Pin State (Z1 = 0x0)	IO Pin State (Z1 = 0x1)
Logic low (0)	Output low	Output low

ADVANCE INFORMATION

Table 14-3. Logic High to Hi-Z Truth Table (continued)

Connected Peripheral Output	IO Pin State (Z1 = 0x0)	IO Pin State (Z1 = 0x1)
Logic high (1)	Output high	High impedance (Hi-Z)

To enable logic high to Hi-Z conversion on a digital IO, set the Z1 bit in the corresponding PINCMx register.

14.1.2.3 Logic Inversion

The IOMUX supports logic inversion of the digital input/output path. Logic inversion is useful for scenarios where opposite polarity is required for UART functions or SPI chip select functions.

To enable logic inversion on a digital IO, set the INV bit in the corresponding PINCMx register. To disable logic inversion, clear the corresponding bit. Logic inversion is disabled by default.

14.1.2.4 SHUTDOWN Mode Wakeup Logic

In SHUTDOWN mode, the entire regulated core supply of the device is disabled and the device wakes only from a wake-capable IO that is configured for wakeup, from NRST, or from a SWD debug connection. The IO wake mechanism for exiting SHUTDOWN is managed by IOMUX and is level based. Specific standard-drive IOs, include the additional wakeup logic that can be used to wake the device from a SHUTDOWN operating mode upon a level match.

To configure a wake-capable IO for wakeup from SHUTDOWN mode:

1. Set the INENA bit to let the input state propagate from the IO to the wakeup logic.
2. Select the compare level to use for wake by setting or clearing the WCOMP bit in the PINCMx register corresponding to the targeted pin.
3. Enable wakeup by setting the WUEN bit in the PINCMx register corresponding to the targeted pin.

After the previous configuration, SHUTDOWN mode can be entered through the appropriate command in SYSCTL. Pins on the device that contain digital IO controlled by IOMUX retain the current state when the device enters into SHUTDOWN. While the digital IO state is latched upon entry into SHUTDOWN mode, the IOMUX configuration registers (all PINCMx registers) lose the contents as the regulated core supply is shut down.

After SHUTDOWN is entered, a level match on any pin configured for wakeup triggers the exit sequence from SHUTDOWN. When the device exits SHUTDOWN, a BOR-level reset occurs but the state of the digital IO remains latched through the reset, keeping the IO state present upon entry into SHUTDOWN. This state is held until the IO are released in SYSCTL. After the BOR, SYSCTL captures the cause of the reset as a SHUTDOWN exit so that software can identify this and take appropriate action to reconfigure the device.

If multiple pins are configured for wakeup from SHUTDOWN, application software can determine which wakeup-configured IO generated the wake by polling the WAKESTATE bit in any IOs enabled for wake before the SHUTDOWN exit.

Application software must apply the following process to restore the IO state upon exit from SHUTDOWN:

1. Check which IO triggered the wakeup from SHUTDOWN, if necessary, as follows:
   a. Reconfigure the PINCMx register corresponding to the IO to be tested for wakeup status and set the peripheral connect (PC) bit (the PC bit gates the WAKESTATE indication).
   b. Test the WAKESTATE bit in the PINCMx register corresponding to the IO to be tested to determine if that particular IO received a WAKE status based on the previously configured WCOMP and WUEN configuration.
2. Reconfigure any remaining IOMUX PINCM registers to the correct states.
3. Reconfigure the peripherals that are connected to pins through IOMUX, and enable them.
4. Release the SHUTDOWN IO lock in SYSCTL.
5. Clear the WUEN bit in the PINCMx register to reset the WAKESTATE status.

Note

After waking from SHUTDOWN, if the WUEN bit is not cleared and the shutdown release bit in SYSCTL is not set, then reentering SHUTDOWN results in an immediate wake event, because the WAKESTATE status has not been cleared from the previous wake event.

14.1.2.5 Pullup/Pulldown Resistors

Programmable pullup/pulldown resistors are provided on most digital IO types, and are connected to VDD/VSS, respectively.

To enable the pullup or pulldown resistor on a digital IO, set the PIPU or PIPD bit, respectively, in the corresponding PINCMx register. To disable the pullup or pulldown resistor, clear the corresponding bit.

The pullup/pulldown resistors can be enabled at any time, and the configuration is independent from the peripheral function configuration. Enable a pullup/pulldown resistor while changing the selected peripheral function is possible.

Note

If both pull resistors are configured at the same time, the pullup resistor takes priority.

14.1.2.6 Drive Strength Control

The high-drive and high-speed digital IO types have programmable drive strength (low drive and high drive). The default drive strength is low drive. Application software can request high drive by setting the DRV bit in the PINCMx register corresponding to the target digital IO. Drive strength control is not available for standard drive IO type.

The drive strength control is completely independent of the selected peripheral function (PF) and can be changed by application software at any time.

For detailed electrical specifications on the drive performance in each drive mode for a given IO, see the Digital IO parameters in the device-specific data sheet.

14.2 IOMUX Registers

This Section describes the IOMUX Registers.

14.2.1 IOMUX Base Address Table

Table 14-4. IOMUX Base Address Table

Bit Field Name		DriverLib Name	Base Address
Instance	Structure		
IomuxRegs	IOMUX_REGS	IOMUX	0x400C_C000

ADVANCE INFORMATION

14.2.2 IOMUX_REGS Registers

Table 14-5 lists the memory-mapped registers for the IOMUX_REGS registers. All register offset addresses not listed in Table 14-5 should be considered as reserved locations and the register contents should not be modified.

Table 14-5. IOMUX_REGS Registers

Offset	Acronym	Register Name	Section
0h	PINCM0	Pin Control Management Register	Go
4h	PINCM1	Pin Control Management Register	Go
8h	PINCM2	Pin Control Management Register	Go
Ch	PINCM3	Pin Control Management Register	Go
10h	PINCM4	Pin Control Management Register	Go
14h	PINCM5	Pin Control Management Register	Go
18h	PINCM6	Pin Control Management Register	Go
1Ch	PINCM7	Pin Control Management Register	Go
20h	PINCM8	Pin Control Management Register	Go
24h	PINCM9	Pin Control Management Register	Go
28h	PINCM10	Pin Control Management Register	Go
2Ch	PINCM11	Pin Control Management Register	Go
30h	PINCM12	Pin Control Management Register	Go
34h	PINCM13	Pin Control Management Register	Go
38h	PINCM14	Pin Control Management Register	Go
3Ch	PINCM15	Pin Control Management Register	Go
40h	PINCM16	Pin Control Management Register	Go
44h	PINCM17	Pin Control Management Register	Go
48h	PINCM18	Pin Control Management Register	Go
4Ch	PINCM19	Pin Control Management Register	Go
50h	PINCM20	Pin Control Management Register	Go
54h	PINCM21	Pin Control Management Register	Go
58h	PINCM22	Pin Control Management Register	Go
5Ch	PINCM23	Pin Control Management Register	Go
60h	PINCM24	Pin Control Management Register	Go
64h	PINCM25	Pin Control Management Register	Go
68h	PINCM26	Pin Control Management Register	Go
6Ch	PINCM27	Pin Control Management Register	Go
70h	PINCM28	Pin Control Management Register	Go
74h	PINCM29	Pin Control Management Register	Go
78h	PINCM30	Pin Control Management Register	Go
7Ch	PINCM31	Pin Control Management Register	Go
80h	PINCM32	Pin Control Management Register	Go
84h	PINCM33	Pin Control Management Register	Go
88h	PINCM34	Pin Control Management Register	Go
8Ch	PINCM35	Pin Control Management Register	Go
90h	PINCM36	Pin Control Management Register	Go
94h	PINCM37	Pin Control Management Register	Go
98h	PINCM38	Pin Control Management Register	Go
9Ch	PINCM39	Pin Control Management Register	Go
A0h	PINCM40	Pin Control Management Register	Go

Table 14-5. IOMUX_REGS Registers (continued)

Offset	Acronym	Register Name	Section
A4h	PINCM41	Pin Control Management Register	Go
A8h	PINCM42	Pin Control Management Register	Go
ACh	PINCM43	Pin Control Management Register	Go
B0h	PINCM44	Pin Control Management Register	Go
B4h	PINCM45	Pin Control Management Register	Go
B8h	PINCM46	Pin Control Management Register	Go
BCh	PINCM47	Pin Control Management Register	Go
C0h	PINCM48	Pin Control Management Register	Go
C4h	PINCM49	Pin Control Management Register	Go
C8h	PINCM50	Pin Control Management Register	Go
CCh	PINCM51	Pin Control Management Register	Go
D0h	PINCM52	Pin Control Management Register	Go
D4h	PINCM53	Pin Control Management Register	Go
D8h	PINCM54	Pin Control Management Register	Go
DCh	PINCM55	Pin Control Management Register	Go
E0h	PINCM56	Pin Control Management Register	Go
E4h	PINCM57	Pin Control Management Register	Go
E8h	PINCM58	Pin Control Management Register	Go
ECh	PINCM59	Pin Control Management Register	Go
F0h	PINCM60	Pin Control Management Register	Go
F4h	PINCM61	Pin Control Management Register	Go
F8h	PINCM62	Pin Control Management Register	Go
FCh	PINCM63	Pin Control Management Register	Go
100h	PINCM64	Pin Control Management Register	Go
104h	PINCM65	Pin Control Management Register	Go
108h	PINCM66	Pin Control Management Register	Go
10Ch	PINCM67	Pin Control Management Register	Go
110h	PINCM68	Pin Control Management Register	Go
114h	PINCM69	Pin Control Management Register	Go
118h	PINCM70	Pin Control Management Register	Go
11Ch	PINCM71	Pin Control Management Register	Go
120h	PINCM72	Pin Control Management Register	Go
124h	PINCM73	Pin Control Management Register	Go
128h	PINCM74	Pin Control Management Register	Go
12Ch	PINCM75	Pin Control Management Register	Go
130h	PINCM76	Pin Control Management Register	Go
134h	PINCM77	Pin Control Management Register	Go
138h	PINCM78	Pin Control Management Register	Go
13Ch	PINCM79	Pin Control Management Register	Go
140h	PINCM80	Pin Control Management Register	Go
144h	PINCM81	Pin Control Management Register	Go
148h	PINCM82	Pin Control Management Register	Go
14Ch	PINCM83	Pin Control Management Register	Go
150h	PINCM84	Pin Control Management Register	Go
154h	PINCM85	Pin Control Management Register	Go

Table 14-5. IOMUX_REGS Registers (continued)

Offset	Acronym	Register Name	Section
158h	PINCM86	Pin Control Management Register	Go
15Ch	PINCM87	Pin Control Management Register	Go
160h	PINCM88	Pin Control Management Register	Go
164h	PINCM89	Pin Control Management Register	Go
168h	PINCM90	Pin Control Management Register	Go
16Ch	PINCM91	Pin Control Management Register	Go
170h	PINCM92	Pin Control Management Register	Go
174h	PINCM93	Pin Control Management Register	Go
178h	PINCM94	Pin Control Management Register	Go
17Ch	PINCM95	Pin Control Management Register	Go
180h	PINCM96	Pin Control Management Register	Go
184h	PINCM97	Pin Control Management Register	Go
188h	PINCM98	Pin Control Management Register	Go
18Ch	PINCM99	Pin Control Management Register	Go
190h	PINCM100	Pin Control Management Register	Go
194h	PINCM101	Pin Control Management Register	Go
198h	PINCM102	Pin Control Management Register	Go
19Ch	PINCM103	Pin Control Management Register	Go
1A0h	PINCM104	Pin Control Management Register	Go
1A4h	PINCM105	Pin Control Management Register	Go
1ACh	PINCM107	Pin Control Management Register	Go

Complex bit access types are encoded to fit into small table cells. Table 14-6 shows the codes that are used for access types in this section.

Table 14-6. IOMUX_REGS Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
RH	R H	Read Set or cleared by hardware
Write Type		
W	W	Write
Reset or Default Value		
-n		Value after reset or the default value

1 PINCM0 Register (Offset = 0h) [Reset = 0XXXXXX0h]

PINCM0 is shown in Figure 14-2 and described in Table 14-7.

Return to the Summary Table.

Pin Control Management Register

Figure 14-2. PINCM0 Register

31	30	29	28	27	26	25	24
RESERVED			WCOMP	WUEN	INV	Z1	RESERVED
R/W-0h			RH/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-Xh
23	22	21	20	19	18	17	16
RESERVED					INENA	PIPU	PIPD
R/W-Xh					R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED		WAKESTATE	RESERVED				
R/W-Xh		R-0h	R/W-Xh				
7	6	5	4	3	2	1	0
PC	RESERVED		PF				
R/W-0h	R/W-Xh		R/W-0h				

Table 14-8. PINCM0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R/W	0h	
28	WCOMP	RH/W	0h	Wakeup Compare Value bit. This bitfield indicates on what data match this IOCELL with intiate a Wake from SHUTDOWN when WUEN = 1. Reset type: SYSRST 0h = 0 1h = 1
27	WUEN	R/W	0h	Wakeup Enable bit. This bitfield indicates whether or not the Wake from Shutdown is enabled on this IOCELL. Reset type: SYSRST 0h = 0 1h = 1
26	INV	R/W	0h	Data Inversion Control Selection. This bitfield controls whether the input and output logic of this IOCELL is inverted. Reset type: SYSRST 0h = 0 1h = 1
25	Z1	R/W	0h	HI-Z Output State Control Selection. This bitfield controls whether or not a high output value will tri-state the output (open-drain functionality). Reset type: SYSRST 0h = 0 1h = 1
24-19	RESERVED	R/W	Xh	

Table 14-8. PINCM0 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
18	INENA	R/W	0h	Input Enable Control Selection. This bitfield controls whether or not the input to this IOCELL propagates to the connected peripheral function. Reset type: SYSRST 0h = 0 1h = 1
17	PIPU	R/W	0h	Pull Up Resistor Control. This bitfield controls whether the pull up resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
16	PIPD	R/W	0h	Pull Down Resistor Control. This bitfield controls whether the pull down resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
15-14	RESERVED	R/W	Xh	
13	WAKESTATE	R	0h	This bitfield has the IOPAD WAKEUP signal as a status bit Reset type: SYSRST
12-8	RESERVED	R/W	Xh	
7	PC	R/W	0h	Peripheral Connect bit. This bit indicates if the designated peripheral function is "Connected" Reset type: SYSRST 0h = 0 1h = 1
6-5	RESERVED	R/W	Xh	
4-0	PF	R/W	0h	Peripheral Function selection bits. This bitfield indicates what peripheral function is selected for a specific pin. Reset type: SYSRST 0h = 0 1h = gpio0_DIO00 2h = mcpwm4_o_pwma1_mscbit 3h = mcpwm3_o_pwma1_mscbit 4h = gptimer32b2cc0_CCP0 5h = gptimer16b2cclp0_CCP0 6h = spgss1_U2_SCL_RX 7h = spgss1_U1_PICO_SDA_TX 8h = spgss0_U1_CS0_CTS 9h = spgss0_U1_PICO_SDA_TX Ah = mcpwm4_o_pwma2_mscbit Bh = spgss0_U2_SCL_RX Dh = spgss0_U0_CS0_CTS 10h = xbarwrapper0_o_output_xbar_output8

2 PINCM1 Register (Offset = 4h) [Reset = 0XXXXXX0h]

PINCM1 is shown in Figure 14-3 and described in Table 14-8.

Return to the Summary Table.

Pin Control Management Register

Figure 14-3. PINCM1 Register

31	30	29	28	27	26	25	24
RESERVED			WCOMP	WUEN	INV	Z1	RESERVED
R/W-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-Xh
23	22	21	20	19	18	17	16
RESERVED					INENA	PIPU	PIPD
R/W-Xh					R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED		WAKESTATE	RESERVED				
R/W-Xh		R-0h	R/W-Xh				
7	6	5	4	3	2	1	0
PC	RESERVED		PF				
R/W-0h	R/W-Xh		R/W-0h				

Table 14-10. PINCM1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R/W	0h	
28	WCOMP	R/W	0h	Wakeup Compare Value bit. This bitfield indicates on what data match this IOCELL with intiate a Wake from SHUTDOWN when WUEN = 1. Reset type: SYSRST 0h = 0 1h = 1
27	WUEN	R/W	0h	Wakeup Enable bit. This bitfield indicates whether or not the Wake from Shutdown is enabled on this IOCELL. Reset type: SYSRST 0h = 0 1h = 1
26	INV	R/W	0h	Data Inversion Control Selection. This bitfield controls whether the input and output logic of this IOCELL is inverted. Reset type: SYSRST 0h = 0 1h = 1
25	Z1	R/W	0h	HI-Z Output State Control Selection. This bitfield controls whether or not a high output value will tri-state the output (open-drain functionality). Reset type: SYSRST 0h = 0 1h = 1
24-19	RESERVED	R/W	Xh	

Table 14-10. PINCM1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
18	INENA	R/W	0h	Input Enable Control Selection. This bitfield controls whether or not the input to this IOCELL propagates to the connected peripheral function. Reset type: SYSRST 0h = 0 1h = 1
17	PIPU	R/W	0h	Pull Up Resistor Control. This bitfield controls whether the pull up resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
16	PIPD	R/W	0h	Pull Down Resistor Control. This bitfield controls whether the pull down resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
15-14	RESERVED	R/W	Xh	
13	WAKESTATE	R	0h	This bitfield has the IOPAD WAKEUP signal as a status bit Reset type: SYSRST
12-8	RESERVED	R/W	Xh	
7	PC	R/W	0h	Peripheral Connect bit. This bit indicates if the designated peripheral function is "Connected" Reset type: SYSRST 0h = 0 1h = 1
6-5	RESERVED	R/W	Xh	
4-0	PF	R/W	0h	Peripheral Function selection bits. This bitfield indicates what peripheral function is selected for a specific pin. Reset type: SYSRST 0h = 0 1h = 1 2h = 2 3h = 3 4h = 4 5h = 5 6h = 6 7h = 7 8h = 8 9h = 9 Ah = 10 Bh = 11 Ch = 12 Dh = 13 10h = 16

3 PINCM2 Register (Offset = 8h) [Reset = 0XXXXXX0h]

PINCM2 is shown in Figure 14-4 and described in Table 14-9.

Return to the Summary Table.

Pin Control Management Register

Figure 14-4. PINCM2 Register

31	30	29	28	27	26	25	24
RESERVED			WCOMP	WUEN	INV	Z1	RESERVED
R/W-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-Xh
23	22	21	20	19	18	17	16
RESERVED					INENA	PIPU	PIPD
R/W-Xh					R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED		WAKESTATE	RESERVED				
R/W-Xh		R-0h	R/W-Xh				
7	6	5	4	3	2	1	0
PC	RESERVED		PF				
R/W-0h	R/W-Xh		R/W-0h				

Table 14-12. PINCM2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R/W	0h	
28	WCOMP	R/W	0h	Wakeup Compare Value bit. This bitfield indicates on what data match this IOCELL with intiate a Wake from SHUTDOWN when WUEN = 1. Reset type: SYSRST 0h = 0 1h = 1
27	WUEN	R/W	0h	Wakeup Enable bit. This bitfield indicates whether or not the Wake from Shutdown is enabled on this IOCELL. Reset type: SYSRST 0h = 0 1h = 1
26	INV	R/W	0h	Data Inversion Control Selection. This bitfield controls whether the input and output logic of this IOCELL is inverted. Reset type: SYSRST 0h = 0 1h = 1
25	Z1	R/W	0h	HI-Z Output State Control Selection. This bitfield controls whether or not a high output value will tri-state the output (open-drain functionality). Reset type: SYSRST 0h = 0 1h = 1
24-19	RESERVED	R/W	Xh	

Table 14-12. PINCM2 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
18	INENA	R/W	0h	Input Enable Control Selection. This bitfield controls whether or not the input to this IOCELL propagates to the connected peripheral function. Reset type: SYSRST 0h = 0 1h = 1
17	PIPU	R/W	0h	Pull Up Resistor Control. This bitfield controls whether the pull up resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
16	PIPD	R/W	0h	Pull Down Resistor Control. This bitfield controls whether the pull down resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
15-14	RESERVED	R/W	Xh	
13	WAKESTATE	R	0h	This bitfield has the IOPAD WAKEUP signal as a status bit Reset type: SYSRST
12-8	RESERVED	R/W	Xh	
7	PC	R/W	0h	Peripheral Connect bit. This bit indicates if the designated peripheral function is "Connected" Reset type: SYSRST 0h = 0 1h = 1
6-5	RESERVED	R/W	Xh	
4-0	PF	R/W	0h	Peripheral Function selection bits. This bitfield indicates what peripheral function is selected for a specific pin. Reset type: SYSRST 0h = 0 1h = 1 2h = 2 3h = 3 4h = 4 5h = 5 6h = 6 8h = 8 Ah = 10 Dh = 13 10h = 16

4 PINCM3 Register (Offset = Ch) [Reset = 0XXXXXX0h]

PINCM3 is shown in Figure 14-5 and described in Table 14-10.

Return to the Summary Table.

Pin Control Management Register

Figure 14-5. PINCM3 Register

31	30	29	28	27	26	25	24
RESERVED			WCOMP	WUEN	INV	Z1	RESERVED
R/W-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-Xh
23	22	21	20	19	18	17	16
RESERVED					INENA	PIPU	PIPD
R/W-Xh					R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED		WAKESTATE	RESERVED				
R/W-Xh		R-0h	R/W-Xh				
7	6	5	4	3	2	1	0
PC	RESERVED		PF				
R/W-0h	R/W-Xh		R/W-0h				

Table 14-14. PINCM3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R/W	0h	
28	WCOMP	R/W	0h	Wakeup Compare Value bit. This bitfield indicates on what data match this IOCELL with intiate a Wake from SHUTDOWN when WUEN = 1. Reset type: SYSRST 0h = 0 1h = 1
27	WUEN	R/W	0h	Wakeup Enable bit. This bitfield indicates whether or not the Wake from Shutdown is enabled on this IOCELL. Reset type: SYSRST 0h = 0 1h = 1
26	INV	R/W	0h	Data Inversion Control Selection. This bitfield controls whether the input and output logic of this IOCELL is inverted. Reset type: SYSRST 0h = 0 1h = 1
25	Z1	R/W	0h	HI-Z Output State Control Selection. This bitfield controls whether or not a high output value will tri-state the output (open-drain functionality). Reset type: SYSRST 0h = 0 1h = 1
24-19	RESERVED	R/W	Xh	

Table 14-14. PINCM3 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
18	INENA	R/W	0h	Input Enable Control Selection. This bitfield controls whether or not the input to this IOCELL propagates to the connected peripheral function. Reset type: SYSRST 0h = 0 1h = 1
17	PIPU	R/W	0h	Pull Up Resistor Control. This bitfield controls whether the pull up resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
16	PIPD	R/W	0h	Pull Down Resistor Control. This bitfield controls whether the pull down resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
15-14	RESERVED	R/W	Xh	
13	WAKESTATE	R	0h	This bitfield has the IOPAD WAKEUP signal as a status bit Reset type: SYSRST
12-8	RESERVED	R/W	Xh	
7	PC	R/W	0h	Peripheral Connect bit. This bit indicates if the designated peripheral function is "Connected" Reset type: SYSRST 0h = 0 1h = 1
6-5	RESERVED	R/W	Xh	
4-0	PF	R/W	0h	Peripheral Function selection bits. This bitfield indicates what peripheral function is selected for a specific pin. Reset type: SYSRST 0h = 0 1h = 1 2h = 2 3h = 3 4h = 4 5h = 5 6h = 6 7h = 7 8h = 8 Ah = 10 Dh = 13 10h = 16

5 PINCM4 Register (Offset = 10h) [Reset = 0XXXXXX0h]

PINCM4 is shown in Figure 14-6 and described in Table 14-11.

Return to the Summary Table.

Pin Control Management Register

Figure 14-6. PINCM4 Register

31	30	29	28	27	26	25	24
RESERVED			WCOMP	WUEN	INV	Z1	RESERVED
R/W-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-Xh
23	22	21	20	19	18	17	16
RESERVED					INENA	PIPU	PIPD
R/W-Xh					R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED		WAKESTATE	RESERVED				
R/W-Xh		R-0h	R/W-Xh				
7	6	5	4	3	2	1	0
PC	RESERVED		PF				
R/W-0h	R/W-Xh		R/W-0h				

Table 14-16. PINCM4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R/W	0h	
28	WCOMP	R/W	0h	Wakeup Compare Value bit. This bitfield indicates on what data match this IOCELL with intiate a Wake from SHUTDOWN when WUEN = 1. Reset type: SYSRST 0h = 0 1h = 1
27	WUEN	R/W	0h	Wakeup Enable bit. This bitfield indicates whether or not the Wake from Shutdown is enabled on this IOCELL. Reset type: SYSRST 0h = 0 1h = 1
26	INV	R/W	0h	Data Inversion Control Selection. This bitfield controls whether the input and output logic of this IOCELL is inverted. Reset type: SYSRST 0h = 0 1h = 1
25	Z1	R/W	0h	HI-Z Output State Control Selection. This bitfield controls whether or not a high output value will tri-state the output (open-drain functionality). Reset type: SYSRST 0h = 0 1h = 1
24-19	RESERVED	R/W	Xh	

Table 14-16. PINCM4 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
18	INENA	R/W	0h	Input Enable Control Selection. This bitfield controls whether or not the input to this IOCELL propagates to the connected peripheral function. Reset type: SYSRST 0h = 0 1h = 1
17	PIPU	R/W	0h	Pull Up Resistor Control. This bitfield controls whether the pull up resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
16	PIPD	R/W	0h	Pull Down Resistor Control. This bitfield controls whether the pull down resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
15-14	RESERVED	R/W	Xh	
13	WAKESTATE	R	0h	This bitfield has the IOPAD WAKEUP signal as a status bit Reset type: SYSRST
12-8	RESERVED	R/W	Xh	
7	PC	R/W	0h	Peripheral Connect bit. This bit indicates if the designated peripheral function is "Connected" Reset type: SYSRST 0h = 0 1h = 1
6-5	RESERVED	R/W	Xh	
4-0	PF	R/W	0h	Peripheral Function selection bits. This bitfield indicates what peripheral function is selected for a specific pin. Reset type: SYSRST 0h = 0 1h = 1 2h = 2 3h = 3 4h = 4 5h = 5 6h = 6 7h = 7 9h = 9 Bh = 11 10h = 16

6 PINCM5 Register (Offset = 14h) [Reset = 0XXXXXX0h]

PINCM5 is shown in Figure 14-7 and described in Table 14-12.

Return to the Summary Table.

Pin Control Management Register

Figure 14-7. PINCM5 Register

31	30	29	28	27	26	25	24
RESERVED			WCOMP	WUEN	INV	Z1	RESERVED
R/W-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-Xh
23	22	21	20	19	18	17	16
RESERVED					INENA	PIPU	PIPD
R/W-Xh					R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED		WAKESTATE	RESERVED				
R/W-Xh		R-0h	R/W-Xh				
7	6	5	4	3	2	1	0
PC	RESERVED		PF				
R/W-0h	R/W-Xh		R/W-0h				

Table 14-18. PINCM5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R/W	0h	
28	WCOMP	R/W	0h	Wakeup Compare Value bit. This bitfield indicates on what data match this IOCELL with intiate a Wake from SHUTDOWN when WUEN = 1. Reset type: SYSRST 0h = 0 1h = 1
27	WUEN	R/W	0h	Wakeup Enable bit. This bitfield indicates whether or not the Wake from Shutdown is enabled on this IOCELL. Reset type: SYSRST 0h = 0 1h = 1
26	INV	R/W	0h	Data Inversion Control Selection. This bitfield controls whether the input and output logic of this IOCELL is inverted. Reset type: SYSRST 0h = 0 1h = 1
25	Z1	R/W	0h	HI-Z Output State Control Selection. This bitfield controls whether or not a high output value will tri-state the output (open-drain functionality). Reset type: SYSRST 0h = 0 1h = 1
24-19	RESERVED	R/W	Xh	

Table 14-18. PINCM5 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
18	INENA	R/W	0h	Input Enable Control Selection. This bitfield controls whether or not the input to this IOCELL propagates to the connected peripheral function. Reset type: SYSRST 0h = 0 1h = 1
17	PIPU	R/W	0h	Pull Up Resistor Control. This bitfield controls whether the pull up resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
16	PIPD	R/W	0h	Pull Down Resistor Control. This bitfield controls whether the pull down resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
15-14	RESERVED	R/W	Xh	
13	WAKESTATE	R	0h	This bitfield has the IOPAD WAKEUP signal as a status bit Reset type: SYSRST
12-8	RESERVED	R/W	Xh	
7	PC	R/W	0h	Peripheral Connect bit. This bit indicates if the designated peripheral function is "Connected" Reset type: SYSRST 0h = 0 1h = 1
6-5	RESERVED	R/W	Xh	
4-0	PF	R/W	0h	Peripheral Function selection bits. This bitfield indicates what peripheral function is selected for a specific pin. Reset type: SYSRST 0h = 0 1h = 1 2h = 2 3h = 3 4h = 4 6h = 6 9h = 9 10h = 16

7 PINCM6 Register (Offset = 18h) [Reset = 0XXXXXX0h]

PINCM6 is shown in Figure 14-8 and described in Table 14-13.

Return to the Summary Table.

Pin Control Management Register

Figure 14-8. PINCM6 Register

31	30	29	28	27	26	25	24
RESERVED			WCOMP	WUEN	INV	Z1	RESERVED
R/W-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-Xh
23	22	21	20	19	18	17	16
RESERVED					INENA	PIPU	PIPD
R/W-Xh					R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED		WAKESTATE	RESERVED				
R/W-Xh		R-0h	R/W-Xh				
7	6	5	4	3	2	1	0
PC	RESERVED		PF				
R/W-0h	R/W-Xh		R/W-0h				

Table 14-20. PINCM6 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R/W	0h	
28	WCOMP	R/W	0h	Wakeup Compare Value bit. This bitfield indicates on what data match this IOCELL with intiate a Wake from SHUTDOWN when WUEN = 1. Reset type: SYSRST 0h = 0 1h = 1
27	WUEN	R/W	0h	Wakeup Enable bit. This bitfield indicates whether or not the Wake from Shutdown is enabled on this IOCELL. Reset type: SYSRST 0h = 0 1h = 1
26	INV	R/W	0h	Data Inversion Control Selection. This bitfield controls whether the input and output logic of this IOCELL is inverted. Reset type: SYSRST 0h = 0 1h = 1
25	Z1	R/W	0h	HI-Z Output State Control Selection. This bitfield controls whether or not a high output value will tri-state the output (open-drain functionality). Reset type: SYSRST 0h = 0 1h = 1
24-19	RESERVED	R/W	Xh	

Table 14-20. PINCM6 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
18	INENA	R/W	0h	Input Enable Control Selection. This bitfield controls whether or not the input to this IOCELL propagates to the connected peripheral function. Reset type: SYSRST 0h = 0 1h = 1
17	PIPU	R/W	0h	Pull Up Resistor Control. This bitfield controls whether the pull up resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
16	PIPD	R/W	0h	Pull Down Resistor Control. This bitfield controls whether the pull down resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
15-14	RESERVED	R/W	Xh	
13	WAKESTATE	R	0h	This bitfield has the IOPAD WAKEUP signal as a status bit Reset type: SYSRST
12-8	RESERVED	R/W	Xh	
7	PC	R/W	0h	Peripheral Connect bit. This bit indicates if the designated peripheral function is "Connected" Reset type: SYSRST 0h = 0 1h = 1
6-5	RESERVED	R/W	Xh	
4-0	PF	R/W	0h	Peripheral Function selection bits. This bitfield indicates what peripheral function is selected for a specific pin. Reset type: SYSRST 0h = 0 1h = 1 2h = 2 3h = 3 4h = 4 6h = 6 9h = 9 Bh = 11 Dh = 13 10h = 16

8 PINCM7 Register (Offset = 1Ch) [Reset = 0XXXXXX0h]

PINCM7 is shown in Figure 14-9 and described in Table 14-14.

Return to the Summary Table.

Pin Control Management Register

Figure 14-9. PINCM7 Register

31	30	29	28	27	26	25	24
RESERVED			WCOMP	WUEN	INV	Z1	RESERVED
R/W-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-Xh
23	22	21	20	19	18	17	16
RESERVED					INENA	PIPU	PIPD
R/W-Xh					R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED		WAKESTATE	RESERVED				
R/W-Xh		R-0h	R/W-Xh				
7	6	5	4	3	2	1	0
PC	RESERVED		PF				
R/W-0h	R/W-Xh		R/W-0h				

Table 14-22. PINCM7 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R/W	0h	
28	WCOMP	R/W	0h	Wakeup Compare Value bit. This bitfield indicates on what data match this IOCELL with intiate a Wake from SHUTDOWN when WUEN = 1. Reset type: SYSRST 0h = 0 1h = 1
27	WUEN	R/W	0h	Wakeup Enable bit. This bitfield indicates whether or not the Wake from Shutdown is enabled on this IOCELL. Reset type: SYSRST 0h = 0 1h = 1
26	INV	R/W	0h	Data Inversion Control Selection. This bitfield controls whether the input and output logic of this IOCELL is inverted. Reset type: SYSRST 0h = 0 1h = 1
25	Z1	R/W	0h	HI-Z Output State Control Selection. This bitfield controls whether or not a high output value will tri-state the output (open-drain functionality). Reset type: SYSRST 0h = 0 1h = 1
24-19	RESERVED	R/W	Xh	

Table 14-22. PINCM7 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
18	INENA	R/W	0h	Input Enable Control Selection. This bitfield controls whether or not the input to this IOCELL propagates to the connected peripheral function. Reset type: SYSRST 0h = 0 1h = 1
17	PIPU	R/W	0h	Pull Up Resistor Control. This bitfield controls whether the pull up resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
16	PIPD	R/W	0h	Pull Down Resistor Control. This bitfield controls whether the pull down resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
15-14	RESERVED	R/W	Xh	
13	WAKESTATE	R	0h	This bitfield has the IOPAD WAKEUP signal as a status bit Reset type: SYSRST
12-8	RESERVED	R/W	Xh	
7	PC	R/W	0h	Peripheral Connect bit. This bit indicates if the designated peripheral function is "Connected" Reset type: SYSRST 0h = 0 1h = 1
6-5	RESERVED	R/W	Xh	
4-0	PF	R/W	0h	Peripheral Function selection bits. This bitfield indicates what peripheral function is selected for a specific pin. Reset type: SYSRST 0h = 0 1h = 1 2h = 2 3h = 3 4h = 4 5h = 5 6h = 6 7h = 7 8h = 8 9h = 9 Ah = 10 Bh = 11 10h = 16

9 PINCM8 Register (Offset = 20h) [Reset = 0XXXXXX0h]

PINCM8 is shown in Figure 14-10 and described in Table 14-15.

Return to the Summary Table.

Pin Control Management Register

Figure 14-10. PINCM8 Register

31	30	29	28	27	26	25	24
RESERVED			WCOMP	WUEN	INV	Z1	RESERVED
R/W-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-Xh
23	22	21	20	19	18	17	16
RESERVED					INENA	PIPU	PIPD
R/W-Xh					R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED		WAKESTATE	RESERVED				
R/W-Xh		R-0h	R/W-Xh				
7	6	5	4	3	2	1	0
PC	RESERVED		PF				
R/W-0h	R/W-Xh		R/W-0h				

Table 14-24. PINCM8 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R/W	0h	
28	WCOMP	R/W	0h	Wakeup Compare Value bit. This bitfield indicates on what data match this IOCELL with intiate a Wake from SHUTDOWN when WUEN = 1. Reset type: SYSRST 0h = 0 1h = 1
27	WUEN	R/W	0h	Wakeup Enable bit. This bitfield indicates whether or not the Wake from Shutdown is enabled on this IOCELL. Reset type: SYSRST 0h = 0 1h = 1
26	INV	R/W	0h	Data Inversion Control Selection. This bitfield controls whether the input and output logic of this IOCELL is inverted. Reset type: SYSRST 0h = 0 1h = 1
25	Z1	R/W	0h	HI-Z Output State Control Selection. This bitfield controls whether or not a high output value will tri-state the output (open-drain functionality). Reset type: SYSRST 0h = 0 1h = 1
24-19	RESERVED	R/W	Xh	

Table 14-24. PINCM8 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
18	INENA	R/W	0h	Input Enable Control Selection. This bitfield controls whether or not the input to this IOCELL propagates to the connected peripheral function. Reset type: SYSRST 0h = 0 1h = 1
17	PIPU	R/W	0h	Pull Up Resistor Control. This bitfield controls whether the pull up resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
16	PIPD	R/W	0h	Pull Down Resistor Control. This bitfield controls whether the pull down resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
15-14	RESERVED	R/W	Xh	
13	WAKESTATE	R	0h	This bitfield has the IOPAD WAKEUP signal as a status bit Reset type: SYSRST
12-8	RESERVED	R/W	Xh	
7	PC	R/W	0h	Peripheral Connect bit. This bit indicates if the designated peripheral function is "Connected" Reset type: SYSRST 0h = 0 1h = 1
6-5	RESERVED	R/W	Xh	
4-0	PF	R/W	0h	Peripheral Function selection bits. This bitfield indicates what peripheral function is selected for a specific pin. Reset type: SYSRST 0h = 0 1h = 1 3h = 3 5h = 5 6h = 6 7h = 7 9h = 9 Ah = 10 Ch = 12 Eh = 14 Fh = 15 10h = 16

10 PINCM9 Register (Offset = 24h) [Reset = 0XXXXXX0h]

PINCM9 is shown in Figure 14-11 and described in Table 14-16.

Return to the Summary Table.

Pin Control Management Register

Figure 14-11. PINCM9 Register

31	30	29	28	27	26	25	24
RESERVED			WCOMP	WUEN	INV	Z1	RESERVED
R/W-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-Xh
23	22	21	20	19	18	17	16
RESERVED					INENA	PIPU	PIPD
R/W-Xh					R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED		WAKESTATE	RESERVED				
R/W-Xh		R-0h	R/W-Xh				
7	6	5	4	3	2	1	0
PC	RESERVED		PF				
R/W-0h	R/W-Xh		R/W-0h				

Table 14-26. PINCM9 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R/W	0h	
28	WCOMP	R/W	0h	Wakeup Compare Value bit. This bitfield indicates on what data match this IOCELL with intiate a Wake from SHUTDOWN when WUEN = 1. Reset type: SYSRST 0h = 0 1h = 1
27	WUEN	R/W	0h	Wakeup Enable bit. This bitfield indicates whether or not the Wake from Shutdown is enabled on this IOCELL. Reset type: SYSRST 0h = 0 1h = 1
26	INV	R/W	0h	Data Inversion Control Selection. This bitfield controls whether the input and output logic of this IOCELL is inverted. Reset type: SYSRST 0h = 0 1h = 1
25	Z1	R/W	0h	HI-Z Output State Control Selection. This bitfield controls whether or not a high output value will tri-state the output (open-drain functionality). Reset type: SYSRST 0h = 0 1h = 1
24-19	RESERVED	R/W	Xh	

Table 14-26. PINCM9 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
18	INENA	R/W	0h	Input Enable Control Selection. This bitfield controls whether or not the input to this IOCELL propagates to the connected peripheral function. Reset type: SYSRST 0h = 0 1h = 1
17	PIPU	R/W	0h	Pull Up Resistor Control. This bitfield controls whether the pull up resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
16	PIPD	R/W	0h	Pull Down Resistor Control. This bitfield controls whether the pull down resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
15-14	RESERVED	R/W	Xh	
13	WAKESTATE	R	0h	This bitfield has the IOPAD WAKEUP signal as a status bit Reset type: SYSRST
12-8	RESERVED	R/W	Xh	
7	PC	R/W	0h	Peripheral Connect bit. This bit indicates if the designated peripheral function is "Connected" Reset type: SYSRST 0h = 0 1h = 1
6-5	RESERVED	R/W	Xh	
4-0	PF	R/W	0h	Peripheral Function selection bits. This bitfield indicates what peripheral function is selected for a specific pin. Reset type: SYSRST 0h = 0 1h = 1 5h = 5 6h = 6 7h = 7 8h = 8 9h = 9 Ah = 10 Bh = 11 Ch = 12 Dh = 13 Eh = 14 Fh = 15 10h = 16

ADVANCE INFORMATION

11 PINCM10 Register (Offset = 28h) [Reset = 0XXXXXX0h]

PINCM10 is shown in Figure 14-12 and described in Table 14-17.

Return to the Summary Table.

Pin Control Management Register

Figure 14-12. PINCM10 Register

31	30	29	28	27	26	25	24
RESERVED			WCOMP	WUEN	INV	Z1	RESERVED
R/W-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-Xh
23	22	21	20	19	18	17	16
RESERVED					INENA	PIPU	PIPD
R/W-Xh					R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED		WAKESTATE	RESERVED				
R/W-Xh		R-0h	R/W-Xh				
7	6	5	4	3	2	1	0
PC	RESERVED		PF				
R/W-0h	R/W-Xh		R/W-0h				

Table 14-28. PINCM10 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R/W	0h	
28	WCOMP	R/W	0h	Wakeup Compare Value bit. This bitfield indicates on what data match this IOCELL with intiate a Wake from SHUTDOWN when WUEN = 1. Reset type: SYSRST 0h = 0 1h = 1
27	WUEN	R/W	0h	Wakeup Enable bit. This bitfield indicates whether or not the Wake from Shutdown is enabled on this IOCELL. Reset type: SYSRST 0h = 0 1h = 1
26	INV	R/W	0h	Data Inversion Control Selection. This bitfield controls whether the input and output logic of this IOCELL is inverted. Reset type: SYSRST 0h = 0 1h = 1
25	Z1	R/W	0h	HI-Z Output State Control Selection. This bitfield controls whether or not a high output value will tri-state the output (open-drain functionality). Reset type: SYSRST 0h = 0 1h = 1
24-19	RESERVED	R/W	Xh	

Table 14-28. PINCM10 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
18	INENA	R/W	0h	Input Enable Control Selection. This bitfield controls whether or not the input to this IOCELL propagates to the connected peripheral function. Reset type: SYSRST 0h = 0 1h = 1
17	PIPU	R/W	0h	Pull Up Resistor Control. This bitfield controls whether the pull up resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
16	PIPD	R/W	0h	Pull Down Resistor Control. This bitfield controls whether the pull down resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
15-14	RESERVED	R/W	Xh	
13	WAKESTATE	R	0h	This bitfield has the IOPAD WAKEUP signal as a status bit Reset type: SYSRST
12-8	RESERVED	R/W	Xh	
7	PC	R/W	0h	Peripheral Connect bit. This bit indicates if the designated peripheral function is "Connected" Reset type: SYSRST 0h = 0 1h = 1
6-5	RESERVED	R/W	Xh	
4-0	PF	R/W	0h	Peripheral Function selection bits. This bitfield indicates what peripheral function is selected for a specific pin. Reset type: SYSRST 0h = 0 1h = 1 4h = 4 5h = 5 6h = 6 7h = 7 8h = 8 9h = 9 Ah = 10 Bh = 11 Ch = 12 Dh = 13 Eh = 14 10h = 16

12 PINCM11 Register (Offset = 2Ch) [Reset = 0XXXXXX0h]

PINCM11 is shown in Figure 14-13 and described in Table 14-18.

Return to the Summary Table.

Pin Control Management Register

Figure 14-13. PINCM11 Register

31	30	29	28	27	26	25	24
RESERVED			WCOMP	WUEN	INV	Z1	RESERVED
R/W-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-Xh
23	22	21	20	19	18	17	16
RESERVED					INENA	PIPU	PIPD
R/W-Xh					R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED		WAKESTATE	RESERVED				
R/W-Xh		R-0h	R/W-Xh				
7	6	5	4	3	2	1	0
PC	RESERVED		PF				
R/W-0h	R/W-Xh		R/W-0h				

Table 14-30. PINCM11 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R/W	0h	
28	WCOMP	R/W	0h	Wakeup Compare Value bit. This bitfield indicates on what data match this IOCELL with intiate a Wake from SHUTDOWN when WUEN = 1. Reset type: SYSRST 0h = 0 1h = 1
27	WUEN	R/W	0h	Wakeup Enable bit. This bitfield indicates whether or not the Wake from Shutdown is enabled on this IOCELL. Reset type: SYSRST 0h = 0 1h = 1
26	INV	R/W	0h	Data Inversion Control Selection. This bitfield controls whether the input and output logic of this IOCELL is inverted. Reset type: SYSRST 0h = 0 1h = 1
25	Z1	R/W	0h	HI-Z Output State Control Selection. This bitfield controls whether or not a high output value will tri-state the output (open-drain functionality). Reset type: SYSRST 0h = 0 1h = 1
24-19	RESERVED	R/W	Xh	

Table 14-30. PINCM11 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
18	INENA	R/W	0h	Input Enable Control Selection. This bitfield controls whether or not the input to this IOCELL propagates to the connected peripheral function. Reset type: SYSRST 0h = 0 1h = 1
17	PIPU	R/W	0h	Pull Up Resistor Control. This bitfield controls whether the pull up resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
16	PIPD	R/W	0h	Pull Down Resistor Control. This bitfield controls whether the pull down resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
15-14	RESERVED	R/W	Xh	
13	WAKESTATE	R	0h	This bitfield has the IOPAD WAKEUP signal as a status bit Reset type: SYSRST
12-8	RESERVED	R/W	Xh	
7	PC	R/W	0h	Peripheral Connect bit. This bit indicates if the designated peripheral function is "Connected" Reset type: SYSRST 0h = 0 1h = 1
6-5	RESERVED	R/W	Xh	
4-0	PF	R/W	0h	Peripheral Function selection bits. This bitfield indicates what peripheral function is selected for a specific pin. Reset type: SYSRST 0h = 0 1h = 1 2h = 2 3h = 3 4h = 4 5h = 5 6h = 6 7h = 7 8h = 8 9h = 9 Ah = 10 Bh = 11 Ch = 12 Dh = 13 Eh = 14 Fh = 15 10h = 16

13 PINCM12 Register (Offset = 30h) [Reset = 0XXXXXX0h]

PINCM12 is shown in Figure 14-14 and described in Table 14-19.

Return to the Summary Table.

Pin Control Management Register

Figure 14-14. PINCM12 Register

31	30	29	28	27	26	25	24
RESERVED			WCOMP	WUEN	INV	Z1	RESERVED
R/W-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-Xh
23	22	21	20	19	18	17	16
RESERVED			DRV	RESERVED	INENA	PIPU	PIPD
R/W-Xh			R/W-0h	R/W-Xh	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED		WAKESTATE	RESERVED				
R/W-Xh		R-0h	R/W-Xh				
7	6	5	4	3	2	1	0
PC	RESERVED		PF				
R/W-0h	R/W-Xh		R/W-0h				

Table 14-32. PINCM12 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R/W	0h	
28	WCOMP	R/W	0h	Wakeup Compare Value bit. This bitfield indicates on what data match this IOCELL with intiate a Wake from SHUTDOWN when WUEN = 1. Reset type: SYSRST 0h = 0 1h = 1
27	WUEN	R/W	0h	Wakeup Enable bit. This bitfield indicates whether or not the Wake from Shutdown is enabled on this IOCELL. Reset type: SYSRST 0h = 0 1h = 1
26	INV	R/W	0h	Data Inversion Control Selection. This bitfield controls whether the input and output logic of this IOCELL is inverted. Reset type: SYSRST 0h = 0 1h = 1
25	Z1	R/W	0h	HI-Z Output State Control Selection. This bitfield controls whether or not a high output value will tri-state the output (open-drain functionality). Reset type: SYSRST 0h = 0 1h = 1
24-21	RESERVED	R/W	Xh	

Table 14-32. PINCM12 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
20	DRV	R/W	0h	bit Reset type: SYSRST 0h = 0 1h = 1
19	RESERVED	R/W	Xh	
18	INENA	R/W	0h	Input Enable Control Selection. This bitfield controls whether or not the input to this IOCELL propagates to the connected peripheral function. Reset type: SYSRST 0h = 0 1h = 1
17	PIPU	R/W	0h	Pull Up Resistor Control. This bitfield controls whether the pull up resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
16	PIPD	R/W	0h	Pull Down Resistor Control. This bitfield controls whether the pull down resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
15-14	RESERVED	R/W	Xh	
13	WAKESTATE	R	0h	This bitfield has the IOPAD WAKEUP signal as a status bit Reset type: SYSRST
12-8	RESERVED	R/W	Xh	
7	PC	R/W	0h	Peripheral Connect bit. This bit indicates if the designated peripheral function is "Connected" Reset type: SYSRST 0h = 0 1h = 1
6-5	RESERVED	R/W	Xh	
4-0	PF	R/W	0h	Peripheral Function selection bits. This bitfield indicates what peripheral function is selected for a specific pin. Reset type: SYSRST 0h = 0 1h = 1 2h = 2 3h = 3 6h = 6 7h = 7 8h = 8 9h = 9 Ah = 10 Bh = 11 Ch = 12 Dh = 13 Eh = 14 Fh = 15 10h = 16

14 PINCM13 Register (Offset = 34h) [Reset = 0XXXXXX2h]

PINCM13 is shown in Figure 14-15 and described in Table 14-20.

Return to the Summary Table.

Pin Control Management Register

Figure 14-15. PINCM13 Register

31	30	29	28	27	26	25	24
RESERVED			WCOMP	WUEN	INV	Z1	RESERVED
R/W-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-Xh
23	22	21	20	19	18	17	16
RESERVED					INENA	PIPU	PIPD
R/W-Xh					R/W-1h	R/W-1h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED		WAKESTATE	RESERVED				
R/W-Xh		R-0h	R/W-Xh				
7	6	5	4	3	2	1	0
PC	RESERVED		PF				
R/W-1h	R/W-Xh		R/W-2h				

Table 14-34. PINCM13 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R/W	0h	
28	WCOMP	R/W	0h	Wakeup Compare Value bit. This bitfield indicates on what data match this IOCELL with intiate a Wake from SHUTDOWN when WUEN = 1. Reset type: SYSRST 0h = 0 1h = 1
27	WUEN	R/W	0h	Wakeup Enable bit. This bitfield indicates whether or not the Wake from Shutdown is enabled on this IOCELL. Reset type: SYSRST 0h = 0 1h = 1
26	INV	R/W	0h	Data Inversion Control Selection. This bitfield controls whether the input and output logic of this IOCELL is inverted. Reset type: SYSRST 0h = 0 1h = 1
25	Z1	R/W	0h	HI-Z Output State Control Selection. This bitfield controls whether or not a high output value will tri-state the output (open-drain functionality). Reset type: SYSRST 0h = 0 1h = 1
24-19	RESERVED	R/W	Xh	

Table 14-34. PINCM13 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
18	INENA	R/W	1h	Input Enable Control Selection. This bitfield controls whether or not the input to this IOCELL propagates to the connected peripheral function. Reset type: SYSRST 0h = 0 1h = 1
17	PIPU	R/W	1h	Pull Up Resistor Control. This bitfield controls whether the pull up resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
16	PIPD	R/W	0h	Pull Down Resistor Control. This bitfield controls whether the pull down resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
15-14	RESERVED	R/W	Xh	
13	WAKESTATE	R	0h	This bitfield has the IOPAD WAKEUP signal as a status bit Reset type: SYSRST
12-8	RESERVED	R/W	Xh	
7	PC	R/W	1h	Peripheral Connect bit. This bit indicates if the designated peripheral function is "Connected" Reset type: SYSRST 0h = 0 1h = 1
6-5	RESERVED	R/W	Xh	
4-0	PF	R/W	2h	Peripheral Function selection bits. This bitfield indicates what peripheral function is selected for a specific pin. Reset type: SYSRST 0h = 0 1h = 1 2h = 2 3h = 3 4h = 4 5h = 5 6h = 6 8h = 8 9h = 9 Ah = 10 Bh = 11 Ch = 12 Dh = 13 10h = 16

15 PINCM14 Register (Offset = 38h) [Reset = 0XXXXXX2h]

PINCM14 is shown in Figure 14-16 and described in Table 14-21.

Return to the Summary Table.

Pin Control Management Register

Figure 14-16. PINCM14 Register

31	30	29	28	27	26	25	24
RESERVED			WCOMP	WUEN	INV	Z1	RESERVED
R/W-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-Xh
23	22	21	20	19	18	17	16
RESERVED					INENA	PIPU	PIPD
R/W-Xh					R/W-1h	R/W-0h	R/W-1h
15	14	13	12	11	10	9	8
RESERVED		WAKESTATE	RESERVED				
R/W-Xh		R-0h	R/W-Xh				
7	6	5	4	3	2	1	0
PC	RESERVED		PF				
R/W-1h	R/W-Xh		R/W-2h				

Table 14-36. PINCM14 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R/W	0h	
28	WCOMP	R/W	0h	Wakeup Compare Value bit. This bitfield indicates on what data match this IOCELL with intiate a Wake from SHUTDOWN when WUEN = 1. Reset type: SYSRST 0h = 0 1h = 1
27	WUEN	R/W	0h	Wakeup Enable bit. This bitfield indicates whether or not the Wake from Shutdown is enabled on this IOCELL. Reset type: SYSRST 0h = 0 1h = 1
26	INV	R/W	0h	Data Inversion Control Selection. This bitfield controls whether the input and output logic of this IOCELL is inverted. Reset type: SYSRST 0h = 0 1h = 1
25	Z1	R/W	0h	HI-Z Output State Control Selection. This bitfield controls whether or not a high output value will tri-state the output (open-drain functionality). Reset type: SYSRST 0h = 0 1h = 1
24-19	RESERVED	R/W	Xh	

Table 14-36. PINCM14 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
18	INENA	R/W	1h	Input Enable Control Selection. This bitfield controls whether or not the input to this IOCELL propagates to the connected peripheral function. Reset type: SYSRST 0h = 0 1h = 1
17	PIPU	R/W	0h	Pull Up Resistor Control. This bitfield controls whether the pull up resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
16	PIPD	R/W	1h	Pull Down Resistor Control. This bitfield controls whether the pull down resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
15-14	RESERVED	R/W	Xh	
13	WAKESTATE	R	0h	This bitfield has the IOPAD WAKEUP signal as a status bit Reset type: SYSRST
12-8	RESERVED	R/W	Xh	
7	PC	R/W	1h	Peripheral Connect bit. This bit indicates if the designated peripheral function is "Connected" Reset type: SYSRST 0h = 0 1h = 1
6-5	RESERVED	R/W	Xh	
4-0	PF	R/W	2h	Peripheral Function selection bits. This bitfield indicates what peripheral function is selected for a specific pin. Reset type: SYSRST 0h = 0 1h = 1 2h = 2 5h = 5 6h = 6 7h = 7 8h = 8 9h = 9 Ah = 10 10h = 16

16 PINCM15 Register (Offset = 3Ch) [Reset = 0XXXXXX2h]

PINCM15 is shown in Figure 14-17 and described in Table 14-22.

Return to the Summary Table.

Pin Control Management Register

Figure 14-17. PINCM15 Register

31	30	29	28	27	26	25	24
RESERVED			WCOMP	WUEN	INV	Z1	RESERVED
R/W-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-Xh
23	22	21	20	19	18	17	16
RESERVED			DRV	RESERVED	INENA	PIPU	PIPD
R/W-Xh			R/W-0h	R/W-Xh	R/W-1h	R/W-1h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED		WAKESTATE	RESERVED				
R/W-Xh		R-0h	R/W-Xh				
7	6	5	4	3	2	1	0
PC	RESERVED		PF				
R/W-1h	R/W-Xh		R/W-2h				

Table 14-38. PINCM15 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R/W	0h	
28	WCOMP	R/W	0h	Wakeup Compare Value bit. This bitfield indicates on what data match this IOCELL with intiate a Wake from SHUTDOWN when WUEN = 1. Reset type: SYSRST 0h = 0 1h = 1
27	WUEN	R/W	0h	Wakeup Enable bit. This bitfield indicates whether or not the Wake from Shutdown is enabled on this IOCELL. Reset type: SYSRST 0h = 0 1h = 1
26	INV	R/W	0h	Data Inversion Control Selection. This bitfield controls whether the input and output logic of this IOCELL is inverted. Reset type: SYSRST 0h = 0 1h = 1
25	Z1	R/W	0h	HI-Z Output State Control Selection. This bitfield controls whether or not a high output value will tri-state the output (open-drain functionality). Reset type: SYSRST 0h = 0 1h = 1
24-21	RESERVED	R/W	Xh	

Table 14-38. PINCM15 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
20	DRV	R/W	0h	bit Reset type: SYSRST 0h = 0 1h = 1
19	RESERVED	R/W	Xh	
18	INENA	R/W	1h	Input Enable Control Selection. This bitfield controls whether or not the input to this IOCELL propagates to the connected peripheral function. Reset type: SYSRST 0h = 0 1h = 1
17	PIPU	R/W	1h	Pull Up Resistor Control. This bitfield controls whether the pull up resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
16	PIPD	R/W	0h	Pull Down Resistor Control. This bitfield controls whether the pull down resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
15-14	RESERVED	R/W	Xh	
13	WAKESTATE	R	0h	This bitfield has the IOPAD WAKEUP signal as a status bit Reset type: SYSRST
12-8	RESERVED	R/W	Xh	
7	PC	R/W	1h	Peripheral Connect bit. This bit indicates if the designated peripheral function is "Connected" Reset type: SYSRST 0h = 0 1h = 1
6-5	RESERVED	R/W	Xh	
4-0	PF	R/W	2h	Peripheral Function selection bits. This bitfield indicates what peripheral function is selected for a specific pin. Reset type: SYSRST 0h = 0 1h = 1 2h = 2 3h = 3 4h = 4 5h = 5 6h = 6 7h = 7 8h = 8 9h = 9 Ah = 10 Bh = 11 Ch = 12 10h = 16

17 PINCM16 Register (Offset = 40h) [Reset = 0XXXXXX0h]

PINCM16 is shown in Figure 14-18 and described in Table 14-23.

Return to the Summary Table.

Pin Control Management Register

Figure 14-18. PINCM16 Register

31	30	29	28	27	26	25	24
RESERVED			WCOMP	WUEN	INV	Z1	RESERVED
R/W-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-Xh
23	22	21	20	19	18	17	16
RESERVED					INENA	PIPU	PIPD
R/W-Xh					R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED		WAKESTATE	RESERVED				
R/W-Xh		R-0h	R/W-Xh				
7	6	5	4	3	2	1	0
PC	RESERVED		PF				
R/W-0h	R/W-Xh		R/W-0h				

Table 14-40. PINCM16 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R/W	0h	
28	WCOMP	R/W	0h	Wakeup Compare Value bit. This bitfield indicates on what data match this IOCELL with intiate a Wake from SHUTDOWN when WUEN = 1. Reset type: SYSRST 0h = 0 1h = 1
27	WUEN	R/W	0h	Wakeup Enable bit. This bitfield indicates whether or not the Wake from Shutdown is enabled on this IOCELL. Reset type: SYSRST 0h = 0 1h = 1
26	INV	R/W	0h	Data Inversion Control Selection. This bitfield controls whether the input and output logic of this IOCELL is inverted. Reset type: SYSRST 0h = 0 1h = 1
25	Z1	R/W	0h	HI-Z Output State Control Selection. This bitfield controls whether or not a high output value will tri-state the output (open-drain functionality). Reset type: SYSRST 0h = 0 1h = 1
24-19	RESERVED	R/W	Xh	

Table 14-40. PINCM16 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
18	INENA	R/W	0h	Input Enable Control Selection. This bitfield controls whether or not the input to this IOCELL propagates to the connected peripheral function. Reset type: SYSRST 0h = 0 1h = 1
17	PIPU	R/W	0h	Pull Up Resistor Control. This bitfield controls whether the pull up resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
16	PIPD	R/W	0h	Pull Down Resistor Control. This bitfield controls whether the pull down resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
15-14	RESERVED	R/W	Xh	
13	WAKESTATE	R	0h	This bitfield has the IOPAD WAKEUP signal as a status bit Reset type: SYSRST
12-8	RESERVED	R/W	Xh	
7	PC	R/W	0h	Peripheral Connect bit. This bit indicates if the designated peripheral function is "Connected" Reset type: SYSRST 0h = 0 1h = 1
6-5	RESERVED	R/W	Xh	
4-0	PF	R/W	0h	Peripheral Function selection bits. This bitfield indicates what peripheral function is selected for a specific pin. Reset type: SYSRST 0h = 0 1h = 1 3h = 3 4h = 4 5h = 5 6h = 6 7h = 7 8h = 8 9h = 9 10h = 16

18 PINCM17 Register (Offset = 44h) [Reset = 0XXXXXX0h]

PINCM17 is shown in Figure 14-19 and described in Table 14-24.

Return to the Summary Table.

Pin Control Management Register

Figure 14-19. PINCM17 Register

31	30	29	28	27	26	25	24
RESERVED			WCOMP	WUEN	INV	Z1	RESERVED
R/W-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-Xh
23	22	21	20	19	18	17	16
RESERVED					INENA	PIPU	PIPD
R/W-Xh					R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED		WAKESTATE	RESERVED				
R/W-Xh		R-0h	R/W-Xh				
7	6	5	4	3	2	1	0
PC	RESERVED		PF				
R/W-0h	R/W-Xh		R/W-0h				

Table 14-42. PINCM17 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R/W	0h	
28	WCOMP	R/W	0h	Wakeup Compare Value bit. This bitfield indicates on what data match this IOCELL with intiate a Wake from SHUTDOWN when WUEN = 1. Reset type: SYSRST 0h = 0 1h = 1
27	WUEN	R/W	0h	Wakeup Enable bit. This bitfield indicates whether or not the Wake from Shutdown is enabled on this IOCELL. Reset type: SYSRST 0h = 0 1h = 1
26	INV	R/W	0h	Data Inversion Control Selection. This bitfield controls whether the input and output logic of this IOCELL is inverted. Reset type: SYSRST 0h = 0 1h = 1
25	Z1	R/W	0h	HI-Z Output State Control Selection. This bitfield controls whether or not a high output value will tri-state the output (open-drain functionality). Reset type: SYSRST 0h = 0 1h = 1
24-19	RESERVED	R/W	Xh	

Table 14-42. PINCM17 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
18	INENA	R/W	0h	Input Enable Control Selection. This bitfield controls whether or not the input to this IOCELL propagates to the connected peripheral function. Reset type: SYSRST 0h = 0 1h = 1
17	PIPU	R/W	0h	Pull Up Resistor Control. This bitfield controls whether the pull up resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
16	PIPD	R/W	0h	Pull Down Resistor Control. This bitfield controls whether the pull down resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
15-14	RESERVED	R/W	Xh	
13	WAKESTATE	R	0h	This bitfield has the IOPAD WAKEUP signal as a status bit Reset type: SYSRST
12-8	RESERVED	R/W	Xh	
7	PC	R/W	0h	Peripheral Connect bit. This bit indicates if the designated peripheral function is "Connected" Reset type: SYSRST 0h = 0 1h = 1
6-5	RESERVED	R/W	Xh	
4-0	PF	R/W	0h	Peripheral Function selection bits. This bitfield indicates what peripheral function is selected for a specific pin. Reset type: SYSRST 0h = 0 1h = 1 3h = 3 4h = 4 5h = 5 6h = 6 7h = 7 8h = 8 Bh = 11 Dh = 13 10h = 16

19 PINCM18 Register (Offset = 48h) [Reset = 0XXXXXX0h]

PINCM18 is shown in Figure 14-20 and described in Table 14-25.

Return to the Summary Table.

Pin Control Management Register

Figure 14-20. PINCM18 Register

31	30	29	28	27	26	25	24
RESERVED			WCOMP	WUEN	INV	Z1	RESERVED
R/W-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-Xh
23	22	21	20	19	18	17	16
RESERVED					INENA	PIPU	PIPD
R/W-Xh					R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED		WAKESTATE	RESERVED				
R/W-Xh		R-0h	R/W-Xh				
7	6	5	4	3	2	1	0
PC	RESERVED		PF				
R/W-0h	R/W-Xh		R/W-0h				

Table 14-44. PINCM18 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R/W	0h	
28	WCOMP	R/W	0h	Wakeup Compare Value bit. This bitfield indicates on what data match this IOCELL with intiate a Wake from SHUTDOWN when WUEN = 1. Reset type: SYSRST 0h = 0 1h = 1
27	WUEN	R/W	0h	Wakeup Enable bit. This bitfield indicates whether or not the Wake from Shutdown is enabled on this IOCELL. Reset type: SYSRST 0h = 0 1h = 1
26	INV	R/W	0h	Data Inversion Control Selection. This bitfield controls whether the input and output logic of this IOCELL is inverted. Reset type: SYSRST 0h = 0 1h = 1
25	Z1	R/W	0h	HI-Z Output State Control Selection. This bitfield controls whether or not a high output value will tri-state the output (open-drain functionality). Reset type: SYSRST 0h = 0 1h = 1
24-19	RESERVED	R/W	Xh	

Table 14-44. PINCM18 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
18	INENA	R/W	0h	Input Enable Control Selection. This bitfield controls whether or not the input to this IOCELL propagates to the connected peripheral function. Reset type: SYSRST 0h = 0 1h = 1
17	PIPU	R/W	0h	Pull Up Resistor Control. This bitfield controls whether the pull up resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
16	PIPD	R/W	0h	Pull Down Resistor Control. This bitfield controls whether the pull down resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
15-14	RESERVED	R/W	Xh	
13	WAKESTATE	R	0h	This bitfield has the IOPAD WAKEUP signal as a status bit Reset type: SYSRST
12-8	RESERVED	R/W	Xh	
7	PC	R/W	0h	Peripheral Connect bit. This bit indicates if the designated peripheral function is "Connected" Reset type: SYSRST 0h = 0 1h = 1
6-5	RESERVED	R/W	Xh	
4-0	PF	R/W	0h	Peripheral Function selection bits. This bitfield indicates what peripheral function is selected for a specific pin. Reset type: SYSRST 0h = 0 1h = 1 3h = 3 4h = 4 5h = 5 10h = 16

20 PINCM19 Register (Offset = 4Ch) [Reset = 0XXXXXX2h]

PINCM19 is shown in Figure 14-21 and described in Table 14-26.

Return to the Summary Table.

Pin Control Management Register

Figure 14-21. PINCM19 Register

31	30	29	28	27	26	25	24
RESERVED			WCOMP	WUEN	INV	Z1	RESERVED
R/W-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-Xh
23	22	21	20	19	18	17	16
RESERVED			DRV	RESERVED	INENA	PIPU	PIPD
R/W-Xh			R/W-0h	R/W-Xh	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED		WAKESTATE	RESERVED				
R/W-Xh		R-0h	R/W-Xh				
7	6	5	4	3	2	1	0
PC	RESERVED		PF				
R/W-1h	R/W-Xh		R/W-2h				

Table 14-46. PINCM19 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R/W	0h	
28	WCOMP	R/W	0h	Wakeup Compare Value bit. This bitfield indicates on what data match this IOCELL with intiate a Wake from SHUTDOWN when WUEN = 1. Reset type: SYSRST 0h = 0 1h = 1
27	WUEN	R/W	0h	Wakeup Enable bit. This bitfield indicates whether or not the Wake from Shutdown is enabled on this IOCELL. Reset type: SYSRST 0h = 0 1h = 1
26	INV	R/W	0h	Data Inversion Control Selection. This bitfield controls whether the input and output logic of this IOCELL is inverted. Reset type: SYSRST 0h = 0 1h = 1
25	Z1	R/W	0h	HI-Z Output State Control Selection. This bitfield controls whether or not a high output value will tri-state the output (open-drain functionality). Reset type: SYSRST 0h = 0 1h = 1
24-21	RESERVED	R/W	Xh	

Table 14-46. PINCM19 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
20	DRV	R/W	0h	bit Reset type: SYSRST 0h = 0 1h = 1
19	RESERVED	R/W	Xh	
18	INENA	R/W	0h	Input Enable Control Selection. This bitfield controls whether or not the input to this IOCELL propagates to the connected peripheral function. Reset type: SYSRST 0h = 0 1h = 1
17	PIPU	R/W	0h	Pull Up Resistor Control. This bitfield controls whether the pull up resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
16	PIPD	R/W	0h	Pull Down Resistor Control. This bitfield controls whether the pull down resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
15-14	RESERVED	R/W	Xh	
13	WAKESTATE	R	0h	This bitfield has the IOPAD WAKEUP signal as a status bit Reset type: SYSRST
12-8	RESERVED	R/W	Xh	
7	PC	R/W	1h	Peripheral Connect bit. This bit indicates if the designated peripheral function is "Connected" Reset type: SYSRST 0h = 0 1h = 1
6-5	RESERVED	R/W	Xh	
4-0	PF	R/W	2h	Peripheral Function selection bits. This bitfield indicates what peripheral function is selected for a specific pin. Reset type: SYSRST 0h = 0 1h = 1 2h = 2 3h = 3 4h = 4 5h = 5 6h = 6 7h = 7 8h = 8 9h = 9 Ch = 12 10h = 16

21 PINCM20 Register (Offset = 50h) [Reset = 0XXXXXX0h]

PINCM20 is shown in Figure 14-22 and described in Table 14-27.

Return to the Summary Table.

Pin Control Management Register

Figure 14-22. PINCM20 Register

31	30	29	28	27	26	25	24
RESERVED			WCOMP	WUEN	INV	Z1	RESERVED
R/W-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-Xh
23	22	21	20	19	18	17	16
RESERVED					INENA	PIPU	PIPD
R/W-Xh					R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED		WAKESTATE	RESERVED				
R/W-Xh		R-0h	R/W-Xh				
7	6	5	4	3	2	1	0
PC	RESERVED		PF				
R/W-0h	R/W-Xh		R/W-0h				

Table 14-48. PINCM20 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R/W	0h	
28	WCOMP	R/W	0h	Wakeup Compare Value bit. This bitfield indicates on what data match this IOCELL with intiate a Wake from SHUTDOWN when WUEN = 1. Reset type: SYSRST 0h = 0 1h = 1
27	WUEN	R/W	0h	Wakeup Enable bit. This bitfield indicates whether or not the Wake from Shutdown is enabled on this IOCELL. Reset type: SYSRST 0h = 0 1h = 1
26	INV	R/W	0h	Data Inversion Control Selection. This bitfield controls whether the input and output logic of this IOCELL is inverted. Reset type: SYSRST 0h = 0 1h = 1
25	Z1	R/W	0h	HI-Z Output State Control Selection. This bitfield controls whether or not a high output value will tri-state the output (open-drain functionality). Reset type: SYSRST 0h = 0 1h = 1
24-19	RESERVED	R/W	Xh	

Table 14-48. PINCM20 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
18	INENA	R/W	0h	Input Enable Control Selection. This bitfield controls whether or not the input to this IOCELL propagates to the connected peripheral function. Reset type: SYSRST 0h = 0 1h = 1
17	PIPU	R/W	0h	Pull Up Resistor Control. This bitfield controls whether the pull up resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
16	PIPD	R/W	0h	Pull Down Resistor Control. This bitfield controls whether the pull down resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
15-14	RESERVED	R/W	Xh	
13	WAKESTATE	R	0h	This bitfield has the IOPAD WAKEUP signal as a status bit Reset type: SYSRST
12-8	RESERVED	R/W	Xh	
7	PC	R/W	0h	Peripheral Connect bit. This bit indicates if the designated peripheral function is "Connected" Reset type: SYSRST 0h = 0 1h = 1
6-5	RESERVED	R/W	Xh	
4-0	PF	R/W	0h	Peripheral Function selection bits. This bitfield indicates what peripheral function is selected for a specific pin. Reset type: SYSRST 0h = 0 1h = 1 2h = 2 3h = 3 4h = 4 5h = 5 6h = 6 7h = 7 8h = 8 9h = 9 Ah = 10 Bh = 11 Ch = 12 Dh = 13 10h = 16

22 PINCM21 Register (Offset = 54h) [Reset = 0XXXXXX0h]

PINCM21 is shown in Figure 14-23 and described in Table 14-28.

Return to the Summary Table.

Pin Control Management Register

Figure 14-23. PINCM21 Register

31	30	29	28	27	26	25	24
RESERVED			WCOMP	WUEN	INV	Z1	RESERVED
R/W-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-Xh
23	22	21	20	19	18	17	16
RESERVED					INENA	PIPU	PIPD
R/W-Xh					R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED		WAKESTATE	RESERVED				
R/W-Xh		R-0h	R/W-Xh				
7	6	5	4	3	2	1	0
PC	RESERVED		PF				
R/W-0h	R/W-Xh		R/W-0h				

Table 14-50. PINCM21 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R/W	0h	
28	WCOMP	R/W	0h	Wakeup Compare Value bit. This bitfield indicates on what data match this IOCELL with intiate a Wake from SHUTDOWN when WUEN = 1. Reset type: SYSRST 0h = 0 1h = 1
27	WUEN	R/W	0h	Wakeup Enable bit. This bitfield indicates whether or not the Wake from Shutdown is enabled on this IOCELL. Reset type: SYSRST 0h = 0 1h = 1
26	INV	R/W	0h	Data Inversion Control Selection. This bitfield controls whether the input and output logic of this IOCELL is inverted. Reset type: SYSRST 0h = 0 1h = 1
25	Z1	R/W	0h	HI-Z Output State Control Selection. This bitfield controls whether or not a high output value will tri-state the output (open-drain functionality). Reset type: SYSRST 0h = 0 1h = 1
24-19	RESERVED	R/W	Xh	

Table 14-50. PINCM21 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
18	INENA	R/W	0h	Input Enable Control Selection. This bitfield controls whether or not the input to this IOCELL propagates to the connected peripheral function. Reset type: SYSRST 0h = 0 1h = 1
17	PIPU	R/W	0h	Pull Up Resistor Control. This bitfield controls whether the pull up resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
16	PIPD	R/W	0h	Pull Down Resistor Control. This bitfield controls whether the pull down resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
15-14	RESERVED	R/W	Xh	
13	WAKESTATE	R	0h	This bitfield has the IOPAD WAKEUP signal as a status bit Reset type: SYSRST
12-8	RESERVED	R/W	Xh	
7	PC	R/W	0h	Peripheral Connect bit. This bit indicates if the designated peripheral function is "Connected" Reset type: SYSRST 0h = 0 1h = 1
6-5	RESERVED	R/W	Xh	
4-0	PF	R/W	0h	Peripheral Function selection bits. This bitfield indicates what peripheral function is selected for a specific pin. Reset type: SYSRST 0h = 0 1h = 1 3h = 3 4h = 4 5h = 5 6h = 6 7h = 7 8h = 8 9h = 9 Ah = 10 Bh = 11 Ch = 12 Eh = 14 Fh = 15 10h = 16

23 PINCM22 Register (Offset = 58h) [Reset = 0XXXXXX0h]

PINCM22 is shown in Figure 14-24 and described in Table 14-29.

Return to the Summary Table.

Pin Control Management Register

Figure 14-24. PINCM22 Register

31	30	29	28	27	26	25	24
RESERVED			WCOMP	WUEN	INV	Z1	RESERVED
R/W-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-Xh
23	22	21	20	19	18	17	16
RESERVED					INENA	PIPU	PIPD
R/W-Xh					R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED		WAKESTATE	RESERVED				
R/W-Xh		R-0h	R/W-Xh				
7	6	5	4	3	2	1	0
PC	RESERVED		PF				
R/W-0h	R/W-Xh		R/W-0h				

Table 14-52. PINCM22 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R/W	0h	
28	WCOMP	R/W	0h	Wakeup Compare Value bit. This bitfield indicates on what data match this IOCELL with intiate a Wake from SHUTDOWN when WUEN = 1. Reset type: SYSRST 0h = 0 1h = 1
27	WUEN	R/W	0h	Wakeup Enable bit. This bitfield indicates whether or not the Wake from Shutdown is enabled on this IOCELL. Reset type: SYSRST 0h = 0 1h = 1
26	INV	R/W	0h	Data Inversion Control Selection. This bitfield controls whether the input and output logic of this IOCELL is inverted. Reset type: SYSRST 0h = 0 1h = 1
25	Z1	R/W	0h	HI-Z Output State Control Selection. This bitfield controls whether or not a high output value will tri-state the output (open-drain functionality). Reset type: SYSRST 0h = 0 1h = 1
24-19	RESERVED	R/W	Xh	

Table 14-52. PINCM22 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
18	INENA	R/W	0h	Input Enable Control Selection. This bitfield controls whether or not the input to this IOCELL propagates to the connected peripheral function. Reset type: SYSRST 0h = 0 1h = 1
17	PIPU	R/W	0h	Pull Up Resistor Control. This bitfield controls whether the pull up resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
16	PIPD	R/W	0h	Pull Down Resistor Control. This bitfield controls whether the pull down resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
15-14	RESERVED	R/W	Xh	
13	WAKESTATE	R	0h	This bitfield has the IOPAD WAKEUP signal as a status bit Reset type: SYSRST
12-8	RESERVED	R/W	Xh	
7	PC	R/W	0h	Peripheral Connect bit. This bit indicates if the designated peripheral function is "Connected" Reset type: SYSRST 0h = 0 1h = 1
6-5	RESERVED	R/W	Xh	
4-0	PF	R/W	0h	Peripheral Function selection bits. This bitfield indicates what peripheral function is selected for a specific pin. Reset type: SYSRST 0h = 0 1h = 1 2h = 2 3h = 3 4h = 4 5h = 5 6h = 6 7h = 7 8h = 8 9h = 9 Ah = 10 Bh = 11 Ch = 12 Dh = 13 Eh = 14 Fh = 15 10h = 16

24 PINCM23 Register (Offset = 5Ch) [Reset = 0XXXXXX0h]

PINCM23 is shown in Figure 14-25 and described in Table 14-30.

Return to the Summary Table.

Pin Control Management Register

Figure 14-25. PINCM23 Register

31	30	29	28	27	26	25	24
RESERVED			WCOMP	WUEN	INV	Z1	RESERVED
R/W-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-Xh
23	22	21	20	19	18	17	16
RESERVED					INENA	PIPU	PIPD
R/W-Xh					R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED		WAKESTATE	RESERVED				
R/W-Xh		R-0h	R/W-Xh				
7	6	5	4	3	2	1	0
PC	RESERVED		PF				
R/W-0h	R/W-Xh		R/W-0h				

Table 14-54. PINCM23 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R/W	0h	
28	WCOMP	R/W	0h	Wakeup Compare Value bit. This bitfield indicates on what data match this IOCELL with intiate a Wake from SHUTDOWN when WUEN = 1. Reset type: SYSRST 0h = 0 1h = 1
27	WUEN	R/W	0h	Wakeup Enable bit. This bitfield indicates whether or not the Wake from Shutdown is enabled on this IOCELL. Reset type: SYSRST 0h = 0 1h = 1
26	INV	R/W	0h	Data Inversion Control Selection. This bitfield controls whether the input and output logic of this IOCELL is inverted. Reset type: SYSRST 0h = 0 1h = 1
25	Z1	R/W	0h	HI-Z Output State Control Selection. This bitfield controls whether or not a high output value will tri-state the output (open-drain functionality). Reset type: SYSRST 0h = 0 1h = 1
24-19	RESERVED	R/W	Xh	

Table 14-54. PINCM23 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
18	INENA	R/W	0h	Input Enable Control Selection. This bitfield controls whether or not the input to this IOCELL propagates to the connected peripheral function. Reset type: SYSRST 0h = 0 1h = 1
17	PIPU	R/W	0h	Pull Up Resistor Control. This bitfield controls whether the pull up resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
16	PIPD	R/W	0h	Pull Down Resistor Control. This bitfield controls whether the pull down resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
15-14	RESERVED	R/W	Xh	
13	WAKESTATE	R	0h	This bitfield has the IOPAD WAKEUP signal as a status bit Reset type: SYSRST
12-8	RESERVED	R/W	Xh	
7	PC	R/W	0h	Peripheral Connect bit. This bit indicates if the designated peripheral function is "Connected" Reset type: SYSRST 0h = 0 1h = 1
6-5	RESERVED	R/W	Xh	
4-0	PF	R/W	0h	Peripheral Function selection bits. This bitfield indicates what peripheral function is selected for a specific pin. Reset type: SYSRST 0h = 0 1h = 1 2h = 2 3h = 3 4h = 4 5h = 5 6h = 6 7h = 7 8h = 8 9h = 9 Ah = 10 Bh = 11 Ch = 12 Dh = 13 Eh = 14 Fh = 15 10h = 16

25 PINCM24 Register (Offset = 60h) [Reset = 0XXXXXX0h]

PINCM24 is shown in Figure 14-26 and described in Table 14-31.

Return to the Summary Table.

Pin Control Management Register

Figure 14-26. PINCM24 Register

31	30	29	28	27	26	25	24
RESERVED			WCOMP	WUEN	INV	Z1	RESERVED
R/W-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-Xh
23	22	21	20	19	18	17	16
RESERVED					INENA	PIPU	PIPD
R/W-Xh					R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED		WAKESTATE	RESERVED				
R/W-Xh		R-0h	R/W-Xh				
7	6	5	4	3	2	1	0
PC	RESERVED		PF				
R/W-0h	R/W-Xh		R/W-0h				

Table 14-56. PINCM24 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R/W	0h	
28	WCOMP	R/W	0h	Wakeup Compare Value bit. This bitfield indicates on what data match this IOCELL with intiate a Wake from SHUTDOWN when WUEN = 1. Reset type: SYSRST 0h = 0 1h = 1
27	WUEN	R/W	0h	Wakeup Enable bit. This bitfield indicates whether or not the Wake from Shutdown is enabled on this IOCELL. Reset type: SYSRST 0h = 0 1h = 1
26	INV	R/W	0h	Data Inversion Control Selection. This bitfield controls whether the input and output logic of this IOCELL is inverted. Reset type: SYSRST 0h = 0 1h = 1
25	Z1	R/W	0h	HI-Z Output State Control Selection. This bitfield controls whether or not a high output value will tri-state the output (open-drain functionality). Reset type: SYSRST 0h = 0 1h = 1
24-19	RESERVED	R/W	Xh	

Table 14-56. PINCM24 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
18	INENA	R/W	0h	Input Enable Control Selection. This bitfield controls whether or not the input to this IOCELL propagates to the connected peripheral function. Reset type: SYSRST 0h = 0 1h = 1
17	PIPU	R/W	0h	Pull Up Resistor Control. This bitfield controls whether the pull up resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
16	PIPD	R/W	0h	Pull Down Resistor Control. This bitfield controls whether the pull down resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
15-14	RESERVED	R/W	Xh	
13	WAKESTATE	R	0h	This bitfield has the IOPAD WAKEUP signal as a status bit Reset type: SYSRST
12-8	RESERVED	R/W	Xh	
7	PC	R/W	0h	Peripheral Connect bit. This bit indicates if the designated peripheral function is "Connected" Reset type: SYSRST 0h = 0 1h = 1
6-5	RESERVED	R/W	Xh	
4-0	PF	R/W	0h	Peripheral Function selection bits. This bitfield indicates what peripheral function is selected for a specific pin. Reset type: SYSRST 0h = 0 1h = 1 2h = 2 3h = 3 4h = 4 5h = 5 6h = 6 7h = 7 8h = 8 9h = 9 Ah = 10 Bh = 11 Ch = 12 10h = 16

26 PINCM25 Register (Offset = 64h) [Reset = 0XXXXXX0h]

PINCM25 is shown in Figure 14-27 and described in Table 14-32.

Return to the Summary Table.

Pin Control Management Register

Figure 14-27. PINCM25 Register

31	30	29	28	27	26	25	24
RESERVED			WCOMP	WUEN	INV	Z1	RESERVED
R/W-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-Xh
23	22	21	20	19	18	17	16
RESERVED					INENA	PIPU	PIPD
R/W-Xh					R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED		WAKESTATE	RESERVED				
R/W-Xh		R-0h	R/W-Xh				
7	6	5	4	3	2	1	0
PC	RESERVED		PF				
R/W-0h	R/W-Xh		R/W-0h				

Table 14-58. PINCM25 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R/W	0h	
28	WCOMP	R/W	0h	Wakeup Compare Value bit. This bitfield indicates on what data match this IOCELL with intiate a Wake from SHUTDOWN when WUEN = 1. Reset type: SYSRST 0h = 0 1h = 1
27	WUEN	R/W	0h	Wakeup Enable bit. This bitfield indicates whether or not the Wake from Shutdown is enabled on this IOCELL. Reset type: SYSRST 0h = 0 1h = 1
26	INV	R/W	0h	Data Inversion Control Selection. This bitfield controls whether the input and output logic of this IOCELL is inverted. Reset type: SYSRST 0h = 0 1h = 1
25	Z1	R/W	0h	HI-Z Output State Control Selection. This bitfield controls whether or not a high output value will tri-state the output (open-drain functionality). Reset type: SYSRST 0h = 0 1h = 1
24-19	RESERVED	R/W	Xh	

Table 14-58. PINCM25 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
18	INENA	R/W	0h	Input Enable Control Selection. This bitfield controls whether or not the input to this IOCELL propagates to the connected peripheral function. Reset type: SYSRST 0h = 0 1h = 1
17	PIPU	R/W	0h	Pull Up Resistor Control. This bitfield controls whether the pull up resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
16	PIPD	R/W	0h	Pull Down Resistor Control. This bitfield controls whether the pull down resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
15-14	RESERVED	R/W	Xh	
13	WAKESTATE	R	0h	This bitfield has the IOPAD WAKEUP signal as a status bit Reset type: SYSRST
12-8	RESERVED	R/W	Xh	
7	PC	R/W	0h	Peripheral Connect bit. This bit indicates if the designated peripheral function is "Connected" Reset type: SYSRST 0h = 0 1h = 1
6-5	RESERVED	R/W	Xh	
4-0	PF	R/W	0h	Peripheral Function selection bits. This bitfield indicates what peripheral function is selected for a specific pin. Reset type: SYSRST 0h = 0 1h = 1 2h = 2 3h = 3 4h = 4 5h = 5 6h = 6 7h = 7 8h = 8 9h = 9 Ah = 10 Bh = 11 Ch = 12 Dh = 13 Eh = 14 10h = 16

27 PINCM26 Register (Offset = 68h) [Reset = 0XXXXXX0h]

PINCM26 is shown in Figure 14-28 and described in Table 14-33.

Return to the Summary Table.

Pin Control Management Register

Figure 14-28. PINCM26 Register

31	30	29	28	27	26	25	24
RESERVED			WCOMP	WUEN	INV	Z1	RESERVED
R/W-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-Xh
23	22	21	20	19	18	17	16
RESERVED					INENA	PIPU	PIPD
R/W-Xh					R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED		WAKESTATE	RESERVED				
R/W-Xh		R-0h	R/W-Xh				
7	6	5	4	3	2	1	0
PC	RESERVED		PF				
R/W-0h	R/W-Xh		R/W-0h				

Table 14-60. PINCM26 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R/W	0h	
28	WCOMP	R/W	0h	Wakeup Compare Value bit. This bitfield indicates on what data match this IOCELL with intiate a Wake from SHUTDOWN when WUEN = 1. Reset type: SYSRST 0h = 0 1h = 1
27	WUEN	R/W	0h	Wakeup Enable bit. This bitfield indicates whether or not the Wake from Shutdown is enabled on this IOCELL. Reset type: SYSRST 0h = 0 1h = 1
26	INV	R/W	0h	Data Inversion Control Selection. This bitfield controls whether the input and output logic of this IOCELL is inverted. Reset type: SYSRST 0h = 0 1h = 1
25	Z1	R/W	0h	HI-Z Output State Control Selection. This bitfield controls whether or not a high output value will tri-state the output (open-drain functionality). Reset type: SYSRST 0h = 0 1h = 1
24-19	RESERVED	R/W	Xh	

Table 14-60. PINCM26 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
18	INENA	R/W	0h	Input Enable Control Selection. This bitfield controls whether or not the input to this IOCELL propagates to the connected peripheral function. Reset type: SYSRST 0h = 0 1h = 1
17	PIPU	R/W	0h	Pull Up Resistor Control. This bitfield controls whether the pull up resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
16	PIPD	R/W	0h	Pull Down Resistor Control. This bitfield controls whether the pull down resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
15-14	RESERVED	R/W	Xh	
13	WAKESTATE	R	0h	This bitfield has the IOPAD WAKEUP signal as a status bit Reset type: SYSRST
12-8	RESERVED	R/W	Xh	
7	PC	R/W	0h	Peripheral Connect bit. This bit indicates if the designated peripheral function is "Connected" Reset type: SYSRST 0h = 0 1h = 1
6-5	RESERVED	R/W	Xh	
4-0	PF	R/W	0h	Peripheral Function selection bits. This bitfield indicates what peripheral function is selected for a specific pin. Reset type: SYSRST 0h = 0 1h = 1 2h = 2 3h = 3 4h = 4 5h = 5 6h = 6 8h = 8 9h = 9 Ah = 10 Bh = 11 Dh = 13 10h = 16

28 PINCM27 Register (Offset = 6Ch) [Reset = 0XXXXXX0h]

PINCM27 is shown in Figure 14-29 and described in Table 14-34.

Return to the Summary Table.

Pin Control Management Register

Figure 14-29. PINCM27 Register

31	30	29	28	27	26	25	24
RESERVED			WCOMP	WUEN	INV	Z1	RESERVED
R/W-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-Xh
23	22	21	20	19	18	17	16
RESERVED					INENA	PIPU	PIPD
R/W-Xh					R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED		WAKESTATE	RESERVED				
R/W-Xh		R-0h	R/W-Xh				
7	6	5	4	3	2	1	0
PC	RESERVED		PF				
R/W-0h	R/W-Xh		R/W-0h				

Table 14-62. PINCM27 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R/W	0h	
28	WCOMP	R/W	0h	Wakeup Compare Value bit. This bitfield indicates on what data match this IOCELL with intiate a Wake from SHUTDOWN when WUEN = 1. Reset type: SYSRST 0h = 0 1h = 1
27	WUEN	R/W	0h	Wakeup Enable bit. This bitfield indicates whether or not the Wake from Shutdown is enabled on this IOCELL. Reset type: SYSRST 0h = 0 1h = 1
26	INV	R/W	0h	Data Inversion Control Selection. This bitfield controls whether the input and output logic of this IOCELL is inverted. Reset type: SYSRST 0h = 0 1h = 1
25	Z1	R/W	0h	HI-Z Output State Control Selection. This bitfield controls whether or not a high output value will tri-state the output (open-drain functionality). Reset type: SYSRST 0h = 0 1h = 1
24-19	RESERVED	R/W	Xh	

Table 14-62. PINCM27 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
18	INENA	R/W	0h	Input Enable Control Selection. This bitfield controls whether or not the input to this IOCELL propagates to the connected peripheral function. Reset type: SYSRST 0h = 0 1h = 1
17	PIPU	R/W	0h	Pull Up Resistor Control. This bitfield controls whether the pull up resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
16	PIPD	R/W	0h	Pull Down Resistor Control. This bitfield controls whether the pull down resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
15-14	RESERVED	R/W	Xh	
13	WAKESTATE	R	0h	This bitfield has the IOPAD WAKEUP signal as a status bit Reset type: SYSRST
12-8	RESERVED	R/W	Xh	
7	PC	R/W	0h	Peripheral Connect bit. This bit indicates if the designated peripheral function is "Connected" Reset type: SYSRST 0h = 0 1h = 1
6-5	RESERVED	R/W	Xh	
4-0	PF	R/W	0h	Peripheral Function selection bits. This bitfield indicates what peripheral function is selected for a specific pin. Reset type: SYSRST 0h = 0 1h = 1 2h = 2 3h = 3 4h = 4 5h = 5 7h = 7 8h = 8 9h = 9 Ah = 10 Bh = 11 10h = 16

29 PINCM28 Register (Offset = 70h) [Reset = 0XXXXXX0h]

PINCM28 is shown in Figure 14-30 and described in Table 14-35.

Return to the Summary Table.

Pin Control Management Register

Figure 14-30. PINCM28 Register

31	30	29	28	27	26	25	24
RESERVED			WCOMP	WUEN	INV	Z1	RESERVED
R/W-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-Xh
23	22	21	20	19	18	17	16
RESERVED					INENA	PIPU	PIPD
R/W-Xh					R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED		WAKESTATE	RESERVED				
R/W-Xh		R-0h	R/W-Xh				
7	6	5	4	3	2	1	0
PC	RESERVED		PF				
R/W-0h	R/W-Xh		R/W-0h				

Table 14-64. PINCM28 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R/W	0h	
28	WCOMP	R/W	0h	Wakeup Compare Value bit. This bitfield indicates on what data match this IOCELL with intiate a Wake from SHUTDOWN when WUEN = 1. Reset type: SYSRST 0h = 0 1h = 1
27	WUEN	R/W	0h	Wakeup Enable bit. This bitfield indicates whether or not the Wake from Shutdown is enabled on this IOCELL. Reset type: SYSRST 0h = 0 1h = 1
26	INV	R/W	0h	Data Inversion Control Selection. This bitfield controls whether the input and output logic of this IOCELL is inverted. Reset type: SYSRST 0h = 0 1h = 1
25	Z1	R/W	0h	HI-Z Output State Control Selection. This bitfield controls whether or not a high output value will tri-state the output (open-drain functionality). Reset type: SYSRST 0h = 0 1h = 1
24-19	RESERVED	R/W	Xh	

Table 14-64. PINCM28 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
18	INENA	R/W	0h	Input Enable Control Selection. This bitfield controls whether or not the input to this IOCELL propagates to the connected peripheral function. Reset type: SYSRST 0h = 0 1h = 1
17	PIPU	R/W	0h	Pull Up Resistor Control. This bitfield controls whether the pull up resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
16	PIPD	R/W	0h	Pull Down Resistor Control. This bitfield controls whether the pull down resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
15-14	RESERVED	R/W	Xh	
13	WAKESTATE	R	0h	This bitfield has the IOPAD WAKEUP signal as a status bit Reset type: SYSRST
12-8	RESERVED	R/W	Xh	
7	PC	R/W	0h	Peripheral Connect bit. This bit indicates if the designated peripheral function is "Connected" Reset type: SYSRST 0h = 0 1h = 1
6-5	RESERVED	R/W	Xh	
4-0	PF	R/W	0h	Peripheral Function selection bits. This bitfield indicates what peripheral function is selected for a specific pin. Reset type: SYSRST 0h = 0 1h = 1 2h = 2 3h = 3 4h = 4 6h = 6 8h = 8 9h = 9 Ah = 10 Bh = 11 Dh = 13 Eh = 14 Fh = 15 10h = 16

30 PINCM29 Register (Offset = 74h) [Reset = 0XXXXXX0h]

PINCM29 is shown in Figure 14-31 and described in Table 14-36.

Return to the Summary Table.

Pin Control Management Register

Figure 14-31. PINCM29 Register

31	30	29	28	27	26	25	24
RESERVED			WCOMP	WUEN	INV	Z1	RESERVED
R/W-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-Xh
23	22	21	20	19	18	17	16
RESERVED					INENA	PIPU	PIPD
R/W-Xh					R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED		WAKESTATE	RESERVED				
R/W-Xh		R-0h	R/W-Xh				
7	6	5	4	3	2	1	0
PC	RESERVED		PF				
R/W-0h	R/W-Xh		R/W-0h				

Table 14-66. PINCM29 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R/W	0h	
28	WCOMP	R/W	0h	Wakeup Compare Value bit. This bitfield indicates on what data match this IOCELL with intiate a Wake from SHUTDOWN when WUEN = 1. Reset type: SYSRST 0h = 0 1h = 1
27	WUEN	R/W	0h	Wakeup Enable bit. This bitfield indicates whether or not the Wake from Shutdown is enabled on this IOCELL. Reset type: SYSRST 0h = 0 1h = 1
26	INV	R/W	0h	Data Inversion Control Selection. This bitfield controls whether the input and output logic of this IOCELL is inverted. Reset type: SYSRST 0h = 0 1h = 1
25	Z1	R/W	0h	HI-Z Output State Control Selection. This bitfield controls whether or not a high output value will tri-state the output (open-drain functionality). Reset type: SYSRST 0h = 0 1h = 1
24-19	RESERVED	R/W	Xh	

Table 14-66. PINCM29 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
18	INENA	R/W	0h	Input Enable Control Selection. This bitfield controls whether or not the input to this IOCELL propagates to the connected peripheral function. Reset type: SYSRST 0h = 0 1h = 1
17	PIPU	R/W	0h	Pull Up Resistor Control. This bitfield controls whether the pull up resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
16	PIPD	R/W	0h	Pull Down Resistor Control. This bitfield controls whether the pull down resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
15-14	RESERVED	R/W	Xh	
13	WAKESTATE	R	0h	This bitfield has the IOPAD WAKEUP signal as a status bit Reset type: SYSRST
12-8	RESERVED	R/W	Xh	
7	PC	R/W	0h	Peripheral Connect bit. This bit indicates if the designated peripheral function is "Connected" Reset type: SYSRST 0h = 0 1h = 1
6-5	RESERVED	R/W	Xh	
4-0	PF	R/W	0h	Peripheral Function selection bits. This bitfield indicates what peripheral function is selected for a specific pin. Reset type: SYSRST 0h = 0 1h = 1 2h = 2 3h = 3 4h = 4 6h = 6 7h = 7 8h = 8 9h = 9 Bh = 11 Dh = 13 Eh = 14 Fh = 15 10h = 16

31 PINCM30 Register (Offset = 78h) [Reset = 0XXXXXX0h]

PINCM30 is shown in Figure 14-32 and described in Table 14-37.

Return to the Summary Table.

Pin Control Management Register

Figure 14-32. PINCM30 Register

31	30	29	28	27	26	25	24
RESERVED			WCOMP	WUEN	INV	Z1	RESERVED
R/W-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-Xh
23	22	21	20	19	18	17	16
RESERVED					INENA	PIPU	PIPD
R/W-Xh					R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED		WAKESTATE	RESERVED				
R/W-Xh		R-0h	R/W-Xh				
7	6	5	4	3	2	1	0
PC	RESERVED		PF				
R/W-0h	R/W-Xh		R/W-0h				

Table 14-68. PINCM30 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R/W	0h	
28	WCOMP	R/W	0h	Wakeup Compare Value bit. This bitfield indicates on what data match this IOCELL with intiate a Wake from SHUTDOWN when WUEN = 1. Reset type: SYSRST 0h = 0 1h = 1
27	WUEN	R/W	0h	Wakeup Enable bit. This bitfield indicates whether or not the Wake from Shutdown is enabled on this IOCELL. Reset type: SYSRST 0h = 0 1h = 1
26	INV	R/W	0h	Data Inversion Control Selection. This bitfield controls whether the input and output logic of this IOCELL is inverted. Reset type: SYSRST 0h = 0 1h = 1
25	Z1	R/W	0h	HI-Z Output State Control Selection. This bitfield controls whether or not a high output value will tri-state the output (open-drain functionality). Reset type: SYSRST 0h = 0 1h = 1
24-19	RESERVED	R/W	Xh	

Table 14-68. PINCM30 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
18	INENA	R/W	0h	Input Enable Control Selection. This bitfield controls whether or not the input to this IOCELL propagates to the connected peripheral function. Reset type: SYSRST 0h = 0 1h = 1
17	PIPU	R/W	0h	Pull Up Resistor Control. This bitfield controls whether the pull up resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
16	PIPD	R/W	0h	Pull Down Resistor Control. This bitfield controls whether the pull down resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
15-14	RESERVED	R/W	Xh	
13	WAKESTATE	R	0h	This bitfield has the IOPAD WAKEUP signal as a status bit Reset type: SYSRST
12-8	RESERVED	R/W	Xh	
7	PC	R/W	0h	Peripheral Connect bit. This bit indicates if the designated peripheral function is "Connected" Reset type: SYSRST 0h = 0 1h = 1
6-5	RESERVED	R/W	Xh	
4-0	PF	R/W	0h	Peripheral Function selection bits. This bitfield indicates what peripheral function is selected for a specific pin. Reset type: SYSRST 0h = 0 1h = 1 2h = 2 3h = 3 4h = 4 5h = 5 6h = 6 7h = 7 8h = 8 Ah = 10 Bh = 11 Eh = 14 10h = 16

32 PINCM31 Register (Offset = 7Ch) [Reset = 0XXXXXX0h]

PINCM31 is shown in Figure 14-33 and described in Table 14-38.

Return to the Summary Table.

Pin Control Management Register

Figure 14-33. PINCM31 Register

31	30	29	28	27	26	25	24
RESERVED			WCOMP	WUEN	INV	Z1	RESERVED
R/W-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-Xh
23	22	21	20	19	18	17	16
RESERVED					INENA	PIPU	PIPD
R/W-Xh					R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED		WAKESTATE	RESERVED				
R/W-Xh		R-0h	R/W-Xh				
7	6	5	4	3	2	1	0
PC	RESERVED		PF				
R/W-0h	R/W-Xh		R/W-0h				

Table 14-70. PINCM31 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R/W	0h	
28	WCOMP	R/W	0h	Wakeup Compare Value bit. This bitfield indicates on what data match this IOCELL with intiate a Wake from SHUTDOWN when WUEN = 1. Reset type: SYSRST 0h = 0 1h = 1
27	WUEN	R/W	0h	Wakeup Enable bit. This bitfield indicates whether or not the Wake from Shutdown is enabled on this IOCELL. Reset type: SYSRST 0h = 0 1h = 1
26	INV	R/W	0h	Data Inversion Control Selection. This bitfield controls whether the input and output logic of this IOCELL is inverted. Reset type: SYSRST 0h = 0 1h = 1
25	Z1	R/W	0h	HI-Z Output State Control Selection. This bitfield controls whether or not a high output value will tri-state the output (open-drain functionality). Reset type: SYSRST 0h = 0 1h = 1
24-19	RESERVED	R/W	Xh	

Table 14-70. PINCM31 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
18	INENA	R/W	0h	Input Enable Control Selection. This bitfield controls whether or not the input to this IOCELL propagates to the connected peripheral function. Reset type: SYSRST 0h = 0 1h = 1
17	PIPU	R/W	0h	Pull Up Resistor Control. This bitfield controls whether the pull up resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
16	PIPD	R/W	0h	Pull Down Resistor Control. This bitfield controls whether the pull down resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
15-14	RESERVED	R/W	Xh	
13	WAKESTATE	R	0h	This bitfield has the IOPAD WAKEUP signal as a status bit Reset type: SYSRST
12-8	RESERVED	R/W	Xh	
7	PC	R/W	0h	Peripheral Connect bit. This bit indicates if the designated peripheral function is "Connected" Reset type: SYSRST 0h = 0 1h = 1
6-5	RESERVED	R/W	Xh	
4-0	PF	R/W	0h	Peripheral Function selection bits. This bitfield indicates what peripheral function is selected for a specific pin. Reset type: SYSRST 0h = 0 1h = 1 2h = 2 3h = 3 4h = 4 5h = 5 6h = 6 7h = 7 Bh = 11 Eh = 14 10h = 16

33 PINCM32 Register (Offset = 80h) [Reset = 0XXXXXX0h]

PINCM32 is shown in Figure 14-34 and described in Table 14-39.

Return to the Summary Table.

Pin Control Management Register

Figure 14-34. PINCM32 Register

31	30	29	28	27	26	25	24
RESERVED			WCOMP	WUEN	INV	Z1	RESERVED
R/W-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-Xh
23	22	21	20	19	18	17	16
RESERVED					INENA	PIPU	PIPD
R/W-Xh					R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED		WAKESTATE	RESERVED				
R/W-Xh		R-0h	R/W-Xh				
7	6	5	4	3	2	1	0
PC	RESERVED		PF				
R/W-0h	R/W-Xh		R/W-0h				

Table 14-72. PINCM32 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R/W	0h	
28	WCOMP	R/W	0h	Wakeup Compare Value bit. This bitfield indicates on what data match this IOCELL with intiate a Wake from SHUTDOWN when WUEN = 1. Reset type: SYSRST 0h = 0 1h = 1
27	WUEN	R/W	0h	Wakeup Enable bit. This bitfield indicates whether or not the Wake from Shutdown is enabled on this IOCELL. Reset type: SYSRST 0h = 0 1h = 1
26	INV	R/W	0h	Data Inversion Control Selection. This bitfield controls whether the input and output logic of this IOCELL is inverted. Reset type: SYSRST 0h = 0 1h = 1
25	Z1	R/W	0h	HI-Z Output State Control Selection. This bitfield controls whether or not a high output value will tri-state the output (open-drain functionality). Reset type: SYSRST 0h = 0 1h = 1
24-19	RESERVED	R/W	Xh	

Table 14-72. PINCM32 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
18	INENA	R/W	0h	Input Enable Control Selection. This bitfield controls whether or not the input to this IOCELL propagates to the connected peripheral function. Reset type: SYSRST 0h = 0 1h = 1
17	PIPU	R/W	0h	Pull Up Resistor Control. This bitfield controls whether the pull up resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
16	PIPD	R/W	0h	Pull Down Resistor Control. This bitfield controls whether the pull down resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
15-14	RESERVED	R/W	Xh	
13	WAKESTATE	R	0h	This bitfield has the IOPAD WAKEUP signal as a status bit Reset type: SYSRST
12-8	RESERVED	R/W	Xh	
7	PC	R/W	0h	Peripheral Connect bit. This bit indicates if the designated peripheral function is "Connected" Reset type: SYSRST 0h = 0 1h = 1
6-5	RESERVED	R/W	Xh	
4-0	PF	R/W	0h	Peripheral Function selection bits. This bitfield indicates what peripheral function is selected for a specific pin. Reset type: SYSRST 0h = 0 1h = 1 3h = 3 4h = 4 9h = 9 Bh = 11 10h = 16

34 PINCM33 Register (Offset = 84h) [Reset = 0XXXXXX0h]

PINCM33 is shown in Figure 14-35 and described in Table 14-40.

Return to the Summary Table.

Pin Control Management Register

Figure 14-35. PINCM33 Register

31	30	29	28	27	26	25	24
RESERVED			WCOMP	WUEN	INV	Z1	RESERVED
R/W-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-Xh
23	22	21	20	19	18	17	16
RESERVED					INENA	PIPU	PIPD
R/W-Xh					R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED		WAKESTATE	RESERVED				
R/W-Xh		R-0h	R/W-Xh				
7	6	5	4	3	2	1	0
PC	RESERVED		PF				
R/W-0h	R/W-Xh		R/W-0h				

Table 14-74. PINCM33 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R/W	0h	
28	WCOMP	R/W	0h	Wakeup Compare Value bit. This bitfield indicates on what data match this IOCELL with intiate a Wake from SHUTDOWN when WUEN = 1. Reset type: SYSRST 0h = 0 1h = 1
27	WUEN	R/W	0h	Wakeup Enable bit. This bitfield indicates whether or not the Wake from Shutdown is enabled on this IOCELL. Reset type: SYSRST 0h = 0 1h = 1
26	INV	R/W	0h	Data Inversion Control Selection. This bitfield controls whether the input and output logic of this IOCELL is inverted. Reset type: SYSRST 0h = 0 1h = 1
25	Z1	R/W	0h	HI-Z Output State Control Selection. This bitfield controls whether or not a high output value will tri-state the output (open-drain functionality). Reset type: SYSRST 0h = 0 1h = 1
24-19	RESERVED	R/W	Xh	

Table 14-74. PINCM33 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
18	INENA	R/W	0h	Input Enable Control Selection. This bitfield controls whether or not the input to this IOCELL propagates to the connected peripheral function. Reset type: SYSRST 0h = 0 1h = 1
17	PIPU	R/W	0h	Pull Up Resistor Control. This bitfield controls whether the pull up resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
16	PIPD	R/W	0h	Pull Down Resistor Control. This bitfield controls whether the pull down resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
15-14	RESERVED	R/W	Xh	
13	WAKESTATE	R	0h	This bitfield has the IOPAD WAKEUP signal as a status bit Reset type: SYSRST
12-8	RESERVED	R/W	Xh	
7	PC	R/W	0h	Peripheral Connect bit. This bit indicates if the designated peripheral function is "Connected" Reset type: SYSRST 0h = 0 1h = 1
6-5	RESERVED	R/W	Xh	
4-0	PF	R/W	0h	Peripheral Function selection bits. This bitfield indicates what peripheral function is selected for a specific pin. Reset type: SYSRST 0h = 0 1h = 1 3h = 3 4h = 4 5h = 5 9h = 9 Bh = 11 10h = 16

35 PINCM34 Register (Offset = 88h) [Reset = 0XXXXXX0h]

PINCM34 is shown in Figure 14-36 and described in Table 14-41.

Return to the Summary Table.

Pin Control Management Register

Figure 14-36. PINCM34 Register

31	30	29	28	27	26	25	24
RESERVED			WCOMP	WUEN	INV	Z1	RESERVED
R/W-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-Xh
23	22	21	20	19	18	17	16
RESERVED					INENA	PIPU	PIPD
R/W-Xh					R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED		WAKESTATE	RESERVED				
R/W-Xh		R-0h	R/W-Xh				
7	6	5	4	3	2	1	0
PC	RESERVED		PF				
R/W-0h	R/W-Xh		R/W-0h				

Table 14-76. PINCM34 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R/W	0h	
28	WCOMP	R/W	0h	Wakeup Compare Value bit. This bitfield indicates on what data match this IOCELL with intiate a Wake from SHUTDOWN when WUEN = 1. Reset type: SYSRST 0h = 0 1h = 1
27	WUEN	R/W	0h	Wakeup Enable bit. This bitfield indicates whether or not the Wake from Shutdown is enabled on this IOCELL. Reset type: SYSRST 0h = 0 1h = 1
26	INV	R/W	0h	Data Inversion Control Selection. This bitfield controls whether the input and output logic of this IOCELL is inverted. Reset type: SYSRST 0h = 0 1h = 1
25	Z1	R/W	0h	HI-Z Output State Control Selection. This bitfield controls whether or not a high output value will tri-state the output (open-drain functionality). Reset type: SYSRST 0h = 0 1h = 1
24-19	RESERVED	R/W	Xh	

Table 14-76. PINCM34 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
18	INENA	R/W	0h	Input Enable Control Selection. This bitfield controls whether or not the input to this IOCELL propagates to the connected peripheral function. Reset type: SYSRST 0h = 0 1h = 1
17	PIPU	R/W	0h	Pull Up Resistor Control. This bitfield controls whether the pull up resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
16	PIPD	R/W	0h	Pull Down Resistor Control. This bitfield controls whether the pull down resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
15-14	RESERVED	R/W	Xh	
13	WAKESTATE	R	0h	This bitfield has the IOPAD WAKEUP signal as a status bit Reset type: SYSRST
12-8	RESERVED	R/W	Xh	
7	PC	R/W	0h	Peripheral Connect bit. This bit indicates if the designated peripheral function is "Connected" Reset type: SYSRST 0h = 0 1h = 1
6-5	RESERVED	R/W	Xh	
4-0	PF	R/W	0h	Peripheral Function selection bits. This bitfield indicates what peripheral function is selected for a specific pin. Reset type: SYSRST 0h = 0 1h = 1 3h = 3 4h = 4 6h = 6 7h = 7 8h = 8 10h = 16

36 PINCM35 Register (Offset = 8Ch) [Reset = 0XXXXXX0h]

PINCM35 is shown in Figure 14-37 and described in Table 14-42.

Return to the Summary Table.

Pin Control Management Register

Figure 14-37. PINCM35 Register

31	30	29	28	27	26	25	24
RESERVED			WCOMP	WUEN	INV	Z1	RESERVED
R/W-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-Xh
23	22	21	20	19	18	17	16
RESERVED					INENA	PIPU	PIPD
R/W-Xh					R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED		WAKESTATE	RESERVED				
R/W-Xh		R-0h	R/W-Xh				
7	6	5	4	3	2	1	0
PC	RESERVED		PF				
R/W-0h	R/W-Xh		R/W-0h				

Table 14-78. PINCM35 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R/W	0h	
28	WCOMP	R/W	0h	Wakeup Compare Value bit. This bitfield indicates on what data match this IOCELL with intiate a Wake from SHUTDOWN when WUEN = 1. Reset type: SYSRST 0h = 0 1h = 1
27	WUEN	R/W	0h	Wakeup Enable bit. This bitfield indicates whether or not the Wake from Shutdown is enabled on this IOCELL. Reset type: SYSRST 0h = 0 1h = 1
26	INV	R/W	0h	Data Inversion Control Selection. This bitfield controls whether the input and output logic of this IOCELL is inverted. Reset type: SYSRST 0h = 0 1h = 1
25	Z1	R/W	0h	HI-Z Output State Control Selection. This bitfield controls whether or not a high output value will tri-state the output (open-drain functionality). Reset type: SYSRST 0h = 0 1h = 1
24-19	RESERVED	R/W	Xh	

Table 14-78. PINCM35 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
18	INENA	R/W	0h	Input Enable Control Selection. This bitfield controls whether or not the input to this IOCELL propagates to the connected peripheral function. Reset type: SYSRST 0h = 0 1h = 1
17	PIPU	R/W	0h	Pull Up Resistor Control. This bitfield controls whether the pull up resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
16	PIPD	R/W	0h	Pull Down Resistor Control. This bitfield controls whether the pull down resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
15-14	RESERVED	R/W	Xh	
13	WAKESTATE	R	0h	This bitfield has the IOPAD WAKEUP signal as a status bit Reset type: SYSRST
12-8	RESERVED	R/W	Xh	
7	PC	R/W	0h	Peripheral Connect bit. This bit indicates if the designated peripheral function is "Connected" Reset type: SYSRST 0h = 0 1h = 1
6-5	RESERVED	R/W	Xh	
4-0	PF	R/W	0h	Peripheral Function selection bits. This bitfield indicates what peripheral function is selected for a specific pin. Reset type: SYSRST 0h = 0 1h = 1 3h = 3 4h = 4 10h = 16

37 PINCM36 Register (Offset = 90h) [Reset = 0XXXXXX0h]

PINCM36 is shown in Figure 14-38 and described in Table 14-43.

Return to the Summary Table.

Pin Control Management Register

Figure 14-38. PINCM36 Register

31	30	29	28	27	26	25	24
RESERVED			WCOMP	WUEN	INV	Z1	RESERVED
R/W-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-Xh
23	22	21	20	19	18	17	16
RESERVED					INENA	PIPU	PIPD
R/W-Xh					R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED		WAKESTATE	RESERVED				
R/W-Xh		R-0h	R/W-Xh				
7	6	5	4	3	2	1	0
PC	RESERVED		PF				
R/W-0h	R/W-Xh		R/W-0h				

Table 14-80. PINCM36 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R/W	0h	
28	WCOMP	R/W	0h	Wakeup Compare Value bit. This bitfield indicates on what data match this IOCELL with intiate a Wake from SHUTDOWN when WUEN = 1. Reset type: SYSRST 0h = 0 1h = 1
27	WUEN	R/W	0h	Wakeup Enable bit. This bitfield indicates whether or not the Wake from Shutdown is enabled on this IOCELL. Reset type: SYSRST 0h = 0 1h = 1
26	INV	R/W	0h	Data Inversion Control Selection. This bitfield controls whether the input and output logic of this IOCELL is inverted. Reset type: SYSRST 0h = 0 1h = 1
25	Z1	R/W	0h	HI-Z Output State Control Selection. This bitfield controls whether or not a high output value will tri-state the output (open-drain functionality). Reset type: SYSRST 0h = 0 1h = 1
24-19	RESERVED	R/W	Xh	

Table 14-80. PINCM36 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
18	INENA	R/W	0h	Input Enable Control Selection. This bitfield controls whether or not the input to this IOCELL propagates to the connected peripheral function. Reset type: SYSRST 0h = 0 1h = 1
17	PIPU	R/W	0h	Pull Up Resistor Control. This bitfield controls whether the pull up resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
16	PIPD	R/W	0h	Pull Down Resistor Control. This bitfield controls whether the pull down resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
15-14	RESERVED	R/W	Xh	
13	WAKESTATE	R	0h	This bitfield has the IOPAD WAKEUP signal as a status bit Reset type: SYSRST
12-8	RESERVED	R/W	Xh	
7	PC	R/W	0h	Peripheral Connect bit. This bit indicates if the designated peripheral function is "Connected" Reset type: SYSRST 0h = 0 1h = 1
6-5	RESERVED	R/W	Xh	
4-0	PF	R/W	0h	Peripheral Function selection bits. This bitfield indicates what peripheral function is selected for a specific pin. Reset type: SYSRST 0h = 0 1h = 1 5h = 5 8h = 8 9h = 9 Ah = 10 Bh = 11 10h = 16

38 PINCM37 Register (Offset = 94h) [Reset = 0XXXXXX0h]

PINCM37 is shown in Figure 14-39 and described in Table 14-44.

Return to the Summary Table.

Pin Control Management Register

Figure 14-39. PINCM37 Register

31	30	29	28	27	26	25	24
RESERVED			WCOMP	WUEN	INV	Z1	RESERVED
R/W-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-Xh
23	22	21	20	19	18	17	16
RESERVED					INENA	PIPU	PIPD
R/W-Xh					R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED		WAKESTATE	RESERVED				
R/W-Xh		R-0h	R/W-Xh				
7	6	5	4	3	2	1	0
PC	RESERVED		PF				
R/W-0h	R/W-Xh		R/W-0h				

Table 14-82. PINCM37 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R/W	0h	
28	WCOMP	R/W	0h	Wakeup Compare Value bit. This bitfield indicates on what data match this IOCELL with intiate a Wake from SHUTDOWN when WUEN = 1. Reset type: SYSRST 0h = 0 1h = 1
27	WUEN	R/W	0h	Wakeup Enable bit. This bitfield indicates whether or not the Wake from Shutdown is enabled on this IOCELL. Reset type: SYSRST 0h = 0 1h = 1
26	INV	R/W	0h	Data Inversion Control Selection. This bitfield controls whether the input and output logic of this IOCELL is inverted. Reset type: SYSRST 0h = 0 1h = 1
25	Z1	R/W	0h	HI-Z Output State Control Selection. This bitfield controls whether or not a high output value will tri-state the output (open-drain functionality). Reset type: SYSRST 0h = 0 1h = 1
24-19	RESERVED	R/W	Xh	

Table 14-82. PINCM37 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
18	INENA	R/W	0h	Input Enable Control Selection. This bitfield controls whether or not the input to this IOCELL propagates to the connected peripheral function. Reset type: SYSRST 0h = 0 1h = 1
17	PIPU	R/W	0h	Pull Up Resistor Control. This bitfield controls whether the pull up resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
16	PIPD	R/W	0h	Pull Down Resistor Control. This bitfield controls whether the pull down resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
15-14	RESERVED	R/W	Xh	
13	WAKESTATE	R	0h	This bitfield has the IOPAD WAKEUP signal as a status bit Reset type: SYSRST
12-8	RESERVED	R/W	Xh	
7	PC	R/W	0h	Peripheral Connect bit. This bit indicates if the designated peripheral function is "Connected" Reset type: SYSRST 0h = 0 1h = 1
6-5	RESERVED	R/W	Xh	
4-0	PF	R/W	0h	Peripheral Function selection bits. This bitfield indicates what peripheral function is selected for a specific pin. Reset type: SYSRST 0h = 0 1h = 1 6h = 6 7h = 7 8h = 8 9h = 9 Ah = 10 10h = 16

39 PINCM38 Register (Offset = 98h) [Reset = 0XXXXXX0h]

PINCM38 is shown in Figure 14-40 and described in Table 14-45.

Return to the Summary Table.

Pin Control Management Register

Figure 14-40. PINCM38 Register

31	30	29	28	27	26	25	24
RESERVED			WCOMP	WUEN	INV	Z1	RESERVED
R/W-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-Xh
23	22	21	20	19	18	17	16
RESERVED					INENA	PIPU	PIPD
R/W-Xh					R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED		WAKESTATE	RESERVED				
R/W-Xh		R-0h	R/W-Xh				
7	6	5	4	3	2	1	0
PC	RESERVED		PF				
R/W-0h	R/W-Xh		R/W-0h				

Table 14-84. PINCM38 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R/W	0h	
28	WCOMP	R/W	0h	Wakeup Compare Value bit. This bitfield indicates on what data match this IOCELL with intiate a Wake from SHUTDOWN when WUEN = 1. Reset type: SYSRST 0h = 0 1h = 1
27	WUEN	R/W	0h	Wakeup Enable bit. This bitfield indicates whether or not the Wake from Shutdown is enabled on this IOCELL. Reset type: SYSRST 0h = 0 1h = 1
26	INV	R/W	0h	Data Inversion Control Selection. This bitfield controls whether the input and output logic of this IOCELL is inverted. Reset type: SYSRST 0h = 0 1h = 1
25	Z1	R/W	0h	HI-Z Output State Control Selection. This bitfield controls whether or not a high output value will tri-state the output (open-drain functionality). Reset type: SYSRST 0h = 0 1h = 1
24-19	RESERVED	R/W	Xh	

Table 14-84. PINCM38 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
18	INENA	R/W	0h	Input Enable Control Selection. This bitfield controls whether or not the input to this IOCELL propagates to the connected peripheral function. Reset type: SYSRST 0h = 0 1h = 1
17	PIPU	R/W	0h	Pull Up Resistor Control. This bitfield controls whether the pull up resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
16	PIPD	R/W	0h	Pull Down Resistor Control. This bitfield controls whether the pull down resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
15-14	RESERVED	R/W	Xh	
13	WAKESTATE	R	0h	This bitfield has the IOPAD WAKEUP signal as a status bit Reset type: SYSRST
12-8	RESERVED	R/W	Xh	
7	PC	R/W	0h	Peripheral Connect bit. This bit indicates if the designated peripheral function is "Connected" Reset type: SYSRST 0h = 0 1h = 1
6-5	RESERVED	R/W	Xh	
4-0	PF	R/W	0h	Peripheral Function selection bits. This bitfield indicates what peripheral function is selected for a specific pin. Reset type: SYSRST 0h = 0 1h = 1 3h = 3 5h = 5 6h = 6 8h = 8 9h = 9 Eh = 14 10h = 16

40 PINCM39 Register (Offset = 9Ch) [Reset = 0XXXXXX0h]

PINCM39 is shown in Figure 14-41 and described in Table 14-46.

Return to the Summary Table.

Pin Control Management Register

Figure 14-41. PINCM39 Register

31	30	29	28	27	26	25	24
RESERVED			WCOMP	WUEN	INV	Z1	RESERVED
R/W-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-Xh
23	22	21	20	19	18	17	16
RESERVED					INENA	PIPU	PIPD
R/W-Xh					R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED		WAKESTATE	RESERVED				
R/W-Xh		R-0h	R/W-Xh				
7	6	5	4	3	2	1	0
PC	RESERVED		PF				
R/W-0h	R/W-Xh		R/W-0h				

Table 14-86. PINCM39 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R/W	0h	
28	WCOMP	R/W	0h	Wakeup Compare Value bit. This bitfield indicates on what data match this IOCELL with intiate a Wake from SHUTDOWN when WUEN = 1. Reset type: SYSRST 0h = 0 1h = 1
27	WUEN	R/W	0h	Wakeup Enable bit. This bitfield indicates whether or not the Wake from Shutdown is enabled on this IOCELL. Reset type: SYSRST 0h = 0 1h = 1
26	INV	R/W	0h	Data Inversion Control Selection. This bitfield controls whether the input and output logic of this IOCELL is inverted. Reset type: SYSRST 0h = 0 1h = 1
25	Z1	R/W	0h	HI-Z Output State Control Selection. This bitfield controls whether or not a high output value will tri-state the output (open-drain functionality). Reset type: SYSRST 0h = 0 1h = 1
24-19	RESERVED	R/W	Xh	

Table 14-86. PINCM39 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
18	INENA	R/W	0h	Input Enable Control Selection. This bitfield controls whether or not the input to this IOCELL propagates to the connected peripheral function. Reset type: SYSRST 0h = 0 1h = 1
17	PIPU	R/W	0h	Pull Up Resistor Control. This bitfield controls whether the pull up resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
16	PIPD	R/W	0h	Pull Down Resistor Control. This bitfield controls whether the pull down resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
15-14	RESERVED	R/W	Xh	
13	WAKESTATE	R	0h	This bitfield has the IOPAD WAKEUP signal as a status bit Reset type: SYSRST
12-8	RESERVED	R/W	Xh	
7	PC	R/W	0h	Peripheral Connect bit. This bit indicates if the designated peripheral function is "Connected" Reset type: SYSRST 0h = 0 1h = 1
6-5	RESERVED	R/W	Xh	
4-0	PF	R/W	0h	Peripheral Function selection bits. This bitfield indicates what peripheral function is selected for a specific pin. Reset type: SYSRST 0h = 0 1h = 1 3h = 3 4h = 4 5h = 5 8h = 8 9h = 9 10h = 16

41 PINCM40 Register (Offset = A0h) [Reset = 0XXXXXX0h]

PINCM40 is shown in Figure 14-42 and described in Table 14-47.

Return to the Summary Table.

Pin Control Management Register

Figure 14-42. PINCM40 Register

31	30	29	28	27	26	25	24
RESERVED			WCOMP	WUEN	INV	Z1	RESERVED
R/W-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-Xh
23	22	21	20	19	18	17	16
RESERVED					INENA	PIPU	PIPD
R/W-Xh					R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED		WAKESTATE	RESERVED				
R/W-Xh		R-0h	R/W-Xh				
7	6	5	4	3	2	1	0
PC	RESERVED		PF				
R/W-0h	R/W-Xh		R/W-0h				

Table 14-88. PINCM40 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R/W	0h	
28	WCOMP	R/W	0h	Wakeup Compare Value bit. This bitfield indicates on what data match this IOCELL with intiate a Wake from SHUTDOWN when WUEN = 1. Reset type: SYSRST 0h = 0 1h = 1
27	WUEN	R/W	0h	Wakeup Enable bit. This bitfield indicates whether or not the Wake from Shutdown is enabled on this IOCELL. Reset type: SYSRST 0h = 0 1h = 1
26	INV	R/W	0h	Data Inversion Control Selection. This bitfield controls whether the input and output logic of this IOCELL is inverted. Reset type: SYSRST 0h = 0 1h = 1
25	Z1	R/W	0h	HI-Z Output State Control Selection. This bitfield controls whether or not a high output value will tri-state the output (open-drain functionality). Reset type: SYSRST 0h = 0 1h = 1
24-19	RESERVED	R/W	Xh	

Table 14-88. PINCM40 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
18	INENA	R/W	0h	Input Enable Control Selection. This bitfield controls whether or not the input to this IOCELL propagates to the connected peripheral function. Reset type: SYSRST 0h = 0 1h = 1
17	PIPU	R/W	0h	Pull Up Resistor Control. This bitfield controls whether the pull up resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
16	PIPD	R/W	0h	Pull Down Resistor Control. This bitfield controls whether the pull down resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
15-14	RESERVED	R/W	Xh	
13	WAKESTATE	R	0h	This bitfield has the IOPAD WAKEUP signal as a status bit Reset type: SYSRST
12-8	RESERVED	R/W	Xh	
7	PC	R/W	0h	Peripheral Connect bit. This bit indicates if the designated peripheral function is "Connected" Reset type: SYSRST 0h = 0 1h = 1
6-5	RESERVED	R/W	Xh	
4-0	PF	R/W	0h	Peripheral Function selection bits. This bitfield indicates what peripheral function is selected for a specific pin. Reset type: SYSRST 0h = 0 1h = 1 2h = 2 3h = 3 4h = 4 5h = 5 6h = 6 7h = 7 8h = 8 9h = 9 Ah = 10 Bh = 11 Dh = 13 Eh = 14 Fh = 15 10h = 16

42 PINCM41 Register (Offset = A4h) [Reset = 0XXXXXX0h]

PINCM41 is shown in Figure 14-43 and described in Table 14-48.

Return to the Summary Table.

Pin Control Management Register

Figure 14-43. PINCM41 Register

31	30	29	28	27	26	25	24
RESERVED			WCOMP	WUEN	INV	Z1	RESERVED
R/W-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-Xh
23	22	21	20	19	18	17	16
RESERVED					INENA	PIPU	PIPD
R/W-Xh					R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED		WAKESTATE	RESERVED				
R/W-Xh		R-0h	R/W-Xh				
7	6	5	4	3	2	1	0
PC	RESERVED		PF				
R/W-0h	R/W-Xh		R/W-0h				

Table 14-90. PINCM41 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R/W	0h	
28	WCOMP	R/W	0h	Wakeup Compare Value bit. This bitfield indicates on what data match this IOCELL with intiate a Wake from SHUTDOWN when WUEN = 1. Reset type: SYSRST 0h = 0 1h = 1
27	WUEN	R/W	0h	Wakeup Enable bit. This bitfield indicates whether or not the Wake from Shutdown is enabled on this IOCELL. Reset type: SYSRST 0h = 0 1h = 1
26	INV	R/W	0h	Data Inversion Control Selection. This bitfield controls whether the input and output logic of this IOCELL is inverted. Reset type: SYSRST 0h = 0 1h = 1
25	Z1	R/W	0h	HI-Z Output State Control Selection. This bitfield controls whether or not a high output value will tri-state the output (open-drain functionality). Reset type: SYSRST 0h = 0 1h = 1
24-19	RESERVED	R/W	Xh	

Table 14-90. PINCM41 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
18	INENA	R/W	0h	Input Enable Control Selection. This bitfield controls whether or not the input to this IOCELL propagates to the connected peripheral function. Reset type: SYSRST 0h = 0 1h = 1
17	PIPU	R/W	0h	Pull Up Resistor Control. This bitfield controls whether the pull up resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
16	PIPD	R/W	0h	Pull Down Resistor Control. This bitfield controls whether the pull down resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
15-14	RESERVED	R/W	Xh	
13	WAKESTATE	R	0h	This bitfield has the IOPAD WAKEUP signal as a status bit Reset type: SYSRST
12-8	RESERVED	R/W	Xh	
7	PC	R/W	0h	Peripheral Connect bit. This bit indicates if the designated peripheral function is "Connected" Reset type: SYSRST 0h = 0 1h = 1
6-5	RESERVED	R/W	Xh	
4-0	PF	R/W	0h	Peripheral Function selection bits. This bitfield indicates what peripheral function is selected for a specific pin. Reset type: SYSRST 0h = 0 1h = 1 3h = 3 4h = 4 5h = 5 6h = 6 7h = 7 8h = 8 9h = 9 Ah = 10 Bh = 11 10h = 16

43 PINCM42 Register (Offset = A8h) [Reset = 0XXXXXX0h]

PINCM42 is shown in Figure 14-44 and described in Table 14-49.

Return to the Summary Table.

Pin Control Management Register

Figure 14-44. PINCM42 Register

31	30	29	28	27	26	25	24
RESERVED			WCOMP	WUEN	INV	Z1	RESERVED
R/W-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-Xh
23	22	21	20	19	18	17	16
RESERVED					INENA	PIPU	PIPD
R/W-Xh					R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED		WAKESTATE	RESERVED				
R/W-Xh		R-0h	R/W-Xh				
7	6	5	4	3	2	1	0
PC	RESERVED		PF				
R/W-0h	R/W-Xh		R/W-0h				

Table 14-92. PINCM42 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R/W	0h	
28	WCOMP	R/W	0h	Wakeup Compare Value bit. This bitfield indicates on what data match this IOCELL with intiate a Wake from SHUTDOWN when WUEN = 1. Reset type: SYSRST 0h = 0 1h = 1
27	WUEN	R/W	0h	Wakeup Enable bit. This bitfield indicates whether or not the Wake from Shutdown is enabled on this IOCELL. Reset type: SYSRST 0h = 0 1h = 1
26	INV	R/W	0h	Data Inversion Control Selection. This bitfield controls whether the input and output logic of this IOCELL is inverted. Reset type: SYSRST 0h = 0 1h = 1
25	Z1	R/W	0h	HI-Z Output State Control Selection. This bitfield controls whether or not a high output value will tri-state the output (open-drain functionality). Reset type: SYSRST 0h = 0 1h = 1
24-19	RESERVED	R/W	Xh	

Table 14-92. PINCM42 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
18	INENA	R/W	0h	Input Enable Control Selection. This bitfield controls whether or not the input to this IOCELL propagates to the connected peripheral function. Reset type: SYSRST 0h = 0 1h = 1
17	PIPU	R/W	0h	Pull Up Resistor Control. This bitfield controls whether the pull up resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
16	PIPD	R/W	0h	Pull Down Resistor Control. This bitfield controls whether the pull down resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
15-14	RESERVED	R/W	Xh	
13	WAKESTATE	R	0h	This bitfield has the IOPAD WAKEUP signal as a status bit Reset type: SYSRST
12-8	RESERVED	R/W	Xh	
7	PC	R/W	0h	Peripheral Connect bit. This bit indicates if the designated peripheral function is "Connected" Reset type: SYSRST 0h = 0 1h = 1
6-5	RESERVED	R/W	Xh	
4-0	PF	R/W	0h	Peripheral Function selection bits. This bitfield indicates what peripheral function is selected for a specific pin. Reset type: SYSRST 0h = 0 1h = 1 4h = 4 5h = 5 6h = 6 7h = 7 8h = 8 9h = 9 Ah = 10 Bh = 11 10h = 16

44 PINCM43 Register (Offset = ACh) [Reset = 0XXXXXX0h]

PINCM43 is shown in Figure 14-45 and described in Table 14-50.

Return to the Summary Table.

Pin Control Management Register

Figure 14-45. PINCM43 Register

31	30	29	28	27	26	25	24
RESERVED			WCOMP	WUEN	INV	Z1	RESERVED
R/W-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-Xh
23	22	21	20	19	18	17	16
RESERVED					INENA	PIPU	PIPD
R/W-Xh					R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED		WAKESTATE	RESERVED				
R/W-Xh		R-0h	R/W-Xh				
7	6	5	4	3	2	1	0
PC	RESERVED		PF				
R/W-0h	R/W-Xh		R/W-0h				

Table 14-94. PINCM43 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R/W	0h	
28	WCOMP	R/W	0h	Wakeup Compare Value bit. This bitfield indicates on what data match this IOCELL with intiate a Wake from SHUTDOWN when WUEN = 1. Reset type: SYSRST 0h = 0 1h = 1
27	WUEN	R/W	0h	Wakeup Enable bit. This bitfield indicates whether or not the Wake from Shutdown is enabled on this IOCELL. Reset type: SYSRST 0h = 0 1h = 1
26	INV	R/W	0h	Data Inversion Control Selection. This bitfield controls whether the input and output logic of this IOCELL is inverted. Reset type: SYSRST 0h = 0 1h = 1
25	Z1	R/W	0h	HI-Z Output State Control Selection. This bitfield controls whether or not a high output value will tri-state the output (open-drain functionality). Reset type: SYSRST 0h = 0 1h = 1
24-19	RESERVED	R/W	Xh	

ADVANCE INFORMATION

Table 14-94. PINCM43 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
18	INENA	R/W	0h	Input Enable Control Selection. This bitfield controls whether or not the input to this IOCELL propagates to the connected peripheral function. Reset type: SYSRST 0h = 0 1h = 1
17	PIPU	R/W	0h	Pull Up Resistor Control. This bitfield controls whether the pull up resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
16	PIPD	R/W	0h	Pull Down Resistor Control. This bitfield controls whether the pull down resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
15-14	RESERVED	R/W	Xh	
13	WAKESTATE	R	0h	This bitfield has the IOPAD WAKEUP signal as a status bit Reset type: SYSRST
12-8	RESERVED	R/W	Xh	
7	PC	R/W	0h	Peripheral Connect bit. This bit indicates if the designated peripheral function is "Connected" Reset type: SYSRST 0h = 0 1h = 1
6-5	RESERVED	R/W	Xh	
4-0	PF	R/W	0h	Peripheral Function selection bits. This bitfield indicates what peripheral function is selected for a specific pin. Reset type: SYSRST 0h = 0 1h = 1 2h = 2 4h = 4 5h = 5 6h = 6 7h = 7 8h = 8 9h = 9 Ah = 10 Bh = 11 Dh = 13 10h = 16

45 PINCM44 Register (Offset = B0h) [Reset = 0XXXXXX0h]

PINCM44 is shown in Figure 14-46 and described in Table 14-51.

Return to the Summary Table.

Pin Control Management Register

Figure 14-46. PINCM44 Register

31	30	29	28	27	26	25	24
RESERVED			WCOMP	WUEN	INV	Z1	RESERVED
R/W-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-Xh
23	22	21	20	19	18	17	16
RESERVED					INENA	PIPU	PIPD
R/W-Xh					R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED		WAKESTATE	RESERVED				
R/W-Xh		R-0h	R/W-Xh				
7	6	5	4	3	2	1	0
PC	RESERVED		PF				
R/W-0h	R/W-Xh		R/W-0h				

Table 14-96. PINCM44 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R/W	0h	
28	WCOMP	R/W	0h	Wakeup Compare Value bit. This bitfield indicates on what data match this IOCELL with intiate a Wake from SHUTDOWN when WUEN = 1. Reset type: SYSRST 0h = 0 1h = 1
27	WUEN	R/W	0h	Wakeup Enable bit. This bitfield indicates whether or not the Wake from Shutdown is enabled on this IOCELL. Reset type: SYSRST 0h = 0 1h = 1
26	INV	R/W	0h	Data Inversion Control Selection. This bitfield controls whether the input and output logic of this IOCELL is inverted. Reset type: SYSRST 0h = 0 1h = 1
25	Z1	R/W	0h	HI-Z Output State Control Selection. This bitfield controls whether or not a high output value will tri-state the output (open-drain functionality). Reset type: SYSRST 0h = 0 1h = 1
24-19	RESERVED	R/W	Xh	

Table 14-96. PINCM44 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
18	INENA	R/W	0h	Input Enable Control Selection. This bitfield controls whether or not the input to this IOCELL propagates to the connected peripheral function. Reset type: SYSRST 0h = 0 1h = 1
17	PIPU	R/W	0h	Pull Up Resistor Control. This bitfield controls whether the pull up resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
16	PIPD	R/W	0h	Pull Down Resistor Control. This bitfield controls whether the pull down resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
15-14	RESERVED	R/W	Xh	
13	WAKESTATE	R	0h	This bitfield has the IOPAD WAKEUP signal as a status bit Reset type: SYSRST
12-8	RESERVED	R/W	Xh	
7	PC	R/W	0h	Peripheral Connect bit. This bit indicates if the designated peripheral function is "Connected" Reset type: SYSRST 0h = 0 1h = 1
6-5	RESERVED	R/W	Xh	
4-0	PF	R/W	0h	Peripheral Function selection bits. This bitfield indicates what peripheral function is selected for a specific pin. Reset type: SYSRST 0h = 0 1h = 1 2h = 2 3h = 3 4h = 4 5h = 5 6h = 6 7h = 7 8h = 8 9h = 9 Bh = 11 Dh = 13 10h = 16

46 PINCM45 Register (Offset = B4h) [Reset = 0XXXXXX0h]

PINCM45 is shown in Figure 14-47 and described in Table 14-52.

Return to the Summary Table.

Pin Control Management Register

Figure 14-47. PINCM45 Register

31	30	29	28	27	26	25	24
RESERVED			WCOMP	WUEN	INV	Z1	RESERVED
R/W-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-Xh
23	22	21	20	19	18	17	16
RESERVED					INENA	PIPU	PIPD
R/W-Xh					R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED		WAKESTATE	RESERVED				
R/W-Xh		R-0h	R/W-Xh				
7	6	5	4	3	2	1	0
PC	RESERVED		PF				
R/W-0h	R/W-Xh		R/W-0h				

Table 14-98. PINCM45 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R/W	0h	
28	WCOMP	R/W	0h	Wakeup Compare Value bit. This bitfield indicates on what data match this IOCELL with intiate a Wake from SHUTDOWN when WUEN = 1. Reset type: SYSRST 0h = 0 1h = 1
27	WUEN	R/W	0h	Wakeup Enable bit. This bitfield indicates whether or not the Wake from Shutdown is enabled on this IOCELL. Reset type: SYSRST 0h = 0 1h = 1
26	INV	R/W	0h	Data Inversion Control Selection. This bitfield controls whether the input and output logic of this IOCELL is inverted. Reset type: SYSRST 0h = 0 1h = 1
25	Z1	R/W	0h	HI-Z Output State Control Selection. This bitfield controls whether or not a high output value will tri-state the output (open-drain functionality). Reset type: SYSRST 0h = 0 1h = 1
24-19	RESERVED	R/W	Xh	

Table 14-98. PINCM45 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
18	INENA	R/W	0h	Input Enable Control Selection. This bitfield controls whether or not the input to this IOCELL propagates to the connected peripheral function. Reset type: SYSRST 0h = 0 1h = 1
17	PIPU	R/W	0h	Pull Up Resistor Control. This bitfield controls whether the pull up resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
16	PIPD	R/W	0h	Pull Down Resistor Control. This bitfield controls whether the pull down resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
15-14	RESERVED	R/W	Xh	
13	WAKESTATE	R	0h	This bitfield has the IOPAD WAKEUP signal as a status bit Reset type: SYSRST
12-8	RESERVED	R/W	Xh	
7	PC	R/W	0h	Peripheral Connect bit. This bit indicates if the designated peripheral function is "Connected" Reset type: SYSRST 0h = 0 1h = 1
6-5	RESERVED	R/W	Xh	
4-0	PF	R/W	0h	Peripheral Function selection bits. This bitfield indicates what peripheral function is selected for a specific pin. Reset type: SYSRST 0h = 0 1h = 1 2h = 2 5h = 5 6h = 6 7h = 7 8h = 8 9h = 9 10h = 16

47 PINCM46 Register (Offset = B8h) [Reset = 0XXXXXX0h]

PINCM46 is shown in Figure 14-48 and described in Table 14-53.

Return to the Summary Table.

Pin Control Management Register

Figure 14-48. PINCM46 Register

31	30	29	28	27	26	25	24
RESERVED			WCOMP	WUEN	INV	Z1	RESERVED
R/W-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-Xh
23	22	21	20	19	18	17	16
RESERVED					INENA	PIPU	PIPD
R/W-Xh					R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED		WAKESTATE	RESERVED				
R/W-Xh		R-0h	R/W-Xh				
7	6	5	4	3	2	1	0
PC	RESERVED		PF				
R/W-0h	R/W-Xh		R/W-0h				

Table 14-100. PINCM46 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R/W	0h	
28	WCOMP	R/W	0h	Wakeup Compare Value bit. This bitfield indicates on what data match this IOCELL with intiate a Wake from SHUTDOWN when WUEN = 1. Reset type: SYSRST 0h = 0 1h = 1
27	WUEN	R/W	0h	Wakeup Enable bit. This bitfield indicates whether or not the Wake from Shutdown is enabled on this IOCELL. Reset type: SYSRST 0h = 0 1h = 1
26	INV	R/W	0h	Data Inversion Control Selection. This bitfield controls whether the input and output logic of this IOCELL is inverted. Reset type: SYSRST 0h = 0 1h = 1
25	Z1	R/W	0h	HI-Z Output State Control Selection. This bitfield controls whether or not a high output value will tri-state the output (open-drain functionality). Reset type: SYSRST 0h = 0 1h = 1
24-19	RESERVED	R/W	Xh	

Table 14-100. PINCM46 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
18	INENA	R/W	0h	Input Enable Control Selection. This bitfield controls whether or not the input to this IOCELL propagates to the connected peripheral function. Reset type: SYSRST 0h = 0 1h = 1
17	PIPU	R/W	0h	Pull Up Resistor Control. This bitfield controls whether the pull up resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
16	PIPD	R/W	0h	Pull Down Resistor Control. This bitfield controls whether the pull down resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
15-14	RESERVED	R/W	Xh	
13	WAKESTATE	R	0h	This bitfield has the IOPAD WAKEUP signal as a status bit Reset type: SYSRST
12-8	RESERVED	R/W	Xh	
7	PC	R/W	0h	Peripheral Connect bit. This bit indicates if the designated peripheral function is "Connected" Reset type: SYSRST 0h = 0 1h = 1
6-5	RESERVED	R/W	Xh	
4-0	PF	R/W	0h	Peripheral Function selection bits. This bitfield indicates what peripheral function is selected for a specific pin. Reset type: SYSRST 0h = 0 1h = 1 2h = 2 8h = 8 Ah = 10 Bh = 11 10h = 16

48 PINCM47 Register (Offset = BCh) [Reset = 0XXXXXX0h]

PINCM47 is shown in Figure 14-49 and described in Table 14-54.

Return to the Summary Table.

Pin Control Management Register

Figure 14-49. PINCM47 Register

31	30	29	28	27	26	25	24
RESERVED			WCOMP	WUEN	INV	Z1	RESERVED
R/W-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-Xh
23	22	21	20	19	18	17	16
RESERVED					INENA	PIPU	PIPD
R/W-Xh					R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED		WAKESTATE	RESERVED				
R/W-Xh		R-0h	R/W-Xh				
7	6	5	4	3	2	1	0
PC	RESERVED		PF				
R/W-0h	R/W-Xh		R/W-0h				

Table 14-102. PINCM47 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R/W	0h	
28	WCOMP	R/W	0h	Wakeup Compare Value bit. This bitfield indicates on what data match this IOCELL with intiate a Wake from SHUTDOWN when WUEN = 1. Reset type: SYSRST 0h = 0 1h = 1
27	WUEN	R/W	0h	Wakeup Enable bit. This bitfield indicates whether or not the Wake from Shutdown is enabled on this IOCELL. Reset type: SYSRST 0h = 0 1h = 1
26	INV	R/W	0h	Data Inversion Control Selection. This bitfield controls whether the input and output logic of this IOCELL is inverted. Reset type: SYSRST 0h = 0 1h = 1
25	Z1	R/W	0h	HI-Z Output State Control Selection. This bitfield controls whether or not a high output value will tri-state the output (open-drain functionality). Reset type: SYSRST 0h = 0 1h = 1
24-19	RESERVED	R/W	Xh	

Table 14-102. PINCM47 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
18	INENA	R/W	0h	Input Enable Control Selection. This bitfield controls whether or not the input to this IOCELL propagates to the connected peripheral function. Reset type: SYSRST 0h = 0 1h = 1
17	PIPU	R/W	0h	Pull Up Resistor Control. This bitfield controls whether the pull up resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
16	PIPD	R/W	0h	Pull Down Resistor Control. This bitfield controls whether the pull down resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
15-14	RESERVED	R/W	Xh	
13	WAKESTATE	R	0h	This bitfield has the IOPAD WAKEUP signal as a status bit Reset type: SYSRST
12-8	RESERVED	R/W	Xh	
7	PC	R/W	0h	Peripheral Connect bit. This bit indicates if the designated peripheral function is "Connected" Reset type: SYSRST 0h = 0 1h = 1
6-5	RESERVED	R/W	Xh	
4-0	PF	R/W	0h	Peripheral Function selection bits. This bitfield indicates what peripheral function is selected for a specific pin. Reset type: SYSRST 0h = 0 1h = 1 8h = 8 Ah = 10 Bh = 11 10h = 16

49 PINCM48 Register (Offset = C0h) [Reset = 0XXXXXX0h]

PINCM48 is shown in Figure 14-50 and described in Table 14-55.

Return to the Summary Table.

Pin Control Management Register

Figure 14-50. PINCM48 Register

31	30	29	28	27	26	25	24
RESERVED			WCOMP	WUEN	INV	Z1	RESERVED
R/W-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-Xh
23	22	21	20	19	18	17	16
RESERVED					INENA	PIPU	PIPD
R/W-Xh					R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED		WAKESTATE	RESERVED				
R/W-Xh		R-0h	R/W-Xh				
7	6	5	4	3	2	1	0
PC	RESERVED		PF				
R/W-0h	R/W-Xh		R/W-0h				

Table 14-104. PINCM48 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R/W	0h	
28	WCOMP	R/W	0h	Wakeup Compare Value bit. This bitfield indicates on what data match this IOCELL with intiate a Wake from SHUTDOWN when WUEN = 1. Reset type: SYSRST 0h = 0 1h = 1
27	WUEN	R/W	0h	Wakeup Enable bit. This bitfield indicates whether or not the Wake from Shutdown is enabled on this IOCELL. Reset type: SYSRST 0h = 0 1h = 1
26	INV	R/W	0h	Data Inversion Control Selection. This bitfield controls whether the input and output logic of this IOCELL is inverted. Reset type: SYSRST 0h = 0 1h = 1
25	Z1	R/W	0h	HI-Z Output State Control Selection. This bitfield controls whether or not a high output value will tri-state the output (open-drain functionality). Reset type: SYSRST 0h = 0 1h = 1
24-19	RESERVED	R/W	Xh	

Table 14-104. PINCM48 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
18	INENA	R/W	0h	Input Enable Control Selection. This bitfield controls whether or not the input to this IOCELL propagates to the connected peripheral function. Reset type: SYSRST 0h = 0 1h = 1
17	PIPU	R/W	0h	Pull Up Resistor Control. This bitfield controls whether the pull up resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
16	PIPD	R/W	0h	Pull Down Resistor Control. This bitfield controls whether the pull down resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
15-14	RESERVED	R/W	Xh	
13	WAKESTATE	R	0h	This bitfield has the IOPAD WAKEUP signal as a status bit Reset type: SYSRST
12-8	RESERVED	R/W	Xh	
7	PC	R/W	0h	Peripheral Connect bit. This bit indicates if the designated peripheral function is "Connected" Reset type: SYSRST 0h = 0 1h = 1
6-5	RESERVED	R/W	Xh	
4-0	PF	R/W	0h	Peripheral Function selection bits. This bitfield indicates what peripheral function is selected for a specific pin. Reset type: SYSRST 0h = 0 1h = 1 4h = 4 7h = 7 8h = 8 Ah = 10 Bh = 11 Dh = 13 10h = 16

50 PINCM49 Register (Offset = C4h) [Reset = 0XXXXXX0h]

PINCM49 is shown in Figure 14-51 and described in Table 14-56.

Return to the Summary Table.

Pin Control Management Register

Figure 14-51. PINCM49 Register

31	30	29	28	27	26	25	24
RESERVED			WCOMP	WUEN	INV	Z1	RESERVED
R/W-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-Xh
23	22	21	20	19	18	17	16
RESERVED					INENA	PIPU	PIPD
R/W-Xh					R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED		WAKESTATE	RESERVED				
R/W-Xh		R-0h	R/W-Xh				
7	6	5	4	3	2	1	0
PC	RESERVED		PF				
R/W-0h	R/W-Xh		R/W-0h				

Table 14-106. PINCM49 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R/W	0h	
28	WCOMP	R/W	0h	Wakeup Compare Value bit. This bitfield indicates on what data match this IOCELL with intiate a Wake from SHUTDOWN when WUEN = 1. Reset type: SYSRST 0h = 0 1h = 1
27	WUEN	R/W	0h	Wakeup Enable bit. This bitfield indicates whether or not the Wake from Shutdown is enabled on this IOCELL. Reset type: SYSRST 0h = 0 1h = 1
26	INV	R/W	0h	Data Inversion Control Selection. This bitfield controls whether the input and output logic of this IOCELL is inverted. Reset type: SYSRST 0h = 0 1h = 1
25	Z1	R/W	0h	HI-Z Output State Control Selection. This bitfield controls whether or not a high output value will tri-state the output (open-drain functionality). Reset type: SYSRST 0h = 0 1h = 1
24-19	RESERVED	R/W	Xh	

Table 14-106. PINCM49 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
18	INENA	R/W	0h	Input Enable Control Selection. This bitfield controls whether or not the input to this IOCELL propagates to the connected peripheral function. Reset type: SYSRST 0h = 0 1h = 1
17	PIPU	R/W	0h	Pull Up Resistor Control. This bitfield controls whether the pull up resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
16	PIPD	R/W	0h	Pull Down Resistor Control. This bitfield controls whether the pull down resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
15-14	RESERVED	R/W	Xh	
13	WAKESTATE	R	0h	This bitfield has the IOPAD WAKEUP signal as a status bit Reset type: SYSRST
12-8	RESERVED	R/W	Xh	
7	PC	R/W	0h	Peripheral Connect bit. This bit indicates if the designated peripheral function is "Connected" Reset type: SYSRST 0h = 0 1h = 1
6-5	RESERVED	R/W	Xh	
4-0	PF	R/W	0h	Peripheral Function selection bits. This bitfield indicates what peripheral function is selected for a specific pin. Reset type: SYSRST 0h = 0 1h = 1 4h = 4 5h = 5 6h = 6 Ah = 10 Bh = 11 Dh = 13 10h = 16

51 PINCM50 Register (Offset = C8h) [Reset = 0XXXXXX0h]

PINCM50 is shown in Figure 14-52 and described in Table 14-57.

Return to the Summary Table.

Pin Control Management Register

Figure 14-52. PINCM50 Register

31	30	29	28	27	26	25	24
RESERVED			WCOMP	WUEN	INV	Z1	RESERVED
R/W-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-Xh
23	22	21	20	19	18	17	16
RESERVED			DRV	RESERVED	INENA	PIPU	PIPD
R/W-Xh			R/W-0h	R/W-Xh	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED		WAKESTATE	RESERVED				
R/W-Xh		R-0h	R/W-Xh				
7	6	5	4	3	2	1	0
PC	RESERVED		PF				
R/W-0h	R/W-Xh		R/W-0h				

Table 14-108. PINCM50 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R/W	0h	
28	WCOMP	R/W	0h	Wakeup Compare Value bit. This bitfield indicates on what data match this IOCELL with intiate a Wake from SHUTDOWN when WUEN = 1. Reset type: SYSRST 0h = 0 1h = 1
27	WUEN	R/W	0h	Wakeup Enable bit. This bitfield indicates whether or not the Wake from Shutdown is enabled on this IOCELL. Reset type: SYSRST 0h = 0 1h = 1
26	INV	R/W	0h	Data Inversion Control Selection. This bitfield controls whether the input and output logic of this IOCELL is inverted. Reset type: SYSRST 0h = 0 1h = 1
25	Z1	R/W	0h	HI-Z Output State Control Selection. This bitfield controls whether or not a high output value will tri-state the output (open-drain functionality). Reset type: SYSRST 0h = 0 1h = 1
24-21	RESERVED	R/W	Xh	

Table 14-108. PINCM50 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
20	DRV	R/W	0h	bit Reset type: SYSRST 0h = 0 1h = 1
19	RESERVED	R/W	Xh	
18	INENA	R/W	0h	Input Enable Control Selection. This bitfield controls whether or not the input to this IOCELL propagates to the connected peripheral function. Reset type: SYSRST 0h = 0 1h = 1
17	PIPU	R/W	0h	Pull Up Resistor Control. This bitfield controls whether the pull up resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
16	PIPD	R/W	0h	Pull Down Resistor Control. This bitfield controls whether the pull down resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
15-14	RESERVED	R/W	Xh	
13	WAKESTATE	R	0h	This bitfield has the IOPAD WAKEUP signal as a status bit Reset type: SYSRST
12-8	RESERVED	R/W	Xh	
7	PC	R/W	0h	Peripheral Connect bit. This bit indicates if the designated peripheral function is "Connected" Reset type: SYSRST 0h = 0 1h = 1
6-5	RESERVED	R/W	Xh	
4-0	PF	R/W	0h	Peripheral Function selection bits. This bitfield indicates what peripheral function is selected for a specific pin. Reset type: SYSRST 0h = 0 1h = 1 2h = 2 4h = 4 6h = 6 7h = 7 8h = 8 9h = 9 Bh = 11 Dh = 13 10h = 16

52 PINCM51 Register (Offset = CCh) [Reset = 0XXXXXX0h]

PINCM51 is shown in Figure 14-53 and described in Table 14-58.

Return to the Summary Table.

Pin Control Management Register

Figure 14-53. PINCM51 Register

31	30	29	28	27	26	25	24
RESERVED			WCOMP	WUEN	INV	Z1	RESERVED
R/W-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-Xh
23	22	21	20	19	18	17	16
RESERVED					INENA	PIPU	PIPD
R/W-Xh					R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED		WAKESTATE	RESERVED				
R/W-Xh		R-0h	R/W-Xh				
7	6	5	4	3	2	1	0
PC	RESERVED		PF				
R/W-0h	R/W-Xh		R/W-0h				

Table 14-110. PINCM51 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R/W	0h	
28	WCOMP	R/W	0h	Wakeup Compare Value bit. This bitfield indicates on what data match this IOCELL with intiate a Wake from SHUTDOWN when WUEN = 1. Reset type: SYSRST 0h = 0 1h = 1
27	WUEN	R/W	0h	Wakeup Enable bit. This bitfield indicates whether or not the Wake from Shutdown is enabled on this IOCELL. Reset type: SYSRST 0h = 0 1h = 1
26	INV	R/W	0h	Data Inversion Control Selection. This bitfield controls whether the input and output logic of this IOCELL is inverted. Reset type: SYSRST 0h = 0 1h = 1
25	Z1	R/W	0h	HI-Z Output State Control Selection. This bitfield controls whether or not a high output value will tri-state the output (open-drain functionality). Reset type: SYSRST 0h = 0 1h = 1
24-19	RESERVED	R/W	Xh	

Table 14-110. PINCM51 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
18	INENA	R/W	0h	Input Enable Control Selection. This bitfield controls whether or not the input to this IOCELL propagates to the connected peripheral function. Reset type: SYSRST 0h = 0 1h = 1
17	PIPU	R/W	0h	Pull Up Resistor Control. This bitfield controls whether the pull up resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
16	PIPD	R/W	0h	Pull Down Resistor Control. This bitfield controls whether the pull down resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
15-14	RESERVED	R/W	Xh	
13	WAKESTATE	R	0h	This bitfield has the IOPAD WAKEUP signal as a status bit Reset type: SYSRST
12-8	RESERVED	R/W	Xh	
7	PC	R/W	0h	Peripheral Connect bit. This bit indicates if the designated peripheral function is "Connected" Reset type: SYSRST 0h = 0 1h = 1
6-5	RESERVED	R/W	Xh	
4-0	PF	R/W	0h	Peripheral Function selection bits. This bitfield indicates what peripheral function is selected for a specific pin. Reset type: SYSRST 0h = 0 1h = 1 3h = 3 4h = 4 8h = 8 Ah = 10 Bh = 11 Dh = 13 10h = 16

53 PINCM52 Register (Offset = D0h) [Reset = 0XXXXXX0h]

PINCM52 is shown in Figure 14-54 and described in Table 14-59.

Return to the Summary Table.

Pin Control Management Register

Figure 14-54. PINCM52 Register

31	30	29	28	27	26	25	24
RESERVED			WCOMP	WUEN	INV	Z1	RESERVED
R/W-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-Xh
23	22	21	20	19	18	17	16
RESERVED					INENA	PIPU	PIPD
R/W-Xh					R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED		WAKESTATE	RESERVED				
R/W-Xh		R-0h	R/W-Xh				
7	6	5	4	3	2	1	0
PC	RESERVED		PF				
R/W-0h	R/W-Xh		R/W-0h				

Table 14-112. PINCM52 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R/W	0h	
28	WCOMP	R/W	0h	Wakeup Compare Value bit. This bitfield indicates on what data match this IOCELL with intiate a Wake from SHUTDOWN when WUEN = 1. Reset type: SYSRST 0h = 0 1h = 1
27	WUEN	R/W	0h	Wakeup Enable bit. This bitfield indicates whether or not the Wake from Shutdown is enabled on this IOCELL. Reset type: SYSRST 0h = 0 1h = 1
26	INV	R/W	0h	Data Inversion Control Selection. This bitfield controls whether the input and output logic of this IOCELL is inverted. Reset type: SYSRST 0h = 0 1h = 1
25	Z1	R/W	0h	HI-Z Output State Control Selection. This bitfield controls whether or not a high output value will tri-state the output (open-drain functionality). Reset type: SYSRST 0h = 0 1h = 1
24-19	RESERVED	R/W	Xh	

Table 14-112. PINCM52 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
18	INENA	R/W	0h	Input Enable Control Selection. This bitfield controls whether or not the input to this IOCELL propagates to the connected peripheral function. Reset type: SYSRST 0h = 0 1h = 1
17	PIPU	R/W	0h	Pull Up Resistor Control. This bitfield controls whether the pull up resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
16	PIPD	R/W	0h	Pull Down Resistor Control. This bitfield controls whether the pull down resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
15-14	RESERVED	R/W	Xh	
13	WAKESTATE	R	0h	This bitfield has the IOPAD WAKEUP signal as a status bit Reset type: SYSRST
12-8	RESERVED	R/W	Xh	
7	PC	R/W	0h	Peripheral Connect bit. This bit indicates if the designated peripheral function is "Connected" Reset type: SYSRST 0h = 0 1h = 1
6-5	RESERVED	R/W	Xh	
4-0	PF	R/W	0h	Peripheral Function selection bits. This bitfield indicates what peripheral function is selected for a specific pin. Reset type: SYSRST 0h = 0 1h = 1 3h = 3 8h = 8 Ah = 10 Dh = 13 10h = 16

54 PINCM53 Register (Offset = D4h) [Reset = 0XXXXXX0h]

PINCM53 is shown in Figure 14-55 and described in Table 14-60.

Return to the Summary Table.

Pin Control Management Register

Figure 14-55. PINCM53 Register

31	30	29	28	27	26	25	24
RESERVED			WCOMP	WUEN	INV	Z1	RESERVED
R/W-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-Xh
23	22	21	20	19	18	17	16
RESERVED					INENA	PIPU	PIPD
R/W-Xh					R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED		WAKESTATE	RESERVED				
R/W-Xh		R-0h	R/W-Xh				
7	6	5	4	3	2	1	0
PC	RESERVED		PF				
R/W-0h	R/W-Xh		R/W-0h				

Table 14-114. PINCM53 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R/W	0h	
28	WCOMP	R/W	0h	Wakeup Compare Value bit. This bitfield indicates on what data match this IOCELL with intiate a Wake from SHUTDOWN when WUEN = 1. Reset type: SYSRST 0h = 0 1h = 1
27	WUEN	R/W	0h	Wakeup Enable bit. This bitfield indicates whether or not the Wake from Shutdown is enabled on this IOCELL. Reset type: SYSRST 0h = 0 1h = 1
26	INV	R/W	0h	Data Inversion Control Selection. This bitfield controls whether the input and output logic of this IOCELL is inverted. Reset type: SYSRST 0h = 0 1h = 1
25	Z1	R/W	0h	HI-Z Output State Control Selection. This bitfield controls whether or not a high output value will tri-state the output (open-drain functionality). Reset type: SYSRST 0h = 0 1h = 1
24-19	RESERVED	R/W	Xh	

Table 14-114. PINCM53 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
18	INENA	R/W	0h	Input Enable Control Selection. This bitfield controls whether or not the input to this IOCELL propagates to the connected peripheral function. Reset type: SYSRST 0h = 0 1h = 1
17	PIPU	R/W	0h	Pull Up Resistor Control. This bitfield controls whether the pull up resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
16	PIPD	R/W	0h	Pull Down Resistor Control. This bitfield controls whether the pull down resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
15-14	RESERVED	R/W	Xh	
13	WAKESTATE	R	0h	This bitfield has the IOPAD WAKEUP signal as a status bit Reset type: SYSRST
12-8	RESERVED	R/W	Xh	
7	PC	R/W	0h	Peripheral Connect bit. This bit indicates if the designated peripheral function is "Connected" Reset type: SYSRST 0h = 0 1h = 1
6-5	RESERVED	R/W	Xh	
4-0	PF	R/W	0h	Peripheral Function selection bits. This bitfield indicates what peripheral function is selected for a specific pin. Reset type: SYSRST 0h = 0 1h = 1 8h = 8 Ah = 10 Dh = 13 10h = 16

55 PINCM54 Register (Offset = D8h) [Reset = 0XXXXXX0h]

PINCM54 is shown in Figure 14-56 and described in Table 14-61.

Return to the Summary Table.

Pin Control Management Register

Figure 14-56. PINCM54 Register

31	30	29	28	27	26	25	24
RESERVED			WCOMP	WUEN	INV	Z1	RESERVED
R/W-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-Xh
23	22	21	20	19	18	17	16
RESERVED					INENA	PIPU	PIPD
R/W-Xh					R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED		WAKESTATE	RESERVED				
R/W-Xh		R-0h	R/W-Xh				
7	6	5	4	3	2	1	0
PC	RESERVED		PF				
R/W-0h	R/W-Xh		R/W-0h				

Table 14-116. PINCM54 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R/W	0h	
28	WCOMP	R/W	0h	Wakeup Compare Value bit. This bitfield indicates on what data match this IOCELL with intiate a Wake from SHUTDOWN when WUEN = 1. Reset type: SYSRST 0h = 0 1h = 1
27	WUEN	R/W	0h	Wakeup Enable bit. This bitfield indicates whether or not the Wake from Shutdown is enabled on this IOCELL. Reset type: SYSRST 0h = 0 1h = 1
26	INV	R/W	0h	Data Inversion Control Selection. This bitfield controls whether the input and output logic of this IOCELL is inverted. Reset type: SYSRST 0h = 0 1h = 1
25	Z1	R/W	0h	HI-Z Output State Control Selection. This bitfield controls whether or not a high output value will tri-state the output (open-drain functionality). Reset type: SYSRST 0h = 0 1h = 1
24-19	RESERVED	R/W	Xh	

Table 14-116. PINCM54 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
18	INENA	R/W	0h	Input Enable Control Selection. This bitfield controls whether or not the input to this IOCELL propagates to the connected peripheral function. Reset type: SYSRST 0h = 0 1h = 1
17	PIPU	R/W	0h	Pull Up Resistor Control. This bitfield controls whether the pull up resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
16	PIPD	R/W	0h	Pull Down Resistor Control. This bitfield controls whether the pull down resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
15-14	RESERVED	R/W	Xh	
13	WAKESTATE	R	0h	This bitfield has the IOPAD WAKEUP signal as a status bit Reset type: SYSRST
12-8	RESERVED	R/W	Xh	
7	PC	R/W	0h	Peripheral Connect bit. This bit indicates if the designated peripheral function is "Connected" Reset type: SYSRST 0h = 0 1h = 1
6-5	RESERVED	R/W	Xh	
4-0	PF	R/W	0h	Peripheral Function selection bits. This bitfield indicates what peripheral function is selected for a specific pin. Reset type: SYSRST 0h = 0 1h = 1 3h = 3 4h = 4 5h = 5 8h = 8 Ah = 10 Dh = 13 10h = 16

56 PINCM55 Register (Offset = DCh) [Reset = 0XXXXXX0h]

PINCM55 is shown in Figure 14-57 and described in Table 14-62.

Return to the Summary Table.

Pin Control Management Register

Figure 14-57. PINCM55 Register

31	30	29	28	27	26	25	24
RESERVED			WCOMP	WUEN	INV	Z1	RESERVED
R/W-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-Xh
23	22	21	20	19	18	17	16
RESERVED					INENA	PIPU	PIPD
R/W-Xh					R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED		WAKESTATE	RESERVED				
R/W-Xh		R-0h	R/W-Xh				
7	6	5	4	3	2	1	0
PC	RESERVED		PF				
R/W-0h	R/W-Xh		R/W-0h				

Table 14-118. PINCM55 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R/W	0h	
28	WCOMP	R/W	0h	Wakeup Compare Value bit. This bitfield indicates on what data match this IOCELL with intiate a Wake from SHUTDOWN when WUEN = 1. Reset type: SYSRST 0h = 0 1h = 1
27	WUEN	R/W	0h	Wakeup Enable bit. This bitfield indicates whether or not the Wake from Shutdown is enabled on this IOCELL. Reset type: SYSRST 0h = 0 1h = 1
26	INV	R/W	0h	Data Inversion Control Selection. This bitfield controls whether the input and output logic of this IOCELL is inverted. Reset type: SYSRST 0h = 0 1h = 1
25	Z1	R/W	0h	HI-Z Output State Control Selection. This bitfield controls whether or not a high output value will tri-state the output (open-drain functionality). Reset type: SYSRST 0h = 0 1h = 1
24-19	RESERVED	R/W	Xh	

Table 14-118. PINCM55 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
18	INENA	R/W	0h	Input Enable Control Selection. This bitfield controls whether or not the input to this IOCELL propagates to the connected peripheral function. Reset type: SYSRST 0h = 0 1h = 1
17	PIPU	R/W	0h	Pull Up Resistor Control. This bitfield controls whether the pull up resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
16	PIPD	R/W	0h	Pull Down Resistor Control. This bitfield controls whether the pull down resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
15-14	RESERVED	R/W	Xh	
13	WAKESTATE	R	0h	This bitfield has the IOPAD WAKEUP signal as a status bit Reset type: SYSRST
12-8	RESERVED	R/W	Xh	
7	PC	R/W	0h	Peripheral Connect bit. This bit indicates if the designated peripheral function is "Connected" Reset type: SYSRST 0h = 0 1h = 1
6-5	RESERVED	R/W	Xh	
4-0	PF	R/W	0h	Peripheral Function selection bits. This bitfield indicates what peripheral function is selected for a specific pin. Reset type: SYSRST 0h = 0 1h = 1 3h = 3 4h = 4 5h = 5 10h = 16

57 PINCM56 Register (Offset = E0h) [Reset = 0XXXXXX0h]

PINCM56 is shown in Figure 14-58 and described in Table 14-63.

Return to the Summary Table.

Pin Control Management Register

Figure 14-58. PINCM56 Register

31	30	29	28	27	26	25	24
RESERVED			WCOMP	WUEN	INV	Z1	RESERVED
R/W-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-Xh
23	22	21	20	19	18	17	16
RESERVED					INENA	PIPU	PIPD
R/W-Xh					R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED		WAKESTATE	RESERVED				
R/W-Xh		R-0h	R/W-Xh				
7	6	5	4	3	2	1	0
PC	RESERVED		PF				
R/W-0h	R/W-Xh		R/W-0h				

Table 14-120. PINCM56 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R/W	0h	
28	WCOMP	R/W	0h	Wakeup Compare Value bit. This bitfield indicates on what data match this IOCELL with intiate a Wake from SHUTDOWN when WUEN = 1. Reset type: SYSRST 0h = 0 1h = 1
27	WUEN	R/W	0h	Wakeup Enable bit. This bitfield indicates whether or not the Wake from Shutdown is enabled on this IOCELL. Reset type: SYSRST 0h = 0 1h = 1
26	INV	R/W	0h	Data Inversion Control Selection. This bitfield controls whether the input and output logic of this IOCELL is inverted. Reset type: SYSRST 0h = 0 1h = 1
25	Z1	R/W	0h	HI-Z Output State Control Selection. This bitfield controls whether or not a high output value will tri-state the output (open-drain functionality). Reset type: SYSRST 0h = 0 1h = 1
24-19	RESERVED	R/W	Xh	

Table 14-120. PINCM56 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
18	INENA	R/W	0h	Input Enable Control Selection. This bitfield controls whether or not the input to this IOCELL propagates to the connected peripheral function. Reset type: SYSRST 0h = 0 1h = 1
17	PIPU	R/W	0h	Pull Up Resistor Control. This bitfield controls whether the pull up resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
16	PIPD	R/W	0h	Pull Down Resistor Control. This bitfield controls whether the pull down resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
15-14	RESERVED	R/W	Xh	
13	WAKESTATE	R	0h	This bitfield has the IOPAD WAKEUP signal as a status bit Reset type: SYSRST
12-8	RESERVED	R/W	Xh	
7	PC	R/W	0h	Peripheral Connect bit. This bit indicates if the designated peripheral function is "Connected" Reset type: SYSRST 0h = 0 1h = 1
6-5	RESERVED	R/W	Xh	
4-0	PF	R/W	0h	Peripheral Function selection bits. This bitfield indicates what peripheral function is selected for a specific pin. Reset type: SYSRST 0h = 0 1h = 1 9h = 9 Bh = 11 Dh = 13 10h = 16

58 PINCM57 Register (Offset = E4h) [Reset = 0XXXXXX0h]

PINCM57 is shown in Figure 14-59 and described in Table 14-64.

Return to the Summary Table.

Pin Control Management Register

Figure 14-59. PINCM57 Register

31	30	29	28	27	26	25	24
RESERVED			WCOMP	WUEN	INV	Z1	RESERVED
R/W-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-Xh
23	22	21	20	19	18	17	16
RESERVED					INENA	PIPU	PIPD
R/W-Xh					R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED		WAKESTATE	RESERVED				
R/W-Xh		R-0h	R/W-Xh				
7	6	5	4	3	2	1	0
PC	RESERVED		PF				
R/W-0h	R/W-Xh		R/W-0h				

Table 14-122. PINCM57 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R/W	0h	
28	WCOMP	R/W	0h	Wakeup Compare Value bit. This bitfield indicates on what data match this IOCELL with intiate a Wake from SHUTDOWN when WUEN = 1. Reset type: SYSRST 0h = 0 1h = 1
27	WUEN	R/W	0h	Wakeup Enable bit. This bitfield indicates whether or not the Wake from Shutdown is enabled on this IOCELL. Reset type: SYSRST 0h = 0 1h = 1
26	INV	R/W	0h	Data Inversion Control Selection. This bitfield controls whether the input and output logic of this IOCELL is inverted. Reset type: SYSRST 0h = 0 1h = 1
25	Z1	R/W	0h	HI-Z Output State Control Selection. This bitfield controls whether or not a high output value will tri-state the output (open-drain functionality). Reset type: SYSRST 0h = 0 1h = 1
24-19	RESERVED	R/W	Xh	

Table 14-122. PINCM57 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
18	INENA	R/W	0h	Input Enable Control Selection. This bitfield controls whether or not the input to this IOCELL propagates to the connected peripheral function. Reset type: SYSRST 0h = 0 1h = 1
17	PIPU	R/W	0h	Pull Up Resistor Control. This bitfield controls whether the pull up resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
16	PIPD	R/W	0h	Pull Down Resistor Control. This bitfield controls whether the pull down resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
15-14	RESERVED	R/W	Xh	
13	WAKESTATE	R	0h	This bitfield has the IOPAD WAKEUP signal as a status bit Reset type: SYSRST
12-8	RESERVED	R/W	Xh	
7	PC	R/W	0h	Peripheral Connect bit. This bit indicates if the designated peripheral function is "Connected" Reset type: SYSRST 0h = 0 1h = 1
6-5	RESERVED	R/W	Xh	
4-0	PF	R/W	0h	Peripheral Function selection bits. This bitfield indicates what peripheral function is selected for a specific pin. Reset type: SYSRST 0h = 0 1h = 1 9h = 9 Bh = 11 Dh = 13 10h = 16

59 PINCM58 Register (Offset = E8h) [Reset = 0XXXXXX0h]

PINCM58 is shown in Figure 14-60 and described in Table 14-65.

Return to the Summary Table.

Pin Control Management Register

Figure 14-60. PINCM58 Register

31	30	29	28	27	26	25	24
RESERVED			WCOMP	WUEN	INV	Z1	RESERVED
R/W-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-Xh
23	22	21	20	19	18	17	16
RESERVED					INENA	PIPU	PIPD
R/W-Xh					R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED		WAKESTATE	RESERVED				
R/W-Xh		R-0h	R/W-Xh				
7	6	5	4	3	2	1	0
PC	RESERVED		PF				
R/W-0h	R/W-Xh		R/W-0h				

Table 14-124. PINCM58 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R/W	0h	
28	WCOMP	R/W	0h	Wakeup Compare Value bit. This bitfield indicates on what data match this IOCELL with intiate a Wake from SHUTDOWN when WUEN = 1. Reset type: SYSRST 0h = 0 1h = 1
27	WUEN	R/W	0h	Wakeup Enable bit. This bitfield indicates whether or not the Wake from Shutdown is enabled on this IOCELL. Reset type: SYSRST 0h = 0 1h = 1
26	INV	R/W	0h	Data Inversion Control Selection. This bitfield controls whether the input and output logic of this IOCELL is inverted. Reset type: SYSRST 0h = 0 1h = 1
25	Z1	R/W	0h	HI-Z Output State Control Selection. This bitfield controls whether or not a high output value will tri-state the output (open-drain functionality). Reset type: SYSRST 0h = 0 1h = 1
24-19	RESERVED	R/W	Xh	

Table 14-124. PINCM58 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
18	INENA	R/W	0h	Input Enable Control Selection. This bitfield controls whether or not the input to this IOCELL propagates to the connected peripheral function. Reset type: SYSRST 0h = 0 1h = 1
17	PIPU	R/W	0h	Pull Up Resistor Control. This bitfield controls whether the pull up resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
16	PIPD	R/W	0h	Pull Down Resistor Control. This bitfield controls whether the pull down resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
15-14	RESERVED	R/W	Xh	
13	WAKESTATE	R	0h	This bitfield has the IOPAD WAKEUP signal as a status bit Reset type: SYSRST
12-8	RESERVED	R/W	Xh	
7	PC	R/W	0h	Peripheral Connect bit. This bit indicates if the designated peripheral function is "Connected" Reset type: SYSRST 0h = 0 1h = 1
6-5	RESERVED	R/W	Xh	
4-0	PF	R/W	0h	Peripheral Function selection bits. This bitfield indicates what peripheral function is selected for a specific pin. Reset type: SYSRST 0h = 0 1h = 1 2h = 2 Dh = 13 10h = 16

60 PINCM59 Register (Offset = ECh) [Reset = 0XXXXXX0h]

PINCM59 is shown in Figure 14-61 and described in Table 14-66.

Return to the Summary Table.

Pin Control Management Register

Figure 14-61. PINCM59 Register

31	30	29	28	27	26	25	24
RESERVED			WCOMP	WUEN	INV	Z1	RESERVED
R/W-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-Xh
23	22	21	20	19	18	17	16
RESERVED					INENA	PIPU	PIPD
R/W-Xh					R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED		WAKESTATE	RESERVED				
R/W-Xh		R-0h	R/W-Xh				
7	6	5	4	3	2	1	0
PC	RESERVED		PF				
R/W-0h	R/W-Xh		R/W-0h				

Table 14-126. PINCM59 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R/W	0h	
28	WCOMP	R/W	0h	Wakeup Compare Value bit. This bitfield indicates on what data match this IOCELL with intiate a Wake from SHUTDOWN when WUEN = 1. Reset type: SYSRST 0h = 0 1h = 1
27	WUEN	R/W	0h	Wakeup Enable bit. This bitfield indicates whether or not the Wake from Shutdown is enabled on this IOCELL. Reset type: SYSRST 0h = 0 1h = 1
26	INV	R/W	0h	Data Inversion Control Selection. This bitfield controls whether the input and output logic of this IOCELL is inverted. Reset type: SYSRST 0h = 0 1h = 1
25	Z1	R/W	0h	HI-Z Output State Control Selection. This bitfield controls whether or not a high output value will tri-state the output (open-drain functionality). Reset type: SYSRST 0h = 0 1h = 1
24-19	RESERVED	R/W	Xh	

Table 14-126. PINCM59 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
18	INENA	R/W	0h	Input Enable Control Selection. This bitfield controls whether or not the input to this IOCELL propagates to the connected peripheral function. Reset type: SYSRST 0h = 0 1h = 1
17	PIPU	R/W	0h	Pull Up Resistor Control. This bitfield controls whether the pull up resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
16	PIPD	R/W	0h	Pull Down Resistor Control. This bitfield controls whether the pull down resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
15-14	RESERVED	R/W	Xh	
13	WAKESTATE	R	0h	This bitfield has the IOPAD WAKEUP signal as a status bit Reset type: SYSRST
12-8	RESERVED	R/W	Xh	
7	PC	R/W	0h	Peripheral Connect bit. This bit indicates if the designated peripheral function is "Connected" Reset type: SYSRST 0h = 0 1h = 1
6-5	RESERVED	R/W	Xh	
4-0	PF	R/W	0h	Peripheral Function selection bits. This bitfield indicates what peripheral function is selected for a specific pin. Reset type: SYSRST 0h = 0 1h = 1 2h = 2 8h = 8 9h = 9 Ah = 10 Bh = 11 Dh = 13 10h = 16

61 PINCM60 Register (Offset = F0h) [Reset = 0XXXXXX0h]

PINCM60 is shown in Figure 14-62 and described in Table 14-67.

Return to the Summary Table.

Pin Control Management Register

Figure 14-62. PINCM60 Register

31	30	29	28	27	26	25	24
RESERVED			WCOMP	WUEN	INV	Z1	RESERVED
R/W-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-Xh
23	22	21	20	19	18	17	16
RESERVED					INENA	PIPU	PIPD
R/W-Xh					R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED		WAKESTATE	RESERVED				
R/W-Xh		R-0h	R/W-Xh				
7	6	5	4	3	2	1	0
PC	RESERVED		PF				
R/W-0h	R/W-Xh		R/W-0h				

Table 14-128. PINCM60 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R/W	0h	
28	WCOMP	R/W	0h	Wakeup Compare Value bit. This bitfield indicates on what data match this IOCELL with intiate a Wake from SHUTDOWN when WUEN = 1. Reset type: SYSRST 0h = 0 1h = 1
27	WUEN	R/W	0h	Wakeup Enable bit. This bitfield indicates whether or not the Wake from Shutdown is enabled on this IOCELL. Reset type: SYSRST 0h = 0 1h = 1
26	INV	R/W	0h	Data Inversion Control Selection. This bitfield controls whether the input and output logic of this IOCELL is inverted. Reset type: SYSRST 0h = 0 1h = 1
25	Z1	R/W	0h	HI-Z Output State Control Selection. This bitfield controls whether or not a high output value will tri-state the output (open-drain functionality). Reset type: SYSRST 0h = 0 1h = 1
24-19	RESERVED	R/W	Xh	

Table 14-128. PINCM60 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
18	INENA	R/W	0h	Input Enable Control Selection. This bitfield controls whether or not the input to this IOCELL propagates to the connected peripheral function. Reset type: SYSRST 0h = 0 1h = 1
17	PIPU	R/W	0h	Pull Up Resistor Control. This bitfield controls whether the pull up resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
16	PIPD	R/W	0h	Pull Down Resistor Control. This bitfield controls whether the pull down resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
15-14	RESERVED	R/W	Xh	
13	WAKESTATE	R	0h	This bitfield has the IOPAD WAKEUP signal as a status bit Reset type: SYSRST
12-8	RESERVED	R/W	Xh	
7	PC	R/W	0h	Peripheral Connect bit. This bit indicates if the designated peripheral function is "Connected" Reset type: SYSRST 0h = 0 1h = 1
6-5	RESERVED	R/W	Xh	
4-0	PF	R/W	0h	Peripheral Function selection bits. This bitfield indicates what peripheral function is selected for a specific pin. Reset type: SYSRST 0h = 0 1h = 1 2h = 2 3h = 3 4h = 4 5h = 5 8h = 8 9h = 9 Ah = 10 Bh = 11 Dh = 13 10h = 16

62 PINCM61 Register (Offset = F4h) [Reset = 0XXXXXX0h]

PINCM61 is shown in Figure 14-63 and described in Table 14-68.

Return to the Summary Table.

Pin Control Management Register

Figure 14-63. PINCM61 Register

31	30	29	28	27	26	25	24
RESERVED			WCOMP	WUEN	INV	Z1	RESERVED
R/W-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-Xh
23	22	21	20	19	18	17	16
RESERVED					INENA	PIPU	PIPD
R/W-Xh					R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED		WAKESTATE	RESERVED				
R/W-Xh		R-0h	R/W-Xh				
7	6	5	4	3	2	1	0
PC	RESERVED		PF				
R/W-0h	R/W-Xh		R/W-0h				

Table 14-130. PINCM61 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R/W	0h	
28	WCOMP	R/W	0h	Wakeup Compare Value bit. This bitfield indicates on what data match this IOCELL with intiate a Wake from SHUTDOWN when WUEN = 1. Reset type: SYSRST 0h = 0 1h = 1
27	WUEN	R/W	0h	Wakeup Enable bit. This bitfield indicates whether or not the Wake from Shutdown is enabled on this IOCELL. Reset type: SYSRST 0h = 0 1h = 1
26	INV	R/W	0h	Data Inversion Control Selection. This bitfield controls whether the input and output logic of this IOCELL is inverted. Reset type: SYSRST 0h = 0 1h = 1
25	Z1	R/W	0h	HI-Z Output State Control Selection. This bitfield controls whether or not a high output value will tri-state the output (open-drain functionality). Reset type: SYSRST 0h = 0 1h = 1
24-19	RESERVED	R/W	Xh	

Table 14-130. PINCM61 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
18	INENA	R/W	0h	Input Enable Control Selection. This bitfield controls whether or not the input to this IOCELL propagates to the connected peripheral function. Reset type: SYSRST 0h = 0 1h = 1
17	PIPU	R/W	0h	Pull Up Resistor Control. This bitfield controls whether the pull up resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
16	PIPD	R/W	0h	Pull Down Resistor Control. This bitfield controls whether the pull down resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
15-14	RESERVED	R/W	Xh	
13	WAKESTATE	R	0h	This bitfield has the IOPAD WAKEUP signal as a status bit Reset type: SYSRST
12-8	RESERVED	R/W	Xh	
7	PC	R/W	0h	Peripheral Connect bit. This bit indicates if the designated peripheral function is "Connected" Reset type: SYSRST 0h = 0 1h = 1
6-5	RESERVED	R/W	Xh	
4-0	PF	R/W	0h	Peripheral Function selection bits. This bitfield indicates what peripheral function is selected for a specific pin. Reset type: SYSRST 0h = 0 1h = 1 2h = 2 3h = 3 4h = 4 5h = 5 6h = 6 8h = 8 Ah = 10 Bh = 11 Dh = 13 10h = 16

63 PINCM62 Register (Offset = F8h) [Reset = 0XXXXXX0h]

PINCM62 is shown in Figure 14-64 and described in Table 14-69.

Return to the Summary Table.

Pin Control Management Register

Figure 14-64. PINCM62 Register

31	30	29	28	27	26	25	24
RESERVED			WCOMP	WUEN	INV	Z1	RESERVED
R/W-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-Xh
23	22	21	20	19	18	17	16
RESERVED					INENA	PIPU	PIPD
R/W-Xh					R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED		WAKESTATE	RESERVED				
R/W-Xh		R-0h	R/W-Xh				
7	6	5	4	3	2	1	0
PC	RESERVED		PF				
R/W-0h	R/W-Xh		R/W-0h				

Table 14-132. PINCM62 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R/W	0h	
28	WCOMP	R/W	0h	Wakeup Compare Value bit. This bitfield indicates on what data match this IOCELL with intiate a Wake from SHUTDOWN when WUEN = 1. Reset type: SYSRST 0h = 0 1h = 1
27	WUEN	R/W	0h	Wakeup Enable bit. This bitfield indicates whether or not the Wake from Shutdown is enabled on this IOCELL. Reset type: SYSRST 0h = 0 1h = 1
26	INV	R/W	0h	Data Inversion Control Selection. This bitfield controls whether the input and output logic of this IOCELL is inverted. Reset type: SYSRST 0h = 0 1h = 1
25	Z1	R/W	0h	HI-Z Output State Control Selection. This bitfield controls whether or not a high output value will tri-state the output (open-drain functionality). Reset type: SYSRST 0h = 0 1h = 1
24-19	RESERVED	R/W	Xh	

Table 14-132. PINCM62 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
18	INENA	R/W	0h	Input Enable Control Selection. This bitfield controls whether or not the input to this IOCELL propagates to the connected peripheral function. Reset type: SYSRST 0h = 0 1h = 1
17	PIPU	R/W	0h	Pull Up Resistor Control. This bitfield controls whether the pull up resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
16	PIPD	R/W	0h	Pull Down Resistor Control. This bitfield controls whether the pull down resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
15-14	RESERVED	R/W	Xh	
13	WAKESTATE	R	0h	This bitfield has the IOPAD WAKEUP signal as a status bit Reset type: SYSRST
12-8	RESERVED	R/W	Xh	
7	PC	R/W	0h	Peripheral Connect bit. This bit indicates if the designated peripheral function is "Connected" Reset type: SYSRST 0h = 0 1h = 1
6-5	RESERVED	R/W	Xh	
4-0	PF	R/W	0h	Peripheral Function selection bits. This bitfield indicates what peripheral function is selected for a specific pin. Reset type: SYSRST 0h = 0 1h = 1 2h = 2 3h = 3 4h = 4 5h = 5 6h = 6 8h = 8 Ah = 10 Bh = 11 Dh = 13 10h = 16

64 PINCM63 Register (Offset = FCh) [Reset = 0XXXXXX0h]

PINCM63 is shown in Figure 14-65 and described in Table 14-70.

Return to the Summary Table.

Pin Control Management Register

Figure 14-65. PINCM63 Register

31	30	29	28	27	26	25	24
RESERVED			WCOMP	WUEN	INV	Z1	RESERVED
R/W-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-Xh
23	22	21	20	19	18	17	16
RESERVED					INENA	PIPU	PIPD
R/W-Xh					R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED		WAKESTATE	RESERVED				
R/W-Xh		R-0h	R/W-Xh				
7	6	5	4	3	2	1	0
PC	RESERVED		PF				
R/W-0h	R/W-Xh		R/W-0h				

Table 14-134. PINCM63 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R/W	0h	
28	WCOMP	R/W	0h	Wakeup Compare Value bit. This bitfield indicates on what data match this IOCELL with intiate a Wake from SHUTDOWN when WUEN = 1. Reset type: SYSRST 0h = 0 1h = 1
27	WUEN	R/W	0h	Wakeup Enable bit. This bitfield indicates whether or not the Wake from Shutdown is enabled on this IOCELL. Reset type: SYSRST 0h = 0 1h = 1
26	INV	R/W	0h	Data Inversion Control Selection. This bitfield controls whether the input and output logic of this IOCELL is inverted. Reset type: SYSRST 0h = 0 1h = 1
25	Z1	R/W	0h	HI-Z Output State Control Selection. This bitfield controls whether or not a high output value will tri-state the output (open-drain functionality). Reset type: SYSRST 0h = 0 1h = 1
24-19	RESERVED	R/W	Xh	

Table 14-134. PINCM63 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
18	INENA	R/W	0h	Input Enable Control Selection. This bitfield controls whether or not the input to this IOCELL propagates to the connected peripheral function. Reset type: SYSRST 0h = 0 1h = 1
17	PIPU	R/W	0h	Pull Up Resistor Control. This bitfield controls whether the pull up resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
16	PIPD	R/W	0h	Pull Down Resistor Control. This bitfield controls whether the pull down resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
15-14	RESERVED	R/W	Xh	
13	WAKESTATE	R	0h	This bitfield has the IOPAD WAKEUP signal as a status bit Reset type: SYSRST
12-8	RESERVED	R/W	Xh	
7	PC	R/W	0h	Peripheral Connect bit. This bit indicates if the designated peripheral function is "Connected" Reset type: SYSRST 0h = 0 1h = 1
6-5	RESERVED	R/W	Xh	
4-0	PF	R/W	0h	Peripheral Function selection bits. This bitfield indicates what peripheral function is selected for a specific pin. Reset type: SYSRST 0h = 0 1h = 1 2h = 2 3h = 3 4h = 4 5h = 5 7h = 7 Bh = 11 Dh = 13 10h = 16

65 PINCM64 Register (Offset = 100h) [Reset = 0XXXXXX0h]

PINCM64 is shown in Figure 14-66 and described in Table 14-71.

Return to the Summary Table.

Pin Control Management Register

Figure 14-66. PINCM64 Register

31	30	29	28	27	26	25	24
RESERVED			WCOMP	WUEN	INV	Z1	RESERVED
R/W-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-Xh
23	22	21	20	19	18	17	16
RESERVED					INENA	PIPU	PIPD
R/W-Xh					R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED		WAKESTATE	RESERVED				
R/W-Xh		R-0h	R/W-Xh				
7	6	5	4	3	2	1	0
PC	RESERVED		PF				
R/W-0h	R/W-Xh		R/W-0h				

Table 14-136. PINCM64 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R/W	0h	
28	WCOMP	R/W	0h	Wakeup Compare Value bit. This bitfield indicates on what data match this IOCELL with intiate a Wake from SHUTDOWN when WUEN = 1. Reset type: SYSRST 0h = 0 1h = 1
27	WUEN	R/W	0h	Wakeup Enable bit. This bitfield indicates whether or not the Wake from Shutdown is enabled on this IOCELL. Reset type: SYSRST 0h = 0 1h = 1
26	INV	R/W	0h	Data Inversion Control Selection. This bitfield controls whether the input and output logic of this IOCELL is inverted. Reset type: SYSRST 0h = 0 1h = 1
25	Z1	R/W	0h	HI-Z Output State Control Selection. This bitfield controls whether or not a high output value will tri-state the output (open-drain functionality). Reset type: SYSRST 0h = 0 1h = 1
24-19	RESERVED	R/W	Xh	

Table 14-136. PINCM64 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
18	INENA	R/W	0h	Input Enable Control Selection. This bitfield controls whether or not the input to this IOCELL propagates to the connected peripheral function. Reset type: SYSRST 0h = 0 1h = 1
17	PIPU	R/W	0h	Pull Up Resistor Control. This bitfield controls whether the pull up resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
16	PIPD	R/W	0h	Pull Down Resistor Control. This bitfield controls whether the pull down resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
15-14	RESERVED	R/W	Xh	
13	WAKESTATE	R	0h	This bitfield has the IOPAD WAKEUP signal as a status bit Reset type: SYSRST
12-8	RESERVED	R/W	Xh	
7	PC	R/W	0h	Peripheral Connect bit. This bit indicates if the designated peripheral function is "Connected" Reset type: SYSRST 0h = 0 1h = 1
6-5	RESERVED	R/W	Xh	
4-0	PF	R/W	0h	Peripheral Function selection bits. This bitfield indicates what peripheral function is selected for a specific pin. Reset type: SYSRST 0h = 0 1h = 1 3h = 3 4h = 4 5h = 5 6h = 6 8h = 8 9h = 9 Dh = 13 10h = 16

66 PINCM65 Register (Offset = 104h) [Reset = 0XXXXXX0h]

PINCM65 is shown in Figure 14-67 and described in Table 14-72.

Return to the Summary Table.

Pin Control Management Register

Figure 14-67. PINCM65 Register

31	30	29	28	27	26	25	24
RESERVED			WCOMP	WUEN	INV	Z1	RESERVED
R/W-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-Xh
23	22	21	20	19	18	17	16
RESERVED					INENA	PIPU	PIPD
R/W-Xh					R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED		WAKESTATE	RESERVED				
R/W-Xh		R-0h	R/W-Xh				
7	6	5	4	3	2	1	0
PC	RESERVED		PF				
R/W-0h	R/W-Xh		R/W-0h				

Table 14-138. PINCM65 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R/W	0h	
28	WCOMP	R/W	0h	Wakeup Compare Value bit. This bitfield indicates on what data match this IOCELL with intiate a Wake from SHUTDOWN when WUEN = 1. Reset type: SYSRST 0h = 0 1h = 1
27	WUEN	R/W	0h	Wakeup Enable bit. This bitfield indicates whether or not the Wake from Shutdown is enabled on this IOCELL. Reset type: SYSRST 0h = 0 1h = 1
26	INV	R/W	0h	Data Inversion Control Selection. This bitfield controls whether the input and output logic of this IOCELL is inverted. Reset type: SYSRST 0h = 0 1h = 1
25	Z1	R/W	0h	HI-Z Output State Control Selection. This bitfield controls whether or not a high output value will tri-state the output (open-drain functionality). Reset type: SYSRST 0h = 0 1h = 1
24-19	RESERVED	R/W	Xh	

Table 14-138. PINCM65 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
18	INENA	R/W	0h	Input Enable Control Selection. This bitfield controls whether or not the input to this IOCELL propagates to the connected peripheral function. Reset type: SYSRST 0h = 0 1h = 1
17	PIPU	R/W	0h	Pull Up Resistor Control. This bitfield controls whether the pull up resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
16	PIPD	R/W	0h	Pull Down Resistor Control. This bitfield controls whether the pull down resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
15-14	RESERVED	R/W	Xh	
13	WAKESTATE	R	0h	This bitfield has the IOPAD WAKEUP signal as a status bit Reset type: SYSRST
12-8	RESERVED	R/W	Xh	
7	PC	R/W	0h	Peripheral Connect bit. This bit indicates if the designated peripheral function is "Connected" Reset type: SYSRST 0h = 0 1h = 1
6-5	RESERVED	R/W	Xh	
4-0	PF	R/W	0h	Peripheral Function selection bits. This bitfield indicates what peripheral function is selected for a specific pin. Reset type: SYSRST 0h = 0 1h = 1 2h = 2 4h = 4 5h = 5 6h = 6 7h = 7 8h = 8 9h = 9 Dh = 13 10h = 16

67 PINCM66 Register (Offset = 108h) [Reset = 0XXXXXX0h]

PINCM66 is shown in Figure 14-68 and described in Table 14-73.

Return to the Summary Table.

Pin Control Management Register

Figure 14-68. PINCM66 Register

31	30	29	28	27	26	25	24
RESERVED			WCOMP	WUEN	INV	Z1	RESERVED
R/W-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-Xh
23	22	21	20	19	18	17	16
RESERVED					INENA	PIPU	PIPD
R/W-Xh					R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED		WAKESTATE	RESERVED				
R/W-Xh		R-0h	R/W-Xh				
7	6	5	4	3	2	1	0
PC	RESERVED		PF				
R/W-0h	R/W-Xh		R/W-0h				

Table 14-140. PINCM66 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R/W	0h	
28	WCOMP	R/W	0h	Wakeup Compare Value bit. This bitfield indicates on what data match this IOCELL with intiate a Wake from SHUTDOWN when WUEN = 1. Reset type: SYSRST 0h = 0 1h = 1
27	WUEN	R/W	0h	Wakeup Enable bit. This bitfield indicates whether or not the Wake from Shutdown is enabled on this IOCELL. Reset type: SYSRST 0h = 0 1h = 1
26	INV	R/W	0h	Data Inversion Control Selection. This bitfield controls whether the input and output logic of this IOCELL is inverted. Reset type: SYSRST 0h = 0 1h = 1
25	Z1	R/W	0h	HI-Z Output State Control Selection. This bitfield controls whether or not a high output value will tri-state the output (open-drain functionality). Reset type: SYSRST 0h = 0 1h = 1
24-19	RESERVED	R/W	Xh	

Table 14-140. PINCM66 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
18	INENA	R/W	0h	Input Enable Control Selection. This bitfield controls whether or not the input to this IOCELL propagates to the connected peripheral function. Reset type: SYSRST 0h = 0 1h = 1
17	PIPU	R/W	0h	Pull Up Resistor Control. This bitfield controls whether the pull up resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
16	PIPD	R/W	0h	Pull Down Resistor Control. This bitfield controls whether the pull down resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
15-14	RESERVED	R/W	Xh	
13	WAKESTATE	R	0h	This bitfield has the IOPAD WAKEUP signal as a status bit Reset type: SYSRST
12-8	RESERVED	R/W	Xh	
7	PC	R/W	0h	Peripheral Connect bit. This bit indicates if the designated peripheral function is "Connected" Reset type: SYSRST 0h = 0 1h = 1
6-5	RESERVED	R/W	Xh	
4-0	PF	R/W	0h	Peripheral Function selection bits. This bitfield indicates what peripheral function is selected for a specific pin. Reset type: SYSRST 0h = 0 1h = 1 2h = 2 3h = 3 5h = 5 6h = 6 7h = 7 8h = 8 9h = 9 Ah = 10 Bh = 11 Dh = 13 10h = 16

68 PINCM67 Register (Offset = 10Ch) [Reset = 0XXXXXX0h]

PINCM67 is shown in Figure 14-69 and described in Table 14-74.

Return to the Summary Table.

Pin Control Management Register

Figure 14-69. PINCM67 Register

31	30	29	28	27	26	25	24
RESERVED			WCOMP	WUEN	INV	Z1	RESERVED
R/W-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-Xh
23	22	21	20	19	18	17	16
RESERVED					INENA	PIPU	PIPD
R/W-Xh					R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED		WAKESTATE	RESERVED				
R/W-Xh		R-0h	R/W-Xh				
7	6	5	4	3	2	1	0
PC	RESERVED		PF				
R/W-0h	R/W-Xh		R/W-0h				

Table 14-142. PINCM67 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R/W	0h	
28	WCOMP	R/W	0h	Wakeup Compare Value bit. This bitfield indicates on what data match this IOCELL with intiate a Wake from SHUTDOWN when WUEN = 1. Reset type: SYSRST 0h = 0 1h = 1
27	WUEN	R/W	0h	Wakeup Enable bit. This bitfield indicates whether or not the Wake from Shutdown is enabled on this IOCELL. Reset type: SYSRST 0h = 0 1h = 1
26	INV	R/W	0h	Data Inversion Control Selection. This bitfield controls whether the input and output logic of this IOCELL is inverted. Reset type: SYSRST 0h = 0 1h = 1
25	Z1	R/W	0h	HI-Z Output State Control Selection. This bitfield controls whether or not a high output value will tri-state the output (open-drain functionality). Reset type: SYSRST 0h = 0 1h = 1
24-19	RESERVED	R/W	Xh	

Table 14-142. PINCM67 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
18	INENA	R/W	0h	Input Enable Control Selection. This bitfield controls whether or not the input to this IOCELL propagates to the connected peripheral function. Reset type: SYSRST 0h = 0 1h = 1
17	PIPU	R/W	0h	Pull Up Resistor Control. This bitfield controls whether the pull up resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
16	PIPD	R/W	0h	Pull Down Resistor Control. This bitfield controls whether the pull down resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
15-14	RESERVED	R/W	Xh	
13	WAKESTATE	R	0h	This bitfield has the IOPAD WAKEUP signal as a status bit Reset type: SYSRST
12-8	RESERVED	R/W	Xh	
7	PC	R/W	0h	Peripheral Connect bit. This bit indicates if the designated peripheral function is "Connected" Reset type: SYSRST 0h = 0 1h = 1
6-5	RESERVED	R/W	Xh	
4-0	PF	R/W	0h	Peripheral Function selection bits. This bitfield indicates what peripheral function is selected for a specific pin. Reset type: SYSRST 0h = 0 1h = 1 2h = 2 3h = 3 4h = 4 5h = 5 6h = 6 7h = 7 8h = 8 Ah = 10 Bh = 11 Dh = 13 10h = 16

69 PINCM68 Register (Offset = 110h) [Reset = 0XXXXXX0h]

PINCM68 is shown in Figure 14-70 and described in Table 14-75.

Return to the Summary Table.

Pin Control Management Register

Figure 14-70. PINCM68 Register

31	30	29	28	27	26	25	24
RESERVED			WCOMP	WUEN	INV	Z1	RESERVED
R/W-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-Xh
23	22	21	20	19	18	17	16
RESERVED					INENA	PIPU	PIPD
R/W-Xh					R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED		WAKESTATE	RESERVED				
R/W-Xh		R-0h	R/W-Xh				
7	6	5	4	3	2	1	0
PC	RESERVED		PF				
R/W-0h	R/W-Xh		R/W-0h				

Table 14-144. PINCM68 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R/W	0h	
28	WCOMP	R/W	0h	Wakeup Compare Value bit. This bitfield indicates on what data match this IOCELL with intiate a Wake from SHUTDOWN when WUEN = 1. Reset type: SYSRST 0h = 0 1h = 1
27	WUEN	R/W	0h	Wakeup Enable bit. This bitfield indicates whether or not the Wake from Shutdown is enabled on this IOCELL. Reset type: SYSRST 0h = 0 1h = 1
26	INV	R/W	0h	Data Inversion Control Selection. This bitfield controls whether the input and output logic of this IOCELL is inverted. Reset type: SYSRST 0h = 0 1h = 1
25	Z1	R/W	0h	HI-Z Output State Control Selection. This bitfield controls whether or not a high output value will tri-state the output (open-drain functionality). Reset type: SYSRST 0h = 0 1h = 1
24-19	RESERVED	R/W	Xh	

Table 14-144. PINCM68 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
18	INENA	R/W	0h	Input Enable Control Selection. This bitfield controls whether or not the input to this IOCELL propagates to the connected peripheral function. Reset type: SYSRST 0h = 0 1h = 1
17	PIPU	R/W	0h	Pull Up Resistor Control. This bitfield controls whether the pull up resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
16	PIPD	R/W	0h	Pull Down Resistor Control. This bitfield controls whether the pull down resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
15-14	RESERVED	R/W	Xh	
13	WAKESTATE	R	0h	This bitfield has the IOPAD WAKEUP signal as a status bit Reset type: SYSRST
12-8	RESERVED	R/W	Xh	
7	PC	R/W	0h	Peripheral Connect bit. This bit indicates if the designated peripheral function is "Connected" Reset type: SYSRST 0h = 0 1h = 1
6-5	RESERVED	R/W	Xh	
4-0	PF	R/W	0h	Peripheral Function selection bits. This bitfield indicates what peripheral function is selected for a specific pin. Reset type: SYSRST 0h = 0 1h = 1 2h = 2 3h = 3 6h = 6 7h = 7 8h = 8 Bh = 11 Dh = 13 10h = 16

70 PINCM69 Register (Offset = 114h) [Reset = 0XXXXXX0h]

PINCM69 is shown in Figure 14-71 and described in Table 14-76.

Return to the Summary Table.

Pin Control Management Register

Figure 14-71. PINCM69 Register

31	30	29	28	27	26	25	24
RESERVED			WCOMP	WUEN	INV	Z1	RESERVED
R/W-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-Xh
23	22	21	20	19	18	17	16
RESERVED					INENA	PIPU	PIPD
R/W-Xh					R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED		WAKESTATE	RESERVED				
R/W-Xh		R-0h	R/W-Xh				
7	6	5	4	3	2	1	0
PC	RESERVED		PF				
R/W-0h	R/W-Xh		R/W-0h				

Table 14-146. PINCM69 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R/W	0h	
28	WCOMP	R/W	0h	Wakeup Compare Value bit. This bitfield indicates on what data match this IOCELL with intiate a Wake from SHUTDOWN when WUEN = 1. Reset type: SYSRST 0h = 0 1h = 1
27	WUEN	R/W	0h	Wakeup Enable bit. This bitfield indicates whether or not the Wake from Shutdown is enabled on this IOCELL. Reset type: SYSRST 0h = 0 1h = 1
26	INV	R/W	0h	Data Inversion Control Selection. This bitfield controls whether the input and output logic of this IOCELL is inverted. Reset type: SYSRST 0h = 0 1h = 1
25	Z1	R/W	0h	HI-Z Output State Control Selection. This bitfield controls whether or not a high output value will tri-state the output (open-drain functionality). Reset type: SYSRST 0h = 0 1h = 1
24-19	RESERVED	R/W	Xh	

Table 14-146. PINCM69 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
18	INENA	R/W	0h	Input Enable Control Selection. This bitfield controls whether or not the input to this IOCELL propagates to the connected peripheral function. Reset type: SYSRST 0h = 0 1h = 1
17	PIPU	R/W	0h	Pull Up Resistor Control. This bitfield controls whether the pull up resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
16	PIPD	R/W	0h	Pull Down Resistor Control. This bitfield controls whether the pull down resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
15-14	RESERVED	R/W	Xh	
13	WAKESTATE	R	0h	This bitfield has the IOPAD WAKEUP signal as a status bit Reset type: SYSRST
12-8	RESERVED	R/W	Xh	
7	PC	R/W	0h	Peripheral Connect bit. This bit indicates if the designated peripheral function is "Connected" Reset type: SYSRST 0h = 0 1h = 1
6-5	RESERVED	R/W	Xh	
4-0	PF	R/W	0h	Peripheral Function selection bits. This bitfield indicates what peripheral function is selected for a specific pin. Reset type: SYSRST 0h = 0 1h = 1 2h = 2 3h = 3 4h = 4 6h = 6 7h = 7 8h = 8 Bh = 11 Dh = 13 10h = 16

71 PINCM70 Register (Offset = 118h) [Reset = 0XXXXXX0h]

PINCM70 is shown in Figure 14-72 and described in Table 14-77.

Return to the Summary Table.

Pin Control Management Register

Figure 14-72. PINCM70 Register

31	30	29	28	27	26	25	24
RESERVED			WCOMP	WUEN	INV	Z1	RESERVED
R/W-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-Xh
23	22	21	20	19	18	17	16
RESERVED					INENA	PIPU	PIPD
R/W-Xh					R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED		WAKESTATE	RESERVED				
R/W-Xh		R-0h	R/W-Xh				
7	6	5	4	3	2	1	0
PC	RESERVED		PF				
R/W-0h	R/W-Xh		R/W-0h				

Table 14-148. PINCM70 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R/W	0h	
28	WCOMP	R/W	0h	Wakeup Compare Value bit. This bitfield indicates on what data match this IOCELL with intiate a Wake from SHUTDOWN when WUEN = 1. Reset type: SYSRST 0h = 0 1h = 1
27	WUEN	R/W	0h	Wakeup Enable bit. This bitfield indicates whether or not the Wake from Shutdown is enabled on this IOCELL. Reset type: SYSRST 0h = 0 1h = 1
26	INV	R/W	0h	Data Inversion Control Selection. This bitfield controls whether the input and output logic of this IOCELL is inverted. Reset type: SYSRST 0h = 0 1h = 1
25	Z1	R/W	0h	HI-Z Output State Control Selection. This bitfield controls whether or not a high output value will tri-state the output (open-drain functionality). Reset type: SYSRST 0h = 0 1h = 1
24-19	RESERVED	R/W	Xh	

Table 14-148. PINCM70 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
18	INENA	R/W	0h	Input Enable Control Selection. This bitfield controls whether or not the input to this IOCELL propagates to the connected peripheral function. Reset type: SYSRST 0h = 0 1h = 1
17	PIPU	R/W	0h	Pull Up Resistor Control. This bitfield controls whether the pull up resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
16	PIPD	R/W	0h	Pull Down Resistor Control. This bitfield controls whether the pull down resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
15-14	RESERVED	R/W	Xh	
13	WAKESTATE	R	0h	This bitfield has the IOPAD WAKEUP signal as a status bit Reset type: SYSRST
12-8	RESERVED	R/W	Xh	
7	PC	R/W	0h	Peripheral Connect bit. This bit indicates if the designated peripheral function is "Connected" Reset type: SYSRST 0h = 0 1h = 1
6-5	RESERVED	R/W	Xh	
4-0	PF	R/W	0h	Peripheral Function selection bits. This bitfield indicates what peripheral function is selected for a specific pin. Reset type: SYSRST 0h = 0 1h = 1 2h = 2 4h = 4 6h = 6 7h = 7 Bh = 11 Dh = 13 10h = 16

72 PINCM71 Register (Offset = 11Ch) [Reset = 0XXXXXX0h]

PINCM71 is shown in Figure 14-73 and described in Table 14-78.

Return to the Summary Table.

Pin Control Management Register

Figure 14-73. PINCM71 Register

31	30	29	28	27	26	25	24
RESERVED			WCOMP	WUEN	INV	Z1	RESERVED
R/W-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-Xh
23	22	21	20	19	18	17	16
RESERVED					INENA	PIPU	PIPD
R/W-Xh					R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED		WAKESTATE	RESERVED				
R/W-Xh		R-0h	R/W-Xh				
7	6	5	4	3	2	1	0
PC	RESERVED		PF				
R/W-0h	R/W-Xh		R/W-0h				

Table 14-150. PINCM71 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R/W	0h	
28	WCOMP	R/W	0h	Wakeup Compare Value bit. This bitfield indicates on what data match this IOCELL with intiate a Wake from SHUTDOWN when WUEN = 1. Reset type: SYSRST 0h = 0 1h = 1
27	WUEN	R/W	0h	Wakeup Enable bit. This bitfield indicates whether or not the Wake from Shutdown is enabled on this IOCELL. Reset type: SYSRST 0h = 0 1h = 1
26	INV	R/W	0h	Data Inversion Control Selection. This bitfield controls whether the input and output logic of this IOCELL is inverted. Reset type: SYSRST 0h = 0 1h = 1
25	Z1	R/W	0h	HI-Z Output State Control Selection. This bitfield controls whether or not a high output value will tri-state the output (open-drain functionality). Reset type: SYSRST 0h = 0 1h = 1
24-19	RESERVED	R/W	Xh	

Table 14-150. PINCM71 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
18	INENA	R/W	0h	Input Enable Control Selection. This bitfield controls whether or not the input to this IOCELL propagates to the connected peripheral function. Reset type: SYSRST 0h = 0 1h = 1
17	PIPU	R/W	0h	Pull Up Resistor Control. This bitfield controls whether the pull up resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
16	PIPD	R/W	0h	Pull Down Resistor Control. This bitfield controls whether the pull down resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
15-14	RESERVED	R/W	Xh	
13	WAKESTATE	R	0h	This bitfield has the IOPAD WAKEUP signal as a status bit Reset type: SYSRST
12-8	RESERVED	R/W	Xh	
7	PC	R/W	0h	Peripheral Connect bit. This bit indicates if the designated peripheral function is "Connected" Reset type: SYSRST 0h = 0 1h = 1
6-5	RESERVED	R/W	Xh	
4-0	PF	R/W	0h	Peripheral Function selection bits. This bitfield indicates what peripheral function is selected for a specific pin. Reset type: SYSRST 0h = 0 1h = 1 7h = 7 8h = 8 Bh = 11 Dh = 13 10h = 16

73 PINCM72 Register (Offset = 120h) [Reset = 0XXXXXX0h]

PINCM72 is shown in Figure 14-74 and described in Table 14-79.

Return to the Summary Table.

Pin Control Management Register

Figure 14-74. PINCM72 Register

31	30	29	28	27	26	25	24
RESERVED			WCOMP	WUEN	INV	Z1	RESERVED
R/W-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-Xh
23	22	21	20	19	18	17	16
RESERVED					INENA	PIPU	PIPD
R/W-Xh					R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED		WAKESTATE	RESERVED				
R/W-Xh		R-0h	R/W-Xh				
7	6	5	4	3	2	1	0
PC	RESERVED		PF				
R/W-0h	R/W-Xh		R/W-0h				

Table 14-152. PINCM72 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R/W	0h	
28	WCOMP	R/W	0h	Wakeup Compare Value bit. This bitfield indicates on what data match this IOCELL with intiate a Wake from SHUTDOWN when WUEN = 1. Reset type: SYSRST 0h = 0 1h = 1
27	WUEN	R/W	0h	Wakeup Enable bit. This bitfield indicates whether or not the Wake from Shutdown is enabled on this IOCELL. Reset type: SYSRST 0h = 0 1h = 1
26	INV	R/W	0h	Data Inversion Control Selection. This bitfield controls whether the input and output logic of this IOCELL is inverted. Reset type: SYSRST 0h = 0 1h = 1
25	Z1	R/W	0h	HI-Z Output State Control Selection. This bitfield controls whether or not a high output value will tri-state the output (open-drain functionality). Reset type: SYSRST 0h = 0 1h = 1
24-19	RESERVED	R/W	Xh	

Table 14-152. PINCM72 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
18	INENA	R/W	0h	Input Enable Control Selection. This bitfield controls whether or not the input to this IOCELL propagates to the connected peripheral function. Reset type: SYSRST 0h = 0 1h = 1
17	PIPU	R/W	0h	Pull Up Resistor Control. This bitfield controls whether the pull up resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
16	PIPD	R/W	0h	Pull Down Resistor Control. This bitfield controls whether the pull down resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
15-14	RESERVED	R/W	Xh	
13	WAKESTATE	R	0h	This bitfield has the IOPAD WAKEUP signal as a status bit Reset type: SYSRST
12-8	RESERVED	R/W	Xh	
7	PC	R/W	0h	Peripheral Connect bit. This bit indicates if the designated peripheral function is "Connected" Reset type: SYSRST 0h = 0 1h = 1
6-5	RESERVED	R/W	Xh	
4-0	PF	R/W	0h	Peripheral Function selection bits. This bitfield indicates what peripheral function is selected for a specific pin. Reset type: SYSRST 0h = 0 1h = 1 2h = 2 3h = 3 4h = 4 7h = 7 8h = 8 Bh = 11 Dh = 13 10h = 16

74 PINCM73 Register (Offset = 124h) [Reset = 0XXXXXX0h]

PINCM73 is shown in Figure 14-75 and described in Table 14-80.

Return to the Summary Table.

Pin Control Management Register

Figure 14-75. PINCM73 Register

31	30	29	28	27	26	25	24
RESERVED			WCOMP	WUEN	INV	Z1	RESERVED
R/W-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-Xh
23	22	21	20	19	18	17	16
RESERVED					INENA	PIPU	PIPD
R/W-Xh					R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED		WAKESTATE	RESERVED				
R/W-Xh		R-0h	R/W-Xh				
7	6	5	4	3	2	1	0
PC	RESERVED		PF				
R/W-0h	R/W-Xh		R/W-0h				

Table 14-154. PINCM73 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R/W	0h	
28	WCOMP	R/W	0h	Wakeup Compare Value bit. This bitfield indicates on what data match this IOCELL with intiate a Wake from SHUTDOWN when WUEN = 1. Reset type: SYSRST 0h = 0 1h = 1
27	WUEN	R/W	0h	Wakeup Enable bit. This bitfield indicates whether or not the Wake from Shutdown is enabled on this IOCELL. Reset type: SYSRST 0h = 0 1h = 1
26	INV	R/W	0h	Data Inversion Control Selection. This bitfield controls whether the input and output logic of this IOCELL is inverted. Reset type: SYSRST 0h = 0 1h = 1
25	Z1	R/W	0h	HI-Z Output State Control Selection. This bitfield controls whether or not a high output value will tri-state the output (open-drain functionality). Reset type: SYSRST 0h = 0 1h = 1
24-19	RESERVED	R/W	Xh	

Table 14-154. PINCM73 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
18	INENA	R/W	0h	Input Enable Control Selection. This bitfield controls whether or not the input to this IOCELL propagates to the connected peripheral function. Reset type: SYSRST 0h = 0 1h = 1
17	PIPU	R/W	0h	Pull Up Resistor Control. This bitfield controls whether the pull up resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
16	PIPD	R/W	0h	Pull Down Resistor Control. This bitfield controls whether the pull down resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
15-14	RESERVED	R/W	Xh	
13	WAKESTATE	R	0h	This bitfield has the IOPAD WAKEUP signal as a status bit Reset type: SYSRST
12-8	RESERVED	R/W	Xh	
7	PC	R/W	0h	Peripheral Connect bit. This bit indicates if the designated peripheral function is "Connected" Reset type: SYSRST 0h = 0 1h = 1
6-5	RESERVED	R/W	Xh	
4-0	PF	R/W	0h	Peripheral Function selection bits. This bitfield indicates what peripheral function is selected for a specific pin. Reset type: SYSRST 0h = 0 1h = 1 2h = 2 3h = 3 4h = 4 7h = 7 Dh = 13 10h = 16

75 PINCM74 Register (Offset = 128h) [Reset = 0XXXXXX0h]

PINCM74 is shown in Figure 14-76 and described in Table 14-81.

Return to the Summary Table.

Pin Control Management Register

Figure 14-76. PINCM74 Register

31	30	29	28	27	26	25	24
RESERVED			WCOMP	WUEN	INV	Z1	RESERVED
R/W-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-Xh
23	22	21	20	19	18	17	16
RESERVED					INENA	PIPU	PIPD
R/W-Xh					R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED		WAKESTATE	RESERVED				
R/W-Xh		R-0h	R/W-Xh				
7	6	5	4	3	2	1	0
PC	RESERVED		PF				
R/W-0h	R/W-Xh		R/W-0h				

Table 14-156. PINCM74 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R/W	0h	
28	WCOMP	R/W	0h	Wakeup Compare Value bit. This bitfield indicates on what data match this IOCELL with intiate a Wake from SHUTDOWN when WUEN = 1. Reset type: SYSRST 0h = 0 1h = 1
27	WUEN	R/W	0h	Wakeup Enable bit. This bitfield indicates whether or not the Wake from Shutdown is enabled on this IOCELL. Reset type: SYSRST 0h = 0 1h = 1
26	INV	R/W	0h	Data Inversion Control Selection. This bitfield controls whether the input and output logic of this IOCELL is inverted. Reset type: SYSRST 0h = 0 1h = 1
25	Z1	R/W	0h	HI-Z Output State Control Selection. This bitfield controls whether or not a high output value will tri-state the output (open-drain functionality). Reset type: SYSRST 0h = 0 1h = 1
24-19	RESERVED	R/W	Xh	

Table 14-156. PINCM74 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
18	INENA	R/W	0h	Input Enable Control Selection. This bitfield controls whether or not the input to this IOCELL propagates to the connected peripheral function. Reset type: SYSRST 0h = 0 1h = 1
17	PIPU	R/W	0h	Pull Up Resistor Control. This bitfield controls whether the pull up resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
16	PIPD	R/W	0h	Pull Down Resistor Control. This bitfield controls whether the pull down resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
15-14	RESERVED	R/W	Xh	
13	WAKESTATE	R	0h	This bitfield has the IOPAD WAKEUP signal as a status bit Reset type: SYSRST
12-8	RESERVED	R/W	Xh	
7	PC	R/W	0h	Peripheral Connect bit. This bit indicates if the designated peripheral function is "Connected" Reset type: SYSRST 0h = 0 1h = 1
6-5	RESERVED	R/W	Xh	
4-0	PF	R/W	0h	Peripheral Function selection bits. This bitfield indicates what peripheral function is selected for a specific pin. Reset type: SYSRST 0h = 0 1h = 1 3h = 3 4h = 4 5h = 5 7h = 7 Dh = 13 10h = 16

76 PINCM75 Register (Offset = 12Ch) [Reset = 0XXXXXX0h]

PINCM75 is shown in Figure 14-77 and described in Table 14-82.

Return to the Summary Table.

Pin Control Management Register

Figure 14-77. PINCM75 Register

31	30	29	28	27	26	25	24
RESERVED			WCOMP	WUEN	INV	Z1	RESERVED
R/W-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-Xh
23	22	21	20	19	18	17	16
RESERVED					INENA	PIPU	PIPD
R/W-Xh					R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED		WAKESTATE	RESERVED				
R/W-Xh		R-0h	R/W-Xh				
7	6	5	4	3	2	1	0
PC	RESERVED		PF				
R/W-0h	R/W-Xh		R/W-0h				

Table 14-158. PINCM75 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R/W	0h	
28	WCOMP	R/W	0h	Wakeup Compare Value bit. This bitfield indicates on what data match this IOCELL with intiate a Wake from SHUTDOWN when WUEN = 1. Reset type: SYSRST 0h = 0 1h = 1
27	WUEN	R/W	0h	Wakeup Enable bit. This bitfield indicates whether or not the Wake from Shutdown is enabled on this IOCELL. Reset type: SYSRST 0h = 0 1h = 1
26	INV	R/W	0h	Data Inversion Control Selection. This bitfield controls whether the input and output logic of this IOCELL is inverted. Reset type: SYSRST 0h = 0 1h = 1
25	Z1	R/W	0h	HI-Z Output State Control Selection. This bitfield controls whether or not a high output value will tri-state the output (open-drain functionality). Reset type: SYSRST 0h = 0 1h = 1
24-19	RESERVED	R/W	Xh	

Table 14-158. PINCM75 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
18	INENA	R/W	0h	Input Enable Control Selection. This bitfield controls whether or not the input to this IOCELL propagates to the connected peripheral function. Reset type: SYSRST 0h = 0 1h = 1
17	PIPU	R/W	0h	Pull Up Resistor Control. This bitfield controls whether the pull up resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
16	PIPD	R/W	0h	Pull Down Resistor Control. This bitfield controls whether the pull down resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
15-14	RESERVED	R/W	Xh	
13	WAKESTATE	R	0h	This bitfield has the IOPAD WAKEUP signal as a status bit Reset type: SYSRST
12-8	RESERVED	R/W	Xh	
7	PC	R/W	0h	Peripheral Connect bit. This bit indicates if the designated peripheral function is "Connected" Reset type: SYSRST 0h = 0 1h = 1
6-5	RESERVED	R/W	Xh	
4-0	PF	R/W	0h	Peripheral Function selection bits. This bitfield indicates what peripheral function is selected for a specific pin. Reset type: SYSRST 0h = 0 1h = 1 3h = 3 4h = 4 5h = 5 6h = 6 Bh = 11 Dh = 13 10h = 16

77 PINCM76 Register (Offset = 130h) [Reset = 0XXXXXX0h]

PINCM76 is shown in Figure 14-78 and described in Table 14-83.

Return to the Summary Table.

Pin Control Management Register

Figure 14-78. PINCM76 Register

31	30	29	28	27	26	25	24
RESERVED			WCOMP	WUEN	INV	Z1	RESERVED
R/W-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-Xh
23	22	21	20	19	18	17	16
RESERVED					INENA	PIPU	PIPD
R/W-Xh					R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED		WAKESTATE	RESERVED				
R/W-Xh		R-0h	R/W-Xh				
7	6	5	4	3	2	1	0
PC	RESERVED		PF				
R/W-0h	R/W-Xh		R/W-0h				

Table 14-160. PINCM76 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R/W	0h	
28	WCOMP	R/W	0h	Wakeup Compare Value bit. This bitfield indicates on what data match this IOCELL with intiate a Wake from SHUTDOWN when WUEN = 1. Reset type: SYSRST 0h = 0 1h = 1
27	WUEN	R/W	0h	Wakeup Enable bit. This bitfield indicates whether or not the Wake from Shutdown is enabled on this IOCELL. Reset type: SYSRST 0h = 0 1h = 1
26	INV	R/W	0h	Data Inversion Control Selection. This bitfield controls whether the input and output logic of this IOCELL is inverted. Reset type: SYSRST 0h = 0 1h = 1
25	Z1	R/W	0h	HI-Z Output State Control Selection. This bitfield controls whether or not a high output value will tri-state the output (open-drain functionality). Reset type: SYSRST 0h = 0 1h = 1
24-19	RESERVED	R/W	Xh	

Table 14-160. PINCM76 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
18	INENA	R/W	0h	Input Enable Control Selection. This bitfield controls whether or not the input to this IOCELL propagates to the connected peripheral function. Reset type: SYSRST 0h = 0 1h = 1
17	PIPU	R/W	0h	Pull Up Resistor Control. This bitfield controls whether the pull up resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
16	PIPD	R/W	0h	Pull Down Resistor Control. This bitfield controls whether the pull down resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
15-14	RESERVED	R/W	Xh	
13	WAKESTATE	R	0h	This bitfield has the IOPAD WAKEUP signal as a status bit Reset type: SYSRST
12-8	RESERVED	R/W	Xh	
7	PC	R/W	0h	Peripheral Connect bit. This bit indicates if the designated peripheral function is "Connected" Reset type: SYSRST 0h = 0 1h = 1
6-5	RESERVED	R/W	Xh	
4-0	PF	R/W	0h	Peripheral Function selection bits. This bitfield indicates what peripheral function is selected for a specific pin. Reset type: SYSRST 0h = 0 1h = 1 3h = 3 4h = 4 6h = 6 Bh = 11 Dh = 13 10h = 16

78 PINCM77 Register (Offset = 134h) [Reset = 0XXXXXX0h]

PINCM77 is shown in Figure 14-79 and described in Table 14-84.

Return to the Summary Table.

Pin Control Management Register

Figure 14-79. PINCM77 Register

31	30	29	28	27	26	25	24
RESERVED			WCOMP	WUEN	INV	Z1	RESERVED
R/W-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-Xh
23	22	21	20	19	18	17	16
RESERVED					INENA	PIPU	PIPD
R/W-Xh					R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED		WAKESTATE	RESERVED				
R/W-Xh		R-0h	R/W-Xh				
7	6	5	4	3	2	1	0
PC	RESERVED		PF				
R/W-0h	R/W-Xh		R/W-0h				

Table 14-162. PINCM77 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R/W	0h	
28	WCOMP	R/W	0h	Wakeup Compare Value bit. This bitfield indicates on what data match this IOCELL with intiate a Wake from SHUTDOWN when WUEN = 1. Reset type: SYSRST 0h = 0 1h = 1
27	WUEN	R/W	0h	Wakeup Enable bit. This bitfield indicates whether or not the Wake from Shutdown is enabled on this IOCELL. Reset type: SYSRST 0h = 0 1h = 1
26	INV	R/W	0h	Data Inversion Control Selection. This bitfield controls whether the input and output logic of this IOCELL is inverted. Reset type: SYSRST 0h = 0 1h = 1
25	Z1	R/W	0h	HI-Z Output State Control Selection. This bitfield controls whether or not a high output value will tri-state the output (open-drain functionality). Reset type: SYSRST 0h = 0 1h = 1
24-19	RESERVED	R/W	Xh	

Table 14-162. PINCM77 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
18	INENA	R/W	0h	Input Enable Control Selection. This bitfield controls whether or not the input to this IOCELL propagates to the connected peripheral function. Reset type: SYSRST 0h = 0 1h = 1
17	PIPU	R/W	0h	Pull Up Resistor Control. This bitfield controls whether the pull up resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
16	PIPD	R/W	0h	Pull Down Resistor Control. This bitfield controls whether the pull down resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
15-14	RESERVED	R/W	Xh	
13	WAKESTATE	R	0h	This bitfield has the IOPAD WAKEUP signal as a status bit Reset type: SYSRST
12-8	RESERVED	R/W	Xh	
7	PC	R/W	0h	Peripheral Connect bit. This bit indicates if the designated peripheral function is "Connected" Reset type: SYSRST 0h = 0 1h = 1
6-5	RESERVED	R/W	Xh	
4-0	PF	R/W	0h	Peripheral Function selection bits. This bitfield indicates what peripheral function is selected for a specific pin. Reset type: SYSRST 0h = 0 1h = 1 3h = 3 4h = 4 6h = 6 Bh = 11 Ch = 12 Dh = 13 10h = 16

79 PINCM78 Register (Offset = 138h) [Reset = 0XXXXXX0h]

PINCM78 is shown in Figure 14-80 and described in Table 14-85.

Return to the Summary Table.

Pin Control Management Register

Figure 14-80. PINCM78 Register

31	30	29	28	27	26	25	24
RESERVED			WCOMP	WUEN	INV	Z1	RESERVED
R/W-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-Xh
23	22	21	20	19	18	17	16
RESERVED					INENA	PIPU	PIPD
R/W-Xh					R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED		WAKESTATE	RESERVED				
R/W-Xh		R-0h	R/W-Xh				
7	6	5	4	3	2	1	0
PC	RESERVED		PF				
R/W-0h	R/W-Xh		R/W-0h				

Table 14-164. PINCM78 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R/W	0h	
28	WCOMP	R/W	0h	Wakeup Compare Value bit. This bitfield indicates on what data match this IOCELL with intiate a Wake from SHUTDOWN when WUEN = 1. Reset type: SYSRST 0h = 0 1h = 1
27	WUEN	R/W	0h	Wakeup Enable bit. This bitfield indicates whether or not the Wake from Shutdown is enabled on this IOCELL. Reset type: SYSRST 0h = 0 1h = 1
26	INV	R/W	0h	Data Inversion Control Selection. This bitfield controls whether the input and output logic of this IOCELL is inverted. Reset type: SYSRST 0h = 0 1h = 1
25	Z1	R/W	0h	HI-Z Output State Control Selection. This bitfield controls whether or not a high output value will tri-state the output (open-drain functionality). Reset type: SYSRST 0h = 0 1h = 1
24-19	RESERVED	R/W	Xh	

Table 14-164. PINCM78 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
18	INENA	R/W	0h	Input Enable Control Selection. This bitfield controls whether or not the input to this IOCELL propagates to the connected peripheral function. Reset type: SYSRST 0h = 0 1h = 1
17	PIPU	R/W	0h	Pull Up Resistor Control. This bitfield controls whether the pull up resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
16	PIPD	R/W	0h	Pull Down Resistor Control. This bitfield controls whether the pull down resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
15-14	RESERVED	R/W	Xh	
13	WAKESTATE	R	0h	This bitfield has the IOPAD WAKEUP signal as a status bit Reset type: SYSRST
12-8	RESERVED	R/W	Xh	
7	PC	R/W	0h	Peripheral Connect bit. This bit indicates if the designated peripheral function is "Connected" Reset type: SYSRST 0h = 0 1h = 1
6-5	RESERVED	R/W	Xh	
4-0	PF	R/W	0h	Peripheral Function selection bits. This bitfield indicates what peripheral function is selected for a specific pin. Reset type: SYSRST 0h = 0 1h = 1 3h = 3 4h = 4 5h = 5 6h = 6 7h = 7 Bh = 11 Dh = 13 10h = 16

80 PINCM79 Register (Offset = 13Ch) [Reset = 0XXXXXX0h]

PINCM79 is shown in Figure 14-81 and described in Table 14-86.

Return to the Summary Table.

Pin Control Management Register

Figure 14-81. PINCM79 Register

31	30	29	28	27	26	25	24
RESERVED			WCOMP	WUEN	INV	Z1	RESERVED
R/W-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-Xh
23	22	21	20	19	18	17	16
RESERVED					INENA	PIPU	PIPD
R/W-Xh					R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED		WAKESTATE	RESERVED				
R/W-Xh		R-0h	R/W-Xh				
7	6	5	4	3	2	1	0
PC	RESERVED		PF				
R/W-0h	R/W-Xh		R/W-0h				

Table 14-166. PINCM79 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R/W	0h	
28	WCOMP	R/W	0h	Wakeup Compare Value bit. This bitfield indicates on what data match this IOCELL with intiate a Wake from SHUTDOWN when WUEN = 1. Reset type: SYSRST 0h = 0 1h = 1
27	WUEN	R/W	0h	Wakeup Enable bit. This bitfield indicates whether or not the Wake from Shutdown is enabled on this IOCELL. Reset type: SYSRST 0h = 0 1h = 1
26	INV	R/W	0h	Data Inversion Control Selection. This bitfield controls whether the input and output logic of this IOCELL is inverted. Reset type: SYSRST 0h = 0 1h = 1
25	Z1	R/W	0h	HI-Z Output State Control Selection. This bitfield controls whether or not a high output value will tri-state the output (open-drain functionality). Reset type: SYSRST 0h = 0 1h = 1
24-19	RESERVED	R/W	Xh	

Table 14-166. PINCM79 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
18	INENA	R/W	0h	Input Enable Control Selection. This bitfield controls whether or not the input to this IOCELL propagates to the connected peripheral function. Reset type: SYSRST 0h = 0 1h = 1
17	PIPU	R/W	0h	Pull Up Resistor Control. This bitfield controls whether the pull up resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
16	PIPD	R/W	0h	Pull Down Resistor Control. This bitfield controls whether the pull down resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
15-14	RESERVED	R/W	Xh	
13	WAKESTATE	R	0h	This bitfield has the IOPAD WAKEUP signal as a status bit Reset type: SYSRST
12-8	RESERVED	R/W	Xh	
7	PC	R/W	0h	Peripheral Connect bit. This bit indicates if the designated peripheral function is "Connected" Reset type: SYSRST 0h = 0 1h = 1
6-5	RESERVED	R/W	Xh	
4-0	PF	R/W	0h	Peripheral Function selection bits. This bitfield indicates what peripheral function is selected for a specific pin. Reset type: SYSRST 0h = 0 1h = 1 5h = 5 7h = 7 8h = 8 9h = 9 Ah = 10 Bh = 11 Dh = 13 10h = 16

81 PINCM80 Register (Offset = 140h) [Reset = 0XXXXXX0h]

PINCM80 is shown in Figure 14-82 and described in Table 14-87.

Return to the Summary Table.

Pin Control Management Register

Figure 14-82. PINCM80 Register

31	30	29	28	27	26	25	24
RESERVED			WCOMP	WUEN	INV	Z1	RESERVED
R/W-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-Xh
23	22	21	20	19	18	17	16
RESERVED					INENA	PIPU	PIPD
R/W-Xh					R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED		WAKESTATE	RESERVED				
R/W-Xh		R-0h	R/W-Xh				
7	6	5	4	3	2	1	0
PC	RESERVED		PF				
R/W-0h	R/W-Xh		R/W-0h				

Table 14-168. PINCM80 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R/W	0h	
28	WCOMP	R/W	0h	Wakeup Compare Value bit. This bitfield indicates on what data match this IOCELL with intiate a Wake from SHUTDOWN when WUEN = 1. Reset type: SYSRST 0h = 0 1h = 1
27	WUEN	R/W	0h	Wakeup Enable bit. This bitfield indicates whether or not the Wake from Shutdown is enabled on this IOCELL. Reset type: SYSRST 0h = 0 1h = 1
26	INV	R/W	0h	Data Inversion Control Selection. This bitfield controls whether the input and output logic of this IOCELL is inverted. Reset type: SYSRST 0h = 0 1h = 1
25	Z1	R/W	0h	HI-Z Output State Control Selection. This bitfield controls whether or not a high output value will tri-state the output (open-drain functionality). Reset type: SYSRST 0h = 0 1h = 1
24-19	RESERVED	R/W	Xh	

Table 14-168. PINCM80 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
18	INENA	R/W	0h	Input Enable Control Selection. This bitfield controls whether or not the input to this IOCELL propagates to the connected peripheral function. Reset type: SYSRST 0h = 0 1h = 1
17	PIPU	R/W	0h	Pull Up Resistor Control. This bitfield controls whether the pull up resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
16	PIPD	R/W	0h	Pull Down Resistor Control. This bitfield controls whether the pull down resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
15-14	RESERVED	R/W	Xh	
13	WAKESTATE	R	0h	This bitfield has the IOPAD WAKEUP signal as a status bit Reset type: SYSRST
12-8	RESERVED	R/W	Xh	
7	PC	R/W	0h	Peripheral Connect bit. This bit indicates if the designated peripheral function is "Connected" Reset type: SYSRST 0h = 0 1h = 1
6-5	RESERVED	R/W	Xh	
4-0	PF	R/W	0h	Peripheral Function selection bits. This bitfield indicates what peripheral function is selected for a specific pin. Reset type: SYSRST 0h = 0 1h = 1 2h = 2 4h = 4 5h = 5 6h = 6 7h = 7 8h = 8 9h = 9 10h = 16

82 PINCM81 Register (Offset = 144h) [Reset = 0XXXXXX0h]

PINCM81 is shown in Figure 14-83 and described in Table 14-88.

Return to the Summary Table.

Pin Control Management Register

Figure 14-83. PINCM81 Register

31	30	29	28	27	26	25	24
RESERVED			WCOMP	WUEN	INV	Z1	RESERVED
R/W-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-Xh
23	22	21	20	19	18	17	16
RESERVED					INENA	PIPU	PIPD
R/W-Xh					R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED		WAKESTATE	RESERVED				
R/W-Xh		R-0h	R/W-Xh				
7	6	5	4	3	2	1	0
PC	RESERVED		PF				
R/W-0h	R/W-Xh		R/W-0h				

Table 14-170. PINCM81 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R/W	0h	
28	WCOMP	R/W	0h	Wakeup Compare Value bit. This bitfield indicates on what data match this IOCELL with intiate a Wake from SHUTDOWN when WUEN = 1. Reset type: SYSRST 0h = 0 1h = 1
27	WUEN	R/W	0h	Wakeup Enable bit. This bitfield indicates whether or not the Wake from Shutdown is enabled on this IOCELL. Reset type: SYSRST 0h = 0 1h = 1
26	INV	R/W	0h	Data Inversion Control Selection. This bitfield controls whether the input and output logic of this IOCELL is inverted. Reset type: SYSRST 0h = 0 1h = 1
25	Z1	R/W	0h	HI-Z Output State Control Selection. This bitfield controls whether or not a high output value will tri-state the output (open-drain functionality). Reset type: SYSRST 0h = 0 1h = 1
24-19	RESERVED	R/W	Xh	

Table 14-170. PINCM81 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
18	INENA	R/W	0h	Input Enable Control Selection. This bitfield controls whether or not the input to this IOCELL propagates to the connected peripheral function. Reset type: SYSRST 0h = 0 1h = 1
17	PIPU	R/W	0h	Pull Up Resistor Control. This bitfield controls whether the pull up resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
16	PIPD	R/W	0h	Pull Down Resistor Control. This bitfield controls whether the pull down resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
15-14	RESERVED	R/W	Xh	
13	WAKESTATE	R	0h	This bitfield has the IOPAD WAKEUP signal as a status bit Reset type: SYSRST
12-8	RESERVED	R/W	Xh	
7	PC	R/W	0h	Peripheral Connect bit. This bit indicates if the designated peripheral function is "Connected" Reset type: SYSRST 0h = 0 1h = 1
6-5	RESERVED	R/W	Xh	
4-0	PF	R/W	0h	Peripheral Function selection bits. This bitfield indicates what peripheral function is selected for a specific pin. Reset type: SYSRST 0h = 0 1h = 1 2h = 2 6h = 6 10h = 16

83 PINCM82 Register (Offset = 148h) [Reset = 0XXXXXX0h]

PINCM82 is shown in Figure 14-84 and described in Table 14-89.

Return to the Summary Table.

Pin Control Management Register

Figure 14-84. PINCM82 Register

31	30	29	28	27	26	25	24
RESERVED			WCOMP	WUEN	INV	Z1	RESERVED
R/W-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-Xh
23	22	21	20	19	18	17	16
RESERVED					INENA	PIPU	PIPD
R/W-Xh					R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED		WAKESTATE	RESERVED				
R/W-Xh		R-0h	R/W-Xh				
7	6	5	4	3	2	1	0
PC	RESERVED		PF				
R/W-0h	R/W-Xh		R/W-0h				

Table 14-172. PINCM82 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R/W	0h	
28	WCOMP	R/W	0h	Wakeup Compare Value bit. This bitfield indicates on what data match this IOCELL with intiate a Wake from SHUTDOWN when WUEN = 1. Reset type: SYSRST 0h = 0 1h = 1
27	WUEN	R/W	0h	Wakeup Enable bit. This bitfield indicates whether or not the Wake from Shutdown is enabled on this IOCELL. Reset type: SYSRST 0h = 0 1h = 1
26	INV	R/W	0h	Data Inversion Control Selection. This bitfield controls whether the input and output logic of this IOCELL is inverted. Reset type: SYSRST 0h = 0 1h = 1
25	Z1	R/W	0h	HI-Z Output State Control Selection. This bitfield controls whether or not a high output value will tri-state the output (open-drain functionality). Reset type: SYSRST 0h = 0 1h = 1
24-19	RESERVED	R/W	Xh	

Table 14-172. PINCM82 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
18	INENA	R/W	0h	Input Enable Control Selection. This bitfield controls whether or not the input to this IOCELL propagates to the connected peripheral function. Reset type: SYSRST 0h = 0 1h = 1
17	PIPU	R/W	0h	Pull Up Resistor Control. This bitfield controls whether the pull up resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
16	PIPD	R/W	0h	Pull Down Resistor Control. This bitfield controls whether the pull down resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
15-14	RESERVED	R/W	Xh	
13	WAKESTATE	R	0h	This bitfield has the IOPAD WAKEUP signal as a status bit Reset type: SYSRST
12-8	RESERVED	R/W	Xh	
7	PC	R/W	0h	Peripheral Connect bit. This bit indicates if the designated peripheral function is "Connected" Reset type: SYSRST 0h = 0 1h = 1
6-5	RESERVED	R/W	Xh	
4-0	PF	R/W	0h	Peripheral Function selection bits. This bitfield indicates what peripheral function is selected for a specific pin. Reset type: SYSRST 0h = 0 1h = 1 3h = 3 4h = 4 5h = 5 Dh = 13 10h = 16

84 PINCM83 Register (Offset = 14Ch) [Reset = 0XXXXXX0h]

PINCM83 is shown in Figure 14-85 and described in Table 14-90.

Return to the Summary Table.

Pin Control Management Register

Figure 14-85. PINCM83 Register

31	30	29	28	27	26	25	24
RESERVED			WCOMP	WUEN	INV	Z1	RESERVED
R/W-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-Xh
23	22	21	20	19	18	17	16
RESERVED					INENA	PIPU	PIPD
R/W-Xh					R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED		WAKESTATE	RESERVED				
R/W-Xh		R-0h	R/W-Xh				
7	6	5	4	3	2	1	0
PC	RESERVED		PF				
R/W-0h	R/W-Xh		R/W-0h				

Table 14-174. PINCM83 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R/W	0h	
28	WCOMP	R/W	0h	Wakeup Compare Value bit. This bitfield indicates on what data match this IOCELL with intiate a Wake from SHUTDOWN when WUEN = 1. Reset type: SYSRST 0h = 0 1h = 1
27	WUEN	R/W	0h	Wakeup Enable bit. This bitfield indicates whether or not the Wake from Shutdown is enabled on this IOCELL. Reset type: SYSRST 0h = 0 1h = 1
26	INV	R/W	0h	Data Inversion Control Selection. This bitfield controls whether the input and output logic of this IOCELL is inverted. Reset type: SYSRST 0h = 0 1h = 1
25	Z1	R/W	0h	HI-Z Output State Control Selection. This bitfield controls whether or not a high output value will tri-state the output (open-drain functionality). Reset type: SYSRST 0h = 0 1h = 1
24-19	RESERVED	R/W	Xh	

Table 14-174. PINCM83 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
18	INENA	R/W	0h	Input Enable Control Selection. This bitfield controls whether or not the input to this IOCELL propagates to the connected peripheral function. Reset type: SYSRST 0h = 0 1h = 1
17	PIPU	R/W	0h	Pull Up Resistor Control. This bitfield controls whether the pull up resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
16	PIPD	R/W	0h	Pull Down Resistor Control. This bitfield controls whether the pull down resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
15-14	RESERVED	R/W	Xh	
13	WAKESTATE	R	0h	This bitfield has the IOPAD WAKEUP signal as a status bit Reset type: SYSRST
12-8	RESERVED	R/W	Xh	
7	PC	R/W	0h	Peripheral Connect bit. This bit indicates if the designated peripheral function is "Connected" Reset type: SYSRST 0h = 0 1h = 1
6-5	RESERVED	R/W	Xh	
4-0	PF	R/W	0h	Peripheral Function selection bits. This bitfield indicates what peripheral function is selected for a specific pin. Reset type: SYSRST 0h = 0 1h = 1 3h = 3 4h = 4 5h = 5 6h = 6 7h = 7 8h = 8 9h = 9 Bh = 11 Dh = 13 10h = 16

85 PINCM84 Register (Offset = 150h) [Reset = 0XXXXXX0h]

PINCM84 is shown in Figure 14-86 and described in Table 14-91.

Return to the Summary Table.

Pin Control Management Register

Figure 14-86. PINCM84 Register

31	30	29	28	27	26	25	24
RESERVED			WCOMP	WUEN	INV	Z1	RESERVED
R/W-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-Xh
23	22	21	20	19	18	17	16
RESERVED					INENA	PIPU	PIPD
R/W-Xh					R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED		WAKESTATE	RESERVED				
R/W-Xh		R-0h	R/W-Xh				
7	6	5	4	3	2	1	0
PC	RESERVED		PF				
R/W-0h	R/W-Xh		R/W-0h				

Table 14-176. PINCM84 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R/W	0h	
28	WCOMP	R/W	0h	Wakeup Compare Value bit. This bitfield indicates on what data match this IOCELL with intiate a Wake from SHUTDOWN when WUEN = 1. Reset type: SYSRST 0h = 0 1h = 1
27	WUEN	R/W	0h	Wakeup Enable bit. This bitfield indicates whether or not the Wake from Shutdown is enabled on this IOCELL. Reset type: SYSRST 0h = 0 1h = 1
26	INV	R/W	0h	Data Inversion Control Selection. This bitfield controls whether the input and output logic of this IOCELL is inverted. Reset type: SYSRST 0h = 0 1h = 1
25	Z1	R/W	0h	HI-Z Output State Control Selection. This bitfield controls whether or not a high output value will tri-state the output (open-drain functionality). Reset type: SYSRST 0h = 0 1h = 1
24-19	RESERVED	R/W	Xh	

Table 14-176. PINCM84 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
18	INENA	R/W	0h	Input Enable Control Selection. This bitfield controls whether or not the input to this IOCELL propagates to the connected peripheral function. Reset type: SYSRST 0h = 0 1h = 1
17	PIPU	R/W	0h	Pull Up Resistor Control. This bitfield controls whether the pull up resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
16	PIPD	R/W	0h	Pull Down Resistor Control. This bitfield controls whether the pull down resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
15-14	RESERVED	R/W	Xh	
13	WAKESTATE	R	0h	This bitfield has the IOPAD WAKEUP signal as a status bit Reset type: SYSRST
12-8	RESERVED	R/W	Xh	
7	PC	R/W	0h	Peripheral Connect bit. This bit indicates if the designated peripheral function is "Connected" Reset type: SYSRST 0h = 0 1h = 1
6-5	RESERVED	R/W	Xh	
4-0	PF	R/W	0h	Peripheral Function selection bits. This bitfield indicates what peripheral function is selected for a specific pin. Reset type: SYSRST 0h = 0 1h = 1 3h = 3 4h = 4 5h = 5 6h = 6 7h = 7 8h = 8 Bh = 11 Dh = 13 10h = 16

86 PINCM85 Register (Offset = 154h) [Reset = 0XXXXXX0h]

PINCM85 is shown in Figure 14-87 and described in Table 14-92.

Return to the Summary Table.

Pin Control Management Register

Figure 14-87. PINCM85 Register

31	30	29	28	27	26	25	24
RESERVED			WCOMP	WUEN	INV	Z1	RESERVED
R/W-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-Xh
23	22	21	20	19	18	17	16
RESERVED					INENA	PIPU	PIPD
R/W-Xh					R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED		WAKESTATE	RESERVED				
R/W-Xh		R-0h	R/W-Xh				
7	6	5	4	3	2	1	0
PC	RESERVED		PF				
R/W-0h	R/W-Xh		R/W-0h				

Table 14-178. PINCM85 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R/W	0h	
28	WCOMP	R/W	0h	Wakeup Compare Value bit. This bitfield indicates on what data match this IOCELL with intiate a Wake from SHUTDOWN when WUEN = 1. Reset type: SYSRST 0h = 0 1h = 1
27	WUEN	R/W	0h	Wakeup Enable bit. This bitfield indicates whether or not the Wake from Shutdown is enabled on this IOCELL. Reset type: SYSRST 0h = 0 1h = 1
26	INV	R/W	0h	Data Inversion Control Selection. This bitfield controls whether the input and output logic of this IOCELL is inverted. Reset type: SYSRST 0h = 0 1h = 1
25	Z1	R/W	0h	HI-Z Output State Control Selection. This bitfield controls whether or not a high output value will tri-state the output (open-drain functionality). Reset type: SYSRST 0h = 0 1h = 1
24-19	RESERVED	R/W	Xh	

Table 14-178. PINCM85 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
18	INENA	R/W	0h	Input Enable Control Selection. This bitfield controls whether or not the input to this IOCELL propagates to the connected peripheral function. Reset type: SYSRST 0h = 0 1h = 1
17	PIPU	R/W	0h	Pull Up Resistor Control. This bitfield controls whether the pull up resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
16	PIPD	R/W	0h	Pull Down Resistor Control. This bitfield controls whether the pull down resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
15-14	RESERVED	R/W	Xh	
13	WAKESTATE	R	0h	This bitfield has the IOPAD WAKEUP signal as a status bit Reset type: SYSRST
12-8	RESERVED	R/W	Xh	
7	PC	R/W	0h	Peripheral Connect bit. This bit indicates if the designated peripheral function is "Connected" Reset type: SYSRST 0h = 0 1h = 1
6-5	RESERVED	R/W	Xh	
4-0	PF	R/W	0h	Peripheral Function selection bits. This bitfield indicates what peripheral function is selected for a specific pin. Reset type: SYSRST 0h = 0 1h = 1 3h = 3 4h = 4 6h = 6 Dh = 13 10h = 16

87 PINCM86 Register (Offset = 158h) [Reset = 0XXXXXX0h]

PINCM86 is shown in Figure 14-88 and described in Table 14-93.

Return to the Summary Table.

Pin Control Management Register

Figure 14-88. PINCM86 Register

31	30	29	28	27	26	25	24
RESERVED			WCOMP	WUEN	INV	Z1	RESERVED
R/W-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-Xh
23	22	21	20	19	18	17	16
RESERVED					INENA	PIPU	PIPD
R/W-Xh					R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED		WAKESTATE	RESERVED				
R/W-Xh		R-0h	R/W-Xh				
7	6	5	4	3	2	1	0
PC	RESERVED		PF				
R/W-0h	R/W-Xh		R/W-0h				

Table 14-180. PINCM86 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R/W	0h	
28	WCOMP	R/W	0h	Wakeup Compare Value bit. This bitfield indicates on what data match this IOCELL with intiate a Wake from SHUTDOWN when WUEN = 1. Reset type: SYSRST 0h = 0 1h = 1
27	WUEN	R/W	0h	Wakeup Enable bit. This bitfield indicates whether or not the Wake from Shutdown is enabled on this IOCELL. Reset type: SYSRST 0h = 0 1h = 1
26	INV	R/W	0h	Data Inversion Control Selection. This bitfield controls whether the input and output logic of this IOCELL is inverted. Reset type: SYSRST 0h = 0 1h = 1
25	Z1	R/W	0h	HI-Z Output State Control Selection. This bitfield controls whether or not a high output value will tri-state the output (open-drain functionality). Reset type: SYSRST 0h = 0 1h = 1
24-19	RESERVED	R/W	Xh	

Table 14-180. PINCM86 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
18	INENA	R/W	0h	Input Enable Control Selection. This bitfield controls whether or not the input to this IOCELL propagates to the connected peripheral function. Reset type: SYSRST 0h = 0 1h = 1
17	PIPU	R/W	0h	Pull Up Resistor Control. This bitfield controls whether the pull up resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
16	PIPD	R/W	0h	Pull Down Resistor Control. This bitfield controls whether the pull down resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
15-14	RESERVED	R/W	Xh	
13	WAKESTATE	R	0h	This bitfield has the IOPAD WAKEUP signal as a status bit Reset type: SYSRST
12-8	RESERVED	R/W	Xh	
7	PC	R/W	0h	Peripheral Connect bit. This bit indicates if the designated peripheral function is "Connected" Reset type: SYSRST 0h = 0 1h = 1
6-5	RESERVED	R/W	Xh	
4-0	PF	R/W	0h	Peripheral Function selection bits. This bitfield indicates what peripheral function is selected for a specific pin. Reset type: SYSRST 0h = 0 1h = 1 3h = 3 4h = 4 5h = 5 6h = 6 7h = 7 8h = 8 9h = 9 Ah = 10 Bh = 11 10h = 16

88 PINCM87 Register (Offset = 15Ch) [Reset = 0XXXXXX0h]

PINCM87 is shown in Figure 14-89 and described in Table 14-94.

Return to the Summary Table.

Pin Control Management Register

Figure 14-89. PINCM87 Register

31	30	29	28	27	26	25	24
RESERVED			WCOMP	WUEN	INV	Z1	RESERVED
R/W-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-Xh
23	22	21	20	19	18	17	16
RESERVED					INENA	PIPU	PIPD
R/W-Xh					R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED		WAKESTATE	RESERVED				
R/W-Xh		R-0h	R/W-Xh				
7	6	5	4	3	2	1	0
PC	RESERVED		PF				
R/W-0h	R/W-Xh		R/W-0h				

Table 14-182. PINCM87 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R/W	0h	
28	WCOMP	R/W	0h	Wakeup Compare Value bit. This bitfield indicates on what data match this IOCELL with intiate a Wake from SHUTDOWN when WUEN = 1. Reset type: SYSRST 0h = 0 1h = 1
27	WUEN	R/W	0h	Wakeup Enable bit. This bitfield indicates whether or not the Wake from Shutdown is enabled on this IOCELL. Reset type: SYSRST 0h = 0 1h = 1
26	INV	R/W	0h	Data Inversion Control Selection. This bitfield controls whether the input and output logic of this IOCELL is inverted. Reset type: SYSRST 0h = 0 1h = 1
25	Z1	R/W	0h	HI-Z Output State Control Selection. This bitfield controls whether or not a high output value will tri-state the output (open-drain functionality). Reset type: SYSRST 0h = 0 1h = 1
24-19	RESERVED	R/W	Xh	

Table 14-182. PINCM87 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
18	INENA	R/W	0h	Input Enable Control Selection. This bitfield controls whether or not the input to this IOCELL propagates to the connected peripheral function. Reset type: SYSRST 0h = 0 1h = 1
17	PIPU	R/W	0h	Pull Up Resistor Control. This bitfield controls whether the pull up resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
16	PIPD	R/W	0h	Pull Down Resistor Control. This bitfield controls whether the pull down resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
15-14	RESERVED	R/W	Xh	
13	WAKESTATE	R	0h	This bitfield has the IOPAD WAKEUP signal as a status bit Reset type: SYSRST
12-8	RESERVED	R/W	Xh	
7	PC	R/W	0h	Peripheral Connect bit. This bit indicates if the designated peripheral function is "Connected" Reset type: SYSRST 0h = 0 1h = 1
6-5	RESERVED	R/W	Xh	
4-0	PF	R/W	0h	Peripheral Function selection bits. This bitfield indicates what peripheral function is selected for a specific pin. Reset type: SYSRST 0h = 0 1h = 1 2h = 2 3h = 3 4h = 4 5h = 5 6h = 6 7h = 7 Ah = 10 Bh = 11 Dh = 13 10h = 16

89 PINCM88 Register (Offset = 160h) [Reset = 0XXXXXX0h]

PINCM88 is shown in Figure 14-90 and described in Table 14-95.

Return to the Summary Table.

Pin Control Management Register

Figure 14-90. PINCM88 Register

31	30	29	28	27	26	25	24
RESERVED			WCOMP	WUEN	INV	Z1	RESERVED
R/W-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-Xh
23	22	21	20	19	18	17	16
RESERVED					INENA	PIPU	PIPD
R/W-Xh					R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED		WAKESTATE	RESERVED				
R/W-Xh		R-0h	R/W-Xh				
7	6	5	4	3	2	1	0
PC	RESERVED		PF				
R/W-0h	R/W-Xh		R/W-0h				

Table 14-184. PINCM88 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R/W	0h	
28	WCOMP	R/W	0h	Wakeup Compare Value bit. This bitfield indicates on what data match this IOCELL with intiate a Wake from SHUTDOWN when WUEN = 1. Reset type: SYSRST 0h = 0 1h = 1
27	WUEN	R/W	0h	Wakeup Enable bit. This bitfield indicates whether or not the Wake from Shutdown is enabled on this IOCELL. Reset type: SYSRST 0h = 0 1h = 1
26	INV	R/W	0h	Data Inversion Control Selection. This bitfield controls whether the input and output logic of this IOCELL is inverted. Reset type: SYSRST 0h = 0 1h = 1
25	Z1	R/W	0h	HI-Z Output State Control Selection. This bitfield controls whether or not a high output value will tri-state the output (open-drain functionality). Reset type: SYSRST 0h = 0 1h = 1
24-19	RESERVED	R/W	Xh	

Table 14-184. PINCM88 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
18	INENA	R/W	0h	Input Enable Control Selection. This bitfield controls whether or not the input to this IOCELL propagates to the connected peripheral function. Reset type: SYSRST 0h = 0 1h = 1
17	PIPU	R/W	0h	Pull Up Resistor Control. This bitfield controls whether the pull up resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
16	PIPD	R/W	0h	Pull Down Resistor Control. This bitfield controls whether the pull down resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
15-14	RESERVED	R/W	Xh	
13	WAKESTATE	R	0h	This bitfield has the IOPAD WAKEUP signal as a status bit Reset type: SYSRST
12-8	RESERVED	R/W	Xh	
7	PC	R/W	0h	Peripheral Connect bit. This bit indicates if the designated peripheral function is "Connected" Reset type: SYSRST 0h = 0 1h = 1
6-5	RESERVED	R/W	Xh	
4-0	PF	R/W	0h	Peripheral Function selection bits. This bitfield indicates what peripheral function is selected for a specific pin. Reset type: SYSRST 0h = 0 1h = 1 2h = 2 4h = 4 5h = 5 7h = 7 Ah = 10 Bh = 11 Dh = 13 10h = 16

90 PINCM89 Register (Offset = 164h) [Reset = 0XXXXXX0h]

PINCM89 is shown in Figure 14-91 and described in Table 14-96.

Return to the Summary Table.

Pin Control Management Register

Figure 14-91. PINCM89 Register

31	30	29	28	27	26	25	24
RESERVED			WCOMP	WUEN	INV	Z1	RESERVED
R/W-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-Xh
23	22	21	20	19	18	17	16
RESERVED					INENA	PIPU	PIPD
R/W-Xh					R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED		WAKESTATE	RESERVED				
R/W-Xh		R-0h	R/W-Xh				
7	6	5	4	3	2	1	0
PC	RESERVED		PF				
R/W-0h	R/W-Xh		R/W-0h				

Table 14-186. PINCM89 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R/W	0h	
28	WCOMP	R/W	0h	Wakeup Compare Value bit. This bitfield indicates on what data match this IOCELL with intiate a Wake from SHUTDOWN when WUEN = 1. Reset type: SYSRST 0h = 0 1h = 1
27	WUEN	R/W	0h	Wakeup Enable bit. This bitfield indicates whether or not the Wake from Shutdown is enabled on this IOCELL. Reset type: SYSRST 0h = 0 1h = 1
26	INV	R/W	0h	Data Inversion Control Selection. This bitfield controls whether the input and output logic of this IOCELL is inverted. Reset type: SYSRST 0h = 0 1h = 1
25	Z1	R/W	0h	HI-Z Output State Control Selection. This bitfield controls whether or not a high output value will tri-state the output (open-drain functionality). Reset type: SYSRST 0h = 0 1h = 1
24-19	RESERVED	R/W	Xh	

Table 14-186. PINCM89 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
18	INENA	R/W	0h	Input Enable Control Selection. This bitfield controls whether or not the input to this IOCELL propagates to the connected peripheral function. Reset type: SYSRST 0h = 0 1h = 1
17	PIPU	R/W	0h	Pull Up Resistor Control. This bitfield controls whether the pull up resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
16	PIPD	R/W	0h	Pull Down Resistor Control. This bitfield controls whether the pull down resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
15-14	RESERVED	R/W	Xh	
13	WAKESTATE	R	0h	This bitfield has the IOPAD WAKEUP signal as a status bit Reset type: SYSRST
12-8	RESERVED	R/W	Xh	
7	PC	R/W	0h	Peripheral Connect bit. This bit indicates if the designated peripheral function is "Connected" Reset type: SYSRST 0h = 0 1h = 1
6-5	RESERVED	R/W	Xh	
4-0	PF	R/W	0h	Peripheral Function selection bits. This bitfield indicates what peripheral function is selected for a specific pin. Reset type: SYSRST 0h = 0 1h = 1 2h = 2 3h = 3 4h = 4 5h = 5 6h = 6 7h = 7 8h = 8 9h = 9 Bh = 11 Dh = 13 10h = 16

91 PINCM90 Register (Offset = 168h) [Reset = 0XXXXXX0h]

PINCM90 is shown in Figure 14-92 and described in Table 14-97.

Return to the Summary Table.

Pin Control Management Register

Figure 14-92. PINCM90 Register

31	30	29	28	27	26	25	24
RESERVED			WCOMP	WUEN	INV	Z1	RESERVED
R/W-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-Xh
23	22	21	20	19	18	17	16
RESERVED					INENA	PIPU	PIPD
R/W-Xh					R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED		WAKESTATE	RESERVED				
R/W-Xh		R-0h	R/W-Xh				
7	6	5	4	3	2	1	0
PC	RESERVED		PF				
R/W-0h	R/W-Xh		R/W-0h				

Table 14-188. PINCM90 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R/W	0h	
28	WCOMP	R/W	0h	Wakeup Compare Value bit. This bitfield indicates on what data match this IOCELL with intiate a Wake from SHUTDOWN when WUEN = 1. Reset type: SYSRST 0h = 0 1h = 1
27	WUEN	R/W	0h	Wakeup Enable bit. This bitfield indicates whether or not the Wake from Shutdown is enabled on this IOCELL. Reset type: SYSRST 0h = 0 1h = 1
26	INV	R/W	0h	Data Inversion Control Selection. This bitfield controls whether the input and output logic of this IOCELL is inverted. Reset type: SYSRST 0h = 0 1h = 1
25	Z1	R/W	0h	HI-Z Output State Control Selection. This bitfield controls whether or not a high output value will tri-state the output (open-drain functionality). Reset type: SYSRST 0h = 0 1h = 1
24-19	RESERVED	R/W	Xh	

Table 14-188. PINCM90 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
18	INENA	R/W	0h	Input Enable Control Selection. This bitfield controls whether or not the input to this IOCELL propagates to the connected peripheral function. Reset type: SYSRST 0h = 0 1h = 1
17	PIPU	R/W	0h	Pull Up Resistor Control. This bitfield controls whether the pull up resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
16	PIPD	R/W	0h	Pull Down Resistor Control. This bitfield controls whether the pull down resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
15-14	RESERVED	R/W	Xh	
13	WAKESTATE	R	0h	This bitfield has the IOPAD WAKEUP signal as a status bit Reset type: SYSRST
12-8	RESERVED	R/W	Xh	
7	PC	R/W	0h	Peripheral Connect bit. This bit indicates if the designated peripheral function is "Connected" Reset type: SYSRST 0h = 0 1h = 1
6-5	RESERVED	R/W	Xh	
4-0	PF	R/W	0h	Peripheral Function selection bits. This bitfield indicates what peripheral function is selected for a specific pin. Reset type: SYSRST 0h = 0 1h = 1 2h = 2 3h = 3 4h = 4 5h = 5 6h = 6 7h = 7 8h = 8 Bh = 11 Dh = 13 10h = 16

92 PINCM91 Register (Offset = 16Ch) [Reset = 0XXXXXX0h]

PINCM91 is shown in Figure 14-93 and described in Table 14-98.

Return to the Summary Table.

Pin Control Management Register

Figure 14-93. PINCM91 Register

31	30	29	28	27	26	25	24
RESERVED			WCOMP	WUEN	INV	Z1	RESERVED
R/W-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-Xh
23	22	21	20	19	18	17	16
RESERVED					INENA	PIPU	PIPD
R/W-Xh					R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED		WAKESTATE	RESERVED				
R/W-Xh		R-0h	R/W-Xh				
7	6	5	4	3	2	1	0
PC	RESERVED		PF				
R/W-0h	R/W-Xh		R/W-0h				

Table 14-190. PINCM91 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R/W	0h	
28	WCOMP	R/W	0h	Wakeup Compare Value bit. This bitfield indicates on what data match this IOCELL with intiate a Wake from SHUTDOWN when WUEN = 1. Reset type: SYSRST 0h = 0 1h = 1
27	WUEN	R/W	0h	Wakeup Enable bit. This bitfield indicates whether or not the Wake from Shutdown is enabled on this IOCELL. Reset type: SYSRST 0h = 0 1h = 1
26	INV	R/W	0h	Data Inversion Control Selection. This bitfield controls whether the input and output logic of this IOCELL is inverted. Reset type: SYSRST 0h = 0 1h = 1
25	Z1	R/W	0h	HI-Z Output State Control Selection. This bitfield controls whether or not a high output value will tri-state the output (open-drain functionality). Reset type: SYSRST 0h = 0 1h = 1
24-19	RESERVED	R/W	Xh	

Table 14-190. PINCM91 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
18	INENA	R/W	0h	Input Enable Control Selection. This bitfield controls whether or not the input to this IOCELL propagates to the connected peripheral function. Reset type: SYSRST 0h = 0 1h = 1
17	PIPU	R/W	0h	Pull Up Resistor Control. This bitfield controls whether the pull up resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
16	PIPD	R/W	0h	Pull Down Resistor Control. This bitfield controls whether the pull down resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
15-14	RESERVED	R/W	Xh	
13	WAKESTATE	R	0h	This bitfield has the IOPAD WAKEUP signal as a status bit Reset type: SYSRST
12-8	RESERVED	R/W	Xh	
7	PC	R/W	0h	Peripheral Connect bit. This bit indicates if the designated peripheral function is "Connected" Reset type: SYSRST 0h = 0 1h = 1
6-5	RESERVED	R/W	Xh	
4-0	PF	R/W	0h	Peripheral Function selection bits. This bitfield indicates what peripheral function is selected for a specific pin. Reset type: SYSRST 0h = 0 1h = 1 Dh = 13 10h = 16

93 PINCM92 Register (Offset = 170h) [Reset = 0XXXXXX0h]

PINCM92 is shown in Figure 14-94 and described in Table 14-99.

Return to the Summary Table.

Pin Control Management Register

Figure 14-94. PINCM92 Register

31	30	29	28	27	26	25	24
RESERVED			WCOMP	WUEN	INV	Z1	RESERVED
R/W-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-Xh
23	22	21	20	19	18	17	16
RESERVED					INENA	PIPU	PIPD
R/W-Xh					R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED		WAKESTATE	RESERVED				
R/W-Xh		R-0h	R/W-Xh				
7	6	5	4	3	2	1	0
PC	RESERVED		PF				
R/W-0h	R/W-Xh		R/W-0h				

Table 14-192. PINCM92 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R/W	0h	
28	WCOMP	R/W	0h	Wakeup Compare Value bit. This bitfield indicates on what data match this IOCELL with intiate a Wake from SHUTDOWN when WUEN = 1. Reset type: SYSRST 0h = 0 1h = 1
27	WUEN	R/W	0h	Wakeup Enable bit. This bitfield indicates whether or not the Wake from Shutdown is enabled on this IOCELL. Reset type: SYSRST 0h = 0 1h = 1
26	INV	R/W	0h	Data Inversion Control Selection. This bitfield controls whether the input and output logic of this IOCELL is inverted. Reset type: SYSRST 0h = 0 1h = 1
25	Z1	R/W	0h	HI-Z Output State Control Selection. This bitfield controls whether or not a high output value will tri-state the output (open-drain functionality). Reset type: SYSRST 0h = 0 1h = 1
24-19	RESERVED	R/W	Xh	

Table 14-192. PINCM92 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
18	INENA	R/W	0h	Input Enable Control Selection. This bitfield controls whether or not the input to this IOCELL propagates to the connected peripheral function. Reset type: SYSRST 0h = 0 1h = 1
17	PIPU	R/W	0h	Pull Up Resistor Control. This bitfield controls whether the pull up resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
16	PIPD	R/W	0h	Pull Down Resistor Control. This bitfield controls whether the pull down resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
15-14	RESERVED	R/W	Xh	
13	WAKESTATE	R	0h	This bitfield has the IOPAD WAKEUP signal as a status bit Reset type: SYSRST
12-8	RESERVED	R/W	Xh	
7	PC	R/W	0h	Peripheral Connect bit. This bit indicates if the designated peripheral function is "Connected" Reset type: SYSRST 0h = 0 1h = 1
6-5	RESERVED	R/W	Xh	
4-0	PF	R/W	0h	Peripheral Function selection bits. This bitfield indicates what peripheral function is selected for a specific pin. Reset type: SYSRST 0h = 0 1h = 1 3h = 3 4h = 4 Dh = 13 10h = 16

94 PINCM93 Register (Offset = 174h) [Reset = 0XXXXXX0h]

PINCM93 is shown in Figure 14-95 and described in Table 14-100.

Return to the Summary Table.

Pin Control Management Register

Figure 14-95. PINCM93 Register

31	30	29	28	27	26	25	24
RESERVED			WCOMP	WUEN	INV	Z1	RESERVED
R/W-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-Xh
23	22	21	20	19	18	17	16
RESERVED					INENA	PIPU	PIPD
R/W-Xh					R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED		WAKESTATE	RESERVED				
R/W-Xh		R-0h	R/W-Xh				
7	6	5	4	3	2	1	0
PC	RESERVED		PF				
R/W-0h	R/W-Xh		R/W-0h				

Table 14-194. PINCM93 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R/W	0h	
28	WCOMP	R/W	0h	Wakeup Compare Value bit. This bitfield indicates on what data match this IOCELL with intiate a Wake from SHUTDOWN when WUEN = 1. Reset type: SYSRST 0h = 0 1h = 1
27	WUEN	R/W	0h	Wakeup Enable bit. This bitfield indicates whether or not the Wake from Shutdown is enabled on this IOCELL. Reset type: SYSRST 0h = 0 1h = 1
26	INV	R/W	0h	Data Inversion Control Selection. This bitfield controls whether the input and output logic of this IOCELL is inverted. Reset type: SYSRST 0h = 0 1h = 1
25	Z1	R/W	0h	HI-Z Output State Control Selection. This bitfield controls whether or not a high output value will tri-state the output (open-drain functionality). Reset type: SYSRST 0h = 0 1h = 1
24-19	RESERVED	R/W	Xh	

Table 14-194. PINCM93 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
18	INENA	R/W	0h	Input Enable Control Selection. This bitfield controls whether or not the input to this IOCELL propagates to the connected peripheral function. Reset type: SYSRST 0h = 0 1h = 1
17	PIPU	R/W	0h	Pull Up Resistor Control. This bitfield controls whether the pull up resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
16	PIPD	R/W	0h	Pull Down Resistor Control. This bitfield controls whether the pull down resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
15-14	RESERVED	R/W	Xh	
13	WAKESTATE	R	0h	This bitfield has the IOPAD WAKEUP signal as a status bit Reset type: SYSRST
12-8	RESERVED	R/W	Xh	
7	PC	R/W	0h	Peripheral Connect bit. This bit indicates if the designated peripheral function is "Connected" Reset type: SYSRST 0h = 0 1h = 1
6-5	RESERVED	R/W	Xh	
4-0	PF	R/W	0h	Peripheral Function selection bits. This bitfield indicates what peripheral function is selected for a specific pin. Reset type: SYSRST 0h = 0 1h = 1 3h = 3 4h = 4 5h = 5 7h = 7 Dh = 13 10h = 16

95 PINCM94 Register (Offset = 178h) [Reset = 0XXXXXX0h]

PINCM94 is shown in Figure 14-96 and described in Table 14-101.

Return to the Summary Table.

Pin Control Management Register

Figure 14-96. PINCM94 Register

31	30	29	28	27	26	25	24
RESERVED			WCOMP	WUEN	INV	Z1	RESERVED
R/W-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-Xh
23	22	21	20	19	18	17	16
RESERVED					INENA	PIPU	PIPD
R/W-Xh					R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED		WAKESTATE	RESERVED				
R/W-Xh		R-0h	R/W-Xh				
7	6	5	4	3	2	1	0
PC	RESERVED		PF				
R/W-0h	R/W-Xh		R/W-0h				

Table 14-196. PINCM94 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R/W	0h	
28	WCOMP	R/W	0h	Wakeup Compare Value bit. This bitfield indicates on what data match this IOCELL with intiate a Wake from SHUTDOWN when WUEN = 1. Reset type: SYSRST 0h = 0 1h = 1
27	WUEN	R/W	0h	Wakeup Enable bit. This bitfield indicates whether or not the Wake from Shutdown is enabled on this IOCELL. Reset type: SYSRST 0h = 0 1h = 1
26	INV	R/W	0h	Data Inversion Control Selection. This bitfield controls whether the input and output logic of this IOCELL is inverted. Reset type: SYSRST 0h = 0 1h = 1
25	Z1	R/W	0h	HI-Z Output State Control Selection. This bitfield controls whether or not a high output value will tri-state the output (open-drain functionality). Reset type: SYSRST 0h = 0 1h = 1
24-19	RESERVED	R/W	Xh	

Table 14-196. PINCM94 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
18	INENA	R/W	0h	Input Enable Control Selection. This bitfield controls whether or not the input to this IOCELL propagates to the connected peripheral function. Reset type: SYSRST 0h = 0 1h = 1
17	PIPU	R/W	0h	Pull Up Resistor Control. This bitfield controls whether the pull up resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
16	PIPD	R/W	0h	Pull Down Resistor Control. This bitfield controls whether the pull down resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
15-14	RESERVED	R/W	Xh	
13	WAKESTATE	R	0h	This bitfield has the IOPAD WAKEUP signal as a status bit Reset type: SYSRST
12-8	RESERVED	R/W	Xh	
7	PC	R/W	0h	Peripheral Connect bit. This bit indicates if the designated peripheral function is "Connected" Reset type: SYSRST 0h = 0 1h = 1
6-5	RESERVED	R/W	Xh	
4-0	PF	R/W	0h	Peripheral Function selection bits. This bitfield indicates what peripheral function is selected for a specific pin. Reset type: SYSRST 0h = 0 1h = 1 3h = 3 4h = 4 5h = 5 6h = 6 7h = 7 8h = 8 Dh = 13 10h = 16

96 PINCM95 Register (Offset = 17Ch) [Reset = 0XXXXXX0h]

PINCM95 is shown in Figure 14-97 and described in Table 14-102.

Return to the Summary Table.

Pin Control Management Register

Figure 14-97. PINCM95 Register

31	30	29	28	27	26	25	24
RESERVED			WCOMP	WUEN	INV	Z1	RESERVED
R/W-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-Xh
23	22	21	20	19	18	17	16
RESERVED					INENA	PIPU	PIPD
R/W-Xh					R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED		WAKESTATE	RESERVED				
R/W-Xh		R-0h	R/W-Xh				
7	6	5	4	3	2	1	0
PC	RESERVED		PF				
R/W-0h	R/W-Xh		R/W-0h				

Table 14-198. PINCM95 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R/W	0h	
28	WCOMP	R/W	0h	Wakeup Compare Value bit. This bitfield indicates on what data match this IOCELL with intiate a Wake from SHUTDOWN when WUEN = 1. Reset type: SYSRST 0h = 0 1h = 1
27	WUEN	R/W	0h	Wakeup Enable bit. This bitfield indicates whether or not the Wake from Shutdown is enabled on this IOCELL. Reset type: SYSRST 0h = 0 1h = 1
26	INV	R/W	0h	Data Inversion Control Selection. This bitfield controls whether the input and output logic of this IOCELL is inverted. Reset type: SYSRST 0h = 0 1h = 1
25	Z1	R/W	0h	HI-Z Output State Control Selection. This bitfield controls whether or not a high output value will tri-state the output (open-drain functionality). Reset type: SYSRST 0h = 0 1h = 1
24-19	RESERVED	R/W	Xh	

Table 14-198. PINCM95 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
18	INENA	R/W	0h	Input Enable Control Selection. This bitfield controls whether or not the input to this IOCELL propagates to the connected peripheral function. Reset type: SYSRST 0h = 0 1h = 1
17	PIPU	R/W	0h	Pull Up Resistor Control. This bitfield controls whether the pull up resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
16	PIPD	R/W	0h	Pull Down Resistor Control. This bitfield controls whether the pull down resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
15-14	RESERVED	R/W	Xh	
13	WAKESTATE	R	0h	This bitfield has the IOPAD WAKEUP signal as a status bit Reset type: SYSRST
12-8	RESERVED	R/W	Xh	
7	PC	R/W	0h	Peripheral Connect bit. This bit indicates if the designated peripheral function is "Connected" Reset type: SYSRST 0h = 0 1h = 1
6-5	RESERVED	R/W	Xh	
4-0	PF	R/W	0h	Peripheral Function selection bits. This bitfield indicates what peripheral function is selected for a specific pin. Reset type: SYSRST 0h = 0 1h = 1 3h = 3 4h = 4 5h = 5 7h = 7 8h = 8 Dh = 13 10h = 16

97 PINCM96 Register (Offset = 180h) [Reset = 0XXXXXX0h]

PINCM96 is shown in Figure 14-98 and described in Table 14-103.

Return to the Summary Table.

Pin Control Management Register

Figure 14-98. PINCM96 Register

31	30	29	28	27	26	25	24
RESERVED			WCOMP	WUEN	INV	Z1	RESERVED
R/W-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-Xh
23	22	21	20	19	18	17	16
RESERVED					INENA	PIPU	PIPD
R/W-Xh					R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED		WAKESTATE	RESERVED				
R/W-Xh		R-0h	R/W-Xh				
7	6	5	4	3	2	1	0
PC	RESERVED		PF				
R/W-0h	R/W-Xh		R/W-0h				

Table 14-200. PINCM96 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R/W	0h	
28	WCOMP	R/W	0h	Wakeup Compare Value bit. This bitfield indicates on what data match this IOCELL with intiate a Wake from SHUTDOWN when WUEN = 1. Reset type: SYSRST 0h = 0 1h = 1
27	WUEN	R/W	0h	Wakeup Enable bit. This bitfield indicates whether or not the Wake from Shutdown is enabled on this IOCELL. Reset type: SYSRST 0h = 0 1h = 1
26	INV	R/W	0h	Data Inversion Control Selection. This bitfield controls whether the input and output logic of this IOCELL is inverted. Reset type: SYSRST 0h = 0 1h = 1
25	Z1	R/W	0h	HI-Z Output State Control Selection. This bitfield controls whether or not a high output value will tri-state the output (open-drain functionality). Reset type: SYSRST 0h = 0 1h = 1
24-19	RESERVED	R/W	Xh	

Table 14-200. PINCM96 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
18	INENA	R/W	0h	Input Enable Control Selection. This bitfield controls whether or not the input to this IOCELL propagates to the connected peripheral function. Reset type: SYSRST 0h = 0 1h = 1
17	PIPU	R/W	0h	Pull Up Resistor Control. This bitfield controls whether the pull up resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
16	PIPD	R/W	0h	Pull Down Resistor Control. This bitfield controls whether the pull down resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
15-14	RESERVED	R/W	Xh	
13	WAKESTATE	R	0h	This bitfield has the IOPAD WAKEUP signal as a status bit Reset type: SYSRST
12-8	RESERVED	R/W	Xh	
7	PC	R/W	0h	Peripheral Connect bit. This bit indicates if the designated peripheral function is "Connected" Reset type: SYSRST 0h = 0 1h = 1
6-5	RESERVED	R/W	Xh	
4-0	PF	R/W	0h	Peripheral Function selection bits. This bitfield indicates what peripheral function is selected for a specific pin. Reset type: SYSRST 0h = 0 1h = 1 3h = 3 4h = 4 5h = 5 8h = 8 Eh = 14 Fh = 15 10h = 16

98 PINCM97 Register (Offset = 184h) [Reset = 0XXXXXX0h]

PINCM97 is shown in Figure 14-99 and described in Table 14-104.

Return to the Summary Table.

Pin Control Management Register

Figure 14-99. PINCM97 Register

31	30	29	28	27	26	25	24
RESERVED			WCOMP	WUEN	INV	Z1	RESERVED
R/W-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-Xh
23	22	21	20	19	18	17	16
RESERVED					INENA	PIPU	PIPD
R/W-Xh					R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED		WAKESTATE	RESERVED				
R/W-Xh		R-0h	R/W-Xh				
7	6	5	4	3	2	1	0
PC	RESERVED		PF				
R/W-0h	R/W-Xh		R/W-0h				

Table 14-202. PINCM97 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R/W	0h	
28	WCOMP	R/W	0h	Wakeup Compare Value bit. This bitfield indicates on what data match this IOCELL with intiate a Wake from SHUTDOWN when WUEN = 1. Reset type: SYSRST 0h = 0 1h = 1
27	WUEN	R/W	0h	Wakeup Enable bit. This bitfield indicates whether or not the Wake from Shutdown is enabled on this IOCELL. Reset type: SYSRST 0h = 0 1h = 1
26	INV	R/W	0h	Data Inversion Control Selection. This bitfield controls whether the input and output logic of this IOCELL is inverted. Reset type: SYSRST 0h = 0 1h = 1
25	Z1	R/W	0h	HI-Z Output State Control Selection. This bitfield controls whether or not a high output value will tri-state the output (open-drain functionality). Reset type: SYSRST 0h = 0 1h = 1
24-19	RESERVED	R/W	Xh	

Table 14-202. PINCM97 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
18	INENA	R/W	0h	Input Enable Control Selection. This bitfield controls whether or not the input to this IOCELL propagates to the connected peripheral function. Reset type: SYSRST 0h = 0 1h = 1
17	PIPU	R/W	0h	Pull Up Resistor Control. This bitfield controls whether the pull up resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
16	PIPD	R/W	0h	Pull Down Resistor Control. This bitfield controls whether the pull down resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
15-14	RESERVED	R/W	Xh	
13	WAKESTATE	R	0h	This bitfield has the IOPAD WAKEUP signal as a status bit Reset type: SYSRST
12-8	RESERVED	R/W	Xh	
7	PC	R/W	0h	Peripheral Connect bit. This bit indicates if the designated peripheral function is "Connected" Reset type: SYSRST 0h = 0 1h = 1
6-5	RESERVED	R/W	Xh	
4-0	PF	R/W	0h	Peripheral Function selection bits. This bitfield indicates what peripheral function is selected for a specific pin. Reset type: SYSRST 0h = 0 1h = 1 3h = 3 4h = 4 5h = 5 8h = 8 Eh = 14 Fh = 15 10h = 16

99 PINCM98 Register (Offset = 188h) [Reset = 0XXXXXX0h]

PINCM98 is shown in Figure 14-100 and described in Table 14-105.

Return to the Summary Table.

Pin Control Management Register

Figure 14-100. PINCM98 Register

31	30	29	28	27	26	25	24
RESERVED			WCOMP	WUEN	INV	Z1	RESERVED
R/W-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-Xh
23	22	21	20	19	18	17	16
RESERVED					INENA	PIPU	PIPD
R/W-Xh					R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED		WAKESTATE	RESERVED				
R/W-Xh		R-0h	R/W-Xh				
7	6	5	4	3	2	1	0
PC	RESERVED		PF				
R/W-0h	R/W-Xh		R/W-0h				

Table 14-204. PINCM98 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R/W	0h	
28	WCOMP	R/W	0h	Wakeup Compare Value bit. This bitfield indicates on what data match this IOCELL with intiate a Wake from SHUTDOWN when WUEN = 1. Reset type: SYSRST 0h = 0 1h = 1
27	WUEN	R/W	0h	Wakeup Enable bit. This bitfield indicates whether or not the Wake from Shutdown is enabled on this IOCELL. Reset type: SYSRST 0h = 0 1h = 1
26	INV	R/W	0h	Data Inversion Control Selection. This bitfield controls whether the input and output logic of this IOCELL is inverted. Reset type: SYSRST 0h = 0 1h = 1
25	Z1	R/W	0h	HI-Z Output State Control Selection. This bitfield controls whether or not a high output value will tri-state the output (open-drain functionality). Reset type: SYSRST 0h = 0 1h = 1
24-19	RESERVED	R/W	Xh	

Table 14-204. PINCM98 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
18	INENA	R/W	0h	Input Enable Control Selection. This bitfield controls whether or not the input to this IOCELL propagates to the connected peripheral function. Reset type: SYSRST 0h = 0 1h = 1
17	PIPU	R/W	0h	Pull Up Resistor Control. This bitfield controls whether the pull up resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
16	PIPD	R/W	0h	Pull Down Resistor Control. This bitfield controls whether the pull down resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
15-14	RESERVED	R/W	Xh	
13	WAKESTATE	R	0h	This bitfield has the IOPAD WAKEUP signal as a status bit Reset type: SYSRST
12-8	RESERVED	R/W	Xh	
7	PC	R/W	0h	Peripheral Connect bit. This bit indicates if the designated peripheral function is "Connected" Reset type: SYSRST 0h = 0 1h = 1
6-5	RESERVED	R/W	Xh	
4-0	PF	R/W	0h	Peripheral Function selection bits. This bitfield indicates what peripheral function is selected for a specific pin. Reset type: SYSRST 0h = 0 1h = 1 3h = 3 4h = 4 5h = 5 6h = 6 Bh = 11 Fh = 15 10h = 16

100 PINCM99 Register (Offset = 18Ch) [Reset = 0XXXXXX0h]

PINCM99 is shown in Figure 14-101 and described in Table 14-106.

Return to the Summary Table.

Pin Control Management Register

Figure 14-101. PINCM99 Register

31	30	29	28	27	26	25	24
RESERVED			WCOMP	WUEN	INV	Z1	RESERVED
R/W-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-Xh
23	22	21	20	19	18	17	16
RESERVED					INENA	PIPU	PIPD
R/W-Xh					R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED		WAKESTATE	RESERVED				
R/W-Xh		R-0h	R/W-Xh				
7	6	5	4	3	2	1	0
PC	RESERVED		PF				
R/W-0h	R/W-Xh		R/W-0h				

Table 14-206. PINCM99 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R/W	0h	
28	WCOMP	R/W	0h	Wakeup Compare Value bit. This bitfield indicates on what data match this IOCELL with intiate a Wake from SHUTDOWN when WUEN = 1. Reset type: SYSRST 0h = 0 1h = 1
27	WUEN	R/W	0h	Wakeup Enable bit. This bitfield indicates whether or not the Wake from Shutdown is enabled on this IOCELL. Reset type: SYSRST 0h = 0 1h = 1
26	INV	R/W	0h	Data Inversion Control Selection. This bitfield controls whether the input and output logic of this IOCELL is inverted. Reset type: SYSRST 0h = 0 1h = 1
25	Z1	R/W	0h	HI-Z Output State Control Selection. This bitfield controls whether or not a high output value will tri-state the output (open-drain functionality). Reset type: SYSRST 0h = 0 1h = 1
24-19	RESERVED	R/W	Xh	

Table 14-206. PINCM99 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
18	INENA	R/W	0h	Input Enable Control Selection. This bitfield controls whether or not the input to this IOCELL propagates to the connected peripheral function. Reset type: SYSRST 0h = 0 1h = 1
17	PIPU	R/W	0h	Pull Up Resistor Control. This bitfield controls whether the pull up resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
16	PIPD	R/W	0h	Pull Down Resistor Control. This bitfield controls whether the pull down resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
15-14	RESERVED	R/W	Xh	
13	WAKESTATE	R	0h	This bitfield has the IOPAD WAKEUP signal as a status bit Reset type: SYSRST
12-8	RESERVED	R/W	Xh	
7	PC	R/W	0h	Peripheral Connect bit. This bit indicates if the designated peripheral function is "Connected" Reset type: SYSRST 0h = 0 1h = 1
6-5	RESERVED	R/W	Xh	
4-0	PF	R/W	0h	Peripheral Function selection bits. This bitfield indicates what peripheral function is selected for a specific pin. Reset type: SYSRST 0h = 0 1h = 1 5h = 5 6h = 6 7h = 7 8h = 8 Ah = 10 Bh = 11 Fh = 15 10h = 16

101 PINCM100 Register (Offset = 190h) [Reset = 0XXXXXX0h]

PINCM100 is shown in Figure 14-102 and described in Table 14-107.

Return to the Summary Table.

Pin Control Management Register

Figure 14-102. PINCM100 Register

31	30	29	28	27	26	25	24
RESERVED			WCOMP	WUEN	INV	Z1	RESERVED
R/W-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-Xh
23	22	21	20	19	18	17	16
RESERVED					INENA	PIPU	PIPD
R/W-Xh					R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED		WAKESTATE	RESERVED				
R/W-Xh		R-0h	R/W-Xh				
7	6	5	4	3	2	1	0
PC	RESERVED		PF				
R/W-0h	R/W-Xh		R/W-0h				

Table 14-208. PINCM100 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R/W	0h	
28	WCOMP	R/W	0h	Wakeup Compare Value bit. This bitfield indicates on what data match this IOCELL with intiate a Wake from SHUTDOWN when WUEN = 1. Reset type: SYSRST 0h = 0 1h = 1
27	WUEN	R/W	0h	Wakeup Enable bit. This bitfield indicates whether or not the Wake from Shutdown is enabled on this IOCELL. Reset type: SYSRST 0h = 0 1h = 1
26	INV	R/W	0h	Data Inversion Control Selection. This bitfield controls whether the input and output logic of this IOCELL is inverted. Reset type: SYSRST 0h = 0 1h = 1
25	Z1	R/W	0h	HI-Z Output State Control Selection. This bitfield controls whether or not a high output value will tri-state the output (open-drain functionality). Reset type: SYSRST 0h = 0 1h = 1
24-19	RESERVED	R/W	Xh	

Table 14-208. PINCM100 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
18	INENA	R/W	0h	Input Enable Control Selection. This bitfield controls whether or not the input to this IOCELL propagates to the connected peripheral function. Reset type: SYSRST 0h = 0 1h = 1
17	PIPU	R/W	0h	Pull Up Resistor Control. This bitfield controls whether the pull up resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
16	PIPD	R/W	0h	Pull Down Resistor Control. This bitfield controls whether the pull down resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
15-14	RESERVED	R/W	Xh	
13	WAKESTATE	R	0h	This bitfield has the IOPAD WAKEUP signal as a status bit Reset type: SYSRST
12-8	RESERVED	R/W	Xh	
7	PC	R/W	0h	Peripheral Connect bit. This bit indicates if the designated peripheral function is "Connected" Reset type: SYSRST 0h = 0 1h = 1
6-5	RESERVED	R/W	Xh	
4-0	PF	R/W	0h	Peripheral Function selection bits. This bitfield indicates what peripheral function is selected for a specific pin. Reset type: SYSRST 0h = 0 1h = 1 5h = 5 6h = 6 Ah = 10 Bh = 11 Dh = 13 Eh = 14 Fh = 15 10h = 16

102 PINCM101 Register (Offset = 194h) [Reset = 0XXXXXX0h]

PINCM101 is shown in Figure 14-103 and described in Table 14-108.

Return to the Summary Table.

Pin Control Management Register

Figure 14-103. PINCM101 Register

31	30	29	28	27	26	25	24
RESERVED			WCOMP	WUEN	INV	Z1	RESERVED
R/W-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-Xh
23	22	21	20	19	18	17	16
RESERVED					INENA	PIPU	PIPD
R/W-Xh					R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED		WAKESTATE	RESERVED				
R/W-Xh		R-0h	R/W-Xh				
7	6	5	4	3	2	1	0
PC	RESERVED		PF				
R/W-0h	R/W-Xh		R/W-0h				

Table 14-210. PINCM101 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R/W	0h	
28	WCOMP	R/W	0h	Wakeup Compare Value bit. This bitfield indicates on what data match this IOCELL with intiate a Wake from SHUTDOWN when WUEN = 1. Reset type: SYSRST 0h = 0 1h = 1
27	WUEN	R/W	0h	Wakeup Enable bit. This bitfield indicates whether or not the Wake from Shutdown is enabled on this IOCELL. Reset type: SYSRST 0h = 0 1h = 1
26	INV	R/W	0h	Data Inversion Control Selection. This bitfield controls whether the input and output logic of this IOCELL is inverted. Reset type: SYSRST 0h = 0 1h = 1
25	Z1	R/W	0h	HI-Z Output State Control Selection. This bitfield controls whether or not a high output value will tri-state the output (open-drain functionality). Reset type: SYSRST 0h = 0 1h = 1
24-19	RESERVED	R/W	Xh	

Table 14-210. PINCM101 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
18	INENA	R/W	0h	Input Enable Control Selection. This bitfield controls whether or not the input to this IOCELL propagates to the connected peripheral function. Reset type: SYSRST 0h = 0 1h = 1
17	PIPU	R/W	0h	Pull Up Resistor Control. This bitfield controls whether the pull up resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
16	PIPD	R/W	0h	Pull Down Resistor Control. This bitfield controls whether the pull down resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
15-14	RESERVED	R/W	Xh	
13	WAKESTATE	R	0h	This bitfield has the IOPAD WAKEUP signal as a status bit Reset type: SYSRST
12-8	RESERVED	R/W	Xh	
7	PC	R/W	0h	Peripheral Connect bit. This bit indicates if the designated peripheral function is "Connected" Reset type: SYSRST 0h = 0 1h = 1
6-5	RESERVED	R/W	Xh	
4-0	PF	R/W	0h	Peripheral Function selection bits. This bitfield indicates what peripheral function is selected for a specific pin. Reset type: SYSRST 0h = 0 1h = 1 6h = 6 8h = 8 9h = 9 10h = 16

103 PINCM102 Register (Offset = 198h) [Reset = 0XXXXXX0h]

PINCM102 is shown in Figure 14-104 and described in Table 14-109.

Return to the Summary Table.

Pin Control Management Register

Figure 14-104. PINCM102 Register

31	30	29	28	27	26	25	24
RESERVED			WCOMP	WUEN	INV	Z1	RESERVED
R/W-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-Xh
23	22	21	20	19	18	17	16
RESERVED					INENA	PIPU	PIPD
R/W-Xh					R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED		WAKESTATE	RESERVED				
R/W-Xh		R-0h	R/W-Xh				
7	6	5	4	3	2	1	0
PC	RESERVED		PF				
R/W-0h	R/W-Xh		R/W-0h				

Table 14-212. PINCM102 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R/W	0h	
28	WCOMP	R/W	0h	Wakeup Compare Value bit. This bitfield indicates on what data match this IOCELL with intiate a Wake from SHUTDOWN when WUEN = 1. Reset type: SYSRST 0h = 0 1h = 1
27	WUEN	R/W	0h	Wakeup Enable bit. This bitfield indicates whether or not the Wake from Shutdown is enabled on this IOCELL. Reset type: SYSRST 0h = 0 1h = 1
26	INV	R/W	0h	Data Inversion Control Selection. This bitfield controls whether the input and output logic of this IOCELL is inverted. Reset type: SYSRST 0h = 0 1h = 1
25	Z1	R/W	0h	HI-Z Output State Control Selection. This bitfield controls whether or not a high output value will tri-state the output (open-drain functionality). Reset type: SYSRST 0h = 0 1h = 1
24-19	RESERVED	R/W	Xh	

Table 14-212. PINCM102 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
18	INENA	R/W	0h	Input Enable Control Selection. This bitfield controls whether or not the input to this IOCELL propagates to the connected peripheral function. Reset type: SYSRST 0h = 0 1h = 1
17	PIPU	R/W	0h	Pull Up Resistor Control. This bitfield controls whether the pull up resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
16	PIPD	R/W	0h	Pull Down Resistor Control. This bitfield controls whether the pull down resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
15-14	RESERVED	R/W	Xh	
13	WAKESTATE	R	0h	This bitfield has the IOPAD WAKEUP signal as a status bit Reset type: SYSRST
12-8	RESERVED	R/W	Xh	
7	PC	R/W	0h	Peripheral Connect bit. This bit indicates if the designated peripheral function is "Connected" Reset type: SYSRST 0h = 0 1h = 1
6-5	RESERVED	R/W	Xh	
4-0	PF	R/W	0h	Peripheral Function selection bits. This bitfield indicates what peripheral function is selected for a specific pin. Reset type: SYSRST 0h = 0 1h = 1 8h = 8 9h = 9 10h = 16

104 PINCM103 Register (Offset = 19Ch) [Reset = 0XXXXXX0h]

PINCM103 is shown in Figure 14-105 and described in Table 14-110.

Return to the Summary Table.

Pin Control Management Register

Figure 14-105. PINCM103 Register

31	30	29	28	27	26	25	24
RESERVED			WCOMP	WUEN	INV	Z1	RESERVED
R/W-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-Xh
23	22	21	20	19	18	17	16
RESERVED					INENA	PIPU	PIPD
R/W-Xh					R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED		WAKESTATE	RESERVED				
R/W-Xh		R-0h	R/W-Xh				
7	6	5	4	3	2	1	0
PC	RESERVED		PF				
R/W-0h	R/W-Xh		R/W-0h				

Table 14-214. PINCM103 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R/W	0h	
28	WCOMP	R/W	0h	Wakeup Compare Value bit. This bitfield indicates on what data match this IOCELL with intiate a Wake from SHUTDOWN when WUEN = 1. Reset type: SYSRST 0h = 0 1h = 1
27	WUEN	R/W	0h	Wakeup Enable bit. This bitfield indicates whether or not the Wake from Shutdown is enabled on this IOCELL. Reset type: SYSRST 0h = 0 1h = 1
26	INV	R/W	0h	Data Inversion Control Selection. This bitfield controls whether the input and output logic of this IOCELL is inverted. Reset type: SYSRST 0h = 0 1h = 1
25	Z1	R/W	0h	HI-Z Output State Control Selection. This bitfield controls whether or not a high output value will tri-state the output (open-drain functionality). Reset type: SYSRST 0h = 0 1h = 1
24-19	RESERVED	R/W	Xh	

Table 14-214. PINCM103 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
18	INENA	R/W	0h	Input Enable Control Selection. This bitfield controls whether or not the input to this IOCELL propagates to the connected peripheral function. Reset type: SYSRST 0h = 0 1h = 1
17	PIPU	R/W	0h	Pull Up Resistor Control. This bitfield controls whether the pull up resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
16	PIPD	R/W	0h	Pull Down Resistor Control. This bitfield controls whether the pull down resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
15-14	RESERVED	R/W	Xh	
13	WAKESTATE	R	0h	This bitfield has the IOPAD WAKEUP signal as a status bit Reset type: SYSRST
12-8	RESERVED	R/W	Xh	
7	PC	R/W	0h	Peripheral Connect bit. This bit indicates if the designated peripheral function is "Connected" Reset type: SYSRST 0h = 0 1h = 1
6-5	RESERVED	R/W	Xh	
4-0	PF	R/W	0h	Peripheral Function selection bits. This bitfield indicates what peripheral function is selected for a specific pin. Reset type: SYSRST 0h = 0 1h = 1 5h = 5 8h = 8 9h = 9 Bh = 11 Dh = 13 10h = 16

105 PINCM104 Register (Offset = 1A0h) [Reset = 0XXXXXX0h]

PINCM104 is shown in Figure 14-106 and described in Table 14-111.

Return to the Summary Table.

Pin Control Management Register

Figure 14-106. PINCM104 Register

31	30	29	28	27	26	25	24
RESERVED			WCOMP	WUEN	INV	Z1	RESERVED
R/W-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-Xh
23	22	21	20	19	18	17	16
RESERVED					INENA	PIPU	PIPD
R/W-Xh					R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED		WAKESTATE	RESERVED				
R/W-Xh		R-0h	R/W-Xh				
7	6	5	4	3	2	1	0
PC	RESERVED		PF				
R/W-0h	R/W-Xh		R/W-0h				

Table 14-216. PINCM104 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R/W	0h	
28	WCOMP	R/W	0h	Wakeup Compare Value bit. This bitfield indicates on what data match this IOCELL with intiate a Wake from SHUTDOWN when WUEN = 1. Reset type: SYSRST 0h = 0 1h = 1
27	WUEN	R/W	0h	Wakeup Enable bit. This bitfield indicates whether or not the Wake from Shutdown is enabled on this IOCELL. Reset type: SYSRST 0h = 0 1h = 1
26	INV	R/W	0h	Data Inversion Control Selection. This bitfield controls whether the input and output logic of this IOCELL is inverted. Reset type: SYSRST 0h = 0 1h = 1
25	Z1	R/W	0h	HI-Z Output State Control Selection. This bitfield controls whether or not a high output value will tri-state the output (open-drain functionality). Reset type: SYSRST 0h = 0 1h = 1
24-19	RESERVED	R/W	Xh	

Table 14-216. PINCM104 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
18	INENA	R/W	0h	Input Enable Control Selection. This bitfield controls whether or not the input to this IOCELL propagates to the connected peripheral function. Reset type: SYSRST 0h = 0 1h = 1
17	PIPU	R/W	0h	Pull Up Resistor Control. This bitfield controls whether the pull up resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
16	PIPD	R/W	0h	Pull Down Resistor Control. This bitfield controls whether the pull down resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
15-14	RESERVED	R/W	Xh	
13	WAKESTATE	R	0h	This bitfield has the IOPAD WAKEUP signal as a status bit Reset type: SYSRST
12-8	RESERVED	R/W	Xh	
7	PC	R/W	0h	Peripheral Connect bit. This bit indicates if the designated peripheral function is "Connected" Reset type: SYSRST 0h = 0 1h = 1
6-5	RESERVED	R/W	Xh	
4-0	PF	R/W	0h	Peripheral Function selection bits. This bitfield indicates what peripheral function is selected for a specific pin. Reset type: SYSRST 0h = 0 1h = 1 5h = 5 8h = 8 9h = 9 Bh = 11 Dh = 13 Eh = 14 Fh = 15 10h = 16

106 PINCM105 Register (Offset = 1A4h) [Reset = 0XXXXXX0h]

PINCM105 is shown in Figure 14-107 and described in Table 14-112.

Return to the Summary Table.

Pin Control Management Register

Figure 14-107. PINCM105 Register

31	30	29	28	27	26	25	24
RESERVED			WCOMP	WUEN	INV	Z1	RESERVED
R/W-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-Xh
23	22	21	20	19	18	17	16
RESERVED					INENA	PIPU	PIPD
R/W-Xh					R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED		WAKESTATE	RESERVED				
R/W-Xh		R-0h	R/W-Xh				
7	6	5	4	3	2	1	0
PC	RESERVED		PF				
R/W-0h	R/W-Xh		R/W-0h				

Table 14-218. PINCM105 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R/W	0h	
28	WCOMP	R/W	0h	Wakeup Compare Value bit. This bitfield indicates on what data match this IOCELL with intiate a Wake from SHUTDOWN when WUEN = 1. Reset type: SYSRST 0h = 0 1h = 1
27	WUEN	R/W	0h	Wakeup Enable bit. This bitfield indicates whether or not the Wake from Shutdown is enabled on this IOCELL. Reset type: SYSRST 0h = 0 1h = 1
26	INV	R/W	0h	Data Inversion Control Selection. This bitfield controls whether the input and output logic of this IOCELL is inverted. Reset type: SYSRST 0h = 0 1h = 1
25	Z1	R/W	0h	HI-Z Output State Control Selection. This bitfield controls whether or not a high output value will tri-state the output (open-drain functionality). Reset type: SYSRST 0h = 0 1h = 1
24-19	RESERVED	R/W	Xh	

Table 14-218. PINCM105 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
18	INENA	R/W	0h	Input Enable Control Selection. This bitfield controls whether or not the input to this IOCELL propagates to the connected peripheral function. Reset type: SYSRST 0h = 0 1h = 1
17	PIPU	R/W	0h	Pull Up Resistor Control. This bitfield controls whether the pull up resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
16	PIPD	R/W	0h	Pull Down Resistor Control. This bitfield controls whether the pull down resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
15-14	RESERVED	R/W	Xh	
13	WAKESTATE	R	0h	This bitfield has the IOPAD WAKEUP signal as a status bit Reset type: SYSRST
12-8	RESERVED	R/W	Xh	
7	PC	R/W	0h	Peripheral Connect bit. This bit indicates if the designated peripheral function is "Connected" Reset type: SYSRST 0h = 0 1h = 1
6-5	RESERVED	R/W	Xh	
4-0	PF	R/W	0h	Peripheral Function selection bits. This bitfield indicates what peripheral function is selected for a specific pin. Reset type: SYSRST 0h = 0 1h = 1 7h = 7 8h = 8 9h = 9 Ah = 10 Dh = 13 Eh = 14 Fh = 15 10h = 16

107 PINCM107 Register (Offset = 1ACh) [Reset = 0XXXXXX0h]

PINCM107 is shown in Figure 14-108 and described in Table 14-113.

Return to the Summary Table.

Pin Control Management Register

Figure 14-108. PINCM107 Register

31	30	29	28	27	26	25	24
RESERVED			WCOMP	WUEN	INV	Z1	RESERVED
R/W-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-Xh
23	22	21	20	19	18	17	16
RESERVED					INENA	PIPU	PIPD
R/W-Xh					R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED		WAKESTATE	RESERVED				
R/W-Xh		R-0h	R/W-Xh				
7	6	5	4	3	2	1	0
PC	RESERVED		PF				
R/W-0h	R/W-Xh		R/W-0h				

Table 14-220. PINCM107 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R/W	0h	
28	WCOMP	R/W	0h	Wakeup Compare Value bit. This bitfield indicates on what data match this IOCELL with intiate a Wake from SHUTDOWN when WUEN = 1. Reset type: SYSRST 0h = 0 1h = 1
27	WUEN	R/W	0h	Wakeup Enable bit. This bitfield indicates whether or not the Wake from Shutdown is enabled on this IOCELL. Reset type: SYSRST 0h = 0 1h = 1
26	INV	R/W	0h	Data Inversion Control Selection. This bitfield controls whether the input and output logic of this IOCELL is inverted. Reset type: SYSRST 0h = 0 1h = 1
25	Z1	R/W	0h	HI-Z Output State Control Selection. This bitfield controls whether or not a high output value will tri-state the output (open-drain functionality). Reset type: SYSRST 0h = 0 1h = 1
24-19	RESERVED	R/W	Xh	

Table 14-220. PINCM107 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
18	INENA	R/W	0h	Input Enable Control Selection. This bitfield controls whether or not the input to this IOCELL propagates to the connected peripheral function. Reset type: SYSRST 0h = 0 1h = 1
17	PIPU	R/W	0h	Pull Up Resistor Control. This bitfield controls whether the pull up resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
16	PIPD	R/W	0h	Pull Down Resistor Control. This bitfield controls whether the pull down resistor on this IOCELL is enabled or disabled. Reset type: SYSRST 0h = 0 1h = 1
15-14	RESERVED	R/W	Xh	
13	WAKESTATE	R	0h	This bitfield has the IOPAD WAKEUP signal as a status bit Reset type: SYSRST
12-8	RESERVED	R/W	Xh	
7	PC	R/W	0h	Peripheral Connect bit. This bit indicates if the designated peripheral function is "Connected" Reset type: SYSRST 0h = 0 1h = 1
6-5	RESERVED	R/W	Xh	
4-0	PF	R/W	0h	Peripheral Function selection bits. This bitfield indicates what peripheral function is selected for a specific pin. Reset type: SYSRST 0h = 0 1h = 1 10h = 16

15.1 General-Purpose Input/Output (GPIO)

The GPIO peripheral provides the user with a means to write data out and read data in to and from the device pins. The GPIO also provides a way to detect wakeup events while the device is in a low power state. This chapter describes the operation of the GPIO peripheral.

15.1.1 GPIO Overview

The GPIO is used to read in digital data from the device pins and to send digital data out to the device pins.

The GPIO module features include:

- Single CLK cycle access to registers (zero wait state MMR access from CPU)
- Set/clear/toggle multiple bits without the need of a read-modify-write construct in software
- Direct writes to individual GPIO output bits (DOUT) without the need of a read-modify-write construct in software
- Direct read comparisons of individual GPIO input bits (DIN) without the need to use masking in software
- DOUT serviceable by DMA to generate a predefined output sequence on specified pins using DMAMASK register
- "FastWake" feature enables low-power wakeup from STOP and STANDBY modes based on input edge detection
- User controlled input filtering (configurable per IO)

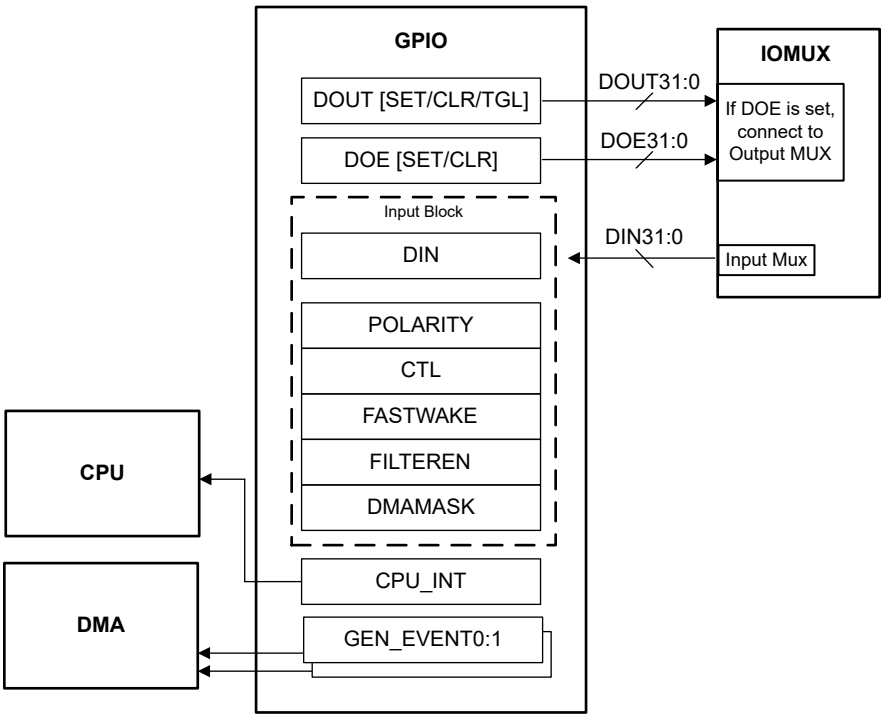


Figure 15-1. GPIO Block Diagram

Note

The GPIO module for the AM13E230x platform does not manage the complete digital IO functionality (for example, pullup, pulldown, or other functional muxing). For complete digital IO control details, refer to IOMUX chapter. Similar to any other peripheral, the GPIO has inputs and outputs (with output enable) that interface with the IOMUX to make connections to the physical device pins.

15.1.2 GPIO Operation

The GPIO peripheral is configured with user software. The GPIO peripheral mainly controls output functionality, but is able to read and filter input signals as long as the digital input on a pin is enabled by the IOMUX. The value of an input or an output on a specified pin is stored within the DIOx bits of the Data Output registers (DOUT31_0) and Data Input registers (DIN31_0) for a given GPIO instance.

ADVANCE INFORMATION

GPIO Outputs

To output a signal on a pin, the output enable bit (DIOx) within the output enable register (DOE31_0) must be set for the specified pin. The output enable bits can be set and cleared directly within the DOE31_0 register, but can also be set and cleared by writing to the data output enable set (DOESET31_0) and data output enable clear (DOECLR 31_0) registers.

Once an output is enabled, the output value can be updated within the DOUT31_0 register. For full explanation of output value manipulation, please see GPIO Read/Write Interface.

GPIO Inputs

To read an input value on a specified pin, the IOMUX must enable the digital input on the pin, allowing the signal to propagate internally to the device. From there, the GPIO can read the value of a specified pin by reading the respective bit within the DIN31_0 register.

As long as the GPIO peripheral is enabled, the GPIO peripheral can read the input of an associated pin as long as the input is enabled by the IOMUX. This means that as long as a pin is enabled for digital input by the IOMUX, regardless of what the peripheral function of the pin is, the GPIO peripheral can read the state of the pin.

In addition to reading the input value of a signal, the GPIO can also filter the input signal to detect pulses greater than a specified amount of ULPCLK cycles. See GPIO Input Glitch Filtering and Synchronization for more details. The GPIO peripheral can also generate interrupts to both the CPU and the DMA based on input edge detection by configuring the polarity registers. More information on interrupts can be found in Event Publishers and Event Manager.

Table 15-1. GPIO Event Publishers

GPIO Instance	Event Publishers
GPIO[0:2]	2x
GPIO3	1x

15.1.2.1 GPIO Ports

Each instance of the GPIO peripheral supports up to 32 data input/output (DIO) bits. For devices with greater than 32 GPIOs, multiple instances of the GPIO peripheral are used to address all of the device pins. Each GPIO instance is associated with a port name to differentiate pins by name. Please see the *Pin Configuration and Functions* section of the device data sheet to determine which GPIO signal is associated with which pin.

Table 15-2. GPIO Instances

GPIO Instance	GPIO Port Name
GPIO0	PA
GPIO1	PB
GPIO2	PC
GPIO3	PD

15.1.2.2 GPIO Read/Write Interface

The GPIO peripheral has features and dedicated registers to allow for advanced bit manipulations without the need to execute a read-modify-write construct in software.

For both input and output data registers, there exists a set of alias registers to allow read/write byte access of a single GPIO pin instead of bit-wise access.

In addition, for output data, there are specific write to set (DOUTSET31_0), clear (DOUTCLR31_0), and toggle (DOUTTGL31_0) registers to set, clear, and toggle the outputs of a pin without the need to read-modify-write, preventing the known state of a GPIO pin from changing while updating the output value.

15.1.2.3 GPIO Input Glitch Filtering and Synchronization

The GPIO module evaluates the state of input pins at the ULPCLK (PD0 bus clock) rate, synchronizing the pin state to ULPCLK through a 2-stage synchronizer before passing the GPIO state to the input glitch filter.

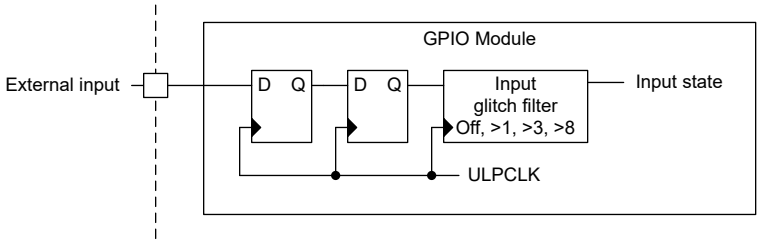


Figure 15-2. GPIO Input Synchronizer

A programmable input glitch filter is provided for suppressing noise on digital input pins. The glitch filter runs at ULPCLK rate. Four levels of user-specified input filtering are possible:

- Sampled input without filtering (the minimum reliably detected pulse width is one ULPCLK cycle due to synchronization)
- Synchronized inputs which are not greater than 1 ULPCLK periods are filtered out (glitch filter value of >1)
- Synchronized inputs which are not greater than 3 ULPCLK periods are filtered out (glitch filter value of >3)
- Synchronized inputs which are not greater than 8 ULPCLK periods are filtered out (glitch filter value of >8)

Note

Inputs are synchronized to ULPCLK, causing up to one ULPCLK cycle of uncertainty in the synchronization. Input signals that are shorter than one ULPCLK cycle can be missed.

Note

For pins with fast wake enable, there is an input delay time equal to the time between the edge of the un-synchronized input edge to the first MCLK edge in STANDBY0/1, STOP1/2 and SLEEP2 modes.

This feature allows users to easily implement input filtering in hardware for cases where fast switching on the input pin is needed to be filtered out. The bit fields in the FILTEREN31_16 and FILTEREN15_0 registers allow users to configure the level of filtering needed for the corresponding GPIO bit.

Input pulses of the same pulse length can be passed in some cases while being filtered in other cases due to the uncertainty in synchronization. GPIO Input Synchronizer and Glitch Filtering Scenarios demonstrates different scenarios and how synchronization affects input detection.

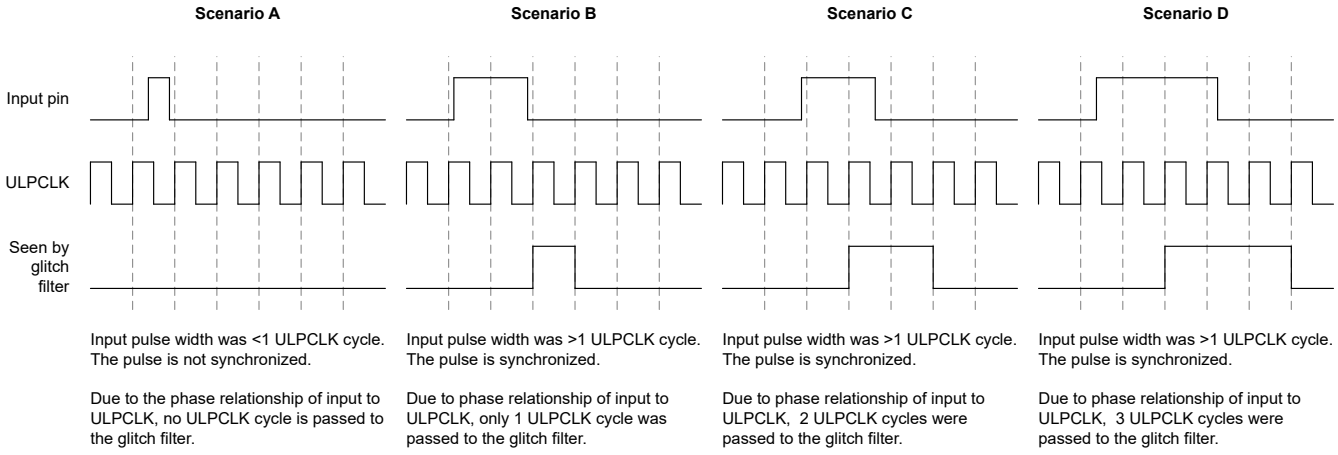


Figure 15-3. GPIO Input Synchronizer and Glitch Filtering Scenarios

ADVANCE INFORMATION

- In Scenario A, the input pulse is less than one ULPCLK cycle. Pulses less than one ULPCLK cycle may not be captured. To ensure that GPIO inputs are always captured, the GPIO input pulse width must be greater than the ULPCLK period.
- In Scenario B, the input pulse is nearly two ULPCLK cycles in length, but because the rising edge occurs just after the ULPCLK edge, the GPIO synchronizer only views the input pin as having been high for 1 ULPCLK period. This scenario would not be filtered out by the glitch filter when the glitch filter is disabled, but a glitch filter value of >1 would result in this pulse being filtered. Conversely, the same input pulse width in Scenario C results in the input pin being considered high for two ULPCLK periods, as the rising edge occurred just before the ULPCLK edge. In this case, this scenario would not be filtered out when a glitch filter value of >1 is specified.
- In Scenario D, three ULPCLK cycles are passed to the glitch filter. In this case, the scenario would not be filtered out when a glitch filter value of >1 is specified, but it would be filtered out for a glitch filter value of >3 or >8.

Note

When the fast wake mode is enabled (SYSOSC is requested asynchronously upon input pin activity), the ULPCLK will switch from off (as would be the case in STANDBY1) or 32kHz (as would be the case in STANDBY0) to 24Mhz/32MHz, resulting in the input synchronization logic and glitch filter running at 24Mhz/32MHz after some latency. See the device specific data sheet for the asynchronous fast clock request wake time, and budget this time into any minimum pulse width calculations when using fast wake.

15.1.2.4 GPIO Fast Wake

The fast wake feature in the GPIO peripheral allows the GPIO module to stay in a low-power state and detect interrupt events on the device pins without requiring a high-speed clock. This allows the device to support fast wakeup from low-power modes, such as STOP and STANDBY, on any GPIO pin.

Fast wake can be enabled on a bit-wise basis using the FASTWAKE register. Setting a bit in the FASTWAKE register enables the corresponding device pin signal to support fast wakeup functionality. The CTL register contains a bit field named FASTWAKEONLY which allows for global control of the fast wake feature. Setting the FASTWAKEONLY bit enables all of the bits in the corresponding GPIO port to support fast wakeup functionality.

Note

Do not enable fast wake in the GPIO while simultaneously blocking asynchronous fast clock requests in SYSCTL. When fast wake is enabled, the GPIO expects to handshake with SYSCTL for the fast clock. If SYSCTL ignores the request, the GPIO does not receive a clock until SYSCTL completes the asynchronous fast clock request handshake.

15.1.2.5 GPIO DMA Interface

The GPIO peripheral allows the DMA write-access to the DOUT31_0 register. This functionality allows users to generate predefined output sequences on specified device pins. Some applications require preloaded sequences of GPIO pin changes. The device also allows for the DMA to run a given sequence so that the CPU can remain asleep and conserve energy.

The DMAMASK register is used to indicate which GPIO bits the DMA is allowed to modify. Setting a bit in the DMAMASK register enables the corresponding DOUT bit to be modified by the DMA.

Note

The CPU can write to any DOUT31_0 bit regardless of the DMAMASK value.

In cases where the DMA and the CPU both attempt to access and modify the DOUT31_0 register concurrently, it is the user's responsibility to manage the DMA and CPU bus transactions that are targeting the same bit to be modified.

- If a DMAMASK bit is set (1), the DMA will have priority when modifying the corresponding DOUT bit.

- If a DMAMASK bit is cleared (0), the CPU will have priority when modifying the corresponding DOUT bit.

15.1.2.6 Event Publishers

The GPIO can trigger interrupt events based on a detected input edge via GPIO event publishers. The Event Manager module provides three event publishers for GPIOx peripherals:

1. GPIOx NVIC IRQ
   - Used for generating a CPU interrupt
   - Interrupt (RIS) flags are cleared upon software reading the IIDX register or writing to the respective ICLR register bits.
   - An event to the CPU can be individually specified for each GPIO bit through the POLARITY registers:
     - 0: Disabled
     - 1: Rise Event
     - 2: Fall Event
     - 3: Rise or Fall Event
2. DMA Fixed Trigger (GPIOx.GEN_EVENT0)
   - Used as a dedicated DMA event trigger for GPIO bits 15 to 0(DIO[15:0])
   - Each GPIO bit can be configured to generate a DMA trigger request based on input detections specified by the POLARITY registers:
     - 0: Disabled
     - 1: Rise Event
     - 2: Fall Event
     - 3: Rise or Fall Event
3. DMA Fixed Trigger (GPIOx.GEN_EVENT1)
   - Used as a dedicated DMA event trigger for GPIO bits 31 to 16(DIO[31:16])
   - Each GPIO bit can be configured to generate a DMA trigger request based on input detection specified by the POLARITY registers:
     - 0: Disabled
     - 1: Rise Event
     - 2: Fall Event
     - 3: Rise or Fall Event

Note

GPIO[0:2] supports both DMA Fixed Trigger Events. GPIO[3] only supports the first DMA Fixed Trigger Event.

15.2 GPIO Registers

This Section describes the GPIO Registers.

15.2.1 GPIO Base Address Table

Table 15-3. GPIO Base Address Table

Bit Field Name		DriverLib Name	Base Address
Instance	Structure		
Gpio0Regs	GPIO_REGS	GPIO0	0x400F_0000
Gpio1Regs	GPIO_REGS	GPIO1	0x400F_2000
Gpio2Regs	GPIO_REGS	GPIO2	0x400F_4000
Gpio3Regs	GPIO_REGS	GPIO3	0x400F_6000

ADVANCE INFORMATION

15.2.2 GPIO_REGS Registers

Table 15-4 lists the memory-mapped registers for the GPIO_REGS registers. All register offset addresses not listed in Table 15-4 should be considered as reserved locations and the register contents should not be modified.

Table 15-4. GPIO_REGS Registers

Offset	Acronym	Register Name	Section
4h + formula	DIO_y	DIO register	Go
204h + formula	DIO_y	DIO register	Go
400h	FSUB_0	Subsciber Port 0	Go
404h	FSUB_1	Subscriber Port 1	Go
444h	FPUB_0	Publisher Port 0	Go
448h	FPUB_1	Publisher Port 1	Go
480h	CPU_CONNECT_0	CPU Connect	Go
800h	PWREN	Power enable	Go
804h	RSTCTL	Reset Control	Go
814h	STAT	Status Register	Go
1010h	CLKOVR	Clock Override	Go
1018h	PDBGCTL	Peripheral Debug Control	Go
1020h	IIDX	Interrupt index	Go
1028h	IMASK	Interrupt mask	Go
1030h	RIS	Raw interrupt status	Go
1038h	MIS	Masked interrupt status	Go
1040h	ISET	Interrupt set	Go
1048h	ICLR	Interrupt clear	Go
1050h	IIDX	Interrupt index	Go
1058h	IMASK	Interrupt mask	Go
1060h	RIS	Raw interrupt status	Go
1068h	MIS	Masked interrupt status	Go
1070h	ISET	Interrupt set	Go
1078h	ICLR	Interrupt clear	Go
1080h	IIDX	Interrupt index	Go
1088h	IMASK	Interrupt mask	Go
1090h	RIS	Raw interrupt status	Go
1098h	MIS	Masked interrupt status	Go
10A0h	ISET	Interrupt set	Go
10A8h	ICLR	Interrupt clear	Go
10E0h	EVT_MODE	Event Mode	Go
10FCh	DESC	Module Description	Go
1200h	DOUT3_0	Data output 3 to 0	Go
1204h	DOUT7_4	Data output 7 to 4	Go
1208h	DOUT11_8	Data output 11 to 8	Go
120Ch	DOUT15_12	Data output 15 to 12	Go
1210h	DOUT19_16	Data output 19 to 16	Go
1214h	DOUT23_20	Data output 23 to 20	Go
1218h	DOUT27_24	Data output 27 to 24	Go
121Ch	DOUT31_28	Data output 31 to 28	Go
1280h	DOUT31_0	Data output 31 to 0	Go

Table 15-4. GPIO_REGS Registers (continued)

Offset	Acronym	Register Name	Section
1290h	DOUTSET31_0	Data output set 31 to 0	Go
12A0h	DOUTCLR31_0	Data output clear 31 to 0	Go
12B0h	DOUTTGL31_0	Data output toggle 31 to 0	Go
12C0h	DOE31_0	Data output enable 31 to 0	Go
12D0h	DOESET31_0	Data output enable set 31 to 0	Go
12E0h	DOECLR31_0	Data output enable clear 31 to 0	Go
1300h	DIN3_0	Data input 3 to 0	Go
1304h	DIN7_4	Data input 7 to 4	Go
1308h	DIN11_8	Data input 11 to 8	Go
130Ch	DIN15_12	Data input 15 to 12	Go
1310h	DIN19_16	Data input 19 to 16	Go
1314h	DIN23_20	Data input 23 to 20	Go
1318h	DIN27_24	Data input 27 to 24	Go
131Ch	DIN31_28	Data input 31 to 28	Go
1380h	DIN31_0	Data input 31 to 0	Go
1390h	POLARITY15_0	Polarity 15 to 0	Go
13A0h	POLARITY31_16	Polarity 31 to 16	Go
1400h	CTL	FAST WAKE GLOBAL EN	Go
1404h	FASTWAKE	FAST WAKE ENABLE	Go
1500h	SUB0CFG	Subscriber 0 configuration	Go
1508h	FILTEREN15_0	Filter Enable 15 to 0	Go
150Ch	FILTEREN31_16	Filter Enable 31 to 16	Go
1510h	DMAMASK	DMA Write MASK	Go
1520h	SUB1CFG	Subscriber 1 configuration	Go
1E00h	TEST0	Test 0	Go

Complex bit access types are encoded to fit into small table cells. Table 15-5 shows the codes that are used for access types in this section.

Table 15-5. GPIO_REGS Access Type Codes

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

Table 15-5. GPIO_REGS Access Type Codes (continued)

Access Type	Code	Description
y		When this variable is used in a register name, an offset, or an address it refers to the value of a register array.

1 DIO_y Register (Offset = 4h + formula) [Reset = 00000000h]

DIO_y is shown in Figure 15-4 and described in Table 15-6.

Return to the Summary Table.

DIO register in the Full Write region.

Offset = 4h + (y * 4h); where y = 0h to 1Fh

Figure 15-4. DIO_y Register

31	30	29	28	27	26	25	24
RESERVED	GFLT	SLEW	WCOMP	WUEN	INV	HIGHZ1	HIGHZ0
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
RESERVED	DRV			HYSTEN	INENA	PIPU	PIPD
R/W-0h	R/W-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
GSTATE		RESERVED					
R/W-0h		R/W-0h					
7	6	5	4	3	2	1	0
PSTATE		RESERVED					
R/W-0h		R/W-0h					

Table 15-7. DIO_y Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	
30	GFLT	R/W	0h	Glitch Filter Enable 0h = No internal glitch filter 1h = Use internal glitch filter
29	SLEW	R/W	0h	Reserved Slew Rate Control 0h = No Slew Rate Control 1h = Use Slew Rate Control
28	WCOMP	R/W	0h	Wake up compare value 0h = Match '0' will wake 1h = Match '1' will wake
27	WUEN	R/W	0h	Wake up enable 0h = Wake up not enabled 1h = Wake up enabled
26	INV	R/W	0h	Invert digital input/output relative to peripheral/GPIO 0h = Input and output are non-inverted 1h = Input and output are inverted
25	HIGHZ1	R/W	0h	High-Z instead of high output 0h = Pin can be driven high 1h = Pin is tri-stated instead of driven high
24	HIGHZ0	R/W	0h	High-Z instead of low output 0h = Pin can be driven low 1h = Pin is tri-stated instead of driven low
23	RESERVED	R/W	0h	

Table 15-7. DIO_y Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
22-20	DRV	R/W	0h	Drive strength options 0h = Lowest drive strength 1h = Drive strength 2/8 2h = Drive strength 3/8 3h = Drive strength 4/8 4h = Drive strength 5/8 5h = Drive strength 6/8 6h = Drive strength 7/8 7h = Highest drive strength
19	HYSTEN	R/W	0h	Hysteresis enable 0h = No hysteresis 1h = Hysteresis on
18	INENA	R/W	0h	Input enable 0h = Inputs 0 to connected core 1h = Inputs IO pad value to connected core
17	PIPU	R/W	0h	Pull up enable 0h = No pull up 1h = Pull up
16	PIPD	R/W	0h	Pull down enable 0h = No pull down 1h = Pull down
15-14	GSTATE	R/W	0h	GPIO Channel State 0h = G-Channel is in Unassigned State 1h = G-Channel is in Handover State 2h = G-Channle is in Connected State and not Locked (That is F field is allowed to change without going back through Unassigned state) 3h = G-Channel is in Connected State and Locked (That is F field is not allowed to change to a different non-Zero value until both G and P channels go to Unassigned)
13-8	RESERVED	R/W	0h	
7-6	PSTATE	R/W	0h	Peripheral-Analog Channel State 0h = P-Channel is in Unassigned State 1h = P-Channel is in Handover State 2h = P-Channle is in Connected State and not Locked (That is F field is allowed to change without going back through Unassigned state) 3h = P-Channel is in Connected State and Locked (That is F field is not allowed to change to a different non-Zero value until both G and P channels go to Unassigned)
5-0	RESERVED	R/W	0h	

2 DIO_y Register (Offset = 204h + formula) [Reset = 00000000h]

DIO_y is shown in Figure 15-5 and described in Table 15-7.

Return to the Summary Table.

FUPDATE version of DIO

Offset = 204h + (y * 4h); where y = 0h to 1Fh

Figure 15-5. DIO_y Register

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

Table 15-9. DIO_y Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	RESERVED	W	0h	
27-2	IOADDR	W	0h	IO Address This is the address that corresponds to the SOC address[27:2] of the module IP instance specific IO signal in the "Full Write" subregion of the pinmux subregion.
1	LOCK	W	0h	Sets lock bit 0h = Writing this value has no effect 1h = Set channel lock bit
0	GSEL	W	0h	GPIO channel Select 0: Select the P-Channel for the F update 1: Select the G-Channel for the F update 0h = Select the P-Channel for the F update 1h = Select the G-Channel for the F update

3 FSUB_0 Register (Offset = 400h) [Reset = 00000000h]

FSUB_0 is shown in Figure 15-6 and described in Table 15-8.

Return to the Summary Table.

Subscriber port

Figure 15-6. FSUB_0 Register

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

Table 15-11. FSUB_0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R/W	0h	
3-0	CHANID	R/W	0h	0 = disconnected. 1-15 = connected to channelID = CHANID. 0h = A value of 0 specifies that the event is not connected Fh = Consult your device datasheet as the actual allowed maximum may be less than 15.

4 FSUB_1 Register (Offset = 404h) [Reset = 00000000h]

FSUB_1 is shown in Figure 15-7 and described in Table 15-9.

Return to the Summary Table.

Subscriber port

Figure 15-7. FSUB_1 Register

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

Table 15-13. FSUB_1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R/W	0h	
3-0	CHANID	R/W	0h	0 = disconnected. 1-15 = connected to channelID = CHANID. 0h = A value of 0 specifies that the event is not connected Fh = Consult your device datasheet as the actual allowed maximum may be less than 15.

5 FPUB_0 Register (Offset = 444h) [Reset = 00000000h]

FPUB_0 is shown in Figure 15-8 and described in Table 15-10.

Return to the Summary Table.

Publisher port

Figure 15-8. FPUB_0 Register

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

Table 15-15. FPUB_0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R/W	0h	
3-0	CHANID	R/W	0h	0 = disconnected. 1-15 = connected to channelID = CHANID. 0h = A value of 0 specifies that the event is not connected Fh = Consult your device datasheet as the actual allowed maximum may be less than 15.

6 FPUB_1 Register (Offset = 448h) [Reset = 00000000h]

FPUB_1 is shown in Figure 15-9 and described in Table 15-11.

Return to the Summary Table.

Publisher port

Figure 15-9. FPUB_1 Register

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

Table 15-17. FPUB_1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R/W	0h	
3-0	CHANID	R/W	0h	0 = disconnected. 1-15 = connected to channelID = CHANID. 0h = A value of 0 specifies that the event is not connected Fh = Consult your device datasheet as the actual allowed maximum may be less than 15.

7 CPU_CONNECT_0 Register (Offset = 480h) [Reset = 00000000h]

CPU_CONNECT_0 is shown in Figure 15-10 and described in Table 15-12.

Return to the Summary Table.

Connect peripheral interrupts / publisher port (FPUB_1) to application processor

Figure 15-10. CPU_CONNECT_0 Register

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

Table 15-19. CPU_CONNECT_0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R/W	0h	
1	RESERVED	R/W	0h	Reserved
0	RESERVED	R/W	0h	

8 PWREN Register (Offset = 800h) [Reset = 00000000h]

PWREN is shown in Figure 15-11 and described in Table 15-13.

Return to the Summary Table.

Register to control the power state

Figure 15-11. PWREN Register

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

Table 15-21. PWREN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	KEY	W	0h	KEY to allow Power State Change 26h = KEY to allow write access to this register
23-1	RESERVED	R/W	0h	
0	ENABLE	R/WK	0h	Enable the power [EXT_GPRCM.GPRCM.PWREN.KEY] must be set to 26h to write to this bit. 0h = Disable Power 1h = Enable Power

ADVANCE INFORMATION

9 RSTCTL Register (Offset = 804h) [Reset = 00000000h]

RSTCTL is shown in Figure 15-12 and described in Table 15-14.

Return to the Summary Table.

Register to control reset assertion and de-assertion

Figure 15-12. RSTCTL Register

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

Table 15-23. RSTCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	KEY	W	0h	Unlock key B1h = KEY to allow write access to this register
23-2	RESERVED	W	0h	
1	RESETSTKYCLR	WK	0h	Clear the RESETSTKY bit in the STAT register [EXT_GPRCM.GPRCM.RSTCTL.KEY] must be set to B1h to write to this bit. 0h = Writing 0 has no effect 1h = Clear reset sticky bit
0	RESETASSERT	WK	0h	Assert reset to the peripheral [EXT_GPRCM.GPRCM.RSTCTL.KEY] must be set to B1h to write to this bit. 0h = Writing 0 has no effect 1h = Assert reset

ADVANCE INFORMATION

10 STAT Register (Offset = 814h) [Reset = 00000000h]

STAT is shown in Figure 15-13 and described in Table 15-15.

Return to the Summary Table.

peripheral enable and reset status

Figure 15-13. STAT Register

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

Table 15-25. STAT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-17	RESERVED	R	0h	
16	RESETSTKY	R	0h	This bit indicates, if the peripheral was reset, since this bit was cleared by RESETSTKYCLR in the RSTCTL register 0h = The peripheral has not been reset since this bit was last cleared by RESETSTKYCLR in the RSTCTL register 1h = The peripheral was reset since the last bit clear
15-0	RESERVED	R	0h	

11 CLKOVR Register (Offset = 1010h) [Reset = 00000000h]

CLKOVR is shown in Figure 15-14 and described in Table 15-16.

Return to the Summary Table.

This register overrides the functional clock request by this peripheral to the system

Figure 15-14. CLKOVR Register

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

Table 15-27. CLKOVR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R/W	0h	
1	RUN_STOP	R/W	0h	If [OVERRIDE] is enabled, this register is used to manually control the peripheral's clock request to the system 0h = Run/ungate functional clock 1h = Stop/gate functional clock
0	OVERRIDE	R/W	0h	Unlocks the functionality of [RUN_STOP] to override the automatic peripheral clock request 0h = Override disabled 1h = Override enabled

ADVANCE INFORMATION

12 PDBGCTL Register (Offset = 1018h) [Reset = 00000001h]

PDBGCTL is shown in Figure 15-15 and described in Table 15-17.

Return to the Summary Table.

This register can be used by the software developer to control the behavior of the peripheral relative to the 'Core Halted' input

Figure 15-15. PDBGCTL Register

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

Table 15-29. PDBGCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R/W	0h	
0	FREE	R/W	1h	Free run control 0h = The peripheral freezes functionality while the Core Halted input is asserted and resumes when it is deasserted. 1h = The peripheral ignores the state of the Core Halted input

ADVANCE INFORMATION

13 IIDX Register (Offset = 1020h) [Reset = 00000000h]

IIDX is shown in Figure 15-16 and described in Table 15-18.

Return to the Summary Table.

This register provides the highest priority enabled interrupt index. Value 0x00 means no event pending. Interrupt 1 is the highest priority, IIDX next highest, 4, 8, ... IIDX^31 is the least priority. That is, the least bit position that is set to 1 denotes the highest priority pending interrupt. The priority order is fixed. However, users can implement their own prioritization schemes using other registers that expose the full set of interrupts that have occurred. On each read, only one interrupt is indicated. On a read, the current interrupt (highest priority) is automatically cleared by the hardware and the corresponding interrupt flag in [RIS] and [MIS] are cleared as well. After a read from the CPU (not from the debug interface), the register is updated with the next highest priority interrupt, if none are pending, then it should display 0x0.

Figure 15-16. IIDX Register

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

Table 15-31. IIDX Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	
7-0	STAT	R	0h	Interrupt index status 0h = No bit is set means there is no pending interrupt request 1h = DIO0 interrupt 2h = DIO1 interrupt 3h = DIO2 interrupt 4h = DIO3 interrupt 5h = DIO4 interrupt 6h = DIO5 interrupt 7h = DIO6 interrupt 8h = DIO7 interrupt 9h = DIO8 interrupt Ah = DIO9 interrupt Bh = DIO10 interrupt Ch = DIO11 interrupt Dh = DIO12 interrupt Eh = DIO13 interrupt Fh = DIO14 interrupt 10h = DIO15 interrupt 11h = DIO16 interrupt 12h = DIO17 interrupt 13h = DIO18 interrupt 14h = DIO19 interrupt 15h = DIO20 interrupt 16h = DIO21 interrupt 17h = DIO22 interrupt 18h = DIO23 interrupt 19h = DIO24 interrupt 1Ah = DIO25 interrupt 1Bh = DIO26 interrupt 1Ch = DIO27 interrupt 1Dh = DIO28 interrupt 1Eh = DIO29 interrupt 1Fh = DIO30 interrupt 20h = DIO31 interrupt

14 IMASK Register (Offset = 1028h) [Reset = 00000000h]

IMASK is shown in Figure 15-17 and described in Table 15-19.

Return to the Summary Table.

Interrupt Mask. If a bit is set, then corresponding interrupt is un-masked. Un-masking the interrupt causes the raw interrupt to be visible in IIDX, as well as MIS.

Figure 15-17. IMASK Register

31	30	29	28	27	26	25	24
DIO31	DIO30	DIO29	DIO28	DIO27	DIO26	DIO25	DIO24
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
DIO23	DIO22	DIO21	DIO20	DIO19	DIO18	DIO17	DIO16
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
DIO15	DIO14	DIO13	DIO12	DIO11	DIO10	DIO9	DIO8
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
DIO7	DIO6	DIO5	DIO4	DIO3	DIO2	DIO1	DIO0
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 15-33. IMASK Register Field Descriptions

Bit	Field	Type	Reset	Description
31	DIO31	R/W	0h	DIO31 event mask 0h = Event is masked 1h = Event is unmasked
30	DIO30	R/W	0h	DIO30 event mask 0h = Event is masked 1h = Event is unmasked
29	DIO29	R/W	0h	DIO29 event mask 0h = Event is masked 1h = Event is unmasked
28	DIO28	R/W	0h	DIO28 event mask 0h = Event is masked 1h = Event is unmasked
27	DIO27	R/W	0h	DIO27 event mask 0h = Event is masked 1h = Event is unmasked
26	DIO26	R/W	0h	DIO26 event mask 0h = Event is masked 1h = Event is unmasked
25	DIO25	R/W	0h	DIO25 event mask 0h = Event is masked 1h = Event is unmasked
24	DIO24	R/W	0h	DIO24 event mask 0h = Event is masked 1h = Event is unmasked
23	DIO23	R/W	0h	DIO23 event mask 0h = Event is masked 1h = Event is unmasked
22	DIO22	R/W	0h	DIO22 event mask 0h = Event is masked 1h = Event is unmasked

Table 15-33. IMASK Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
21	DIO21	R/W	0h	DIO21 event mask 0h = Event is masked 1h = Event is unmasked
20	DIO20	R/W	0h	DIO20 event mask 0h = Event is masked 1h = Event is unmasked
19	DIO19	R/W	0h	DIO19 event mask 0h = Event is masked 1h = Event is unmasked
18	DIO18	R/W	0h	DIO18 event mask 0h = Event is masked 1h = Event is unmasked
17	DIO17	R/W	0h	DIO17 event mask 0h = Event is masked 1h = Event is unmasked
16	DIO16	R/W	0h	DIO16 event mask 0h = Event is masked 1h = Event is unmasked
15	DIO15	R/W	0h	DIO15 event mask 0h = Event is masked 1h = Event is unmasked
14	DIO14	R/W	0h	DIO14 event mask 0h = Event is masked 1h = Event is unmasked
13	DIO13	R/W	0h	DIO13 event mask 0h = Event is masked 1h = Event is unmasked
12	DIO12	R/W	0h	DIO12 event mask 0h = Event is masked 1h = Event is unmasked
11	DIO11	R/W	0h	DIO11 event mask 0h = Event is masked 1h = Event is unmasked
10	DIO10	R/W	0h	DIO10 event mask 0h = Event is masked 1h = Event is unmasked
9	DIO9	R/W	0h	DIO9 event mask 0h = Event is masked 1h = Event is unmasked
8	DIO8	R/W	0h	DIO8 event mask 0h = Event is masked 1h = Event is unmasked
7	DIO7	R/W	0h	DIO7 event mask 0h = Event is masked 1h = Event is unmasked
6	DIO6	R/W	0h	DIO6 event mask 0h = Event is masked 1h = Event is unmasked
5	DIO5	R/W	0h	DIO5 event mask 0h = Event is masked 1h = Event is unmasked
4	DIO4	R/W	0h	DIO4 event mask 0h = Event is masked 1h = Event is unmasked
3	DIO3	R/W	0h	DIO3 event mask 0h = Event is masked 1h = Event is unmasked

Table 15-33. IMASK Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
2	DIO2	R/W	0h	DIO2 event mask 0h = Event is masked 1h = Event is unmasked
1	DIO1	R/W	0h	DIO1 event mask 0h = Event is masked 1h = Event is unmasked
0	DIO0	R/W	0h	DIO0 event mask 0h = Event is masked 1h = Event is unmasked

15 RIS Register (Offset = 1030h) [Reset = 00000000h]

RIS is shown in Figure 15-18 and described in Table 15-20.

Return to the Summary Table.

Raw interrupt status. Reflects all pending interrupts, regardless of masking. The RIS register allows the user to implement a poll scheme. A flag set in this register can be cleared by writing 1 to the ICLR register bit even if the corresponding IMASK bit is not enabled.

Figure 15-18. RIS Register

31	30	29	28	27	26	25	24
DIO31	DIO30	DIO29	DIO28	DIO27	DIO26	DIO25	DIO24
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
23	22	21	20	19	18	17	16
DIO23	DIO22	DIO21	DIO20	DIO19	DIO18	DIO17	DIO16
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
15	14	13	12	11	10	9	8
DIO15	DIO14	DIO13	DIO12	DIO11	DIO10	DIO9	DIO8
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
7	6	5	4	3	2	1	0
DIO7	DIO6	DIO5	DIO4	DIO3	DIO2	DIO1	DIO0
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

Table 15-35. RIS Register Field Descriptions

Bit	Field	Type	Reset	Description
31	DIO31	R	0h	DIO31 event 0h = DIO31 event did not occur 1h = DIO31 event occurred
30	DIO30	R	0h	DIO30 event 0h = DIO30 event did not occur 1h = DIO30 event occurred
29	DIO29	R	0h	DIO29 event 0h = DIO29 event did not occur 1h = DIO29 event occurred
28	DIO28	R	0h	DIO28 event 0h = DIO28 event did not occur 1h = DIO28 event occurred
27	DIO27	R	0h	DIO27 event 0h = DIO27 event did not occur 1h = DIO27 event occurred
26	DIO26	R	0h	DIO26 event 0h = DIO26 event did not occur 1h = DIO26 event occurred
25	DIO25	R	0h	DIO25 event 0h = DIO25 event did not occur 1h = DIO25 event occurred
24	DIO24	R	0h	DIO24 event 0h = DIO24 event did not occur 1h = DIO24 event occurred
23	DIO23	R	0h	DIO23 event 0h = DIO23 event did not occur 1h = DIO23 event occurred
22	DIO22	R	0h	DIO22 event 0h = DIO22 event did not occur 1h = DIO22 event occurred

Table 15-35. RIS Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
21	DIO21	R	0h	DIO21 event 0h = DIO21 event did not occur 1h = DIO21 event occurred
20	DIO20	R	0h	DIO20 event 0h = DIO20 event did not occur 1h = DIO20 event occurred
19	DIO19	R	0h	DIO19 event 0h = DIO19 event did not occur 1h = DIO19 event occurred
18	DIO18	R	0h	DIO18 event 0h = DIO18 event did not occur 1h = DIO18 event occurred
17	DIO17	R	0h	DIO17 event 0h = DIO17 event did not occur 1h = DIO17 event occurred
16	DIO16	R	0h	DIO16 event 0h = DIO16 event did not occur 1h = DIO16 event occurred
15	DIO15	R	0h	DIO15 event 0h = DIO15 event did not occur 1h = DIO15 event occurred
14	DIO14	R	0h	DIO14 event 0h = DIO14 event did not occur 1h = DIO14 event occurred
13	DIO13	R	0h	DIO13 event 0h = DIO13 event did not occur 1h = DIO13 event occurred
12	DIO12	R	0h	DIO12 event 0h = DIO12 event did not occur 1h = DIO12 event occurred
11	DIO11	R	0h	DIO11 event 0h = DIO11 event did not occur 1h = DIO11 event occurred
10	DIO10	R	0h	DIO10 event 0h = DIO10 event did not occur 1h = DIO10 event occurred
9	DIO9	R	0h	DIO9 event 0h = DIO9 event did not occur 1h = DIO9 event occurred
8	DIO8	R	0h	DIO8 event 0h = DIO8 event did not occur 1h = DIO8 event occurred
7	DIO7	R	0h	DIO7 event 0h = DIO7 event did not occur 1h = DIO7 event occurred
6	DIO6	R	0h	DIO6 event 0h = DIO6 event did not occur 1h = DIO6 event occurred
5	DIO5	R	0h	DIO5 event 0h = DIO5 event did not occur 1h = DIO5 event occurred
4	DIO4	R	0h	DIO4 event 0h = DIO4 event did not occur 1h = DIO4 event occurred
3	DIO3	R	0h	DIO3 event 0h = DIO3 event did not occur 1h = DIO3 event occurred

Table 15-35. RIS Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
2	DIO2	R	0h	DIO2 event 0h = DIO2 event did not occur 1h = DIO2 event occurred
1	DIO1	R	0h	DIO1 event 0h = DIO1 event did not occur 1h = DIO1 event occurred
0	DIO0	R	0h	DIO0 event 0h = DIO0 event did not occur 1h = DIO0 event occurred

16 MIS Register (Offset = 1038h) [Reset = 00000000h]

MIS is shown in Figure 15-19 and described in Table 15-21.

Return to the Summary Table.

Masked interrupt status. This is an AND of the IMASK and RIS registers.

Figure 15-19. MIS Register

31	30	29	28	27	26	25	24
DIO31	DIO30	DIO29	DIO28	DIO27	DIO26	DIO25	DIO24
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
23	22	21	20	19	18	17	16
DIO23	DIO22	DIO21	DIO20	DIO19	DIO18	DIO17	DIO16
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
15	14	13	12	11	10	9	8
DIO15	DIO14	DIO13	DIO12	DIO11	DIO10	DIO9	DIO8
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
7	6	5	4	3	2	1	0
DIO7	DIO6	DIO5	DIO4	DIO3	DIO2	DIO1	DIO0
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

Table 15-37. MIS Register Field Descriptions

Bit	Field	Type	Reset	Description
31	DIO31	R	0h	DIO31 event 0h = DIO31 event did not occur 1h = DIO31 event occurred
30	DIO30	R	0h	DIO30 event 0h = DIO30 event did not occur 1h = DIO30 event occurred
29	DIO29	R	0h	DIO29 event 0h = DIO29 event did not occur 1h = DIO29 event occurred
28	DIO28	R	0h	DIO28 event 0h = DIO28 event did not occur 1h = DIO28 event occurred
27	DIO27	R	0h	DIO27 event 0h = DIO27 event did not occur 1h = DIO27 event occurred
26	DIO26	R	0h	DIO26 event 0h = DIO26 event did not occur 1h = DIO26 event occurred
25	DIO25	R	0h	DIO25 event 0h = DIO25 event did not occur 1h = DIO25 event occurred
24	DIO24	R	0h	DIO24 event 0h = DIO24 event did not occur 1h = DIO24 event occurred
23	DIO23	R	0h	DIO23 event 0h = DIO23 event did not occur 1h = DIO23 event occurred
22	DIO22	R	0h	DIO22 event 0h = DIO22 event did not occur 1h = DIO22 event occurred

Table 15-37. MIS Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
21	DIO21	R	0h	DIO21 event 0h = DIO21 event did not occur 1h = DIO21 event occurred
20	DIO20	R	0h	DIO20 event 0h = DIO20 event did not occur 1h = DIO20 event occurred
19	DIO19	R	0h	DIO19 event 0h = DIO19 event did not occur 1h = DIO19 event occurred
18	DIO18	R	0h	DIO18 event 0h = DIO18 event did not occur 1h = DIO18 event occurred
17	DIO17	R	0h	DIO17 event 0h = DIO17 event did not occur 1h = DIO17 event occurred
16	DIO16	R	0h	DIO16 event 0h = DIO16 event did not occur 1h = DIO16 event occurred
15	DIO15	R	0h	DIO15 event 0h = DIO15 event did not occur 1h = DIO15 event occurred
14	DIO14	R	0h	DIO14 event 0h = DIO14 event did not occur 1h = DIO14 event occurred
13	DIO13	R	0h	DIO13 event 0h = DIO13 event did not occur 1h = DIO13 event occurred
12	DIO12	R	0h	DIO12 event 0h = DIO12 event did not occur 1h = DIO12 event occurred
11	DIO11	R	0h	DIO11 event 0h = DIO11 event did not occur 1h = DIO11 event occurred
10	DIO10	R	0h	DIO10 event 0h = DIO10 event did not occur 1h = DIO10 event occurred
9	DIO9	R	0h	DIO9 event 0h = DIO9 event did not occur 1h = DIO9 event occurred
8	DIO8	R	0h	DIO8 event 0h = DIO8 event did not occur 1h = DIO8 event occurred
7	DIO7	R	0h	DIO7 event 0h = DIO7 event did not occur 1h = DIO7 event occurred
6	DIO6	R	0h	DIO6 event 0h = DIO6 event did not occur 1h = DIO6 event occurred
5	DIO5	R	0h	DIO5 event 0h = DIO5 event did not occur 1h = DIO5 event occurred
4	DIO4	R	0h	DIO4 event 0h = DIO4 event did not occur 1h = DIO4 event occurred
3	DIO3	R	0h	DIO3 event 0h = DIO3 event did not occur 1h = DIO3 event occurred

Table 15-37. MIS Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
2	DIO2	R	0h	DIO2 event 0h = DIO2 event did not occur 1h = DIO2 event occurred
1	DIO1	R	0h	DIO1 event 0h = DIO1 event did not occur 1h = DIO1 event occurred
0	DIO0	R	0h	DIO0 event 0h = DIO0 event did not occur 1h = DIO0 event occurred

17 ISET Register (Offset = 1040h) [Reset = 00000000h]

ISET is shown in Figure 15-20 and described in Table 15-22.

Return to the Summary Table.

Interrupt set. Allows interrupts to be set by software (useful in diagnostics and safety checks). Writing a 1 to a bit in ISET will set the event and therefore the related RIS bit also gets set. If the interrupt is enabled through the mask, then the corresponding MIS bit is also set.

Figure 15-20. ISET Register

31	30	29	28	27	26	25	24
DIO31	DIO30	DIO29	DIO28	DIO27	DIO26	DIO25	DIO24
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h
23	22	21	20	19	18	17	16
DIO23	DIO22	DIO21	DIO20	DIO19	DIO18	DIO17	DIO16
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h
15	14	13	12	11	10	9	8
DIO15	DIO14	DIO13	DIO12	DIO11	DIO10	DIO9	DIO8
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h
7	6	5	4	3	2	1	0
DIO7	DIO6	DIO5	DIO4	DIO3	DIO2	DIO1	DIO0
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h

Table 15-39. ISET Register Field Descriptions

Bit	Field	Type	Reset	Description
31	DIO31	W	0h	DIO31 event 0h = No effect 1h = Sets DIO31 in RIS register
30	DIO30	W	0h	DIO30 event 0h = No effect 1h = Sets DIO30 in RIS register
29	DIO29	W	0h	DIO29 event 0h = No effect 1h = Sets DIO29 in RIS register
28	DIO28	W	0h	DIO28 event 0h = No effect 1h = Sets DIO28 in RIS register
27	DIO27	W	0h	DIO27 event 0h = No effect 1h = Sets DIO27 in RIS register
26	DIO26	W	0h	DIO26 event 0h = No effect 1h = Sets DIO26 in RIS register
25	DIO25	W	0h	DIO25 event 0h = No effect 1h = Sets DIO25 in RIS register
24	DIO24	W	0h	DIO24 event 0h = No effect 1h = Sets DIO24 in RIS register
23	DIO23	W	0h	DIO23 event 0h = No effect 1h = Sets DIO23 in RIS register
22	DIO22	W	0h	DIO22 event 0h = No effect 1h = Sets DIO22 in RIS register

Table 15-39. ISET Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
21	DIO21	W	0h	DIO21 event 0h = No effect 1h = Sets DIO21 in RIS register
20	DIO20	W	0h	DIO20 event 0h = No effect 1h = Sets DIO20 in RIS register
19	DIO19	W	0h	DIO19 event 0h = No effect 1h = Sets DIO19 in RIS register
18	DIO18	W	0h	DIO18 event 0h = No effect 1h = Sets DIO18 in RIS register
17	DIO17	W	0h	DIO17 event 0h = No effect 1h = Sets DIO17 in RIS register
16	DIO16	W	0h	DIO16 event 0h = No effect 1h = Sets DIO16 in RIS register
15	DIO15	W	0h	DIO15 event 0h = No effect 1h = Sets DIO15 in RIS register
14	DIO14	W	0h	DIO14 event 0h = No effect 1h = Sets DIO14 in RIS register
13	DIO13	W	0h	DIO13 event 0h = No effect 1h = Sets DIO13 in RIS register
12	DIO12	W	0h	DIO12 event 0h = No effect 1h = Sets DIO12 in RIS register
11	DIO11	W	0h	DIO11 event 0h = No effect 1h = Sets DIO11 in RIS register
10	DIO10	W	0h	DIO10 event 0h = No effect 1h = Sets DIO10 in RIS register
9	DIO9	W	0h	DIO9 event 0h = No effect 1h = Sets DIO9 in RIS register
8	DIO8	W	0h	DIO8 event 0h = No effect 1h = Sets DIO8 in RIS register
7	DIO7	W	0h	DIO7 event 0h = No effect 1h = Sets DIO7 in RIS register
6	DIO6	W	0h	DIO6 event 0h = No effect 1h = Sets DIO6 in RIS register
5	DIO5	W	0h	DIO5 event 0h = No effect 1h = Sets DIO5 in RIS register
4	DIO4	W	0h	DIO4 event 0h = No effect 1h = Sets DIO4 in RIS register
3	DIO3	W	0h	DIO3 event 0h = No effect 1h = Sets DIO3 in RIS register

Table 15-39. ISET Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
2	DIO2	W	0h	DIO2 event 0h = No effect 1h = Sets DIO2 in RIS register
1	DIO1	W	0h	DIO1 event 0h = No effect 1h = Sets DIO1 in RIS register
0	DIO0	W	0h	DIO0 event 0h = No effect 1h = Sets DIO0 in RIS register

18 ICLR Register (Offset = 1048h) [Reset = 00000000h]

ICLR is shown in Figure 15-21 and described in Table 15-23.

Return to the Summary Table.

Interrupt clear. Write a 1 to clear corresponding Interrupt.

Figure 15-21. ICLR Register

31	30	29	28	27	26	25	24
DIO31	DIO30	DIO29	DIO28	DIO27	DIO26	DIO25	DIO24
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h
23	22	21	20	19	18	17	16
DIO23	DIO22	DIO21	DIO20	DIO19	DIO18	DIO17	DIO16
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h
15	14	13	12	11	10	9	8
DIO15	DIO14	DIO13	DIO12	DIO11	DIO10	DIO9	DIO8
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h
7	6	5	4	3	2	1	0
DIO7	DIO6	DIO5	DIO4	DIO3	DIO2	DIO1	DIO0
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h

Table 15-41. ICLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31	DIO31	W	0h	DIO31 event 0h = No effect 1h = Clears DIO31 in RIS register
30	DIO30	W	0h	DIO30 event 0h = No effect 1h = Clears DIO30 in RIS register
29	DIO29	W	0h	DIO29 event 0h = No effect 1h = Clears DIO29 in RIS register
28	DIO28	W	0h	DIO28 event 0h = No effect 1h = Clears DIO28 in RIS register
27	DIO27	W	0h	DIO27 event 0h = No effect 1h = Clears DIO27 in RIS register
26	DIO26	W	0h	DIO26 event 0h = No effect 1h = Clears DIO26 in RIS register
25	DIO25	W	0h	DIO25 event 0h = No effect 1h = Clears DIO25 in RIS register
24	DIO24	W	0h	DIO24 event 0h = No effect 1h = Clears DIO24 in RIS register
23	DIO23	W	0h	DIO23 event 0h = No effect 1h = Clears DIO23 in RIS register
22	DIO22	W	0h	DIO22 event 0h = No effect 1h = Clears DIO22 in RIS register

Table 15-41. ICLR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
21	DIO21	W	0h	DIO21 event 0h = No effect 1h = Clears DIO21 in RIS register
20	DIO20	W	0h	DIO20 event 0h = No effect 1h = Clears DIO20 in RIS register
19	DIO19	W	0h	DIO19 event 0h = No effect 1h = Clears DIO19 in RIS register
18	DIO18	W	0h	DIO18 event 0h = No effect 1h = Clears DIO18 in RIS register
17	DIO17	W	0h	DIO17 event 0h = No effect 1h = Clears DIO17 in RIS register
16	DIO16	W	0h	DIO16 event 0h = No effect 1h = Clears DIO16 in RIS register
15	DIO15	W	0h	DIO15 event 0h = No effect 1h = Clears DIO15 in RIS register
14	DIO14	W	0h	DIO14 event 0h = No effect 1h = Clears DIO14 in RIS register
13	DIO13	W	0h	DIO13 event 0h = No effect 1h = Clears DIO13 in RIS register
12	DIO12	W	0h	DIO12 event 0h = No effect 1h = Clears DIO12 in RIS register
11	DIO11	W	0h	DIO11 event 0h = No effect 1h = Clears DIO11 in RIS register
10	DIO10	W	0h	DIO10 event 0h = No effect 1h = Clears DIO10 in RIS register
9	DIO9	W	0h	DIO9 event 0h = No effect 1h = Clears DIO9 in RIS register
8	DIO8	W	0h	DIO8 event 0h = No effect 1h = Clears DIO8 in RIS register
7	DIO7	W	0h	DIO7 event 0h = No effect 1h = Clears DIO7 in RIS register
6	DIO6	W	0h	DIO6 event 0h = No effect 1h = Clears DIO6 in RIS register
5	DIO5	W	0h	DIO5 event 0h = No effect 1h = Clears DIO5 in RIS register
4	DIO4	W	0h	DIO4 event 0h = No effect 1h = Clears DIO4 in RIS register
3	DIO3	W	0h	DIO3 event 0h = No effect 1h = Clears DIO3 in RIS register

Table 15-41. ICLR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
2	DIO2	W	0h	DIO2 event 0h = No effect 1h = Clears DIO2 in RIS register
1	DIO1	W	0h	DIO1 event 0h = No effect 1h = Clears DIO1 in RIS register
0	DIO0	W	0h	DIO0 event 0h = No effect 1h = Clears DIO0 in RIS register

19 IIDX Register (Offset = 1050h) [Reset = 00000000h]

IIDX is shown in Figure 15-22 and described in Table 15-24.

Return to the Summary Table.

This register provides the highest priority enabled interrupt index. Value 0x00 means no event pending. Interrupt 1 is the highest priority, IIDX next highest, 4, 8, ... IIDX^31 is the least priority. That is, the least bit position that is set to 1 denotes the highest priority pending interrupt. The priority order is fixed. However, users can implement their own prioritization schemes using other registers that expose the full set of interrupts that have occurred. On each read, only one interrupt is indicated. On a read, the current interrupt (highest priority) is automatically cleared by the hardware and the corresponding interrupt flag in [RIS] and [MIS] are cleared as well. After a read from the CPU (not from the debug interface), the register is updated with the next highest priority interrupt, if none are pending, then it should display 0x0.

Figure 15-22. IIDX Register

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

Table 15-43. IIDX Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	
7-0	STAT	R	0h	Interrupt index status 0h = No bit is set means there is no pending interrupt request 1h = DIO0 interrupt 2h = DIO1 interrupt 3h = DIO2 interrupt 4h = DIO3 interrupt 5h = DIO4 interrupt 6h = DIO5 interrupt 7h = DIO6 interrupt 8h = DIO7 interrupt 9h = DIO8 interrupt Ah = DIO9 interrupt Bh = DIO10 interrupt Ch = DIO11 interrupt Dh = DIO12 interrupt Eh = DIO13 interrupt Fh = DIO14 interrupt 10h = DIO15 interrupt

20 IMASK Register (Offset = 1058h) [Reset = 00000000h]

IMASK is shown in Figure 15-23 and described in Table 15-25.

Return to the Summary Table.

Interrupt Mask. If a bit is set, then corresponding interrupt is un-masked. Un-masking the interrupt causes the raw interrupt to be visible in IIDX, as well as MIS.

Figure 15-23. IMASK Register

31	30	29	28	27	26	25	24
RESERVED							
R/W-0h							
23	22	21	20	19	18	17	16
RESERVED							
R/W-0h							
15	14	13	12	11	10	9	8
DIO15	DIO14	DIO13	DIO12	DIO11	DIO10	DIO9	DIO8
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
DIO7	DIO6	DIO5	DIO4	DIO3	DIO2	DIO1	DIO0
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 15-45. IMASK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R/W	0h	
15	DIO15	R/W	0h	DIO15 event mask 0h = Event is masked 1h = Event is unmasked
14	DIO14	R/W	0h	DIO14 event mask 0h = Event is masked 1h = Event is unmasked
13	DIO13	R/W	0h	DIO13 event mask 0h = Event is masked 1h = Event is unmasked
12	DIO12	R/W	0h	DIO12 event mask 0h = Event is masked 1h = Event is unmasked
11	DIO11	R/W	0h	DIO11 event mask 0h = Event is masked 1h = Event is unmasked
10	DIO10	R/W	0h	DIO10 event mask 0h = Event is masked 1h = Event is unmasked
9	DIO9	R/W	0h	DIO9 event mask 0h = Event is masked 1h = Event is unmasked
8	DIO8	R/W	0h	DIO8 event mask 0h = Event is masked 1h = Event is unmasked
7	DIO7	R/W	0h	DIO7 event mask 0h = Event is masked 1h = Event is unmasked
6	DIO6	R/W	0h	DIO6 event mask 0h = Event is masked 1h = Event is unmasked

Table 15-45. IMASK Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
5	DIO5	R/W	0h	DIO5 event mask 0h = Event is masked 1h = Event is unmasked
4	DIO4	R/W	0h	DIO4 event mask 0h = Event is masked 1h = Event is unmasked
3	DIO3	R/W	0h	DIO3 event mask 0h = Event is masked 1h = Event is unmasked
2	DIO2	R/W	0h	DIO2 event mask 0h = Event is masked 1h = Event is unmasked
1	DIO1	R/W	0h	DIO1 event mask 0h = Event is masked 1h = Event is unmasked
0	DIO0	R/W	0h	DIO0 event mask 0h = Event is masked 1h = Event is unmasked

21 RIS Register (Offset = 1060h) [Reset = 00000000h]

RIS is shown in Figure 15-24 and described in Table 15-26.

Return to the Summary Table.

Raw interrupt status. Reflects all pending interrupts, regardless of masking. The RIS register allows the user to implement a poll scheme. A flag set in this register can be cleared by writing 1 to the ICLR register bit even if the corresponding IMASK bit is not enabled.

Figure 15-24. RIS Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
DIO15	DIO14	DIO13	DIO12	DIO11	DIO10	DIO9	DIO8
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
7	6	5	4	3	2	1	0
DIO7	DIO6	DIO5	DIO4	DIO3	DIO2	DIO1	DIO0
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

Table 15-47. RIS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	
15	DIO15	R	0h	DIO15 event 0h = DIO15 event did not occur 1h = DIO15 event occurred
14	DIO14	R	0h	DIO14 event 0h = DIO14 event did not occur 1h = DIO14 event occurred
13	DIO13	R	0h	DIO13 event 0h = DIO13 event did not occur 1h = DIO13 event occurred
12	DIO12	R	0h	DIO12 event 0h = DIO12 event did not occur 1h = DIO12 event occurred
11	DIO11	R	0h	DIO11 event 0h = DIO11 event did not occur 1h = DIO11 event occurred
10	DIO10	R	0h	DIO10 event 0h = DIO10 event did not occur 1h = DIO10 event occurred
9	DIO9	R	0h	DIO9 event 0h = DIO9 event did not occur 1h = DIO9 event occurred
8	DIO8	R	0h	DIO8 event 0h = DIO8 event did not occur 1h = DIO8 event occurred
7	DIO7	R	0h	DIO7 event 0h = DIO7 event did not occur 1h = DIO7 event occurred

Table 15-47. RIS Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
6	DIO6	R	0h	DIO6 event 0h = DIO6 event did not occur 1h = DIO6 event occurred
5	DIO5	R	0h	DIO5 event 0h = DIO5 event did not occur 1h = DIO5 event occurred
4	DIO4	R	0h	DIO4 event 0h = DIO4 event did not occur 1h = DIO4 event occurred
3	DIO3	R	0h	DIO3 event 0h = DIO3 event did not occur 1h = DIO3 event occurred
2	DIO2	R	0h	DIO2 event 0h = DIO2 event did not occur 1h = DIO2 event occurred
1	DIO1	R	0h	DIO1 event 0h = DIO1 event did not occur 1h = DIO1 event occurred
0	DIO0	R	0h	DIO0 event 0h = DIO0 event did not occur 1h = DIO0 event occurred

22 MIS Register (Offset = 1068h) [Reset = 00000000h]

MIS is shown in Figure 15-25 and described in Table 15-27.

Return to the Summary Table.

Masked interrupt status. This is an AND of the IMASK and RIS registers.

Figure 15-25. MIS Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
DIO15	DIO14	DIO13	DIO12	DIO11	DIO10	DIO9	DIO8
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
7	6	5	4	3	2	1	0
DIO7	DIO6	DIO5	DIO4	DIO3	DIO2	DIO1	DIO0
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

Table 15-49. MIS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	
15	DIO15	R	0h	DIO15 event 0h = DIO15 event did not occur 1h = DIO15 event occurred
14	DIO14	R	0h	DIO14 event 0h = DIO14 event did not occur 1h = DIO14 event occurred
13	DIO13	R	0h	DIO13 event 0h = DIO13 event did not occur 1h = DIO13 event occurred
12	DIO12	R	0h	DIO12 event 0h = DIO12 event did not occur 1h = DIO12 event occurred
11	DIO11	R	0h	DIO11 event 0h = DIO11 event did not occur 1h = DIO11 event occurred
10	DIO10	R	0h	DIO10 event 0h = DIO10 event did not occur 1h = DIO10 event occurred
9	DIO9	R	0h	DIO9 event 0h = DIO9 event did not occur 1h = DIO9 event occurred
8	DIO8	R	0h	DIO8 event 0h = DIO8 event did not occur 1h = DIO8 event occurred
7	DIO7	R	0h	DIO7 event 0h = DIO7 event did not occur 1h = DIO7 event occurred
6	DIO6	R	0h	DIO6 event 0h = DIO6 event did not occur 1h = DIO6 event occurred

Table 15-49. MIS Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
5	DIO5	R	0h	DIO5 event 0h = DIO5 event did not occur 1h = DIO5 event occurred
4	DIO4	R	0h	DIO4 event 0h = DIO4 event did not occur 1h = DIO4 event occurred
3	DIO3	R	0h	DIO3 event 0h = DIO3 event did not occur 1h = DIO3 event occurred
2	DIO2	R	0h	DIO2 event 0h = DIO2 event did not occur 1h = DIO2 event occurred
1	DIO1	R	0h	DIO1 event 0h = DIO1 event did not occur 1h = DIO1 event occurred
0	DIO0	R	0h	DIO0 event 0h = DIO0 event did not occur 1h = DIO0 event occurred

23 ISET Register (Offset = 1070h) [Reset = 00000000h]

ISET is shown in Figure 15-26 and described in Table 15-28.

Return to the Summary Table.

Interrupt set. Allows interrupts to be set by software (useful in diagnostics and safety checks). Writing a 1 to a bit in ISET will set the event and therefore the related RIS bit also gets set. If the interrupt is enabled through the mask, then the corresponding MIS bit is also set.

Figure 15-26. ISET Register

31	30	29	28	27	26	25	24
RESERVED							
W-0h							
23	22	21	20	19	18	17	16
RESERVED							
W-0h							
15	14	13	12	11	10	9	8
DIO15	DIO14	DIO13	DIO12	DIO11	DIO10	DIO9	DIO8
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h
7	6	5	4	3	2	1	0
DIO7	DIO6	DIO5	DIO4	DIO3	DIO2	DIO1	DIO0
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h

Table 15-51. ISET Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	W	0h	
15	DIO15	W	0h	DIO15 event 0h = No effect 1h = Sets DIO15 in RIS register
14	DIO14	W	0h	DIO14 event 0h = No effect 1h = Sets DIO14 in RIS register
13	DIO13	W	0h	DIO13 event 0h = No effect 1h = Sets DIO13 in RIS register
12	DIO12	W	0h	DIO12 event 0h = No effect 1h = Sets DIO12 in RIS register
11	DIO11	W	0h	DIO11 event 0h = No effect 1h = Sets DIO11 in RIS register
10	DIO10	W	0h	DIO10 event 0h = No effect 1h = Sets DIO10 in RIS register
9	DIO9	W	0h	DIO9 event 0h = No effect 1h = Sets DIO9 in RIS register
8	DIO8	W	0h	DIO8 event 0h = No effect 1h = Sets DIO8 in RIS register
7	DIO7	W	0h	DIO7 event 0h = No effect 1h = Sets DIO7 in RIS register

Table 15-51. ISET Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
6	DIO6	W	0h	DIO6 event 0h = No effect 1h = Sets DIO6 in RIS register
5	DIO5	W	0h	DIO5 event 0h = No effect 1h = Sets DIO5 in RIS register
4	DIO4	W	0h	DIO4 event 0h = No effect 1h = Sets DIO4 in RIS register
3	DIO3	W	0h	DIO3 event 0h = No effect 1h = Sets DIO3 in RIS register
2	DIO2	W	0h	DIO2 event 0h = No effect 1h = Sets DIO2 in RIS register
1	DIO1	W	0h	DIO1 event 0h = No effect 1h = Sets DIO1 in RIS register
0	DIO0	W	0h	DIO0 event 0h = No effect 1h = Sets DIO0 in RIS register

24 ICLR Register (Offset = 1078h) [Reset = 00000000h]

ICLR is shown in Figure 15-27 and described in Table 15-29.

Return to the Summary Table.

Interrupt clear. Write a 1 to clear corresponding Interrupt.

Figure 15-27. ICLR Register

31	30	29	28	27	26	25	24
RESERVED							
W-0h							
23	22	21	20	19	18	17	16
RESERVED							
W-0h							
15	14	13	12	11	10	9	8
DIO15	DIO14	DIO13	DIO12	DIO11	DIO10	DIO9	DIO8
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h
7	6	5	4	3	2	1	0
DIO7	DIO6	DIO5	DIO4	DIO3	DIO2	DIO1	DIO0
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h

Table 15-53. ICLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	W	0h	
15	DIO15	W	0h	DIO15 event 0h = No effect 1h = Clears DIO15 in RIS register
14	DIO14	W	0h	DIO14 event 0h = No effect 1h = Clears DIO14 in RIS register
13	DIO13	W	0h	DIO13 event 0h = No effect 1h = Clears DIO13 in RIS register
12	DIO12	W	0h	DIO12 event 0h = No effect 1h = Clears DIO12 in RIS register
11	DIO11	W	0h	DIO11 event 0h = No effect 1h = Clears DIO11 in RIS register
10	DIO10	W	0h	DIO10 event 0h = No effect 1h = Clears DIO10 in RIS register
9	DIO9	W	0h	DIO9 event 0h = No effect 1h = Clears DIO9 in RIS register
8	DIO8	W	0h	DIO8 event 0h = No effect 1h = Clears DIO8 in RIS register
7	DIO7	W	0h	DIO7 event 0h = No effect 1h = Clears DIO7 in RIS register
6	DIO6	W	0h	DIO6 event 0h = No effect 1h = Clears DIO6 in RIS register

Table 15-53. ICLR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
5	DIO5	W	0h	DIO5 event 0h = No effect 1h = Clears DIO5 in RIS register
4	DIO4	W	0h	DIO4 event 0h = No effect 1h = Clears DIO4 in RIS register
3	DIO3	W	0h	DIO3 event 0h = No effect 1h = Clears DIO3 in RIS register
2	DIO2	W	0h	DIO2 event 0h = No effect 1h = Clears DIO2 in RIS register
1	DIO1	W	0h	DIO1 event 0h = No effect 1h = Clears DIO1 in RIS register
0	DIO0	W	0h	DIO0 event 0h = No effect 1h = Clears DIO0 in RIS register

25 IIDX Register (Offset = 1080h) [Reset = 00000000h]

IIDX is shown in Figure 15-28 and described in Table 15-30.

Return to the Summary Table.

This register provides the highest priority enabled interrupt index. Value 0x00 means no event pending. Interrupt 1 is the highest priority, IIDX next highest, 4, 8, ... IIDX^31 is the least priority. That is, the least bit position that is set to 1 denotes the highest priority pending interrupt. The priority order is fixed. However, users can implement their own prioritization schemes using other registers that expose the full set of interrupts that have occurred. On each read, only one interrupt is indicated. On a read, the current interrupt (highest priority) is automatically cleared by the hardware and the corresponding interrupt flag in [RIS] and [MIS] are cleared as well. After a read from the CPU (not from the debug interface), the register is updated with the next highest priority interrupt, if none are pending, then it should display 0x0.

Figure 15-28. IIDX Register

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

Table 15-55. IIDX Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	
7-0	STAT	R	0h	Interrupt index status 0h = No bit is set means there is no pending interrupt request 1h = DIO0 interrupt 2h = DIO1 interrupt 3h = DIO2 interrupt 4h = DIO3 interrupt 5h = DIO4 interrupt 6h = DIO5 interrupt 7h = DIO6 interrupt 8h = DIO7 interrupt 9h = DIO8 interrupt Ah = DIO9 interrupt Bh = DIO10 interrupt Ch = DIO11 interrupt Dh = DIO12 interrupt Eh = DIO13 interrupt Fh = DIO14 interrupt 10h = DIO15 interrupt

26 IMASK Register (Offset = 1088h) [Reset = 00000000h]

IMASK is shown in Figure 15-29 and described in Table 15-31.

Return to the Summary Table.

Interrupt Mask. If a bit is set, then corresponding interrupt is un-masked. Un-masking the interrupt causes the raw interrupt to be visible in IIDX, as well as MIS.

Figure 15-29. IMASK Register

31	30	29	28	27	26	25	24
DIO31	DIO30	DIO29	DIO28	DIO27	DIO26	DIO25	DIO24
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
DIO23	DIO22	DIO21	DIO20	DIO19	DIO18	DIO17	DIO16
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED							
R/W-0h							
7	6	5	4	3	2	1	0
RESERVED							
R/W-0h							

Table 15-57. IMASK Register Field Descriptions

Bit	Field	Type	Reset	Description
31	DIO31	R/W	0h	DIO31 event mask 0h = Event is masked 1h = Event is unmasked
30	DIO30	R/W	0h	DIO30 event mask 0h = Event is masked 1h = Event is unmasked
29	DIO29	R/W	0h	DIO29 event mask 0h = Event is masked 1h = Event is unmasked
28	DIO28	R/W	0h	DIO28 event mask 0h = Event is masked 1h = Event is unmasked
27	DIO27	R/W	0h	DIO27 event mask 0h = Event is masked 1h = Event is unmasked
26	DIO26	R/W	0h	DIO26 event mask 0h = Event is masked 1h = Event is unmasked
25	DIO25	R/W	0h	DIO25 event mask 0h = Event is masked 1h = Event is unmasked
24	DIO24	R/W	0h	DIO24 event mask 0h = Event is masked 1h = Event is unmasked
23	DIO23	R/W	0h	DIO23 event mask 0h = Event is masked 1h = Event is unmasked
22	DIO22	R/W	0h	DIO22 event mask 0h = Event is masked 1h = Event is unmasked

Table 15-57. IMASK Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
21	DIO21	R/W	0h	DIO21 event mask 0h = Event is masked 1h = Event is unmasked
20	DIO20	R/W	0h	DIO20 event mask 0h = Event is masked 1h = Event is unmasked
19	DIO19	R/W	0h	DIO19 event mask 0h = Event is masked 1h = Event is unmasked
18	DIO18	R/W	0h	DIO18 event mask 0h = Event is masked 1h = Event is unmasked
17	DIO17	R/W	0h	DIO17 event mask 0h = Event is masked 1h = Event is unmasked
16	DIO16	R/W	0h	DIO16 event mask 0h = Event is masked 1h = Event is unmasked
15-0	RESERVED	R/W	0h	

27 RIS Register (Offset = 1090h) [Reset = 00000000h]

RIS is shown in Figure 15-30 and described in Table 15-32.

Return to the Summary Table.

Raw interrupt status. Reflects all pending interrupts, regardless of masking. The RIS register allows the user to implement a poll scheme. A flag set in this register can be cleared by writing 1 to the ICLR register bit even if the corresponding IMASK bit is not enabled.

Figure 15-30. RIS Register

31	30	29	28	27	26	25	24
DIO31	DIO30	DIO29	DIO28	DIO27	DIO26	DIO25	DIO24
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
23	22	21	20	19	18	17	16
DIO23	DIO22	DIO21	DIO20	DIO19	DIO18	DIO17	DIO16
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED							
R-0h							

Table 15-59. RIS Register Field Descriptions

Bit	Field	Type	Reset	Description
31	DIO31	R	0h	DIO31 event 0h = DIO31 event did not occur 1h = DIO31 event occurred
30	DIO30	R	0h	DIO30 event 0h = DIO30 event did not occur 1h = DIO30 event occurred
29	DIO29	R	0h	DIO29 event 0h = DIO29 event did not occur 1h = DIO29 event occurred
28	DIO28	R	0h	DIO28 event 0h = DIO28 event did not occur 1h = DIO28 event occurred
27	DIO27	R	0h	DIO27 event 0h = DIO27 event did not occur 1h = DIO27 event occurred
26	DIO26	R	0h	DIO26 event 0h = DIO26 event did not occur 1h = DIO26 event occurred
25	DIO25	R	0h	DIO25 event 0h = DIO25 event did not occur 1h = DIO25 event occurred
24	DIO24	R	0h	DIO24 event 0h = DIO24 event did not occur 1h = DIO24 event occurred
23	DIO23	R	0h	DIO23 event 0h = DIO23 event did not occur 1h = DIO23 event occurred
22	DIO22	R	0h	DIO22 event 0h = DIO22 event did not occur 1h = DIO22 event occurred

Table 15-59. RIS Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
21	DIO21	R	0h	DIO21 event 0h = DIO21 event did not occur 1h = DIO21 event occurred
20	DIO20	R	0h	DIO20 event 0h = DIO20 event did not occur 1h = DIO20 event occurred
19	DIO19	R	0h	DIO19 event 0h = DIO19 event did not occur 1h = DIO19 event occurred
18	DIO18	R	0h	DIO18 event 0h = DIO18 event did not occur 1h = DIO18 event occurred
17	DIO17	R	0h	DIO17 event 0h = DIO17 event did not occur 1h = DIO17 event occurred
16	DIO16	R	0h	DIO16 event 0h = DIO16 event did not occur 1h = DIO16 event occurred
15-0	RESERVED	R	0h	

28 MIS Register (Offset = 1098h) [Reset = 00000000h]

MIS is shown in Figure 15-31 and described in Table 15-33.

Return to the Summary Table.

Masked interrupt status. This is an AND of the IMASK and RIS registers.

Figure 15-31. MIS Register

31	30	29	28	27	26	25	24
DIO31	DIO30	DIO29	DIO28	DIO27	DIO26	DIO25	DIO24
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
23	22	21	20	19	18	17	16
DIO23	DIO22	DIO21	DIO20	DIO19	DIO18	DIO17	DIO16
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED							
R-0h							

Table 15-61. MIS Register Field Descriptions

Bit	Field	Type	Reset	Description
31	DIO31	R	0h	DIO31 event 0h = DIO31 event did not occur 1h = DIO31 event occurred
30	DIO30	R	0h	DIO30 event 0h = DIO30 event did not occur 1h = DIO30 event occurred
29	DIO29	R	0h	DIO29 event 0h = DIO29 event did not occur 1h = DIO29 event occurred
28	DIO28	R	0h	DIO28 event 0h = DIO28 event did not occur 1h = DIO28 event occurred
27	DIO27	R	0h	DIO27 event 0h = DIO27 event did not occur 1h = DIO27 event occurred
26	DIO26	R	0h	DIO26 event 0h = DIO26 event did not occur 1h = DIO26 event occurred
25	DIO25	R	0h	DIO25 event 0h = DIO25 event did not occur 1h = DIO25 event occurred
24	DIO24	R	0h	DIO24 event 0h = DIO24 event did not occur 1h = DIO24 event occurred
23	DIO23	R	0h	DIO23 event 0h = DIO23 event did not occur 1h = DIO23 event occurred
22	DIO22	R	0h	DIO22 event 0h = DIO22 event did not occur 1h = DIO22 event occurred

Table 15-61. MIS Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
21	DIO21	R	0h	DIO21 event 0h = DIO21 event did not occur 1h = DIO21 event occurred
20	DIO20	R	0h	DIO20 event 0h = DIO20 event did not occur 1h = DIO20 event occurred
19	DIO19	R	0h	DIO19 event 0h = DIO19 event did not occur 1h = DIO19 event occurred
18	DIO18	R	0h	DIO18 event 0h = DIO18 event did not occur 1h = DIO18 event occurred
17	DIO17	R	0h	DIO17 event 0h = DIO17 event did not occur 1h = DIO17 event occurred
16	DIO16	R	0h	DIO16 event 0h = DIO16 event did not occur 1h = DIO16 event occurred
15-0	RESERVED	R	0h	

29 ISET Register (Offset = 10A0h) [Reset = 00000000h]

ISET is shown in Figure 15-32 and described in Table 15-34.

Return to the Summary Table.

Interrupt set. Allows interrupts to be set by software (useful in diagnostics and safety checks). Writing a 1 to a bit in ISET will set the event and therefore the related RIS bit also gets set. If the interrupt is enabled through the mask, then the corresponding MIS bit is also set.

Figure 15-32. ISET Register

31	30	29	28	27	26	25	24
DIO31	DIO30	DIO29	DIO28	DIO27	DIO26	DIO25	DIO24
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h
23	22	21	20	19	18	17	16
DIO23	DIO22	DIO21	DIO20	DIO19	DIO18	DIO17	DIO16
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h
15	14	13	12	11	10	9	8
RESERVED							
W-0h							
7	6	5	4	3	2	1	0
RESERVED							
W-0h							

Table 15-63. ISET Register Field Descriptions

Bit	Field	Type	Reset	Description
31	DIO31	W	0h	DIO31 event 0h = No effect 1h = Sets DIO31 in RIS register
30	DIO30	W	0h	DIO30 event 0h = No effect 1h = Sets DIO30 in RIS register
29	DIO29	W	0h	DIO29 event 0h = No effect 1h = Sets DIO29 in RIS register
28	DIO28	W	0h	DIO28 event 0h = No effect 1h = Sets DIO28 in RIS register
27	DIO27	W	0h	DIO27 event 0h = No effect 1h = Sets DIO27 in RIS register
26	DIO26	W	0h	DIO26 event 0h = No effect 1h = Sets DIO26 in RIS register
25	DIO25	W	0h	DIO25 event 0h = No effect 1h = Sets DIO25 in RIS register
24	DIO24	W	0h	DIO24 event 0h = No effect 1h = Sets DIO24 in RIS register
23	DIO23	W	0h	DIO23 event 0h = No effect 1h = Sets DIO23 in RIS register
22	DIO22	W	0h	DIO22 event 0h = No effect 1h = Sets DIO22 in RIS register

Table 15-63. ISET Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
21	DIO21	W	0h	DIO21 event 0h = No effect 1h = Sets DIO21 in RIS register
20	DIO20	W	0h	DIO20 event 0h = No effect 1h = Sets DIO20 in RIS register
19	DIO19	W	0h	DIO19 event 0h = No effect 1h = Sets DIO19 in RIS register
18	DIO18	W	0h	DIO18 event 0h = No effect 1h = Sets DIO18 in RIS register
17	DIO17	W	0h	DIO17 event 0h = No effect 1h = Sets DIO17 in RIS register
16	DIO16	W	0h	DIO16 event 0h = No effect 1h = Sets DIO16 in RIS register
15-0	RESERVED	W	0h	

30 ICLR Register (Offset = 10A8h) [Reset = 00000000h]

ICLR is shown in Figure 15-33 and described in Table 15-35.

Return to the Summary Table.

Interrupt clear. Write a 1 to clear corresponding Interrupt.

Figure 15-33. ICLR Register

31	30	29	28	27	26	25	24
DIO31	DIO30	DIO29	DIO28	DIO27	DIO26	DIO25	DIO24
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h
23	22	21	20	19	18	17	16
DIO23	DIO22	DIO21	DIO20	DIO19	DIO18	DIO17	DIO16
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h
15	14	13	12	11	10	9	8
RESERVED							
W-0h							
7	6	5	4	3	2	1	0
RESERVED							
W-0h							

Table 15-65. ICLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31	DIO31	W	0h	DIO31 event 0h = No effect 1h = Clears DIO31 in RIS register
30	DIO30	W	0h	DIO30 event 0h = No effect 1h = Clears DIO30 in RIS register
29	DIO29	W	0h	DIO29 event 0h = No effect 1h = Clears DIO29 in RIS register
28	DIO28	W	0h	DIO28 event 0h = No effect 1h = Clears DIO28 in RIS register
27	DIO27	W	0h	DIO27 event 0h = No effect 1h = Clears DIO27 in RIS register
26	DIO26	W	0h	DIO26 event 0h = No effect 1h = Clears DIO26 in RIS register
25	DIO25	W	0h	DIO25 event 0h = No effect 1h = Clears DIO25 in RIS register
24	DIO24	W	0h	DIO24 event 0h = No effect 1h = Clears DIO24 in RIS register
23	DIO23	W	0h	DIO23 event 0h = No effect 1h = Clears DIO23 in RIS register
22	DIO22	W	0h	DIO22 event 0h = No effect 1h = Clears DIO22 in RIS register

Table 15-65. ICLR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
21	DIO21	W	0h	DIO21 event 0h = No effect 1h = Clears DIO21 in RIS register
20	DIO20	W	0h	DIO20 event 0h = No effect 1h = Clears DIO20 in RIS register
19	DIO19	W	0h	DIO19 event 0h = No effect 1h = Clears DIO19 in RIS register
18	DIO18	W	0h	DIO18 event 0h = No effect 1h = Clears DIO18 in RIS register
17	DIO17	W	0h	DIO17 event 0h = No effect 1h = Clears DIO17 in RIS register
16	DIO16	W	0h	DIO16 event 0h = No effect 1h = Clears DIO16 in RIS register
15-0	RESERVED	W	0h	

31 EVT_MODE Register (Offset = 10E0h) [Reset = 00000000h]

EVT_MODE is shown in Figure 15-34 and described in Table 15-36.

Return to the Summary Table.

Event mode register. It is used to select whether each line is disabled, in software mode (software clears the RIS) or in hardware mode (hardware clears the RIS)

Figure 15-34. EVT_MODE Register

31	30	29	28	27	26	25	24
RESERVED							
R/W-0h							
23	22	21	20	19	18	17	16
RESERVED							
R/W-0h							
15	14	13	12	11	10	9	8
RESERVED							
R/W-0h							
7	6	5	4	3	2	1	0
RESERVED		EVT2_CFG		EVT1_CFG		INT0_CFG	
R/W-0h		R-0h		R-0h		R-0h	

Table 15-67. EVT_MODE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-6	RESERVED	R/W	0h	
5-4	EVT2_CFG	R	2h	Event line mode select for event corresponding to [IPSTANDARD.GEN_EVENT1] 0h = The interrupt or event line is disabled. 1h = The interrupt or event line is in software mode. Software must clear the RIS. 2h = The interrupt or event line is in hardware mode. The hardware (another module) clears automatically the associated RIS flag.
3-2	EVT1_CFG	R	2h	Event line mode select for event corresponding to [IPSTANDARD.GEN_EVENT0] 0h = The interrupt or event line is disabled. 1h = The interrupt or event line is in software mode. Software must clear the RIS. 2h = The interrupt or event line is in hardware mode. The hardware (another module) clears automatically the associated RIS flag.
1-0	INT0_CFG	R	1h	Event line mode select for event corresponding to [IPSTANDARD.CPU_INT] 0h = The interrupt or event line is disabled. 1h = The interrupt or event line is in software mode. Software must clear the RIS. 2h = The interrupt or event line is in hardware mode. The hardware (another module) clears automatically the associated RIS flag.

32 DESC Register (Offset = 10FCh) [Reset = 00000000h]

DESC is shown in Figure 15-35 and described in Table 15-37.

Return to the Summary Table.

This register identifies the peripheral and its exact version.

Figure 15-35. DESC Register

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

Table 15-69. DESC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	MODULEID	R	1611h	Module identification contains a unique peripheral identification number. The assignments are maintained in a central database for all of the platform modules to ensure uniqueness. 0h = Smallest value FFFFh = Highest possible value
15-12	FEATUREVER	R	0h	Feature Set for the module *instance* 0h = Smallest value Fh = Highest possible value
11-8	RESERVED	R	0h	
7-4	MAJREV	R	0h	Major rev of the IP 0h = Smallest value Fh = Highest possible value
3-0	MINREV	R	0h	Minor rev of the IP 0h = Smallest value Fh = Highest possible value

33 DOUT3_0 Register (Offset = 1200h) [Reset = 00000000h]

DOUT3_0 is shown in Figure 15-36 and described in Table 15-38.

Return to the Summary Table.

Data output for pins configured as DIO3 to DIO0. This is an alias register for byte access to bits 3 to 0 in DOUT31_0 register.

Figure 15-36. DOUT3_0 Register

31	30	29	28	27	26	25	24
RESERVED							DIO3
W-0h							W-0h
23	22	21	20	19	18	17	16
RESERVED							DIO2
W-0h							W-0h
15	14	13	12	11	10	9	8
RESERVED							DIO1
W-0h							W-0h
7	6	5	4	3	2	1	0
RESERVED							DIO0
W-0h							W-0h

Table 15-71. DOUT3_0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-25	RESERVED	W	0h	
24	DIO3	W	0h	This bit sets the value of the pin configured as DIO3 when the output is enabled through DOE31_0 register. 0h = Output is set to 0 1h = Output is set to 1
23-17	RESERVED	W	0h	
16	DIO2	W	0h	This bit sets the value of the pin configured as DIO2 when the output is enabled through DOE31_0 register. 0h = Output is set to 0 1h = Output is set to 1
15-9	RESERVED	W	0h	
8	DIO1	W	0h	This bit sets the value of the pin configured as DIO1 when the output is enabled through DOE31_0 register. 0h = Output is set to 0 1h = Output is set to 1
7-1	RESERVED	W	0h	
0	DIO0	W	0h	This bit sets the value of the pin configured as DIO0 when the output is enabled through DOE31_0 register. 0h = Output is set to 0 1h = Output is set to 1

34 DOUT7_4 Register (Offset = 1204h) [Reset = 00000000h]

DOUT7_4 is shown in Figure 15-37 and described in Table 15-39.

Return to the Summary Table.

Data output for pins configured as DIO7 to DIO4. This is an alias register for byte access to bits 7 to 4 in DOUT31_0 register.

Figure 15-37. DOUT7_4 Register

31	30	29	28	27	26	25	24
RESERVED							DIO7
W-0h							W-0h
23	22	21	20	19	18	17	16
RESERVED							DIO6
W-0h							W-0h
15	14	13	12	11	10	9	8
RESERVED							DIO5
W-0h							W-0h
7	6	5	4	3	2	1	0
RESERVED							DIO4
W-0h							W-0h

Table 15-73. DOUT7_4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-25	RESERVED	W	0h	
24	DIO7	W	0h	This bit sets the value of the pin configured as DIO7 when the output is enabled through DOE31_0 register. 0h = Output is set to 0 1h = Output is set to 1
23-17	RESERVED	W	0h	
16	DIO6	W	0h	This bit sets the value of the pin configured as DIO6 when the output is enabled through DOE31_0 register. 0h = Output is set to 0 1h = Output is set to 1
15-9	RESERVED	W	0h	
8	DIO5	W	0h	This bit sets the value of the pin configured as DIO5 when the output is enabled through DOE31_0 register. 0h = Output is set to 0 1h = Output is set to 1
7-1	RESERVED	W	0h	
0	DIO4	W	0h	This bit sets the value of the pin configured as DIO4 when the output is enabled through DOE31_0 register. 0h = Output is set to 0 1h = Output is set to 1

35 DOUT11_8 Register (Offset = 1208h) [Reset = 00000000h]

DOUT11_8 is shown in Figure 15-38 and described in Table 15-40.

Return to the Summary Table.

Data output for pins configured as DIO11 to DIO8. This is an alias register for byte access to bits 11 to 8 in DOUT31_0 register.

Figure 15-38. DOUT11_8 Register

31	30	29	28	27	26	25	24
RESERVED							DIO11
W-0h							W-0h
23	22	21	20	19	18	17	16
RESERVED							DIO10
W-0h							W-0h
15	14	13	12	11	10	9	8
RESERVED							DIO9
W-0h							W-0h
7	6	5	4	3	2	1	0
RESERVED							DIO8
W-0h							W-0h

Table 15-75. DOUT11_8 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-25	RESERVED	W	0h	
24	DIO11	W	0h	This bit sets the value of the pin configured as DIO11 when the output is enabled through DOE31_0 register. 0h = Output is set to 0 1h = Output is set to 1
23-17	RESERVED	W	0h	
16	DIO10	W	0h	This bit sets the value of the pin configured as DIO10 when the output is enabled through DOE31_0 register. 0h = Output is set to 0 1h = Output is set to 1
15-9	RESERVED	W	0h	
8	DIO9	W	0h	This bit sets the value of the pin configured as DIO9 when the output is enabled through DOE31_0 register. 0h = Output is set to 0 1h = Output is set to 1
7-1	RESERVED	W	0h	
0	DIO8	W	0h	This bit sets the value of the pin configured as DIO8 when the output is enabled through DOE31_0 register. 0h = Output is set to 0 1h = Output is set to 1

36 DOUT15_12 Register (Offset = 120Ch) [Reset = 00000000h]

DOUT15_12 is shown in Figure 15-39 and described in Table 15-41.

Return to the Summary Table.

Data output for pins configured as DIO15 to DIO12. This is an alias register for byte access to bits 15 to 12 in DOUT31_0 register.

Figure 15-39. DOUT15_12 Register

31	30	29	28	27	26	25	24
RESERVED							DIO15
W-0h							W-0h
23	22	21	20	19	18	17	16
RESERVED							DIO14
W-0h							W-0h
15	14	13	12	11	10	9	8
RESERVED							DIO13
W-0h							W-0h
7	6	5	4	3	2	1	0
RESERVED							DIO12
W-0h							W-0h

Table 15-77. DOUT15_12 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-25	RESERVED	W	0h	
24	DIO15	W	0h	This bit sets the value of the pin configured as DIO15 when the output is enabled through DOE31_0 register. 0h = Output is set to 0 1h = Output is set to 1
23-17	RESERVED	W	0h	
16	DIO14	W	0h	This bit sets the value of the pin configured as DIO14 when the output is enabled through DOE31_0 register. 0h = Output is set to 0 1h = Output is set to 1
15-9	RESERVED	W	0h	
8	DIO13	W	0h	This bit sets the value of the pin configured as DIO13 when the output is enabled through DOE31_0 register. 0h = Output is set to 0 1h = Output is set to 1
7-1	RESERVED	W	0h	
0	DIO12	W	0h	This bit sets the value of the pin configured as DIO12 when the output is enabled through DOE31_0 register. 0h = Output is set to 0 1h = Output is set to 1

37 DOUT19_16 Register (Offset = 1210h) [Reset = 00000000h]

DOUT19_16 is shown in Figure 15-40 and described in Table 15-42.

Return to the Summary Table.

Data output for pins configured as DIO19 to DIO16. This is an alias register for byte access to bits 19 to 16 in DOUT31_0 register.

Figure 15-40. DOUT19_16 Register

31	30	29	28	27	26	25	24
RESERVED							DIO19
W-0h							W-0h
23	22	21	20	19	18	17	16
RESERVED							DIO18
W-0h							W-0h
15	14	13	12	11	10	9	8
RESERVED							DIO17
W-0h							W-0h
7	6	5	4	3	2	1	0
RESERVED							DIO16
W-0h							W-0h

Table 15-79. DOUT19_16 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-25	RESERVED	W	0h	
24	DIO19	W	0h	This bit sets the value of the pin configured as DIO19 when the output is enabled through DOE31_0 register. 0h = Output is set to 0 1h = Output is set to 1
23-17	RESERVED	W	0h	
16	DIO18	W	0h	This bit sets the value of the pin configured as DIO18 when the output is enabled through DOE31_0 register. 0h = Output is set to 0 1h = Output is set to 1
15-9	RESERVED	W	0h	
8	DIO17	W	0h	This bit sets the value of the pin configured as DIO17 when the output is enabled through DOE31_0 register. 0h = Output is set to 0 1h = Output is set to 1
7-1	RESERVED	W	0h	
0	DIO16	W	0h	This bit sets the value of the pin configured as DIO16 when the output is enabled through DOE31_0 register. 0h = Output is set to 0 1h = Output is set to 1

38 DOUT23_20 Register (Offset = 1214h) [Reset = 00000000h]

DOUT23_20 is shown in Figure 15-41 and described in Table 15-43.

Return to the Summary Table.

Data output for pins configured as DIO23 to DIO20. This is an alias register for byte access to bits 23 to 20 in DOUT31_0 register.

Figure 15-41. DOUT23_20 Register

31	30	29	28	27	26	25	24
RESERVED							DIO23
W-0h							W-0h
23	22	21	20	19	18	17	16
RESERVED							DIO22
W-0h							W-0h
15	14	13	12	11	10	9	8
RESERVED							DIO21
W-0h							W-0h
7	6	5	4	3	2	1	0
RESERVED							DIO20
W-0h							W-0h

Table 15-81. DOUT23_20 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-25	RESERVED	W	0h	
24	DIO23	W	0h	This bit sets the value of the pin configured as DIO23 when the output is enabled through DOE31_0 register. 0h = Output is set to 0 1h = Output is set to 1
23-17	RESERVED	W	0h	
16	DIO22	W	0h	This bit sets the value of the pin configured as DIO22 when the output is enabled through DOE31_0 register. 0h = Output is set to 0 1h = Output is set to 1
15-9	RESERVED	W	0h	
8	DIO21	W	0h	This bit sets the value of the pin configured as DIO21 when the output is enabled through DOE31_0 register. 0h = Output is set to 0 1h = Output is set to 1
7-1	RESERVED	W	0h	
0	DIO20	W	0h	This bit sets the value of the pin configured as DIO20 when the output is enabled through DOE31_0 register. 0h = Output is set to 0 1h = Output is set to 1

39 DOUT27_24 Register (Offset = 1218h) [Reset = 00000000h]

DOUT27_24 is shown in Figure 15-42 and described in Table 15-44.

Return to the Summary Table.

Data output for pins configured as DIO27 to DIO24. This is an alias register for byte access to bits 27 to 24 in DOUT31_0 register.

Figure 15-42. DOUT27_24 Register

31	30	29	28	27	26	25	24
RESERVED							DIO27
W-0h							W-0h
23	22	21	20	19	18	17	16
RESERVED							DIO26
W-0h							W-0h
15	14	13	12	11	10	9	8
RESERVED							DIO25
W-0h							W-0h
7	6	5	4	3	2	1	0
RESERVED							DIO24
W-0h							W-0h

Table 15-83. DOUT27_24 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-25	RESERVED	W	0h	
24	DIO27	W	0h	This bit sets the value of the pin configured as DIO27 when the output is enabled through DOE31_0 register. 0h = Output is set to 0 1h = Output is set to 1
23-17	RESERVED	W	0h	
16	DIO26	W	0h	This bit sets the value of the pin configured as DIO26 when the output is enabled through DOE31_0 register. 0h = Output is set to 0 1h = Output is set to 1
15-9	RESERVED	W	0h	
8	DIO25	W	0h	This bit sets the value of the pin configured as DIO25 when the output is enabled through DOE31_0 register. 0h = Output is set to 0 1h = Output is set to 1
7-1	RESERVED	W	0h	
0	DIO24	W	0h	This bit sets the value of the pin configured as DIO24 when the output is enabled through DOE31_0 register. 0h = Output is set to 0 1h = Output is set to 1

40 DOUT31_28 Register (Offset = 121Ch) [Reset = 00000000h]

DOUT31_28 is shown in Figure 15-43 and described in Table 15-45.

Return to the Summary Table.

Data output for pins configured as DIO31 to DIO28. This is an alias register for byte access to bits 31 to 28 in DOUT31_0 register.

Figure 15-43. DOUT31_28 Register

31	30	29	28	27	26	25	24
RESERVED							DIO31
W-0h							W-0h
23	22	21	20	19	18	17	16
RESERVED							DIO30
W-0h							W-0h
15	14	13	12	11	10	9	8
RESERVED							DIO29
W-0h							W-0h
7	6	5	4	3	2	1	0
RESERVED							DIO28
W-0h							W-0h

Table 15-85. DOUT31_28 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-25	RESERVED	W	0h	
24	DIO31	W	0h	This bit sets the value of the pin configured as DIO31 when the output is enabled through DOE31_0 register. 0h = Output is set to 0 1h = Output is set to 1
23-17	RESERVED	W	0h	
16	DIO30	W	0h	This bit sets the value of the pin configured as DIO30 when the output is enabled through DOE31_0 register. 0h = Output is set to 0 1h = Output is set to 1
15-9	RESERVED	W	0h	
8	DIO29	W	0h	This bit sets the value of the pin configured as DIO29 when the output is enabled through DOE31_0 register. 0h = Output is set to 0 1h = Output is set to 1
7-1	RESERVED	W	0h	
0	DIO28	W	0h	This bit sets the value of the pin configured as DIO28 when the output is enabled through DOE31_0 register. 0h = Output is set to 0 1h = Output is set to 1

41 DOUT31_0 Register (Offset = 1280h) [Reset = 00000000h]

DOUT31_0 is shown in Figure 15-44 and described in Table 15-46.

Return to the Summary Table.

Data output for pins configured as DIO31 to DIO0.

Figure 15-44. DOUT31_0 Register

31	30	29	28	27	26	25	24
DIO31	DIO30	DIO29	DIO28	DIO27	DIO26	DIO25	DIO24
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
DIO23	DIO22	DIO21	DIO20	DIO19	DIO18	DIO17	DIO16
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
DIO15	DIO14	DIO13	DIO12	DIO11	DIO10	DIO9	DIO8
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
DIO7	DIO6	DIO5	DIO4	DIO3	DIO2	DIO1	DIO0
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 15-87. DOUT31_0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	DIO31	R/W	0h	This bit sets the value of the pin configured as DIO31 when the output is enabled through DOE31_0 register. 0h = Output is set to 0 1h = Output is set to 1
30	DIO30	R/W	0h	This bit sets the value of the pin configured as DIO30 when the output is enabled through DOE31_0 register. 0h = Output is set to 0 1h = Output is set to 1
29	DIO29	R/W	0h	This bit sets the value of the pin configured as DIO29 when the output is enabled through DOE31_0 register. 0h = Output is set to 0 1h = Output is set to 1
28	DIO28	R/W	0h	This bit sets the value of the pin configured as DIO28 when the output is enabled through DOE31_0 register. 0h = Output is set to 0 1h = Output is set to 1
27	DIO27	R/W	0h	This bit sets the value of the pin configured as DIO27 when the output is enabled through DOE31_0 register. 0h = Output is set to 0 1h = Output is set to 1
26	DIO26	R/W	0h	This bit sets the value of the pin configured as DIO26 when the output is enabled through DOE31_0 register. 0h = Output is set to 0 1h = Output is set to 1
25	DIO25	R/W	0h	This bit sets the value of the pin configured as DIO25 when the output is enabled through DOE31_0 register. 0h = Output is set to 0 1h = Output is set to 1
24	DIO24	R/W	0h	This bit sets the value of the pin configured as DIO24 when the output is enabled through DOE31_0 register. 0h = Output is set to 0 1h = Output is set to 1

Table 15-87. DOUT31_0 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
23	DIO23	R/W	0h	This bit sets the value of the pin configured as DIO23 when the output is enabled through DOE31_0 register. 0h = Output is set to 0 1h = Output is set to 1
22	DIO22	R/W	0h	This bit sets the value of the pin configured as DIO22 when the output is enabled through DOE31_0 register. 0h = Output is set to 0 1h = Output is set to 1
21	DIO21	R/W	0h	This bit sets the value of the pin configured as DIO21 when the output is enabled through DOE31_0 register. 0h = Output is set to 0 1h = Output is set to 1
20	DIO20	R/W	0h	This bit sets the value of the pin configured as DIO20 when the output is enabled through DOE31_0 register. 0h = Output is set to 0 1h = Output is set to 1
19	DIO19	R/W	0h	This bit sets the value of the pin configured as DIO19 when the output is enabled through DOE31_0 register. 0h = Output is set to 0 1h = Output is set to 1
18	DIO18	R/W	0h	This bit sets the value of the pin configured as DIO18 when the output is enabled through DOE31_0 register. 0h = Output is set to 0 1h = Output is set to 1
17	DIO17	R/W	0h	This bit sets the value of the pin configured as DIO17 when the output is enabled through DOE31_0 register. 0h = Output is set to 0 1h = Output is set to 1
16	DIO16	R/W	0h	This bit sets the value of the pin configured as DIO16 when the output is enabled through DOE31_0 register. 0h = Output is set to 0 1h = Output is set to 1
15	DIO15	R/W	0h	This bit sets the value of the pin configured as DIO15 when the output is enabled through DOE31_0 register. 0h = Output is set to 0 1h = Output is set to 1
14	DIO14	R/W	0h	This bit sets the value of the pin configured as DIO14 when the output is enabled through DOE31_0 register. 0h = Output is set to 0 1h = Output is set to 1
13	DIO13	R/W	0h	This bit sets the value of the pin configured as DIO13 when the output is enabled through DOE31_0 register. 0h = Output is set to 0 1h = Output is set to 1
12	DIO12	R/W	0h	This bit sets the value of the pin configured as DIO12 when the output is enabled through DOE31_0 register. 0h = Output is set to 0 1h = Output is set to 1
11	DIO11	R/W	0h	This bit sets the value of the pin configured as DIO11 when the output is enabled through DOE31_0 register. 0h = Output is set to 0 1h = Output is set to 1
10	DIO10	R/W	0h	This bit sets the value of the pin configured as DIO10 when the output is enabled through DOE31_0 register. 0h = Output is set to 0 1h = Output is set to 1

Table 15-87. DOUT31_0 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
9	DIO9	R/W	0h	This bit sets the value of the pin configured as DIO9 when the output is enabled through DOE31_0 register. 0h = Output is set to 0 1h = Output is set to 1
8	DIO8	R/W	0h	This bit sets the value of the pin configured as DIO8 when the output is enabled through DOE31_0 register. 0h = Output is set to 0 1h = Output is set to 1
7	DIO7	R/W	0h	This bit sets the value of the pin configured as DIO7 when the output is enabled through DOE31_0 register. 0h = Output is set to 0 1h = Output is set to 1
6	DIO6	R/W	0h	This bit sets the value of the pin configured as DIO6 when the output is enabled through DOE31_0 register. 0h = Output is set to 0 1h = Output is set to 1
5	DIO5	R/W	0h	This bit sets the value of the pin configured as DIO5 when the output is enabled through DOE31_0 register. 0h = Output is set to 0 1h = Output is set to 1
4	DIO4	R/W	0h	This bit sets the value of the pin configured as DIO4 when the output is enabled through DOE31_0 register. 0h = Output is set to 0 1h = Output is set to 1
3	DIO3	R/W	0h	This bit sets the value of the pin configured as DIO3 when the output is enabled through DOE31_0 register. 0h = Output is set to 0 1h = Output is set to 1
2	DIO2	R/W	0h	This bit sets the value of the pin configured as DIO2 when the output is enabled through DOE31_0 register. 0h = Output is set to 0 1h = Output is set to 1
1	DIO1	R/W	0h	This bit sets the value of the pin configured as DIO1 when the output is enabled through DOE31_0 register. 0h = Output is set to 0 1h = Output is set to 1
0	DIO0	R/W	0h	This bit sets the value of the pin configured as DIO0 when the output is enabled through DOE31_0 register. 0h = Output is set to 0 1h = Output is set to 1

42 DOUTSET31_0 Register (Offset = 1290h) [Reset = 00000000h]

DOUTSET31_0 is shown in Figure 15-45 and described in Table 15-47.

Return to the Summary Table.

Writing 1 to a bit position in this register sets the corresponding bit in the DOUT31_0 register.

Figure 15-45. DOUTSET31_0 Register

31	30	29	28	27	26	25	24
DIO31	DIO30	DIO29	DIO28	DIO27	DIO26	DIO25	DIO24
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h
23	22	21	20	19	18	17	16
DIO23	DIO22	DIO21	DIO20	DIO19	DIO18	DIO17	DIO16
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h
15	14	13	12	11	10	9	8
DIO15	DIO14	DIO13	DIO12	DIO11	DIO10	DIO9	DIO8
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h
7	6	5	4	3	2	1	0
DIO7	DIO6	DIO5	DIO4	DIO3	DIO2	DIO1	DIO0
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h

Table 15-89. DOUTSET31_0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	DIO31	W	0h	Writing 1 to this bit sets the DIO31 bit in the DOUT31_0 register. Writing 0 has no effect. 0h = No effect 1h = Sets DIO31 in DOUT31_0
30	DIO30	W	0h	Writing 1 to this bit sets the DIO30 bit in the DOUT31_0 register. Writing 0 has no effect. 0h = No effect 1h = Sets DIO30 in DOUT31_0
29	DIO29	W	0h	Writing 1 to this bit sets the DIO29 bit in the DOUT31_0 register. Writing 0 has no effect. 0h = No effect 1h = Sets DIO29 in DOUT31_0
28	DIO28	W	0h	Writing 1 to this bit sets the DIO28 bit in the DOUT31_0 register. Writing 0 has no effect. 0h = No effect 1h = Sets DIO28 in DOUT31_0
27	DIO27	W	0h	Writing 1 to this bit sets the DIO27 bit in the DOUT31_0 register. Writing 0 has no effect. 0h = No effect 1h = Sets DIO27 in DOUT31_0
26	DIO26	W	0h	Writing 1 to this bit sets the DIO26 bit in the DOUT31_0 register. Writing 0 has no effect. 0h = No effect 1h = Sets DIO26 in DOUT31_0
25	DIO25	W	0h	Writing 1 to this bit sets the DIO25 bit in the DOUT31_0 register. Writing 0 has no effect. 0h = No effect 1h = Sets DIO25 in DOUT31_0
24	DIO24	W	0h	Writing 1 to this bit sets the DIO24 bit in the DOUT31_0 register. Writing 0 has no effect. 0h = No effect 1h = Sets DIO24 in DOUT31_0

Table 15-89. DOUTSET31_0 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
23	DIO23	W	0h	Writing 1 to this bit sets the DIO23 bit in the DOUT31_0 register. Writing 0 has no effect. 0h = No effect 1h = Sets DIO23 in DOUT31_0
22	DIO22	W	0h	Writing 1 to this bit sets the DIO22 bit in the DOUT31_0 register. Writing 0 has no effect. 0h = No effect 1h = Sets DIO22 in DOUT31_0
21	DIO21	W	0h	Writing 1 to this bit sets the DIO21 bit in the DOUT31_0 register. Writing 0 has no effect. 0h = No effect 1h = Sets DIO21 in DOUT31_0
20	DIO20	W	0h	Writing 1 to this bit sets the DIO20 bit in the DOUT31_0 register. Writing 0 has no effect. 0h = No effect 1h = Sets DIO20 in DOUT31_0
19	DIO19	W	0h	Writing 1 to this bit sets the DIO19 bit in the DOUT31_0 register. Writing 0 has no effect. 0h = No effect 1h = Sets DIO19 in DOUT31_0
18	DIO18	W	0h	Writing 1 to this bit sets the DIO18 bit in the DOUT31_0 register. Writing 0 has no effect. 0h = No effect 1h = Sets DIO18 in DOUT31_0
17	DIO17	W	0h	Writing 1 to this bit sets the DIO17 bit in the DOUT31_0 register. Writing 0 has no effect. 0h = No effect 1h = Sets DIO17 in DOUT31_0
16	DIO16	W	0h	Writing 1 to this bit sets the DIO16 bit in the DOUT31_0 register. Writing 0 has no effect. 0h = No effect 1h = Sets DIO16 in DOUT31_0
15	DIO15	W	0h	Writing 1 to this bit sets the DIO15 bit in the DOUT31_0 register. Writing 0 has no effect. 0h = No effect 1h = Sets DIO15 in DOUT31_0
14	DIO14	W	0h	Writing 1 to this bit sets the DIO14 bit in the DOUT31_0 register. Writing 0 has no effect. 0h = No effect 1h = Sets DIO14 in DOUT31_0
13	DIO13	W	0h	Writing 1 to this bit sets the DIO13 bit in the DOUT31_0 register. Writing 0 has no effect. 0h = No effect 1h = Sets DIO13 in DOUT31_0
12	DIO12	W	0h	Writing 1 to this bit sets the DIO12 bit in the DOUT31_0 register. Writing 0 has no effect. 0h = No effect 1h = Sets DIO12 in DOUT31_0
11	DIO11	W	0h	Writing 1 to this bit sets the DIO11 bit in the DOUT31_0 register. Writing 0 has no effect. 0h = No effect 1h = Sets DIO11 in DOUT31_0
10	DIO10	W	0h	Writing 1 to this bit sets the DIO10 bit in the DOUT31_0 register. Writing 0 has no effect. 0h = No effect 1h = Sets DIO10 in DOUT31_0

Table 15-89. DOUTSET31_0 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
9	DIO9	W	0h	Writing 1 to this bit sets the DIO9 bit in the DOUT31_0 register. Writing 0 has no effect. 0h = No effect 1h = Sets DIO9 in DOUT31_0
8	DIO8	W	0h	Writing 1 to this bit sets the DIO8 bit in the DOUT31_0 register. Writing 0 has no effect. 0h = No effect 1h = Sets DIO8 in DOUT31_0
7	DIO7	W	0h	Writing 1 to this bit sets the DIO7 bit in the DOUT31_0 register. Writing 0 has no effect. 0h = No effect 1h = Sets DIO7 in DOUT31_0
6	DIO6	W	0h	Writing 1 to this bit sets the DIO6 bit in the DOUT31_0 register. Writing 0 has no effect. 0h = No effect 1h = Sets DIO6 in DOUT31_0
5	DIO5	W	0h	Writing 1 to this bit sets the DIO5 bit in the DOUT31_0 register. Writing 0 has no effect. 0h = No effect 1h = Sets DIO5 in DOUT31_0
4	DIO4	W	0h	Writing 1 to this bit sets the DIO4 bit in the DOUT31_0 register. Writing 0 has no effect. 0h = No effect 1h = Sets DIO4 in DOUT31_0
3	DIO3	W	0h	Writing 1 to this bit sets the DIO3 bit in the DOUT31_0 register. Writing 0 has no effect. 0h = No effect 1h = Sets DIO3 in DOUT31_0
2	DIO2	W	0h	Writing 1 to this bit sets the DIO2 bit in the DOUT31_0 register. Writing 0 has no effect. 0h = No effect 1h = Sets DIO2 in DOUT31_0
1	DIO1	W	0h	Writing 1 to this bit sets the DIO1 bit in the DOUT31_0 register. Writing 0 has no effect. 0h = No effect 1h = Sets DIO1 in DOUT31_0
0	DIO0	W	0h	Writing 1 to this bit sets the DIO0 bit in the DOUT31_0 register. Writing 0 has no effect. 0h = No effect 1h = Sets DIO0 in DOUT31_0

43 DOUTCLR31_0 Register (Offset = 12A0h) [Reset = 00000000h]

DOUTCLR31_0 is shown in Figure 15-46 and described in Table 15-48.

Return to the Summary Table.

Writing 1 to a bit position in this register clears the corresponding bit in the DOUT31_0 register.

Figure 15-46. DOUTCLR31_0 Register

31	30	29	28	27	26	25	24
DIO31	DIO30	DIO29	DIO28	DIO27	DIO26	DIO25	DIO24
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h
23	22	21	20	19	18	17	16
DIO23	DIO22	DIO21	DIO20	DIO19	DIO18	DIO17	DIO16
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h
15	14	13	12	11	10	9	8
DIO15	DIO14	DIO13	DIO12	DIO11	DIO10	DIO9	DIO8
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h
7	6	5	4	3	2	1	0
DIO7	DIO6	DIO5	DIO4	DIO3	DIO2	DIO1	DIO0
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h

Table 15-91. DOUTCLR31_0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	DIO31	W	0h	Writing 1 to this bit clears the DIO31 bit in the DOUT31_0 register. Writing 0 has no effect. 0h = No effect 1h = Clears DIO31 in DOUT31_0
30	DIO30	W	0h	Writing 1 to this bit clears the DIO30 bit in the DOUT31_0 register. Writing 0 has no effect. 0h = No effect 1h = Clears DIO30 in DOUT31_0
29	DIO29	W	0h	Writing 1 to this bit clears the DIO29 bit in the DOUT31_0 register. Writing 0 has no effect. 0h = No effect 1h = Clears DIO29 in DOUT31_0
28	DIO28	W	0h	Writing 1 to this bit clears the DIO28 bit in the DOUT31_0 register. Writing 0 has no effect. 0h = No effect 1h = Clears DIO28 in DOUT31_0
27	DIO27	W	0h	Writing 1 to this bit clears the DIO27 bit in the DOUT31_0 register. Writing 0 has no effect. 0h = No effect 1h = Clears DIO27 in DOUT31_0
26	DIO26	W	0h	Writing 1 to this bit clears the DIO26 bit in the DOUT31_0 register. Writing 0 has no effect. 0h = No effect 1h = Clears DIO26 in DOUT31_0
25	DIO25	W	0h	Writing 1 to this bit clears the DIO25 bit in the DOUT31_0 register. Writing 0 has no effect. 0h = No effect 1h = Clears DIO25 in DOUT31_0
24	DIO24	W	0h	Writing 1 to this bit clears the DIO24 bit in the DOUT31_0 register. Writing 0 has no effect. 0h = No effect 1h = Clears DIO24 in DOUT31_0

Table 15-91. DOUTCLR31_0 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
23	DIO23	W	0h	Writing 1 to this bit clears the DIO23 bit in the DOUT31_0 register. Writing 0 has no effect. 0h = No effect 1h = Clears DIO23 in DOUT31_0
22	DIO22	W	0h	Writing 1 to this bit clears the DIO22 bit in the DOUT31_0 register. Writing 0 has no effect. 0h = No effect 1h = Clears DIO22 in DOUT31_0
21	DIO21	W	0h	Writing 1 to this bit clears the DIO21 bit in the DOUT31_0 register. Writing 0 has no effect. 0h = No effect 1h = Clears DIO21 in DOUT31_0
20	DIO20	W	0h	Writing 1 to this bit clears the DIO20 bit in the DOUT31_0 register. Writing 0 has no effect. 0h = No effect 1h = Clears DIO20 in DOUT31_0
19	DIO19	W	0h	Writing 1 to this bit clears the DIO19 bit in the DOUT31_0 register. Writing 0 has no effect. 0h = No effect 1h = Clears DIO19 in DOUT31_0
18	DIO18	W	0h	Writing 1 to this bit clears the DIO18 bit in the DOUT31_0 register. Writing 0 has no effect. 0h = No effect 1h = Clears DIO18 in DOUT31_0
17	DIO17	W	0h	Writing 1 to this bit clears the DIO17 bit in the DOUT31_0 register. Writing 0 has no effect. 0h = No effect 1h = Clears DIO17 in DOUT31_0
16	DIO16	W	0h	Writing 1 to this bit clears the DIO16 bit in the DOUT31_0 register. Writing 0 has no effect. 0h = No effect 1h = Clears DIO16 in DOUT31_0
15	DIO15	W	0h	Writing 1 to this bit clears the DIO15 bit in the DOUT31_0 register. Writing 0 has no effect. 0h = No effect 1h = Clears DIO15 in DOUT31_0
14	DIO14	W	0h	Writing 1 to this bit clears the DIO14 bit in the DOUT31_0 register. Writing 0 has no effect. 0h = No effect 1h = Clears DIO14 in DOUT31_0
13	DIO13	W	0h	Writing 1 to this bit clears the DIO13 bit in the DOUT31_0 register. Writing 0 has no effect. 0h = No effect 1h = Clears DIO13 in DOUT31_0
12	DIO12	W	0h	Writing 1 to this bit clears the DIO12 bit in the DOUT31_0 register. Writing 0 has no effect. 0h = No effect 1h = Clears DIO12 in DOUT31_0
11	DIO11	W	0h	Writing 1 to this bit clears the DIO11 bit in the DOUT31_0 register. Writing 0 has no effect. 0h = No effect 1h = Clears DIO11 in DOUT31_0
10	DIO10	W	0h	Writing 1 to this bit clears the DIO10 bit in the DOUT31_0 register. Writing 0 has no effect. 0h = No effect 1h = Clears DIO10 in DOUT31_0

Table 15-91. DOUTCLR31_0 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
9	DIO9	W	0h	Writing 1 to this bit clears the DIO9 bit in the DOUT31_0 register. Writing 0 has no effect. 0h = No effect 1h = Clears DIO9 in DOUT31_0
8	DIO8	W	0h	Writing 1 to this bit clears the DIO8 bit in the DOUT31_0 register. Writing 0 has no effect. 0h = No effect 1h = Clears DIO8 in DOUT31_0
7	DIO7	W	0h	Writing 1 to this bit clears the DIO7 bit in the DOUT31_0 register. Writing 0 has no effect. 0h = No effect 1h = Clears DIO7 in DOUT31_0
6	DIO6	W	0h	Writing 1 to this bit clears the DIO6 bit in the DOUT31_0 register. Writing 0 has no effect. 0h = No effect 1h = Clears DIO6 in DOUT31_0
5	DIO5	W	0h	Writing 1 to this bit clears the DIO5 bit in the DOUT31_0 register. Writing 0 has no effect. 0h = No effect 1h = Clears DIO5 in DOUT31_0
4	DIO4	W	0h	Writing 1 to this bit clears the DIO4 bit in the DOUT31_0 register. Writing 0 has no effect. 0h = No effect 1h = Clears DIO4 in DOUT31_0
3	DIO3	W	0h	Writing 1 to this bit clears the DIO3 bit in the DOUT31_0 register. Writing 0 has no effect. 0h = No effect 1h = Clears DIO3 in DOUT31_0
2	DIO2	W	0h	Writing 1 to this bit clears the DIO2 bit in the DOUT31_0 register. Writing 0 has no effect. 0h = No effect 1h = Clears DIO2 in DOUT31_0
1	DIO1	W	0h	Writing 1 to this bit clears the DIO1 bit in the DOUT31_0 register. Writing 0 has no effect. 0h = No effect 1h = Clears DIO1 in DOUT31_0
0	DIO0	W	0h	Writing 1 to this bit clears the DIO0 bit in the DOUT31_0 register. Writing 0 has no effect. 0h = No effect 1h = Clears DIO0 in DOUT31_0

44 DOUTTGL31_0 Register (Offset = 12B0h) [Reset = 00000000h]

DOUTTGL31_0 is shown in Figure 15-47 and described in Table 15-49.

Return to the Summary Table.

Writing 1 to a bit position in this register will invert the corresponding DIO output.

Figure 15-47. DOUTTGL31_0 Register

31	30	29	28	27	26	25	24
DIO31	DIO30	DIO29	DIO28	DIO27	DIO26	DIO25	DIO24
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h
23	22	21	20	19	18	17	16
DIO23	DIO22	DIO21	DIO20	DIO19	DIO18	DIO17	DIO16
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h
15	14	13	12	11	10	9	8
DIO15	DIO14	DIO13	DIO12	DIO11	DIO10	DIO9	DIO8
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h
7	6	5	4	3	2	1	0
DIO7	DIO6	DIO5	DIO4	DIO3	DIO2	DIO1	DIO0
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h

Table 15-93. DOUTTGL31_0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	DIO31	W	0h	This bit is used to toggle DIO31 output. 0h = No effect 1h = Toggle output
30	DIO30	W	0h	This bit is used to toggle DIO30 output. 0h = No effect 1h = Toggle output
29	DIO29	W	0h	This bit is used to toggle DIO29 output. 0h = No effect 1h = Toggle output
28	DIO28	W	0h	This bit is used to toggle DIO28 output. 0h = No effect 1h = Toggle output
27	DIO27	W	0h	This bit is used to toggle DIO27 output. 0h = No effect 1h = Toggle output
26	DIO26	W	0h	This bit is used to toggle DIO26 output. 0h = No effect 1h = Toggle output
25	DIO25	W	0h	This bit is used to toggle DIO25 output. 0h = No effect 1h = Toggle output
24	DIO24	W	0h	This bit is used to toggle DIO24 output. 0h = No effect 1h = Toggle output
23	DIO23	W	0h	This bit is used to toggle DIO23 output. 0h = No effect 1h = Toggle output
22	DIO22	W	0h	This bit is used to toggle DIO22 output. 0h = No effect 1h = Toggle output

Table 15-93. DOUTTGL31_0 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
21	DIO21	W	0h	This bit is used to toggle DIO21 output. 0h = No effect 1h = Toggle output
20	DIO20	W	0h	This bit is used to toggle DIO20 output. 0h = No effect 1h = Toggle output
19	DIO19	W	0h	This bit is used to toggle DIO19 output. 0h = No effect 1h = Toggle output
18	DIO18	W	0h	This bit is used to toggle DIO18 output. 0h = No effect 1h = Toggle output
17	DIO17	W	0h	This bit is used to toggle DIO17 output. 0h = No effect 1h = Toggle output
16	DIO16	W	0h	This bit is used to toggle DIO16 output. 0h = No effect 1h = Toggle output
15	DIO15	W	0h	This bit is used to toggle DIO15 output. 0h = No effect 1h = Toggle output
14	DIO14	W	0h	This bit is used to toggle DIO14 output. 0h = No effect 1h = Toggle output
13	DIO13	W	0h	This bit is used to toggle DIO13 output. 0h = No effect 1h = Toggle output
12	DIO12	W	0h	This bit is used to toggle DIO12 output. 0h = No effect 1h = Toggle output
11	DIO11	W	0h	This bit is used to toggle DIO11 output. 0h = No effect 1h = Toggle output
10	DIO10	W	0h	This bit is used to toggle DIO10 output. 0h = No effect 1h = Toggle output
9	DIO9	W	0h	This bit is used to toggle DIO9 output. 0h = No effect 1h = Toggle output
8	DIO8	W	0h	This bit is used to toggle DIO8 output. 0h = No effect 1h = Toggle output
7	DIO7	W	0h	This bit is used to toggle DIO7 output. 0h = No effect 1h = Toggle output
6	DIO6	W	0h	This bit is used to toggle DIO6 output. 0h = No effect 1h = Toggle output
5	DIO5	W	0h	This bit is used to toggle DIO5 output. 0h = No effect 1h = Toggle output
4	DIO4	W	0h	This bit is used to toggle DIO4 output. 0h = No effect 1h = Toggle output
3	DIO3	W	0h	This bit is used to toggle DIO3 output. 0h = No effect 1h = Toggle output

Table 15-93. DOUTTGL31_0 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
2	DIO2	W	0h	This bit is used to toggle DIO2 output. 0h = No effect 1h = Toggle output
1	DIO1	W	0h	This bit is used to toggle DIO1 output. 0h = No effect 1h = Toggle output
0	DIO0	W	0h	This bit is used to toggle DIO0 output. 0h = No effect 1h = Toggle output

45 DOE31_0 Register (Offset = 12C0h) [Reset = 00000000h]

DOE31_0 is shown in Figure 15-48 and described in Table 15-50.

Return to the Summary Table.

This register is used to enable the data outputs for DIO31 to DIO0.

Figure 15-48. DOE31_0 Register

31	30	29	28	27	26	25	24
DIO31	DIO30	DIO29	DIO28	DIO27	DIO26	DIO25	DIO24
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
DIO23	DIO22	DIO21	DIO20	DIO19	DIO18	DIO17	DIO16
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
DIO15	DIO14	DIO13	DIO12	DIO11	DIO10	DIO9	DIO8
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
DIO7	DIO6	DIO5	DIO4	DIO3	DIO2	DIO1	DIO0
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 15-95. DOE31_0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	DIO31	R/W	0h	Enables data output for DIO31. 0h = Output disabled 1h = Output enabled
30	DIO30	R/W	0h	Enables data output for DIO30. 0h = Output disabled 1h = Output enabled
29	DIO29	R/W	0h	Enables data output for DIO29. 0h = Output disabled 1h = Output enabled
28	DIO28	R/W	0h	Enables data output for DIO28. 0h = Output disabled 1h = Output enabled
27	DIO27	R/W	0h	Enables data output for DIO27. 0h = Output disabled 1h = Output enabled
26	DIO26	R/W	0h	Enables data output for DIO26. 0h = Output disabled 1h = Output enabled
25	DIO25	R/W	0h	Enables data output for DIO25. 0h = Output disabled 1h = Output enabled
24	DIO24	R/W	0h	Enables data output for DIO24. 0h = Output disabled 1h = Output enabled
23	DIO23	R/W	0h	Enables data output for DIO23. 0h = Output disabled 1h = Output enabled
22	DIO22	R/W	0h	Enables data output for DIO22. 0h = Output disabled 1h = Output enabled

Table 15-95. DOE31_0 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
21	DIO21	R/W	0h	Enables data output for DIO21. 0h = Output disabled 1h = Output enabled
20	DIO20	R/W	0h	Enables data output for DIO20. 0h = Output disabled 1h = Output enabled
19	DIO19	R/W	0h	Enables data output for DIO19. 0h = Output disabled 1h = Output enabled
18	DIO18	R/W	0h	Enables data output for DIO18. 0h = Output disabled 1h = Output enabled
17	DIO17	R/W	0h	Enables data output for DIO17. 0h = Output disabled 1h = Output enabled
16	DIO16	R/W	0h	Enables data output for DIO16. 0h = Output disabled 1h = Output enabled
15	DIO15	R/W	0h	Enables data output for DIO15. 0h = Output disabled 1h = Output enabled
14	DIO14	R/W	0h	Enables data output for DIO14. 0h = Output disabled 1h = Output enabled
13	DIO13	R/W	0h	Enables data output for DIO13. 0h = Output disabled 1h = Output enabled
12	DIO12	R/W	0h	Enables data output for DIO12. 0h = Output disabled 1h = Output enabled
11	DIO11	R/W	0h	Enables data output for DIO11. 0h = Output disabled 1h = Output enabled
10	DIO10	R/W	0h	Enables data output for DIO10. 0h = Output disabled 1h = Output enabled
9	DIO9	R/W	0h	Enables data output for DIO9. 0h = Output disabled 1h = Output enabled
8	DIO8	R/W	0h	Enables data output for DIO8. 0h = Output disabled 1h = Output enabled
7	DIO7	R/W	0h	Enables data output for DIO7. 0h = Output disabled 1h = Output enabled
6	DIO6	R/W	0h	Enables data output for DIO6. 0h = Output disabled 1h = Output enabled
5	DIO5	R/W	0h	Enables data output for DIO5. 0h = Output disabled 1h = Output enabled
4	DIO4	R/W	0h	Enables data output for DIO4. 0h = Output disabled 1h = Output enabled
3	DIO3	R/W	0h	Enables data output for DIO3. 0h = Output disabled 1h = Output enabled

Table 15-95. DOE31_0 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
2	DIO2	R/W	0h	Enables data output for DIO2. 0h = Output disabled 1h = Output enabled
1	DIO1	R/W	0h	Enables data output for DIO1. 0h = Output disabled 1h = Output enabled
0	DIO0	R/W	0h	Enables data output for DIO0. 0h = Output disabled 1h = Output enabled

46 DOESET31_0 Register (Offset = 12D0h) [Reset = 00000000h]

DOESET31_0 is shown in Figure 15-49 and described in Table 15-51.

Return to the Summary Table.

Writing 1 to a bit position in this register sets the corresponding bit in the DOE31_0 register.

Figure 15-49. DOESET31_0 Register

31	30	29	28	27	26	25	24
DIO31	DIO30	DIO29	DIO28	DIO27	DIO26	DIO25	DIO24
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h
23	22	21	20	19	18	17	16
DIO23	DIO22	DIO21	DIO20	DIO19	DIO18	DIO17	DIO16
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h
15	14	13	12	11	10	9	8
DIO15	DIO14	DIO13	DIO12	DIO11	DIO10	DIO9	DIO8
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h
7	6	5	4	3	2	1	0
DIO7	DIO6	DIO5	DIO4	DIO3	DIO2	DIO1	DIO0
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h

Table 15-97. DOESET31_0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	DIO31	W	0h	Writing 1 to this bit sets the DIO31 bit in the DOE31_0 register. Writing 0 has no effect. 0h = No effect 1h = Sets DIO31 in DOE31_0
30	DIO30	W	0h	Writing 1 to this bit sets the DIO30 bit in the DOE31_0 register. Writing 0 has no effect. 0h = No effect 1h = Sets DIO30 in DOE31_0
29	DIO29	W	0h	Writing 1 to this bit sets the DIO29 bit in the DOE31_0 register. Writing 0 has no effect. 0h = No effect 1h = Sets DIO29 in DOE31_0
28	DIO28	W	0h	Writing 1 to this bit sets the DIO28 bit in the DOE31_0 register. Writing 0 has no effect. 0h = No effect 1h = Sets DIO28 in DOE31_0
27	DIO27	W	0h	Writing 1 to this bit sets the DIO27 bit in the DOE31_0 register. Writing 0 has no effect. 0h = No effect 1h = Sets DIO27 in DOE31_0
26	DIO26	W	0h	Writing 1 to this bit sets the DIO26 bit in the DOE31_0 register. Writing 0 has no effect. 0h = No effect 1h = Sets DIO26 in DOE31_0
25	DIO25	W	0h	Writing 1 to this bit sets the DIO25 bit in the DOE31_0 register. Writing 0 has no effect. 0h = No effect 1h = Sets DIO25 in DOE31_0
24	DIO24	W	0h	Writing 1 to this bit sets the DIO24 bit in the DOE31_0 register. Writing 0 has no effect. 0h = No effect 1h = Sets DIO24 in DOE31_0

Table 15-97. DOESET31_0 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
23	DIO23	W	0h	Writing 1 to this bit sets the DIO23 bit in the DOE31_0 register. Writing 0 has no effect. 0h = No effect 1h = Sets DIO23 in DOE31_0
22	DIO22	W	0h	Writing 1 to this bit sets the DIO22 bit in the DOE31_0 register. Writing 0 has no effect. 0h = No effect 1h = Sets DIO22 in DOE31_0
21	DIO21	W	0h	Writing 1 to this bit sets the DIO21 bit in the DOE31_0 register. Writing 0 has no effect. 0h = No effect 1h = Sets DIO21 in DOE31_0
20	DIO20	W	0h	Writing 1 to this bit sets the DIO20 bit in the DOE31_0 register. Writing 0 has no effect. 0h = No effect 1h = Sets DIO20 in DOE31_0
19	DIO19	W	0h	Writing 1 to this bit sets the DIO19 bit in the DOE31_0 register. Writing 0 has no effect. 0h = No effect 1h = Sets DIO19 in DOE31_0
18	DIO18	W	0h	Writing 1 to this bit sets the DIO18 bit in the DOE31_0 register. Writing 0 has no effect. 0h = No effect 1h = Sets DIO18 in DOE31_0
17	DIO17	W	0h	Writing 1 to this bit sets the DIO17 bit in the DOE31_0 register. Writing 0 has no effect. 0h = No effect 1h = Sets DIO17 in DOE31_0
16	DIO16	W	0h	Writing 1 to this bit sets the DIO16 bit in the DOE31_0 register. Writing 0 has no effect. 0h = No effect 1h = Sets DIO16 in DOE31_0
15	DIO15	W	0h	Writing 1 to this bit sets the DIO15 bit in the DOE31_0 register. Writing 0 has no effect. 0h = No effect 1h = Sets DIO15 in DOE31_0
14	DIO14	W	0h	Writing 1 to this bit sets the DIO14 bit in the DOE31_0 register. Writing 0 has no effect. 0h = No effect 1h = Sets DIO14 in DOE31_0
13	DIO13	W	0h	Writing 1 to this bit sets the DIO13 bit in the DOE31_0 register. Writing 0 has no effect. 0h = No effect 1h = Sets DIO13 in DOE31_0
12	DIO12	W	0h	Writing 1 to this bit sets the DIO12 bit in the DOE31_0 register. Writing 0 has no effect. 0h = No effect 1h = Sets DIO12 in DOE31_0
11	DIO11	W	0h	Writing 1 to this bit sets the DIO11 bit in the DOE31_0 register. Writing 0 has no effect. 0h = No effect 1h = Sets DIO11 in DOE31_0
10	DIO10	W	0h	Writing 1 to this bit sets the DIO10 bit in the DOE31_0 register. Writing 0 has no effect. 0h = No effect 1h = Sets DIO10 in DOE31_0

Table 15-97. DOESET31_0 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
9	DIO9	W	0h	Writing 1 to this bit sets the DIO9 bit in the DOE31_0 register. Writing 0 has no effect. 0h = No effect 1h = Sets DIO9 in DOE31_0
8	DIO8	W	0h	Writing 1 to this bit sets the DIO8 bit in the DOE31_0 register. Writing 0 has no effect. 0h = No effect 1h = Sets DIO8 in DOE31_0
7	DIO7	W	0h	Writing 1 to this bit sets the DIO7 bit in the DOE31_0 register. Writing 0 has no effect. 0h = No effect 1h = Sets DIO7 in DOE31_0
6	DIO6	W	0h	Writing 1 to this bit sets the DIO6 bit in the DOE31_0 register. Writing 0 has no effect. 0h = No effect 1h = Sets DIO6 in DOE31_0
5	DIO5	W	0h	Writing 1 to this bit sets the DIO5 bit in the DOE31_0 register. Writing 0 has no effect. 0h = No effect 1h = Sets DIO5 in DOE31_0
4	DIO4	W	0h	Writing 1 to this bit sets the DIO4 bit in the DOE31_0 register. Writing 0 has no effect. 0h = No effect 1h = Sets DIO4 in DOE31_0
3	DIO3	W	0h	Writing 1 to this bit sets the DIO3 bit in the DOE31_0 register. Writing 0 has no effect. 0h = No effect 1h = Sets DIO3 in DOE31_0
2	DIO2	W	0h	Writing 1 to this bit sets the DIO2 bit in the DOE31_0 register. Writing 0 has no effect. 0h = No effect 1h = Sets DIO2 in DOE31_0
1	DIO1	W	0h	Writing 1 to this bit sets the DIO1 bit in the DOE31_0 register. Writing 0 has no effect. 0h = No effect 1h = Sets DIO1 in DOE31_0
0	DIO0	W	0h	Writing 1 to this bit sets the DIO0 bit in the DOE31_0 register. Writing 0 has no effect. 0h = No effect 1h = Sets DIO0 in DOE31_0

47 DOECLR31_0 Register (Offset = 12E0h) [Reset = 00000000h]

DOECLR31_0 is shown in Figure 15-50 and described in Table 15-52.

Return to the Summary Table.

Writing 1 to a bit position in this register clears the corresponding bit in the DOE31_0 register.

Figure 15-50. DOECLR31_0 Register

31	30	29	28	27	26	25	24
DIO31	DIO30	DIO29	DIO28	DIO27	DIO26	DIO25	DIO24
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h
23	22	21	20	19	18	17	16
DIO23	DIO22	DIO21	DIO20	DIO19	DIO18	DIO17	DIO16
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h
15	14	13	12	11	10	9	8
DIO15	DIO14	DIO13	DIO12	DIO11	DIO10	DIO9	DIO8
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h
7	6	5	4	3	2	1	0
DIO7	DIO6	DIO5	DIO4	DIO3	DIO2	DIO1	DIO0
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h

Table 15-99. DOECLR31_0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	DIO31	W	0h	Writing 1 to this bit clears the DIO31 bit in the DOE31_0 register. Writing 0 has no effect. 0h = No effect 1h = Clears DIO31 in DOE31_0
30	DIO30	W	0h	Writing 1 to this bit clears the DIO30 bit in the DOE31_0 register. Writing 0 has no effect. 0h = No effect 1h = Clears DIO30 in DOE31_0
29	DIO29	W	0h	Writing 1 to this bit clears the DIO29 bit in the DOE31_0 register. Writing 0 has no effect. 0h = No effect 1h = Clears DIO29 in DOE31_0
28	DIO28	W	0h	Writing 1 to this bit clears the DIO28 bit in the DOE31_0 register. Writing 0 has no effect. 0h = No effect 1h = Clears DIO28 in DOE31_0
27	DIO27	W	0h	Writing 1 to this bit clears the DIO27 bit in the DOE31_0 register. Writing 0 has no effect. 0h = No effect 1h = Clears DIO27 in DOE31_0
26	DIO26	W	0h	Writing 1 to this bit clears the DIO26 bit in the DOE31_0 register. Writing 0 has no effect. 0h = No effect 1h = Clears DIO26 in DOE31_0
25	DIO25	W	0h	Writing 1 to this bit clears the DIO25 bit in the DOE31_0 register. Writing 0 has no effect. 0h = No effect 1h = Clears DIO25 in DOE31_0
24	DIO24	W	0h	Writing 1 to this bit clears the DIO24 bit in the DOE31_0 register. Writing 0 has no effect. 0h = No effect 1h = Clears DIO24 in DOE31_0

Table 15-99. DOECLR31_0 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
23	DIO23	W	0h	Writing 1 to this bit clears the DIO23 bit in the DOE31_0 register. Writing 0 has no effect. 0h = No effect 1h = Clears DIO23 in DOE31_0
22	DIO22	W	0h	Writing 1 to this bit clears the DIO22 bit in the DOE31_0 register. Writing 0 has no effect. 0h = No effect 1h = Clears DIO22 in DOE31_0
21	DIO21	W	0h	Writing 1 to this bit clears the DIO21 bit in the DOE31_0 register. Writing 0 has no effect. 0h = No effect 1h = Clears DIO21 in DOE31_0
20	DIO20	W	0h	Writing 1 to this bit clears the DIO20 bit in the DOE31_0 register. Writing 0 has no effect. 0h = No effect 1h = Clears DIO20 in DOE31_0
19	DIO19	W	0h	Writing 1 to this bit clears the DIO19 bit in the DOE31_0 register. Writing 0 has no effect. 0h = No effect 1h = Clears DIO19 in DOE31_0
18	DIO18	W	0h	Writing 1 to this bit clears the DIO18 bit in the DOE31_0 register. Writing 0 has no effect. 0h = No effect 1h = Clears DIO18 in DOE31_0
17	DIO17	W	0h	Writing 1 to this bit clears the DIO17 bit in the DOE31_0 register. Writing 0 has no effect. 0h = No effect 1h = Clears DIO17 in DOE31_0
16	DIO16	W	0h	Writing 1 to this bit clears the DIO16 bit in the DOE31_0 register. Writing 0 has no effect. 0h = No effect 1h = Clears DIO16 in DOE31_0
15	DIO15	W	0h	Writing 1 to this bit clears the DIO15 bit in the DOE31_0 register. Writing 0 has no effect. 0h = No effect 1h = Clears DIO15 in DOE31_0
14	DIO14	W	0h	Writing 1 to this bit clears the DIO14 bit in the DOE31_0 register. Writing 0 has no effect. 0h = No effect 1h = Clears DIO14 in DOE31_0
13	DIO13	W	0h	Writing 1 to this bit clears the DIO13 bit in the DOE31_0 register. Writing 0 has no effect. 0h = No effect 1h = Clears DIO13 in DOE31_0
12	DIO12	W	0h	Writing 1 to this bit clears the DIO12 bit in the DOE31_0 register. Writing 0 has no effect. 0h = No effect 1h = Clears DIO12 in DOE31_0
11	DIO11	W	0h	Writing 1 to this bit clears the DIO11 bit in the DOE31_0 register. Writing 0 has no effect. 0h = No effect 1h = Clears DIO11 in DOE31_0
10	DIO10	W	0h	Writing 1 to this bit clears the DIO10 bit in the DOE31_0 register. Writing 0 has no effect. 0h = No effect 1h = Clears DIO10 in DOE31_0

Table 15-99. DOECLR31_0 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
9	DIO9	W	0h	Writing 1 to this bit clears the DIO9 bit in the DOE31_0 register. Writing 0 has no effect. 0h = No effect 1h = Clears DIO9 in DOE31_0
8	DIO8	W	0h	Writing 1 to this bit clears the DIO8 bit in the DOE31_0 register. Writing 0 has no effect. 0h = No effect 1h = Clears DIO8 in DOE31_0
7	DIO7	W	0h	Writing 1 to this bit clears the DIO7 bit in the DOE31_0 register. Writing 0 has no effect. 0h = No effect 1h = Clears DIO7 in DOE31_0
6	DIO6	W	0h	Writing 1 to this bit clears the DIO6 bit in the DOE31_0 register. Writing 0 has no effect. 0h = No effect 1h = Clears DIO6 in DOE31_0
5	DIO5	W	0h	Writing 1 to this bit clears the DIO5 bit in the DOE31_0 register. Writing 0 has no effect. 0h = No effect 1h = Clears DIO5 in DOE31_0
4	DIO4	W	0h	Writing 1 to this bit clears the DIO4 bit in the DOE31_0 register. Writing 0 has no effect. 0h = No effect 1h = Clears DIO4 in DOE31_0
3	DIO3	W	0h	Writing 1 to this bit clears the DIO3 bit in the DOE31_0 register. Writing 0 has no effect. 0h = No effect 1h = Clears DIO3 in DOE31_0
2	DIO2	W	0h	Writing 1 to this bit clears the DIO2 bit in the DOE31_0 register. Writing 0 has no effect. 0h = No effect 1h = Clears DIO2 in DOE31_0
1	DIO1	W	0h	Writing 1 to this bit clears the DIO1 bit in the DOE31_0 register. Writing 0 has no effect. 0h = No effect 1h = Clears DIO1 in DOE31_0
0	DIO0	W	0h	Writing 1 to this bit clears the DIO0 bit in the DOE31_0 register. Writing 0 has no effect. 0h = No effect 1h = Clears DIO0 in DOE31_0

48 DIN3_0 Register (Offset = 1300h) [Reset = 00000000h]

DIN3_0 is shown in Figure 15-51 and described in Table 15-53.

Return to the Summary Table.

Data input from pins configured as DIO3 to DIO0. This is an alias register for byte access to bits 3 to 0 in DIN31_0 register.

Figure 15-51. DIN3_0 Register

31	30	29	28	27	26	25	24
RESERVED							DIO3
R-0h							R-0h
23	22	21	20	19	18	17	16
RESERVED							DIO2
R-0h							R-0h
15	14	13	12	11	10	9	8
RESERVED							DIO1
R-0h							R-0h
7	6	5	4	3	2	1	0
RESERVED							DIO0
R-0h							R-0h

Table 15-101. DIN3_0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-25	RESERVED	R	0h	
24	DIO3	R	0h	This bit reads the data input value of DIO3. 0h = Input value is 0 1h = Input value is 1
23-17	RESERVED	R	0h	
16	DIO2	R	0h	This bit reads the data input value of DIO2. 0h = Input value is 0 1h = Input value is 1
15-9	RESERVED	R	0h	
8	DIO1	R	0h	This bit reads the data input value of DIO1. 0h = Input value is 0 1h = Input value is 1
7-1	RESERVED	R	0h	
0	DIO0	R	0h	This bit reads the data input value of DIO0. 0h = Input value is 0 1h = Input value is 1

49 DIN7_4 Register (Offset = 1304h) [Reset = 00000000h]

DIN7_4 is shown in Figure 15-52 and described in Table 15-54.

Return to the Summary Table.

Data input from pins configured as DIO7 to DIO4. This is an alias register for byte access to bits 7 to 4 in DIN31_0 register.

Figure 15-52. DIN7_4 Register

31	30	29	28	27	26	25	24
RESERVED							DIO7
R-0h							R-0h
23	22	21	20	19	18	17	16
RESERVED							DIO6
R-0h							R-0h
15	14	13	12	11	10	9	8
RESERVED							DIO5
R-0h							R-0h
7	6	5	4	3	2	1	0
RESERVED							DIO4
R-0h							R-0h

Table 15-103. DIN7_4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-25	RESERVED	R	0h	
24	DIO7	R	0h	This bit reads the data input value of DIO7. 0h = Input value is 0 1h = Input value is 1
23-17	RESERVED	R	0h	
16	DIO6	R	0h	This bit reads the data input value of DIO6. 0h = Input value is 0 1h = Input value is 1
15-9	RESERVED	R	0h	
8	DIO5	R	0h	This bit reads the data input value of DIO5. 0h = Input value is 0 1h = Input value is 1
7-1	RESERVED	R	0h	
0	DIO4	R	0h	This bit reads the data input value of DIO4. 0h = Input value is 0 1h = Input value is 1

50 DIN11_8 Register (Offset = 1308h) [Reset = 00000000h]

DIN11_8 is shown in Figure 15-53 and described in Table 15-55.

Return to the Summary Table.

Data input from pins configured as DIO11 to DIO8. This is an alias register for byte access to bits 11 to 8 in DIN31_0 register.

Figure 15-53. DIN11_8 Register

31	30	29	28	27	26	25	24
RESERVED							DIO11
R-0h							R-0h
23	22	21	20	19	18	17	16
RESERVED							DIO10
R-0h							R-0h
15	14	13	12	11	10	9	8
RESERVED							DIO9
R-0h							R-0h
7	6	5	4	3	2	1	0
RESERVED							DIO8
R-0h							R-0h

Table 15-105. DIN11_8 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-25	RESERVED	R	0h	
24	DIO11	R	0h	This bit reads the data input value of DIO11. 0h = Input value is 0 1h = Input value is 1
23-17	RESERVED	R	0h	
16	DIO10	R	0h	This bit reads the data input value of DIO10. 0h = Input value is 0 1h = Input value is 1
15-9	RESERVED	R	0h	
8	DIO9	R	0h	This bit reads the data input value of DIO9. 0h = Input value is 0 1h = Input value is 1
7-1	RESERVED	R	0h	
0	DIO8	R	0h	This bit reads the data input value of DIO8. 0h = Input value is 0 1h = Input value is 1

51 DIN15_12 Register (Offset = 130Ch) [Reset = 00000000h]

DIN15_12 is shown in Figure 15-54 and described in Table 15-56.

Return to the Summary Table.

Data input from pins configured as DIO15 to DIO12. This is an alias register for byte access to bits 15 to 12 in DIN31_0 register.

Figure 15-54. DIN15_12 Register

31	30	29	28	27	26	25	24
RESERVED							DIO15
R-0h							R-0h
23	22	21	20	19	18	17	16
RESERVED							DIO14
R-0h							R-0h
15	14	13	12	11	10	9	8
RESERVED							DIO13
R-0h							R-0h
7	6	5	4	3	2	1	0
RESERVED							DIO12
R-0h							R-0h

Table 15-107. DIN15_12 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-25	RESERVED	R	0h	
24	DIO15	R	0h	This bit reads the data input value of DIO15. 0h = Input value is 0 1h = Input value is 1
23-17	RESERVED	R	0h	
16	DIO14	R	0h	This bit reads the data input value of DIO14. 0h = Input value is 0 1h = Input value is 1
15-9	RESERVED	R	0h	
8	DIO13	R	0h	This bit reads the data input value of DIO13. 0h = Input value is 0 1h = Input value is 1
7-1	RESERVED	R	0h	
0	DIO12	R	0h	This bit reads the data input value of DIO12. 0h = Input value is 0 1h = Input value is 1

52 DIN19_16 Register (Offset = 1310h) [Reset = 00000000h]

DIN19_16 is shown in Figure 15-55 and described in Table 15-57.

Return to the Summary Table.

Data input from pins configured as DIO19 to DIO16. This is an alias register for byte access to bits 19 to 16 in DIN31_0 register.

Figure 15-55. DIN19_16 Register

31	30	29	28	27	26	25	24
RESERVED							DIO19
R-0h							R-0h
23	22	21	20	19	18	17	16
RESERVED							DIO18
R-0h							R-0h
15	14	13	12	11	10	9	8
RESERVED							DIO17
R-0h							R-0h
7	6	5	4	3	2	1	0
RESERVED							DIO16
R-0h							R-0h

Table 15-109. DIN19_16 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-25	RESERVED	R	0h	
24	DIO19	R	0h	This bit reads the data input value of DIO19. 0h = Input value is 0 1h = Input value is 1
23-17	RESERVED	R	0h	
16	DIO18	R	0h	This bit reads the data input value of DIO18. 0h = Input value is 0 1h = Input value is 1
15-9	RESERVED	R	0h	
8	DIO17	R	0h	This bit reads the data input value of DIO17. 0h = Input value is 0 1h = Input value is 1
7-1	RESERVED	R	0h	
0	DIO16	R	0h	This bit reads the data input value of DIO16. 0h = Input value is 0 1h = Input value is 1

53 DIN23_20 Register (Offset = 1314h) [Reset = 00000000h]

DIN23_20 is shown in Figure 15-56 and described in Table 15-58.

Return to the Summary Table.

Data input from pins configured as DIO23 to DIO20. This is an alias register for byte access to bits 23 to 20 in DIN31_0 register.

Figure 15-56. DIN23_20 Register

31	30	29	28	27	26	25	24
RESERVED							DIO23
R-0h							R-0h
23	22	21	20	19	18	17	16
RESERVED							DIO22
R-0h							R-0h
15	14	13	12	11	10	9	8
RESERVED							DIO21
R-0h							R-0h
7	6	5	4	3	2	1	0
RESERVED							DIO20
R-0h							R-0h

Table 15-111. DIN23_20 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-25	RESERVED	R	0h	
24	DIO23	R	0h	This bit reads the data input value of DIO23. 0h = Input value is 0 1h = Input value is 1
23-17	RESERVED	R	0h	
16	DIO22	R	0h	This bit reads the data input value of DIO22. 0h = Input value is 0 1h = Input value is 1
15-9	RESERVED	R	0h	
8	DIO21	R	0h	This bit reads the data input value of DIO21. 0h = Input value is 0 1h = Input value is 1
7-1	RESERVED	R	0h	
0	DIO20	R	0h	This bit reads the data input value of DIO20. 0h = Input value is 0 1h = Input value is 1

54 DIN27_24 Register (Offset = 1318h) [Reset = 00000000h]

DIN27_24 is shown in Figure 15-57 and described in Table 15-59.

Return to the Summary Table.

Data input from pins configured as DIO27 to DIO24. This is an alias register for byte access to bits 27 to 24 in DIN31_0 register.

Figure 15-57. DIN27_24 Register

31	30	29	28	27	26	25	24
RESERVED							DIO27
R-0h							R-0h
23	22	21	20	19	18	17	16
RESERVED							DIO26
R-0h							R-0h
15	14	13	12	11	10	9	8
RESERVED							DIO25
R-0h							R-0h
7	6	5	4	3	2	1	0
RESERVED							DIO24
R-0h							R-0h

Table 15-113. DIN27_24 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-25	RESERVED	R	0h	
24	DIO27	R	0h	This bit reads the data input value of DIO27. 0h = Input value is 0 1h = Input value is 1
23-17	RESERVED	R	0h	
16	DIO26	R	0h	This bit reads the data input value of DIO26. 0h = Input value is 0 1h = Input value is 1
15-9	RESERVED	R	0h	
8	DIO25	R	0h	This bit reads the data input value of DIO25. 0h = Input value is 0 1h = Input value is 1
7-1	RESERVED	R	0h	
0	DIO24	R	0h	This bit reads the data input value of DIO24. 0h = Input value is 0 1h = Input value is 1

55 DIN31_28 Register (Offset = 131Ch) [Reset = 00000000h]

DIN31_28 is shown in Figure 15-58 and described in Table 15-60.

Return to the Summary Table.

Data input from pins configured as DIO31 to DIO28. This is an alias register for byte access to bits 31 to 28 in DIN31_0 register.

Figure 15-58. DIN31_28 Register

31	30	29	28	27	26	25	24
RESERVED							DIO31
R-0h							R-0h
23	22	21	20	19	18	17	16
RESERVED							DIO30
R-0h							R-0h
15	14	13	12	11	10	9	8
RESERVED							DIO29
R-0h							R-0h
7	6	5	4	3	2	1	0
RESERVED							DIO28
R-0h							R-0h

Table 15-115. DIN31_28 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-25	RESERVED	R	0h	
24	DIO31	R	0h	This bit reads the data input value of DIO31. 0h = Input value is 0 1h = Input value is 1
23-17	RESERVED	R	0h	
16	DIO30	R	0h	This bit reads the data input value of DIO30. 0h = Input value is 0 1h = Input value is 1
15-9	RESERVED	R	0h	
8	DIO29	R	0h	This bit reads the data input value of DIO29. 0h = Input value is 0 1h = Input value is 1
7-1	RESERVED	R	0h	
0	DIO28	R	0h	This bit reads the data input value of DIO28. 0h = Input value is 0 1h = Input value is 1

56 DIN31_0 Register (Offset = 1380h) [Reset = 00000000h]

DIN31_0 is shown in Figure 15-59 and described in Table 15-61.

Return to the Summary Table.

Data input value for pins configured as DIO31 to DIO0.

Figure 15-59. DIN31_0 Register

31	30	29	28	27	26	25	24
DIO31	DIO30	DIO29	DIO28	DIO27	DIO26	DIO25	DIO24
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
23	22	21	20	19	18	17	16
DIO23	DIO22	DIO21	DIO20	DIO19	DIO18	DIO17	DIO16
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
15	14	13	12	11	10	9	8
DIO15	DIO14	DIO13	DIO12	DIO11	DIO10	DIO9	DIO8
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
7	6	5	4	3	2	1	0
DIO7	DIO6	DIO5	DIO4	DIO3	DIO2	DIO1	DIO0
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

Table 15-117. DIN31_0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	DIO31	R	0h	This bit reads the data input value of DIO31. 0h = Input value is 0 1h = Input value is 1
30	DIO30	R	0h	This bit reads the data input value of DIO30. 0h = Input value is 0 1h = Input value is 1
29	DIO29	R	0h	This bit reads the data input value of DIO29. 0h = Input value is 0 1h = Input value is 1
28	DIO28	R	0h	This bit reads the data input value of DIO28. 0h = Input value is 0 1h = Input value is 1
27	DIO27	R	0h	This bit reads the data input value of DIO27. 0h = Input value is 0 1h = Input value is 1
26	DIO26	R	0h	This bit reads the data input value of DIO26. 0h = Input value is 0 1h = Input value is 1
25	DIO25	R	0h	This bit reads the data input value of DIO25. 0h = Input value is 0 1h = Input value is 1
24	DIO24	R	0h	This bit reads the data input value of DIO24. 0h = Input value is 0 1h = Input value is 1
23	DIO23	R	0h	This bit reads the data input value of DIO23. 0h = Input value is 0 1h = Input value is 1
22	DIO22	R	0h	This bit reads the data input value of DIO22. 0h = Input value is 0 1h = Input value is 1

Table 15-117. DIN31_0 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
21	DIO21	R	0h	This bit reads the data input value of DIO21. 0h = Input value is 0 1h = Input value is 1
20	DIO20	R	0h	This bit reads the data input value of DIO20. 0h = Input value is 0 1h = Input value is 1
19	DIO19	R	0h	This bit reads the data input value of DIO19. 0h = Input value is 0 1h = Input value is 1
18	DIO18	R	0h	This bit reads the data input value of DIO18. 0h = Input value is 0 1h = Input value is 1
17	DIO17	R	0h	This bit reads the data input value of DIO17. 0h = Input value is 0 1h = Input value is 1
16	DIO16	R	0h	This bit reads the data input value of DIO16. 0h = Input value is 0 1h = Input value is 1
15	DIO15	R	0h	This bit reads the data input value of DIO15. 0h = Input value is 0 1h = Input value is 1
14	DIO14	R	0h	This bit reads the data input value of DIO14. 0h = Input value is 0 1h = Input value is 1
13	DIO13	R	0h	This bit reads the data input value of DIO13. 0h = Input value is 0 1h = Input value is 1
12	DIO12	R	0h	This bit reads the data input value of DIO12. 0h = Input value is 0 1h = Input value is 1
11	DIO11	R	0h	This bit reads the data input value of DIO11. 0h = Input value is 0 1h = Input value is 1
10	DIO10	R	0h	This bit reads the data input value of DIO10. 0h = Input value is 0 1h = Input value is 1
9	DIO9	R	0h	This bit reads the data input value of DIO9. 0h = Input value is 0 1h = Input value is 1
8	DIO8	R	0h	This bit reads the data input value of DIO8. 0h = Input value is 0 1h = Input value is 1
7	DIO7	R	0h	This bit reads the data input value of DIO7. 0h = Input value is 0 1h = Input value is 1
6	DIO6	R	0h	This bit reads the data input value of DIO6. 0h = Input value is 0 1h = Input value is 1
5	DIO5	R	0h	This bit reads the data input value of DIO5. 0h = Input value is 0 1h = Input value is 1
4	DIO4	R	0h	This bit reads the data input value of DIO4. 0h = Input value is 0 1h = Input value is 1
3	DIO3	R	0h	This bit reads the data input value of DIO3. 0h = Input value is 0 1h = Input value is 1

Table 15-117. DIN31_0 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
2	DIO2	R	0h	This bit reads the data input value of DIO2. 0h = Input value is 0 1h = Input value is 1
1	DIO1	R	0h	This bit reads the data input value of DIO1. 0h = Input value is 0 1h = Input value is 1
0	DIO0	R	0h	This bit reads the data input value of DIO0. 0h = Input value is 0 1h = Input value is 1

57 POLARITY15_0 Register (Offset = 1390h) [Reset = 00000000h]

POLARITY15_0 is shown in Figure 15-60 and described in Table 15-62.

Return to the Summary Table.

This register is used to enable and configure the polarity for input edge detection on DIO15 to DIO0. The corresponding DIO bits in RIS register will be set when the input event matches the configured polarity.

Figure 15-60. POLARITY15_0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DIO15		DIO14		DIO13		DIO12		DIO11		DIO10		DIO9		DIO8	
R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DIO7		DIO6		DIO5		DIO4		DIO3		DIO2		DIO1		DIO0	
R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h	

Table 15-119. POLARITY15_0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	DIO15	R/W	0h	Enables and configures edge detection polarity for DIO15. 0h = Edge detection disabled 1h = Detects rising edge of input event 2h = Detects falling edge of input event 3h = Detects both rising and falling edge of input event
29-28	DIO14	R/W	0h	Enables and configures edge detection polarity for DIO14. 0h = Edge detection disabled 1h = Detects rising edge of input event 2h = Detects falling edge of input event 3h = Detects both rising and falling edge of input event
27-26	DIO13	R/W	0h	Enables and configures edge detection polarity for DIO13. 0h = Edge detection disabled 1h = Detects rising edge of input event 2h = Detects falling edge of input event 3h = Detects both rising and falling edge of input event
25-24	DIO12	R/W	0h	Enables and configures edge detection polarity for DIO12. 0h = Edge detection disabled 1h = Detects rising edge of input event 2h = Detects falling edge of input event 3h = Detects both rising and falling edge of input event
23-22	DIO11	R/W	0h	Enables and configures edge detection polarity for DIO11. 0h = Edge detection disabled 1h = Detects rising edge of input event 2h = Detects falling edge of input event 3h = Detects both rising and falling edge of input event
21-20	DIO10	R/W	0h	Enables and configures edge detection polarity for DIO10. 0h = Edge detection disabled 1h = Detects rising edge of input event 2h = Detects falling edge of input event 3h = Detects both rising and falling edge of input event
19-18	DIO9	R/W	0h	Enables and configures edge detection polarity for DIO9. 0h = Edge detection disabled 1h = Detects rising edge of input event 2h = Detects falling edge of input event 3h = Detects both rising and falling edge of input event
17-16	DIO8	R/W	0h	Enables and configures edge detection polarity for DIO8. 0h = Edge detection disabled 1h = Detects rising edge of input event 2h = Detects falling edge of input event 3h = Detects both rising and falling edge of input event

Table 15-119. POLARITY15_0 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
15-14	DIO7	R/W	0h	Enables and configures edge detection polarity for DIO7. 0h = Edge detection disabled 1h = Detects rising edge of input event 2h = Detects falling edge of input event 3h = Detects both rising and falling edge of input event
13-12	DIO6	R/W	0h	Enables and configures edge detection polarity for DIO6. 0h = Edge detection disabled 1h = Detects rising edge of input event 2h = Detects falling edge of input event 3h = Detects both rising and falling edge of input event
11-10	DIO5	R/W	0h	Enables and configures edge detection polarity for DIO5. 0h = Edge detection disabled 1h = Detects rising edge of input event 2h = Detects falling edge of input event 3h = Detects both rising and falling edge of input event
9-8	DIO4	R/W	0h	Enables and configures edge detection polarity for DIO4. 0h = Edge detection disabled 1h = Detects rising edge of input event 2h = Detects falling edge of input event 3h = Detects both rising and falling edge of input event
7-6	DIO3	R/W	0h	Enables and configures edge detection polarity for DIO3. 0h = Edge detection disabled 1h = Detects rising edge of input event 2h = Detects falling edge of input event 3h = Detects both rising and falling edge of input event
5-4	DIO2	R/W	0h	Enables and configures edge detection polarity for DIO2. 0h = Edge detection disabled 1h = Detects rising edge of input event 2h = Detects falling edge of input event 3h = Detects both rising and falling edge of input event
3-2	DIO1	R/W	0h	Enables and configures edge detection polarity for DIO1. 0h = Edge detection disabled 1h = Detects rising edge of input event 2h = Detects falling edge of input event 3h = Detects both rising and falling edge of input event
1-0	DIO0	R/W	0h	Enables and configures edge detection polarity for DIO0. 0h = Edge detection disabled 1h = Detects rising edge of input event 2h = Detects falling edge of input event 3h = Detects both rising and falling edge of input event

58 POLARITY31_16 Register (Offset = 13A0h) [Reset = 00000000h]

POLARITY31_16 is shown in Figure 15-61 and described in Table 15-63.

Return to the Summary Table.

This register is used to enable and configure the polarity for input edge detection on DIO31 to DIO16. The corresponding DIO bits in RIS register will be set when the input event matches the configured polarity.

Figure 15-61. POLARITY31_16 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DIO31		DIO30		DIO29		DIO28		DIO27		DIO26		DIO25		DIO24	
R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DIO23		DIO22		DIO21		DIO20		DIO19		DIO18		DIO17		DIO16	
R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h	

Table 15-121. POLARITY31_16 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	DIO31	R/W	0h	Enables and configures edge detection polarity for DIO31. 0h = Edge detection disabled 1h = Detects rising edge of input event 2h = Detects falling edge of input event 3h = Detects both rising and falling edge of input event
29-28	DIO30	R/W	0h	Enables and configures edge detection polarity for DIO30. 0h = Edge detection disabled 1h = Detects rising edge of input event 2h = Detects falling edge of input event 3h = Detects both rising and falling edge of input event
27-26	DIO29	R/W	0h	Enables and configures edge detection polarity for DIO29. 0h = Edge detection disabled 1h = Detects rising edge of input event 2h = Detects falling edge of input event 3h = Detects both rising and falling edge of input event
25-24	DIO28	R/W	0h	Enables and configures edge detection polarity for DIO28. 0h = Edge detection disabled 1h = Detects rising edge of input event 2h = Detects falling edge of input event 3h = Detects both rising and falling edge of input event
23-22	DIO27	R/W	0h	Enables and configures edge detection polarity for DIO27. 0h = Edge detection disabled 1h = Detects rising edge of input event 2h = Detects falling edge of input event 3h = Detects both rising and falling edge of input event
21-20	DIO26	R/W	0h	Enables and configures edge detection polarity for DIO26. 0h = Edge detection disabled 1h = Detects rising edge of input event 2h = Detects falling edge of input event 3h = Detects both rising and falling edge of input event
19-18	DIO25	R/W	0h	Enables and configures edge detection polarity for DIO25. 0h = Edge detection disabled 1h = Detects rising edge of input event 2h = Detects falling edge of input event 3h = Detects both rising and falling edge of input event
17-16	DIO24	R/W	0h	Enables and configures edge detection polarity for DIO24. 0h = Edge detection disabled 1h = Detects rising edge of input event 2h = Detects falling edge of input event 3h = Detects both rising and falling edge of input event

Table 15-121. POLARITY31_16 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
15-14	DIO23	R/W	0h	Enables and configures edge detection polarity for DIO23. 0h = Edge detection disabled 1h = Detects rising edge of input event 2h = Detects falling edge of input event 3h = Detects both rising and falling edge of input event
13-12	DIO22	R/W	0h	Enables and configures edge detection polarity for DIO22. 0h = Edge detection disabled 1h = Detects rising edge of input event 2h = Detects falling edge of input event 3h = Detects both rising and falling edge of input event
11-10	DIO21	R/W	0h	Enables and configures edge detection polarity for DIO21. 0h = Edge detection disabled 1h = Detects rising edge of input event 2h = Detects falling edge of input event 3h = Detects both rising and falling edge of input event
9-8	DIO20	R/W	0h	Enables and configures edge detection polarity for DIO20. 0h = Edge detection disabled 1h = Detects rising edge of input event 2h = Detects falling edge of input event 3h = Detects both rising and falling edge of input event
7-6	DIO19	R/W	0h	Enables and configures edge detection polarity for DIO19. 0h = Edge detection disabled 1h = Detects rising edge of input event 2h = Detects falling edge of input event 3h = Detects both rising and falling edge of input event
5-4	DIO18	R/W	0h	Enables and configures edge detection polarity for DIO18. 0h = Edge detection disabled 1h = Detects rising edge of input event 2h = Detects falling edge of input event 3h = Detects both rising and falling edge of input event
3-2	DIO17	R/W	0h	Enables and configures edge detection polarity for DIO17. 0h = Edge detection disabled 1h = Detects rising edge of input event 2h = Detects falling edge of input event 3h = Detects both rising and falling edge of input event
1-0	DIO16	R/W	0h	Enables and configures edge detection polarity for DIO16. 0h = Edge detection disabled 1h = Detects rising edge of input event 2h = Detects falling edge of input event 3h = Detects both rising and falling edge of input event

59 CTL Register (Offset = 1400h) [Reset = 00000000h]

CTL is shown in Figure 15-62 and described in Table 15-64.

Return to the Summary Table.

GPIO Control Register

Figure 15-62. CTL Register

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

Table 15-123. CTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R/W	0h	
0	FASTWAKEONLY	R/W	0h	FASTWAKEONLY for the global control of fastwake 0h = The global control of fastwake is not enabled, per bit fast wake feature depends on FASTWAKE.DIN 1h = The global control of fastwake is enabled

ADVANCE INFORMATION

60 FASTWAKE Register (Offset = 1404h) [Reset = 00000000h]

FASTWAKE is shown in Figure 15-63 and described in Table 15-65.

Return to the Summary Table.

This is per bit fast wake enable for the bit slice, allows the GPIO module to stay in a low power state and not require high speed clocking of the input synchronizer or filter

Figure 15-63. FASTWAKE Register

31	30	29	28	27	26	25	24
DIN31	DIN30	DIN29	DIN28	DIN27	DIN26	DIN25	DIN24
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
DIN23	DIN22	DIN21	DIN20	DIN19	DIN18	DIN17	DIN16
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
DIN15	DIN14	DIN13	DIN12	DIN11	DIN10	DIN9	DIN8
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
DIN7	DIN6	DIN5	DIN4	DIN3	DIN2	DIN1	DIN0
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 15-125. FASTWAKE Register Field Descriptions

Bit	Field	Type	Reset	Description
31	DIN31	R/W	0h	Enable fastwake feature for DIN31 0h = fastwake feature is disabled 1h = fastwake feature is enabled
30	DIN30	R/W	0h	Enable fastwake feature for DIN30 0h = fastwake feature is disabled 1h = fastwake feature is enabled
29	DIN29	R/W	0h	Enable fastwake feature for DIN29 0h = fastwake feature is disabled 1h = fastwake feature is enabled
28	DIN28	R/W	0h	Enable fastwake feature for DIN29 0h = fastwake feature is disabled 1h = fastwake feature is enabled
27	DIN27	R/W	0h	Enable fastwake feature for DIN27 0h = fastwake feature is disabled 1h = fastwake feature is enabled
26	DIN26	R/W	0h	Enable fastwake feature for DIN26 0h = fastwake feature is disabled 1h = fastwake feature is enabled
25	DIN25	R/W	0h	Enable fastwake feature for DIN25 0h = fastwake feature is disabled 1h = fastwake feature is enabled
24	DIN24	R/W	0h	Enable fastwake feature for DIN24 0h = fastwake feature is disabled 1h = fastwake feature is enabled
23	DIN23	R/W	0h	Enable fastwake feature for DIN23 0h = fastwake feature is disabled 1h = fastwake feature is enabled
22	DIN22	R/W	0h	Enable fastwake feature for DIN22 0h = fastwake feature is disabled 1h = fastwake feature is enabled

Table 15-125. FASTWAKE Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
21	DIN21	R/W	0h	Enable fastwake feature for DIN21 0h = fastwake feature is disabled 1h = fastwake feature is enabled
20	DIN20	R/W	0h	Enable fastwake feature for DIN20 0h = fastwake feature is disabled 1h = fastwake feature is enabled
19	DIN19	R/W	0h	Enable fastwake feature for DIN19 0h = fastwake feature is disabled 1h = fastwake feature is enabled
18	DIN18	R/W	0h	Enable fastwake feature for DIN18 0h = fastwake feature is disabled 1h = fastwake feature is enabled
17	DIN17	R/W	0h	Enable fastwake feature for DIN17 0h = fastwake feature is disabled 1h = fastwake feature is enabled
16	DIN16	R/W	0h	Enable fastwake feature for DIN16 0h = fastwake feature is disabled 1h = fastwake feature is enabled
15	DIN15	R/W	0h	Enable fastwake feature for DIN15 0h = fastwake feature is disabled 1h = fastwake feature is enabled
14	DIN14	R/W	0h	Enable fastwake feature for DIN14 0h = fastwake feature is disabled 1h = fastwake feature is enabled
13	DIN13	R/W	0h	Enable fastwake feature for DIN13 0h = fastwake feature is disabled 1h = fastwake feature is enabled
12	DIN12	R/W	0h	Enable fastwake feature for DIN12 0h = fastwake feature is disabled 1h = fastwake feature is enabled
11	DIN11	R/W	0h	Enable fastwake feature for DIN11 0h = fastwake feature is disabled 1h = fastwake feature is enabled
10	DIN10	R/W	0h	Enable fastwake feature for DIN10 0h = fastwake feature is disabled 1h = fastwake feature is enabled
9	DIN9	R/W	0h	Enable fastwake feature for DIN9 0h = fastwake feature is disabled 1h = fastwake feature is enabled
8	DIN8	R/W	0h	Enable fastwake feature for DIN8 0h = fastwake feature is disabled 1h = fastwake feature is enabled
7	DIN7	R/W	0h	Enable fastwake feature for DIN7 0h = fastwake feature is disabled 1h = fastwake feature is enabled
6	DIN6	R/W	0h	Enable fastwake feature for DIN6 0h = fastwake feature is disabled 1h = fastwake feature is enabled
5	DIN5	R/W	0h	Enable fastwake feature for DIN5 0h = fastwake feature is disabled 1h = fastwake feature is enabled
4	DIN4	R/W	0h	Enable fastwake feature for DIN4 0h = fastwake feature is disabled 1h = fastwake feature is enabled
3	DIN3	R/W	0h	Enable fastwake feature for DIN3 0h = fastwake feature is disabled 1h = fastwake feature is enabled

Table 15-125. FASTWAKE Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
2	DIN2	R/W	0h	Enable fastwake feature for DIN2 0h = fastwake feature is disabled 1h = fastwake feature is enabled
1	DIN1	R/W	0h	Enable fastwake feature for DIN1 0h = fastwake feature is disabled 1h = fastwake feature is enabled
0	DIN0	R/W	0h	Enable fastwake feature for DIN0 0h = fastwake feature is disabled 1h = fastwake feature is enabled

61 SUB0CFG Register (Offset = 1500h) [Reset = 00000000h]

SUB0CFG is shown in Figure 15-64 and described in Table 15-66.

Return to the Summary Table.

This register is used to enable the subscriber 0 event and define the output policy on the selected DIO 0-15 pins.

Figure 15-64. SUB0CFG Register

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

Table 15-127. SUB0CFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-20	RESERVED	R/W	0h	
19-16	INDEX	R/W	0h	Indicates the specific bit among lower 16 bits that is targeted by the subscriber action 0h = specific bit targeted by the subscriber action is bit0 Fh = specific bit targeted by the subscriber action is bit15
15-10	RESERVED	R/W	0h	
9-8	OUTPOLICY	R/W	0h	These bits configure the output policy for subscriber 0 event. 0h = Selected DIO pins are set 1h = Selected DIO pins are cleared 2h = Selected DIO pins are toggled
7-1	RESERVED	R/W	0h	
0	ENABLE	R/W	0h	This bit is used to enable subscriber 0 event. 0h = Subscriber 0 event is disabled 1h = Subscriber 0 event is enabled

62 FILTEREN15_0 Register (Offset = 1508h) [Reset = 00000000h]

FILTEREN15_0 is shown in Figure 15-65 and described in Table 15-67.

Return to the Summary Table.

Programmable counter length of digital glitch filter for DIN0~15

Figure 15-65. FILTEREN15_0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DIN15		DIN14		DIN13		DIN12		DIN11		DIN10		DIN9		DIN8	
R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DIN7		DIN6		DIN5		DIN4		DIN3		DIN2		DIN1		DIN0	
R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h	

Table 15-129. FILTEREN15_0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	DIN15	R/W	0h	Programmable counter length of digital glitch filter for DIN15 0h = No additional filter beyond the CDC synchronization sample 1h = 1 ULPCLK minimum sample 2h = 3 ULPCLK minimum sample 3h = 8 ULPCLK minimum sample
29-28	DIN14	R/W	0h	Programmable counter length of digital glitch filter for DIN14 0h = No additional filter beyond the CDC synchronization sample 1h = 1 ULPCLK minimum sample 2h = 3 ULPCLK minimum sample 3h = 8 ULPCLK minimum sample
27-26	DIN13	R/W	0h	Programmable counter length of digital glitch filter for DIN13 0h = No additional filter beyond the CDC synchronization sample 1h = 1 ULPCLK minimum sample 2h = 3 ULPCLK minimum sample 3h = 8 ULPCLK minimum sample
25-24	DIN12	R/W	0h	Programmable counter length of digital glitch filter for DIN12 0h = No additional filter beyond the CDC synchronization sample 1h = 1 ULPCLK minimum sample 2h = 3 ULPCLK minimum sample 3h = 8 ULPCLK minimum sample
23-22	DIN11	R/W	0h	Programmable counter length of digital glitch filter for DIN11 0h = No additional filter beyond the CDC synchronization sample 1h = 1 ULPCLK minimum sample 2h = 3 ULPCLK minimum sample 3h = 8 ULPCLK minimum sample
21-20	DIN10	R/W	0h	Programmable counter length of digital glitch filter for DIN10 0h = No additional filter beyond the CDC synchronization sample 1h = 1 ULPCLK minimum sample 2h = 3 ULPCLK minimum sample 3h = 8 ULPCLK minimum sample
19-18	DIN9	R/W	0h	Programmable counter length of digital glitch filter for DIN9 0h = No additional filter beyond the CDC synchronization sample 1h = 1 ULPCLK minimum sample 2h = 3 ULPCLK minimum sample 3h = 8 ULPCLK minimum sample
17-16	DIN8	R/W	0h	Programmable counter length of digital glitch filter for DIN8 0h = No additional filter beyond the CDC synchronization sample 1h = 1 ULPCLK minimum sample 2h = 3 ULPCLK minimum sample 3h = 8 ULPCLK minimum sample

Table 15-129. FILTEREN15_0 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
15-14	DIN7	R/W	0h	Programmable counter length of digital glitch filter for DIN7 0h = No additional filter beyond the CDC synchronization sample 1h = 1 ULPCLK minimum sample 2h = 3 ULPCLK minimum sample 3h = 8 ULPCLK minimum sample
13-12	DIN6	R/W	0h	Programmable counter length of digital glitch filter for DIN6 0h = No additional filter beyond the CDC synchronization sample 1h = 1 ULPCLK minimum sample 2h = 3 ULPCLK minimum sample 3h = 8 ULPCLK minimum sample
11-10	DIN5	R/W	0h	Programmable counter length of digital glitch filter for DIN5 0h = No additional filter beyond the CDC synchronization sample 1h = 1 ULPCLK minimum sample 2h = 3 ULPCLK minimum sample 3h = 8 ULPCLK minimum sample
9-8	DIN4	R/W	0h	Programmable counter length of digital glitch filter for DIN4 0h = No additional filter beyond the CDC synchronization sample 1h = 1 ULPCLK minimum sample 2h = 3 ULPCLK minimum sample 3h = 8 ULPCLK minimum sample
7-6	DIN3	R/W	0h	Programmable counter length of digital glitch filter for DIN3 0h = No additional filter beyond the CDC synchronization sample 1h = 1 ULPCLK minimum sample 2h = 3 ULPCLK minimum sample 3h = 8 ULPCLK minimum sample
5-4	DIN2	R/W	0h	Programmable counter length of digital glitch filter for DIN2 0h = No additional filter beyond the CDC synchronization sample 1h = 1 ULPCLK minimum sample 2h = 3 ULPCLK minimum sample 3h = 8 ULPCLK minimum sample
3-2	DIN1	R/W	0h	Programmable counter length of digital glitch filter for DIN1 0h = No additional filter beyond the CDC synchronization sample 1h = 1 ULPCLK minimum sample 2h = 3 ULPCLK minimum sample 3h = 8 ULPCLK minimum sample
1-0	DIN0	R/W	0h	Programmable counter length of digital glitch filter for DIN0 0h = No additional filter beyond the CDC synchronization sample 1h = 1 ULPCLK minimum sample 2h = 3 ULPCLK minimum sample 3h = 8 ULPCLK minimum sample

63 FILTEREN31_16 Register (Offset = 150Ch) [Reset = 00000000h]

FILTEREN31_16 is shown in Figure 15-66 and described in Table 15-68.

Return to the Summary Table.

Programmable counter length of digital glitch filter for DIN16~31

Figure 15-66. FILTEREN31_16 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DIN31		DIN30		DIN29		DIN28		DIN27		DIN26		DIN25		DIN24	
R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DIN23		DIN22		DIN21		DIN20		DIN19		DIN18		DIN17		DIN16	
R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h	

Table 15-131. FILTEREN31_16 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	DIN31	R/W	0h	Programmable counter length of digital glitch filter for DIN31 0h = No additional filter beyond the CDC synchronization sample 1h = 1 ULPCLK minimum sample 2h = 3 ULPCLK minimum sample 3h = 8 ULPCLK minimum sample
29-28	DIN30	R/W	0h	Programmable counter length of digital glitch filter for DIN30 0h = No additional filter beyond the CDC synchronization sample 1h = 1 ULPCLK minimum sample 2h = 3 ULPCLK minimum sample 3h = 8 ULPCLK minimum sample
27-26	DIN29	R/W	0h	Programmable counter length of digital glitch filter for DIN29 0h = No additional filter beyond the CDC synchronization sample 1h = 1 ULPCLK minimum sample 2h = 3 ULPCLK minimum sample 3h = 8 ULPCLK minimum sample
25-24	DIN28	R/W	0h	Programmable counter length of digital glitch filter for DIN28 0h = No additional filter beyond the CDC synchronization sample 1h = 1 ULPCLK minimum sample 2h = 3 ULPCLK minimum sample 3h = 8 ULPCLK minimum sample
23-22	DIN27	R/W	0h	Programmable counter length of digital glitch filter for DIN27 0h = No additional filter beyond the CDC synchronization sample 1h = 1 ULPCLK minimum sample 2h = 3 ULPCLK minimum sample 3h = 8 ULPCLK minimum sample
21-20	DIN26	R/W	0h	Programmable counter length of digital glitch filter for DIN26 0h = No additional filter beyond the CDC synchronization sample 1h = 1 ULPCLK minimum sample 2h = 3 ULPCLK minimum sample 3h = 8 ULPCLK minimum sample
19-18	DIN25	R/W	0h	Programmable counter length of digital glitch filter for DIN25 0h = No additional filter beyond the CDC synchronization sample 1h = 1 ULPCLK minimum sample 2h = 3 ULPCLK minimum sample 3h = 8 ULPCLK minimum sample
17-16	DIN24	R/W	0h	Programmable counter length of digital glitch filter for DIN24 0h = No additional filter beyond the CDC synchronization sample 1h = 1 ULPCLK minimum sample 2h = 3 ULPCLK minimum sample 3h = 8 ULPCLK minimum sample

Table 15-131. FILTEREN31_16 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
15-14	DIN23	R/W	0h	Programmable counter length of digital glitch filter for DIN23 0h = No additional filter beyond the CDC synchronization sample 1h = 1 ULPCLK minimum sample 2h = 3 ULPCLK minimum sample 3h = 8 ULPCLK minimum sample
13-12	DIN22	R/W	0h	Programmable counter length of digital glitch filter for DIN22 0h = No additional filter beyond the CDC synchronization sample 1h = 1 ULPCLK minimum sample 2h = 3 ULPCLK minimum sample 3h = 8 ULPCLK minimum sample
11-10	DIN21	R/W	0h	Programmable counter length of digital glitch filter for DIN21 0h = No additional filter beyond the CDC synchronization sample 1h = 1 ULPCLK minimum sample 2h = 3 ULPCLK minimum sample 3h = 8 ULPCLK minimum sample
9-8	DIN20	R/W	0h	Programmable counter length of digital glitch filter for DIN20 0h = No additional filter beyond the CDC synchronization sample 1h = 1 ULPCLK minimum sample 2h = 3 ULPCLK minimum sample 3h = 8 ULPCLK minimum sample
7-6	DIN19	R/W	0h	Programmable counter length of digital glitch filter for DIN19 0h = No additional filter beyond the CDC synchronization sample 1h = 1 ULPCLK minimum sample 2h = 3 ULPCLK minimum sample 3h = 8 ULPCLK minimum sample
5-4	DIN18	R/W	0h	Programmable counter length of digital glitch filter for DIN18 0h = No additional filter beyond the CDC synchronization sample 1h = 1 ULPCLK minimum sample 2h = 3 ULPCLK minimum sample 3h = 8 ULPCLK minimum sample
3-2	DIN17	R/W	0h	Programmable counter length of digital glitch filter for DIN17 0h = No additional filter beyond the CDC synchronization sample 1h = 1 ULPCLK minimum sample 2h = 3 ULPCLK minimum sample 3h = 8 ULPCLK minimum sample
1-0	DIN16	R/W	0h	Programmable counter length of digital glitch filter for DIN16 0h = No additional filter beyond the CDC synchronization sample 1h = 1 ULPCLK minimum sample 2h = 3 ULPCLK minimum sample 3h = 8 ULPCLK minimum sample

64 DMAMASK Register (Offset = 1510h) [Reset = 00000000h]

DMAMASK is shown in Figure 15-67 and described in Table 15-69.

Return to the Summary Table.

DMA MASK which indicates which bit lanes the DMA is allowed to modify.

Figure 15-67. DMAMASK Register

31	30	29	28	27	26	25	24
DOUT31	DOUT30	DOUT29	DOUT28	DOUT27	DOUT26	DOUT25	DOUT24
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
DOUT23	DOUT22	DOUT21	DOUT20	DOUT19	DOUT18	DOUT17	DOUT16
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
DOUT15	DOUT14	DOUT13	DOUT12	DOUT11	DOUT10	DOUT9	DOUT8
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
DOUT7	DOUT6	DOUT5	DOUT4	DOUT3	DOUT2	DOUT1	DOUT0
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 15-133. DMAMASK Register Field Descriptions

Bit	Field	Type	Reset	Description
31	DOUT31	R/W	0h	DMA is allowed to modify DOUT31 0h = DMA is not allowed to modify this bit lane 1h = DMA is allowed to modify this bit lane
30	DOUT30	R/W	0h	DMA is allowed to modify DOUT30 0h = DMA is not allowed to modify this bit lane 1h = DMA is allowed to modify this bit lane
29	DOUT29	R/W	0h	DMA is allowed to modify DOUT29 0h = DMA is not allowed to modify this bit lane 1h = DMA is allowed to modify this bit lane
28	DOUT28	R/W	0h	DMA is allowed to modify DOUT28 0h = DMA is not allowed to modify this bit lane 1h = DMA is allowed to modify this bit lane
27	DOUT27	R/W	0h	DMA is allowed to modify DOUT27 0h = DMA is not allowed to modify this bit lane 1h = DMA is allowed to modify this bit lane
26	DOUT26	R/W	0h	DMA is allowed to modify DOUT26 0h = DMA is not allowed to modify this bit lane 1h = DMA is allowed to modify this bit lane
25	DOUT25	R/W	0h	DMA is allowed to modify DOUT25 0h = DMA is not allowed to modify this bit lane 1h = DMA is allowed to modify this bit lane
24	DOUT24	R/W	0h	DMA is allowed to modify DOUT24 0h = DMA is not allowed to modify this bit lane 1h = DMA is allowed to modify this bit lane
23	DOUT23	R/W	0h	DMA is allowed to modify DOUT23 0h = DMA is not allowed to modify this bit lane 1h = DMA is allowed to modify this bit lane
22	DOUT22	R/W	0h	DMA is allowed to modify DOUT22 0h = DMA is not allowed to modify this bit lane 1h = DMA is allowed to modify this bit lane

Table 15-133. DMAMASK Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
21	DOUT21	R/W	0h	DMA is allowed to modify DOUT21 0h = DMA is not allowed to modify this bit lane 1h = DMA is allowed to modify this bit lane
20	DOUT20	R/W	0h	DMA is allowed to modify DOUT20 0h = DMA is not allowed to modify this bit lane 1h = DMA is allowed to modify this bit lane
19	DOUT19	R/W	0h	DMA is allowed to modify DOUT19 0h = DMA is not allowed to modify this bit lane 1h = DMA is allowed to modify this bit lane
18	DOUT18	R/W	0h	DMA is allowed to modify DOUT18 0h = DMA is not allowed to modify this bit lane 1h = DMA is allowed to modify this bit lane
17	DOUT17	R/W	0h	DMA is allowed to modify DOUT17 0h = DMA is not allowed to modify this bit lane 1h = DMA is allowed to modify this bit lane
16	DOUT16	R/W	0h	DMA is allowed to modify DOUT16 0h = DMA is not allowed to modify this bit lane 1h = DMA is allowed to modify this bit lane
15	DOUT15	R/W	0h	DMA is allowed to modify DOUT15 0h = DMA is not allowed to modify this bit lane 1h = DMA is allowed to modify this bit lane
14	DOUT14	R/W	0h	DMA is allowed to modify DOUT14 0h = DMA is not allowed to modify this bit lane 1h = DMA is allowed to modify this bit lane
13	DOUT13	R/W	0h	DMA is allowed to modify DOUT13 0h = DMA is not allowed to modify this bit lane 1h = DMA is allowed to modify this bit lane
12	DOUT12	R/W	0h	DMA is allowed to modify DOUT12 0h = DMA is not allowed to modify this bit lane 1h = DMA is allowed to modify this bit lane
11	DOUT11	R/W	0h	DMA is allowed to modify DOUT11 0h = DMA is not allowed to modify this bit lane 1h = DMA is allowed to modify this bit lane
10	DOUT10	R/W	0h	DMA is allowed to modify DOUT10 0h = DMA is not allowed to modify this bit lane 1h = DMA is allowed to modify this bit lane
9	DOUT9	R/W	0h	DMA is allowed to modify DOUT9 0h = DMA is not allowed to modify this bit lane 1h = DMA is allowed to modify this bit lane
8	DOUT8	R/W	0h	DMA is allowed to modify DOUT8 0h = DMA is not allowed to modify this bit lane 1h = DMA is allowed to modify this bit lane
7	DOUT7	R/W	0h	DMA is allowed to modify DOUT7 0h = DMA is not allowed to modify this bit lane 1h = DMA is allowed to modify this bit lane
6	DOUT6	R/W	0h	DMA is allowed to modify DOUT6 0h = DMA is not allowed to modify this bit lane 1h = DMA is allowed to modify this bit lane
5	DOUT5	R/W	0h	DMA is allowed to modify DOUT5 0h = DMA is not allowed to modify this bit lane 1h = DMA is allowed to modify this bit lane
4	DOUT4	R/W	0h	DMA is allowed to modify DOUT4 0h = DMA is not allowed to modify this bit lane 1h = DMA is allowed to modify this bit lane
3	DOUT3	R/W	0h	DMA is allowed to modify DOUT3 0h = DMA is not allowed to modify this bit lane 1h = DMA is allowed to modify this bit lane

Table 15-133. DMAMASK Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
2	DOUT2	R/W	0h	DMA is allowed to modify DOUT2 0h = DMA is not allowed to modify this bit lane 1h = DMA is allowed to modify this bit lane
1	DOUT1	R/W	0h	DMA is allowed to modify DOUT1 0h = DMA is not allowed to modify this bit lane 1h = DMA is allowed to modify this bit lane
0	DOUT0	R/W	0h	DMA is allowed to modify DOUT0 0h = DMA is not allowed to modify this bit lane 1h = DMA is allowed to modify this bit lane

65 SUB1CFG Register (Offset = 1520h) [Reset = 00000000h]

SUB1CFG is shown in Figure 15-68 and described in Table 15-70.

Return to the Summary Table.

This register is used to enable the subscriber 1 event and define the output policy on the selected DIO 16-31 pins.

Figure 15-68. SUB1CFG Register

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

Table 15-135. SUB1CFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-20	RESERVED	R/W	0h	
19-16	INDEX	R/W	0h	indicates the specific bit in the upper 16 bits that is targeted by the subscriber action 0h = specific bit targeted by the subscriber action is bit16 Fh = specific bit targeted by the subscriber action is bit31
15-10	RESERVED	R/W	0h	
9-8	OUTPOLICY	R/W	0h	These bits configure the output policy for subscriber 1 event. 0h = Selected DIO pins are set 1h = Selected DIO pins are cleared 2h = Selected DIO pins are toggled
7-1	RESERVED	R/W	0h	
0	ENABLE	R/W	0h	This bit is used to enable subscriber 1 event. 0h = Subscriber 1 event is disabled 1h = Subscriber 1 event is enabled

66 TEST0 Register (Offset = 1E00h) [Reset = 00000000h]

TEST0 is shown in Figure 15-69 and described in Table 15-71.

Return to the Summary Table.

Test 0 register.

Figure 15-69. TEST0 Register

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

Table 15-137. TEST0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R/W	0h	
3-0	DTB_MUX_SEL	R/W	0h	This bit field is used to select DTB mux digital inputs. 0h = Selects test group 0 1h = Selects test group 1 2h = Selects test group 2 Fh = Selects test group 15

Chapter 16

Analog-to-Digital Converter (ADC)

The analog-to-digital converter (ADC) module described in this chapter is a Type 7 ADC.

ADVANCE INFORMATION

16.1 Introduction

The ADC module is a 12-bit successive approximation (SAR) style ADC . The ADC is composed of a core and a wrapper. The core is composed of the analog circuits which include the channel select MUX, the sample-and-hold (S/H) circuit, the successive approximation circuits, voltage reference circuits, and other analog support circuits. The wrapper is composed of the digital circuits that configure and control the ADC. These circuits include the logic for programmable conversions, result registers, interfaces to analog circuits, interfaces to the peripheral buses, post-processing circuits, and interfaces to other on-chip modules.

Each ADC module consists of a single sample-and-hold (S/H) circuit. The ADC module is designed to be duplicated multiple times on the same chip, allowing simultaneous sampling or independent operation of multiple ADCs. The ADC wrapper is start-of-conversion (SOC) based (see Section 16.3).

16.1.1 Features

The ADC supports the following features:

- 12-bit resolution
- Selectable internal reference of 2.5V or 3.3V
- Single-ended (SE) signal conversions
- Input multiplexer with up to 32 channels
- 16 configurable SOCs
- Type 4 digital wrappers that enhances the ADC capabilities to include:
  - Over and Under Sampling
  - External Channel support with at least 2 bit external select per ADC
- 16 individually addressable result registers
- One trigger repeater module, enabling customizable hardware oversampling modes with little or no CPU overhead
- Multiple trigger sources:
  - S/W - software immediate start
  - All MCPWMs- ADCSOC A, B, C or D
  - TIMG4_0.GEN_EVENT0/TIMG12_0_GEN_EVENT0
  - GPIO: INPUTXBAR[5]
  - ADCINT1/2
- Three flexible interrupts and two DMA triggers
- Configurable interrupt placement
- Three post-processing blocks, each with:
  - Saturating offset calibration
  - Error from set-point calculation
  - High, low, zero-crossing and in-limit compare, with interrupt and MCPWM trip capability
  - Aggregation functions: sum and average (binary shift) (on PPB1 only)

16.1.2 ADC Related Collateral

Foundational Materials

- *ADC Input Circuit Evaluation for C2000 MCUs (PSPICE for TI) Application Report*
- *ADC Input Circuit Evaluation for C2000 MCUs (TINA-TI) Application Report*
- PSpice for TI design and simulation tool
- *Real-Time Control Reference Guide*
  - Refer to the ADC section
- TI Precision Labs - ADCs
- TI Precision Labs: Driving the reference input on a SAR ADC (Video)
- TI Precision Labs: Introduction to analog-to-digital converters (ADCs) (Video)
- TI Precision Labs: SAR ADC input driver design (Video)
- TI e2e: Connecting VDDA to VREFHI
- TI e2e: Topologies for ADC Input Protection

ADVANCE INFORMATION

- TI e2e: Why does the ADC Input Voltage drop with sampling?
  - Sampling a high impedance voltage divider with ADC
- *Understanding Data Converters Application Report*
- *VREFHI Input Driver Design for C2000 MCUs Application Report*

Getting Started Materials

- ADC-PWM Synchronization Using ADC Interrupt
  - NOTE: This is a non-TI (third party) site.
- *Hardware Design Guide for F2800x C2000 Real-Time MCU Series*

Expert Materials

- *A potential firmware mistake may lead to control instability*
- Analog Engineer's Calculator
- *Analog Engineer's Pocket Reference*
- Debugging an integrated ADC in a microcontroller using an oscilloscope
- TI Precision Labs: ADC AC specifications (Video)
- TI Precision Labs: ADC Error sources (Video)
- TI Precision Labs: ADC Noise (Video)
- TI Precision Labs: Analog-to-digital converter (ADC) drive topologies (Video)
- TI Precision Labs: Electrical overstress on data converters (Video)
- TI Precision Labs: High-speed ADC fundamentals (Video)
- TI Precision Labs: SAR & Delta-Sigma: Understanding the Difference (Video)
- TI e2e: ADC Bandwidth Clarification
- TI e2e: ADC Resolution with Oversampling
- TI e2e: ADC configuration for interleaved mode
- TI e2e: Simultaneous Sampling with Single ADC

16.1.3 Block Diagram

Figure 16-1 shows the block diagram for the ADC core and ADC wrapper.

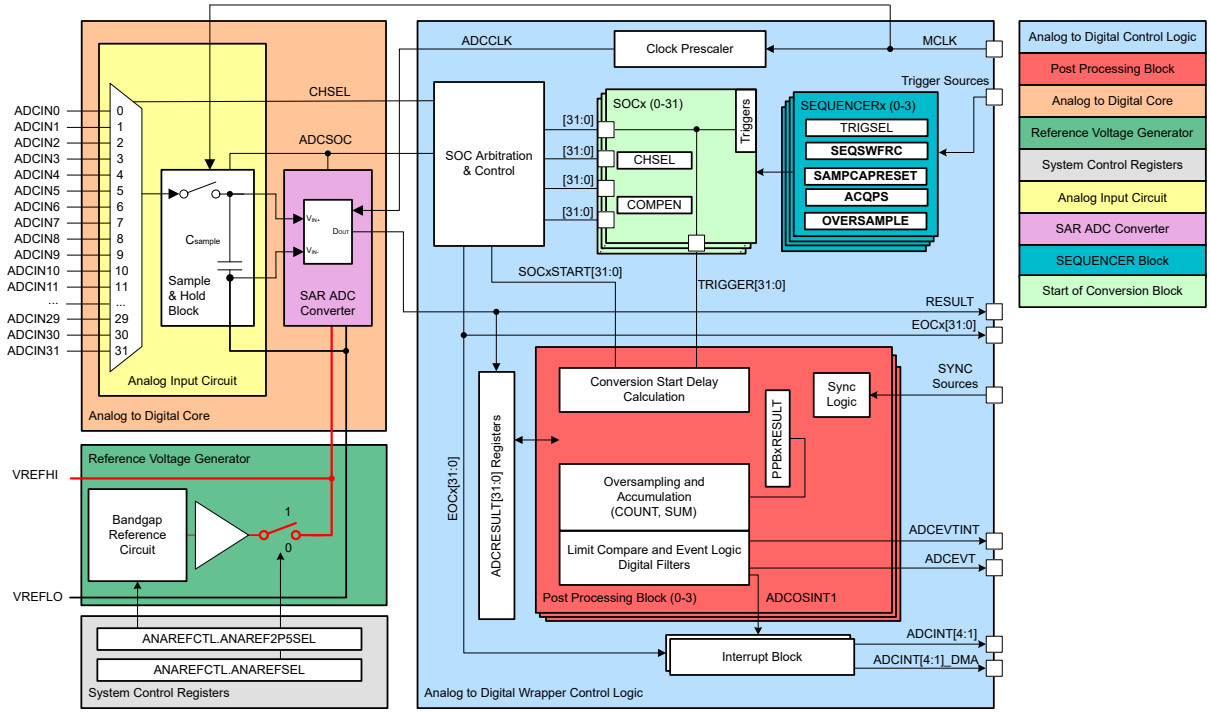


Figure 16-1. ADC Module Block Diagram

Note

- The ADC block diagram reflects the number of ADC channels internally configurable on the device. The actual number of available external ADC inputs varies depending on device part number and package type.

ADVANCE INFORMATION

16.2 ADC Configurability

Some ADC configurations are individually controlled by the SOCs, while others are globally controlled per ADC module. Table 16-1 summarizes the basic ADC options and the level of configurability. The subsequent sections discuss these configurations.

Table 16-1. ADC Options and Configuration Levels

Options	Configurability
Clock	Per module(1)
Resolution	Not configurable (12-bit only)
Signal Mode	Not configurable (single-ended only)
Reference Voltage Source	Either external or internal for all modules
Trigger Source	Per SOC(1)
Converted Channel	Per SOC
Acquisition Window Duration	Per SOC(1)
EOC Location	Per module
Burst Mode	Per module(1)
Sample capacitor reset	Per SOC(1)

(1) Writing these values differently to different ADC modules can cause the ADCs to operate asynchronously. See *Ensuring Synchronous Operation* for guidance on when the ADCs are operating synchronously or asynchronously.

16.2.1 ADC Clock Configuration

The base ADC clock is provided directly by the system clock. MCLK is used to generate the ADC acquisition window. The register ADCCTL2 has a PRESCALE field that determines the ADCCLK. ADCCLK is used to clock the converter, and is only active during the conversion phase. At all other times, including during the sample-and-hold window, the ADCCLK signal is gated off.

The ADC core requires approximately 11 ADCCLK cycles to process a voltage into a conversion result. The user must determine the required duration of the acquisition window, see *Choosing an Acquisition Window Duration*.

Note

To determine an appropriate value for ADCCTL2.PRESCALE, see the device data sheet for the maximum allowable ADCCLK frequencies.

16.2.2 Resolution

The resolution of the ADC determines how finely the analog range is quantized into digital values. Each ADC module supports a fixed resolution of 12 bits.

ADVANCE INFORMATION

16.2.3 Voltage Reference

16.2.3.1 External Reference Mode

The ADC module share VREFHI and VREFLO inputs. In external reference mode, these pins are used as a ratiometric reference to determine the ADC conversion input range.

See Section 16.13.6 for information on how to supply the reference voltage.

Note

- On devices with no external VREFHI and VREFLO pin, VREFHI is internally tied to VDDA and VREFLO is internally connected to the device analog ground, VSSA. Internal reference mode is not available for packages lacking VREFHI and VREFLO pins. See the device data sheet for packages with VREFHI and VREFLO pins available.
- To enable the external reference mode, the bit ANAREFSEL in register REFCONFIGA from AnalogSubsysRegs must be set to 1 (AnalogSubsysRegs.REFCONFIGA.bit.ANAREFSEL = 1).
- See the device data sheet to determine the allowable voltage range for VREFHI and VREFLO.
- The external reference mode requires an external capacitor on the VREFHI pin. See the device data sheet for the specific value required.

16.2.3.2 Internal Reference Mode

In internal reference mode, the device drives a voltage out onto the VREFHI pin. The VREFHI and VREFLO pins then set the ADC conversion range. Internal reference mode is not available for packages without VREFHI and VREFLO pins. See the device data sheet for packages with VREFHI and VREFLO pins available.

The internal reference voltage can be configured to be either 2.5V or 1.65V. When the 1.65V internal reference voltage is selected, the ADC input signal is internally divided by 2 before conversion, which effectively makes the ADC conversion range from VREFLO to 3.3V.

Note

The internal reference mode also requires an external capacitor on the VREFHI pin. See the device data sheet for the specific value required.

16.2.3.3 Selecting Reference Mode

The voltage reference mode must be configured by using either the ADC_setVREF() or the SetVREF() functions, depending on the header files used, provided in the Device SDK. Using either of these functions makes sure that the correct trim is loaded in the ADC trim registers. This function must be called at least once after a device reset. Do not configure the voltage reference mode by directly writing to the ANAREFCTL register.

16.2.4 Signal Mode

The ADC supports single-ended signaling.

In single-ended mode, the input voltage to the converter is sampled through a single pin (ADCINx), referenced to VREFLO.

ADVANCE INFORMATION

16.2.4.1 Expected Conversion Results

Based on a given analog input voltage, the expected digital conversion is given in Table 16-2. Fractional values are truncated.

Table 16-2. Analog to 12-bit Digital Formulas

Analog Input	Digital Result
when ADCINy ≤ VREFLO	ADCRESULTx = 0
when VREFLO < ADCINy < VREFHI	$\text{ADCRESULTx} = 4096 \left(\frac{\text{ADCINy - VREFLO}}{\text{VREFHI - VREFLO}} \right)$
when ADCINy ≥ VREFHI	ADCRESULTx = 4095

16.2.4.2 Interpreting Conversion Results

Based on a given ADC conversion result, the corresponding analog input is given in Table 16-3. This corresponds to the center of the possible range of analog voltages that can produce this conversion result.

Table 16-3. 12-Bit Digital-to-Analog Formulas

Digital Value	Analog Equivalent	
when ADCRESULTy = 0	$\text{ADCINx} \leq \text{VREFLO}$	(11)
when 0 < ADCRESULTy < 4095	$\text{ADCINx} = (\text{VREFHI} - \text{VREFLO})\left(\frac{\text{ADCRESULTy}}{4096}\right) + \text{VREFLO}$	(12)
when ADCRESULTy = 4095	$\text{ADCINx} \geq \text{VREFHI}$	(13)

ADVANCE INFORMATION

16.3 SOC Principle of Operation

The ADC triggering and conversion sequencing is accomplished through configurable start-of-conversions (SOCs). Each SOC is a configuration set defining the single conversion of a single channel. In that set, there are three configurations: the trigger source that starts the conversion, the channel to convert, and the acquisition (sample) window duration. Upon receiving the trigger configured for a SOC, the wrapper makes sure that the specified channel is captured using the specified acquisition window duration.

Multiple SOCs can be configured for the same trigger, channel, and acquisition window as desired. Configuring multiple SOCs to use the same trigger allows the trigger to generate a sequence of conversions. Configuring multiple SOCs to use the same trigger and channel allows for oversampling. Oversampling can also be achieved using a single trigger source by configuring the trigger repeater module. See *Trigger Repeaters* for more information.

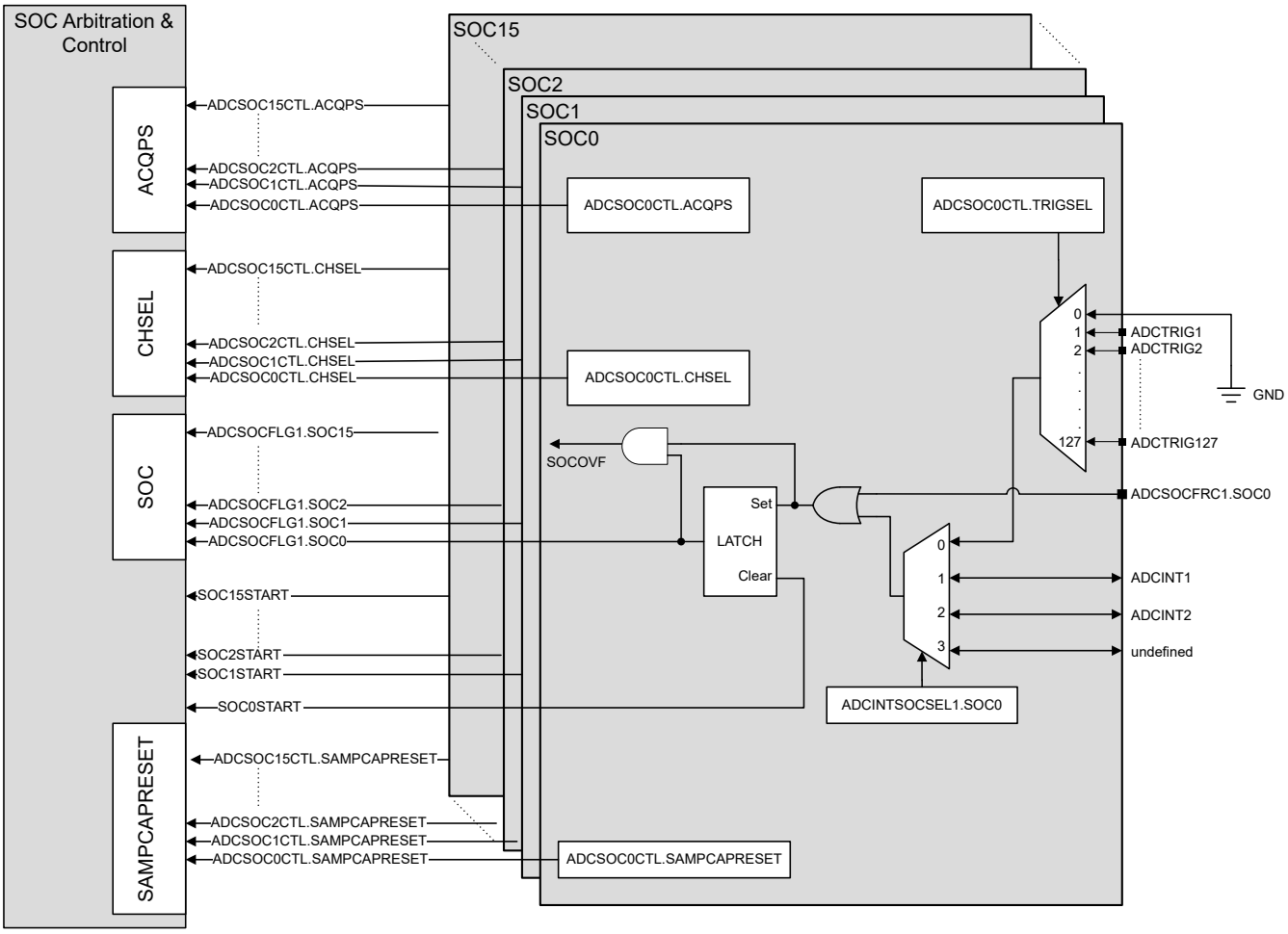


Figure 16-2. SOC Block Diagram

16.3.1 ADC Sequencer

Sequencer Overview

The ADC supports up to four sequencers, which are used to sequence the order in which ADC channels are converted. Each sequencer is associated with multiple SOCs, and the number of SOCs within a sequencer is configured using the ADCSEQ{#}CONFIG.SEQSTART and ADCSEQ{#+1}CONFIG.SEQSTART-1 or ADCSEQCTL.SEQEND registers. Sequencers are enabled using the ADCSEQ{#}CONFIG.SEQENABLE register.

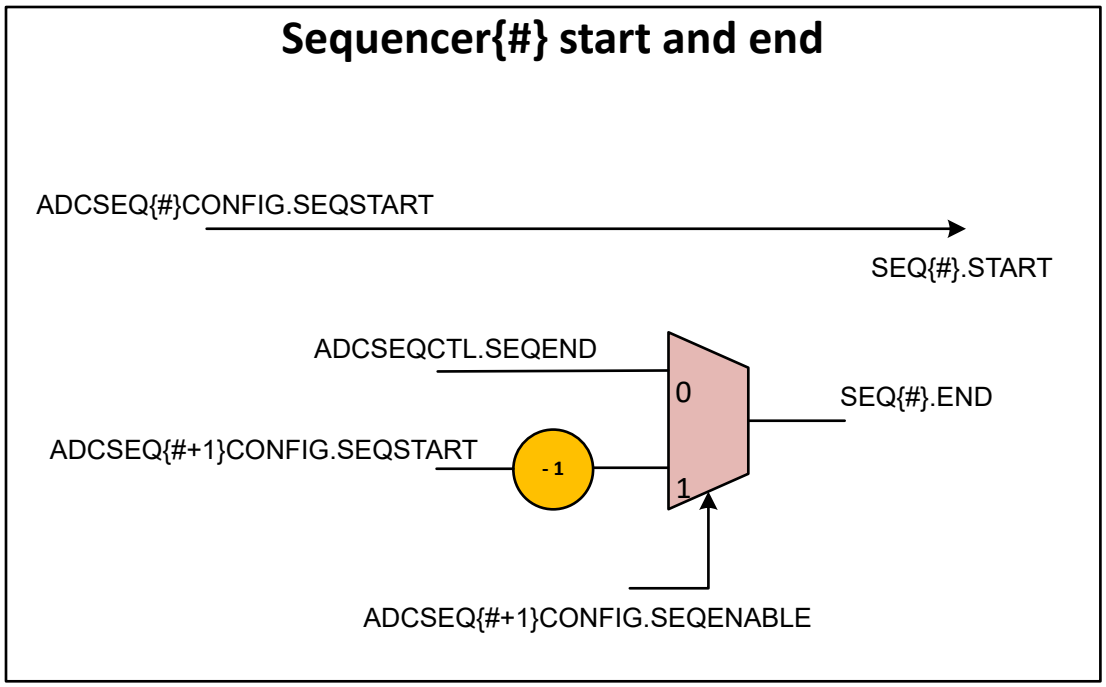


Figure 16-3. Sequencer start and end

Once a trigger is received for a sequencer and the ADC Wrapper is ready for arbitration, the sequencer starts converting in ascending order of SOCs. The first SOC in the sequence to convert is determined by ADCSEQ{#}CONFIG.SEQSTART, followed by subsequent SOCs until the end of the sequencer is reached.

Multiple sequencers can be configured to use the same trigger if desired. If multiple sequencers are triggered at the same time, the conversions within the sequence happens in the order of priority: SEQ1 followed by SEQ2, SEQ3, and SEQ4.

If a trigger for higher priority sequence occurs while a lower priority sequence is ongoing, the higher priority sequence can either be serviced as soon as the ongoing SOC conversion is complete or serviced after the lower priority sequence is complete. This can be configured using ADCSEQCTL.SEQPREEMPT register.

The ADCSEQCTL.SEQPREEMPT register configuration is as follows

- (2'b0x) Pre-empt disabled: The ongoing sequence fully completes before switching to higher priority sequence.
- (2'b10)Pre-empt enabled without restart of aborted SEQ : The ongoing sequence is discarded (after completing any ongoing conversion) and arbitration switches to higher priority sequence. The discarded/ aborted sequence is not be re-started after completing the higher priority sequence.
- (2'b11)Pre-empt enabled with restart of aborted SEQ: The ongoing sequence is discarded (after completing any ongoing conversion) and arbitration switches to the higher priority sequence. The discarded/aborted sequence is re-started from the beginning of the sequence after completing all higher priority sequences.

Figure 16-4 illustrates the logical representation of sequencer{#} enable and overflow logic, which generates the overflow condition when the new trigger is received while a previous trigger is ongoing or pending.

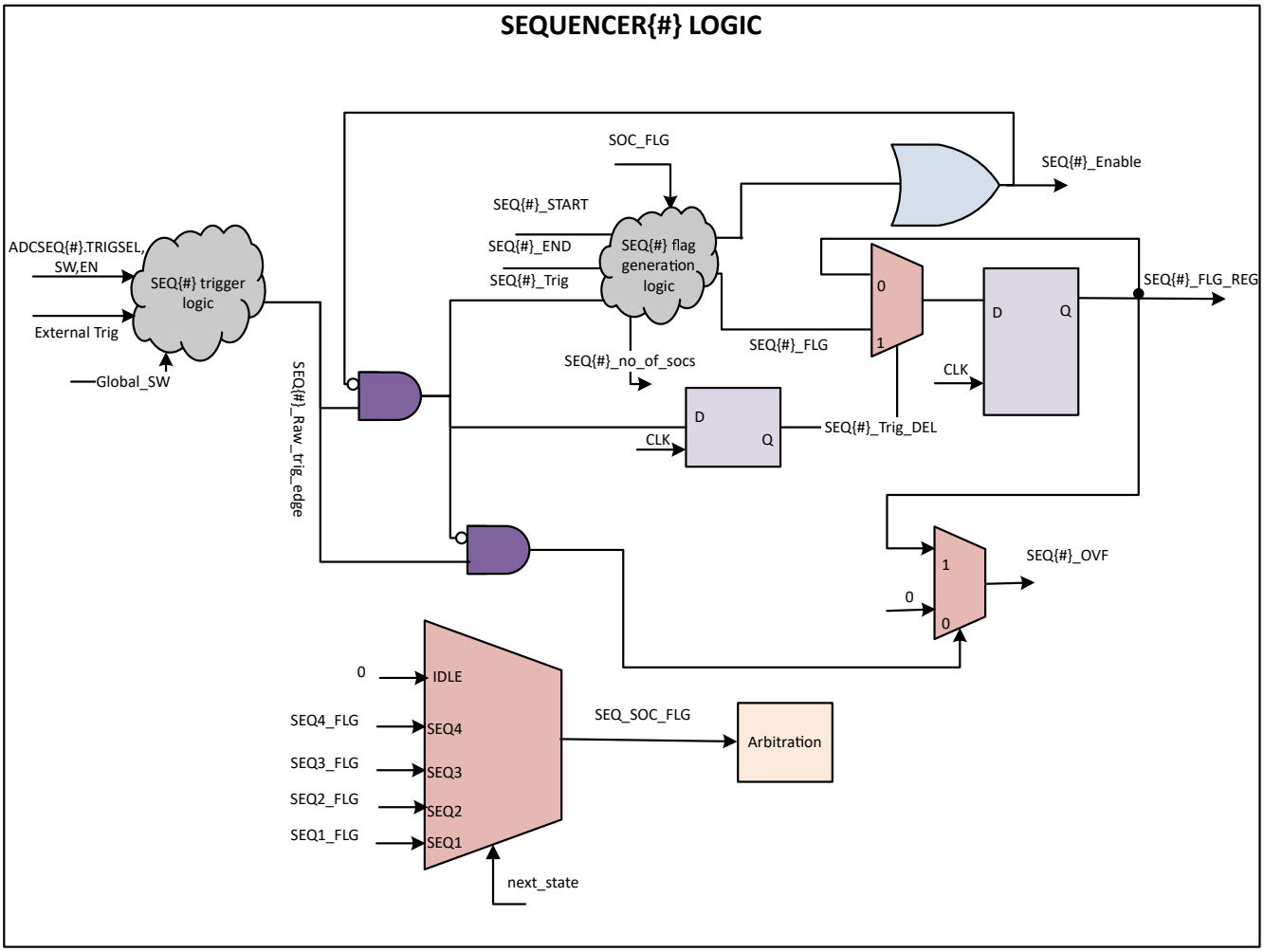


Figure 16-4. Sequencer Enable and Overflow Logic

Figure 16-5 illustrates the logical representation of sequencer{#} restart & discard logic.

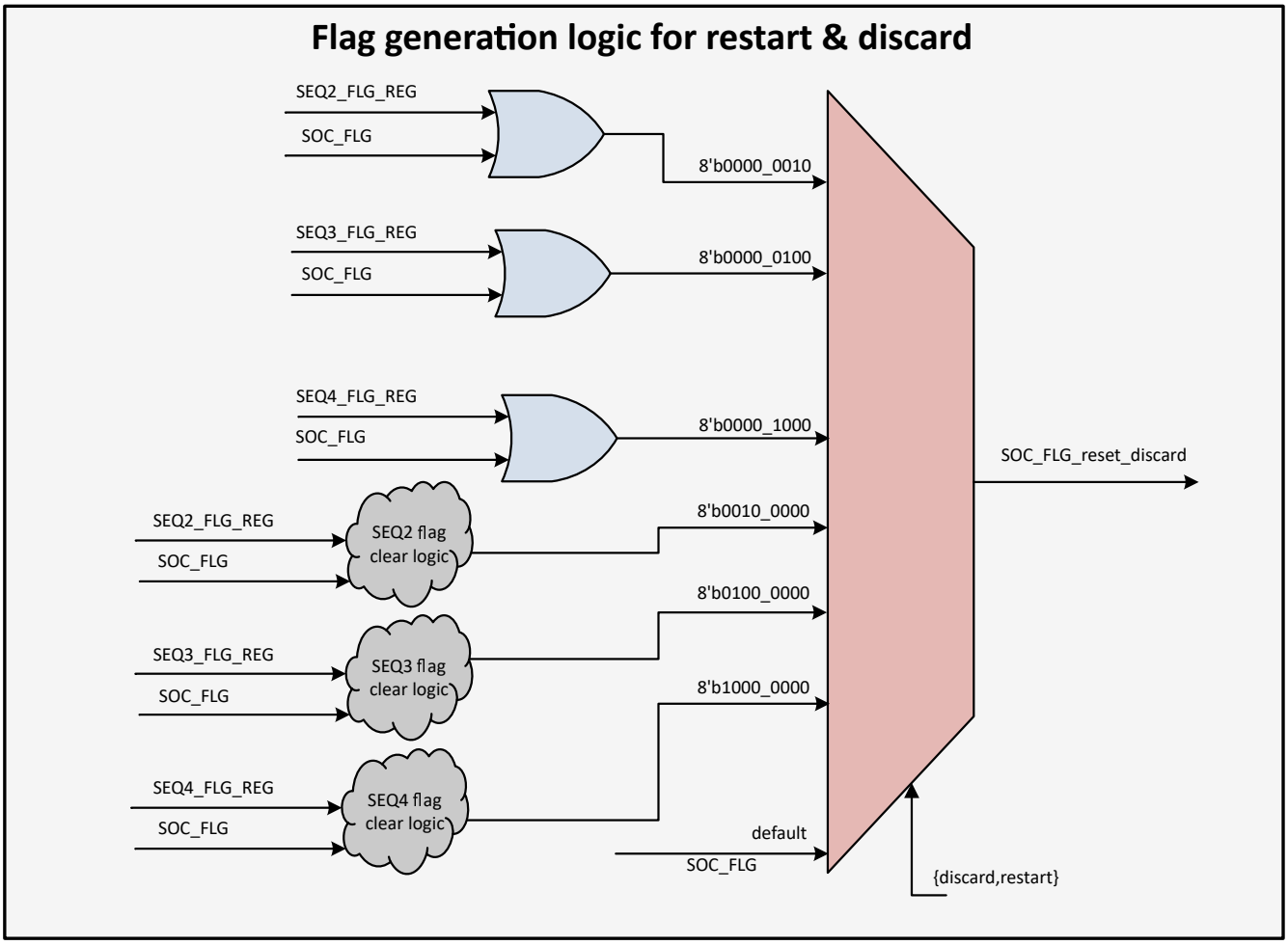


Figure 16-5. Sequencer Pre-empt Restart & Discard Logic

Additionally, oversampling and averaging can be enabled for the SOCs within the sequence using the Post Processing Blocking (PPB) averaging feature. Each sequencer is associated with the corresponding PPB block. PPB{#}LIMIT defines the number of samples per SOCs.

If oversampling is enabled, each SOC is sampled and converted 'N' times (N – total number of samples configured) before moving to the next SOC.

Each sequencer is associated with the corresponding ADCPPB{#}TRIPHI and ADCPPB{#}TRIPLO registers. This can be used to compare whether ADCRESULT is within the configured TRIPHI and TRIPLO limits or exceeded either of the limits.

Since the limit comparator settings are common across all SOCs within a Sequencer, limit comparison for a given SOC can be enabled or disabled using the ADCSOC{#}CTL.COMPEN field of the corresponding SOC.

16.3.2 SOC Configuration

Each SOC has a configuration register, ADCSOCxCTL. Within this register, SOCx can be configured to select the specific channel to convert and enable or disable the threshold comparator.

16.3.3 Trigger Operation

Each SOC can be configured to start on one of many input triggers. The primary trigger select for SOCx is in the ADCSOCxCTL.TRIGSEL register, which can select between:

- Disabled (Software only)
- GPIO: INPUTXBAR.OUT[5]

- ADCSOCA, ADCSOCB, ADCSOCC, or ADCSOCD from each MCPWM module
- The trigger repeater block. This can be used to achieve oversampling and re-trigger spread

In addition, each SOC can also be triggered when the ADCINT1 flag or ADCINT2 flag is set. This is achieved by configuring the ADCINTSOCSEL1 register (for SOC0 to SOC7) or the ADCINTSOCSEL2 register (for SOC8 to SOC15). This can be useful for creating continuous conversions.

Table 16-4. ADC SOC Trigger Selection

ADCSEQxCTL.TRIGSEL	Input Signal	Trigger Type Mapping
0	ADC_SOFTWARE_TRIGGER	Reserved
1	TIMG4_0.GEN_EVENT0	SUBSCRIBER
2	Reserved	Reserved
3	TIMG12_0.GEN_EVENT0	SUBSCRIBER
4	Reserved	Reserved
5	INPUTXBAR5	RAW
6	PWM0_SOCA	RAW
7	PWM0_SOCB	RAW
8	PWM0_SOCC	RAW
9	PWM0_SOCD	RAW
10	PWM1_SOCA	RAW
11	PWM1_SOCB	RAW
12	PWM1_SOCC	RAW
13	PWM1_SOCD	RAW
14	PWM2_SOCA	RAW
15	PWM2_SOCB	RAW
16	PWM2_SOCC	RAW
17	PWM2_SOCD	RAW
18	PWM3_SOCA	RAW
19	PWM3_SOCB	RAW
20	PWM3_SOCC	RAW
21	PWM3_SOCD	RAW
22	PWM4_SOCA	RAW
23	PWM4_SOCB	RAW
24	PWM4_SOCC	RAW
25	PWM4_SOCD	RAW
26	ADC0_SEQ0	RAW
27	ADC1_SEQ0	RAW
28	ADC2_SEQ0	RAW
29-31	Reserved	Reserved

16.3.3.1 Global Software Trigger

This ADC only supports a synchronous global software trigger through use of the INPUTXBAR5 signal. Synchronous global triggers allow the application to trigger SOCs on multiple ADC instances that are exactly simultaneous in time.

16.3.4 ADC Acquisition (Sample and Hold) Window

External signal sources vary in the ability to drive an analog signal quickly and effectively. To achieve rated resolution, the signal source needs to charge the sampling capacitor in the ADC core to within 0.5 LSBs of the signal voltage. The acquisition window is the amount of time the sampling capacitor is allowed to charge and is configurable per SOCx by the ADCSOCxCTL.ACQPSADCSEQxCTONFIG.ACQPS register.

ACQPS is a 9-bit8-bit register field that can be set to a value between 0 and 255.511, resulting in an acquisition window duration of:The 2 upper bits (ADCSOCxCTL.ACQPS[7:6]) configure the base duration and cycle prescalers. The base duration can be set to values of 0, 64, 192, or 448 with corresponding cycle prescalers of 1, 2, 4, and 16 respectively. The lower 6 bits (ADCSOCxCTL.ACQPS[5:0]) in the register field configure the additional cycles to be multiplied by the prescaler and added to the base duration.

Acquisition Window = (ACQPS + 1) × (System Clock (MCLK) cycle time)

Acquisition Window [MCLK cycles] = (Base duration) + ((Additional cycles + 1)*(Prescaler))

- The acquisition window duration is based on the System Clock (MCLK), not the ADC clock (ADCCLK).
- The selected acquisition window duration must be at least as long as one ADCCLK cycle.
- The data sheet specifies a minimum acquisition window duration (in nanoseconds). The user is responsible for selecting an acquisition window duration that meets this requirement.

16.3.5 Sample Capacitor Reset

In certain systems where the ADC successively samples multiple signal sources, memory crosstalk can occur. Memory crosstalk is the tendency of the ADC conversion to be pulled towards the value of the previous conversion, due to inadequate acquisition/settling time. This happens because the ADC sample capacitor voltage starts near the previously converted voltage, then settles towards the newly applied voltage on the current channel. If the acquisition window is not long enough for the sample capacitor to settle, this can result in some sample error reflected in the ADC conversion.

The 12-bit ADC modules in this device include a sample capacitor reset feature to help mitigate memory crosstalk. When sample capacitor reset is enabled, after every conversion, the sampling capacitor voltage is reset to the VREFLOeither the VREFLO or VREFHI/2 voltage, depending on the configuration of the ADCSOCxCTL.SAMPCAPRESETSEL bit. This reset takes an extra ADCCLK cycle to complete. The sample capacitor reset function is inactive by default for each SOC. If desired, the application can activate sample capacitor reset by writing 0 to the SAMPCAPRESETDISABLE bit in the ADCSOCxCTL register. When sample capacitor reset is active, overall ADC throughput is slightly decreased due to the extra ADCCLK cycle in the conversion period.

16.3.6 ADC Input Models

For single-ended operation, the ADC input characteristics for values in the single-ended input model (see Figure 16-6) can be found in the device data sheet.

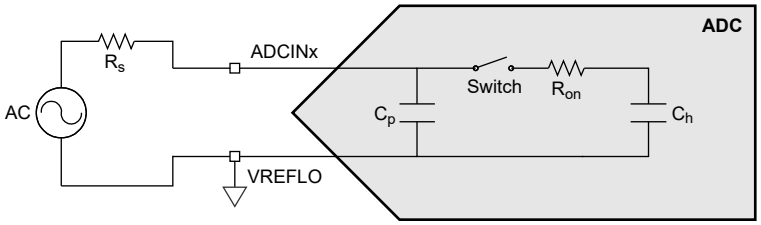


Figure 16-6. Single-Ended Input Model

These input models must be used along with actual signal source impedance to determine the acquisition window duration. See *Choosing an Acquisition Window Duration* for more information.

16.3.7 Channel Selection

Each SOC can be configured to convert any of the ADC channels. This behavior is selected for SOCx by the ADCSOCxCTL.CHSEL register. This is summarized in Table 16-5. For pin location of ADC inputs, refer to the device specific data sheet.

Table 16-5. Channel Selection of Input Pins

Input Mode	CHSEL	Input
Single-Ended	0	ADCIN0
	1	ADCIN1
	2	ADCIN2
	3	ADCIN3
	4	ADCIN4
	5	ADCIN5
	6	ADCIN6
	7	ADCIN7
	8	ADCIN8
	9	ADCIN9
	10	ADCIN10
	11	ADCIN11
	12	ADCIN12
	13	ADCIN13
	14	ADCIN14
	15	ADCIN15

16.4 SOC Configuration Examples

The following sections provide some specific examples of how to configure the SOCs to produce some conversions.

16.4.1 Single Conversion from MCPWM Trigger

To configure ADCA to perform a single conversion on channel ADCIN1 when the MCPWM timer reaches the period match, a few things are necessary. First, MCPWM3 must be configured to generate an SOCA, SOCB, SOCC, or SOCD signal (in this statement, SOC refers to a signal in the MCPWM module). See the *Multi-Channel Pulse Width Modulator Module (MCPWM)* chapter on how to do this. Assume that SOCB was chosen.

SOC5 is chosen arbitrarily. Any of the 16 SOCs can be used.

Assuming a 100ns sample window is desired with a MCLK frequency of 200MHz, then the acquisition window duration must be 100ns/6.25ns = 16 cycles. The ACQPS field must be set to 0 in the upper two bits and 16 - 1 in the lower six bits (ACQPS[7:6] = 0, ACQPS[5:0]=15)..

```
AdcaRegs.ADCSOC5CTL.bit.CHSEL = 1;      //SOC5 converts ADCINA1
              AdcaRegs.ADCSOC5CTL.bit.ACQPS = 15;     //SOC5 uses a sample duration of 16 MCLK
cycles
AdcaRegs.ADCSOC5CTL.bit.TRIGSEL = 14;   //SOC5 begins conversion on PWM3 SOCB
```

As configured, when MCPWM3 matches the period and generates the SOCB signal, the ADC begins sampling channel ADCINA1 (SOC5) immediately if the ADC is idle. If the ADC is busy, ADCINA1 begins sampling when SOC5 gains priority (see Section 16.5). The ADC control logic samples ADCINA1 with the specified acquisition window width of 100ns. Immediately after the acquisition is complete, the ADC begins converting the sampled voltage to a digital value. When the ADC conversion is complete, the results are available in the ADCRESULT5 register (see Section 16.12 for exact sample, conversion, and result latch timings).

16.4.2 Oversampled Conversion from MCPWM Trigger

To configure the ADC to oversample ADCINA1 four times, we use the same configurations as the previous example, but apply them to SOC5, SOC6, SOC7, and SOC8.

As configured, when MCPWM3 matches the period and generates the SOCB signal, the ADC begins sampling channel ADCINA1 (SOC5) immediately if the ADC is idle. If the ADC is busy, ADCINA1 begins sampling when SOC5 gains priority (see Section 16.5). Once the conversion is complete for SOC5, SOC6 begins converting ADCINA1 and the results for SOC5 are placed in the ADCRESULT5 register. All four conversions eventually are completed sequentially, with the results in ADCRESULT5, ADCRESULT6, ADCRESULT7, and ADCRESULT8 for SOC5, SOC6, SOC7, and SOC8, respectively.

Note

Possible, but unlikely, that the ADC can begin converting SOC6, SOC7, or SOC8 before SOC5 depending on the position of the round-robin pointer when the MCPWM trigger is received. See Section 16.5 to understand how the next SOC to be converted is chosen.

16.4.3 Software Triggering of SOCs

At any point, whether or not the SOCs have been configured to accept a specific trigger, a software trigger can set the SOCs to be converted. This is accomplished by writing bits in the ADCSOCFRC1 register.

Software triggering of the previous example without waiting for the CPU1 Timer 2 to generate the trigger can be accomplished by the statement:

```
AdcaRegs.ADCSOCFRC1.all = 0x000F;        //set SOC flags for SOC0 to SOC3
```

16.5 ADC Conversion Priority

When multiple SOC flags are set at the same time, one of two forms of priority determines the converted order. The default priority method is round-robin. In this scheme, no SOC has an inherent higher priority than another. Priority depends on the round-robin pointer (RRPOINTER). The RRPOINTER reflected in the ADCSOCPRIORITYCTL register points to the last SOC converted. The highest priority SOC is given to the next value greater than the RRPOINTER value, wrapping around back to SOC0 after SOC15. At reset the value is 16 since 0 indicates a conversion has already occurred. When RRPOINTER equals 16 the highest priority is given to SOC0. The RRPOINTER is reset when the ADC module is reset or when the reset value is written to the SOCPRICTL register. The ADC module is reset by writing and clearing the SOFTPRES bit corresponding to the ADC instance.

An example of the round-robin priority method is given in Figure 16-7.

The SOCPRIORITY field in the ADCSOCPRIORITYCTL register can be used to assign high priority from a single to all of the SOCs. When configured as high priority, an SOC interrupts the round-robin wheel after any current conversion completes and inserts in as the next conversion. After the conversion completes, the round-robin wheel continues where the conversion was interrupted. If two high priority SOCs are triggered at the same time, the SOC with the lower number takes precedence.

High priority mode is assigned first to SOC0, then in increasing numerical order. The value written in the SOCPRIORITY field defines the first SOC that is not high priority. In other words, if a value of 4 is written into SOCPRIORITY, then SOC0, SOC1, SOC2, and SOC3 are defined as high priority, with SOC0 the highest.

An example using high priority SOC's is given in Figure 16-8.

A After reset, SOC0 is highest priority SOC; SOC7 receives trigger; SOC7 configured channel is converted immediately.

B RRPOINTER changes to point to SOC 7; SOC8 is now highest priority SOC.

C SOC2 & SOC12 triggers rcvd. simultaneously; SOC12 is first on round robin wheel; SOC12 configured channel is converted while SOC2 stays pending.

D RRPOINTER changes to point to SOC 12; SOC2 configured channel is now converted.

E RRPOINTER changes to point to SOC 2; SOC3 is now highest priority SOC.

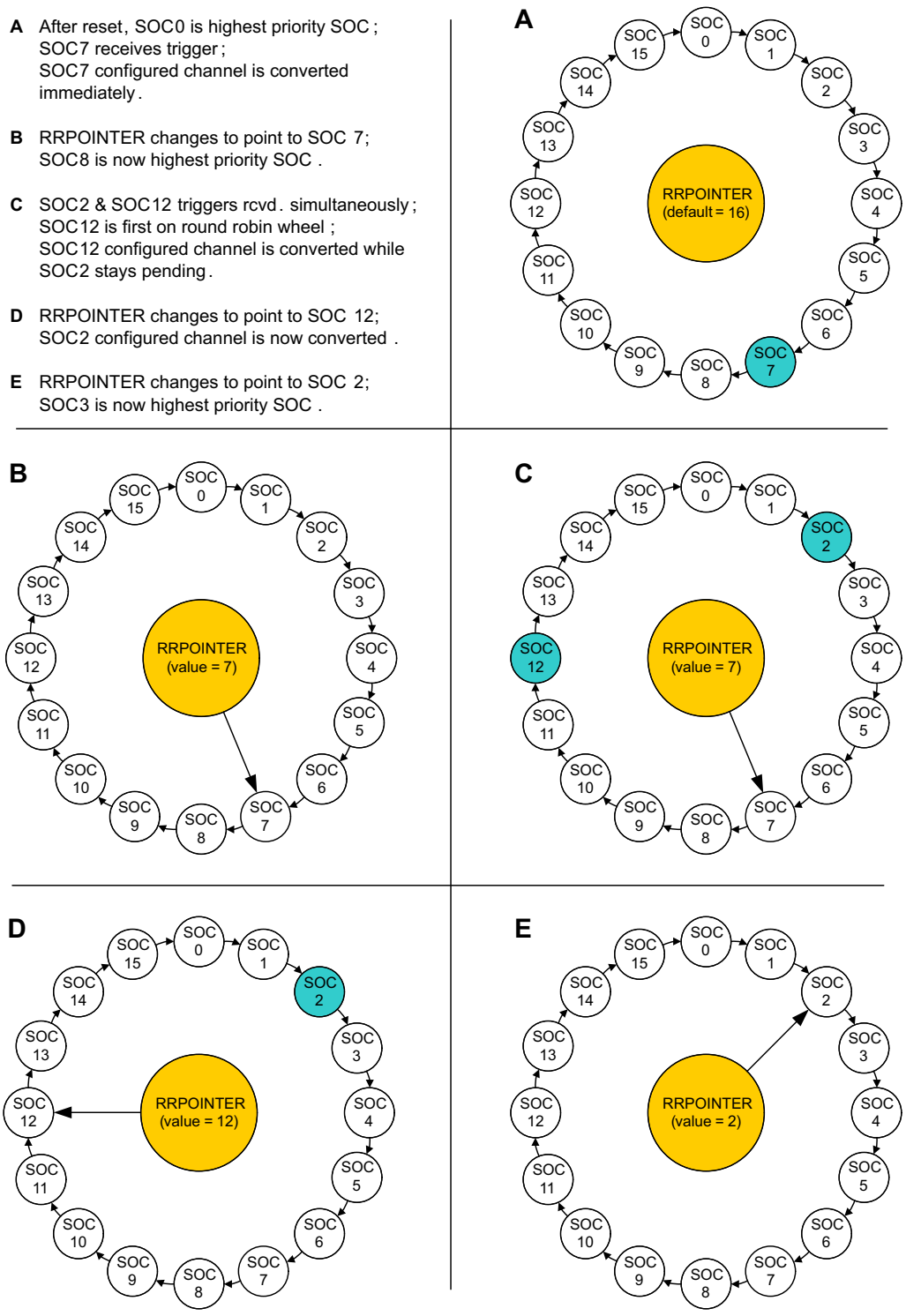


Figure 16-7. Round Robin Priority Example

Example when SOCPRIORITY = 4

A After reset, SOC4 is 1st on round robin wheel ;
SOC7 receives trigger ;
SOC7 configured channel is converted immediately .

B RRPOINTER changes to point to SOC 7;
SOC8 is now 1st on round robin wheel .

C SOC2 & SOC12 triggers rcvd . simultaneously ;
SOC2 interrupts round robin wheel and SOC 2 configured channel is converted while SOC 12 stays pending .

D RRPOINTER stays pointing to 7;
SOC12 configured channel is now converted .

E RRPOINTER changes to point to SOC 12;
SOC13 is now 1st on round robin wheel .

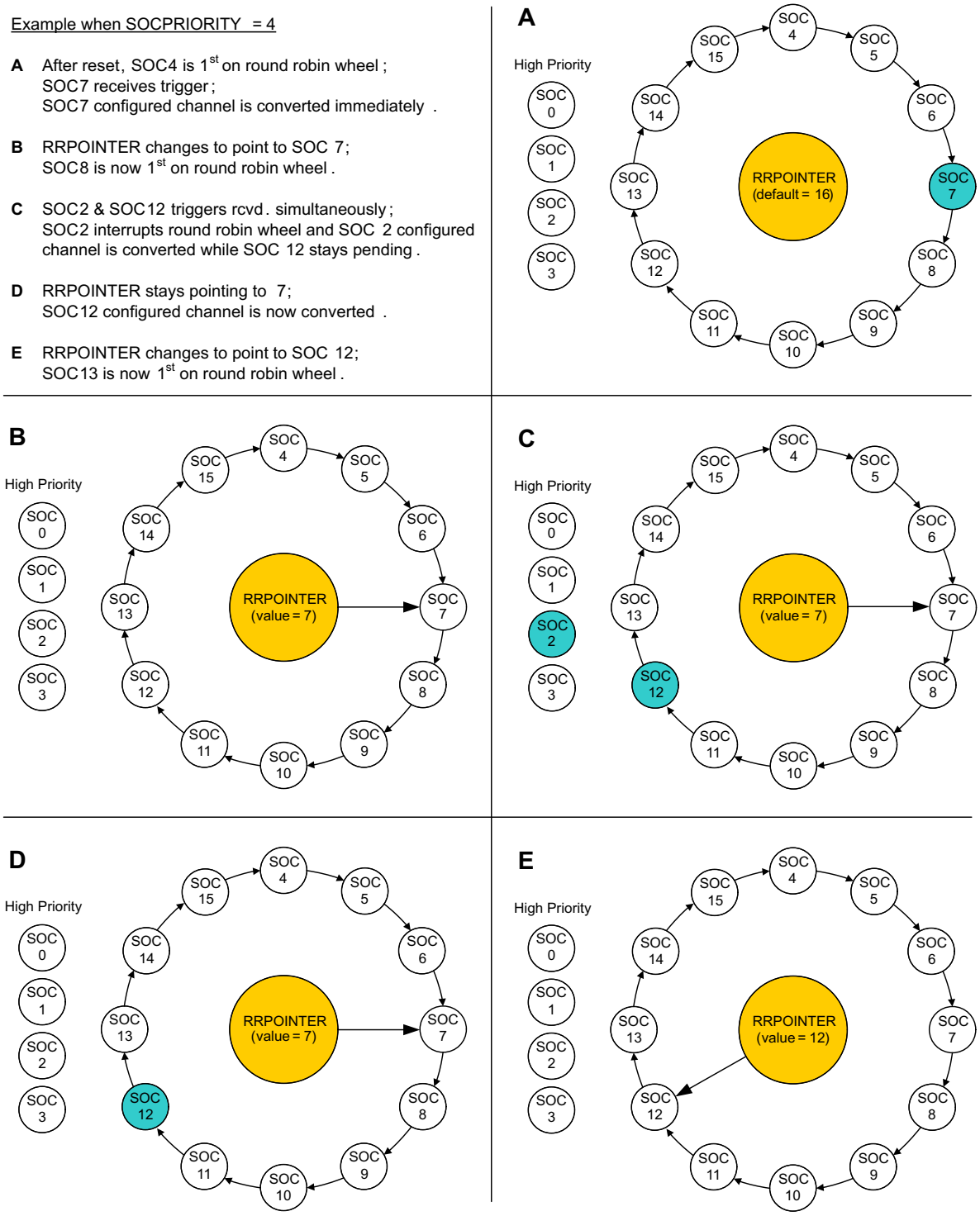


Figure 16-8. High Priority Example

ADVANCE INFORMATION

16.6 Burst Mode

Burst mode allows a single trigger to walk through the round-robin SOCs one or more at a time. Setting the bit BURSTEN in the ADCBURSTCTL register configures the ADC wrapper for burst mode. This causes the TRIGSEL field to be ignored, but only for SOCs that are configured for round-robin operation (not high priority). Instead of the TRIGSEL field, all round-robin SOCs are triggered based on the BURSTTRIG field in the ADCBURSTCTL register. Upon reception of the burst trigger, the ADC wrapper does not set all round-robin SOCs to be converted, but only (ADCBURSTCTL.BURSTSIZE + 1) SOCs. The first SOC to be set is the SOC with the highest priority based on the round-robin pointer, and subsequent SOCs are set until BURSTSIZE SOCs have been set.

Note

When configuring the ADC for burst mode, the user is responsible for ensuring that each burst of conversions is allowed to complete before the next burst trigger is received. The value of (ADCBURSTCTL.BURSTSIZE + 1) must be less than or equal to the number of SOCs configured for round-robin priority.

For example, if SOCPRIORITY = 12, that is, SOC12, SOC13, SOC14, and SOC15 are in round-robin, ADCBURSTCTL.BURSTSIZE setting must be ≤3 for burst mode to operate correctly.

16.6.1 Burst Mode Example

Burst mode can be used to sample a different set of signals on every other trigger. In the following example, ADCIN7 and ADCIN5 are converted on the first trigger from Timer 2 and every other trigger thereafter. ADCIN2 and ACIN3 are converted on the second trigger from Timer 2 and every other trigger thereafter. All signals are converted with 20 MCLK cycle wide acquisition windows, but different durations can be configured for each SOC as desired.

```
AdcaRegs.BURSTCTL.BURSTEN = 1;              //Enable ADC burst mode
AdcaRegs.BURSTCTL.BURSTTRIG = 3;            //CPU1 Timer 2 triggers burst of conversions
AdcaRegs.BURSTCTL.BURSTSIZE = 1;            //conversion bursts are 1 + 1 = 2 conversions long
AdcaRegs.SOCPRICTL.bit.SOCPRIORITY = 12;    //SOC0 to SOC11 are high priority
AdcaRegs.ADCSOC12CTL.bit.CHSEL = 7;         //SOC12 converts ADCINA7
AdcaRegs.ADCSOC12CTL.bit.ACQPS = 19;        //SOC12 uses sample duration of 20 MCLK cycles
AdcaRegs.ADCSOC13CTL.bit.CHSEL = 5;         //SOC13 converts ADCINA5
AdcaRegs.ADCSOC13CTL.bit.ACQPS = 19;        //SOC13 uses sample duration of 20 MCLK cycles
AdcaRegs.ADCSOC14CTL.bit.CHSEL = 2;         //SOC14 converts ADCINA2
AdcaRegs.ADCSOC14CTL.bit.ACQPS = 19;        //SOC14 uses sample duration of 20 MCLK cycles
AdcaRegs.ADCSOC15CTL.bit.CHSEL = 3;         //SOC15 converts ADCINA3
AdcaRegs.ADCSOC15CTL.bit.ACQPS = 19;        //SOC15 uses sample duration of 20 MCLK cycles
```

When the first Timer 2 trigger is received, SOC12 and SOC13 are converted immediately if the ADC is idle. If the ADC is busy, SOC12 and SOC13 are converted once the SOCs gain priority. The results for SOC12 and SOC13 are in ADCRESULT12 and ADCRESULT13, respectively. After SOC13 completes, the round-robin pointer gives the highest priority to SOC14. Because of this, when the next CPU1 Timer 2 trigger is received, SOC14 and SOC15 is set as pending and eventually converted. The results for SOC14 and SOC15 are in ADCRESULT14 and ADCRESULT15, respectively. Subsequent triggers continue to toggle between converting SOC12 and SOC13, and converting SOC14 and SOC15.

While the above example toggles between two sets of conversions, three or more different sets of conversions can be achieved using a similar approach.

16.6.2 Burst Mode Priority Example

An example of priority resolution using burst mode and high-priority SOCs is presented in Figure 16-9.

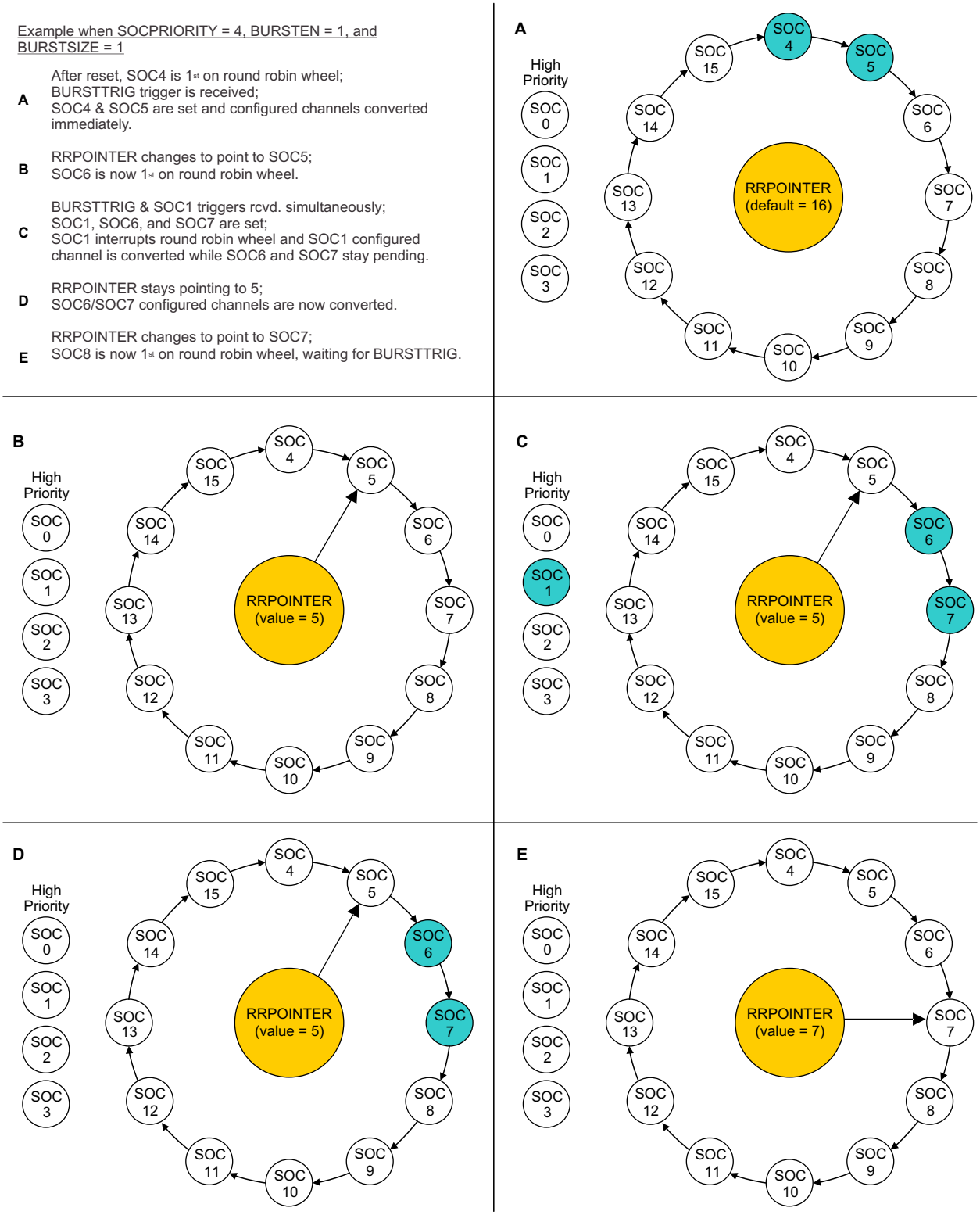


Figure 16-9. Burst Priority Example

16.7 EOC and Interrupt Operation

Each SOC has a corresponding end-of-conversion (EOC) signal. This EOC signal can be used to trigger an ADC interrupt. The ADC can be configured to generate the EOC pulse at either the end of the acquisition window or at the end of the voltage conversion. This is configured using the bit INTPULSEPOS in the ADCCTL1 register. See Section 16.12 for exact EOC pulse location.

The ADC module has 2 configurable ADC interrupts. These interrupts can be triggered by any of the 16 EOC signals. The flag bit for each ADCINT can be read directly to determine if the associated SOC is complete or the interrupt can be passed on to the NVIC.

Note

The ADCCTL1.ADCBSY bit being clear does not indicate that all conversions in a set of SOCs have completed, only that the ADC is ready to process the next conversion. To determine if a sequence of SOCs is complete, link an ADCINT flag to the last SOC in the sequence and monitor that ADCINT flag.

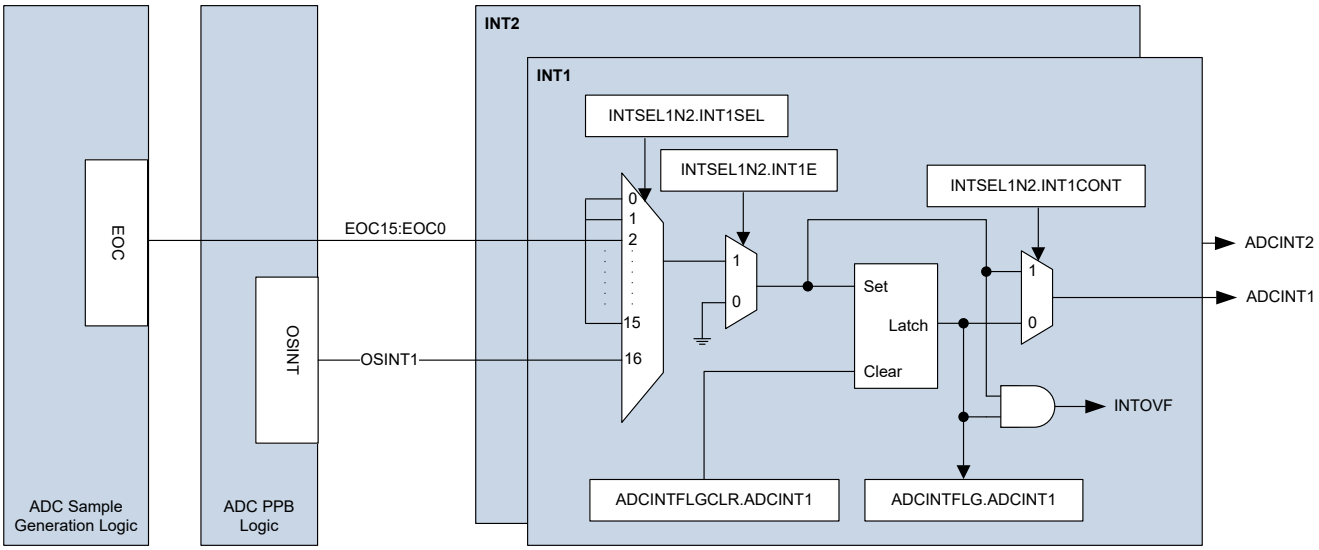


Figure 16-10. ADC End-of-Conversion (EOC) Signal Interrupts

ADVANCE INFORMATION

16.7.1 Interrupt Overflow

If the EOC signal sets a flag in the ADCINTFLG register, but that flag is already set, an interrupt overflow occurs. By default, overflow interrupts are not passed on to the module. When an overflow occurs on a given flag in the ADCINTFLG register, the corresponding flag in the ADCINOVF register is set. This overflow flag is only used to detect that an overflow has occurred; the flag does not block further interrupts from propagating to the module.

When an ADC interrupt overflow occurs, the application must check the appropriate ADCINTOVF flag inside the ISR or in the background loop and take appropriate action when an overflow is detected.

16.7.2 Continue to Interrupt Mode

The INTxCONT bits in the ADCINTSEL1N2 register configures how interrupts are handled when an ADCINTFLG has not yet been cleared from a prior interrupt. This mode is disabled by default and additional overlapping interrupts are not issued to the NVIC. By activating this mode, ADC interrupts always reach the NVIC. If interrupts occur while ADCINTFLG is set, the ADCINTOVF register remains set regardless of the configuration of the INTxCONT bits.

16.7.3 Early Interrupt Configuration Mode

Enabling early interrupt mode can allow the application to enter the ADC interrupt service routine before the ADC results are ready. This allows the application to do any necessary pre-work so that the application can act on the ADC results immediately when the ADC results become available. If the timing of the early interrupt is too early, then the application needs to waste time until the updated ADC results become available. To prevent this situation, the time the ADC interrupt is entered in early interrupt mode is configurable by way of the DELAY field in the ADCINTCYCLE register.

- To use the configurable interrupt time, the ADC must be in early interrupt mode. To achieve this, clear the bit INTPULSEPOS to 0 in ADCCTL1.
- The DELAY value in the ADCINTCYCLE register sets the number of additional MCLK cycles after the falling edge of the SOC pulse before the ADCINT flag is set.
- If the value of DELAY goes beyond EOC, the ADC interrupt is generated along with EOC.
- Writing values to DELAY when INTPULSEPOS is set to 1 does not have any effect on the interrupt generation.

16.8 Post-Processing Blocks

Each ADC module contains four post-processing blocks (PPB). These blocks can be associated with any of the 16 RESULT registers using the ADCPPBxCONFIG.CONFIG bit field. The post-processing blocks have the ability to:

- Remove an offset associated with the ADCIN channel
- Subtract out a reference value
- Flag a zero-crossing point, with the option to trip a PWM and generate an interrupt
- Flag a high or low compare limit, with the option to trip a PWM and generate an interrupt

Figure 16-11 presents the structure of each PPB. Subsequent sections explain the use of each submodule.

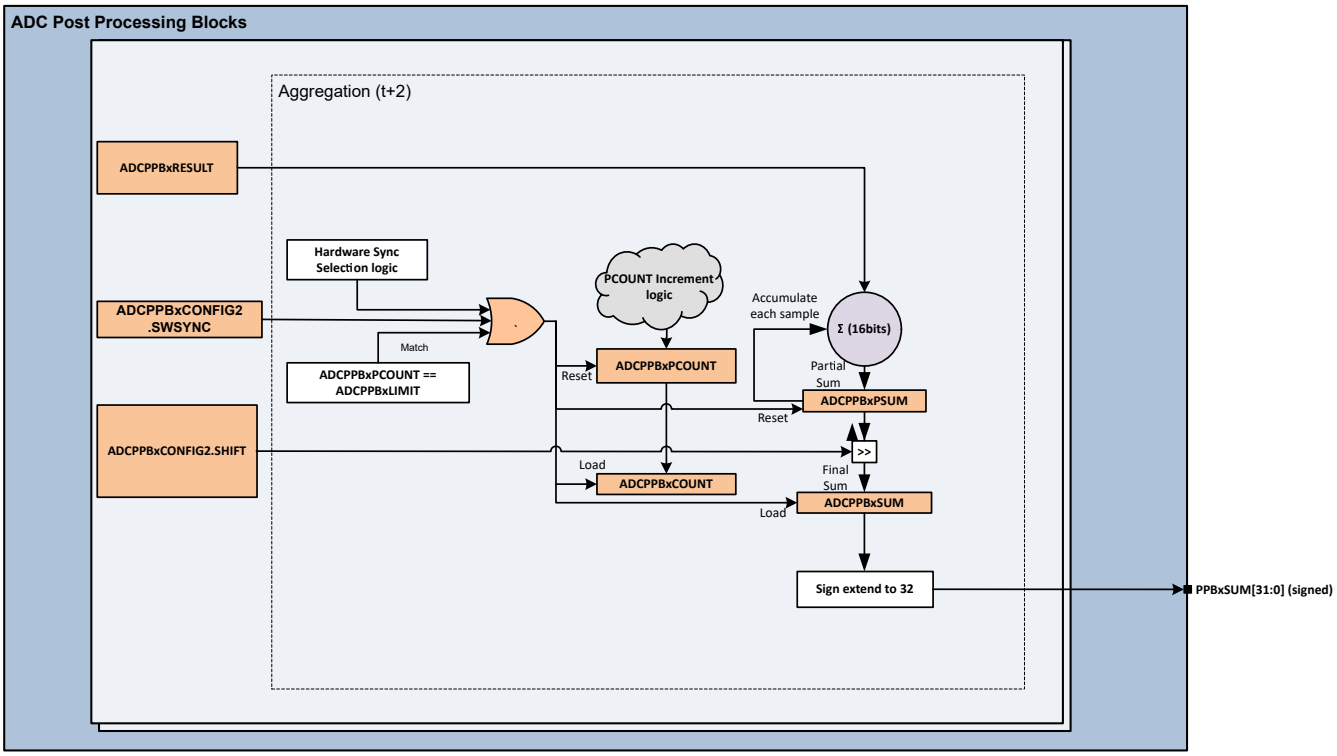


Figure 16-11. ADC Post-Processing Blocks (PPB) Block Diagram

ADVANCE INFORMATION

16.8.1 PPB Offset Correction

In many applications, external sensors and signal sources produce an offset. A global trimming of the ADC offset is not enough to compensate for these offsets, which vary from channel to channel. The post-processing block can remove these offsets with zero overhead, saving numerous cycles in tight control loops.

Offset correction is accomplished by first pointing the ADCPPBxCONFIG.CONFIG to the desired SOC, then writing an offset correction value to the ADCPPBxOFFCAL.OFFCAL register. The post-processing block automatically adds or subtracts the value in the OFFCAL register from the raw conversion result and stores the value in the ADCRESULT register. This addition/subtraction saturates at 0 on the low end and 4095 on the high end.

Note

- Writing a 0 to the OFFCAL register effectively disables the offset correction feature, passing the raw result unchanged to the ADCRESULT register.
- To point multiple PPBs to the same SOC is possible. In this case, the OFFCAL value that is actually applied comes from the PPB with the number.

16.8.2 PPB Error Calculation

In many applications, an error from a set point or expected value must be computed from the digital output of an ADC conversion. In other cases, a bipolar signal is necessary or convenient for control calculations. The PPB can perform these functions automatically, reducing the sample to output latency and reducing software overhead.

Error calculation is accomplished by first pointing the ADCPPBxCONFIG.CONFIG to the desired SOC, then writing a value to the ADCPPBxOFFREF.OFFREF register. The post-processing block automatically subtracts the value in the OFFREF register from the ADCRESULT value and stores the value in the ADCPPBxRESULT register. This subtraction produces a sign-extended 32-bit result. It is also possible to selectively invert the calculated value before storing in the ADCPPBxRESULT register by setting the TWOSCOMPEN bit in the ADCPPBxCONFIG register.

Note

- Do not write a value larger than 12 bits to the ADCPPBxOFFREF register.
- Since the ADCPPBxRESULT register is unique for each PPB, to point multiple PPBs to the same SOC and get different results for each PPB is possible.
- Writing a 0 to the ADCPPBxOFFREF register effectively disables the error calculation feature, passing the ADCRESULT value unchanged to the ADCPPBxRESULT register.

16.8.3 PPB Limit Detection and Zero-Crossing Detection

Many applications perform a limit check against the ADC conversion results. The PPB can automatically perform a check against a high limit, a low limit, high and low limits simultaneously, or whenever ADCPPBxRESULT changes sign. Based on these comparisons, the PPB can generate a trip to the PWM and an interrupt automatically, lowering the sample to MCPWM latency and reducing software overhead. This functionality also enables safety-conscious applications to trip the MCPWM based on an out-of-range ADC conversion without any CPU intervention.

To enable this functionality, first point the ADCPPBxCONFIG.CONFIG to the desired SOC, then write a value to one or both of the registers ADCPPBxTRIPHI.LIMITHI and ADCPPBxTRIPLO.LIMITLO (zero-crossing detection does not require further configuration). Whenever these limits are exceeded, the PPBxTRIPHI bit or PPBxTRIPLO bit is set in the ADCEVTSTAT register. When the PPB result is either in between or equal to the PPBxTRIPHI and PPBxTRIPLO thresholds, the PPBxINLIMIT bit is set. Note that the PPBxZERO bit in the ADCEVTSTAT register is gated by end-of-conversion (EOC), not by the sign change in the ADCPPBxRESULT register. The ADCEVTCLR register has corresponding bits to clear these event flags. The ADCEVTSEL register

has corresponding bits that allow the events to propagate through to the PWM. The ADCEVTINTSEL register has corresponding bits that allow the events to propagate through to the .

One interrupt is shared between all the PPBs for a given ADC module as shown in the following figure.

Note

- If different actions need to be taken for different PPB events from the same ADC module, then the ADCEVTINT ISR has to read the PPB event flags in the ADCEVTSTAT register to determine which event caused the interrupt.
- If different MCPWM trips need to be generated separately for high compare, low compare, in-limit compare, or zero-crossing, this can be achieved by pointing multiple PPBs to the same SOC.
- The zero-crossing detect circuit considers a result of zero to be positive.

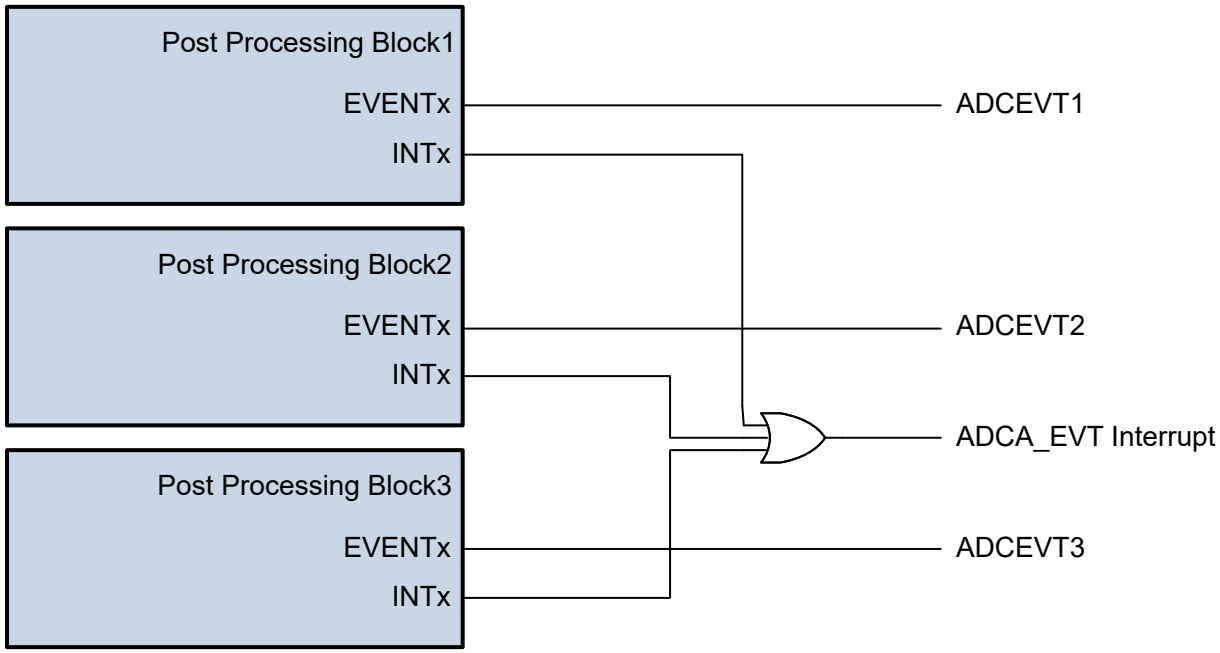


Figure 16-12. ADC Post-Processing Blocks (PPB) Interrupt Event

16.8.4 PPB Sample Delay Capture

When multiple control loops are running asynchronously on the same ADC, there is a chance that an ADC request from two or more loops collide, causing one of the samples to be delayed. This shows up as a measurement error in the system. By knowing when this delay occurs and the amount of delay that has occurred, software can employ extrapolation techniques to reduce the error.

To this effect, each PPB has the field DLYSTAMP in the ADCPPBxSTAMP register. This field contains the number of MCLK cycles between when the associate SOC was triggered and when the SOC began converting.

This is achieved by having a global 12-bit free running counter based off of MCLK, which is in the field FREECOUNT in the ADCCOUNTER register. When the trigger for the associated SOC arrives, the value of this counter is loaded into the bit field ADCPPBxTRIPLO.REQSTAMP. When the actual sample window for that SOC begins, the value in REQSTAMP is subtracted from the current FREECOUNT value and stored in DLYSTAMP.

Note

If more than 4096 MCLK cycles elapse between the SOC trigger and the actual start of the SOC acquisition, the FREECOUNT register can overflow more than once, leading to incorrect DLYSTAMP value. Be cautious when using very slow conversions to prevent this from happening.

The sample delay capture does not function, if the associated SOC is triggered using software. The sample delay capture, however, correctly records the delay, if the software triggering of a different SOC causes the SOC associated with the PPB to be delayed

16.8.5 PPB Oversampling

This ADC has built-in support for oversampling in the first post-processing block(PPB1). The oversampling support module exists at the output of the sample correction module, as shown in Figure 16-11. The oversampling module works by accumulating results in partial registers until either the sample count limit defined in the ADCPPBxLIMIT register is reached , an external hardware sync event occurs, or the software forces a sync event by writing to the SWSYNC bit in the ADCPPBxCONFIG2 register.

16.8.5.1 Accumulation and Average Functions

At the end of each ADC sample conversion, the PPB updates the partial result registers ADCPPBxPSUM with the newly processed conversion result from the ADCPPBxRESULT register, and the partial conversion count register (ADCPPBxPCOUNT) increments by 1. When the partial conversion count equals the limit defined in ADCPPBxLIMIT, or the PPB receives a software sync signal, the PPB takes the following actions:

1. The PPB loads the values of the respective partial result registers into the final result registers ADCPPBxPSUM.
2. The PPB loads the partial count in ADCPPBxPCOUNT into the final conversion count register ADCPPBxCOUNT.
3. The partial count register and partial result registers reset to zero.
4. The ADC generates an oversampling interrupt (OSINTx) event pulse, which triggers a CPU interrupt if so configured in the ADCINTSEL1N2 or ADCINTSEL3N4 registers.

The PPB can also be configured to generate an oversampling interrupt when there is a hardware or software sync event. To trigger an OSINTx pulse when a sync event occurs, write 1 to the OSINTSEL bit in the ADCPPBxCONFIG2 register.

The PPB can automatically compute the average of the accumulated samples if ADCPPBxLIMIT is set to a power of 2 (up to a maximum of 1024 samples). To perform automatic averaging over 2^n samples, set the SHIFT field in the ADCPPBxCONFIG2 register to n. When this field is set, the PPB divides the value of ADCPPBxPSUM by 2^n before loading into ADCPPBxSUM.

To compute an average from the accumulated sum when the number of samples is not a power of 2, divide the value of ADCPPBxSUM by the value of ADCPPBxCOUNT using the CPU.

Note

When using a sync signal to the repeater module and post-processing block to reset the ADC, note that the repeater sync signal does not stop or abort any pending SOCs. If both sync signals are issued simultaneously, any additional pending SOCs can propagate through the post-processing block after the sync signal has been issued. To fully clear or reset the ADC when using the repeater and PPB accumulation logic together:

1. Disable the repeater module trigger source.
2. Reset the trigger repeater by issuing a sync signal to the repeater module.
3. Wait for any pending SOCs to complete.
4. Finally, issue a sync signal to the post-processing block to complete the ADC reset.

16.8.5.2 Outlier Rejection

The post-processing block enables the application to easily perform outlier rejection, by eliminating the largest and smallest samples during each SOC burst. To eliminate outliers, the following formula can be used in a software routine or ISR:

$$\text{Average} = \frac{(\text{ADCPPBxSUM} - \text{ADCPPBxMAX} - \text{ADCPPBxMIN})}{(\text{ADCPPBxCOUNT} - 2)} \quad (14)$$

16.9 Opens/Shorts Detection Circuit (OSDETECT)

The opens/shorts detection circuit (OSDETECT) can be used to detect pin faults in the system. The circuit connects to the ADC input after the channel select multiplexer but before the S+H circuit as shown in Figure 16-13.

Note

- The divider resistance tolerances can vary widely; hence, this feature must not be used to check for conversion accuracy.
- See the data sheet for implementation and availability of analog input channels.
- Due to high drive impedance, a S+H duration much longer than the ADC minimum is needed.

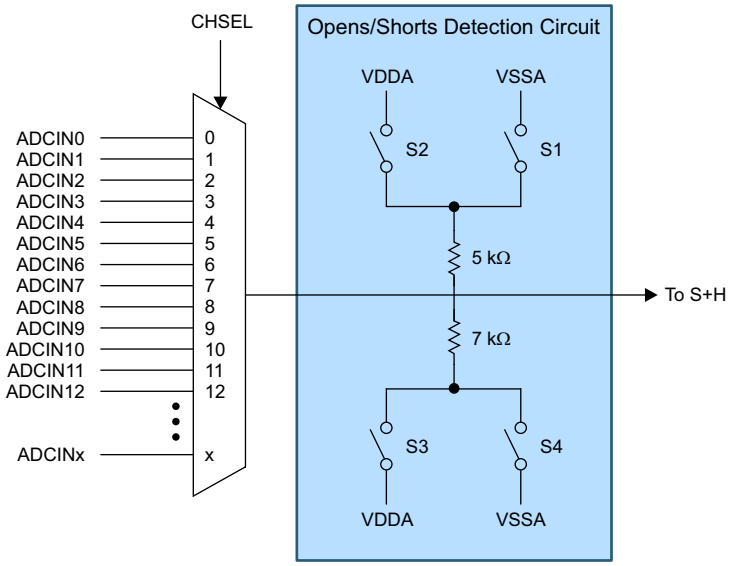


Figure 16-13. Opens/Shorts Detection Circuit

The circuit can be operated by writing a value to the DETECTCFG field in the ADCOSDETECT register. This causes the circuit to source a voltage onto the input during the S+H phase of any conversion. The voltage and drive strength of the OSDETECT circuit for different DETECTCFG settings is given in Table 16-6.

Table 16-6. DETECTCFG Settings

ADCOSDETECT. DETECTCFG	Source Voltage	S4	S3	S2	S1	Drive Impedance
0	Off	Open	Open	Open	Open	Open
1	Zero Scale	Closed	Open	Open	Closed	5K || 7K
2	Full Scale	Open	Closed	Closed	Open	5K || 7K
3	5/12 VDDA	Open	Closed	Open	Closed	5K || 7K
4	7/12 VDDA	Closed	Open	Closed	Open	5K || 7K
5	Zero Scale	Open	Open	Open	Closed	5K
6	Full Scale	Open	Open	Closed	Open	5K
7	Zero Scale	Closed	Open	Open	Open	7K

16.9.1 Open Short Detection Implementation

A representative circuit with the Open-short Detection (OSDETECT) implementation consists of the signal source with series resistance R_S, shunt capacitor C_P, the equivalent OSDETECT resistance $R_{OSDETECT}$ and

ADVANCE INFORMATION

voltage $V_{OSDETECT}$ is shown in Figure 16-14 and can be used as a basis to calculate the signal level going in to the sampling capacitor. $R_{OSDETECT}$ and $V_{OSDETECT}$ are the equivalent input resistance and voltage source contributed by the OSDETECT circuit with values shown in Table 16-6 for the different configuration settings. Refer to Figure 16-14 when deriving the input signal to S/H if signal source V_S is driving while the OSDETECT feature is enabled.

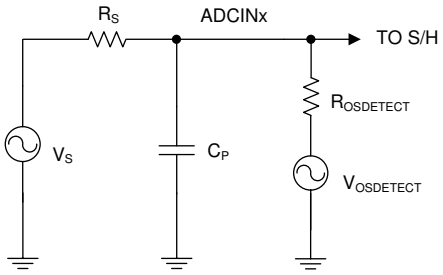


Figure 16-14. Input Circuit Equivalent with OSDETECT Enabled

The input impedance R_S and C_P are integral parts of the signal source or can have been implemented in the design to precondition the signal or to control signal settling time to meet S/H requirements. The input path has to be considered when using the OSDETECT feature, as this affects the conversion results. For instance, driving an input signal when this feature is enabled connects signal V_S to the OSDETECT circuit through R_S and affects the ADC results. Larger C_P values (in the order greater than hundreds of pF) require using higher ACQPS to make sure the signal at the input has settled prior to conversion.

To enable the circuit:

1. Configure the ADC for conversion (for example, channel, SOC, ACQPS, prescaler, trigger, and so on).
2. Set up the ADCOSDETECT register for the desired voltage divider connection as shown in Table 16-6.
3. Initiate a conversion and inspect the conversion result.

Interpret the results based on what is driving on the input side and what are the values of R_S and C_P. If the V_S signal can be disconnected from the input pin, the circuit can be used to detect open and shorted input pins as described in the following sections.

16.9.2 Detecting an Open Input Pin

By cycling through the various OSDETECT settings, the input signal is pulled towards the sourced voltages. An input with good drive strength (pin not open) is minimally affected. However, if the pin is open, the sampled voltages is close to the source voltages specified in Table 16-6.

16.9.3 Detecting a Shorted Input Pin

By cycling through the various OSDETECT settings, the input signal is pulled towards the sourced voltages. An input with finite drive strength (pin not shorted) is pulled toward each sourced voltage. However, if the pin is shorted, the signal remains at the same voltage.

16.10 Power-Up Sequence

Upon device power-up or system level reset, the ADC is powered down and disabled. When powering up the ADC, the following sequence must be used:

1. Set the desired ADC clock divider in the PRESCALE field of ADCCTL2.
2. Power up the ADC by setting the ADCPWDNZ bit in ADCCTL1.
3. Allow a delay before sampling. See the data sheet for the necessary time.

Only one delay is necessary as long as the delay occurs after the ADC has begun powering up.

16.11 ADC Calibration

During the fabrication and test process, Texas Instruments calibrates the gain, offset, and linearity of the ADCs. These trim settings are stored in TI reserved OTP memory, and can be loaded using C-callable functions.

- The Device_cal() function copies the trim values for ADC from OTP memory to the respective trim registers.

Until the appropriate factory trim is loaded, the ADC and other analog modules are not specified to operate within the data sheet specifications. Similarly, if trim values other than the factory settings are placed into the trim registers, the ADC (and other modules) is not specified to operate within the data sheet specifications.

The boot ROM calls the calibration functions, so trim values are initially populated without user intervention. However, if the trims are cleared due to a module reset or modified for some other reason, then the user must call the calibration functions (defined in the software development kit header files).

16.12 ADC Timings

The process of converting an analog voltage to a digital value is broken down into an S+H phase and a conversion phase. The ADC sample and hold circuits (S+H) are clocked by MCLK while the ADC conversion process is clocked by ADCCLK. ADCCLK is generated by dividing down MCLK based on the PRESCALE field in the ADCCTL2 register.

The S+H duration is the value of the ACQPS field of the SOC being converted, plus one, times the MCLK period. The user must make sure that this duration exceeds both 1 ADCCLK period and the minimum S+H duration specified in the data sheet. The conversion time is approximately 10.5 ADCCLK cycles. See the timing diagrams and tables in Section 16.12.1 for exact timings.

16.12.1 ADC Timing Diagrams

The following diagrams show the ADC conversion timings for two SOCs given the following assumptions:

- SOC0 and SOC1 are configured to use the same trigger.
- No other SOCs are converting or pending when the trigger occurs.
- The round robin pointer is in a state that causes SOC0 to convert first.
- ADCINTSEL is configured to set an ADCINT flag upon end of conversion for SOC0 (whether this flag propagates through to the CPU to cause an interrupt is determined by the configurations in the NVIC module).

Table 16-8 and Table 16-9 lists the ADC timings.

Table 16-7. ADC Timing Parameter Descriptions

Parameter	Description
t_{SH}	The duration of the S+H window. At the end of this window, the value on the S+H capacitor becomes the voltage to be converted into a digital value. The duration is given by (ACQPS + 1) MCLK cycles. ACQPS can be configured individually for each SOC, so t_{SH} is not necessarily the same for different SOCs. **Note:** The value on the S+H capacitor is captured approximately 5ns before the end of the S+H window regardless of device clock settings.
t_{LAT}	The time from the end of the S+H window until the ADC results latch in the ADCRESULTx register. If the ADCRESULTx register is read before this time, the previous conversion results are returned.

ADVANCE INFORMATION

Table 16-7. ADC Timing Parameter Descriptions (continued)

Parameter	Description
t_{EOC}	The time from the end of the S+H window until the S+H window for the next ADC conversion can begin. The subsequent sample can start before the conversion results are latched.
t_{INT}	The time from the end of the S+H window until an ADCINT flag is set (if configured). If the INTPULSEPOS bit in the ADCCTL1 register is set, t_{INT} coincides with the end of conversion (EOC) signal. If the INTPULSEPOS bit is 0, and the OFFSET field in the ADCINTCYCLE register is not 0, then there is a delay of OFFSET MCLK cycles before the ADCINT flag is set. This delay can be used to enter the ISR when the sample is ready. If the INTPULSEPOS bit is 0, t_{INT} coincides with the end of the S+H window. If t_{INT} triggers a read of the ADC result register (by triggering an ISR that reads the result), care must be taken to make sure the read occurs after the results latch (otherwise, the previous results are read).
t_{DMA}	The time from the end of the S+H window until a DMA read of the ADC conversion result is triggered.

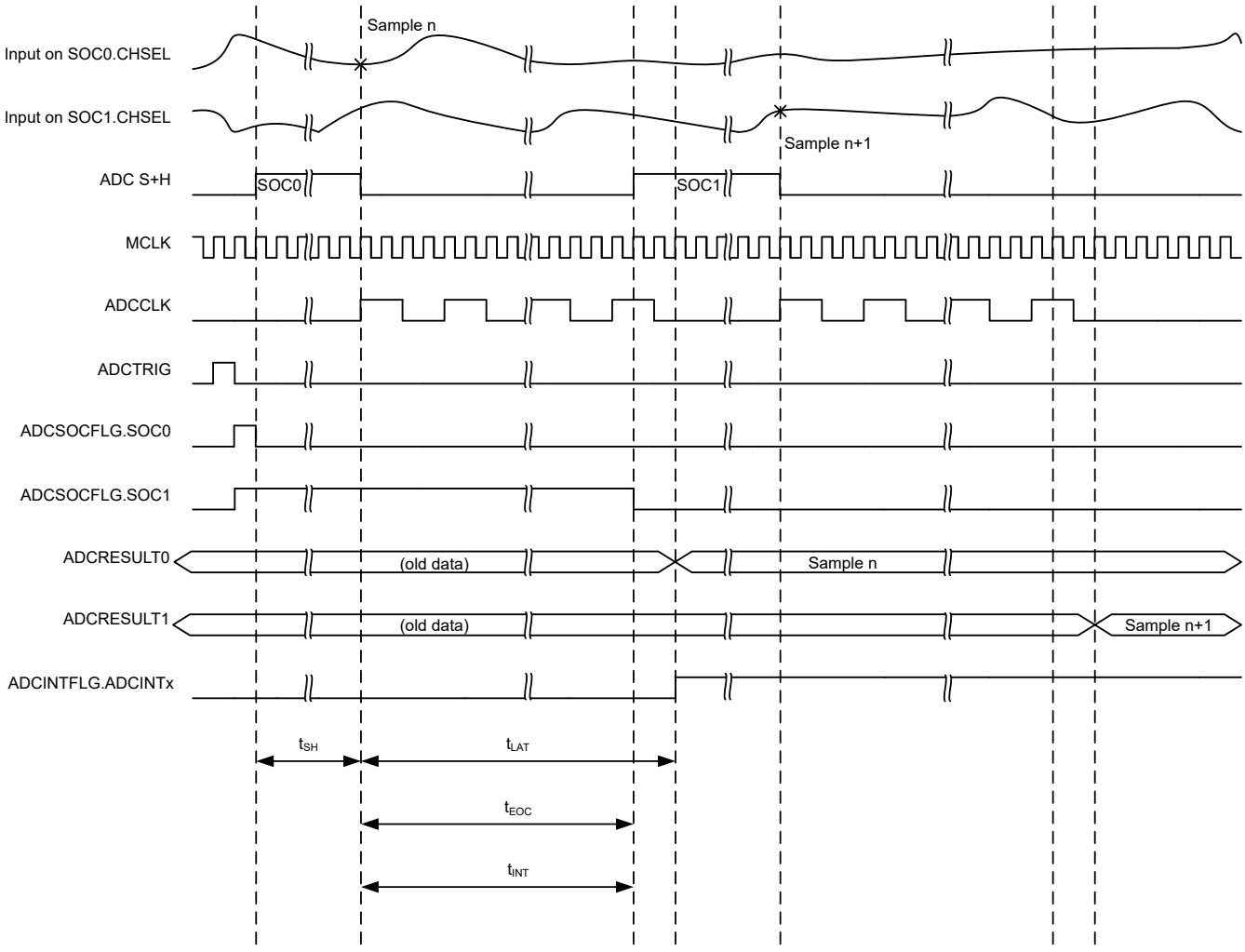


Figure 16-15. ADC Timings for 12-bit Mode in Early Interrupt Mode

ADVANCE INFORMATION

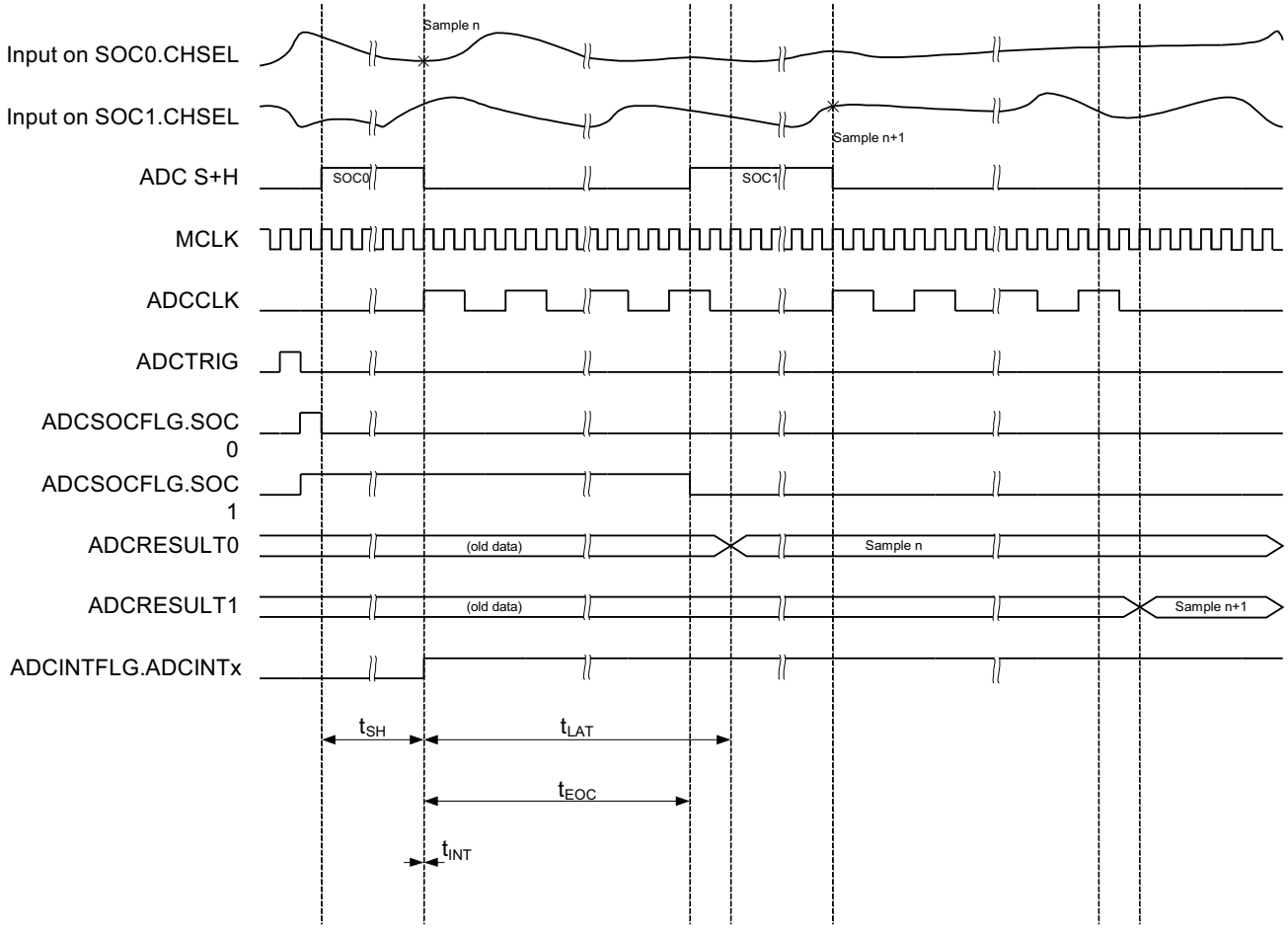


Figure 16-16. ADC Timings for 12-bit Mode in Late Interrupt Mode

ADVANCE INFORMATION

Table 16-8. ADC Timings in 12-bit Mode with SAMPCAPRESETSEL = 0

ADCCLK Prescale		MCLK Cycles				
ADCCTL2. PRESCALE	Prescale Ratio	t_{EOC}	t_{LAT}	t_{INT} (Early)(1)	t_{INT} (Late)	t_{DMA}
0	1	15	20	1	15	20
2	2	30	35	1	30	35
3	2.5	38	46	1	38	46
4	3	45	50	1	45	50
5	3.5	53	58	1	53	58
6	4	60	65	1	60	65
7	4.5	68	73	1	68	73
8	5	75	80	1	75	80
9	5.5	83	88	1	83	88
10	6	90	95	1	90	95
11	6.5	98	103	1	98	103
12	7	105	110	1	105	110
13	7.5	113	118	1	113	118
14	8	120	125	1	120	125
15	8.5	128	133	1	128	133

(1) By default, t_{INT} occurs one MCLK cycle after the S+H window, if INTPULSEPOS is 0. This can be changed by writing to the OFFSET field in the ADCINTCYCLE register.

Table 16-9. ADC Timings in 12-bit Mode with SAMPCAPRESETSEL = 1

ADCCLK Prescale		MCLK Cycles				
ADCCTL2. PRESCALE	Prescale Ratio	t_{EOC}	t_{LAT}	t_{INT} (Early)(1)	t_{INT} (Late)	t_{DMA}
0	1	14	19	1	14	19
2	2	28	33	1	28	33
3	2.5	35	40	1	35	40
4	3	42	47	1	42	47
5	3.5	49	54	1	49	54
6	4	56	61	1	56	61
7	4.5	63	68	1	63	68
8	5	70	75	1	70	75
9	5.5	77	82	1	77	82
10	6	84	89	1	84	89
11	6.5	91	96	1	91	96
12	7	98	103	1	98	103
13	7.5	105	110	1	105	110
14	8	112	117	1	112	117
15	8.5	119	124	1	119	124

(1) By default, t_{INT} occurs one MCLK cycle after the S+H window, if INTPULSEPOS is 0. This can be changed by writing to the OFFSET field in the ADCINTCYCLE register.

ADVANCE INFORMATION

16.12.2 Post-Processing Block Timings

The value of ADCRESULT is always available at time t_{LAT}, as specified in Section 16.12.1 . The value of ADCPPBxRESULT, and the limit and zero-crossing comparisons are available 1 MCLK cycle later, as long as multiple PPB instances do not point to the same SOC. Table 16-10 shows PPB result availability timings when there are no SOCs shared between multiple PPBs. In cases where multiple PPBs point to the same SOC, PPB results become available sequentially, starting with the lowest numbered PPB. The first ADCPPBxRESULT becomes available 1 MCLK cycle after t_{LAT}, and each subsequent ADCPPBxRESULT becomes available 1 MCLK cycle after the previous PPB has completed the limit, in-line, and zero-crossing comparison. The serialized PPB results are therefore spaced 2 or 3 MCLK cycles apart, depending on whether the comparison uses ADCPPBxRESULT or PSUM/SUM respectively. This timing is as shown in Table 16-11. PPB aggregation values (PSUM, SUM, PCOUNT, COUNT) are available 1 cycle after the associated ADCPPBxRESULT becomes available.

Table 16-10. PPB Result Timings (One PPB per SOC)

Register	Description	Results Available
ADCRESULTy	ADC result	t_{LAT}
ADCPPBxRESULT	PPB result	t_{LAT} + 1 MCLK
ADCPPBxPSUM	Oversampling partial sum	t_{LAT} + 2 MCLK
ADCPPBxSUM	Oversampling sum	t_{LAT} + 2 MCLK
ADCPPBxPCOUNT	Oversampling partial sample count	t_{LAT} + 2 MCLK
ADCPPBxCOUNT	Oversampling sample count	t_{LAT} + 2 MCLK
Comparison	Limit and zero-crossing comparison (if using PPBxRESULT)	t_{LAT} + 2 MCLK
Comparison	Limit and zero-crossing comparison (if using PSUM or SUM)	t_{LAT} + 3 MCLK

Table 16-11. PPB Result Timings (Multiple PPBs Configured to Same SOC)

Register	Description	Results Available
ADCRESULTy	ADC result	t_{LAT}
ADCPPBxRESULT	PPB result	Varies (PPBs process serially)
ADCPPBxPSUM	Oversampling partial sum	ADCPPBxRESULT+ 1 MCLK
ADCPPBxSUM	Oversampling sum	ADCPPBxRESULT+ 1 MCLK
ADCPPBxPSUM	Oversampling partial sum	ADCPPBxRESULT+ 1 MCLK
ADCPPBxSUM	Oversampling sum	ADCPPBxRESULT+ 1 MCLK
Comparison	Limit and zero-crossing comparison (if using PPBxRESULT)	t_{LAT} + 2 MCLK
Comparison	Limit and zero-crossing comparison (if using PSUM or SUM)	t_{LAT} + 3 MCLK

16.13 Additional Information

The following sections contain additional practical information.

16.13.1 Ensuring Synchronous Operation

For best performance, all ADCs on the device must be operated synchronously. The device data sheet specifies the performance in both synchronous and asynchronous mode for those parameters which differ between the modes of operation.

To make sure of synchronous operation, all ADCs on the device must operate in lockstep. This is accomplished by writing configurations to all ADCs that cause the sampling and conversion phases of all ADCs to be exactly aligned. The easiest way to accomplish this is to write identical values to the SOC configurations for each ADC for trigger select and ACQPS (S+H duration). In addition, synchronous ADCs must also configure identical values for the SOC priority control, burst mode, burst trigger, and burst size.

On some products, ADC types can be combined on the device for versatility. Synchronous operation can be restricted when operating different ADC types simultaneously. Consult the device data sheet, if there are restrictions on simultaneous operation of ADCs of different types to understand if ADC performance is impacted.

16.13.1.1 Basic Synchronous Operation

The following example configures two SOCs each on ADCA and with identical trigger select and ACQPS values. This results in synchronous operation between ADCA and . For devices with more than two ADCs, the same principles can be used to synchronize all the ADCs.

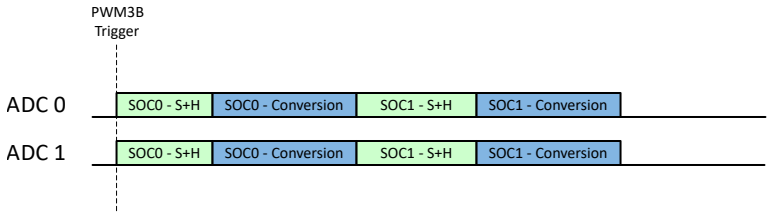


Figure 16-17. Example: Basic Synchronous Operation

Several things can be noted from Figure 16-17. First, while the ACQPS values must be the same for SOCs with the same number, different ACQPS values can be used for SOCs with different numbers. Because of this, synchronous operation does not require a single global S+H time, but instead only channels sampled simultaneously require identical S+H durations. Another important point from this example is that any channel select value can be used for any SOC. Finally, this example assumes round-robin operation. If high-priority SOCs are to be used, the priority must be configured the same on all ADCs.

16.13.1.2 Synchronous Operation with Multiple Trigger Sources

As long as each set of SOCs has identical trigger select and ACQPS settings, multiple trigger sources can be used while still achieving synchronous operation.

The following example demonstrates synchronous operation between ADCA and ADCB while using three SOCs and two trigger sources. Figure 16-18 demonstrates that any combination of relative trigger timings still results in synchronous operation.

Note

In the diagram below:

- ADCA and ADCB can be any two ADC[0:x]
- CPU Timer can be any RTI[0:x]
- PWM3 SOCB can be any PWM[0:x]_SOCB

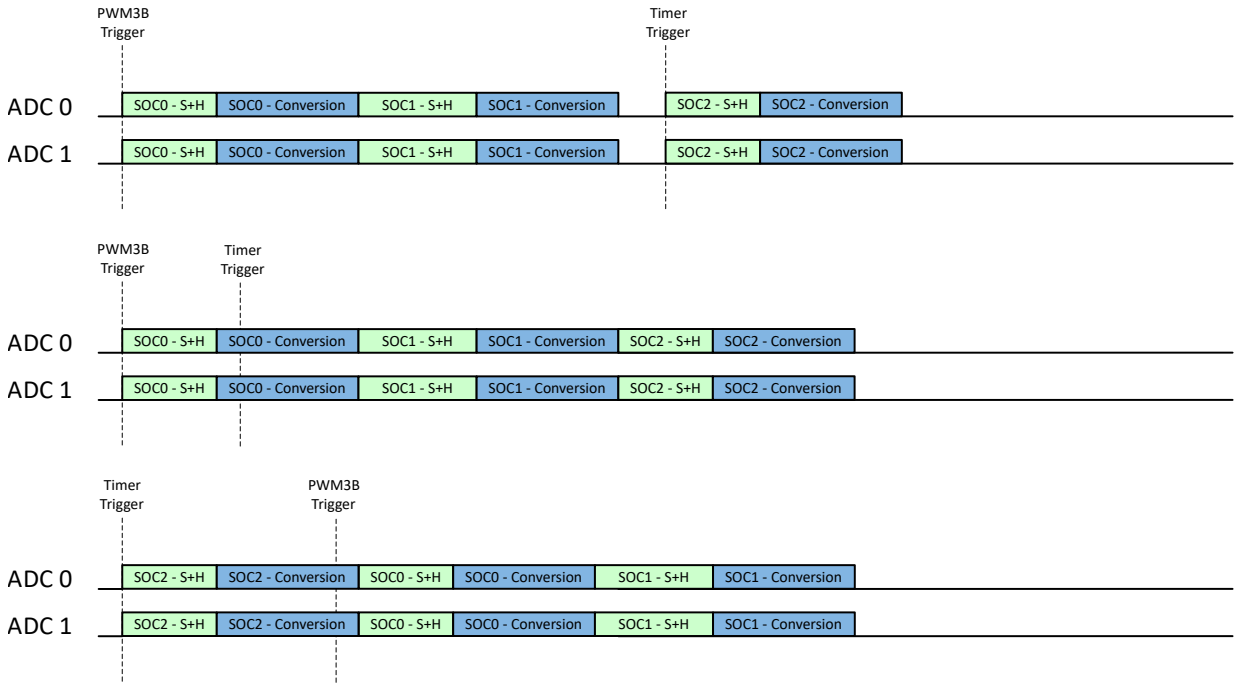


Figure 16-18. Example: Synchronous Operation with Multiple Trigger Sources

Note that any trigger source that can be selected in the TRIGSEL field can be used except for software triggering. There is no way to issue the software triggers for all ADCs simultaneously, so this results in asynchronous operation. ADCINT1 or ADCINT2 can also be used as a trigger when the ADCINTSOCSEL1 and ADCINTSOCSEL2 registers are configured identically for all ADCs and software triggering is not used to start the chain of conversions.

16.13.1.3 Synchronous Operation with Uneven SOC Numbers

If only one trigger source is used, one ADC can use more SOCs than the other ADCs while still operating synchronously.

Note

In the diagrams below:

- ADCA and ADCB can be any two ADC[0:x]
- CPU Timer can be any RTI[0:x]
- PWM3 SOCB can be any PWM[0:x]_SOCB

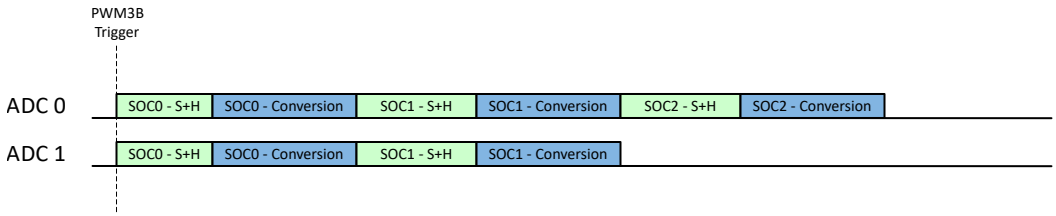


Figure 16-19. Example: Synchronous Operation with Uneven SOC Numbers

Note that if the trigger comes again before all SOCs have completed the conversions, ADCB begins converting immediately on SOC0 while ADCA does not start converting SOC0 again until SOC2 is complete. This results in asynchronous operation, so care must be taken to not overflow the trigger.

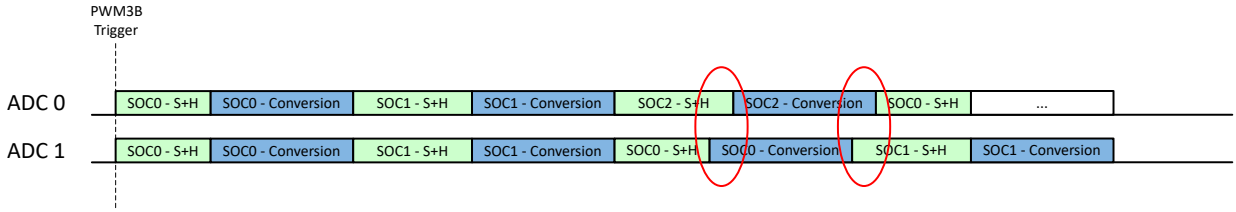


Figure 16-20. Example: Asynchronous Operation with Uneven SOC Numbers – Trigger Overflow

ADVANCE INFORMATION

16.13.1.4 Non-overlapping Conversions

If conversion timings can be made sure to not overlap by the user, then all the SOCs on all ADCs do not have to be identically configured to achieve performance equivalent to synchronous operation. For example, if the two ADC triggers in a system come from two PWM sources that are always 180-degrees out-of-phase, then SOC0 can be used for both ADCA and ADCB with different trigger sources and different ACQPS values.

Note

In the diagram below:

- ADCA and ADCB can be any two ADC[0:x]
- CPU Timer can be any RTI[0:x]
- PWM3 SOCA can be any PWM[0:x]_SOCA
- PWM3 SOCB can be any PWM[0:x]_SOCB

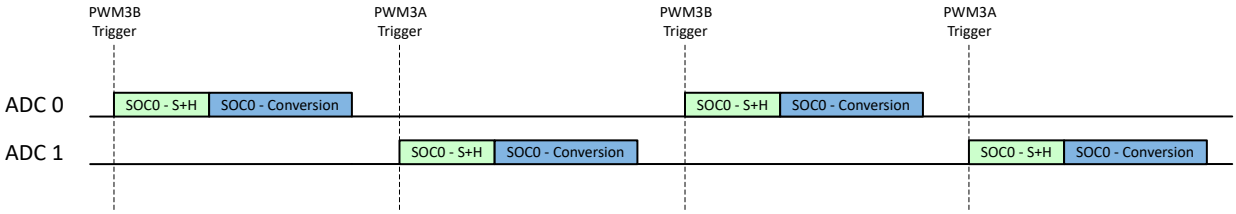


Figure 16-21. Example: Synchronous Equivalent Operation with Non-Overlapping Conversions

16.13.2 Choosing an Acquisition Window Duration

For correct operation, the input signal to the ADC must be allowed adequate time to charge the sample and hold capacitor, Ch. Typically, the S+H duration is chosen such that the sampling capacitor is charged to within ½ LSB or ¼ LSB of the final value, depending on the tolerable settling error.

The best methodology to determine the required settling time is to simulate the ADC and ADC driving circuits to make sure adequate settling performance. See *ADC Input Circuit Evaluation for C2000 MCUs* and *Charge-Sharing Driving Circuits for C2000 ADCs* for additional guidance on ADC signal conditioning circuit design and evaluation.

An approximation of the required settling time can also be determined using an RC settling model. The time constant for the model is given by the equation:

$$\tau = (R_S + R_{on}) \times C_h + R_S \times \left(C_S + C_p\right) \quad (15)$$

And the number of time constants needed is given by the equation:

$$k = ln\left(\frac{2^n}{settling\ error}\right) - ln\left(\frac{C_S + C_P}{CH}\right) \quad (16)$$

So the total S+H time must be set to at least:

$$t = k \cdot \tau \quad (17)$$

ADVANCE INFORMATION

Where the following parameters are provided by the ADC input model in the device data sheet:

- n = ADC resolution (in bits)
- R_{ON} = ADC sampling switch resistance (provided in Ω)
- C_H = ADC sampling capacitor (provided in pF)
- C_p = ADC channel parasitic input capacitance (provided in pF)

And the following parameters are dependent on the application design:

- settling error = tolerable settling error (in LSBs)
- R_s = ADC driving circuit source impedance (typically in Ω or kΩ)
- C_S = capacitance on ADC input pin (typically in pF or nF)

For example, assuming the following parameters:

- n = 12-bits
- R_{ON} = 500Ω
- C_H = 12.5pF
- C_p = 12.7pF
- settling error = ¼ LSB
- R_s = 180Ω
- C_s = 150pF

The time constant is calculated as:

$$\tau = (180\Omega + 500\Omega) \times 12.5pF + 180\Omega \times (150pF + 12.7pF) = 37.8ns \quad (18)$$

And the number of required time constants is:

$$k = ln\left(\frac{2^{12}}{0.25}\right) - ln\left(\frac{150pF + 12.7pF}{12.5pF}\right) = 9.70 - 2.57 = 7.13 \quad (19)$$

So the S+H time must be set to at least: 37.8ns × 7.13 = 270ns

If MCLK = 160MHz , then each MCLK cycle is 6.25ns. S+H duration is 270ns/6.25ns = 43.2 MCLK cycles, so ACQPS for this input is set to at least CEILING(43.2) – 1 = 42.

While this gives a rough estimate of the required acquisition window, a better method is to setup a circuit with the ADC input model, a model of the source impedance/capacitance, and any board parasitics in SPICE (or similar software) and simulate to verify that the sampling capacitor settles to the desired accuracy.

Note

The device data sheet specifies a minimum ADC S+H window duration. Do not use an ACQPS value that gives a duration less than this specification.

ADVANCE INFORMATION

16.13.3 Achieving Simultaneous Sampling

While each ADC does not have dual S+H circuits, achieving simultaneous sampling is accomplished by setting the SOC triggers on two or more ADC modules to use the same trigger source. The following example demonstrates simultaneous sampling on 3 ADCs based on the MCPWM3 event. A0_3 and A2_5 are sampled. An acquisition window of 20 MCLK cycles is used, but different durations are possible.

```
AdcORegs.ADCSOCOCTL.bit.CHSEL = 3;        //SOC0 converts ADC0 Channel 3
AdcORegs.ADCSOCOCTL.bit.ACQPS = 19;       //SOC0 uses sample duration of 20 MCLK cycles
AdcORegs.ADCSOCOCTL.bit.TRIGSEL = 10;     //SOC0 begins conversion on MCPWM3 SOCB
Adc2Regs.ADCSOCOCTL.bit.CHSEL = 5;        //SOC0 converts ADC2 Channel 5
Adc2Regs.ADCSOCOCTL.bit.ACQPS = 19;       //SOC0 uses sample duration of 20 MCLK cycles
Adc2Regs.ADCSOCOCTL.bit.TRIGSEL = 10;     //SOC0 begins conversion on MCPWM3 SOCB
```

When the MCPWM3 trigger is received, all 3 ADCs begin converting in parallel immediately. All results are stored in the ADCRESULT0 register for each ADC. Note that this assumes that all ADCs are idle when the trigger is received. If one or more ADCs is busy, the samples do not happen at exactly the same time.

16.13.4 Result Register Mapping

The ADC results and the ADC PPB results are duplicated for each memory bus controller in the system. Bus controllers include all CPUs, DMAs, and CLAs present on the specific part family and part number. For each bus controller, no access configuration is needed to allow read access to the result registers, and no contention occurs in cases where multiple bus controllers try to read the ADC results simultaneously.

16.13.5 Internal Temperature Sensor

The internal temperature sensor measures the junction temperature of the device. The output of the sensor can be sampled with the ADC through an internal connection. This can be enabled on channel A0_13 on ADC0 or A2_13 on ADC2 and on the CMP0_HP5 input by setting the ENABLE bit in the TSNSCTL register.

16.13.6 Designing an External Reference Circuit

Figure 16-22 shows the basic organization of the external voltage reference generation circuitry. For best performance, the externally generated reference voltage must be buffered by a precision op-amp with good bandwidth and low output impedance before being driven into the ADC reference pin. A capacitor between the high and low reference pins must be placed on the PCB as close to the pins as practical to help absorb high-frequency currents. A series resistor (typically <1Ω) in series with this capacitor is sometimes necessary to maintain op-amp stability.

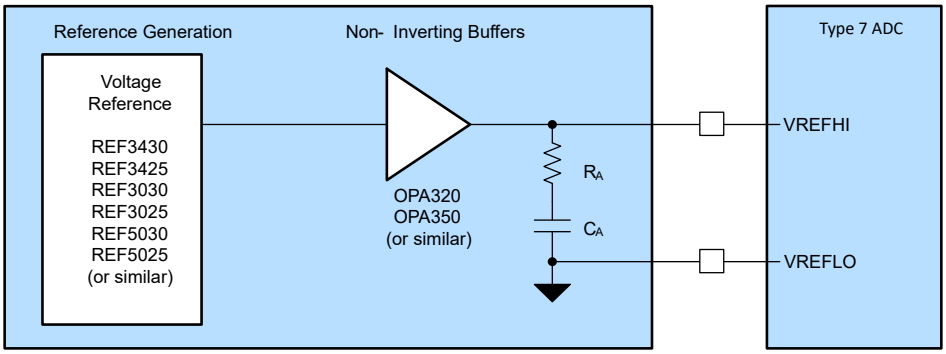


Figure 16-22. ADC Reference System

16.13.7 ADC-DAC Loopback Testing

For system diagnostic or functional safety purposes, the user application can perform a loopback test of the ADC module to verify that the ADC is converting correctly. Using the output of the DAC in the first CMPSS module, the device can be configured to supply a series of known voltages to the input of the converter, and the conversion result verified against expected results. Loopback test mode is enabled by setting the bit corresponding to the ADC module under test in the ADCLOOPBACK register in the analog subsystem module to 1. Figure 16-23 shows the connection between the CMPSS DAC output and the ADC.

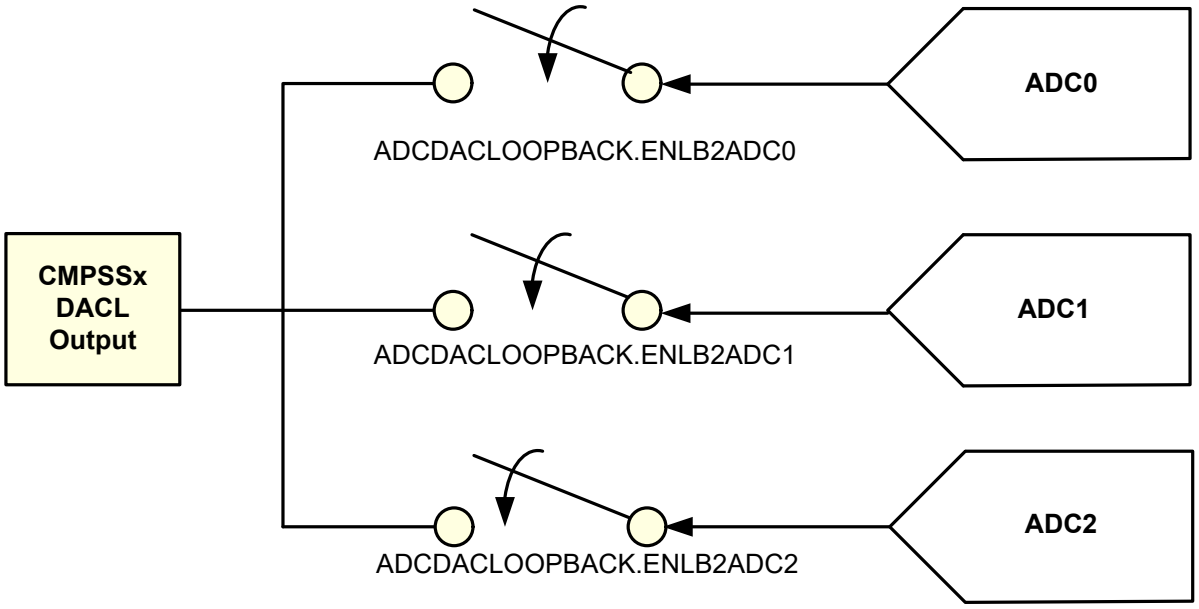


Figure 16-23. CMPSS to ADC Loopback Connection

In ADC loopback test mode, the following special considerations apply:

- The ADC module always samples the CMPSS1 DACL output, regardless of what channel is selected in the ADCSOCxCTL.CHSEL field.
- The minimum sampling window size (ACQPS) when converting the DAC output is 2.5µs (512 MCLK cycles at 200MHz MCLK).
- The output resolution of the CMPSS DAC is 6 bits. The lower 6 bits of the input DACVAL are discarded.
- ADC loopback test mode affects CMPSS trip voltages. Avoid enabling ADC loopback mode during regular CMPSS operation.

For more information on the CMPSS module and how to configure the CMPSS DAC, see the *Comparator Subsystem (CMPSS)* chapter.

16.13.8 Internal Test Mode

For diagnostic purposes, the ADC can sample various internal node voltages using a special input selection mux called INTERNALTEST. When internal test mode is enabled, the INTERNALTEST mux selection overrides the ADC-A input channel mux: ADC-A samples the INTERNALTEST selection instead of the channel selected by ADCSOCxCTL.CHSEL. Internal test mode can be used to sample the VDDCORE voltage, VREFLO, VDDA, VSSA, and the CMPSS DAC outputs.

To enable internal test mode, write the desired node selection to the TESTSEL field of the INTERNALTESTCTL analog subsystem register. For details of internal test mux connections to various internal device voltage nodes, refer to the INTERNALTESTCTL register description.

To disable internal test mode, write 0 to the TESTSEL field of the INTERNALTESTCTL register.

When using internal test mode, the following special considerations apply:

- INTERNALTESTCTL.TESTSEL overrides the value of ADCSOCxCTL.CHSEL on ADC0 when a non-zero value is configured.
- The minimum sampling window size (ACQPS) when converting INTERNALTEST is 2.56µs (512 MCLK cycles at 200MHz MCLK).
- The effective resolution of the CMPSS DAC outputs to INTERNALTEST is 6 bits.

For more information on the CMPSS module and how to configure the CMPSS DAC, see the *Comparator Subsystem (CMPSS)* chapter.

16.13.9 ADC Zero Offset Calibration

ADC offset error is determined and calibrated during factory testing. However, the user still has the option to perform offset calibration if the end application specifically requires this. This section describes how to perform offset calibration using the VREFLO connection (see your device DS for the ADC channel connection to VREFLO) for single-ended operation.

Zero offset error is defined as the difference from 0 that occurs when converting a voltage at VREFLO. The zero offset error can be positive or negative. To correct this error, an adjustment of equal magnitude and opposite polarity is written into the ADCOFFTRIM register. The value contained in this register is applied before the results are available in the ADC result registers. This operation is fully contained within the ADC core, so the timing of the results is not affected, and the full dynamic range of the ADC is maintained for any trim value.

Note

Regardless of the converter resolution, the size of each ADCOFFTRIM step is (VREFHI-VREFLO)/65536.

Use the following procedure to re-calibrate the ADC offset in 12-bit single-ended mode:

1. Set ADCOFFTRIM to +112 steps (0x70). This adds an artificial offset to account for negative offset that can reside in the ADC core.
2. Perform some multiple of 16 conversions on VREFLO (internal connection, see device DS for the ADC channel connected to VREFLO), accumulating the results (for example, 32 × 16 conversions = 512 conversions). Use the maximum value of ACQPS to make sure longer settling time to account for parasitic impedance of internal VREFLO connections.
3. Divide the accumulated result by the multiple of 16 (for example, for 512 conversions, divide by 32).
4. Set ADCOFFTRIM to 112 – result from step 3.

ADVANCE INFORMATION

16.14 ADC Registers

This Section describes the ADC Registers.

16.14.1 ADC Base Address Table

Table 16-12. ADC Base Address Table

Bit Field Name		DriverLib Name	Base Address
Instance	Structure		
Adc0Regs	ADC_LITE_REGS	ADC0	0x4000_0000
Adc1Regs	ADC_LITE_REGS	ADC1	0x4000_2000
Adc2Regs	ADC_LITE_REGS	ADC2	0x4000_4000
Adc0ResultRegs	ADC_LITE_RESULT_REGS	ADC0RESULT	0x4000_A000
Adc1ResultRegs	ADC_LITE_RESULT_REGS	ADC1RESULT	0x4000_B000
Adc2ResultRegs	ADC_LITE_RESULT_REGS	ADC2RESULT	0x4000_C000

16.14.2 ADC_LITE_REGS Registers

Table 16-13 lists the memory-mapped registers for the ADC_LITE_REGS registers. All register offset addresses not listed in Table 16-13 should be considered as reserved locations and the register contents should not be modified.

Table 16-13. ADC_LITE_REGS Registers

Offset	Acronym	Register Name	Write Protection	Section
0h	ADCCTL1	ADC Control 1 Register	EALLOW	Go
4h	ADCCTL2	ADC Control 2 Register	EALLOW	Go
10h	ADCINTSEL	ADC Interrupt 1, 2, 3 and 4 Selection Register	EALLOW	Go
14h	ADCDMAINTSEL	ADC DMA Interrupt 1, 2, 3 and 4 Selection Register	EALLOW	Go
18h	ADCRAWINTFLG	ADC Raw Interrupt Flag Register		Go
1Ch	ADCINTFLG	ADC Interrupt Flag Register		Go
20h	ADCINTFLGFRC	ADC Interrupt Flag Force Register		Go
24h	ADCINTFLGCLR	ADC Interrupt Flag Clear Register		Go
28h	ADCINTOVF	ADC Interrupt Overflow Register		Go
2Ch	ADCINTOVFCLR	ADC Interrupt Overflow Clear Register		Go
3Ch	ADCSOCFLG1	ADC SOC Flag 1 Register		Go
44h	ADCSOCOVF1	ADC SOC Overflow 1 Register		Go
48h	ADCSOCOVFCLR1	ADC SOC Overflow Clear 1 Register		Go
4Ch	ADCSOC0CTL	ADC SOC0 Control Register	EALLOW	Go
50h	ADCSOC1CTL	ADC SOC1 Control Register	EALLOW	Go
54h	ADCSOC2CTL	ADC SOC2 Control Register	EALLOW	Go
58h	ADCSOC3CTL	ADC SOC3 Control Register	EALLOW	Go
5Ch	ADCSOC4CTL	ADC SOC4 Control Register	EALLOW	Go
60h	ADCSOC5CTL	ADC SOC5 Control Register	EALLOW	Go
64h	ADCSOC6CTL	ADC SOC6 Control Register	EALLOW	Go
68h	ADCSOC7CTL	ADC SOC7 Control Register	EALLOW	Go
6Ch	ADCSOC8CTL	ADC SOC8 Control Register	EALLOW	Go
70h	ADCSOC9CTL	ADC SOC9 Control Register	EALLOW	Go
74h	ADCSOC10CTL	ADC SOC10 Control Register	EALLOW	Go
78h	ADCSOC11CTL	ADC SOC11 Control Register	EALLOW	Go
7Ch	ADCSOC12CTL	ADC SOC12 Control Register	EALLOW	Go
80h	ADCSOC13CTL	ADC SOC13 Control Register	EALLOW	Go
84h	ADCSOC14CTL	ADC SOC14 Control Register	EALLOW	Go
88h	ADCSOC15CTL	ADC SOC15 Control Register	EALLOW	Go
CCh	ADCEVTSTAT	ADC Event Status Register		Go
D0h	ADCEVTCLR	ADC Event Clear Register		Go
D4h	ADCEVTSEL	ADC Event Selection Register	EALLOW	Go
D8h	ADCEVTINTSEL	ADC Event Interrupt Selection Register	EALLOW	Go
E4h	ADCREV	ADC Revision Register		Go
E8h	ADCOFFTRIM	ADC Offset Trim Register 1	EALLOW	Go
100h	ADCPPB1CONFIG	ADC PPB1 Config Register	EALLOW	Go
110h	ADCPPB1TRIPHI	ADC PPB1 Trip High Register	EALLOW	Go
114h	ADCPPB1TRIPLO	ADC PPB1 Trip Low/Trigger Time Stamp Register	EALLOW	Go
120h	ADCPPB2CONFIG	ADC PPB2 Config Register	EALLOW	Go
130h	ADCPPB2TRIPHI	ADC PPB2 Trip High Register	EALLOW	Go

Table 16-13. ADC_LITE_REGS Registers (continued)

Offset	Acronym	Register Name	Write Protection	Section
134h	ADCPPB2TRIPLO	ADC PPB2 Trip Low/Trigger Time Stamp Register	EALLOW	Go
140h	ADCPPB3CONFIG	ADC PPB3 Config Register	EALLOW	Go
150h	ADCPPB3TRIPHI	ADC PPB3 Trip High Register	EALLOW	Go
154h	ADCPPB3TRIPLO	ADC PPB3 Trip Low/Trigger Time Stamp Register	EALLOW	Go
160h	ADCPPB4CONFIG	ADC PPB4 Config Register	EALLOW	Go
170h	ADCPPB4TRIPHI	ADC PPB4 Trip High Register	EALLOW	Go
174h	ADCPPB4TRIPLO	ADC PPB4 Trip Low/Trigger Time Stamp Register	EALLOW	Go
180h	ADCINTCYCLE	ADC Early Interrupt Generation Cycle	EALLOW	Go
19Ch	ADCREV2	ADC Wrapper Revision Register		Go
200h	ADCPPB1LIMIT	ADC PPB1Conversion Count Limit Register	EALLOW	Go
204h	ADCPPB1PCOUNT	ADC PPB1 Partial Conversion Count Register		Go
208h	ADCPPB1CONFIG2	ADC PPB1 Sum Shift Register		Go
20Ch	ADCPPB1PSUM	ADC PPB1 Partial Sum Register		Go
240h	ADCPPB2LIMIT	ADC PPB2Conversion Count Limit Register	EALLOW	Go
244h	ADCPPB2PCOUNT	ADC PPB2 Partial Conversion Count Register		Go
248h	ADCPPB2CONFIG2	ADC PPB2 Sum Shift Register		Go
24Ch	ADCPPB2PSUM	ADC PPB2 Partial Sum Register		Go
280h	ADCPPB3LIMIT	ADC PPB3Conversion Count Limit Register	EALLOW	Go
284h	ADCPPB3PCOUNT	ADC PPB3 Partial Conversion Count Register		Go
288h	ADCPPB3CONFIG2	ADC PPB3 Sum Shift Register		Go
28Ch	ADCPPB3PSUM	ADC PPB3 Partial Sum Register		Go
2C0h	ADCPPB4LIMIT	ADC PPB4Conversion Count Limit Register	EALLOW	Go
2C4h	ADCPPB4PCOUNT	ADC PPB4 Partial Conversion Count Register		Go
2C8h	ADCPPB4CONFIG2	ADC PPB4 Sum Shift Register		Go
2CCh	ADCPPB4PSUM	ADC PPB4 Partial Sum Register		Go
320h	ADCSEQCTL	ADC Sequencer common control Register		Go
324h	ADCSEQ1CONFIG_AM13E2X	ADC Sequencer 1 Config register		Go
328h	ADCSEQ2CONFIG_AM13E2X	ADC Sequencer 2 Config register		Go
32Ch	ADCSEQ3CONFIG_AM13E2X	ADC Sequencer 3 Config register		Go
330h	ADCSEQ4CONFIG_AM13E2X	ADC Sequencer 4 Config register		Go
800h	PWREN	Power enable		Go
804h	RSTCTL	Reset Control		Go
814h	STAT	Status Register		Go

Complex bit access types are encoded to fit into small table cells. Table 16-14 shows the codes that are used for access types in this section.

Table 16-14. ADC_LITE_REGS Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
R-0	R -0	Read Returns 0s
Write Type		
W	W	Write

Table 16-14. ADC_LITE_REGS Access Type Codes (continued)

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

1 ADCCTL1 Register (Offset = 0h) [Reset = 00000000h]

ADCCTL1 is shown in Figure 16-24 and described in Table 16-15.

Return to the Summary Table.

ADC Control 1 Register

Figure 16-24. ADCCTL1 Register

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

Table 16-16. ADCCTL1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-14	RESERVED	R	0h	Reserved
13	ADCBSY	R	0h	ADC Busy. Set when ADC SOC is generated, cleared by hardware four ADC clocks after negative edge of S/H pulse. Used by the ADC state machine to determine if ADC is available to sample. 0 ADC is available to sample next channel 1 ADC is busy and cannot sample another channel Reset type: SYSRSn
12	RESERVED	R	0h	Reserved
11-8	ADCBSYCHN	R	0h	ADC Busy Channel. Set when an ADC Start of Conversion (SOC) is generated. When ADCBSY=0: holds the value of the last converted SOC When ADCBSY=1: reflects the SOC currently being processed 0h SOC0 is currently processing or was last SOC converted 1h SOC1 is currently processing or was last SOC converted 2h SOC2 is currently processing or was last SOC converted 3h SOC3 is currently processing or was last SOC converted 4h SOC4 is currently processing or was last SOC converted 5h SOC5 is currently processing or was last SOC converted 6h SOC6 is currently processing or was last SOC converted 7h SOC7 is currently processing or was last SOC converted 8h SOC8 is currently processing or was last SOC converted 9h SOC9 is currently processing or was last SOC converted Ah SOC10 is currently processing or was last SOC converted Bh SOC11 is currently processing or was last SOC converted Ch SOC12 is currently processing or was last SOC converted Dh SOC13 is currently processing or was last SOC converted Eh SOC14 is currently processing or was last SOC converted Fh SOC15 is currently processing or was last SOC converted Reset type: SYSRSn

Table 16-16. ADCCTL1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
7	ADCPWDNZ	R/W	0h	ADC Power Down (active low). This bit controls the power up and power down of all the analog circuitry inside the analog core. 0 All analog circuitry inside the core is powered down 1 All analog circuitry inside the core is powered up Reset type: SYSRSn
6-3	RESERVED	R	0h	Reserved
2	INTPULSEPOS	R/W	0h	ADC Interrupt Pulse Position. 0 Interrupt pulse generation occurs when ADC begins conversion (at the end of the acquisition window) plus a number of MCLK cycles as specified in the ADCINTCYCLE.OFFSET register. 1 Interrupt pulse generation occurs at the end of the conversion, 1 cycle prior to the ADC result latching into its result register Reset type: SYSRSn
1-0	RESERVED	R	0h	Reserved

2 ADCCTL2 Register (Offset = 4h) [Reset = 00000000h]

ADCCTL2 is shown in Figure 16-25 and described in Table 16-16.

Return to the Summary Table.

ADC Control 2 Register

Figure 16-25. ADCCTL2 Register

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

Table 16-18. ADCCTL2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-13	RESERVED	R	0h	Reserved
12-9	RESERVED	R	0h	Reserved
8	RESERVED	R/W	0h	Reserved
7	RESERVED	R/W	0h	Reserved
6	RESERVED	R/W	0h	Reserved
5-4	RESERVED	R	0h	Reserved
3-0	PRESCALE	R/W	0h	ADC Clock Prescaler. 0000 ADCCLK = Input Clock / 1.0 0001 ADCCLK = Input Clock / 1.5 0010 ADCCLK = Input Clock / 2.0 0011 ADCCLK = Input Clock / 2.5 0100 ADCCLK = Input Clock / 3.0 0101 ADCCLK = Input Clock / 3.5 0110 ADCCLK = Input Clock / 4.0 0111 ADCCLK = Input Clock / 4.5 1000 ADCCLK = Input Clock / 5.0 1001 ADCCLK = Input Clock / 5.5 1010 ADCCLK = Input Clock / 6.0 1011 ADCCLK = Input Clock / 6.5 1100 ADCCLK = Input Clock / 7.0 1101 ADCCLK = Input Clock / 7.5 1110 ADCCLK = Input Clock / 8.0 1111 ADCCLK = Input Clock / 8.5 Reset type: SYSRSn

3 ADCINTSEL Register (Offset = 10h) [Reset = 00000000h]

ADCINTSEL is shown in Figure 16-26 and described in Table 16-17.

Return to the Summary Table.

ADC Interrupt 1, 2, 3 and 4 Selection Register

Figure 16-26. ADCINTSEL Register

31	30	29	28	27	26	25	24
INT4E	INT4CONT	RESERVED		INT4SEL			
R/W-0h	R/W-0h	R-0h		R/W-0h			
23	22	21	20	19	18	17	16
INT3E	INT3CONT	RESERVED		INT3SEL			
R/W-0h	R/W-0h	R-0h		R/W-0h			
15	14	13	12	11	10	9	8
INT2E	INT2CONT	RESERVED		INT2SEL			
R/W-0h	R/W-0h	R-0h		R/W-0h			
7	6	5	4	3	2	1	0
INT1E	INT1CONT	RESERVED		INT1SEL			
R/W-0h	R/W-0h	R-0h		R/W-0h			

Table 16-20. ADCINTSEL Register Field Descriptions

Bit	Field	Type	Reset	Description
31	INT4E	R/W	0h	ADCINT4 Interrupt Enable 0 ADCINT4 is disabled 1 ADCINT4 is enabled Reset type: SYSRSn
30	INT4CONT	R/W	0h	ADCINT4 Continue to Interrupt Mode 0 No further ADCINT4 pulses are generated until ADCINT4 flag (in ADCINTFLG register) is cleared by user. 1 ADCINT4 pulses are generated whenever an EOC pulse is generated irrespective of whether the flag bit is cleared or not. Reset type: SYSRSn
29-28	RESERVED	R	0h	Reserved
27-24	INT4SEL	R/W	0h	ADCINT4 EOC Source Select 00h EOC0 is trigger for ADCINT4 01h EOC1 is trigger for ADCINT4 02h EOC2 is trigger for ADCINT4 03h EOC3 is trigger for ADCINT4 04h EOC4 is trigger for ADCINT4 05h EOC5 is trigger for ADCINT4 06h EOC6 is trigger for ADCINT4 07h EOC7 is trigger for ADCINT4 08h EOC8 is trigger for ADCINT4 09h EOC9 is trigger for ADCINT4 0Ah EOC10 is trigger for ADCINT4 0Bh EOC11 is trigger for ADCINT4 0Ch EOC12 is trigger for ADCINT4 0Dh EOC13 is trigger for ADCINT4 0Eh EOC14 is trigger for ADCINT4 0Fh EOC15 is trigger for ADCINT4 Note : When oversampling is enabled, the end of the last oversampled coversion will trigger the interrupts Reset type: SYSRSn

Table 16-20. ADCINTSEL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
23	INT3E	R/W	0h	ADCINT3 Interrupt Enable 0 ADCINT3 is disabled 1 ADCINT3 is enabled Reset type: SYSRSn
22	INT3CONT	R/W	0h	ADCINT3 Continue to Interrupt Mode 0 No further ADCINT3 pulses are generated until ADCINT3 flag (in ADCINTFLG register) is cleared by user. 1 ADCINT3 pulses are generated whenever an EOC pulse is generated irrespective of whether the flag bit is cleared or not. Reset type: SYSRSn
21-20	RESERVED	R	0h	Reserved
19-16	INT3SEL	R/W	0h	ADCINT3 EOC Source Select 00h EOC0 is trigger for ADCINT3 01h EOC1 is trigger for ADCINT3 02h EOC2 is trigger for ADCINT3 03h EOC3 is trigger for ADCINT3 04h EOC4 is trigger for ADCINT3 05h EOC5 is trigger for ADCINT3 06h EOC6 is trigger for ADCINT3 07h EOC7 is trigger for ADCINT3 08h EOC8 is trigger for ADCINT3 09h EOC9 is trigger for ADCINT3 0Ah EOC10 is trigger for ADCINT3 0Bh EOC11 is trigger for ADCINT3 0Ch EOC12 is trigger for ADCINT3 0Dh EOC13 is trigger for ADCINT3 0Eh EOC14 is trigger for ADCINT3 0Fh EOC15 is trigger for ADCINT3 Note : When oversampling is enabled, the end of the last oversampled coversion will trigger the interrupts Reset type: SYSRSn
15	INT2E	R/W	0h	ADCINT2 Interrupt Enable 0 ADCINT2 is disabled 1 ADCINT2 is enabled Reset type: SYSRSn
14	INT2CONT	R/W	0h	ADCINT2 Continue to Interrupt Mode 0 No further ADCINT2 pulses are generated until ADCINT2 flag (in ADCINTFLG register) is cleared by user. 1 ADCINT2 pulses are generated whenever an EOC pulse is generated irrespective of whether the flag bit is cleared or not. Reset type: SYSRSn
13-12	RESERVED	R	0h	Reserved

Table 16-20. ADCINTSEL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
11-8	INT2SEL	R/W	0h	ADCINT2 EOC Source Select 00h EOC0 is trigger for ADCINT2 01h EOC1 is trigger for ADCINT2 02h EOC2 is trigger for ADCINT2 03h EOC3 is trigger for ADCINT2 04h EOC4 is trigger for ADCINT2 05h EOC5 is trigger for ADCINT2 06h EOC6 is trigger for ADCINT2 07h EOC7 is trigger for ADCINT2 08h EOC8 is trigger for ADCINT2 09h EOC9 is trigger for ADCINT2 0Ah EOC10 is trigger for ADCINT2 0Bh EOC11 is trigger for ADCINT2 0Ch EOC12 is trigger for ADCINT2 0Dh EOC13 is trigger for ADCINT2 0Eh EOC14 is trigger for ADCINT2 0Fh EOC15 is trigger for ADCINT2 Note : When oversampling is enabled, the end of the last oversampled coversion will trigger the interrupts Reset type: SYSRSn
7	INT1E	R/W	0h	ADCINT1 Interrupt Enable 0 ADCINT1 is disabled 1 ADCINT1 is enabled Reset type: SYSRSn
6	INT1CONT	R/W	0h	ADCINT1 Continue to Interrupt Mode 0 No further ADCINT1 pulses are generated until ADCINT1 flag (in ADCINTFLG register) is cleared by user. 1 ADCINT1 pulses are generated whenever an EOC pulse is generated irrespective of whether the flag bit is cleared or not. Reset type: SYSRSn
5-4	RESERVED	R	0h	Reserved
3-0	INT1SEL	R/W	0h	ADCINT1 EOC Source Select 00h EOC0 is trigger for ADCINT1 01h EOC1 is trigger for ADCINT1 02h EOC2 is trigger for ADCINT1 03h EOC3 is trigger for ADCINT1 04h EOC4 is trigger for ADCINT1 05h EOC5 is trigger for ADCINT1 06h EOC6 is trigger for ADCINT1 07h EOC7 is trigger for ADCINT1 08h EOC8 is trigger for ADCINT1 09h EOC9 is trigger for ADCINT1 0Ah EOC10 is trigger for ADCINT1 0Bh EOC11 is trigger for ADCINT1 0Ch EOC12 is trigger for ADCINT1 0Dh EOC13 is trigger for ADCINT1 0Eh EOC14 is trigger for ADCINT1 0Fh EOC15 is trigger for ADCINT1 Note : When oversampling is enabled, the end of the last oversampled coversion will trigger the interrupts Reset type: SYSRSn

4 ADCDMAINTSEL Register (Offset = 14h) [Reset = 00000000h]

ADCDMAINTSEL is shown in Figure 16-27 and described in Table 16-18.

Return to the Summary Table.

ADC DMA Interrupt 1, 2, 3 and 4 Selection Register

Figure 16-27. ADCDMAINTSEL Register

31	30	29	28	27	26	25	24
DMAINT4E	DMAINT4CONT	RESERVED		DMAINT4SEL			
R/W-0h	R/W-0h	R-0h		R/W-0h			
23	22	21	20	19	18	17	16
DMAINT3E	DMAINT3CONT	RESERVED		DMAINT3SEL			
R/W-0h	R/W-0h	R-0h		R/W-0h			
15	14	13	12	11	10	9	8
DMAINT2E	DMAINT2CONT	RESERVED		DMAINT2SEL			
R/W-0h	R/W-0h	R-0h		R/W-0h			
7	6	5	4	3	2	1	0
DMAINT1E	DMAINT1CONT	RESERVED		DMAINT1SEL			
R/W-0h	R/W-0h	R-0h		R/W-0h			

Table 16-22. ADCDMAINTSEL Register Field Descriptions

Bit	Field	Type	Reset	Description
31	DMAINT4E	R/W	0h	ADCDMAINT4 Interrupt Enable 0 ADCDMAINT4 is disabled 1 ADCDMAINT4 is enabled Reset type: SYSRSn
30	DMAINT4CONT	R/W	0h	ADCDMAINT4 Continue to Interrupt Mode 0 No further ADCDMAINT4 pulses are generated until ADCDMAINT4 flag (in ADCDMAINTFLG register) is cleared by user. 1 ADCDMAINT4 pulses are generated whenever an EOC pulse is generated irrespective of whether the flag bit is cleared or not. Reset type: SYSRSn
29-28	RESERVED	R	0h	Reserved
27-24	DMAINT4SEL	R/W	0h	ADCDMAINT4 EOC Source Select 00h EOC0 is trigger for ADCDMAINT4 01h EOC1 is trigger for ADCDMAINT4 02h EOC2 is trigger for ADCDMAINT4 03h EOC3 is trigger for ADCDMAINT4 04h EOC4 is trigger for ADCDMAINT4 05h EOC5 is trigger for ADCDMAINT4 06h EOC6 is trigger for ADCDMAINT4 07h EOC7 is trigger for ADCDMAINT4 08h EOC8 is trigger for ADCDMAINT4 09h EOC9 is trigger for ADCDMAINT4 0Ah EOC10 is trigger for ADCDMAINT4 0Bh EOC11 is trigger for ADCDMAINT4 0Ch EOC12 is trigger for ADCDMAINT4 0Dh EOC13 is trigger for ADCDMAINT4 0Eh EOC14 is trigger for ADCDMAINT4 0Fh EOC15 is trigger for ADCDMAINT4 Note : When oversampling is enabled, the end of the last oversampled coversion will trigger the DMAINTerrupts Reset type: SYSRSn

Table 16-22. ADCDMAINTSEL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
23	DMAINT3E	R/W	0h	ADCDMAINT3 Interrupt Enable 0 ADCDMAINT3 is disabled 1 ADCDMAINT3 is enabled Reset type: SYSRSn
22	DMAINT3CONT	R/W	0h	ADCDMAINT3 Continue to Interrupt Mode 0 No further ADCDMAINT3 pulses are generated until ADCDMAINT3 flag (in ADCDMAINTFLG register) is cleared by user. 1 ADCDMAINT3 pulses are generated whenever an EOC pulse is generated irrespective of whether the flag bit is cleared or not. Reset type: SYSRSn
21-20	RESERVED	R	0h	Reserved
19-16	DMAINT3SEL	R/W	0h	ADCDMAINT3 EOC Source Select 00h EOC0 is trigger for ADCDMAINT3 01h EOC1 is trigger for ADCDMAINT3 02h EOC2 is trigger for ADCDMAINT3 03h EOC3 is trigger for ADCDMAINT3 04h EOC4 is trigger for ADCDMAINT3 05h EOC5 is trigger for ADCDMAINT3 06h EOC6 is trigger for ADCDMAINT3 07h EOC7 is trigger for ADCDMAINT3 08h EOC8 is trigger for ADCDMAINT3 09h EOC9 is trigger for ADCDMAINT3 0Ah EOC10 is trigger for ADCDMAINT3 0Bh EOC11 is trigger for ADCDMAINT3 0Ch EOC12 is trigger for ADCDMAINT3 0Dh EOC13 is trigger for ADCDMAINT3 0Eh EOC14 is trigger for ADCDMAINT3 0Fh EOC15 is trigger for ADCDMAINT3 Note : When oversampling is enabled, the end of the last oversampled coversion will trigger the DMAINTerrupts Reset type: SYSRSn
15	DMAINT2E	R/W	0h	ADCDMAINT2 Interrupt Enable 0 ADCDMAINT2 is disabled 1 ADCDMAINT2 is enabled Reset type: SYSRSn
14	DMAINT2CONT	R/W	0h	ADCDMAINT2 Continue to Interrupt Mode 0 No further ADCDMAINT2 pulses are generated until ADCDMAINT2 flag (in ADCDMAINTFLG register) is cleared by user. 1 ADCDMAINT2 pulses are generated whenever an EOC pulse is generated irrespective of whether the flag bit is cleared or not. Reset type: SYSRSn
13-12	RESERVED	R	0h	Reserved

Table 16-22. ADCDMAINTSEL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
11-8	DMAINT2SEL	R/W	0h	ADCDMAINT2 EOC Source Select 00h EOC0 is trigger for ADCDMAINT2 01h EOC1 is trigger for ADCDMAINT2 02h EOC2 is trigger for ADCDMAINT2 03h EOC3 is trigger for ADCDMAINT2 04h EOC4 is trigger for ADCDMAINT2 05h EOC5 is trigger for ADCDMAINT2 06h EOC6 is trigger for ADCDMAINT2 07h EOC7 is trigger for ADCDMAINT2 08h EOC8 is trigger for ADCDMAINT2 09h EOC9 is trigger for ADCDMAINT2 0Ah EOC10 is trigger for ADCDMAINT2 0Bh EOC11 is trigger for ADCDMAINT2 0Ch EOC12 is trigger for ADCDMAINT2 0Dh EOC13 is trigger for ADCDMAINT2 0Eh EOC14 is trigger for ADCDMAINT2 0Fh EOC15 is trigger for ADCDMAINT2 Note : When oversampling is enabled, the end of the last oversampled coversion will trigger the DMAINTerrupts Reset type: SYSRSn
7	DMAINT1E	R/W	0h	ADCDMAINT1 Interrupt Enable 0 ADCDMAINT1 is disabled 1 ADCDMAINT1 is enabled Reset type: SYSRSn
6	DMAINT1CONT	R/W	0h	ADCDMAINT1 Continue to Interrupt Mode 0 No further ADCDMAINT1 pulses are generated until ADCDMAINT1 flag (in ADCDMAINTFLG register) is cleared by user. 1 ADCDMAINT1 pulses are generated whenever an EOC pulse is generated irrespective of whether the flag bit is cleared or not. Reset type: SYSRSn
5-4	RESERVED	R	0h	Reserved
3-0	DMAINT1SEL	R/W	0h	ADCDMAINT1 EOC Source Select 00h EOC0 is trigger for ADCDMAINT1 01h EOC1 is trigger for ADCDMAINT1 02h EOC2 is trigger for ADCDMAINT1 03h EOC3 is trigger for ADCDMAINT1 04h EOC4 is trigger for ADCDMAINT1 05h EOC5 is trigger for ADCDMAINT1 06h EOC6 is trigger for ADCDMAINT1 07h EOC7 is trigger for ADCDMAINT1 08h EOC8 is trigger for ADCDMAINT1 09h EOC9 is trigger for ADCDMAINT1 0Ah EOC10 is trigger for ADCDMAINT1 0Bh EOC11 is trigger for ADCDMAINT1 0Ch EOC12 is trigger for ADCDMAINT1 0Dh EOC13 is trigger for ADCDMAINT1 0Eh EOC14 is trigger for ADCDMAINT1 0Fh EOC15 is trigger for ADCDMAINT1 Note : When oversampling is enabled, the end of the last oversampled conversion will trigger the DMAINTerrupts Reset type: SYSRSn

5 ADCRAWINTFLG Register (Offset = 18h) [Reset = 00000000h]

ADCRAWINTFLG is shown in Figure 16-28 and described in Table 16-19.

Return to the Summary Table.

ADC Raw Interrupt Flag Register

Figure 16-28. ADCRAWINTFLG Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED				ADCDMARAWINT4	ADCDMARAWINT3	ADCDMARAWINT2	ADCDMARAWINT1
R-0h				R-0h	R-0h	R-0h	R-0h
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED				ADCRAWINT4	ADCRAWINT3	ADCRAWINT2	ADCRAWINT1
R-0h				R-0h	R-0h	R-0h	R-0h

Table 16-24. ADCRAWINTFLG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-20	RESERVED	R	0h	Reserved
19	ADCDMARAWINT4	R	0h	ADC DMA Raw Interrupt 4 Flag. Reading these flags indicates if the associated ADCDMAINT condition occurred. This flag will be set irrespective of corresponding DMAINTE setting 0 Selected EOC/OSINT event did not occur 1 Selected EOC/OSINT event occurred Writing corresponding INTCLR bit in ADCINTFLGCLR register will clear this bit. Reset type: SYSRSn
18	ADCDMARAWINT3	R	0h	ADC DMA Raw Interrupt 3 Flag. Reading these flags indicates if the associated ADCDMAINT condition occurred. This flag will be set irrespective of corresponding DMAINTE setting 0 Selected EOC/OSINT event did not occur 1 Selected EOC/OSINT event occurred Writing corresponding INTCLR bit in ADCINTFLGCLR register will clear this bit. Reset type: SYSRSn
17	ADCDMARAWINT2	R	0h	ADC DMA Raw Interrupt 2 Flag. Reading these flags indicates if the associated ADCDMAINT condition occurred. This flag will be set irrespective of corresponding DMAINTE setting 0 Selected EOC/OSINT event did not occur 1 Selected EOC/OSINT event occurred Writing corresponding INTCLR bit in ADCINTFLGCLR register will clear this bit. Reset type: SYSRSn

Table 16-24. ADCRAWINTFLG Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	ADCDMARAWINT1	R	0h	ADC DMA Raw Interrupt 1 Flag. Reading these flags indicates if the associated ADCDMAINT condition occurred. This flag will be set irrespective of corresponding DMAINTE setting 0 Selected EOC/OSINT event did not occur 1 Selected EOC/OSINT event occurred Writing corresponding INTCLR bit in ADCINTFLGCLR register will clear this bit. Reset type: SYSRSn
15-4	RESERVED	R	0h	Reserved
3	ADCRAWINT4	R	0h	ADC RAW Interrupt 4 Flag. Reading these flags indicates if the associated INT condition occurred. This flag will be set irrespective of corresponding INTE setting 0 Selected EOC/OSINT event did not occur 1 Selected EOC/OSINT event occured Writing corresponding INTCLR bit in ADCINTFLGCLR register will clear this bit. Reset type: SYSRSn
2	ADCRAWINT3	R	0h	ADC RAW Interrupt 3 Flag. Reading these flags indicates if the associated INT condition occurred. This flag will be set irrespective of corresponding INTE setting 0 Selected EOC/OSINT event did not occur 1 Selected EOC/OSINT event occured Writing corresponding INTCLR bit in ADCINTFLGCLR register will clear this bit. Reset type: SYSRSn
1	ADCRAWINT2	R	0h	ADC RAW Interrupt 2 Flag. Reading these flags indicates if the associated INT condition occurred. This flag will be set irrespective of corresponding INTE setting 0 Selected EOC/OSINT event did not occur 1 Selected EOC/OSINT event occured Writing corresponding INTCLR bit in ADCINTFLGCLR register will clear this bit. Reset type: SYSRSn
0	ADCRAWINT1	R	0h	ADC RAW Interrupt 1 Flag. Reading these flags indicates if the associated INT condition occurred. This flag will be set irrespective of corresponding INTE setting 0 Selected EOC/OSINT event did not occur 1 Selected EOC/OSINT event occurred Writing corresponding INTCLR bit in ADCINTFLGCLR register will clear this bit. Reset type: SYSRSn

6 ADCINTFLG Register (Offset = 1Ch) [Reset = 00000000h]

ADCINTFLG is shown in Figure 16-29 and described in Table 16-20.

Return to the Summary Table.

ADC Interrupt Flag Register

Figure 16-29. ADCINTFLG Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED				ADCDMAINT4	ADCDMAINT3	ADCDMAINT2	ADCDMAINT1
R-0h				R-0h	R-0h	R-0h	R-0h
15	14	13	12	11	10	9	8
RESERVED				ADCINT4RESULT	ADCINT3RESULT	ADCINT2RESULT	ADCINT1RESULT
R-0h				R-0h	R-0h	R-0h	R-0h
7	6	5	4	3	2	1	0
RESERVED				ADCINT4	ADCINT3	ADCINT2	ADCINT1
R-0h				R-0h	R-0h	R-0h	R-0h

Table 16-26. ADCINTFLG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-20	RESERVED	R	0h	Reserved
19	ADCDMAINT4	R	0h	ADC DMA Interrupt 4 Flag. Reading these flags indicates if the associated ADCDMAINT pulse was generated since the last clear. 0 No ADC DMA interru4pt pulse generated 1 ADC DMA interrupt pulse generated If the ADC interrupt is placed in continue to interrupt mode (DMAINTSEL register) then further interrupt pulses are generated whenever a selected EOC event occurs even if the flag bit is set. If the continuous mode is not enabled, then no further interrupt pulses are generated until the user clears this flag bit using the ADCINTFLGCLR register. Rather, an ADC interrupt overflow event occurs in the ADCINTOVF register. Reset type: SYSRSn
18	ADCDMAINT3	R	0h	ADC DMA Interrupt 3 Flag. Reading these flags indicates if the associated ADCDMAINT pulse was generated since the last clear. 0 No ADC DMA interrupt pulse generated 1 ADC DMA interrupt pulse generated If the ADC interrupt is placed in continue to interrupt mode (DMAINTSEL register) then further interrupt pulses are generated whenever a selected EOC event occurs even if the flag bit is set. If the continuous mode is not enabled, then no further interrupt pulses are generated until the user clears this flag bit using the ADCINTFLGCLR register. Rather, an ADC interrupt overflow event occurs in the ADCINTOVF register. Reset type: SYSRSn

ADVANCE INFORMATION

Table 16-26. ADCINTFLG Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
17	ADCDMAINT2	R	0h	ADC DMA Interrupt 2 Flag. Reading these flags indicates if the associated ADCDMAINT pulse was generated since the last clear. 0 No ADC DMA interrupt pulse generated 1 ADC DMA interrupt pulse generated If the ADC DMA interrupt is placed in continue to interrupt mode (DMAINTSEL register) then further interrupt pulses are generated whenever a selected EOC event occurs even if the flag bit is set. If the continuous mode is not enabled, then no further interrupt pulses are generated until the user clears this flag bit using the ADCINTFLGCLR register. Rather, an ADC interrupt overflow event occurs in the ADCINTOVF register. Reset type: SYSRSn
16	ADCDMAINT1	R	0h	ADC DMA Interrupt 1 Flag. Reading these flags indicates if the associated ADCDMAINT pulse was generated since the last clear. 0 No ADC DMA interrupt pulse generated 1 ADC DMA interrupt pulse generated If the ADC interrupt is placed in continue to interrupt mode (DMAINTSEL register) then further interrupt pulses are generated whenever a selected EOC event occurs even if the flag bit is set. If the continuous mode is not enabled, then no further interrupt pulses are generated until the user clears this flag bit using the ADCINTFLGCLR register. Rather, an ADC interrupt overflow event occurs in the ADCINTOVF register. Reset type: SYSRSn
15-12	RESERVED	R	0h	Reserved
11	ADCINT4RESULT	R	0h	ADC Interrupt 4 Results Ready Flag. This flag is set when the conversions results associated with ADCINT4 latch into the corresponding results register. 0 Conversion results have not latched 1 Conversion results have latched This flag can be used in an ISR that is entered in early interrupt mode to ensure that the corresponding results are ready before proceeding to read the result register. This flag can be cleared via the ACK bit in the ADCINTFLGCLR that also clears the ADCINT4 flag. In case results latch and this flag is already set, the corresponding flag in ADCINTOVF is NOT set. This flag does NOT have to be cleared in order for ADCINT ISRs to propagate to the NVIC . In case the associated SOC is associated with a PPB or PPBs, the flag will not be set until all associated PPB results latch. Reset type: SYSRSn
10	ADCINT3RESULT	R	0h	ADC Interrupt 3 Results Ready Flag. This flag is set when the conversions results associated with ADCINT3 latch into the corresponding results register. 0 Conversion results have not latched 1 Conversion results have latched This flag can be used in an ISR that is entered in early interrupt mode to ensure that the corresponding results are ready before proceeding to read the result register. This flag can be cleared via the ACK bit in the ADCINTFLGCLR that also clears the ADCINT3 flag. In case results latch and this flag is already set, the corresponding flag in ADCINTOVF is NOT set. This flag does NOT have to be cleared in order for ADCINT ISRs to propagate to the NVIC . In case the associated SOC is associated with a PPB or PPBs, the flag will not be set until all associated PPB results latch. Reset type: SYSRSn

Table 16-26. ADCINTFLG Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
9	ADCINT2RESULT	R	0h	ADC Interrupt 2 Results Ready Flag. This flag is set when the conversions results associated with ADCINT2 latch into the corresponding results register. 0 Conversion results have not latched 1 Conversion results have latched This flag can be used in an ISR that is entered in early interrupt mode to ensure that the corresponding results are ready before proceeding to read the result register. This flag can be cleared via the ACK bit in the ADCINTFLGCLR that also clears the ADCINT2 flag. In case results latch and this flag is already set, the corresponding flag in ADCINTOVF is NOT set. This flag does NOT have to be cleared in order for ADCINT ISRs to propagate to the NVIC . In case the associated SOC is associated with a PPB or PPBs, the flag will not be set until all associated PPB results latch. Reset type: SYSRSn
8	ADCINT1RESULT	R	0h	ADC Interrupt 1 Results Ready Flag. This flag is set when the conversions results associated with ADCINT1 latch into the corresponding results register. 0 Conversion results have not latched 1 Conversion results have latched This flag can be used in an ISR that is entered in early interrupt mode to ensure that the corresponding results are ready before proceeding to read the result register. This flag can be cleared via the ACK bit in the ADCINTFLGCLR that also clears the ADCINT1 flag. In case results latch and this flag is already set, the corresponding flag in ADCINTOVF is NOT set. This flag does NOT have to be cleared in order for ADCINT ISRs to propagate to the NVIC . In case the associated SOC is associated with a PPB or PPBs, the flag will not be set until all associated PPB results latch. Reset type: SYSRSn
7-4	RESERVED	R	0h	Reserved
3	ADCINT4	R	0h	ADC Interrupt 4 Flag. Reading these flags indicates if the associated ADCINT pulse was generated since the last clear. 0 No ADC interrupt pulse generated 1 ADC interrupt pulse generated If the ADC interrupt is placed in continue to interrupt mode (INTSEL register) then further interrupt pulses are generated whenever a selected EOC event occurs even if the flag bit is set. If the continuous mode is not enabled, then no further interrupt pulses are generated until the user clears this flag bit using the ADCINFLGCLR register. Rather, an ADC interrupt overflow event occurs in the ADCINTOVF register. Reset type: SYSRSn
2	ADCINT3	R	0h	ADC Interrupt 3 Flag. Reading these flags indicates if the associated ADCINT pulse was generated since the last clear. 0 No ADC interrupt pulse generated 1 ADC interrupt pulse generated If the ADC interrupt is placed in continue to interrupt mode (INTSEL register) then further interrupt pulses are generated whenever a selected EOC event occurs even if the flag bit is set. If the continuous mode is not enabled, then no further interrupt pulses are generated until the user clears this flag bit using the ADCINTFLGCLR register. Rather, an ADC interrupt overflow event occurs in the ADCINTOVF register. Reset type: SYSRSn

Table 16-26. ADCINTFLG Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
1	ADCINT2	R	0h	ADC Interrupt 2 Flag. Reading these flags indicates if the associated ADCINT pulse was generated since the last clear. 0 No ADC interrupt pulse generated 1 ADC interrupt pulse generated If the ADC interrupt is placed in continue to interrupt mode (INTSEL register) then further interrupt pulses are generated whenever a selected EOC event occurs even if the flag bit is set. If the continuous mode is not enabled, then no further interrupt pulses are generated until the user clears this flag bit using the ADCINTFLGCLR register. Rather, an ADC interrupt overflow event occurs in the ADCINTOVF register. Reset type: SYSRSn
0	ADCINT1	R	0h	ADC Interrupt 1 Flag. Reading these flags indicates if the associated ADCINT pulse was generated since the last clear. 0 No ADC interrupt pulse generated 1 ADC interrupt pulse generated If the ADC interrupt is placed in continue to interrupt mode (INTSEL register) then further interrupt pulses are generated whenever a selected EOC event occurs even if the flag bit is set. If the continuous mode is not enabled, then no further interrupt pulses are generated until the user clears this flag bit using the ADCINTFLGCLR register. Rather, an ADC interrupt overflow event occurs in the ADCINTOVF register. Reset type: SYSRSn

7 ADCINTFLGFRC Register (Offset = 20h) [Reset = 00000000h]

ADCINTFLGFRC is shown in Figure 16-30 and described in Table 16-21.

Return to the Summary Table.

ADC Interrupt Flag Force Register

Figure 16-30. ADCINTFLGFRC Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED				ADCDMAINT4	ADCDMAINT3	ADCDMAINT2	ADCDMAINT1
R-0h				R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED				ADCINT4	ADCINT3	ADCINT2	ADCINT1
R-0h				R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h

Table 16-28. ADCINTFLGFRC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-20	RESERVED	R	0h	Reserved
19	ADCDMAINT4	R-0/W1S	0h	ADC DMA interrupt 4 Flag Force. Reads return 0. 0 No action 1 Forces ADCDMAINT4 flags in the ADCINTFLG and ADCRAWINTFLG registers. Reset type: SYSRSn
18	ADCDMAINT3	R-0/W1S	0h	ADC DMA interrupt 3 Flag Force. Reads return 0. 0 No action 1 Forces ADCDMAINT3 flags in the ADCINTFLG and ADCRAWINTFLG registers. Reset type: SYSRSn
17	ADCDMAINT2	R-0/W1S	0h	ADC DMA interrupt 2 Flag Force. Reads return 0. 0 No action 1 Forces ADCDMAINT2 flags in the ADCINTFLG and ADCRAWINTFLG registers. Reset type: SYSRSn
16	ADCDMAINT1	R-0/W1S	0h	ADC DMA interrupt 1 Flag Force. Reads return 0. 0 No action 1 Forces ADCDMAINT1 flags in the ADCINTFLG and ADCRAWINTFLG registers. Reset type: SYSRSn
15-4	RESERVED	R	0h	Reserved
3	ADCINT4	R-0/W1S	0h	ADC Interrupt 4 Flag Force. Reads return 0. 0 No action 1 Forces ADCINT4 flags in the ADCINTFLG and ADCRAWINTFLG registers. Reset type: SYSRSn

Table 16-28. ADCINTFLGFRC Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
2	ADCINT3	R-0/W1S	0h	ADC Interrupt 3 Flag Force. Reads return 0. 0 No action 1 Forces ADCINT3 flags in the ADCINTFLG and ADCRAWINTFLG registers. Reset type: SYSRSn
1	ADCINT2	R-0/W1S	0h	ADC Interrupt 2 Flag Force. Reads return 0. 0 No action 1 Forces ADCINT2 flags in the ADCINTFLG and ADCRAWINTFLG registers. Reset type: SYSRSn
0	ADCINT1	R-0/W1S	0h	ADC Interrupt 1 Flag Force. Reads return 0. 0 No action 1 Forces ADCINT1 flags in the ADCINTFLG and ADCRAWINTFLG registers. Reset type: SYSRSn

8 ADCINTFLGCLR Register (Offset = 24h) [Reset = 00000000h]

ADCINTFLGCLR is shown in Figure 16-31 and described in Table 16-22.

Return to the Summary Table.

ADC Interrupt Flag Clear Register

Figure 16-31. ADCINTFLGCLR Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED				ADCDMAINT4	ADCDMAINT3	ADCDMAINT2	ADCDMAINT1
R-0h				R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED				ADCINT4	ADCINT3	ADCINT2	ADCINT1
R-0h				R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h

Table 16-30. ADCINTFLGCLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-20	RESERVED	R	0h	Reserved
19	ADCDMAINT4	R-0/W1C	0h	ADC DMA Interrupt 4 Flag Clear. Reads return 0. 0 No action 1 Clears ADDMACINT4 flags in the ADCINTFLG ,ADCRAWINTFLG registers. If software sets the clear bit on the same cycle that hardware is trying to set the flag bit, then hardware has priority and the overflow bit will not be set Reset type: SYSRSn
18	ADCDMAINT3	R-0/W1C	0h	ADC DMA Interrupt 3 Flag Clear. Reads return 0. 0 No action 1 Clears ADDMACINT3 flags in the ADCINTFLG ,ADCRAWINTFLG registers. If software sets the clear bit on the same cycle that hardware is trying to set the flag bit, then hardware has priority and the overflow bit will not be set Reset type: SYSRSn
17	ADCDMAINT2	R-0/W1C	0h	ADC DMA Interrupt 2 Flag Clear. Reads return 0. 0 No action 1 Clears ADDMACINT2 flags in the ADCINTFLG ,ADCRAWINTFLG registers. If software sets the clear bit on the same cycle that hardware is trying to set the flag bit, then hardware has priority and the overflow bit will not be set Reset type: SYSRSn

Table 16-30. ADCINTFLGCLR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	ADCDMAINT1	R-0/W1C	0h	ADC DMA Interrupt 1 Flag Clear. Reads return 0. 0 No action 1 Clears ADDMACINT1 flags in the ADCINTFLG ,ADCRAWINTFLG registers. If software sets the clear bit on the same cycle that hardware is trying to set the flag bit, then hardware has priority and the overflow bit will not be set Reset type: SYSRSn
15-4	RESERVED	R	0h	Reserved
3	ADCINT4	R-0/W1C	0h	ADC Interrupt 4 Flag Clear. Reads return 0. 0 No action 1 Clears ADCINT4 and ADCINT4RESULT flags in the ADCINTFLG,, ADCRAWINTFLG registers. If software sets the clear bit on the same cycle that hardware is trying to set the flag bit, then hardware has priority and the overflow bit will not be set Reset type: SYSRSn
2	ADCINT3	R-0/W1C	0h	ADC Interrupt 3 Flag Clear. Reads return 0. 0 No action 1 Clears ADCINT3 and ADCINT3RESULT flags in the ADCINTFLG, ADCRAWINTFLG registers. If software sets the clear bit on the same cycle that hardware is trying to set the flag bit, then hardware has priority and the overflow bit will not be set Reset type: SYSRSn
1	ADCINT2	R-0/W1C	0h	ADC Interrupt 2 Flag Clear. Reads return 0. 0 No action 1 Clears ADCINT2 and ADCINT2RESULT flags in the ADCINTFLG, ADCRAWINTFLG registers. If software sets the clear bit on the same cycle that hardware is trying to set the flag bit, then hardware has priority and the overflow bit will not be set Reset type: SYSRSn
0	ADCINT1	R-0/W1C	0h	ADC Interrupt 1 Flag Clear. Reads return 0. 0 No action 1 Clears ADCINT1 and ADCINT1RESULT flags in the ADCINTFLG, ADCRAWINTFLG registers. If software sets the clear bit on the same cycle that hardware is trying to set the flag bit, then hardware has priority and the overflow bit will not be set Reset type: SYSRSn

9 ADCINTOVF Register (Offset = 28h) [Reset = 00000000h]

ADCINTOVF is shown in Figure 16-32 and described in Table 16-23.

Return to the Summary Table.

ADC Interrupt Overflow Register

Figure 16-32. ADCINTOVF Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED				ADCDMAINT4OVF	ADCDMAINT3OVF	ADCDMAINT2OVF	ADCDMAINT1OVF
R-0h				R-0h	R-0h	R-0h	R-0h
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED				ADCINT4OVF	ADCINT3OVF	ADCINT2OVF	ADCINT1OVF
R-0h				R-0h	R-0h	R-0h	R-0h

Table 16-32. ADCINTOVF Register Field Descriptions

Bit	Field	Type	Reset	Description
31-20	RESERVED	R	0h	Reserved
19	ADCDMAINT4OVF	R	0h	ADC DMA Interrupt 4 Overflow Flags Indicates if an overflow occurred when generating ADCINT pulses. If the respective ADCINTFLG bit is set and a selected additional EOC trigger is generated, then an overflow condition occurs. 0 No ADC DMA Interrupt overflow event detected. 1 ADC DMA Interrupt overflow event detected. The overflow bit does not care about the continuous mode bit state. An overflow condition is generated irrespective of this mode selection. Reset type: SYSRSn
18	ADCDMAINT3OVF	R	0h	ADC DMA Interrupt 3 Overflow Flags Indicates if an overflow occurred when generating ADCINT pulses. If the respective ADCINTFLG bit is set and a selected additional EOC trigger is generated, then an overflow condition occurs. 0 No ADC DMA Interrupt overflow event detected. 1 ADC DMA Interrupt overflow event detected. The overflow bit does not care about the continuous mode bit state. An overflow condition is generated irrespective of this mode selection. Reset type: SYSRSn
17	ADCDMAINT2OVF	R	0h	ADC DMA Interrupt 2 Overflow Flags Indicates if an overflow occurred when generating ADCINT pulses. If the respective ADCINTFLG bit is set and a selected additional EOC trigger is generated, then an overflow condition occurs. 0 No ADC DMA Interrupt overflow event detected. 1 ADC DMA Interrupt overflow event detected. The overflow bit does not care about the continuous mode bit state. An overflow condition is generated irrespective of this mode selection. Reset type: SYSRSn

Table 16-32. ADCINTOVF Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	ADCDMAINT1OVF	R	0h	ADC DMA Interrupt 1 Overflow Flags Indicates if an overflow occurred when generating ADCINT pulses. If the respective ADCINTFLG bit is set and a selected additional EOC trigger is generated, then an overflow condition occurs. 0 No ADC DMA Interrupt overflow event detected. 1 ADC DMA Interrupt overflow event detected. The overflow bit does not care about the continuous mode bit state. An overflow condition is generated irrespective of this mode selection. Reset type: SYSRSn
15-4	RESERVED	R	0h	Reserved
3	ADCINT4OVF	R	0h	ADC Interrupt 4 Overflow Flags Indicates if an overflow occurred when generating ADCINT pulses. If the respective ADCINTFLG bit is set and a selected additional EOC trigger is generated, then an overflow condition occurs. 0 No ADC interrupt overflow event detected. 1 ADC Interrupt overflow event detected. The overflow bit does not care about the continuous mode bit state. An overflow condition is generated irrespective of this mode selection. Reset type: SYSRSn
2	ADCINT3OVF	R	0h	ADC Interrupt 3 Overflow Flags Indicates if an overflow occurred when generating ADCINT pulses. If the respective ADCINTFLG bit is set and a selected additional EOC trigger is generated, then an overflow condition occurs. 0 No ADC interrupt overflow event detected. 1 ADC Interrupt overflow event detected. The overflow bit does not care about the continuous mode bit state. An overflow condition is generated irrespective of this mode selection. Reset type: SYSRSn
1	ADCINT2OVF	R	0h	ADC Interrupt 2 Overflow Flags Indicates if an overflow occurred when generating ADCINT pulses. If the respective ADCINTFLG bit is set and a selected additional EOC trigger is generated, then an overflow condition occurs. 0 No ADC interrupt overflow event detected. 1 ADC Interrupt overflow event detected. The overflow bit does not care about the continuous mode bit state. An overflow condition is generated irrespective of this mode selection. Reset type: SYSRSn
0	ADCINT1OVF	R	0h	ADC Interrupt 1 Overflow Flags Indicates if an overflow occurred when generating ADCINT pulses. If the respective ADCINTFLG bit is set and a selected additional EOC trigger is generated, then an overflow condition occurs. 0 No ADC interrupt overflow event detected. 1 ADC Interrupt overflow event detected. The overflow bit does not care about the continuous mode bit state. An overflow condition is generated irrespective of this mode selection. Reset type: SYSRSn

10 ADCINTOVFCLR Register (Offset = 2Ch) [Reset = 00000000h]

ADCINTOVFCLR is shown in Figure 16-33 and described in Table 16-24.

Return to the Summary Table.

ADC Interrupt Overflow Clear Register

Figure 16-33. ADCINTOVFCLR Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED				ADCDMAINT4OVF	ADCDMAINT3OVF	ADCDMAINT2OVF	ADCDMAINT1OVF
R-0h				R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED				ADCINT4OVF	ADCINT3OVF	ADCINT2OVF	ADCINT1OVF
R-0h				R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h

Table 16-34. ADCINTOVFCLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-20	RESERVED	R	0h	Reserved
19	ADCDMAINT4OVF	R-0/W1C	0h	ADC DMA Interrupt 4 Overflow Clear Bits 0 No action. 1 Clears the respective overflow bit in the ADCINTOVF register. If software tries to set this bit on the same clock cycle that hardware tries to set the overflow bit in the ADCINTOVF register, then hardware has priority and the ADCINTOVF bit will be set. Reset type: SYSRSn
18	ADCDMAINT3OVF	R-0/W1C	0h	ADC DMA Interrupt 3 Overflow Clear Bits 0 No action. 1 Clears the respective overflow bit in the ADCINTOVF register. If software tries to set this bit on the same clock cycle that hardware tries to set the overflow bit in the ADCINTOVF register, then hardware has priority and the ADCINTOVF bit will be set. Reset type: SYSRSn
17	ADCDMAINT2OVF	R-0/W1C	0h	ADC DMA Interrupt 2 Overflow Clear Bits 0 No action. 1 Clears the respective overflow bit in the ADCINTOVF register. If software tries to set this bit on the same clock cycle that hardware tries to set the overflow bit in the ADCINTOVF register, then hardware has priority and the ADCINTOVF bit will be set. Reset type: SYSRSn
16	ADCDMAINT1OVF	R-0/W1C	0h	ADC DMA Interrupt 1 Overflow Clear Bits 0 No action. 1 Clears the respective overflow bit in the ADCINTOVF register. If software tries to set this bit on the same clock cycle that hardware tries to set the overflow bit in the ADCINTOVF register, then hardware has priority and the ADCINTOVF bit will be set. Reset type: SYSRSn

Table 16-34. ADCINTOVFCLR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
15-4	RESERVED	R	0h	Reserved
3	ADCINT4OVF	R-0/W1C	0h	ADC Interrupt 4 Overflow Clear Bits 0 No action. 1 Clears the respective overflow bit in the ADCINTOVF register. If software tries to set this bit on the same clock cycle that hardware tries to set the overflow bit in the ADCINTOVF register, then hardware has priority and the ADCINTOVF bit will be set. Reset type: SYSRSn
2	ADCINT3OVF	R-0/W1C	0h	ADC Interrupt 3 Overflow Clear Bits 0 No action. 1 Clears the respective overflow bit in the ADCINTOVF register. If software tries to set this bit on the same clock cycle that hardware tries to set the overflow bit in the ADCINTOVF register, then hardware has priority and the ADCINTOVF bit will be set. Reset type: SYSRSn
1	ADCINT2OVF	R-0/W1C	0h	ADC Interrupt 2 Overflow Clear Bits 0 No action. 1 Clears the respective overflow bit in the ADCINTOVF register. If software tries to set this bit on the same clock cycle that hardware tries to set the overflow bit in the ADCINTOVF register, then hardware has priority and the ADCINTOVF bit will be set. Reset type: SYSRSn
0	ADCINT1OVF	R-0/W1C	0h	ADC Interrupt 1 Overflow Clear Bits 0 No action. 1 Clears the respective overflow bit in the ADCINTOVF register. If software tries to set this bit on the same clock cycle that hardware tries to set the overflow bit in the ADCINTOVF register, then hardware has priority and the ADCINTOVF bit will be set. Reset type: SYSRSn

11 ADCSOCFLG1 Register (Offset = 3Ch) [Reset = 00000000h]

ADCSOCFLG1 is shown in Figure 16-34 and described in Table 16-25.

Return to the Summary Table.

ADC SOC Flag 1 Register

Figure 16-34. ADCSOCFLG1 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
SOC15	SOC14	SOC13	SOC12	SOC11	SOC10	SOC9	SOC8
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
7	6	5	4	3	2	1	0
SOC7	SOC6	SOC5	SOC4	SOC3	SOC2	SOC1	SOC0
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

Table 16-36. ADCSOCFLG1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15	SOC15	R	0h	SOC15 Start of Conversion Flag. Indicates the state of SOC15 conversions. 0 No sample pending for SOC15. 1 Trigger has been received and sample is pending for SOC15. This bit will be automatically cleared when the SOC15 conversion is started. If contention exists where this bit receives both a request to set and a request to clear on the same cycle, regardless of the source of either, this bit will be set and the request to clear will be ignored. In this case the overflow bit in the ADCSOCOVF1 register will not be affected regardless of whether this bit was previously set or not. Reset type: SYSRSn
14	SOC14	R	0h	SOC14 Start of Conversion Flag. Indicates the state of SOC14 conversions. 0 No sample pending for SOC14. 1 Trigger has been received and sample is pending for SOC14. This bit will be automatically cleared when the SOC14 conversion is started. If contention exists where this bit receives both a request to set and a request to clear on the same cycle, regardless of the source of either, this bit will be set and the request to clear will be ignored. In this case the overflow bit in the ADCSOCOVF1 register will not be affected regardless of whether this bit was previously set or not. Reset type: SYSRSn

Table 16-36. ADCSOCFLG1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
13	SOC13	R	0h	SOC13 Start of Conversion Flag. Indicates the state of SOC13 conversions. 0 No sample pending for SOC13. 1 Trigger has been received and sample is pending for SOC13. This bit will be automatically cleared when the SOC13 conversion is started. If contention exists where this bit receives both a request to set and a request to clear on the same cycle, regardless of the source of either, this bit will be set and the request to clear will be ignored. In this case the overflow bit in the ADCSOCOVF1 register will not be affected regardless of whether this bit was previously set or not. Reset type: SYSRSn
12	SOC12	R	0h	SOC12 Start of Conversion Flag. Indicates the state of SOC12 conversions. 0 No sample pending for SOC12. 1 Trigger has been received and sample is pending for SOC12. This bit will be automatically cleared when the SOC12 conversion is started. If contention exists where this bit receives both a request to set and a request to clear on the same cycle, regardless of the source of either, this bit will be set and the request to clear will be ignored. In this case the overflow bit in the ADCSOCOVF1 register will not be affected regardless of whether this bit was previously set or not. Reset type: SYSRSn
11	SOC11	R	0h	SOC11 Start of Conversion Flag. Indicates the state of SOC11 conversions. 0 No sample pending for SOC11. 1 Trigger has been received and sample is pending for SOC11. This bit will be automatically cleared when the SOC11 conversion is started. If contention exists where this bit receives both a request to set and a request to clear on the same cycle, regardless of the source of either, this bit will be set and the request to clear will be ignored. In this case the overflow bit in the ADCSOCOVF1 register will not be affected regardless of whether this bit was previously set or not. Reset type: SYSRSn
10	SOC10	R	0h	SOC10 Start of Conversion Flag. Indicates the state of SOC10 conversions. 0 No sample pending for SOC10. 1 Trigger has been received and sample is pending for SOC10. This bit will be automatically cleared when the SOC10 conversion is started. If contention exists where this bit receives both a request to set and a request to clear on the same cycle, regardless of the source of either, this bit will be set and the request to clear will be ignored. In this case the overflow bit in the ADCSOCOVF1 register will not be affected regardless of whether this bit was previously set or not. Reset type: SYSRSn

Table 16-36. ADCSOCFLG1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
9	SOC9	R	0h	SOC9 Start of Conversion Flag. Indicates the state of SOC9 conversions. 0 No sample pending for SOC9. 1 Trigger has been received and sample is pending for SOC9. This bit will be automatically cleared when the SOC9 conversion is started. If contention exists where this bit receives both a request to set and a request to clear on the same cycle, regardless of the source of either, this bit will be set and the request to clear will be ignored. In this case the overflow bit in the ADCSOCOVF1 register will not be affected regardless of whether this bit was previously set or not. Reset type: SYSRSn
8	SOC8	R	0h	SOC8 Start of Conversion Flag. Indicates the state of SOC8 conversions. 0 No sample pending for SOC8. 1 Trigger has been received and sample is pending for SOC8. This bit will be automatically cleared when the SOC8 conversion is started. If contention exists where this bit receives both a request to set and a request to clear on the same cycle, regardless of the source of either, this bit will be set and the request to clear will be ignored. In this case the overflow bit in the ADCSOCOVF1 register will not be affected regardless of whether this bit was previously set or not. Reset type: SYSRSn
7	SOC7	R	0h	SOC7 Start of Conversion Flag. Indicates the state of SOC7 conversions. 0 No sample pending for SOC7. 1 Trigger has been received and sample is pending for SOC7. This bit will be automatically cleared when the SOC7 conversion is started. If contention exists where this bit receives both a request to set and a request to clear on the same cycle, regardless of the source of either, this bit will be set and the request to clear will be ignored. In this case the overflow bit in the ADCSOCOVF1 register will not be affected regardless of whether this bit was previously set or not. Reset type: SYSRSn
6	SOC6	R	0h	SOC6 Start of Conversion Flag. Indicates the state of SOC6 conversions. 0 No sample pending for SOC6. 1 Trigger has been received and sample is pending for SOC6. This bit will be automatically cleared when the SOC6 conversion is started. If contention exists where this bit receives both a request to set and a request to clear on the same cycle, regardless of the source of either, this bit will be set and the request to clear will be ignored. In this case the overflow bit in the ADCSOCOVF1 register will not be affected regardless of whether this bit was previously set or not. Reset type: SYSRSn

Table 16-36. ADCSOCFLG1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
5	SOC5	R	0h	SOC5 Start of Conversion Flag. Indicates the state of SOC5 conversions. 0 No sample pending for SOC5. 1 Trigger has been received and sample is pending for SOC5. This bit will be automatically cleared when the SOC5 conversion is started. If contention exists where this bit receives both a request to set and a request to clear on the same cycle, regardless of the source of either, this bit will be set and the request to clear will be ignored. In this case the overflow bit in the ADCSOCOVF1 register will not be affected regardless of whether this bit was previously set or not. Reset type: SYSRSn
4	SOC4	R	0h	SOC4 Start of Conversion Flag. Indicates the state of SOC4 conversions. 0 No sample pending for SOC4. 1 Trigger has been received and sample is pending for SOC4. This bit will be automatically cleared when the SOC4 conversion is started. If contention exists where this bit receives both a request to set and a request to clear on the same cycle, regardless of the source of either, this bit will be set and the request to clear will be ignored. In this case the overflow bit in the ADCSOCOVF1 register will not be affected regardless of whether this bit was previously set or not. Reset type: SYSRSn
3	SOC3	R	0h	SOC3 Start of Conversion Flag. Indicates the state of SOC3 conversions. 0 No sample pending for SOC3. 1 Trigger has been received and sample is pending for SOC3. This bit will be automatically cleared when the SOC3 conversion is started. If contention exists where this bit receives both a request to set and a request to clear on the same cycle, regardless of the source of either, this bit will be set and the request to clear will be ignored. In this case the overflow bit in the ADCSOCOVF1 register will not be affected regardless of whether this bit was previously set or not. Reset type: SYSRSn
2	SOC2	R	0h	SOC2 Start of Conversion Flag. Indicates the state of SOC2 conversions. 0 No sample pending for SOC2. 1 Trigger has been received and sample is pending for SOC2. This bit will be automatically cleared when the SOC2 conversion is started. If contention exists where this bit receives both a request to set and a request to clear on the same cycle, regardless of the source of either, this bit will be set and the request to clear will be ignored. In this case the overflow bit in the ADCSOCOVF1 register will not be affected regardless of whether this bit was previously set or not. Reset type: SYSRSn

Table 16-36. ADCSOCFLG1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
1	SOC1	R	0h	SOC1 Start of Conversion Flag. Indicates the state of SOC1 conversions. 0 No sample pending for SOC1. 1 Trigger has been received and sample is pending for SOC1. This bit will be automatically cleared when the SOC1 conversion is started. If contention exists where this bit receives both a request to set and a request to clear on the same cycle, regardless of the source of either, this bit will be set and the request to clear will be ignored. In this case the overflow bit in the ADCSOCOVF1 register will not be affected regardless of whether this bit was previously set or not. Reset type: SYSRSn
0	SOC0	R	0h	SOC0 Start of Conversion Flag. Indicates the state of SOC0 conversions. 0 No sample pending for SOC0. 1 Trigger has been received and sample is pending for SOC0. This bit will be automatically cleared when the SOC0 conversion is started. If contention exists where this bit receives both a request to set and a request to clear on the same cycle, regardless of the source of either, this bit will be set and the request to clear will be ignored. In this case the overflow bit in the ADCSOCOVF1 register will not be affected regardless of whether this bit was previously set or not. Reset type: SYSRSn

12 ADCSOCOVF1 Register (Offset = 44h) [Reset = 00000000h]

ADCSOCOVF1 is shown in Figure 16-35 and described in Table 16-26.

Return to the Summary Table.

ADC SOC Overflow 1 Register

Figure 16-35. ADCSOCOVF1 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
SOC15OVF	SOC14OVF	SOC13OVF	SOC12OVF	SOC11OVF	SOC10OVF	SOC9OVF	SOC8OVF
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
7	6	5	4	3	2	1	0
SOC7OVF	SOC6OVF	SOC5OVF	SOC4OVF	SOC3OVF	SOC2OVF	SOC1OVF	SOC0OVF
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

Table 16-38. ADCSOCOVF1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15	SOC15OVF	R	0h	SOC15 Start of Conversion Overflow Flag. Indicates an SOC15 event was generated in hardware while an existing SOC15 event was already pending. 0 No SOC15 event overflow. 1 SOC15 event overflow. An overflow condition does not stop SOC15 events from being processed. It simply is an indication that a hardware trigger was missed. Reset type: SYSRSn
14	SOC14OVF	R	0h	SOC14 Start of Conversion Overflow Flag. Indicates an SOC14 event was generated in hardware while an existing SOC14 event was already pending. 0 No SOC14 event overflow. 1 SOC14 event overflow. An overflow condition does not stop SOC14 events from being processed. It simply is an indication that a hardware trigger was missed. Reset type: SYSRSn
13	SOC13OVF	R	0h	SOC13 Start of Conversion Overflow Flag. Indicates an SOC13 event was generated in hardware while an existing SOC13 event was already pending. 0 No SOC13 event overflow. 1 SOC13 event overflow. An overflow condition does not stop SOC13 events from being processed. It simply is an indication that a hardware trigger was missed. Reset type: SYSRSn

Table 16-38. ADCSOCOVF1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
12	SOC12OVF	R	0h	SOC12 Start of Conversion Overflow Flag. Indicates an SOC12 event was generated in hardware while an existing SOC12 event was already pending. 0 No SOC12 event overflow. 1 SOC12 event overflow. An overflow condition does not stop SOC12 events from being processed. It simply is an indication that a hardware trigger was missed. Reset type: SYSRSn
11	SOC11OVF	R	0h	SOC11 Start of Conversion Overflow Flag. Indicates an SOC11 event was generated in hardware while an existing SOC11 event was already pending. 0 No SOC11 event overflow. 1 SOC11 event overflow. An overflow condition does not stop SOC11 events from being processed. It simply is an indication that a hardware trigger was missed. Reset type: SYSRSn
10	SOC10OVF	R	0h	SOC10 Start of Conversion Overflow Flag. Indicates an SOC10 event was generated in hardware while an existing SOC10 event was already pending. 0 No SOC10 event overflow. 1 SOC10 event overflow. An overflow condition does not stop SOC10 events from being processed. It simply is an indication that a hardware trigger was missed. Reset type: SYSRSn
9	SOC9OVF	R	0h	SOC9 Start of Conversion Overflow Flag. Indicates an SOC9 event was generated in hardware while an existing SOC9 event was already pending. 0 No SOC9 event overflow. 1 SOC9 event overflow. An overflow condition does not stop SOC9 events from being processed. It simply is an indication that a hardware trigger was missed. Reset type: SYSRSn
8	SOC8OVF	R	0h	SOC8 Start of Conversion Overflow Flag. Indicates an SOC8 event was generated in hardware while an existing SOC8 event was already pending. 0 No SOC8 event overflow. 1 SOC8 event overflow. An overflow condition does not stop SOC8 events from being processed. It simply is an indication that a hardware trigger was missed. Reset type: SYSRSn
7	SOC7OVF	R	0h	SOC7 Start of Conversion Overflow Flag. Indicates an SOC7 event was generated in hardware while an existing SOC7 event was already pending. 0 No SOC7 event overflow. 1 SOC7 event overflow. An overflow condition does not stop SOC7 events from being processed. It simply is an indication that a hardware trigger was missed. Reset type: SYSRSn

Table 16-38. ADCSOCOVF1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
6	SOC6OVF	R	0h	SOC6 Start of Conversion Overflow Flag. Indicates an SOC6 event was generated in hardware while an existing SOC6 event was already pending. 0 No SOC6 event overflow. 1 SOC6 event overflow. An overflow condition does not stop SOC6 events from being processed. It simply is an indication that a hardware trigger was missed. Reset type: SYSRSn
5	SOC5OVF	R	0h	SOC5 Start of Conversion Overflow Flag. Indicates an SOC5 event was generated in hardware while an existing SOC5 event was already pending. 0 No SOC5 event overflow. 1 SOC5 event overflow. An overflow condition does not stop SOC5 events from being processed. It simply is an indication that a hardware trigger was missed. Reset type: SYSRSn
4	SOC4OVF	R	0h	SOC4 Start of Conversion Overflow Flag. Indicates an SOC4 event was generated in hardware while an existing SOC4 event was already pending. 0 No SOC4 event overflow. 1 SOC4 event overflow. An overflow condition does not stop SOC4 events from being processed. It simply is an indication that a hardware trigger was missed. Reset type: SYSRSn
3	SOC3OVF	R	0h	SOC3 Start of Conversion Overflow Flag. Indicates an SOC3 event was generated in hardware while an existing SOC3 event was already pending. 0 No SOC3 event overflow. 1 SOC3 event overflow. An overflow condition does not stop SOC3 events from being processed. It simply is an indication that a hardware trigger was missed. Reset type: SYSRSn
2	SOC2OVF	R	0h	SOC2 Start of Conversion Overflow Flag. Indicates an SOC2 event was generated in hardware while an existing SOC2 event was already pending. 0 No SOC2 event overflow. 1 SOC2 event overflow. An overflow condition does not stop SOC2 events from being processed. It simply is an indication that a hardware trigger was missed. Reset type: SYSRSn
1	SOC1OVF	R	0h	SOC1 Start of Conversion Overflow Flag. Indicates an SOC1 event was generated in hardware while an existing SOC1 event was already pending. 0 No SOC1 event overflow. 1 SOC1 event overflow. An overflow condition does not stop SOC1 events from being processed. It simply is an indication that a hardware trigger was missed. Reset type: SYSRSn

Table 16-38. ADCSOCOVF1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
0	SOC0OVF	R	0h	SOC0 Start of Conversion Overflow Flag. Indicates an SOC0 event was generated in hardware while an existing SOC0 event was already pending. 0 No SOC0 event overflow. 1 SOC0 event overflow. An overflow condition does not stop SOC0 events from being processed. It simply is an indication that a hardware trigger was missed. Reset type: SYSRSn

13 ADCSOCOVFCLR1 Register (Offset = 48h) [Reset = 00000000h]

ADCSOCOVFCLR1 is shown in Figure 16-36 and described in Table 16-27.

Return to the Summary Table.

ADC SOC Overflow Clear 1 Register

Figure 16-36. ADCSOCOVFCLR1 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
SOC15OVF	SOC14OVF	SOC13OVF	SOC12OVF	SOC11OVF	SOC10OVF	SOC9OVF	SOC8OVF
R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h
7	6	5	4	3	2	1	0
SOC7OVF	SOC6OVF	SOC5OVF	SOC4OVF	SOC3OVF	SOC2OVF	SOC1OVF	SOC0OVF
R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h

Table 16-40. ADCSOCOVFCLR1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15	SOC15OVF	R-0/W1C	0h	SOC15 Clear Start of Conversion Overflow Bit. Writing a 1 will clear the SOC15 overflow flag in the ADCSOCOVF1 register. Writes of 0 are ignored. Reads will always return a 0. 0 No action. 1 Clear SOC15 overflow flag. If software tries to set this bit on the same clock cycle that hardware tries to set the overflow bit in the ADCSOCOVF1 register, then hardware has priority and the ADCSOCOVF1 bit will be set.. Reset type: SYSRSn
14	SOC14OVF	R-0/W1C	0h	SOC14 Clear Start of Conversion Overflow Bit. Writing a 1 will clear the SOC14 overflow flag in the ADCSOCOVF1 register. Writes of 0 are ignored. Reads will always return a 0. 0 No action. 1 Clear SOC14 overflow flag. If software tries to set this bit on the same clock cycle that hardware tries to set the overflow bit in the ADCSOCOVF1 register, then hardware has priority and the ADCSOCOVF1 bit will be set.. Reset type: SYSRSn
13	SOC13OVF	R-0/W1C	0h	SOC13 Clear Start of Conversion Overflow Bit. Writing a 1 will clear the SOC13 overflow flag in the ADCSOCOVF1 register. Writes of 0 are ignored. Reads will always return a 0. 0 No action. 1 Clear SOC13 overflow flag. If software tries to set this bit on the same clock cycle that hardware tries to set the overflow bit in the ADCSOCOVF1 register, then hardware has priority and the ADCSOCOVF1 bit will be set.. Reset type: SYSRSn

Table 16-40. ADCSOCOVFCLR1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
12	SOC12OVF	R-0/W1C	0h	SOC12 Clear Start of Conversion Overflow Bit. Writing a 1 will clear the SOC12 overflow flag in the ADCSOCOVF1 register. Writes of 0 are ignored. Reads will always return a 0. 0 No action. 1 Clear SOC12 overflow flag. If software tries to set this bit on the same clock cycle that hardware tries to set the overflow bit in the ADCSOCOVF1 register, then hardware has priority and the ADCSOCOVF1 bit will be set.. Reset type: SYSRSn
11	SOC11OVF	R-0/W1C	0h	SOC11 Clear Start of Conversion Overflow Bit. Writing a 1 will clear the SOC11 overflow flag in the ADCSOCOVF1 register. Writes of 0 are ignored. Reads will always return a 0. 0 No action. 1 Clear SOC11 overflow flag. If software tries to set this bit on the same clock cycle that hardware tries to set the overflow bit in the ADCSOCOVF1 register, then hardware has priority and the ADCSOCOVF1 bit will be set.. Reset type: SYSRSn
10	SOC10OVF	R-0/W1C	0h	SOC10 Clear Start of Conversion Overflow Bit. Writing a 1 will clear the SOC10 overflow flag in the ADCSOCOVF1 register. Writes of 0 are ignored. Reads will always return a 0. 0 No action. 1 Clear SOC10 overflow flag. If software tries to set this bit on the same clock cycle that hardware tries to set the overflow bit in the ADCSOCOVF1 register, then hardware has priority and the ADCSOCOVF1 bit will be set.. Reset type: SYSRSn
9	SOC9OVF	R-0/W1C	0h	SOC9 Clear Start of Conversion Overflow Bit. Writing a 1 will clear the SOC9 overflow flag in the ADCSOCOVF1 register. Writes of 0 are ignored. Reads will always return a 0. 0 No action. 1 Clear SOC9 overflow flag. If software tries to set this bit on the same clock cycle that hardware tries to set the overflow bit in the ADCSOCOVF1 register, then hardware has priority and the ADCSOCOVF1 bit will be set.. Reset type: SYSRSn
8	SOC8OVF	R-0/W1C	0h	SOC8 Clear Start of Conversion Overflow Bit. Writing a 1 will clear the SOC8 overflow flag in the ADCSOCOVF1 register. Writes of 0 are ignored. Reads will always return a 0. 0 No action. 1 Clear SOC8 overflow flag. If software tries to set this bit on the same clock cycle that hardware tries to set the overflow bit in the ADCSOCOVF1 register, then hardware has priority and the ADCSOCOVF1 bit will be set.. Reset type: SYSRSn
7	SOC7OVF	R-0/W1C	0h	SOC7 Clear Start of Conversion Overflow Bit. Writing a 1 will clear the SOC7 overflow flag in the ADCSOCOVF1 register. Writes of 0 are ignored. Reads will always return a 0. 0 No action. 1 Clear SOC7 overflow flag. If software tries to set this bit on the same clock cycle that hardware tries to set the overflow bit in the ADCSOCOVF1 register, then hardware has priority and the ADCSOCOVF1 bit will be set.. Reset type: SYSRSn

Table 16-40. ADCSOCOVFCLR1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
6	SOC6OVF	R-0/W1C	0h	SOC6 Clear Start of Conversion Overflow Bit. Writing a 1 will clear the SOC6 overflow flag in the ADCSOCOVF1 register. Writes of 0 are ignored. Reads will always return a 0. 0 No action. 1 Clear SOC6 overflow flag. If software tries to set this bit on the same clock cycle that hardware tries to set the overflow bit in the ADCSOCOVF1 register, then hardware has priority and the ADCSOCOVF1 bit will be set.. Reset type: SYSRSn
5	SOC5OVF	R-0/W1C	0h	SOC5 Clear Start of Conversion Overflow Bit. Writing a 1 will clear the SOC5 overflow flag in the ADCSOCOVF1 register. Writes of 0 are ignored. Reads will always return a 0. 0 No action. 1 Clear SOC5 overflow flag. If software tries to set this bit on the same clock cycle that hardware tries to set the overflow bit in the ADCSOCOVF1 register, then hardware has priority and the ADCSOCOVF1 bit will be set.. Reset type: SYSRSn
4	SOC4OVF	R-0/W1C	0h	SOC4 Clear Start of Conversion Overflow Bit. Writing a 1 will clear the SOC4 overflow flag in the ADCSOCOVF1 register. Writes of 0 are ignored. Reads will always return a 0. 0 No action. 1 Clear SOC4 overflow flag. If software tries to set this bit on the same clock cycle that hardware tries to set the overflow bit in the ADCSOCOVF1 register, then hardware has priority and the ADCSOCOVF1 bit will be set.. Reset type: SYSRSn
3	SOC3OVF	R-0/W1C	0h	SOC3 Clear Start of Conversion Overflow Bit. Writing a 1 will clear the SOC3 overflow flag in the ADCSOCOVF1 register. Writes of 0 are ignored. Reads will always return a 0. 0 No action. 1 Clear SOC3 overflow flag. If software tries to set this bit on the same clock cycle that hardware tries to set the overflow bit in the ADCSOCOVF1 register, then hardware has priority and the ADCSOCOVF1 bit will be set.. Reset type: SYSRSn
2	SOC2OVF	R-0/W1C	0h	SOC2 Clear Start of Conversion Overflow Bit. Writing a 1 will clear the SOC2 overflow flag in the ADCSOCOVF1 register. Writes of 0 are ignored. Reads will always return a 0. 0 No action. 1 Clear SOC2 overflow flag. If software tries to set this bit on the same clock cycle that hardware tries to set the overflow bit in the ADCSOCOVF1 register, then hardware has priority and the ADCSOCOVF1 bit will be set.. Reset type: SYSRSn
1	SOC1OVF	R-0/W1C	0h	SOC1 Clear Start of Conversion Overflow Bit. Writing a 1 will clear the SOC1 overflow flag in the ADCSOCOVF1 register. Writes of 0 are ignored. Reads will always return a 0. 0 No action. 1 Clear SOC1 overflow flag. If software tries to set this bit on the same clock cycle that hardware tries to set the overflow bit in the ADCSOCOVF1 register, then hardware has priority and the ADCSOCOVF1 bit will be set.. Reset type: SYSRSn

Table 16-40. ADCSOCOVFCLR1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
0	SOC0OVF	R-0/W1C	0h	SOC0 Clear Start of Conversion Overflow Bit. Writing a 1 will clear the SOC0 overflow flag in the ADCSOCOVF1 register. Writes of 0 are ignored. Reads will always return a 0. 0 No action. 1 Clear SOC0 overflow flag. If software tries to set this bit on the same clock cycle that hardware tries to set the overflow bit in the ADCSOCOVF1 register, then hardware has priority and the ADCSOCOVF1 bit will be set.. Reset type: SYSRSn

14 ADCSOC0CTL Register (Offset = 4Ch) [Reset = 00000000h]

ADCSOC0CTL is shown in Figure 16-37 and described in Table 16-28.

Return to the Summary Table.

ADC SOC0 Control Register

Figure 16-37. ADCSOC0CTL Register

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

Table 16-42. ADCSOC0CTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	RESERVED	R	0h	Reserved
25	COMPEN	R/W	0h	SOC0 Threshold comparator enable. Reset type: SYSRSn
24-20	RESERVED	R	0h	Reserved
19-15	CHSEL	R/W	0h	SOC0 Channel Select. Selects the channel to be converted when SOC0 is received by the ADC. 00h ADCIN0 01h ADCIN1 02h ADCIN2 03h ADCIN3 ... 1Dh ADCIN29 1Eh ADCIN30 1Fh ADCIN31 Reset type: SYSRSn
14-0	RESERVED	R	0h	Reserved

ADVANCE INFORMATION

15 ADCSOC1CTL Register (Offset = 50h) [Reset = 00000000h]

ADCSOC1CTL is shown in Figure 16-38 and described in Table 16-29.

Return to the Summary Table.

ADC SOC1 Control Register

Figure 16-38. ADCSOC1CTL Register

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

Table 16-44. ADCSOC1CTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	RESERVED	R	0h	Reserved
25	COMPEN	R/W	0h	SOC1 Threshold comparator enable. Reset type: SYSRSn
24-20	RESERVED	R	0h	Reserved
19-15	CHSEL	R/W	0h	SOC1 Channel Select. Selects the channel to be converted when SOC1 is received by the ADC. 00h ADCIN0 01h ADCIN1 02h ADCIN2 03h ADCIN3 ... 1Dh ADCIN29 1Eh ADCIN30 1Fh ADCIN31 Reset type: SYSRSn
14-0	RESERVED	R	0h	Reserved

ADVANCE INFORMATION

16 ADCSOC2CTL Register (Offset = 54h) [Reset = 00000000h]

ADCSOC2CTL is shown in Figure 16-39 and described in Table 16-30.

Return to the Summary Table.

ADC SOC2 Control Register

Figure 16-39. ADCSOC2CTL Register

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

Table 16-46. ADCSOC2CTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	RESERVED	R	0h	Reserved
25	COMPEN	R/W	0h	SOC2 Threshold comparator enable. Reset type: SYSRSn
24-20	RESERVED	R	0h	Reserved
19-15	CHSEL	R/W	0h	SOC2 Channel Select. Selects the channel to be converted when SOC2 is received by the ADC. 00h ADCIN0 01h ADCIN1 02h ADCIN2 03h ADCIN3 ... 1Dh ADCIN29 1Eh ADCIN30 1Fh ADCIN31 Reset type: SYSRSn
14-0	RESERVED	R	0h	Reserved

ADVANCE INFORMATION

17 ADCSOC3CTL Register (Offset = 58h) [Reset = 00000000h]

ADCSOC3CTL is shown in Figure 16-40 and described in Table 16-31.

Return to the Summary Table.

ADC SOC3 Control Register

Figure 16-40. ADCSOC3CTL Register

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

Table 16-48. ADCSOC3CTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	RESERVED	R	0h	Reserved
25	COMPEN	R/W	0h	SOC3 Threshold comparator enable. Reset type: SYSRSn
24-20	RESERVED	R	0h	Reserved
19-15	CHSEL	R/W	0h	SOC3 Channel Select. Selects the channel to be converted when SOC3 is received by the ADC. 00h ADCIN0 01h ADCIN1 02h ADCIN2 03h ADCIN3 ... 1Dh ADCIN29 1Eh ADCIN30 1Fh ADCIN31 Reset type: SYSRSn
14-0	RESERVED	R	0h	Reserved

ADVANCE INFORMATION

18 ADCSOC4CTL Register (Offset = 5Ch) [Reset = 00000000h]

ADCSOC4CTL is shown in Figure 16-41 and described in Table 16-32.

Return to the Summary Table.

ADC SOC4 Control Register

Figure 16-41. ADCSOC4CTL Register

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

Table 16-50. ADCSOC4CTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	RESERVED	R	0h	Reserved
25	COMPEN	R/W	0h	SOC4 Threshold comparator enable. Reset type: SYSRSn
24-20	RESERVED	R	0h	Reserved
19-15	CHSEL	R/W	0h	SOC4 Channel Select. Selects the channel to be converted when SOC4 is received by the ADC. 00h ADCIN0 01h ADCIN1 02h ADCIN2 03h ADCIN3 ... 1Dh ADCIN29 1Eh ADCIN30 1Fh ADCIN31 Reset type: SYSRSn
14-0	RESERVED	R	0h	Reserved

ADVANCE INFORMATION

19 ADCSOC5CTL Register (Offset = 60h) [Reset = 00000000h]

ADCSOC5CTL is shown in Figure 16-42 and described in Table 16-33.

Return to the Summary Table.

ADC SOC5 Control Register

Figure 16-42. ADCSOC5CTL Register

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

Table 16-52. ADCSOC5CTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	RESERVED	R	0h	Reserved
25	COMPEN	R/W	0h	SOC5 Threshold comparator enable. Reset type: SYSRSn
24-20	RESERVED	R	0h	Reserved
19-15	CHSEL	R/W	0h	SOC5 Channel Select. Selects the channel to be converted when SOC5 is received by the ADC. 00h ADCIN0 01h ADCIN1 02h ADCIN2 03h ADCIN3 ... 1Dh ADCIN29 1Eh ADCIN30 1Fh ADCIN31 Reset type: SYSRSn
14-0	RESERVED	R	0h	Reserved

ADVANCE INFORMATION

20 ADCSOC6CTL Register (Offset = 64h) [Reset = 00000000h]

ADCSOC6CTL is shown in Figure 16-43 and described in Table 16-34.

Return to the Summary Table.

ADC SOC6 Control Register

Figure 16-43. ADCSOC6CTL Register

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

Table 16-54. ADCSOC6CTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	RESERVED	R	0h	Reserved
25	COMPEN	R/W	0h	SOC6 Threshold comparator enable. Reset type: SYSRSn
24-20	RESERVED	R	0h	Reserved
19-15	CHSEL	R/W	0h	SOC6 Channel Select. Selects the channel to be converted when SOC6 is received by the ADC. 00h ADCIN0 01h ADCIN1 02h ADCIN2 03h ADCIN3 ... 1Dh ADCIN29 1Eh ADCIN30 1Fh ADCIN31 Reset type: SYSRSn
14-0	RESERVED	R	0h	Reserved

ADVANCE INFORMATION

21 ADCSOC7CTL Register (Offset = 68h) [Reset = 00000000h]

ADCSOC7CTL is shown in Figure 16-44 and described in Table 16-35.

Return to the Summary Table.

ADC SOC7 Control Register

Figure 16-44. ADCSOC7CTL Register

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

Table 16-56. ADCSOC7CTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	RESERVED	R	0h	Reserved
25	COMPEN	R/W	0h	SOC7 Threshold comparator enable. Reset type: SYSRSn
24-20	RESERVED	R	0h	Reserved
19-15	CHSEL	R/W	0h	SOC7 Channel Select. Selects the channel to be converted when SOC7 is received by the ADC. 00h ADCIN0 01h ADCIN1 02h ADCIN2 03h ADCIN3 ... 1Dh ADCIN29 1Eh ADCIN30 1Fh ADCIN31 Reset type: SYSRSn
14-0	RESERVED	R	0h	Reserved

ADVANCE INFORMATION

22 ADCSOC8CTL Register (Offset = 6Ch) [Reset = 00000000h]

ADCSOC8CTL is shown in Figure 16-45 and described in Table 16-36.

Return to the Summary Table.

ADC SOC8 Control Register

Figure 16-45. ADCSOC8CTL Register

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

Table 16-58. ADCSOC8CTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	RESERVED	R	0h	Reserved
25	COMPEN	R/W	0h	SOC8 Threshold comparator enable. Reset type: SYSRSn
24-20	RESERVED	R	0h	Reserved
19-15	CHSEL	R/W	0h	SOC8 Channel Select. Selects the channel to be converted when SOC8 is received by the ADC. 00h ADCIN0 01h ADCIN1 02h ADCIN2 03h ADCIN3 ... 1Dh ADCIN29 1Eh ADCIN30 1Fh ADCIN31 Reset type: SYSRSn
14-0	RESERVED	R	0h	Reserved

ADVANCE INFORMATION

23 ADCSOC9CTL Register (Offset = 70h) [Reset = 00000000h]

ADCSOC9CTL is shown in Figure 16-46 and described in Table 16-37.

Return to the Summary Table.

ADC SOC9 Control Register

Figure 16-46. ADCSOC9CTL Register

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

Table 16-60. ADCSOC9CTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	RESERVED	R	0h	Reserved
25	COMPEN	R/W	0h	SOC9 Threshold comparator enable. Reset type: SYSRSn
24-20	RESERVED	R	0h	Reserved
19-15	CHSEL	R/W	0h	SOC9 Channel Select. Selects the channel to be converted when SOC9 is received by the ADC. 00h ADCIN0 01h ADCIN1 02h ADCIN2 03h ADCIN3 ... 1Dh ADCIN29 1Eh ADCIN30 1Fh ADCIN31 Reset type: SYSRSn
14-0	RESERVED	R	0h	Reserved

ADVANCE INFORMATION

24 ADCSOC10CTL Register (Offset = 74h) [Reset = 00000000h]

ADCSOC10CTL is shown in Figure 16-47 and described in Table 16-38.

Return to the Summary Table.

ADC SOC10 Control Register

Figure 16-47. ADCSOC10CTL Register

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

Table 16-62. ADCSOC10CTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	RESERVED	R	0h	Reserved
25	COMPEN	R/W	0h	SOC10 Threshold comparator enable. Reset type: SYSRSn
24-20	RESERVED	R	0h	Reserved
19-15	CHSEL	R/W	0h	SOC10 Channel Select. Selects the channel to be converted when SOC10 is received by the ADC. 00h ADCIN0 01h ADCIN1 02h ADCIN2 03h ADCIN3 ... 1Dh ADCIN29 1Eh ADCIN30 1Fh ADCIN31 Reset type: SYSRSn
14-0	RESERVED	R	0h	Reserved

ADVANCE INFORMATION

25 ADCSOC11CTL Register (Offset = 78h) [Reset = 00000000h]

ADCSOC11CTL is shown in Figure 16-48 and described in Table 16-39.

Return to the Summary Table.

ADC SOC11 Control Register

Figure 16-48. ADCSOC11CTL Register

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

Table 16-64. ADCSOC11CTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	RESERVED	R	0h	Reserved
25	COMPEN	R/W	0h	SOC11 Threshold comparator enable. Reset type: SYSRSn
24-20	RESERVED	R	0h	Reserved
19-15	CHSEL	R/W	0h	SOC11 Channel Select. Selects the channel to be converted when SOC11 is received by the ADC. 00h ADCIN0 01h ADCIN1 02h ADCIN2 03h ADCIN3 ... 1Dh ADCIN29 1Eh ADCIN30 1Fh ADCIN31 Reset type: SYSRSn
14-0	RESERVED	R	0h	Reserved

ADVANCE INFORMATION

26 ADCSOC12CTL Register (Offset = 7Ch) [Reset = 00000000h]

ADCSOC12CTL is shown in Figure 16-49 and described in Table 16-40.

Return to the Summary Table.

ADC SOC12 Control Register

Figure 16-49. ADCSOC12CTL Register

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

Table 16-66. ADCSOC12CTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	RESERVED	R	0h	Reserved
25	COMPEN	R/W	0h	SOC12 Threshold comparator enable. Reset type: SYSRSn
24-20	RESERVED	R	0h	Reserved
19-15	CHSEL	R/W	0h	SOC12 Channel Select. Selects the channel to be converted when SOC12 is received by the ADC. 00h ADCIN0 01h ADCIN1 02h ADCIN2 03h ADCIN3 ... 1Dh ADCIN29 1Eh ADCIN30 1Fh ADCIN31 Reset type: SYSRSn
14-0	RESERVED	R	0h	Reserved

27 ADCSOC13CTL Register (Offset = 80h) [Reset = 00000000h]

ADCSOC13CTL is shown in Figure 16-50 and described in Table 16-41.

Return to the Summary Table.

ADC SOC13 Control Register

Figure 16-50. ADCSOC13CTL Register

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

Table 16-68. ADCSOC13CTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	RESERVED	R	0h	Reserved
25	COMPEN	R/W	0h	SOC13 Threshold comparator enable. Reset type: SYSRSn
24-20	RESERVED	R	0h	Reserved
19-15	CHSEL	R/W	0h	SOC13 Channel Select. Selects the channel to be converted when SOC13 is received by the ADC. 00h ADCIN0 01h ADCIN1 02h ADCIN2 03h ADCIN3 ... 1Dh ADCIN29 1Eh ADCIN30 1Fh ADCIN31 Reset type: SYSRSn
14-0	RESERVED	R	0h	Reserved

28 ADCSOC14CTL Register (Offset = 84h) [Reset = 00000000h]

ADCSOC14CTL is shown in Figure 16-51 and described in Table 16-42.

Return to the Summary Table.

ADC SOC14 Control Register

Figure 16-51. ADCSOC14CTL Register

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

Table 16-70. ADCSOC14CTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	RESERVED	R	0h	Reserved
25	COMPEN	R/W	0h	SOC14 Threshold comparator enable. Reset type: SYSRSn
24-20	RESERVED	R	0h	Reserved
19-15	CHSEL	R/W	0h	SOC14 Channel Select. Selects the channel to be converted when SOC14 is received by the ADC. 00h ADCIN0 01h ADCIN1 02h ADCIN2 03h ADCIN3 ... 1Dh ADCIN29 1Eh ADCIN30 1Fh ADCIN31 Reset type: SYSRSn
14-0	RESERVED	R	0h	Reserved

29 ADCSOC15CTL Register (Offset = 88h) [Reset = 00000000h]

ADCSOC15CTL is shown in Figure 16-52 and described in Table 16-43.

Return to the Summary Table.

ADC SOC15 Control Register

Figure 16-52. ADCSOC15CTL Register

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

Table 16-72. ADCSOC15CTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	RESERVED	R	0h	Reserved
25	COMPEN	R/W	0h	SOC15 Threshold comparator enable. Reset type: SYSRSn
24-20	RESERVED	R	0h	Reserved
19-15	CHSEL	R/W	0h	SOC15 Channel Select. Selects the channel to be converted when SOC15 is received by the ADC. 00h ADCIN0 01h ADCIN1 02h ADCIN2 03h ADCIN3 ... 1Dh ADCIN29 1Eh ADCIN30 1Fh ADCIN31 Reset type: SYSRSn
14-0	RESERVED	R	0h	Reserved

ADVANCE INFORMATION

30 ADCEVTSTAT Register (Offset = CCh) [Reset = 00000000h]

ADCEVTSTAT is shown in Figure 16-53 and described in Table 16-44.

Return to the Summary Table.

ADC Event Status Register

Figure 16-53. ADCEVTSTAT Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
PPB4INLIMIT	PPB4ZERO	PPB4TRIPLO	PPB4TRIPHI	PPB3INLIMIT	PPB3ZERO	PPB3TRIPLO	PPB3TRIPHI
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
7	6	5	4	3	2	1	0
PPB2INLIMIT	PPB2ZERO	PPB2TRIPLO	PPB2TRIPHI	PPB1INLIMIT	PPB1ZERO	PPB1TRIPLO	PPB1TRIPHI
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

Table 16-74. ADCEVTSTAT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15	PPB4INLIMIT	R	0h	Post Processing Block 4 Within trip limit Flag. When set indicates a digital compare within limit event has occurred. This will be set when the PPB result is either in between or equal to the TRIPHI and TRIPLO thresholds. Note: these bits are set even when the corresponding enable in ADCEVTINTSEL is not set. Because of this, an ISR may need to examine both the ADCEVTSTAT and ADCEVTINTSEL registers to determine the source of the interrupt. Note: If software sets the clear bit on the same cycle that hardware is trying to set the flag bit, then hardware has priority Reset type: SYSRSn
14	PPB4ZERO	R	0h	Post Processing Block 4 Zero Crossing Flag. When set indicates the ADCPPB4RESULT register has changed sign. This bit is gated by EOC signal. Note: these bits are set even when the corresponding enable in ADCEVTINTSEL is not set. Because of this, an ISR may need to examine both the ADCEVTSTAT and ADCEVTINTSEL registers to determine the source of the interrupt. Note: If software sets the clear bit on the same cycle that hardware is trying to set the flag bit, then hardware has priority Reset type: SYSRSn
13	PPB4TRIPLO	R	0h	Post Processing Block 4 Trip Low Flag. When set indicates a digital compare trip low event has occurred. Note: these bits are set even when the corresponding enable in ADCEVTINTSEL is not set. Because of this, an ISR may need to examine both the ADCEVTSTAT and ADCEVTINTSEL registers to determine the source of the interrupt. Note: If software sets the clear bit on the same cycle that hardware is trying to set the flag bit, then hardware has priority Reset type: SYSRSn

Table 16-74. ADCEVTSTAT Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
12	PPB4TRIPHI	R	0h	Post Processing Block 4 Trip High Flag. When set indicates a digital compare trip high event has occurred. Note: these bits are set even when the corresponding enable in ADCEVTINTSEL is not set. Because of this, an ISR may need to examine both the ADCEVTSTAT and ADCEVTINTSEL registers to determine the source of the interrupt. Note: If software sets the clear bit on the same cycle that hardware is trying to set the flag bit, then hardware has priority Reset type: SYSRSn
11	PPB3INLIMIT	R	0h	Post Processing Block 3 Within trip limit Flag. When set indicates a digital compare within limit event has occurred. This will be set when the PPB result is either in between or equal to the TRIPHI and TRIPLO thresholds. Note: these bits are set even when the corresponding enable in ADCEVTINTSEL is not set. Because of this, an ISR may need to examine both the ADCEVTSTAT and ADCEVTINTSEL registers to determine the source of the interrupt. Note: If software sets the clear bit on the same cycle that hardware is trying to set the flag bit, then hardware has priority Reset type: SYSRSn
10	PPB3ZERO	R	0h	Post Processing Block 3 Zero Crossing Flag. When set indicates the ADCPPB3RESULT register has changed sign. This bit is gated by EOC signal. Note: these bits are set even when the corresponding enable in ADCEVTINTSEL is not set. Because of this, an ISR may need to examine both the ADCEVTSTAT and ADCEVTINTSEL registers to determine the source of the interrupt. Note: If software sets the clear bit on the same cycle that hardware is trying to set the flag bit, then hardware has priority Reset type: SYSRSn
9	PPB3TRIPLO	R	0h	Post Processing Block 3 Trip Low Flag. When set indicates a digital compare trip low event has occurred. Note: these bits are set even when the corresponding enable in ADCEVTINTSEL is not set. Because of this, an ISR may need to examine both the ADCEVTSTAT and ADCEVTINTSEL registers to determine the source of the interrupt. Note: If software sets the clear bit on the same cycle that hardware is trying to set the flag bit, then hardware has priority Reset type: SYSRSn
8	PPB3TRIPHI	R	0h	Post Processing Block 3 Trip High Flag. When set indicates a digital compare trip high event has occurred. Note: these bits are set even when the corresponding enable in ADCEVTINTSEL is not set. Because of this, an ISR may need to examine both the ADCEVTSTAT and ADCEVTINTSEL registers to determine the source of the interrupt. Note: If software sets the clear bit on the same cycle that hardware is trying to set the flag bit, then hardware has priority Reset type: SYSRSn

Table 16-74. ADCEVTSTAT Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
7	PPB2INLIMIT	R	0h	Post Processing Block 2 Within trip limit Flag. When set indicates a digital compare within limit event has occurred. This will be set when the PPB result is either in between or equal to the TRIPHI and TRIPLO thresholds. Note: these bits are set even when the corresponding enable in ADCEVTINTSEL is not set. Because of this, an ISR may need to examine both the ADCEVTSTAT and ADCEVTINTSEL registers to determine the source of the interrupt. Note: If software sets the clear bit on the same cycle that hardware is trying to set the flag bit, then hardware has priority Reset type: SYSRSn
6	PPB2ZERO	R	0h	Post Processing Block 2 Zero Crossing Flag. When set indicates the ADCPPB2RESULT register has changed sign. This bit is gated by EOC signal. Note: these bits are set even when the corresponding enable in ADCEVTINTSEL is not set. Because of this, an ISR may need to examine both the ADCEVTSTAT and ADCEVTINTSEL registers to determine the source of the interrupt. Note: If software sets the clear bit on the same cycle that hardware is trying to set the flag bit, then hardware has priority Reset type: SYSRSn
5	PPB2TRIPLO	R	0h	Post Processing Block 2 Trip Low Flag. When set indicates a digital compare trip low event has occurred. Note: these bits are set even when the corresponding enable in ADCEVTINTSEL is not set. Because of this, an ISR may need to examine both the ADCEVTSTAT and ADCEVTINTSEL registers to determine the source of the interrupt. Note: If software sets the clear bit on the same cycle that hardware is trying to set the flag bit, then hardware has priority Reset type: SYSRSn
4	PPB2TRIPHI	R	0h	Post Processing Block 2 Trip High Flag. When set indicates a digital compare trip high event has occurred. Note: these bits are set even when the corresponding enable in ADCEVTINTSEL is not set. Because of this, an ISR may need to examine both the ADCEVTSTAT and ADCEVTINTSEL registers to determine the source of the interrupt. Note: If software sets the clear bit on the same cycle that hardware is trying to set the flag bit, then hardware has priority Reset type: SYSRSn
3	PPB1INLIMIT	R	0h	Post Processing Block 1 Within trip limit Flag. When set indicates a digital compare within limit event has occurred. This will be set when the PPB result is either in between or equal to the TRIPHI and TRIPLO thresholds. Note: these bits are set even when the corresponding enable in ADCEVTINTSEL is not set. Because of this, an ISR may need to examine both the ADCEVTSTAT and ADCEVTINTSEL registers to determine the source of the interrupt. Note: If software sets the clear bit on the same cycle that hardware is trying to set the flag bit, then hardware has priority Reset type: SYSRSn

Table 16-74. ADCEVTSTAT Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
2	PPB1ZERO	R	0h	Post Processing Block 1 Zero Crossing Flag. When set indicates the ADCPPB1RESULT register has changed sign. This bit is gated by EOC signal. Note: these bits are set even when the corresponding enable in ADCEVTINTSEL is not set. Because of this, an ISR may need to examine both the ADCEVTSTAT and ADCEVTINTSEL registers to determine the source of the interrupt. Note: If software sets the clear bit on the same cycle that hardware is trying to set the flag bit, then hardware has priority Reset type: SYSRSn
1	PPB1TRIPLO	R	0h	Post Processing Block 1 Trip Low Flag. When set indicates a digital compare trip low event has occurred. Note: these bits are set even when the corresponding enable in ADCEVTINTSEL is not set. Because of this, an ISR may need to examine both the ADCEVTSTAT and ADCEVTINTSEL registers to determine the source of the interrupt. Note: If software sets the clear bit on the same cycle that hardware is trying to set the flag bit, then hardware has priority Reset type: SYSRSn
0	PPB1TRIPHI	R	0h	Post Processing Block 1 Trip High Flag. When set indicates a digital compare trip high event has occurred. Note: these bits are set even when the corresponding enable in ADCEVTINTSEL is not set. Because of this, an ISR may need to examine both the ADCEVTSTAT and ADCEVTINTSEL registers to determine the source of the interrupt. Note: If software sets the clear bit on the same cycle that hardware is trying to set the flag bit, then hardware has priority Reset type: SYSRSn

31 ADCEVTCLR Register (Offset = D0h) [Reset = 00000000h]

ADCEVTCLR is shown in Figure 16-54 and described in Table 16-45.

Return to the Summary Table.

ADC Event Clear Register

Figure 16-54. ADCEVTCLR Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
PPB4INLIMIT	PPB4ZERO	PPB4TRIPLO	PPB4TRIPHI	PPB3INLIMIT	PPB3ZERO	PPB3TRIPLO	PPB3TRIPHI
R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h
7	6	5	4	3	2	1	0
PPB2INLIMIT	PPB2ZERO	PPB2TRIPLO	PPB2TRIPHI	PPB1INLIMIT	PPB1ZERO	PPB1TRIPLO	PPB1TRIPHI
R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h

Table 16-76. ADCEVTCLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15	PPB4INLIMIT	R-0/W1C	0h	Post Processing Block 4 Within trip limit flag Clear, Clears the corresponding within trip limit flag in the ADCEVTSTAT register. Note: If software sets the clear bit on the same cycle that hardware is trying to set the flag bit, then hardware has priority Reset type: SYSRSn
14	PPB4ZERO	R-0/W1C	0h	Post Processing Block 4 Zero Crossing Clear. Clears the corresponding zero crossing flag in the ADCEVTSTAT register. Note: If software sets the clear bit on the same cycle that hardware is trying to set the flag bit, then hardware has priority Reset type: SYSRSn
13	PPB4TRIPLO	R-0/W1C	0h	Post Processing Block 4 Trip Low Clear. Clears the corresponding trip low flag in the ADCEVTSTAT register. Note: If software sets the clear bit on the same cycle that hardware is trying to set the flag bit, then hardware has priority Reset type: SYSRSn
12	PPB4TRIPHI	R-0/W1C	0h	Post Processing Block 4 Trip High Clear. Clears the corresponding trip high flag in the ADCEVTSTAT register. Note: If software sets the clear bit on the same cycle that hardware is trying to set the flag bit, then hardware has priority Reset type: SYSRSn
11	PPB3INLIMIT	R-0/W1C	0h	Post Processing Block 3 Within trip limit flag Clear, Clears the corresponding within trip limit flag in the ADCEVTSTAT register. Note: If software sets the clear bit on the same cycle that hardware is trying to set the flag bit, then hardware has priority Reset type: SYSRSn

Table 16-76. ADCEVTCLR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
10	PPB3ZERO	R-0/W1C	0h	Post Processing Block 3 Zero Crossing Clear. Clears the corresponding zero crossing flag in the ADCEVTSTAT register. Note: If software sets the clear bit on the same cycle that hardware is trying to set the flag bit, then hardware has priority Reset type: SYSRSn
9	PPB3TRIPLO	R-0/W1C	0h	Post Processing Block 3 Trip Low Clear. Clears the corresponding trip low flag in the ADCEVTSTAT register. Note: If software sets the clear bit on the same cycle that hardware is trying to set the flag bit, then hardware has priority Reset type: SYSRSn
8	PPB3TRIPHI	R-0/W1C	0h	Post Processing Block 3 Trip High Clear. Clears the corresponding trip high flag in the ADCEVTSTAT register. Note: If software sets the clear bit on the same cycle that hardware is trying to set the flag bit, then hardware has priority Reset type: SYSRSn
7	PPB2INLIMIT	R-0/W1C	0h	Post Processing Block 2 Within trip limit flag Clear, Clears the corresponding within trip limit flag in the ADCEVTSTAT register. Note: If software sets the clear bit on the same cycle that hardware is trying to set the flag bit, then hardware has priority Reset type: SYSRSn
6	PPB2ZERO	R-0/W1C	0h	Post Processing Block 2 Zero Crossing Clear. Clears the corresponding zero crossing flag in the ADCEVTSTAT register. Note: If software sets the clear bit on the same cycle that hardware is trying to set the flag bit, then hardware has priority Reset type: SYSRSn
5	PPB2TRIPLO	R-0/W1C	0h	Post Processing Block 2 Trip Low Clear. Clears the corresponding trip low flag in the ADCEVTSTAT register. Note: If software sets the clear bit on the same cycle that hardware is trying to set the flag bit, then hardware has priority Reset type: SYSRSn
4	PPB2TRIPHI	R-0/W1C	0h	Post Processing Block 2 Trip High Clear. Clears the corresponding trip high flag in the ADCEVTSTAT register. Note: If software sets the clear bit on the same cycle that hardware is trying to set the flag bit, then hardware has priority Reset type: SYSRSn
3	PPB1INLIMIT	R-0/W1C	0h	Post Processing Block 1 Within trip limit flag Clear, Clears the corresponding within trip limit flag in the ADCEVTSTAT register. Note: If software sets the clear bit on the same cycle that hardware is trying to set the flag bit, then hardware has priority Reset type: SYSRSn
2	PPB1ZERO	R-0/W1C	0h	Post Processing Block 1 Zero Crossing Clear. Clears the corresponding zero crossing flag in the ADCEVTSTAT register. Note: If software sets the clear bit on the same cycle that hardware is trying to set the flag bit, then hardware has priority Reset type: SYSRSn
1	PPB1TRIPLO	R-0/W1C	0h	Post Processing Block 1 Trip Low Clear. Clears the corresponding trip low flag in the ADCEVTSTAT register. Note: If software sets the clear bit on the same cycle that hardware is trying to set the flag bit, then hardware has priority Reset type: SYSRSn

Table 16-76. ADCEVTCLR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
0	PPB1TRIPHI	R-0/W1C	0h	Post Processing Block 1 Trip High Clear. Clears the corresponding trip high flag in the ADCEVTSTAT register. Note: If software sets the clear bit on the same cycle that hardware is trying to set the flag bit, then hardware has priority Reset type: SYSRSn

32 ADCEVTSEL Register (Offset = D4h) [Reset = 00000000h]

ADCEVTSEL is shown in Figure 16-55 and described in Table 16-46.

Return to the Summary Table.

ADC Event Selection Register

Figure 16-55. ADCEVTSEL Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
PPB4INLIMIT	PPB4ZERO	PPB4TRIPLO	PPB4TRIPHI	PPB3INLIMIT	PPB3ZERO	PPB3TRIPLO	PPB3TRIPHI
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
PPB2INLIMIT	PPB2ZERO	PPB2TRIPLO	PPB2TRIPHI	PPB1INLIMIT	PPB1ZERO	PPB1TRIPLO	PPB1TRIPHI
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 16-78. ADCEVTSEL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15	PPB4INLIMIT	R/W	0h	Post Processing Block 4 Within trip limit event enable. Setting this bit allows the corresponding rising zero crossing flag to activate the event signal to the PWM blocks. The flag must be cleared before it can produce additional events to the PWM blocks. Reset type: SYSRSn
14	PPB4ZERO	R/W	0h	Post Processing Block 4 Zero Crossing Event Enable. Setting this bit allows the corresponding rising zero crossing flag to activate the event signal to the PWM blocks. The flag must be cleared before it can produce additional events to the PWM blocks. Reset type: SYSRSn
13	PPB4TRIPLO	R/W	0h	Post Processing Block 4 Trip Low Event Enable. Setting this bit allows the corresponding rising trip low flag to activate the event signal to the PWM blocks. The flag must be cleared before it can produce additional events to the PWM blocks. Reset type: SYSRSn
12	PPB4TRIPHI	R/W	0h	Post Processing Block 4 Trip High Event Enable. Setting this bit allows the corresponding rising trip high flag to activate the event signal to the PWM blocks. The flag must be cleared before it can produce additional events to the PWM blocks. Reset type: SYSRSn
11	PPB3INLIMIT	R/W	0h	Post Processing Block 3 Within trip limit event enable. Setting this bit allows the corresponding rising zero crossing flag to activate the event signal to the PWM blocks. The flag must be cleared before it can produce additional events to the PWM blocks. Reset type: SYSRSn

Table 16-78. ADCEVTSEL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
10	PPB3ZERO	R/W	0h	Post Processing Block 3 Zero Crossing Event Enable. Setting this bit allows the corresponding rising zero crossing flag to activate the event signal to the PWM blocks. The flag must be cleared before it can produce additional events to the PWM blocks. Reset type: SYSRSn
9	PPB3TRIPLO	R/W	0h	Post Processing Block 3 Trip Low Event Enable. Setting this bit allows the corresponding rising trip low flag to activate the event signal to the PWM blocks. The flag must be cleared before it can produce additional events to the PWM blocks. Reset type: SYSRSn
8	PPB3TRIPHI	R/W	0h	Post Processing Block 3 Trip High Event Enable. Setting this bit allows the corresponding rising trip high flag to activate the event signal to the PWM blocks. The flag must be cleared before it can produce additional events to the PWM blocks. Reset type: SYSRSn
7	PPB2INLIMIT	R/W	0h	Post Processing Block 2 Within trip limit event enable. Setting this bit allows the corresponding rising zero crossing flag to activate the event signal to the PWM blocks. The flag must be cleared before it can produce additional events to the PWM blocks. Reset type: SYSRSn
6	PPB2ZERO	R/W	0h	Post Processing Block 2 Zero Crossing Event Enable. Setting this bit allows the corresponding rising zero crossing flag to activate the event signal to the PWM blocks. The flag must be cleared before it can produce additional events to the PWM blocks. Reset type: SYSRSn
5	PPB2TRIPLO	R/W	0h	Post Processing Block 2 Trip Low Event Enable. Setting this bit allows the corresponding rising trip low flag to activate the event signal to the PWM blocks. The flag must be cleared before it can produce additional events to the PWM blocks. Reset type: SYSRSn
4	PPB2TRIPHI	R/W	0h	Post Processing Block 2 Trip High Event Enable. Setting this bit allows the corresponding rising trip high flag to activate the event signal to the PWM blocks. The flag must be cleared before it can produce additional events to the PWM blocks. Reset type: SYSRSn
3	PPB1INLIMIT	R/W	0h	Post Processing Block 1 Within trip limit event enable. Setting this bit allows the corresponding rising zero crossing flag to activate the event signal to the PWM blocks. The flag must be cleared before it can produce additional events to the PWM blocks. Reset type: SYSRSn
2	PPB1ZERO	R/W	0h	Post Processing Block 1 Zero Crossing Event Enable. Setting this bit allows the corresponding rising zero crossing flag to activate the event signal to the PWM blocks. The flag must be cleared before it can produce additional events to the PWM blocks. Reset type: SYSRSn
1	PPB1TRIPLO	R/W	0h	Post Processing Block 1 Trip Low Event Enable. Setting this bit allows the corresponding rising trip low flag to activate the event signal to the PWM blocks. The flag must be cleared before it can produce additional events to the PWM blocks. Reset type: SYSRSn

Table 16-78. ADCEVTSEL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
0	PPB1TRIPHI	R/W	0h	Post Processing Block 1 Trip High Event Enable. Setting this bit allows the corresponding rising trip high flag to activate the event signal to the PWM blocks. The flag must be cleared before it can produce additional events to the PWM blocks. Reset type: SYSRSn

33 ADCEVTINTSEL Register (Offset = D8h) [Reset = 00000000h]

ADCEVTINTSEL is shown in Figure 16-56 and described in Table 16-47.

Return to the Summary Table.

ADC Event Interrupt Selection Register

Figure 16-56. ADCEVTINTSEL Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
PPB4INLIMIT	PPB4ZERO	PPB4TRIPLO	PPB4TRIPHI	PPB3INLIMIT	PPB3ZERO	PPB3TRIPLO	PPB3TRIPHI
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
PPB2INLIMIT	PPB2ZERO	PPB2TRIPLO	PPB2TRIPHI	PPB1INLIMIT	PPB1ZERO	PPB1TRIPLO	PPB1TRIPHI
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 16-80. ADCEVTINTSEL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15	PPB4INLIMIT	R/W	0h	Post Processing Block 4 Within trip limit Interrupt Enable. Setting this bit allows the corresponding rising zero crossing flag to activate the event interrupt signal to the NVIC . The flag must be cleared before it can produce additional interrupts to the NVIC . Reset type: SYSRSn
14	PPB4ZERO	R/W	0h	Post Processing Block 4 Zero Crossing Interrupt Enable. Setting this bit allows the corresponding rising zero crossing flag to activate the event interrupt signal to the NVIC . The flag must be cleared before it can produce additional interrupts to the NVIC . Reset type: SYSRSn
13	PPB4TRIPLO	R/W	0h	Post Processing Block 4 Trip Low Interrupt Enable. Setting this bit allows the corresponding rising trip low flag to activate the event interrupt signal to the NVIC . The flag must be cleared before it can produce additional interrupts to the NVIC . Reset type: SYSRSn
12	PPB4TRIPHI	R/W	0h	Post Processing Block 4 Trip High Interrupt Enable. Setting this bit allows the corresponding rising trip high flag to activate the event interrupt signal to the NVIC . The flag must be cleared before it can produce additional interrupts to the NVIC . Reset type: SYSRSn
11	PPB3INLIMIT	R/W	0h	Post Processing Block 3 Within trip limit Interrupt Enable. Setting this bit allows the corresponding rising zero crossing flag to activate the event interrupt signal to the NVIC . The flag must be cleared before it can produce additional interrupts to the NVIC . Reset type: SYSRSn

Table 16-80. ADCEVTINTSEL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
10	PPB3ZERO	R/W	0h	Post Processing Block 3 Zero Crossing Interrupt Enable. Setting this bit allows the corresponding rising zero crossing flag to activate the event interrupt signal to the NVIC . The flag must be cleared before it can produce additional interrupts to the NVIC . Reset type: SYSRSn
9	PPB3TRIPLO	R/W	0h	Post Processing Block 3 Trip Low Interrupt Enable. Setting this bit allows the corresponding rising trip low flag to activate the event interrupt signal to the NVIC . The flag must be cleared before it can produce additional interrupts to the NVIC . Reset type: SYSRSn
8	PPB3TRIPHI	R/W	0h	Post Processing Block 3 Trip High Interrupt Enable. Setting this bit allows the corresponding rising trip high flag to activate the event interrupt signal to the NVIC . The flag must be cleared before it can produce additional interrupts to the NVIC . Reset type: SYSRSn
7	PPB2INLIMIT	R/W	0h	Post Processing Block 2 Within trip limit Interrupt Enable. Setting this bit allows the corresponding rising zero crossing flag to activate the event interrupt signal to the NVIC . The flag must be cleared before it can produce additional interrupts to the NVIC . Reset type: SYSRSn
6	PPB2ZERO	R/W	0h	Post Processing Block 2 Zero Crossing Interrupt Enable. Setting this bit allows the corresponding rising zero crossing flag to activate the event interrupt signal to the NVIC . The flag must be cleared before it can produce additional interrupts to the NVIC . Reset type: SYSRSn
5	PPB2TRIPLO	R/W	0h	Post Processing Block 2 Trip Low Interrupt Enable. Setting this bit allows the corresponding rising trip low flag to activate the event interrupt signal to the NVIC . The flag must be cleared before it can produce additional interrupts to the NVIC . Reset type: SYSRSn
4	PPB2TRIPHI	R/W	0h	Post Processing Block 2 Trip High Interrupt Enable. Setting this bit allows the corresponding rising trip high flag to activate the event interrupt signal to the NVIC . The flag must be cleared before it can produce additional interrupts to the NVIC . Reset type: SYSRSn
3	PPB1INLIMIT	R/W	0h	Post Processing Block 1 Within trip limit Interrupt Enable. Setting this bit allows the corresponding rising zero crossing flag to activate the event interrupt signal to the NVIC . The flag must be cleared before it can produce additional interrupts to the NVIC . Reset type: SYSRSn
2	PPB1ZERO	R/W	0h	Post Processing Block 1 Zero Crossing Interrupt Enable. Setting this bit allows the corresponding rising zero crossing flag to activate the event interrupt signal to the NVIC . The flag must be cleared before it can produce additional interrupts to the NVIC . Reset type: SYSRSn
1	PPB1TRIPLO	R/W	0h	Post Processing Block 1 Trip Low Interrupt Enable. Setting this bit allows the corresponding rising trip low flag to activate the event interrupt signal to the NVIC . The flag must be cleared before it can produce additional interrupts to the NVIC . Reset type: SYSRSn

Table 16-80. ADCEVTINTSEL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
0	PPB1TRIPHI	R/W	0h	Post Processing Block 1 Trip High Interrupt Enable. Setting this bit allows the corresponding rising trip high flag to activate the event interrupt signal to the NVIC . The flag must be cleared before it can produce additional interrupts to the NVIC . Reset type: SYSRSn

34 ADCREV Register (Offset = E4h) [Reset = 00000006h]

ADCREV is shown in Figure 16-57 and described in Table 16-48.

Return to the Summary Table.

ADC Revision Register

Figure 16-57. ADCREV Register

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

Table 16-82. ADCREV Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	REV	R	0h	ADC Revision. To allow documentation of differences between revisions. First version is labeled as 00h. Reset type: SYSRSn
7-0	TYPE	R	6h	ADC Type. Always set to 6 for this HSADC-12b. Reset type: SYSRSn

35 ADCOFFTRIM Register (Offset = E8h) [Reset = 00000000h]

ADCOFFTRIM is shown in Figure 16-58 and described in Table 16-49.

Return to the Summary Table.

ADC Offset Trim Register 1

Figure 16-58. ADCOFFTRIM Register

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

Table 16-84. ADCOFFTRIM Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	Reserved
7-0	OFFTRIM	R/W	0h	ADC Offset Trim Adjusts the conversion results of the converter up or down to account for offset error in the ADC. A factory trim setting will be loaded during device boot. Offset can be corrected in the range of +7 to -8 LSBs. Value is 16*Offset in 8-bit 2's complement: 7 LSB (16*7) = 112 6 LSB (16*6) = 96 5 LSB (16*5) = 80 4 LSB (16*4) = 64 3 LSB (16*3) = 48 2 LSB (16*2) = 32 1 LSB (16*1) = 16 0 LSB (16*0) = 0 -1 LSB (16*(-1)) = 240 : : -7LSB(16*(-7)) = 144 Reset type: XRSn

36 ADCPPB1CONFIG Register (Offset = 100h) [Reset = 00000000h]

ADCPPB1CONFIG is shown in Figure 16-59 and described in Table 16-50.

Return to the Summary Table.

ADC PPB1 Config Register

Figure 16-59. ADCPPB1CONFIG Register

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

Table 16-86. ADCPPB1CONFIG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	Reserved
7	RESERVED	R/W	0h	Reserved
6	RESERVED	R	0h	Reserved
5	CBCEN	R/W	0h	ADC Post Processing Block Cycle By Cycle Enable. When set, this bit enables the post conversion hardware processing circuit to automatically clear the ADCEVTSTAT on a conversion if the event condition is no longer present. Reset type: SYSRSn
4-0	RESERVED	R	0h	Reserved

37 ADCPPB1TRIPHI Register (Offset = 110h) [Reset = 00000000h]

ADCPPB1TRIPHI is shown in Figure 16-60 and described in Table 16-51.

Return to the Summary Table.

ADC PPB1 Trip High Register

Figure 16-60. ADCPPB1TRIPHI Register

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

Table 16-88. ADCPPB1TRIPHI Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15-0	LIMITHI	R/W	0h	ADC Post Processing Block 1 Trip High Limit. This value sets the digital comparator trip high limit. When comparing to an ADCPPBxRESULT register, the upper bits will be ignored: - TRIPHI[15:13] will be ignored Reset type: SYSRSn

38 ADCPPB1TRIPLO Register (Offset = 114h) [Reset = 00000000h]

ADCPPB1TRIPLO is shown in Figure 16-61 and described in Table 16-52.

Return to the Summary Table.

ADC PPB1 Trip Low/Trigger Time Stamp Register

Figure 16-61. ADCPPB1TRIPLO Register

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

Table 16-90. ADCPPB1TRIPLO Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15-0	LIMITLO	R/W	0h	ADC Post Processing Block 1 Trip High Low Limit. This value sets the digital comparator trip low limit if ADCPPB1TRIPLO.LIMITLO2EN = 1. When comparing to an ADCPPBxRESULT register, the upper bits will be ignored: - TRIPLO[15:13] will be ignored in 12 bit mode Reset type: SYSRSn

39 ADCPPB2CONFIG Register (Offset = 120h) [Reset = 00000000h]

ADCPPB2CONFIG is shown in Figure 16-62 and described in Table 16-53.

Return to the Summary Table.

ADC PPB2 Config Register

Figure 16-62. ADCPPB2CONFIG Register

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

Table 16-92. ADCPPB2CONFIG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	Reserved
7	RESERVED	R/W	0h	Reserved
6	RESERVED	R	0h	Reserved
5	CBCEN	R/W	0h	ADC Post Processing Block Cycle By Cycle Enable. When set, this bit enables the post conversion hardware processing circuit to automatically clear the ADCEVTSTAT on a conversion if the event condition is no longer present. Reset type: SYSRSn
4-0	RESERVED	R	0h	Reserved

40 ADCPPB2TRIPHI Register (Offset = 130h) [Reset = 00000000h]

ADCPPB2TRIPHI is shown in Figure 16-63 and described in Table 16-54.

Return to the Summary Table.

ADC PPB2 Trip High Register

Figure 16-63. ADCPPB2TRIPHI Register

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

Table 16-94. ADCPPB2TRIPHI Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15-0	LIMITHI	R/W	0h	ADC Post Processing Block 2 Trip High Limit. This value sets the digital comparator trip high limit. When comparing to an ADCPPBxRESULT register, the upper bits will be ignored: - TRIPHI[15:13] will be ignored Reset type: SYSRSn

ADVANCE INFORMATION

41 ADCPPB2TRIPLO Register (Offset = 134h) [Reset = 00000000h]

ADCPPB2TRIPLO is shown in Figure 16-64 and described in Table 16-55.

Return to the Summary Table.

ADC PPB2 Trip Low/Trigger Time Stamp Register

Figure 16-64. ADCPPB2TRIPLO Register

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

Table 16-96. ADCPPB2TRIPLO Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15-0	LIMITLO	R/W	0h	ADC Post Processing Block 2 Trip High Low Limit. This value sets the digital comparator trip low limit if ADCPPB2TRIPLO.LIMITLO2EN = 1. When comparing to an ADCPPBxRESULT register, the upper bits will be ignored: - TRIPLO[15:13] will be ignored in 12 bit mode Reset type: SYSRSn

42 ADCPPB3CONFIG Register (Offset = 140h) [Reset = 00000000h]

ADCPPB3CONFIG is shown in Figure 16-65 and described in Table 16-56.

Return to the Summary Table.

ADC PPB3 Config Register

Figure 16-65. ADCPPB3CONFIG Register

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

Table 16-98. ADCPPB3CONFIG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	Reserved
7	RESERVED	R/W	0h	Reserved
6	RESERVED	R	0h	Reserved
5	CBCEN	R/W	0h	ADC Post Processing Block Cycle By Cycle Enable. When set, this bit enables the post conversion hardware processing circuit to automatically clear the ADCEVTSTAT on a conversion if the event condition is no longer present. Reset type: SYSRSn
4-0	RESERVED	R	0h	Reserved

43 ADCPPB3TRIPHI Register (Offset = 150h) [Reset = 00000000h]

ADCPPB3TRIPHI is shown in Figure 16-66 and described in Table 16-57.

Return to the Summary Table.

ADC PPB3 Trip High Register

Figure 16-66. ADCPPB3TRIPHI Register

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

Table 16-100. ADCPPB3TRIPHI Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15-0	LIMITHI	R/W	0h	ADC Post Processing Block 3 Trip High Limit. This value sets the digital comparator trip high limit. When comparing to an ADCPPBxRESULT register, the upper bits will be ignored: - TRIPHI[15:13] will be ignored Reset type: SYSRSn

ADVANCE INFORMATION

44 ADCPPB3TRIPLO Register (Offset = 154h) [Reset = 00000000h]

ADCPPB3TRIPLO is shown in Figure 16-67 and described in Table 16-58.

Return to the Summary Table.

ADC PPB3 Trip Low/Trigger Time Stamp Register

Figure 16-67. ADCPPB3TRIPLO Register

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

Table 16-102. ADCPPB3TRIPLO Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15-0	LIMITLO	R/W	0h	ADC Post Processing Block 3 Trip High Low Limit. This value sets the digital comparator trip low limit if ADCPPB3TRIPLO.LIMITLO2EN = 1. When comparing to an ADCPPBxRESULT register, the upper bits will be ignored: - TRIPLO[15:13] will be ignored in 12 bit mode Reset type: SYSRSn

45 ADCPPB4CONFIG Register (Offset = 160h) [Reset = 00000000h]

ADCPPB4CONFIG is shown in Figure 16-68 and described in Table 16-59.

Return to the Summary Table.

ADC PPB4 Config Register

Figure 16-68. ADCPPB4CONFIG Register

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

Table 16-104. ADCPPB4CONFIG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	Reserved
7	RESERVED	R/W	0h	Reserved
6	RESERVED	R	0h	Reserved
5	CBCEN	R/W	0h	ADC Post Processing Block Cycle By Cycle Enable. When set, this bit enables the post conversion hardware processing circuit to automatically clear the ADCEVTSTAT on a conversion if the event condition is no longer present. Reset type: SYSRSn
4-0	RESERVED	R	0h	Reserved

46 ADCPPB4TRIPHI Register (Offset = 170h) [Reset = 00000000h]

ADCPPB4TRIPHI is shown in Figure 16-69 and described in Table 16-60.

Return to the Summary Table.

ADC PPB4 Trip High Register

Figure 16-69. ADCPPB4TRIPHI Register

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

Table 16-106. ADCPPB4TRIPHI Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15-0	LIMITHI	R/W	0h	ADC Post Processing Block 4 Trip High Limit. This value sets the digital comparator trip high limit. When comparing to an ADCPPBxRESULT register, the upper bits will be ignored: - TRIPHI[15:13] will be ignored Reset type: SYSRSn

ADVANCE INFORMATION

47 ADCPPB4TRIPLO Register (Offset = 174h) [Reset = 00000000h]

ADCPPB4TRIPLO is shown in Figure 16-70 and described in Table 16-61.

Return to the Summary Table.

ADC PPB4 Trip Low/Trigger Time Stamp Register

Figure 16-70. ADCPPB4TRIPLO Register

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

Table 16-108. ADCPPB4TRIPLO Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15-0	LIMITLO	R/W	0h	ADC Post Processing Block 4 Trip High Low Limit. This value sets the digital comparator trip low limit if ADCPPB4TRIPLO.LIMITLO2EN = 1. When comparing to an ADCPPBxRESULT register, the upper bits will be ignored: - TRIPLO[15:13] will be ignored in 12 bit mode Reset type: SYSRSn

48 ADCINTCYCLE Register (Offset = 180h) [Reset = 00000000h]

ADCINTCYCLE is shown in Figure 16-71 and described in Table 16-62.

Return to the Summary Table.

ADC Early Interrupt Generation Cycle

Figure 16-71. ADCINTCYCLE Register

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

Table 16-110. ADCINTCYCLE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-6	RESERVED	R	0h	Reserved
5-0	DELAY	R/W	0h	ADC Early Interrupt Generation Cycle Delay: Defines the delay from the fall edge of ADCSOC in terms of system clock cycles, for the interrupt to be generated. Reset type: SYSRSn

49 ADCREV2 Register (Offset = 19Ch) [Reset = 00000006h]

ADCREV2 is shown in Figure 16-72 and described in Table 16-63.

Return to the Summary Table.

ADC Wrapper Revision Register

Figure 16-72. ADCREV2 Register

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

Table 16-112. ADCREV2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	WRAPPERREV	R	0h	ADC Revision. To allow documentation of differences between revisions. First version is labeled as 00h. Reset type: SYSRSn
7-0	WRAPPERTYPE	R	6h	ADC Wrapper Type. Always set to 6 for this ADC. Reset type: SYSRSn

50 ADCPPB1LIMIT Register (Offset = 200h) [Reset = 00000000h]

ADCPPB1LIMIT is shown in Figure 16-73 and described in Table 16-64.

Return to the Summary Table.

ADC PPB1Conversion Count Limit Register

Figure 16-73. ADCPPB1LIMIT Register

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

Table 16-114. ADCPPB1LIMIT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R	0h	Reserved
1-0	LIMIT	R/W	0h	Post Processing Block 1 Oversampling Limit. Defines the number of conversions to accumulate before PSUM is automatically loaded into SUM. 0 = No - accumulation 1 = 2 conversions are accumulated 2 = 4 conversions are accumulated 3 = 8 conversions are accumulated Reset type: SYSRSn

ADVANCE INFORMATION

51 ADCPPB1PCOUNT Register (Offset = 204h) [Reset = 00000000h]

ADCPPB1PCOUNT is shown in Figure 16-74 and described in Table 16-65.

Return to the Summary Table.

ADC PPB1 Partial Conversion Count Register

Figure 16-74. ADCPPB1PCOUNT Register

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

Table 16-116. ADCPPB1PCOUNT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R	0h	Reserved
2-0	PCOUNT	R	0h	Post Processing Block 1 Oversampling Partial Count. Each time a new result propagates through the PPB signal chain and accumulates into ADCPPB1PSUM this register is incremented by 1. This register is reset when either a count-match event occurs (PCOUNT = LIMIT) or PPB1 receives a sync. event. This result is available 1 MCLK cycle after the associated ADCPPB1RESULT is available. This will be 2 MCLK cycles after the associated ADCRESULT is available, unless multiple PPBs point to the same SOC (refer to the ADCPPB1RESULT timing information). Reset type: SYSRSn

52 ADCPPB1CONFIG2 Register (Offset = 208h) [Reset = 00000000h]

ADCPPB1CONFIG2 is shown in Figure 16-75 and described in Table 16-66.

Return to the Summary Table.

ADC PPB1 Sum Shift Register

Figure 16-75. ADCPPB1CONFIG2 Register

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

Table 16-118. ADCPPB1CONFIG2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15	COMPSEL	R/W	0h	Post Processing Block 1 Compare Source Select. This field determines whether ADCPPB1RESULT or ADCPPB1SUM is used for the threshold compare. 0 = ADCPPB1RESULT is used for compare logic 1 = ADCPPB1SUM is used for compare logic Reset type: SYSRSn
14-13	RESERVED	R	0h	Reserved
12	RESERVED	R/W	0h	Reserved
11	SWSYNC	R-0/W1S	0h	PPB 1 software force sync. On a sync. event, all partial registers transfer to the final registers and are then reset. Note: In the case where the software force occurs at the same time that a new sample is added to the PSUM and the PSUM is being used for a high or low limit compare, then the comparison will not occur. Reset type: SYSRSn
10-9	RESERVED	R	0h	Reserved
8-4	SYNCINSEL	R/W	0h	PPB 1 sync. input select. On a sync. event, all partial registers transfer to the final registers and are then reset. Refer to SOC spec for details Reset type: SYSRSn
3	RESERVED	R	0h	Reserved

Table 16-118. ADCPPB1CONFIG2 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
2-0	SHIFT	R/W	0h	Post Processing Block 1 right shift. Defines the number of bits to right shift PSUM before loading into the final SUM. 0 : no right shift 1 : SUM = PSUM >> 1 2 : SUM = PSUM >> 2 ... 7 : SUM = PSUM >> 7 Reset type: SYSRSn

53 ADCPPB1PSUM Register (Offset = 20Ch) [Reset = 00000000h]

ADCPPB1PSUM is shown in Figure 16-76 and described in Table 16-67.

Return to the Summary Table.

ADC PPB1 Partial Sum Register

Figure 16-76. ADCPPB1PSUM Register

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

Table 16-120. ADCPPB1PSUM Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	SIGN	R	0h	Sign Extended Bits. These bits reflect the same value as bit 15. Reset type: SYSRSn
15-0	PSUM	R	0h	Post Processing Block 1 Oversampling Partial Sum. Each time a new result propagates through the PPB signal chain and latches into ADCPPB1RESULT the result is subsequently accumulated into this register. This register is reset when either a count-match event occurs (PCOUNT = LIMIT) or PPB1 receives a sync. event. This result is available 1 MCLK cycle after the associated ADCPPB1RESULT is available. This will be 2 MCLK cycles after the associated ADCRESULT is available, unless multiple PPBs point to the same SOC. In the case of multiple PPBs associated with the same SOC, the lowest numbered PPB's result will be availble 2 MCLK cycle after the associated ADCRESULT and subsequent results (in order from lowest numbered PPB to highest) will each become available every 2-3 MCLK cycles (refer to the TRM for detailed timing information). Reset type: SYSRSn

54 ADCPPB2LIMIT Register (Offset = 240h) [Reset = 00000000h]

ADCPPB2LIMIT is shown in Figure 16-77 and described in Table 16-68.

Return to the Summary Table.

ADC PPB2Conversion Count Limit Register

Figure 16-77. ADCPPB2LIMIT Register

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

Table 16-122. ADCPPB2LIMIT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R	0h	Reserved
1-0	LIMIT	R/W	0h	Post Processing Block 2 Oversampling Limit. Defines the number of conversions to accumulate before PSUM is automatically loaded into SUM. 0 = No - accumulation 1 = 2 conversions are accumulated 2 = 4 conversions are accumulated 3 = 8 conversions are accumulated Reset type: SYSRSn

ADVANCE INFORMATION

55 ADCPPB2PCOUNT Register (Offset = 244h) [Reset = 00000000h]

ADCPPB2PCOUNT is shown in Figure 16-78 and described in Table 16-69.

Return to the Summary Table.

ADC PPB2 Partial Conversion Count Register

Figure 16-78. ADCPPB2PCOUNT Register

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

Table 16-124. ADCPPB2PCOUNT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R	0h	Reserved
2-0	PCOUNT	R	0h	Post Processing Block 2 Oversampling Partial Count. Each time a new result propagates through the PPB signal chain and accumulates into ADCPPB2PSUM this register is incremented by 1. This register is reset when either a count-match event occurs (PCOUNT = LIMIT) or PPB2 receives a sync. event. This result is available 1 MCLK cycle after the associated ADCPPB2RESULT is available. This will be 2 MCLK cycles after the associated ADCRESULT is available, unless multiple PPBs point to the same SOC (refer to the ADCPPB2RESULT timing information). Reset type: SYSRSn

56 ADCPPB2CONFIG2 Register (Offset = 248h) [Reset = 00000000h]

ADCPPB2CONFIG2 is shown in Figure 16-79 and described in Table 16-70.

Return to the Summary Table.

ADC PPB2 Sum Shift Register

Figure 16-79. ADCPPB2CONFIG2 Register

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

Table 16-126. ADCPPB2CONFIG2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15	COMPSEL	R/W	0h	Post Processing Block 2 Compare Source Select. This field determines whether ADCPPB2RESULT or ADCPPB2SUM is used for the threshold compare. 0 = ADCPPB2RESULT is used for compare logic 1 = ADCPPB2SUM is used for compare logic Reset type: SYSRSn
14-13	RESERVED	R	0h	Reserved
12	RESERVED	R/W	0h	Reserved
11	SWSYNC	R-0/W1S	0h	PPB 2 software force sync. On a sync. event, all partial registers transfer to the final registers and are then reset. Note: In the case where the software force occurs at the same time that a new sample is added to the PSUM and the PSUM is being used for a high or low limit compare, then the comparison will not occur. Reset type: SYSRSn
10-9	RESERVED	R	0h	Reserved
8-4	SYNCINSEL	R/W	0h	PPB 2 sync. input select. On a sync. event, all partial registers transfer to the final registers and are then reset. Refer to SOC spec for details Reset type: SYSRSn
3	RESERVED	R	0h	Reserved

ADVANCE INFORMATION

Table 16-126. ADCPPB2CONFIG2 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
2-0	SHIFT	R/W	0h	Post Processing Block 2 right shift. Defines the number of bits to right shift PSUM before loading into the final SUM. 0 : no right shift 1 : SUM = PSUM >> 1 2 : SUM = PSUM >> 2 ... 7 : SUM = PSUM >> 7 Reset type: SYSRSn

57 ADCPPB2PSUM Register (Offset = 24Ch) [Reset = 00000000h]

ADCPPB2PSUM is shown in Figure 16-80 and described in Table 16-71.

Return to the Summary Table.

ADC PPB2 Partial Sum Register

Figure 16-80. ADCPPB2PSUM Register

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

Table 16-128. ADCPPB2PSUM Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	SIGN	R	0h	Sign Extended Bits. These bits reflect the same value as bit 15. Reset type: SYSRSn
15-0	PSUM	R	0h	Post Processing Block 2 Oversampling Partial Sum. Each time a new result propagates through the PPB signal chain and latches into ADCPPB2RESULT the result is subsequently accumulated into this register. This register is reset when either a count-match event occurs (PCOUNT = LIMIT) or PPB2 receives a sync. event. This result is available 1 MCLK cycle after the associated ADCPPB2RESULT is available. This will be 2 MCLK cycles after the associated ADCRESULT is available, unless multiple PPBs point to the same SOC. In the case of multiple PPBs associated with the same SOC, the lowest numbered PPB's result will be availble 2 MCLK cycle after the associated ADCRESULT and subsequent results (in order from lowest numbered PPB to highest) will each become available every 2-3 MCLK cycles (refer to the TRM for detailed timing information). Reset type: SYSRSn

58 ADCPPB3LIMIT Register (Offset = 280h) [Reset = 00000000h]

ADCPPB3LIMIT is shown in Figure 16-81 and described in Table 16-72.

Return to the Summary Table.

ADC PPB3Conversion Count Limit Register

Figure 16-81. ADCPPB3LIMIT Register

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

Table 16-130. ADCPPB3LIMIT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R	0h	Reserved
1-0	LIMIT	R/W	0h	Post Processing Block 3 Oversampling Limit. Defines the number of conversions to accumulate before PSUM is automatically loaded into SUM. 0 = No - accumulation 1 = 2 conversions are accumulated 2 = 4 conversions are accumulated 3 = 8 conversions are accumulated Reset type: SYSRSn

ADVANCE INFORMATION

59 ADCPPB3PCOUNT Register (Offset = 284h) [Reset = 00000000h]

ADCPPB3PCOUNT is shown in Figure 16-82 and described in Table 16-73.

Return to the Summary Table.

ADC PPB3 Partial Conversion Count Register

Figure 16-82. ADCPPB3PCOUNT Register

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

Table 16-132. ADCPPB3PCOUNT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R	0h	Reserved
2-0	PCOUNT	R	0h	Post Processing Block 3 Oversampling Partial Count. Each time a new result propagates through the PPB signal chain and accumulates into ADCPPB3PSUM this register is incremented by 1. This register is reset when either a count-match event occurs (PCOUNT = LIMIT) or PPB3 receives a sync. event. This result is available 1 MCLK cycle after the associated ADCPPB3RESULT is available. This will be 2 MCLK cycles after the associated ADCRESULT is available, unless multiple PPBs point to the same SOC (refer to the ADCPPB3RESULT timing information). Reset type: SYSRSn

60 ADCPPB3CONFIG2 Register (Offset = 288h) [Reset = 00000000h]

ADCPPB3CONFIG2 is shown in Figure 16-83 and described in Table 16-74.

Return to the Summary Table.

ADC PPB3 Sum Shift Register

Figure 16-83. ADCPPB3CONFIG2 Register

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

Table 16-134. ADCPPB3CONFIG2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15	COMPSEL	R/W	0h	Post Processing Block 3 Compare Source Select. This field determines whether ADCPPB3RESULT or ADCPPB3SUM is used for the threshold compare. 0 = ADCPPB3RESULT is used for compare logic 1 = ADCPPB3SUM is used for compare logic Reset type: SYSRSn
14-13	RESERVED	R	0h	Reserved
12	RESERVED	R/W	0h	Reserved
11	SWSYNC	R-0/W1S	0h	PPB 3 software force sync. On a sync. event, all partial registers transfer to the final registers and are then reset. Note: In the case where the software force occurs at the same time that a new sample is added to the PSUM and the PSUM is being used for a high or low limit compare, then the comparison will not occur. Reset type: SYSRSn
10-9	RESERVED	R	0h	Reserved
8-4	SYNCINSEL	R/W	0h	PPB 3 sync. input select. On a sync. event, all partial registers transfer to the final registers and are then reset. Refer to SOC spec for details Reset type: SYSRSn
3	RESERVED	R	0h	Reserved

Table 16-134. ADCPPB3CONFIG2 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
2-0	SHIFT	R/W	0h	Post Processing Block 3 right shift. Defines the number of bits to right shift PSUM before loading into the final SUM. 0 : no right shift 1 : SUM = PSUM >> 1 2 : SUM = PSUM >> 2 ... 7 : SUM = PSUM >> 7 Reset type: SYSRSn

61 ADCPPB3PSUM Register (Offset = 28Ch) [Reset = 00000000h]

ADCPPB3PSUM is shown in Figure 16-84 and described in Table 16-75.

Return to the Summary Table.

ADC PPB3 Partial Sum Register

Figure 16-84. ADCPPB3PSUM Register

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

Table 16-136. ADCPPB3PSUM Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	SIGN	R	0h	Sign Extended Bits. These bits reflect the same value as bit 15. Reset type: SYSRSn
15-0	PSUM	R	0h	Post Processing Block 3 Oversampling Partial Sum. Each time a new result propagates through the PPB signal chain and latches into ADCPPB3RESULT the result is subsequently accumulated into this register. This register is reset when either a count-match event occurs (PCOUNT = LIMIT) or PPB3 receives a sync. event. This result is available 1 MCLK cycle after the associated ADCPPB3RESULT is available. This will be 2 MCLK cycles after the associated ADCRESULT is available, unless multiple PPBs point to the same SOC. In the case of multiple PPBs associated with the same SOC, the lowest numbered PPB's result will be availble 2 MCLK cycle after the associated ADCRESULT and subsequent results (in order from lowest numbered PPB to highest) will each become available every 2-3 MCLK cycles (refer to the TRM for detailed timing information). Reset type: SYSRSn

62 ADCPPB4LIMIT Register (Offset = 2C0h) [Reset = 00000000h]

ADCPPB4LIMIT is shown in Figure 16-85 and described in Table 16-76.

Return to the Summary Table.

ADC PPB4Conversion Count Limit Register

Figure 16-85. ADCPPB4LIMIT Register

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

Table 16-138. ADCPPB4LIMIT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R	0h	Reserved
1-0	LIMIT	R/W	0h	Post Processing Block 4 Oversampling Limit. Defines the number of conversions to accumulate before PSUM is automatically loaded into SUM. 0 = No - accumulation 1 = 2 conversions are accumulated 2 = 4 conversions are accumulated 3 = 8 conversions are accumulated Reset type: SYSRSn

ADVANCE INFORMATION

63 ADCPPB4PCOUNT Register (Offset = 2C4h) [Reset = 00000000h]

ADCPPB4PCOUNT is shown in Figure 16-86 and described in Table 16-77.

Return to the Summary Table.

ADC PPB4 Partial Conversion Count Register

Figure 16-86. ADCPPB4PCOUNT Register

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

Table 16-140. ADCPPB4PCOUNT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R	0h	Reserved
2-0	PCOUNT	R	0h	Post Processing Block 4 Oversampling Partial Count. Each time a new result propagates through the PPB signal chain and accumulates into ADCPPB4PSUM this register is incremented by 1. This register is reset when either a count-match event occurs (PCOUNT = LIMIT) or PPB4 receives a sync. event. This result is available 1 MCLK cycle after the associated ADCPPB4RESULT is available. This will be 2 MCLK cycles after the associated ADCRESULT is available, unless multiple PPBs point to the same SOC (refer to the ADCPPB4RESULT timing information). Reset type: SYSRSn

64 ADCPPB4CONFIG2 Register (Offset = 2C8h) [Reset = 00000000h]

ADCPPB4CONFIG2 is shown in Figure 16-87 and described in Table 16-78.

Return to the Summary Table.

ADC PPB4 Sum Shift Register

Figure 16-87. ADCPPB4CONFIG2 Register

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

Table 16-142. ADCPPB4CONFIG2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved
15	COMPSEL	R/W	0h	Post Processing Block 4 Compare Source Select. This field determines whether ADCPPB4RESULT or ADCPPB4SUM is used for the threshold compare. 0 = ADCPPB4RESULT is used for compare logic 1 = ADCPPB4SUM is used for compare logic Reset type: SYSRSn
14-13	RESERVED	R	0h	Reserved
12	RESERVED	R/W	0h	Reserved
11	SWSYNC	R-0/W1S	0h	PPB 4 software force sync. On a sync. event, all partial registers transfer to the final registers and are then reset. Note: In the case where the software force occurs at the same time that a new sample is added to the PSUM and the PSUM is being used for a high or low limit compare, then the comparison will not occur. Reset type: SYSRSn
10-9	RESERVED	R	0h	Reserved
8-4	SYNCINSEL	R/W	0h	PPB 4 sync. input select. On a sync. event, all partial registers transfer to the final registers and are then reset. Refer to SOC spec for details Reset type: SYSRSn
3	RESERVED	R	0h	Reserved

Table 16-142. ADCPPB4CONFIG2 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
2-0	SHIFT	R/W	0h	Post Processing Block 4 right shift. Defines the number of bits to right shift PSUM before loading into the final SUM. 0 : no right shift 1 : SUM = PSUM >> 1 2 : SUM = PSUM >> 2 ... 7 : SUM = PSUM >> 7 Reset type: SYSRSn

65 ADCPPB4PSUM Register (Offset = 2CCh) [Reset = 00000000h]

ADCPPB4PSUM is shown in Figure 16-88 and described in Table 16-79.

Return to the Summary Table.

ADC PPB4 Partial Sum Register

Figure 16-88. ADCPPB4PSUM Register

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

Table 16-144. ADCPPB4PSUM Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	SIGN	R	0h	Sign Extended Bits. These bits reflect the same value as bit 15. Reset type: SYSRSn
15-0	PSUM	R	0h	Post Processing Block 4 Oversampling Partial Sum. Each time a new result propagates through the PPB signal chain and latches into ADCPPB4RESULT the result is subsequently accumulated into this register. This register is reset when either a count-match event occurs (PCOUNT = LIMIT) or PPB4 receives a sync. event. This result is available 1 MCLK cycle after the associated ADCPPB4RESULT is available. This will be 2 MCLK cycles after the associated ADCRESULT is available, unless multiple PPBs point to the same SOC. In the case of multiple PPBs associated with the same SOC, the lowest numbered PPB's result will be availble 2 MCLK cycle after the associated ADCRESULT and subsequent results (in order from lowest numbered PPB to highest) will each become available every 2-3 MCLK cycles (refer to the TRM for detailed timing information). Reset type: SYSRSn

66 ADCSEQCTL Register (Offset = 320h) [Reset = 00000000h]

ADCSEQCTL is shown in Figure 16-89 and described in Table 16-80.

Return to the Summary Table.

ADC Sequencer common control Register

Figure 16-89. ADCSEQCTL Register

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

Table 16-146. ADCSEQCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	RESERVED	R	0h	Reserved
17-16	SEQPREEMPT	R/W	0h	SoC Sequence PREEMPT 0X: Pre-empt disabled 10: Pre-empt enabled and will not restart aborted Sequence 11: Pre-empt enabled and will restart aborted Sequence Reset type: SYSRSn
15-4	RESERVED	R	0h	Reserved
3-0	SEQEND	R/W	0h	END SOC of last enabled sequence Reset type: SYSRSn

67 ADCSEQ1CONFIG_AM13E2X Register (Offset = 324h) [Reset = 00000200h]

ADCSEQ1CONFIG_AM13E2X is shown in Figure 16-90 and described in Table 16-81.

Return to the Summary Table.

ADC Sequencer 1 Config register

Figure 16-90. ADCSEQ1CONFIG_AM13E2X Register

31	30	29	28	27	26	25	24
SEQENABLE	SEQSWFRC	RESERVED					TRIGSEL
R/W-0h	R-0/W1S-0h	R-0h					R/W-0h
23	22	21	20	19	18	17	16
TRIGSEL				RESERVED	SEQSTART		
R/W-0h				R-0h	R/W-0h		
15	14	13	12	11	10	9	8
SEQSTART	RESERVED				SAMPCAPRESETSEL	SAMPCAPRESETDISABLE	RESERVED
R/W-0h	R-0h				R/W-0h	R/W-1h	R-0h
7	6	5	4	3	2	1	0
ACQPS							
R/W-0h							

Table 16-148. ADCSEQ1CONFIG_AM13E2X Register Field Descriptions

Bit	Field	Type	Reset	Description
31	SEQENABLE	R/W	0h	SEQ1Enable Indicates whether the Sequence1 is enabled or not Reset type: SYSRSn
30	SEQSWFRC	R-0/W1S	0h	Write 1 to force a trigger to Sequencer 1 regardless of the value of Hardware TRIGGER. Always reads 0. Reset type: SYSRSn
29-25	RESERVED	R	0h	Reserved
24-20	TRIGSEL	R/W	0h	SOC1 Trigger Source Select. Along with the SOC1 field in the ADCINTSOCSEL1 register, this bit field configures which trigger will set the SOC1 flag in the ADCSOCFLG1 register to initiate a conversion to start once priority is given to it. Note: SOCFRC1 register can always be used to software trigger SOCs in addition to any hardware trigger configuration. 0h ADCTRIG0 - Software only 1h - 31h Hardware triggers Reset type: SYSRSn
19	RESERVED	R	0h	Reserved
18-15	SEQSTART	R/W	0h	Beginning SOC of SEQ1 Reset type: SYSRSn
14-11	RESERVED	R	0h	Reserved
10	SAMPCAPRESETSEL	R/W	0h	SOC1 Sample Cap Reset Select : Resets sample cap after conversion to either vrefhi/2 or vreflo 0 - The sample cap is reset to Vreflo after each conversion 1 - The sample cap is reset to Vrefhi/2 after each conversion Reset type: SYSRSn

ADVANCE INFORMATION

Table 16-148. ADCSEQ1CONFIG_AM13E2X Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
9	SAMPCAPRESETDISABLE	R/W	1h	SOC1 Sample Cap Reset enable : Resets sample cap after conversion. 0 - The sample cap is reset after each conversion 1 - The sample cap is not reset after each conversion Reset type: SYSRSn
8	RESERVED	R	0h	Reserved
7-0	ACQPS	R/W	0h	SOC1 Acquisition Prescale. Controls the sample and hold window for each SOC in this Seqeunce. The configured acquisition time must be at least as long as one ADCCLK cycle for correct ADC operation. The device datasheet will also specify a minimum sample and hold window duration. S+H window values is defined based on a combonation of ACQPS[7:6] and ACQPS[5:0] values If ACQPS[7:6] value is "00" S+H window = ACQPS[5:0] + 1 MCLK cycles "01" S+H window = 64 + ((ACQPS[5:0] +1) * 2) MCLK cycles "10" S+H window = 192 + ((ACQPS[5:0] +1) * 4) MCLK cycles "11" S+H window = 448 + ((ACQPS[5:0] +1) * 16) MCLK cycles Reset type: SYSRSn

68 ADCSEQ2CONFIG_AM13E2X Register (Offset = 328h) [Reset = 00000200h]

ADCSEQ2CONFIG_AM13E2X is shown in Figure 16-91 and described in Table 16-82.

Return to the Summary Table.

ADC Sequencer 2 Config register

Figure 16-91. ADCSEQ2CONFIG_AM13E2X Register

31	30	29	28	27	26	25	24
SEQENABLE	SEQSWFRC	RESERVED					TRIGSEL
R/W-0h	R-0/W1S-0h	R-0h					R/W-0h
23	22	21	20	19	18	17	16
TRIGSEL				RESERVED	SEQSTART		
R/W-0h				R-0h	R/W-0h		
15	14	13	12	11	10	9	8
SEQSTART	RESERVED				SAMPCAPRESETSEL	SAMPCAPRESETDISABLE	RESERVED
R/W-0h	R-0h				R/W-0h	R/W-1h	R-0h
7	6	5	4	3	2	1	0
ACQPS							
R/W-0h							

Table 16-150. ADCSEQ2CONFIG_AM13E2X Register Field Descriptions

Bit	Field	Type	Reset	Description
31	SEQENABLE	R/W	0h	SEQ2Enable Indicates whether the Sequence2 is enabled or not Reset type: SYSRSn
30	SEQSWFRC	R-0/W1S	0h	Write 1 to force a trigger to Sequencer 2 regardless of the value of Hardware TRIGGER. Always reads 0. Reset type: SYSRSn
29-25	RESERVED	R	0h	Reserved
24-20	TRIGSEL	R/W	0h	SOC2 Trigger Source Select. Along with the SOC2 field in the ADCINTSOCSEL1 register, this bit field configures which trigger will set the SOC2 flag in the ADCSOCFLG1 register to initiate a conversion to start once priority is given to it. Note: SOCFRC1 register can always be used to software trigger SOCs in addition to any hardware trigger configuration. 0h ADCTRIG0 - Software only 1h - 31h Hardware triggers Reset type: SYSRSn
19	RESERVED	R	0h	Reserved
18-15	SEQSTART	R/W	0h	Beginning SOC of SEQ2 Reset type: SYSRSn
14-11	RESERVED	R	0h	Reserved
10	SAMPCAPRESETSEL	R/W	0h	SOC2 Sample Cap Reset Select : Resets sample cap after conversion to either vrefhi/2 or vreflo 0 - The sample cap is reset to Vreflo after each conversion 1 - The sample cap is reset to Vrefhi/2 after each conversion Reset type: SYSRSn

ADVANCE INFORMATION

Table 16-150. ADCSEQ2CONFIG_AM13E2X Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
9	SAMPCAPRESETDISABLE	R/W	1h	SOC2 Sample Cap Reset enable : Resets sample cap after conversion. 0 - The sample cap is reset after each conversion 1 - The sample cap is not reset after each conversion Reset type: SYSRSn
8	RESERVED	R	0h	Reserved
7-0	ACQPS	R/W	0h	SOC2 Acquisition Prescale. Controls the sample and hold window for each SOC in this Seqeunce. The configured acquisition time must be at least as long as one ADCCLK cycle for correct ADC operation. The device datasheet will also specify a minimum sample and hold window duration. S+H window values is defined based on a combonation of ACQPS[7:6] and ACQPS[5:0] values If ACQPS[7:6] value is "00" S+H window = ACQPS[5:0] + 1 MCLK cycles "01" S+H window = 64 + ((ACQPS[5:0] +1) * 2) MCLK cycles "10" S+H window = 192 + ((ACQPS[5:0] +1) * 4) MCLK cycles "11" S+H window = 448 + ((ACQPS[5:0] +1) * 16) MCLK cycles Reset type: SYSRSn

69 ADCSEQ3CONFIG_AM13E2X Register (Offset = 32Ch) [Reset = 00000200h]

ADCSEQ3CONFIG_AM13E2X is shown in Figure 16-92 and described in Table 16-83.

Return to the Summary Table.

ADC Sequencer 3 Config register

Figure 16-92. ADCSEQ3CONFIG_AM13E2X Register

31	30	29	28	27	26	25	24
SEQENABLE	SEQSWFRC	RESERVED					TRIGSEL
R/W-0h	R-0/W1S-0h	R-0h					R/W-0h
23	22	21	20	19	18	17	16
TRIGSEL				RESERVED	SEQSTART		
R/W-0h				R-0h	R/W-0h		
15	14	13	12	11	10	9	8
SEQSTART	RESERVED				SAMPCAPRESETSEL	SAMPCAPRESETDISABLE	RESERVED
R/W-0h	R-0h				R/W-0h	R/W-1h	R-0h
7	6	5	4	3	2	1	0
ACQPS							
R/W-0h							

Table 16-152. ADCSEQ3CONFIG_AM13E2X Register Field Descriptions

Bit	Field	Type	Reset	Description
31	SEQENABLE	R/W	0h	SEQ3Enable Indicates whether the Sequence3 is enabled or not Reset type: SYSRSn
30	SEQSWFRC	R-0/W1S	0h	Write 1 to force a trigger to Sequencer 3 regardless of the value of Hardware TRIGGER. Always reads 0. Reset type: SYSRSn
29-25	RESERVED	R	0h	Reserved
24-20	TRIGSEL	R/W	0h	SOC3 Trigger Source Select. Along with the SOC3 field in the ADCINTSOCSEL1 register, this bit field configures which trigger will set the SOC3 flag in the ADCSOCFLG1 register to initiate a conversion to start once priority is given to it. Note: SOCFRC1 register can always be used to software trigger SOCs in addition to any hardware trigger configuration. 0h ADCTRIG0 - Software only 1h - 31h Hardware triggers Reset type: SYSRSn
19	RESERVED	R	0h	Reserved
18-15	SEQSTART	R/W	0h	Beginning SOC of SEQ3 Reset type: SYSRSn
14-11	RESERVED	R	0h	Reserved
10	SAMPCAPRESETSEL	R/W	0h	SOC3 Sample Cap Reset Select : Resets sample cap after conversion to either vrefhi/2 or vreflo 0 - The sample cap is reset to Vreflo after each conversion 1 - The sample cap is reset to Vrefhi/2 after each conversion Reset type: SYSRSn

ADVANCE INFORMATION

Table 16-152. ADCSEQ3CONFIG_AM13E2X Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
9	SAMPCAPRESETDISABLE	R/W	1h	SOC3 Sample Cap Reset enable : Resets sample cap after conversion. 0 - The sample cap is reset after each conversion 1 - The sample cap is not reset after each conversion Reset type: SYSRSn
8	RESERVED	R	0h	Reserved
7-0	ACQPS	R/W	0h	SOC3 Acquisition Prescale. Controls the sample and hold window for each SOC in this Seqeunce. The configured acquisition time must be at least as long as one ADCCLK cycle for correct ADC operation. The device datasheet will also specify a minimum sample and hold window duration. S+H window values is defined based on a combonation of ACQPS[7:6] and ACQPS[5:0] values If ACQPS[7:6] value is "00" S+H window = ACQPS[5:0] + 1 MCLK cycles "01" S+H window = 64 + ((ACQPS[5:0] +1) * 2) MCLK cycles "10" S+H window = 192 + ((ACQPS[5:0] +1) * 4) MCLK cycles "11" S+H window = 448 + ((ACQPS[5:0] +1) * 16) MCLK cycles Reset type: SYSRSn

70 ADCSEQ4CONFIG_AM13E2X Register (Offset = 330h) [Reset = 00000200h]

ADCSEQ4CONFIG_AM13E2X is shown in Figure 16-93 and described in Table 16-84.

Return to the Summary Table.

ADC Sequencer 4 Config register

Figure 16-93. ADCSEQ4CONFIG_AM13E2X Register

31	30	29	28	27	26	25	24
SEQENABLE	SEQSWFRC	RESERVED					TRIGSEL
R/W-0h	R-0/W1S-0h	R-0h					R/W-0h
23	22	21	20	19	18	17	16
TRIGSEL				RESERVED	SEQSTART		
R/W-0h				R-0h	R/W-0h		
15	14	13	12	11	10	9	8
SEQSTART	RESERVED				SAMPCAPRESETSEL	SAMPCAPRESETDISABLE	RESERVED
R/W-0h	R-0h				R/W-0h	R/W-1h	R-0h
7	6	5	4	3	2	1	0
ACQPS							
R/W-0h							

Table 16-154. ADCSEQ4CONFIG_AM13E2X Register Field Descriptions

Bit	Field	Type	Reset	Description
31	SEQENABLE	R/W	0h	SEQ4Enable Indicates whether the Sequence4 is enabled or not Reset type: SYSRSn
30	SEQSWFRC	R-0/W1S	0h	Write 1 to force a trigger to Sequencer 4 regardless of the value of Hardware TRIGGER. Always reads 0. Reset type: SYSRSn
29-25	RESERVED	R	0h	Reserved
24-20	TRIGSEL	R/W	0h	SOC4 Trigger Source Select. Along with the SOC4 field in the ADCINTSOCSEL1 register, this bit field configures which trigger will set the SOC4 flag in the ADCSOCFLG1 register to initiate a conversion to start once priority is given to it. Note: SOCFRC1 register can always be used to software trigger SOCs in addition to any hardware trigger configuration. 0h ADCTRIG0 - Software only 1h - 31h Hardware triggers Reset type: SYSRSn
19	RESERVED	R	0h	Reserved
18-15	SEQSTART	R/W	0h	Beginning SOC of SEQ4 Reset type: SYSRSn
14-11	RESERVED	R	0h	Reserved
10	SAMPCAPRESETSEL	R/W	0h	SOC4 Sample Cap Reset Select : Resets sample cap after conversion to either vrefhi/2 or vreflo 0 - The sample cap is reset to Vreflo after each conversion 1 - The sample cap is reset to Vrefhi/2 after each conversion Reset type: SYSRSn

ADVANCE INFORMATION

Table 16-154. ADCSEQ4CONFIG_AM13E2X Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
9	SAMPCAPRESETDISABLE	R/W	1h	SOC4 Sample Cap Reset enable : Resets sample cap after conversion. 0 - The sample cap is reset after each conversion 1 - The sample cap is not reset after each conversion Reset type: SYSRSn
8	RESERVED	R	0h	Reserved
7-0	ACQPS	R/W	0h	SOC4 Acquisition Prescale. Controls the sample and hold window for each SOC in this Seqeunce. The configured acquisition time must be at least as long as one ADCCLK cycle for correct ADC operation. The device datasheet will also specify a minimum sample and hold window duration. S+H window values is defined based on a combonation of ACQPS[7:6] and ACQPS[5:0] values If ACQPS[7:6] value is "00" S+H window = ACQPS[5:0] + 1 MCLK cycles "01" S+H window = 64 + ((ACQPS[5:0] +1) * 2) MCLK cycles "10" S+H window = 192 + ((ACQPS[5:0] +1) * 4) MCLK cycles "11" S+H window = 448 + ((ACQPS[5:0] +1) * 16) MCLK cycles Reset type: SYSRSn

71 PWREN Register (Offset = 800h) [Reset = 00000000h]

PWREN is shown in Figure 16-94 and described in Table 16-85.

Return to the Summary Table.

Register to control the power state

Figure 16-94. PWREN Register

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

Table 16-156. PWREN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	KEY	W	0h	KEY to allow Power State Change 26h = KEY to allow write access to this register
23-1	RESERVED	R/W	0h	
0	ENABLE	R/WK	0h	Enable the power [EXT_GPRCM.GPRCM.PWREN.KEY] must be set to 26h to write to this bit. 0h = Disable Power 1h = Enable Power

ADVANCE INFORMATION

72 RSTCTL Register (Offset = 804h) [Reset = 00000000h]

RSTCTL is shown in Figure 16-95 and described in Table 16-86.

Return to the Summary Table.

Register to control reset assertion and de-assertion

Figure 16-95. RSTCTL Register

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

Table 16-158. RSTCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	KEY	W	0h	Unlock key B1h = KEY to allow write access to this register
23-2	RESERVED	W	0h	
1	RESETSTKYCLR	WK	0h	Clear the RESETSTKY bit in the STAT register [EXT_GPRCM.GPRCM.RSTCTL.KEY] must be set to B1h to write to this bit. 0h = Writing 0 has no effect 1h = Clear reset sticky bit
0	RESETASSERT	WK	0h	Assert reset to the peripheral [EXT_GPRCM.GPRCM.RSTCTL.KEY] must be set to B1h to write to this bit. 0h = Writing 0 has no effect 1h = Assert reset

ADVANCE INFORMATION

73 STAT Register (Offset = 814h) [Reset = 00000000h]

STAT is shown in Figure 16-96 and described in Table 16-87.

Return to the Summary Table.

peripheral enable and reset status

Figure 16-96. STAT Register

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

Table 16-160. STAT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-17	RESERVED	R	0h	
16	RESETSTKYCLR	R	0h	This bit indicates, if the peripheral was reset, since this bit was cleared by RESETSTKYCLR in the RSTCTL register 0h = The peripheral has not been reset since this bit was last cleared by RESETSTKYCLR in the RSTCTL register 1h = The peripheral was reset since the last bit clear
15-0	RESERVED	R	0h	

16.14.3 ADC_LITE_RESULT_REGS Registers

Table 16-88 lists the memory-mapped registers for the ADC_LITE_RESULT_REGS registers. All register offset addresses not listed in Table 16-88 should be considered as reserved locations and the register contents should not be modified.

Table 16-161. ADC_LITE_RESULT_REGS Registers

Offset	Acronym	Register Name	Write Protection	Section
0h	ADCRESULT0	ADC Result 0 Register		Go
2h	ADCRESULT1	ADC Result 1 Register		Go
4h	ADCRESULT2	ADC Result 2 Register		Go
6h	ADCRESULT3	ADC Result 3 Register		Go
8h	ADCRESULT4	ADC Result 4 Register		Go
Ah	ADCRESULT5	ADC Result 5 Register		Go
Ch	ADCRESULT6	ADC Result 6 Register		Go
Eh	ADCRESULT7	ADC Result 7 Register		Go
10h	ADCRESULT8	ADC Result 8 Register		Go
12h	ADCRESULT9	ADC Result 9 Register		Go
14h	ADCRESULT10	ADC Result 10 Register		Go
16h	ADCRESULT11	ADC Result 11 Register		Go
18h	ADCRESULT12	ADC Result 12 Register		Go
1Ah	ADCRESULT13	ADC Result 13 Register		Go
1Ch	ADCRESULT14	ADC Result 14 Register		Go
1Eh	ADCRESULT15	ADC Result 15 Register		Go
40h	ADCPPB1RESULT	ADC Post Processing Block 1 Result Register		Go
44h	ADCPPB2RESULT	ADC Post Processing Block 2 Result Register		Go
48h	ADCPPB3RESULT	ADC Post Processing Block 3 Result Register		Go
4Ch	ADCPPB4RESULT	ADC Post Processing Block 4 Result Register		Go
50h	ADCPPB1SUM	ADC PPB 1 Final Sum Result Register		Go
54h	ADCPPB1COUNT	ADC PPB1 Final Conversion Count Register		Go
58h	ADCPPB2SUM	ADC PPB 2 Final Sum Result Register		Go
5Ch	ADCPPB2COUNT	ADC PPB2 Final Conversion Count Register		Go
60h	ADCPPB3SUM	ADC PPB 3 Final Sum Result Register		Go
64h	ADCPPB3COUNT	ADC PPB3 Final Conversion Count Register		Go
68h	ADCPPB4SUM	ADC PPB 4 Final Sum Result Register		Go
6Ch	ADCPPB4COUNT	ADC PPB4 Final Conversion Count Register		Go
C0h	ADCSEQ1FIFORESULT	ADC Sequence 1 FIFO Result Register		Go
C4h	ADCSEQ2FIFORESULT	ADC Sequence 2 FIFO Result Register		Go
C8h	ADCSEQ3FIFORESULT	ADC Sequence 3 FIFO Result Register		Go
CCh	ADCSEQ4FIFORESULT	ADC Sequence 4 FIFO Result Register		Go
D0h	ADCSEQ1FIFOSTATUS	ADC Sequence 1 FIFO Status		Go
D4h	ADCSEQ2FIFOSTATUS	ADC Sequence 2 FIFO Status		Go
D8h	ADCSEQ3FIFOSTATUS	ADC Sequence 3 FIFO Status		Go
DCh	ADCSEQ4FIFOSTATUS	ADC Sequence 4 FIFO Status		Go

Complex bit access types are encoded to fit into small table cells. Table 16-89 shows the codes that are used for access types in this section.

Table 16-162. ADC_LITE_RESULT_REGS Access Type Codes

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

1 ADCRESULT0 Register (Offset = 0h) [Reset = 0000h]

ADCRESULT0 is shown in Figure 16-97 and described in Table 16-90.

Return to the Summary Table.

ADC Result 0 Register

Figure 16-97. ADCRESULT0 Register

15	14	13	12	11	10	9	8
RESULT							
R-0h							
7	6	5	4	3	2	1	0
RESULT							
R-0h							

Table 16-164. ADCRESULT0 Register Field Descriptions

Bit	Field	Type	Reset	Description
15-0	RESULT	R	0h	ADC Result 0 12-bit ADC result. After the ADC completes a conversion of SOC0, the digital result is placed in this bit field. If oversampling is enabled, Final SUM (After any Shift if enabled) will be stored in this register after completing the specified number of samples. Reset type: SYSRSn

2 ADCRESULT1 Register (Offset = 2h) [Reset = 0000h]

ADCRESULT1 is shown in Figure 16-98 and described in Table 16-91.

Return to the Summary Table.

ADC Result 1 Register

Figure 16-98. ADCRESULT1 Register

15	14	13	12	11	10	9	8
RESULT							
R-0h							
7	6	5	4	3	2	1	0
RESULT							
R-0h							

Table 16-166. ADCRESULT1 Register Field Descriptions

Bit	Field	Type	Reset	Description
15-0	RESULT	R	0h	ADC Result 1 12-bit ADC result. After the ADC completes a conversion of SOC1, the digital result is placed in this bit field. If oversampling is enabled, Final SUM (After any Shift if enabled) will be stored in this register after completing the specified number of samples. Reset type: SYSRSn

3 ADCRESULT2 Register (Offset = 4h) [Reset = 0000h]

ADCRESULT2 is shown in Figure 16-99 and described in Table 16-92.

Return to the Summary Table.

ADC Result 2 Register

Figure 16-99. ADCRESULT2 Register

15	14	13	12	11	10	9	8
RESULT							
R-0h							
7	6	5	4	3	2	1	0
RESULT							
R-0h							

Table 16-168. ADCRESULT2 Register Field Descriptions

Bit	Field	Type	Reset	Description
15-0	RESULT	R	0h	ADC Result 2 12-bit ADC result. After the ADC completes a conversion of SOC2, the digital result is placed in this bit field. If oversampling is enabled, Final SUM (After any Shift if enabled) will be stored in this register after completing the specified number of samples. Reset type: SYSRSn

4 ADCRESULT3 Register (Offset = 6h) [Reset = 0000h]

ADCRESULT3 is shown in Figure 16-100 and described in Table 16-93.

Return to the Summary Table.

ADC Result 3 Register

Figure 16-100. ADCRESULT3 Register

15	14	13	12	11	10	9	8
RESULT							
R-0h							
7	6	5	4	3	2	1	0
RESULT							
R-0h							

Table 16-170. ADCRESULT3 Register Field Descriptions

Bit	Field	Type	Reset	Description
15-0	RESULT	R	0h	ADC Result 3 12-bit ADC result. After the ADC completes a conversion of SOC3, the digital result is placed in this bit field. If oversampling is enabled, Final SUM (After any Shift if enabled) will be stored in this register after completing the specified number of samples. Reset type: SYSRSn

5 ADCRESULT4 Register (Offset = 8h) [Reset = 0000h]

ADCRESULT4 is shown in Figure 16-101 and described in Table 16-94.

Return to the Summary Table.

ADC Result 4 Register

Figure 16-101. ADCRESULT4 Register

15	14	13	12	11	10	9	8
RESULT							
R-0h							
7	6	5	4	3	2	1	0
RESULT							
R-0h							

Table 16-172. ADCRESULT4 Register Field Descriptions

Bit	Field	Type	Reset	Description
15-0	RESULT	R	0h	ADC Result 4 12-bit ADC result. After the ADC completes a conversion of SOC4, the digital result is placed in this bit field. If oversampling is enabled, Final SUM (After any Shift if enabled) will be stored in this register after completing the specified number of samples. Reset type: SYSRSn

6 ADCRESULT5 Register (Offset = Ah) [Reset = 0000h]

ADCRESULT5 is shown in Figure 16-102 and described in Table 16-95.

Return to the Summary Table.

ADC Result 5 Register

Figure 16-102. ADCRESULT5 Register

15	14	13	12	11	10	9	8
RESULT							
R-0h							
7	6	5	4	3	2	1	0
RESULT							
R-0h							

Table 16-174. ADCRESULT5 Register Field Descriptions

Bit	Field	Type	Reset	Description
15-0	RESULT	R	0h	ADC Result 5 12-bit ADC result. After the ADC completes a conversion of SOC5, the digital result is placed in this bit field. If oversampling is enabled, Final SUM (After any Shift if enabled) will be stored in this register after completing the specified number of samples. Reset type: SYSRSn

7 ADCRESULT6 Register (Offset = Ch) [Reset = 0000h]

ADCRESULT6 is shown in Figure 16-103 and described in Table 16-96.

Return to the Summary Table.

ADC Result 6 Register

Figure 16-103. ADCRESULT6 Register

15	14	13	12	11	10	9	8
RESULT							
R-0h							
7	6	5	4	3	2	1	0
RESULT							
R-0h							

Table 16-176. ADCRESULT6 Register Field Descriptions

Bit	Field	Type	Reset	Description
15-0	RESULT	R	0h	ADC Result 6 12-bit ADC result. After the ADC completes a conversion of SOC6, the digital result is placed in this bit field. If oversampling is enabled, Final SUM (After any Shift if enabled) will be stored in this register after completing the specified number of samples. Reset type: SYSRSn

8 ADCRESULT7 Register (Offset = Eh) [Reset = 0000h]

ADCRESULT7 is shown in Figure 16-104 and described in Table 16-97.

Return to the Summary Table.

ADC Result 7 Register

Figure 16-104. ADCRESULT7 Register

15	14	13	12	11	10	9	8
RESULT							
R-0h							
7	6	5	4	3	2	1	0
RESULT							
R-0h							

Table 16-178. ADCRESULT7 Register Field Descriptions

Bit	Field	Type	Reset	Description
15-0	RESULT	R	0h	ADC Result 7 12-bit ADC result. After the ADC completes a conversion of SOC7, the digital result is placed in this bit field. If oversampling is enabled, Final SUM (After any Shift if enabled) will be stored in this register after completing the specified number of samples. Reset type: SYSRSn

9 ADCRESULT8 Register (Offset = 10h) [Reset = 0000h]

ADCRESULT8 is shown in Figure 16-105 and described in Table 16-98.

Return to the Summary Table.

ADC Result 8 Register

Figure 16-105. ADCRESULT8 Register

15	14	13	12	11	10	9	8
RESULT							
R-0h							
7	6	5	4	3	2	1	0
RESULT							
R-0h							

Table 16-180. ADCRESULT8 Register Field Descriptions

Bit	Field	Type	Reset	Description
15-0	RESULT	R	0h	ADC Result 8 12-bit ADC result. After the ADC completes a conversion of SOC8, the digital result is placed in this bit field. If oversampling is enabled, Final SUM (After any Shift if enabled) will be stored in this register after completing the specified number of samples. Reset type: SYSRSn

ADVANCE INFORMATION

10 ADCRESULT9 Register (Offset = 12h) [Reset = 0000h]

ADCRESULT9 is shown in Figure 16-106 and described in Table 16-99.

Return to the Summary Table.

ADC Result 9 Register

Figure 16-106. ADCRESULT9 Register

15	14	13	12	11	10	9	8
RESULT							
R-0h							
7	6	5	4	3	2	1	0
RESULT							
R-0h							

Table 16-182. ADCRESULT9 Register Field Descriptions

Bit	Field	Type	Reset	Description
15-0	RESULT	R	0h	ADC Result 9 12-bit ADC result. After the ADC completes a conversion of SOC9, the digital result is placed in this bit field. If oversampling is enabled, Final SUM (After any Shift if enabled) will be stored in this register after completing the specified number of samples. Reset type: SYSRSn

11 ADCRESULT10 Register (Offset = 14h) [Reset = 0000h]

ADCRESULT10 is shown in Figure 16-107 and described in Table 16-100.

Return to the Summary Table.

ADC Result 10 Register

Figure 16-107. ADCRESULT10 Register

15	14	13	12	11	10	9	8
RESULT							
R-0h							
7	6	5	4	3	2	1	0
RESULT							
R-0h							

Table 16-184. ADCRESULT10 Register Field Descriptions

Bit	Field	Type	Reset	Description
15-0	RESULT	R	0h	ADC Result 10 12-bit ADC result. After the ADC completes a conversion of SOC10, the digital result is placed in this bit field. If oversampling is enabled, Final SUM (After any Shift if enabled) will be stored in this register after completing the specified number of samples. Reset type: SYSRSn

12 ADCRESULT11 Register (Offset = 16h) [Reset = 0000h]

ADCRESULT11 is shown in Figure 16-108 and described in Table 16-101.

Return to the Summary Table.

ADC Result 11 Register

Figure 16-108. ADCRESULT11 Register

15	14	13	12	11	10	9	8
RESULT							
R-0h							
7	6	5	4	3	2	1	0
RESULT							
R-0h							

Table 16-186. ADCRESULT11 Register Field Descriptions

Bit	Field	Type	Reset	Description
15-0	RESULT	R	0h	ADC Result 11 12-bit ADC result. After the ADC completes a conversion of SOC11, the digital result is placed in this bit field. If oversampling is enabled, Final SUM (After any Shift if enabled) will be stored in this register after completing the specified number of samples. Reset type: SYSRSn

ADVANCE INFORMATION

13 ADCRESULT12 Register (Offset = 18h) [Reset = 0000h]

ADCRESULT12 is shown in Figure 16-109 and described in Table 16-102.

Return to the Summary Table.

ADC Result 12 Register

Figure 16-109. ADCRESULT12 Register

15	14	13	12	11	10	9	8
RESULT							
R-0h							
7	6	5	4	3	2	1	0
RESULT							
R-0h							

Table 16-188. ADCRESULT12 Register Field Descriptions

Bit	Field	Type	Reset	Description
15-0	RESULT	R	0h	ADC Result 12 12-bit ADC result. After the ADC completes a conversion of SOC12, the digital result is placed in this bit field. If oversampling is enabled, Final SUM (After any Shift if enabled) will be stored in this register after completing the specified number of samples. Reset type: SYSRSn

14 ADCRESULT13 Register (Offset = 1Ah) [Reset = 0000h]

ADCRESULT13 is shown in Figure 16-110 and described in Table 16-103.

Return to the Summary Table.

ADC Result 13 Register

Figure 16-110. ADCRESULT13 Register

15	14	13	12	11	10	9	8
RESULT							
R-0h							
7	6	5	4	3	2	1	0
RESULT							
R-0h							

Table 16-190. ADCRESULT13 Register Field Descriptions

Bit	Field	Type	Reset	Description
15-0	RESULT	R	0h	ADC Result 13 12-bit ADC result. After the ADC completes a conversion of SOC13, the digital result is placed in this bit field. If oversampling is enabled, Final SUM (After any Shift if enabled) will be stored in this register after completing the specified number of samples. Reset type: SYSRSn

ADVANCE INFORMATION

15 ADCRESULT14 Register (Offset = 1Ch) [Reset = 0000h]

ADCRESULT14 is shown in Figure 16-111 and described in Table 16-104.

Return to the Summary Table.

ADC Result 14 Register

Figure 16-111. ADCRESULT14 Register

15	14	13	12	11	10	9	8
RESULT							
R-0h							
7	6	5	4	3	2	1	0
RESULT							
R-0h							

Table 16-192. ADCRESULT14 Register Field Descriptions

Bit	Field	Type	Reset	Description
15-0	RESULT	R	0h	ADC Result 14 12-bit ADC result. After the ADC completes a conversion of SOC14, the digital result is placed in this bit field. If oversampling is enabled, Final SUM (After any Shift if enabled) will be stored in this register after completing the specified number of samples. Reset type: SYSRSn

16 ADCRESULT15 Register (Offset = 1Eh) [Reset = 0000h]

ADCRESULT15 is shown in Figure 16-112 and described in Table 16-105.

Return to the Summary Table.

ADC Result 15 Register

Figure 16-112. ADCRESULT15 Register

15	14	13	12	11	10	9	8
RESULT							
R-0h							
7	6	5	4	3	2	1	0
RESULT							
R-0h							

Table 16-194. ADCRESULT15 Register Field Descriptions

Bit	Field	Type	Reset	Description
15-0	RESULT	R	0h	ADC Result 15 12-bit ADC result. After the ADC completes a conversion of SOC15, the digital result is placed in this bit field. If oversampling is enabled, Final SUM (After any Shift if enabled) will be stored in this register after completing the specified number of samples. Reset type: SYSRSn

ADVANCE INFORMATION

17 ADCPPB1RESULT Register (Offset = 40h) [Reset = 00000000h]

ADCPPB1RESULT is shown in Figure 16-113 and described in Table 16-106.

Return to the Summary Table.

ADC Post Processing Block 1 Result Register

Figure 16-113. ADCPPB1RESULT Register

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

Table 16-196. ADCPPB1RESULT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-13	SIGN	R	0h	Sign Extended Bits. These bits reflect the same value as bit 12. NOTE: If the conversion associated with this Post Processing Block is a 12-bit conversion, the SIGN bits extend down to bit 12, and all reflect the same value as bit 12. Reset type: SYSRSn
12-0	PPBRESULT	R	0h	ADC Post Processing Block Result 1 The result of the offset/reference subtraction post conversion processing is stored in this register. This result is available 1 MCLK cycle after the associated ADCRESULT is available, unless multiple PPBs point to the same SOC. In the case of multiple PPBs associated with the same SOC, the lowest numbered PPB's result will be availble 1 MCLK cycle after the associated ADCRESULT and subsequent results (in order from lowest numbered PPB to highest) will each become available every 2-3 MCLK cycles (refer to the TRM for more detailed timing information). If ADCINTFLG is polled to determine when to read the PPBRESULT, it may be necessary to add one or more NOP instructions to ensure that the updated post conversion processing result has posted to the register. NOTE: If the conversion associated with this Post Processing Block is a 12-bit conversion, the PPBRESULT bits are limited to bits 12:0. Reset type: SYSRSn

18 ADCPPB2RESULT Register (Offset = 44h) [Reset = 00000000h]

ADCPPB2RESULT is shown in Figure 16-114 and described in Table 16-107.

Return to the Summary Table.

ADC Post Processing Block 2 Result Register

Figure 16-114. ADCPPB2RESULT Register

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

Table 16-198. ADCPPB2RESULT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-13	SIGN	R	0h	Sign Extended Bits. These bits reflect the same value as bit 12. NOTE: If the conversion associated with this Post Processing Block is a 12-bit conversion, the SIGN bits extend down to bit 12, and all reflect the same value as bit 12. Reset type: SYSRSn
12-0	PPBRESULT	R	0h	ADC Post Processing Block Result 2 The result of the offset/reference subtraction post conversion processing is stored in this register. This result is available 1 MCLK cycle after the associated ADCRESULT is available, unless multiple PPBs point to the same SOC. In the case of multiple PPBs associated with the same SOC, the lowest numbered PPB's result will be availble 1 MCLK cycle after the associated ADCRESULT and subsequent results (in order from lowest numbered PPB to highest) will each become available every 2-3 MCLK cycles (refer to the TRM for more detailed timing information). If ADCINTFLG is polled to determine when to read the PPBRESULT, it may be necessary to add one or more NOP instructions to ensure that the updated post conversion processing result has posted to the register. NOTE: If the conversion associated with this Post Processing Block is a 12-bit conversion, the PPBRESULT bits are limited to bits 12:0. Reset type: SYSRSn

19 ADCPPB3RESULT Register (Offset = 48h) [Reset = 00000000h]

ADCPPB3RESULT is shown in Figure 16-115 and described in Table 16-108.

Return to the Summary Table.

ADC Post Processing Block 3 Result Register

Figure 16-115. ADCPPB3RESULT Register

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

Table 16-200. ADCPPB3RESULT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-13	SIGN	R	0h	Sign Extended Bits. These bits reflect the same value as bit 12. NOTE: If the conversion associated with this Post Processing Block is a 12-bit conversion, the SIGN bits extend down to bit 12, and all reflect the same value as bit 12. Reset type: SYSRSn
12-0	PPBRESULT	R	0h	ADC Post Processing Block Result 3 The result of the offset/reference subtraction post conversion processing is stored in this register. This result is available 1 MCLK cycle after the associated ADCRESULT is available, unless multiple PPBs point to the same SOC. In the case of multiple PPBs associated with the same SOC, the lowest numbered PPB's result will be availble 1 MCLK cycle after the associated ADCRESULT and subsequent results (in order from lowest numbered PPB to highest) will each become available every 2-3 MCLK cycles (refer to the TRM for more detailed timing information). If ADCINTFLG is polled to determine when to read the PPBRESULT, it may be necessary to add one or more NOP instructions to ensure that the updated post conversion processing result has posted to the register. NOTE: If the conversion associated with this Post Processing Block is a 12-bit conversion, the PPBRESULT bits are limited to bits 12:0. Reset type: SYSRSn

20 ADCPPB4RESULT Register (Offset = 4Ch) [Reset = 00000000h]

ADCPPB4RESULT is shown in Figure 16-116 and described in Table 16-109.

Return to the Summary Table.

ADC Post Processing Block 4 Result Register

Figure 16-116. ADCPPB4RESULT Register

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

Table 16-202. ADCPPB4RESULT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-13	SIGN	R	0h	Sign Extended Bits. These bits reflect the same value as bit 12. NOTE: If the conversion associated with this Post Processing Block is a 12-bit conversion, the SIGN bits extend down to bit 12, and all reflect the same value as bit 12. Reset type: SYSRSn
12-0	PPBRESULT	R	0h	ADC Post Processing Block Result 4 The result of the offset/reference subtraction post conversion processing is stored in this register. This result is available 1 MCLK cycle after the associated ADCRESULT is available, unless multiple PPBs point to the same SOC. In the case of multiple PPBs associated with the same SOC, the lowest numbered PPB's result will be availble 1 MCLK cycle after the associated ADCRESULT and subsequent results (in order from lowest numbered PPB to highest) will each become available every 2-3 MCLK cycles (refer to the TRM for more detailed timing information). If ADCINTFLG is polled to determine when to read the PPBRESULT, it may be necessary to add one or more NOP instructions to ensure that the updated post conversion processing result has posted to the register. NOTE: If the conversion associated with this Post Processing Block is a 12-bit conversion, the PPBRESULT bits are limited to bits 12:0. Reset type: SYSRSn

21 ADCPPB1SUM Register (Offset = 50h) [Reset = 00000000h]

ADCPPB1SUM is shown in Figure 16-117 and described in Table 16-110.

Return to the Summary Table.

ADC PPB 1 Final Sum Result Register

Figure 16-117. ADCPPB1SUM Register

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

Table 16-204. ADCPPB1SUM Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	SIGN	R	0h	Sign Extended Bits. These bits reflect the same value as bit 15. Reset type: SYSRSn
15-0	SUM	R	0h	Post Processing Block 1 Oversampling Final Sum. When either a count-match event occurs (PCOUNT = LIMIT) or PPB1 receives a sync. event, the value of PSUM is loaded into this register. In the case of a count-match event, the sum loaded into this register includes the value from the most recent conversion. The value from PSUM will be right shifted by the amount specified in the SHIFT register before being loaded into the final SUM result register. This result is available 1 MCLK cycle after the associated ADCPPB1RESULT is available (only in case of a count-match event). This will be 2 MCLK cycles after the associated ADCRESULT is available, unless multiple PPBs point to the same SOC (refer to the ADCPPB1RESULT timing information). Reset type: SYSRSn

22 ADCPPB1COUNT Register (Offset = 54h) [Reset = 0000h]

ADCPPB1COUNT is shown in Figure 16-118 and described in Table 16-111.

Return to the Summary Table.

ADC PPB1 Final Conversion Count Register

Figure 16-118. ADCPPB1COUNT Register

15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED				COUNT			
R-0h				R-0h			

Table 16-206. ADCPPB1COUNT Register Field Descriptions

Bit	Field	Type	Reset	Description
15-4	RESERVED	R	0h	Reserved
3-0	COUNT	R	0h	Post Processing Block 1 Oversampling Final Count. When either a count-match event occurs (PCOUNT = LIMIT) or PPB1 receives a sync. event, the value of PCOUNT is loaded into this register. This result is available 1 MCLK cycle after the associated ADCPPB1RESULT is available (only in case of a count-match event). This will be 2 MCLK cycles after the associated ADCRESULT is available, unless multiple PPBs point to the same SOC (refer to the ADCPPB1RESULT timing information). Reset type: SYSRSn

23 ADCPPB2SUM Register (Offset = 58h) [Reset = 00000000h]

ADCPPB2SUM is shown in Figure 16-119 and described in Table 16-112.

Return to the Summary Table.

ADC PPB 2 Final Sum Result Register

Figure 16-119. ADCPPB2SUM Register

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

Table 16-208. ADCPPB2SUM Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	SIGN	R	0h	Sign Extended Bits. These bits reflect the same value as bit 15. Reset type: SYSRSn
15-0	SUM	R	0h	Post Processing Block 2 Oversampling Final Sum. When either a count-match event occurs (PCOUNT = LIMIT) or PPB2 receives a sync. event, the value of PSUM is loaded into this register. In the case of a count-match event, the sum loaded into this register includes the value from the most recent conversion. The value from PSUM will be right shifted by the amount specified in the SHIFT register before being loaded into the final SUM result register. This result is available 1 MCLK cycle after the associated ADCPPB2RESULT is available (only in case of a count-match event). This will be 2 MCLK cycles after the associated ADCRESULT is available, unless multiple PPBs point to the same SOC (refer to the ADCPPB2RESULT timing information). Reset type: SYSRSn

24 ADCPPB2COUNT Register (Offset = 5Ch) [Reset = 0000h]

ADCPPB2COUNT is shown in Figure 16-120 and described in Table 16-113.

Return to the Summary Table.

ADC PPB2 Final Conversion Count Register

Figure 16-120. ADCPPB2COUNT Register

15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED				COUNT			
R-0h				R-0h			

Table 16-210. ADCPPB2COUNT Register Field Descriptions

Bit	Field	Type	Reset	Description
15-4	RESERVED	R	0h	Reserved
3-0	COUNT	R	0h	Post Processing Block 2 Oversampling Final Count. When either a count-match event occurs (PCOUNT = LIMIT) or PPB2 receives a sync. event, the value of PCOUNT is loaded into this register. This result is available 1 MCLK cycle after the associated ADCPPB2RESULT is available (only in case of a count-match event). This will be 2 MCLK cycles after the associated ADCRESULT is available, unless multiple PPBs point to the same SOC (refer to the ADCPPB2RESULT timing information). Reset type: SYSRSn

25 ADCPPB3SUM Register (Offset = 60h) [Reset = 00000000h]

ADCPPB3SUM is shown in Figure 16-121 and described in Table 16-114.

Return to the Summary Table.

ADC PPB 3 Final Sum Result Register

Figure 16-121. ADCPPB3SUM Register

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

Table 16-212. ADCPPB3SUM Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	SIGN	R	0h	Sign Extended Bits. These bits reflect the same value as bit 15. Reset type: SYSRSn
15-0	SUM	R	0h	Post Processing Block 3 Oversampling Final Sum. When either a count-match event occurs (PCOUNT = LIMIT) or PPB3 receives a sync. event, the value of PSUM is loaded into this register. In the case of a count-match event, the sum loaded into this register includes the value from the most recent conversion. The value from PSUM will be right shifted by the amount specified in the SHIFT register before being loaded into the final SUM result register. This result is available 1 MCLK cycle after the associated ADCPPB3RESULT is available (only in case of a count-match event). This will be 2 MCLK cycles after the associated ADCRESULT is available, unless multiple PPBs point to the same SOC (refer to the ADCPPB3RESULT timing information). Reset type: SYSRSn

26 ADCPPB3COUNT Register (Offset = 64h) [Reset = 0000h]

ADCPPB3COUNT is shown in Figure 16-122 and described in Table 16-115.

Return to the Summary Table.

ADC PPB3 Final Conversion Count Register

Figure 16-122. ADCPPB3COUNT Register

15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED				COUNT			
R-0h				R-0h			

Table 16-214. ADCPPB3COUNT Register Field Descriptions

Bit	Field	Type	Reset	Description
15-4	RESERVED	R	0h	Reserved
3-0	COUNT	R	0h	Post Processing Block 3 Oversampling Final Count. When either a count-match event occurs (PCOUNT = LIMIT) or PPB3 receives a sync. event, the value of PCOUNT is loaded into this register. This result is available 1 MCLK cycle after the associated ADCPPB3RESULT is available (only in case of a count-match event). This will be 2 MCLK cycles after the associated ADCRESULT is available, unless multiple PPBs point to the same SOC (refer to the ADCPPB3RESULT timing information). Reset type: SYSRSn

27 ADCPPB4SUM Register (Offset = 68h) [Reset = 00000000h]

ADCPPB4SUM is shown in Figure 16-123 and described in Table 16-116.

Return to the Summary Table.

ADC PPB 4 Final Sum Result Register

Figure 16-123. ADCPPB4SUM Register

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

Table 16-216. ADCPPB4SUM Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	SIGN	R	0h	Sign Extended Bits. These bits reflect the same value as bit 15. Reset type: SYSRSn
15-0	SUM	R	0h	Post Processing Block 4 Oversampling Final Sum. When either a count-match event occurs (PCOUNT = LIMIT) or PPB4 receives a sync. event, the value of PSUM is loaded into this register. In the case of a count-match event, the sum loaded into this register includes the value from the most recent conversion. The value from PSUM will be right shifted by the amount specified in the SHIFT register before being loaded into the final SUM result register. This result is available 1 MCLK cycle after the associated ADCPPB4RESULT is available (only in case of a count-match event). This will be 2 MCLK cycles after the associated ADCRESULT is available, unless multiple PPBs point to the same SOC (refer to the ADCPPB4RESULT timing information). Reset type: SYSRSn

28 ADCPPB4COUNT Register (Offset = 6Ch) [Reset = 0000h]

ADCPPB4COUNT is shown in Figure 16-124 and described in Table 16-117.

Return to the Summary Table.

ADC PPB4 Final Conversion Count Register

Figure 16-124. ADCPPB4COUNT Register

15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED				COUNT			
R-0h				R-0h			

Table 16-218. ADCPPB4COUNT Register Field Descriptions

Bit	Field	Type	Reset	Description
15-4	RESERVED	R	0h	Reserved
3-0	COUNT	R	0h	Post Processing Block 4 Oversampling Final Count. When either a count-match event occurs (PCOUNT = LIMIT) or PPB4 receives a sync. event, the value of PCOUNT is loaded into this register. This result is available 1 MCLK cycle after the associated ADCPPB4RESULT is available (only in case of a count-match event). This will be 2 MCLK cycles after the associated ADCRESULT is available, unless multiple PPBs point to the same SOC (refer to the ADCPPB4RESULT timing information). Reset type: SYSRSn

29 ADCSEQ1FIFORESULT Register (Offset = C0h) [Reset = 00000000h]

ADCSEQ1FIFORESULT is shown in Figure 16-125 and described in Table 16-118.

Return to the Summary Table.

ADC Sequence 1 FIFO Result Register

Figure 16-125. ADCSEQ1FIFORESULT Register

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

Table 16-220. ADCSEQ1FIFORESULT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	SEQFIFORESULT	R	0h	Sequencer 1 Result in FIFO mode Reset type: SYSRSn

30 ADCSEQ2FIFORESULT Register (Offset = C4h) [Reset = 00000000h]

ADCSEQ2FIFORESULT is shown in Figure 16-126 and described in Table 16-119.

Return to the Summary Table.

ADC Sequence 2 FIFO Result Register

Figure 16-126. ADCSEQ2FIFORESULT Register

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

Table 16-222. ADCSEQ2FIFORESULT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	SEQFIFORESULT	R	0h	Sequencer 2 Result in FIFO mode Reset type: SYSRSn

ADVANCE INFORMATION

31 ADCSEQ3FIFORESULT Register (Offset = C8h) [Reset = 00000000h]

ADCSEQ3FIFORESULT is shown in Figure 16-127 and described in Table 16-120.

Return to the Summary Table.

ADC Sequence 3 FIFO Result Register

Figure 16-127. ADCSEQ3FIFORESULT Register

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

Table 16-224. ADCSEQ3FIFORESULT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	SEQFIFORESULT	R	0h	Sequencer 3 Result in FIFO mode Reset type: SYSRSn

ADVANCE INFORMATION

32 ADCSEQ4FIFORESULT Register (Offset = CCh) [Reset = 00000000h]

ADCSEQ4FIFORESULT is shown in Figure 16-128 and described in Table 16-121.

Return to the Summary Table.

ADC Sequence 4 FIFO Result Register

Figure 16-128. ADCSEQ4FIFORESULT Register

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

Table 16-226. ADCSEQ4FIFORESULT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	SEQFIFORESULT	R	0h	Sequencer 4 Result in FIFO mode Reset type: SYSRSn

ADVANCE INFORMATION

33 ADCSEQ1FIFOSTATUS Register (Offset = D0h) [Reset = 00000000h]

ADCSEQ1FIFOSTATUS is shown in Figure 16-129 and described in Table 16-122.

Return to the Summary Table.

ADC Sequence 1 FIFO Status

Figure 16-129. ADCSEQ1FIFOSTATUS Register

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

Table 16-228. ADCSEQ1FIFOSTATUS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-5	RESERVED	R	0h	Reserved
4-1	SEQFIFOPENDREAD	R	0h	Sequencer 1 FIFO pending reads Note: 1. This field is only updated when after the last byte of ADCSEQ1FIFORESULT is read. 2. The Pending read information will not be updated for debug reads. 3. Once all the data in the FIFO has been read, FIFO will wrap back. Subsequent reads will read from the beginning of the FIFO. Reset type: SYSRSn
0	SEQFIFOVALID	R	0h	Sequencer 1 FIFO Valid 0 - All Conversions associated with the current Sequencer are NOT complete. 1 - Conversion from all the SOCs associated with the Sequencer are complete and result is available to be read. Reset type: SYSRSn

34 ADCSEQ2FIFOSTATUS Register (Offset = D4h) [Reset = 00000000h]

ADCSEQ2FIFOSTATUS is shown in Figure 16-130 and described in Table 16-123.

Return to the Summary Table.

ADC Sequence 2 FIFO Status

Figure 16-130. ADCSEQ2FIFOSTATUS Register

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

Table 16-230. ADCSEQ2FIFOSTATUS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-5	RESERVED	R	0h	Reserved
4-1	SEQFIFOPENDREAD	R	0h	Sequencer 2 FIFO pending reads Note: 1. This field is only updated when after the last byte of ADCSEQ2FIFORESULT is read. 2. The Pending read information will not be updated for debug reads. 3. Once all the data in the FIFO has been read, FIFO will wrap back. Subsequent reads will read from the beginning of the FIFO. Reset type: SYSRSn
0	SEQFIFOVALID	R	0h	Sequencer 2 FIFO Valid 0 - All Conversions associated with the current Sequencer are NOT complete. 1 - Conversion from all the SOCs associated with the Sequencer are complete and result is available to be read. Reset type: SYSRSn

35 ADCSEQ3FIFOSTATUS Register (Offset = D8h) [Reset = 00000000h]

ADCSEQ3FIFOSTATUS is shown in Figure 16-131 and described in Table 16-124.

Return to the Summary Table.

ADC Sequence 3 FIFO Status

Figure 16-131. ADCSEQ3FIFOSTATUS Register

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

Table 16-232. ADCSEQ3FIFOSTATUS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-5	RESERVED	R	0h	Reserved
4-1	SEQFIFOPENDREAD	R	0h	Sequencer 3 FIFO pending reads Note: 1. This field is only updated when after the last byte of ADCSEQ3FIFORESULT is read. 2. The Pending read information will not be updated for debug reads. 3. Once all the data in the FIFO has been read, FIFO will wrap back. Subsequent reads will read from the beginning of the FIFO. Reset type: SYSRSn
0	SEQFIFOVALID	R	0h	Sequencer 3 FIFO Valid 0 - All Conversions associated with the current Sequencer are NOT complete. 1 - Conversion from all the SOCs associated with the Sequencer are complete and result is available to be read. Reset type: SYSRSn

36 ADCSEQ4FIFOSTATUS Register (Offset = DCh) [Reset = 00000000h]

ADCSEQ4FIFOSTATUS is shown in Figure 16-132 and described in Table 16-125.

Return to the Summary Table.

ADC Sequence 4 FIFO Status

Figure 16-132. ADCSEQ4FIFOSTATUS Register

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

Table 16-234. ADCSEQ4FIFOSTATUS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-5	RESERVED	R	0h	Reserved
4-1	SEQFIFOPENDREAD	R	0h	Sequencer 4 FIFO pending reads Note: 1. This field is only updated when after the last byte of ADCSEQ4FIFORESULT is read. 2. The Pending read information will not be updated for debug reads. 3. Once all the data in the FIFO has been read, FIFO will wrap back. Subsequent reads will read from the beginning of the FIFO. Reset type: SYSRSn
0	SEQFIFOVALID	R	0h	Sequencer 4 FIFO Valid 0 - All Conversions associated with the current Sequencer are NOT complete. 1 - Conversion from all the SOCs associated with the Sequencer are complete and result is available to be read. Reset type: SYSRSn

Chapter 17

Comparator Subsystem (CMPSS)

The Comparator Subsystem (CMPSS) consists of analog comparators and supporting circuits that are useful for power applications such as peak current mode control, switched-mode power, power factor correction, voltage trip monitoring, and so forth.

ADVANCE INFORMATION

17.1 Introduction

The comparator subsystem is built around a number of modules. Each subsystem contains two comparators, two reference 12-bit DACs, and two digital filters. Comparators are denoted "H" or "L" within each module where “H” and “L” represent high and low, respectively. Each comparator generates a digital output which indicates whether the voltage on the positive input is greater than the voltage on the negative input. The positive input of the comparator is driven from or by the PGA (see the *Analog Subsystem* chapter for mux options available to the CMPSS).. Each comparator output passes through a programmable digital filter that can remove spurious trip signals. An unfiltered output is also available if filtering is not required.

17.1.1 Features

Each CMPSS includes:

- Two analog comparators
- Two programmable reference 12-bit DACs (These DAC instances are 12-bit with a lower effective resolution as described in the data sheet)
- Two digital filters, max filter clock prescale = 2^{16}
- Ability to synchronize submodules with MCPWMSYNCPER
- Ability to synchronize output with MCLK
- Ability to latch output
- Ability to invert output
- Option to use hysteresis on the input
- Option for negative input of comparator to be driven by an external signal or by the reference DAC
- VDDA is the DAC reference voltage
- Option to use the low comparator DAC output on an external pin (select instances only, mutually exclusive with use of comparator functionality)

17.1.2 CMPSS Related Collateral

Foundational Materials

- *Real-Time Control Reference Guide*
  - Refer to the Comparator section

Expert Materials

- *Peak Current Mode Controlled PSFB Converter Reference Design Using C2000™ Real-time MCU*
- *Understanding and Applying Current-Mode Control Theory Application Report*

17.1.3 Block Diagram

The block diagram for the CMPSS is shown in Figure 17-1Figure 17-2.

- CTRIPx (x= "H" or "L") signals are connected to the PWM X-BAR for MCPWM trip response. See the *Multi-Channel Pulse Width Modulator (MCPWM)* chapter for more details on the PWM X-BAR mux configuration.
- CTRIPxOUTx(x= "H" or "L") signals are connected to the Output X-BAR for external signaling. See the *General-Purpose Input/Output (GPIO)* chapter for more details on the Output X-BAR mux configuration.
- CMP3_LITE_DACL is the low comparator DAC output that is available only in the CMPSS3 module. To enable this DAC output, set the CMPSSCTL.CMP3LDACOUTEN register field in SYSCTL.
- CMP4_LITE_DACL is the low comparator DAC output that is available only in the CMPSS4 module. To enable this DAC output, set the CMPSSCTL.CMP4LDACOUTEN register field in SYSCTL.

Note

Enabling the CMP[3:4]_LITE_DACLCMP3_LITE_DACL to a pin disables the functionality to the associated COMPL in CMPSS. In this case, the inverting input of the COMPL must come from the CMPx_LN input pin. All other functions of the CMPSS module are retained.

Figure 17-1. CMPSS Module Block Diagram

ADVANCE INFORMATION

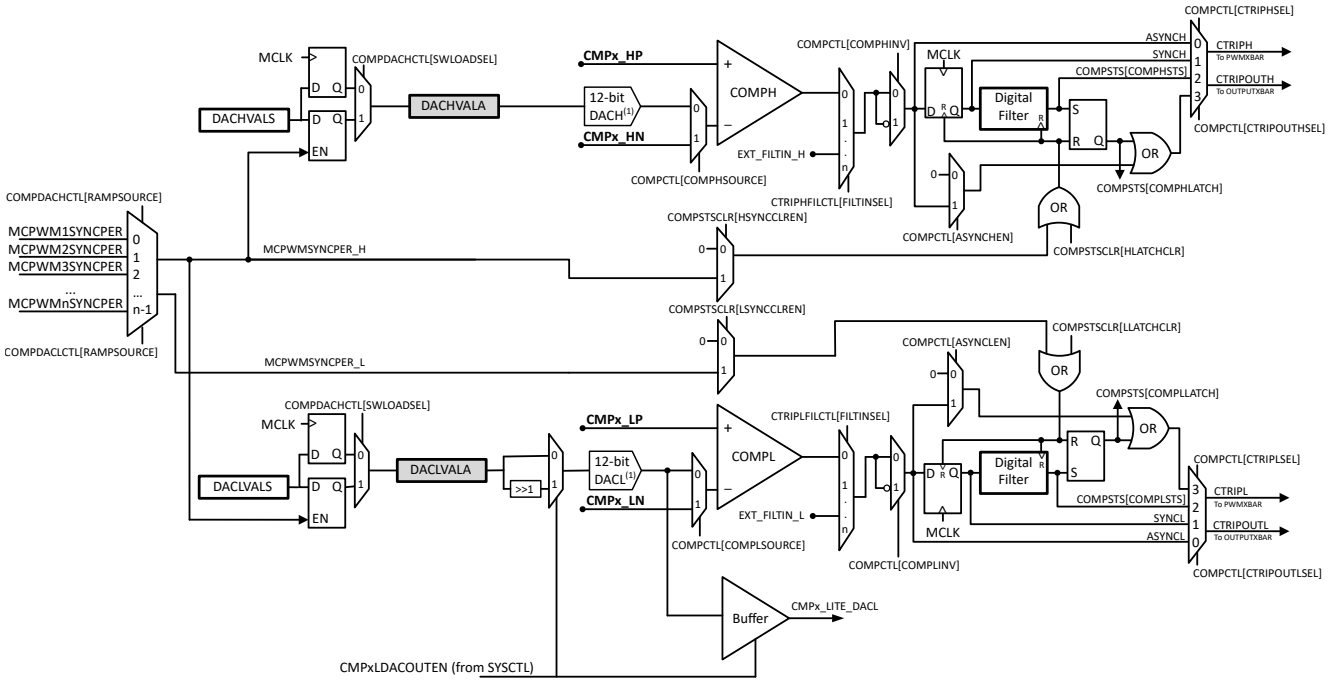


Figure 17-2. CMPSS_LITE Module Block Diagram

17.2 Comparator

The comparator generates a high digital output when the voltage on the positive input is greater than the voltage on the negative input, and a low digital output when the voltage on the positive input is less than the voltage on the negative input. The comparator is illustrated in Figure 17-3.

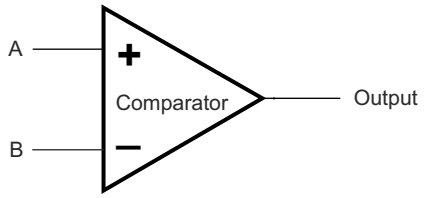


Figure 17-3. Comparator Block Diagram

Voltages	Output
Voltage A > Voltage B	1
Voltage A < Voltage B	0

ADVANCE INFORMATION

17.3 Reference DAC

Each 12-bit reference DAC can be configured to drive a reference voltage into the negative input of the respective comparator. Some CMPSS instances allow the low DAC output to be routed to a pin to act as an external DAC. In this case, all other CMPSS module functionality is not useable, including the high DAC, both comparators, and the digital filters.

Two sets of DACxVAL registers, DACxVALA and DACxVALS, are present for each reference 12-bit DAC. DACxVALA is a read-only register that actively controls the reference 12-bit DAC value. DACxVALS is a writable shadow register that loads into DACxVALA either immediately or synchronized with the next MCPWMSYNCPER event.

The operating range of the reference 12-bit DAC is bounded by DACREF and VSSA. The reference 12-bit DAC is illustrated in Figure 17-4.

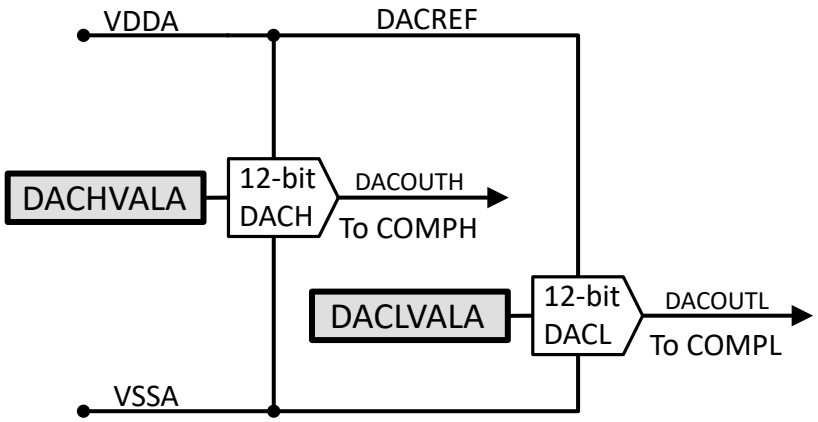


Figure 17-4. Reference DAC Block Diagram

The output of the reference 12-bit DAC can be calculated as:

$$DACOUT = \frac{(1 + DACVALA)*DACREF}{4096} \quad (20)$$

Note

- In the situations where both the DACH and DACL are driving the high and low comparators, a trip on one comparator can temporarily disturb the DAC output of the other comparator. The amount and length of time of this disturbance is specified in the device data sheet as “CMPSS DAC Output Disturbance” and “CMPSS DAC Disturbance Time”, respectively.

  Users must carefully design the system so that if the input signal crosses either DACH or DACL and trips the associated comparator, the input signal stays more than a “*CMPSS DAC output disturbance*” away from the other comparator trip point for the minimum length of “*CMPSS DAC disturbance time*”.
- If the user is not using DACL, the DACLVALS register must be set to 0 to avoid COMPL from affecting and tripping DACH. In this case, there is no limitation on the DACHVALS setting. Similarly, when not using DACH, the user must set DACHVALS to the maximum value.
- The CMPSS instance can be enabled before programming the reference DAC values.

17.4 Digital Filter

The digital filter works on a window of FIFO samples (SAMPWIN) taken from the input. The filter output resolves to the majority value of the sample window, where majority is defined by the threshold (THRESH) value. If the majority threshold is not satisfied, the filter output remains unchanged.

For proper operation, the value of THRESH must be greater than SAMPWIN / 2 and less than or equal to SAMPWIN.

A prescale function (CLKPRESCALE) determines the filter sampling rate, where the filter FIFO captures one sample every prescaled system clock cycle. Old data from the FIFO is discarded.

Note that for SAMPWIN, THRESH, and CLKPRESCALE, the internal number used by the digital filter is + 1 in all cases. In essence, samples = SAMPWIN + 1, threshold = THRESH + 1 and prescale = CLKPRESCALE + 1.

A conceptual model of the digital filter is shown in Figure 17-5.

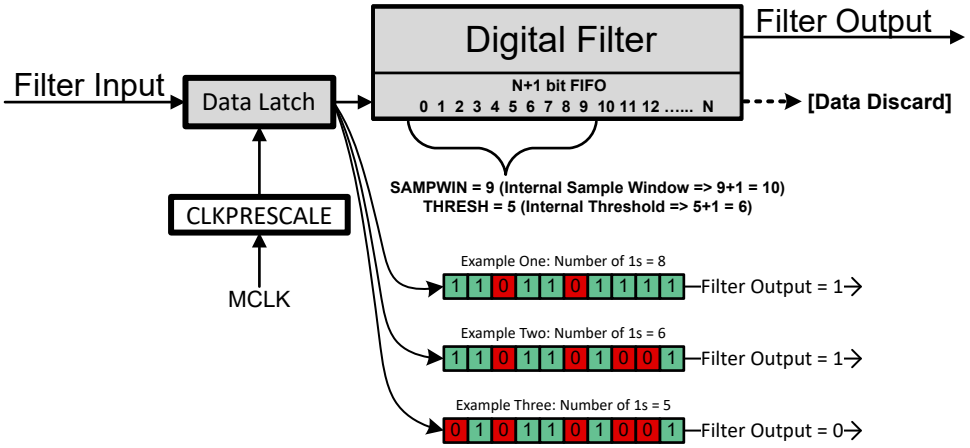


Figure 17-5. Digital Filter Behavior

Equivalent C code of the filter implementation is:

```
if (FILTER_OUTPUT == 0) {
    if (Num_1s_in_SAMPWIN >= THRESH) {
        FILTER_OUTPUT = 1;
    }
}
else {
    if (Num_0s_in_SAMPWIN >= THRESH) {
        FILTER_OUTPUT = 0;
    }
}
```

ADVANCE INFORMATION

17.4.1 Filter Initialization Sequence

For proper operation of the digital filter, the following initialization sequence is recommended:

1. Configure and enable the comparator for operation.
2. Configure the digital filter parameters for operation:
   - Set SAMPWIN for the number of samples to monitor in the FIFO window.
   - Set THRESH for the threshold required for majority qualification.
   - Set CLKPRESCALE for the digital filter clock prescale value.
3. Initialize the sample values in the digital FIFO window by setting FILINIT.
4. Clear COMPSTS latch using COMPSTSCLR, if the latched path is desired.
5. Configure the CTRIP and CTRIPOUT signal paths.
6. If desired, configure the destination module, for example, MCPWM, GPIO, and so on to accept the filtered signals.

17.5 Using the CMPSS

17.5.1 LATCHCLR, and MCPWMSYNCPER Signals

The LATCHCLR signal holds the digital filter, synchronization block, and the latch output in reset (0) after the required delays. The LATCHCLR signal is activated in software using xLATCHCLR (x = H or L). The LATCHCLR signal can also be activated by MCPWMSYNCPER when xSYNCCLREN (x = H or L) is set. If a longer LATCHCLR signal is required, the EPWMBLANK signal can be used to extend the LATCHCLR signal by setting BLANKEN.

MCPWMxSYNCPER comes from the Time-Base submodule of the MCPWM, respectively. For a detailed description of how these two signals are generated, refer to the Time Base Counter Synchronization subsection in the *Multi-Channel Pulse Width Modulator (MCPWM)* chapter.

The MCPWMxSYNCPER signal that loads DACxVALA when COMPDACCTL[SWLOADSEL] = 1 is a level trigger load. If TBCTR and TBPRD of the MCPWM are both 0, MCPWMSYNCPER is held at level high and DACxVALA is loaded immediately from DACxVALS irrespective of the value of COMPDACCTL[SWLOADSEL]. Due to this, configure the MCPWM first before setting COMPDACCTL[SWLOADSEL] to 1.

Note

The name of the sync signal that the CMPSS receives from the MCPWM is referred to as MCPWMSYNCPER, PWMSYNCPER, and MCPWMxSYNCPER interchangeably in this document.

17.5.2 Synchronizer, Digital Filter, and Latch Delays

The synchronization block adds 1-2 MCLK cycle delays. If the digital filter is bypassed (all filter settings are 0), the digital filter adds 2 MCLK cycle delays. The latch adds 1 MCLK cycle delay.

17.5.3 Calibrating the CMPSS

The CMPSS has two sources of offset errors: comparator offset error and compdac offset error. In the data sheet, the comparator offset error is referred to as ***Input referred offset error*** and compdac offset error is referred to as ***Static offset error***. See the device data sheet for the values.

If both inputs of the comparator are driven from a pin, only the comparator offset error applies. However if the inverting input of the comparator is driven from the compdac, then only the compdac offset error applies. This is because the compdac offset error includes comparator offset error.

Due to the offset errors, the CMPSS must be calibrated to make sure trips happen at the expected levels. The following flow outlines how the calibration can be performed if the inverting input of the comparator is driven from the compdac.

Notes before calibration:

1. A static DC signal is required on the non-inverting input of the comparator.
2. Hysteresis can be disabled for calibration and can be re-enabled after calibration is complete.
3. A noisy input can make calibration difficult, so use the latch with non-zero filter settings depending on how noisy is the signal on the non-inverting input.

This approach sweeps down the compdac:

1. Set the starting compdac value to max, 0xFFF.
   - Optional: Instead of setting the starting compdac value to maximum, set to ***Vtarget*** + ***Static offset error*** + ***Margin***. Where ***Vtarget*** is the approximate DC voltage on the non-inverting input, ***Static offset error*** is the compdac offset error specification and ***Margin*** is some amount of guard band. This can lead to a faster calibration but only works if ***Vtarget*** is known. Alternatively, if ***Vtarget*** is unknown, the ADC can be used to convert ***Vtarget***.
2. Decrement compdac value by 1.
3. Wait for compdac to settle.
4. Clear latch.
5. Wait for possible latch set.
6. If latch is set, trip code is found exit.
   - Optional: The trip code can be double checked by:
     a. Increasing compdac value by 1.
     b. Clear latch.
     c. Wait for possible latch set.
     d. Latch can be unset.
7. If latch is unset, go back to step 2 and repeat.

It is also possible to calibrate the CMPSS, if both inputs of the comparator are driven from a pin. For this case, the flow stays the same but the voltage on the inverting pin of the comparator is swept externally.

17.5.4 Enabling and Disabling the CMPSS Clock

If the clock to the CMPSS module is disabled while the comparator is active, the following behavior can be expected:

- The comparator remains unaffected and continues to trip from voltages on the inputs.
- If the reference 12-bit DAC is driving the negative input of the comparator, the voltage on the negative input remains static and unaffected but DACVALA can no longer be updated from the ramp generator or DACVALS.
- The ramp generator, synchronize block and digital filter freeze on the current states.

Enabling the clock to the CMPSS restores the clock to the state before the clock was disabled.

17.6 CMPSS DAC Output

Select instances of the CMPSS_LITE module can optionally route the low DAC output to an external pin to be used as an external DAC. This is subject to the following constraints:

- This is available only on select instances; consult the device data sheet to determine if the CMPSS DAC has been pinned out for a particular instance.
- In this case, the DAC output is not available to the COMPL inside the CMPSS module. The negative input to COMPL must be driven from the device pin in this case, if used. All other CMPSS module functionality, including the COMPH DAC source, is usable in this case.
- See the device data sheet for effective resolution of the CMPSS DAC output.

Note

The CMPSS low DAC output does not necessarily appear on the same pins that are available for low comparator input. Consult the pinout of the device data sheet to determine which pin the CMPSS output DAC is available on.

17.7 CMPSS Registers

This Section describes the CMPSS Registers.

17.7.1 CMPSS Base Address Table

Table 17-1. CMPSS Base Address Table

Bit Field Name		DriverLib Name	Base Address
Instance	Structure		
Cmpss0Regs	CMPSS_LITE_REGS	CMPSS0	0x4046_0000
Cmpss1Regs	CMPSS_LITE_REGS	CMPSS1	0x4046_1000
Cmpss2Regs	CMPSS_LITE_REGS	CMPSS2	0x4046_2000
Cmpss3Regs	CMPSS_LITE_REGS	CMPSS3	0x4046_3000

17.7.2 CMPSS_LITE_REGS Registers

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

Table 17-2. CMPSS_LITE_REGS Registers

Offset	Acronym	Register Name	Write Protection	Section
0h	COMPCTL	CMPSS Comparator Control Register	EALLOW	Go
2h	COMPHYSCTL	CMPSS Comparator Hysteresis Control Register	EALLOW	Go
4h	COMPSTS	CMPSS Comparator Status Register		Go
6h	COMPSTSCLR	CMPSS Comparator Status Clear Register	EALLOW	Go
8h	COMPDACHCTL	CMPSS High DAC Control Register	EALLOW	Go
Ch	DACHVALS	CMPSS High DAC Value Shadow Register		Go
Eh	DACHVALA	CMPSS High DAC Value Active Register		Go
24h	DACLVALS	CMPSS Low DAC Value Shadow Register		Go
26h	DACLVALA	CMPSS Low DAC Value Active Register		Go
2Ch	CTRIPLFILCTL	CTRIPL Filter Control Register	EALLOW	Go
2Eh	CTRIPLFILCLKCTL	CTRIPL Filter Clock Control Register	EALLOW	Go
30h	CTRIPHFILCTL	CTRIPH Filter Control Register	EALLOW	Go
32h	CTRIPHFILCLKCTL	CTRIPH Filter Clock Control Register	EALLOW	Go
34h	COMPLOCK	CMPSS Lock Register	EALLOW	Go
48h	COMPDACLCTL	CMPSS Low DAC Control Register	EALLOW	Go
6Eh	CTRIPLFILCLKCTL2	CTRIPL Filter Clock Control Register 2	EALLOW	Go
72h	CTRIPHFILCLKCTL2	CTRIPH Filter Clock Control Register 2	EALLOW	Go

Complex bit access types are encoded to fit into small table cells. Table 17-3 shows the codes that are used for access types in this section.

Table 17-3. CMPSS_LITE_REGS Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
R-0	R -0	Read Returns 0s
Write Type		
W	W	Write
W1S	W 1S	Write 1 to set
WSonce	W Sonce	Write Set once
Reset or Default Value		
-n		Value after reset or the default value

1 COMPCTL Register (Offset = 0h) [Reset = 0000h]

COMPCTL is shown in Figure 17-6 and described in Table 17-4.

Return to the Summary Table.

CMPSS Comparator Control Register

Figure 17-6. COMPCTL Register

15	14	13	12	11	10	9	8
COMPDACE	ASYNCLEN	CTRIPOUTLSEL		CTRIPLSEL		COMPLINV	COMPLSOURCE
R/W-0h	R/W-0h	R/W-0h		R/W-0h		R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
RESERVED	ASYNCHEN	CTRIPOUTHSEL		CTRIPHSEL		COMPHINV	COMPHSOURCE
R-0h	R/W-0h	R/W-0h		R/W-0h		R/W-0h	R/W-0h

Table 17-5. COMPCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
15	COMPDACE	R/W	0h	Comparator/DAC enable. 0 Comparator/DAC disabled 1 Comparator/DAC enabled Reset type: SYSRSn
14	ASYNCLEN	R/W	0h	Low comparator asynchronous path enable. Allows asynchronous comparator output to feed into OR gate with latched digital filter signal when CTRIPLSEL=3 or CTRIPOUTLSEL=3. 0 Asynchronous comparator output does not feed into OR gate with latched digital filter output 1 Asynchronous comparator output feeds into OR gate with latched digital filter output Reset type: SYSRSn
13-12	CTRIPOUTLSEL	R/W	0h	Low comparator CTRIPOUTL source select. 0 Asynchronous comparator output drives CTRIPOUTL 1 Synchronous comparator output drives CTRIPOUTL 2 Output of digital filter drives CTRIPOUTL 3 Latched output of digital filter drives CTRIPOUTL Reset type: SYSRSn
11-10	CTRIPLSEL	R/W	0h	Low comparator CTRIPL source select. 0 Asynchronous comparator output drives CTRIPL 1 Synchronous comparator output drives CTRIPL 2 Output of digital filter drives CTRIPL 3 Latched output of digital filter drives CTRIPL Reset type: SYSRSn
9	COMPLINV	R/W	0h	Low comparator output invert. 0 Output of comparator is not inverted 1 Output of comparator is inverted Reset type: SYSRSn
8	COMPLSOURCE	R/W	0h	Low comparator input source. 0 Inverting input of comparator driven by internal DAC 1 Inverting input of comparator driven through external pin Reset type: SYSRSn
7	RESERVED	R	0h	Reserved

ADVANCE INFORMATION

Table 17-5. COMPCTL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
6	ASYNCHEN	R/W	0h	High comparator asynchronous path enable. Allows asynchronous comparator output to feed into OR gate with latched digital filter signal when CTRIPHSEL=3 or CTRIPOUTHSEL=3. 0 Asynchronous comparator output does not feed into OR gate with latched digital filter output 1 Asynchronous comparator output feeds into OR gate with latched digital filter output Reset type: SYSRSn
5-4	CTRIPOUTHSEL	R/W	0h	High comparator CTRIPOUTH source select. 0 Asynchronous comparator output drives CTRIPOUTH 1 Synchronous comparator output drives CTRIPOUTH 2 Output of digital filter drives CTRIPOUTH 3 Latched output of digital filter drives CTRIPOUTH Reset type: SYSRSn
3-2	CTRIPHSEL	R/W	0h	High comparator CTRIPH source select. 0 Asynchronous comparator output drives CTRIPH 1 Synchronous comparator output drives CTRIPH 2 Output of digital filter drives CTRIPH 3 Latched output of digital filter drives CTRIPH Reset type: SYSRSn
1	COMPHINV	R/W	0h	High comparator output invert. 0 Output of comparator is not inverted 1 Output of comparator is inverted Reset type: SYSRSn
0	COMPHSOURCE	R/W	0h	High comparator input source. 0 Inverting input of comparator driven by internal DAC 1 Inverting input of comparator driven through external pin Reset type: SYSRSn

2 COMPHYSCTL Register (Offset = 2h) [Reset = 0000h]

COMPHYSCTL is shown in Figure 17-7 and described in Table 17-5.

Return to the Summary Table.

CMPSS Comparator Hysteresis Control Register

Figure 17-7. COMPHYSCTL Register

15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED				COMPHYS			
R-0h				R/W-0h			

Table 17-7. COMPHYSCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
15-4	RESERVED	R	0h	Reserved
3-0	COMPHYS	R/W	0h	Comparator hysteresis. Sets the amount of hysteresis on the comparator inputs. 0 None 1 Set to typical hysteresis 2 Set to 2x of typical hysteresis 3 Set to 3x of typical hysteresis 4 Set to 4x of typical hysteresis others : undefined Reset type: SYSRSn

3 COMPSTS Register (Offset = 4h) [Reset = 0000h]

COMPSTS is shown in Figure 17-8 and described in Table 17-6.

Return to the Summary Table.

CMPSS Comparator Status Register

Figure 17-8. COMPSTS Register

15	14	13	12	11	10	9	8
RESERVED						COMPLLATCH	COMPLSTS
R-0h						R-0h	R-0h
7	6	5	4	3	2	1	0
RESERVED						COMPHLATCH	COMPHSTS
R-0h						R-0h	R-0h

Table 17-9. COMPSTS Register Field Descriptions

Bit	Field	Type	Reset	Description
15-10	RESERVED	R	0h	Reserved
9	COMPLLATCH	R	0h	Latched value of low comparator digital filter output Reset type: SYSRSn
8	COMPLSTS	R	0h	Low comparator digital filter output Reset type: SYSRSn
7-2	RESERVED	R	0h	Reserved
1	COMPHLATCH	R	0h	Latched value of high comparator digital filter output Reset type: SYSRSn
0	COMPHSTS	R	0h	High comparator digital filter output Reset type: SYSRSn

4 COMPSTSCLR Register (Offset = 6h) [Reset = 0000h]

COMPSTSCLR is shown in Figure 17-9 and described in Table 17-7.

Return to the Summary Table.

CMPSS Comparator Status Clear Register

Figure 17-9. COMPSTSCLR Register

15	14	13	12	11	10	9	8
RESERVED					LSYNCCLREN	LLATCHCLR	RESERVED
R-0h					R/W-0h	R-0/W1S-0h	R-0h
7	6	5	4	3	2	1	0
RESERVED					HSYNCCLREN	HLATCHCLR	RESERVED
R-0h					R/W-0h	R-0/W1S-0h	R-0h

Table 17-11. COMPSTSCLR Register Field Descriptions

Bit	Field	Type	Reset	Description
15-11	RESERVED	R	0h	Reserved
10	LSYNCCLREN	R/W	0h	Low comparator latch PWMSYNCPER clear. Enable PWMSYNCPER reset of low comparator digital filter output latch COMPSTS[COMPLLATCH]. 0 PWMSYNCPER will not reset latch 1 PWMSYNCPER will reset latch Reset type: SYSRSn
9	LLATCHCLR	R-0/W1S	0h	Low comparator latch software clear. Perform software reset of low comparator digital filter output latch COMPSTS[COMPLLATCH]. Reads always return 0. 0 No effect 1 Generate a single pulse of latch reset signal for COMPSTS[COMPLLATCH] Reset type: SYSRSn
8-3	RESERVED	R	0h	Reserved
2	HSYNCCLREN	R/W	0h	High comparator latch PWMSYNCPER clear. Enable PWMSYNCPER reset of high comparator digital filter output latch COMPSTS[COMPHLATCH]. 0 PWMSYNCPER will not reset latch 1 PWMSYNCPER will reset latch Reset type: SYSRSn
1	HLATCHCLR	R-0/W1S	0h	High comparator latch software clear. Perform software reset of high comparator digital filter output latch COMPSTS[COMPHLATCH]. Reads always return 0. 0 No effect 1 Generate a single pulse of latch reset signal for COMPSTS[COMPHLATCH] Reset type: SYSRSn
0	RESERVED	R	0h	Reserved

5 COMPDACHCTL Register (Offset = 8h) [Reset = 0000h]

COMPDACHCTL is shown in Figure 17-10 and described in Table 17-8.

Return to the Summary Table.

CMPSS High DAC Control Register

Figure 17-10. COMPDACHCTL Register

15	14	13	12	11	10	9	8
RESERVED		RESERVED	RESERVED	RESERVED			
R/W-0h		R/W-0h	R/W-0h	R/W-0h			
7	6	5	4	3	2	1	0
SWLOADSEL	RESERVED	RESERVED	RAMPSOURCE				RESERVED
R/W-0h	R/W-0h	R/W-0h	R/W-0h				R/W-0h

Table 17-13. COMPDACHCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
15-14	RESERVED	R/W	0h	Reserved
13	RESERVED	R/W	0h	Reserved
12	RESERVED	R/W	0h	Reserved
11-8	RESERVED	R/W	0h	Reserved
7	SWLOADSEL	R/W	0h	Software load select. Determines whether DACxVALA is updated from DACxVALS on SYSCLK or MCPWMSYNCPER. 0 DACxVALA is updated from DACxVALS on SYSCLK 1 DACxVALA is updated from DACxVALS on MCPWMSYNCPER Reset type: SYSRSn
6	RESERVED	R/W	0h	Reserved
5	RESERVED	R/W	0h	Reserved
4-1	RAMPSOURCE	R/W	0h	MCPWMSYNCPER source select. Determines which MCPWMnSYNCPER signal is used within the COMPH Where n represents the maximum number of MCPWMSYNCPER signals available on the device: 0 MCPWM1SYNCPER 1 MCPWM2SYNCPER 2 MCPWM3SYNCPER ... n-1 MCPWMnSYNCPER Reset type: SYSRSn
0	RESERVED	R/W	0h	Reserved

6 DACHVALS Register (Offset = Ch) [Reset = 0000h]

DACHVALS is shown in Figure 17-11 and described in Table 17-9.

Return to the Summary Table.

CMPSS High DAC Value Shadow Register

Figure 17-11. DACHVALS Register

15	14	13	12	11	10	9	8
RESERVED				DACVAL			
R-0h				R/W-0h			
7	6	5	4	3	2	1	0
DACVAL							
R/W-0h							

Table 17-15. DACHVALS Register Field Descriptions

Bit	Field	Type	Reset	Description
15-12	RESERVED	R	0h	Reserved
11-0	DACVAL	R/W	0h	High DAC shadow value. When COMPDACCTL[DACSOURCE]=0, the value of DACHVALS is loaded into DACHVALA on the trigger signal selected by COMPDACCTL[SWLOADSEL]. Reset type: SYSRSn

7 DACHVALA Register (Offset = Eh) [Reset = 0000h]

DACHVALA is shown in Figure 17-12 and described in Table 17-10.

Return to the Summary Table.

CMPSS High DAC Value Active Register

Figure 17-12. DACHVALA Register

15	14	13	12	11	10	9	8
RESERVED				DACVAL			
R-0h				R-0h			
7	6	5	4	3	2	1	0
DACVAL							
R-0h							

Table 17-17. DACHVALA Register Field Descriptions

Bit	Field	Type	Reset	Description
15-12	RESERVED	R	0h	Reserved
11-0	DACVAL	R	0h	High DAC active value. Value that is actively driven by the high DAC. Reset type: SYSRSn

8 DACLVALS Register (Offset = 24h) [Reset = 0000h]

DACLVALS is shown in Figure 17-13 and described in Table 17-11.

Return to the Summary Table.

CMPSS Low DAC Value Shadow Register

Figure 17-13. DACLVALS Register

15	14	13	12	11	10	9	8
RESERVED				DACVAL			
R-0h				R/W-0h			
7	6	5	4	3	2	1	0
DACVAL							
R/W-0h							

Table 17-19. DACLVALS Register Field Descriptions

Bit	Field	Type	Reset	Description
15-12	RESERVED	R	0h	Reserved
11-0	DACVAL	R/W	0h	Low DAC shadow value. value to be loaded into DACLVALA on the trigger signal selected by COMPDACCTL[SWLOADSEL]. Reset type: SYSRSn

9 DACLVALA Register (Offset = 26h) [Reset = 0000h]

DACLVALA is shown in Figure 17-14 and described in Table 17-12.

Return to the Summary Table.

CMPSS Low DAC Value Active Register

Figure 17-14. DACLVALA Register

15	14	13	12	11	10	9	8
RESERVED				DACVAL			
R-0h				R-0h			
7	6	5	4	3	2	1	0
DACVAL							
R-0h							

Table 17-21. DACLVALA Register Field Descriptions

Bit	Field	Type	Reset	Description
15-12	RESERVED	R	0h	Reserved
11-0	DACVAL	R	0h	Low DAC active value. Value that is actively driven by the low DAC. Reset type: SYSRSn

10 CTRIPLFILCTL Register (Offset = 2Ch) [Reset = 0000h]

CTRIPLFILCTL is shown in Figure 17-15 and described in Table 17-13.

Return to the Summary Table.

CTRIPL Filter Control Register

Figure 17-15. CTRIPLFILCTL Register

15	14	13	12	11	10	9	8
FILINIT	THRESH						SAMPWIN
R-0/W1S-0h	R/W-0h						R/W-0h
7	6	5	4	3	2	1	0
SAMPWIN						FILTINSEL	
R/W-0h						R/W-0h	

Table 17-23. CTRIPLFILCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
15	FILINIT	R-0/W1S	0h	Low filter initialization. 0 No effect 1 Initialize all samples to the filter input value Reset type: SYSRSn
14-9	THRESH	R/W	0h	Low filter majority voting threshold. At least THRESH samples of the opposite state must appear within the sample window in order for the output to change state. Threshold used is THRESH+1. Reset type: SYSRSn
8-3	SAMPWIN	R/W	0h	Low filter sample window size. Number of samples to monitor is SAMPWIN+1. Reset type: SYSRSn
2-0	FILTINSEL	R/W	0h	Low filter Input Mux Select Bit 0 Selects the COMPL output as the filter input 1 Selects the external signal EXT_FILTIN_L[1] as the filter input 2 Selects the external signal EXT_FILTIN_L[2] as the filter input ... ... 7 Selects the external signal EXT_FILTIN_L[7] as the filter input Reset type: SYSRSn

11 CTRIPLFILCLKCTL Register (Offset = 2Eh) [Reset = 0000h]

CTRIPLFILCLKCTL is shown in Figure 17-16 and described in Table 17-14.

Return to the Summary Table.

CTRIPL Filter Clock Control Register

Figure 17-16. CTRIPLFILCLKCTL Register

15	14	13	12	11	10	9	8
CLKPRESCALE							
R/W-0h							
7	6	5	4	3	2	1	0
CLKPRESCALE							
R/W-0h							

Table 17-25. CTRIPLFILCLKCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
15-0	CLKPRESCALE	R/W	0h	Low filter sample clock prescale. Number of system clocks between samples is CLKPRESCALE+1. Reset type: SYSRSn

12 CTRIPHFILCTL Register (Offset = 30h) [Reset = 0000h]

CTRIPHFILCTL is shown in Figure 17-17 and described in Table 17-15.

Return to the Summary Table.

CTRIPH Filter Control Register

Figure 17-17. CTRIPHFILCTL Register

15	14	13	12	11	10	9	8
FILINIT	THRESH						SAMPWIN
R-0/W1S-0h	R/W-0h						R/W-0h
7	6	5	4	3	2	1	0
SAMPWIN						FILTINSEL	
R/W-0h						R/W-0h	

Table 17-27. CTRIPHFILCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
15	FILINIT	R-0/W1S	0h	High filter initialization. 0 No effect 1 Initialize all samples to the filter input value Reset type: SYSRSn
14-9	THRESH	R/W	0h	High filter majority voting threshold. At least THRESH samples of the opposite state must appear within the sample window in order for the output to change state. Threshold used is THRESH+1. Reset type: SYSRSn
8-3	SAMPWIN	R/W	0h	High filter sample window size. Number of samples to monitor is SAMPWIN+1. Reset type: SYSRSn
2-0	FILTINSEL	R/W	0h	High filter Input Mux Select Bit 0 Selects the COMPL output as the filter input 1 Selects the external signal EXT_FILTIN_H[1] as the filter input 2 Selects the external signal EXT_FILTIN_H[2] as the filter input ... ... 7 Selects the external signal EXT_FILTIN_H[7] as the filter input Reset type: SYSRSn

13 CTRIPHFILCLKCTL Register (Offset = 32h) [Reset = 0000h]

CTRIPHFILCLKCTL is shown in Figure 17-18 and described in Table 17-16.

Return to the Summary Table.

CTRIPH Filter Clock Control Register

Figure 17-18. CTRIPHFILCLKCTL Register

15	14	13	12	11	10	9	8
CLKPRESCALE							
R/W-0h							
7	6	5	4	3	2	1	0
CLKPRESCALE							
R/W-0h							

Table 17-29. CTRIPHFILCLKCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
15-0	CLKPRESCALE	R/W	0h	High filter sample clock prescale. Number of system clocks between samples is CLKPRESCALE+1. Reset type: SYSRSn

14 COMPLOCK Register (Offset = 34h) [Reset = 0000h]

COMPLOCK is shown in Figure 17-19 and described in Table 17-17.

Return to the Summary Table.

CMPSS Lock Register

Figure 17-19. COMPLOCK Register

15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED			RESERVED	CTRIP	DACCTL	COMPHYSCTL	COMPCTL
R-0h			R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h

Table 17-31. COMPLOCK Register Field Descriptions

Bit	Field	Type	Reset	Description
15-5	RESERVED	R	0h	Reserved
4	RESERVED	R/WSonce	0h	Reserved
3	CTRIP	R/WSonce	0h	Lock write-access to the CTRIPxFILTCTL and CTRIPxFILCLKCTL* registers. 0 CTRIPxFILCTL and CTRIPxFILCLKCTL* registers are not locked. Write 0 to this bit has no effect. 1 CTRIPxFILCTL and CTRIPxFILCLKCTL* registers are locked. Only a system reset can clear this bit. Reset type: SYSRSn
2	DACCTL	R/WSonce	0h	Lock write-access to the COMPDAC*CTL* register(s). 0 COMPDAC*CTL* register(s) not locked. Write 0 to this bit has no effect. 1 COMPDAC*CTL* register(s) locked. Only a system reset can clear this bit. Reset type: SYSRSn
1	COMPHYSCTL	R/WSonce	0h	Lock write-access to the COMPHYSCTL register. 0 COMPHYSCTL register is not locked. Write 0 to this bit has no effect. 1 COMPHYSCTL register is locked. Only a system reset can clear this bit. Reset type: SYSRSn
0	COMPCTL	R/WSonce	0h	Lock write-access to the COMPCTL register. 0 COMPCTL register is not locked. Write 0 to this bit has no effect. 1 COMPCTL register is locked. Only a system reset can clear this bit. Reset type: SYSRSn

15 COMPDACLCTL Register (Offset = 48h) [Reset = 0000h]

COMPDACLCTL is shown in Figure 17-20 and described in Table 17-18.

Return to the Summary Table.

CMPSS Low DAC Control Register

Figure 17-20. COMPDACLCTL Register

15	14	13	12	11	10	9	8
RESERVED		RESERVED	RESERVED	RESERVED			
R-0h		R/W-0h	R/W-0h	R/W-0h			
7	6	5	4	3	2	1	0
RESERVED	RESERVED	RESERVED	RAMPSOURCE				RESERVED
R-0h	R/W-0h	R-0h	R/W-0h				R/W-0h

Table 17-33. COMPDACLCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
15-14	RESERVED	R	0h	Reserved
13	RESERVED	R/W	0h	Reserved
12	RESERVED	R/W	0h	Reserved
11-8	RESERVED	R/W	0h	Reserved
7	RESERVED	R	0h	Reserved
6	RESERVED	R/W	0h	Reserved
5	RESERVED	R	0h	Reserved
4-1	RAMPSOURCE	R/W	0h	MCPWMSYNCPER source select. Determines which MCPWMnSYNCPER signal is used within the COMPL Where n represents the maximum number of MCPWMSYNCPER signals available on the device: 0 MCPWM1SYNCPER 1 MCPWM2SYNCPER 2 MCPWM3SYNCPER ... n-1 MCPWMnSYNCPER Reset type: SYSRSn
0	RESERVED	R/W	0h	Reserved

ADVANCE INFORMATION

16 CTRIPLFILCLKCTL2 Register (Offset = 6Eh) [Reset = 0000h]

CTRIPLFILCLKCTL2 is shown in Figure 17-21 and described in Table 17-19.

Return to the Summary Table.

CTRIPL Filter Clock Control Register 2

Figure 17-21. CTRIPLFILCLKCTL2 Register

15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
CLKPRESCALEU							
R/W-0h							

Table 17-35. CTRIPLFILCLKCTL2 Register Field Descriptions

Bit	Field	Type	Reset	Description
15-8	RESERVED	R	0h	Reserved
7-0	CLKPRESCALEU	R/W	0h	COMP Low filter sample clock prescale Upper Bits. The effective prescale value is (CLKPRESCALEH:CLKPRESCALE)+1 Reset type: SYSRSn

17 CTRIPHFILCLKCTL2 Register (Offset = 72h) [Reset = 0000h]

CTRIPHFILCLKCTL2 is shown in Figure 17-22 and described in Table 17-20.

Return to the Summary Table.

CTRIPH Filter Clock Control Register 2

Figure 17-22. CTRIPHFILCLKCTL2 Register

15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
CLKPRESCALEU							
R/W-0h							

Table 17-37. CTRIPHFILCLKCTL2 Register Field Descriptions

Bit	Field	Type	Reset	Description
15-8	RESERVED	R	0h	Reserved
7-0	CLKPRESCALEU	R/W	0h	COMP High filter sample clock prescale Upper Bits. The effective prescale value is (CLKPRESCALEH:CLKPRESCALE)+1 Reset type: SYSRSn

Chapter 18

Programmable Gain Amplifier (PGA)

The Programmable Gain Amplifier (PGA) is used to amplify an input voltage for the purpose of increasing the dynamic range of the downstream ADC and CMPSS modules.

18.1 Programmable Gain Amplifier (PGA) Overview

The integrated PGA helps to reduce cost and design effort for many control applications that traditionally require external, standalone amplifiers. On-chip integration makes sure that the PGA is compatible with the downstream analog-to-digital converter (ADC) and comparator subsystem (CMPSS) modules. Software selectable gain, filter settings, and different operational modes make the PGA adaptable to various performance needs.

18.1.1 Features

Features available to PGA modules are:

- Rail to rail input and output voltage within VDDA and VSSA range
- Programmable gain modes including unity gain and other values from x2 to x64
- Standalone gain mode using off-chip passive components
- Post-gain filtering using on-chip resistors
- Differential input support
- Hardware assisted chopping for offset reduction
- Support for Kelvin ground connections using PGA_INM pins
- 4-to-1 mux on positive input pins
- 4-to-1 mux on negative input pins

18.1.2 Block Diagram

Figure 18-1 shows the block diagram of the PGA. The active component in the PGA is an embedded operational amplifier (op-amp) that is configurable as a non-inverting or inverting amplifier with internal feedback resistors. These internal feedback resistor values are paired to produce software selectable voltage gains.

Six PGA signals are available at the device pins:

- PGA_INP0 through PGA_INP8 are multiplexed to get the positive input to the PGA op-amp.
- PGA_INM0 through PGA_INM3 are multiplexed to get the negative input to the PGA op-amp.
- PGA_OUT supports op-amp output filtering with RC components. The filtered signal is available for sampling and monitoring by on-chip ADC and CMPSS modules.

Note

PGA_OUT_INT is an internal signal at the op-amp output, which is available for sampling and monitoring by the internal ADC and CMPSS modules.

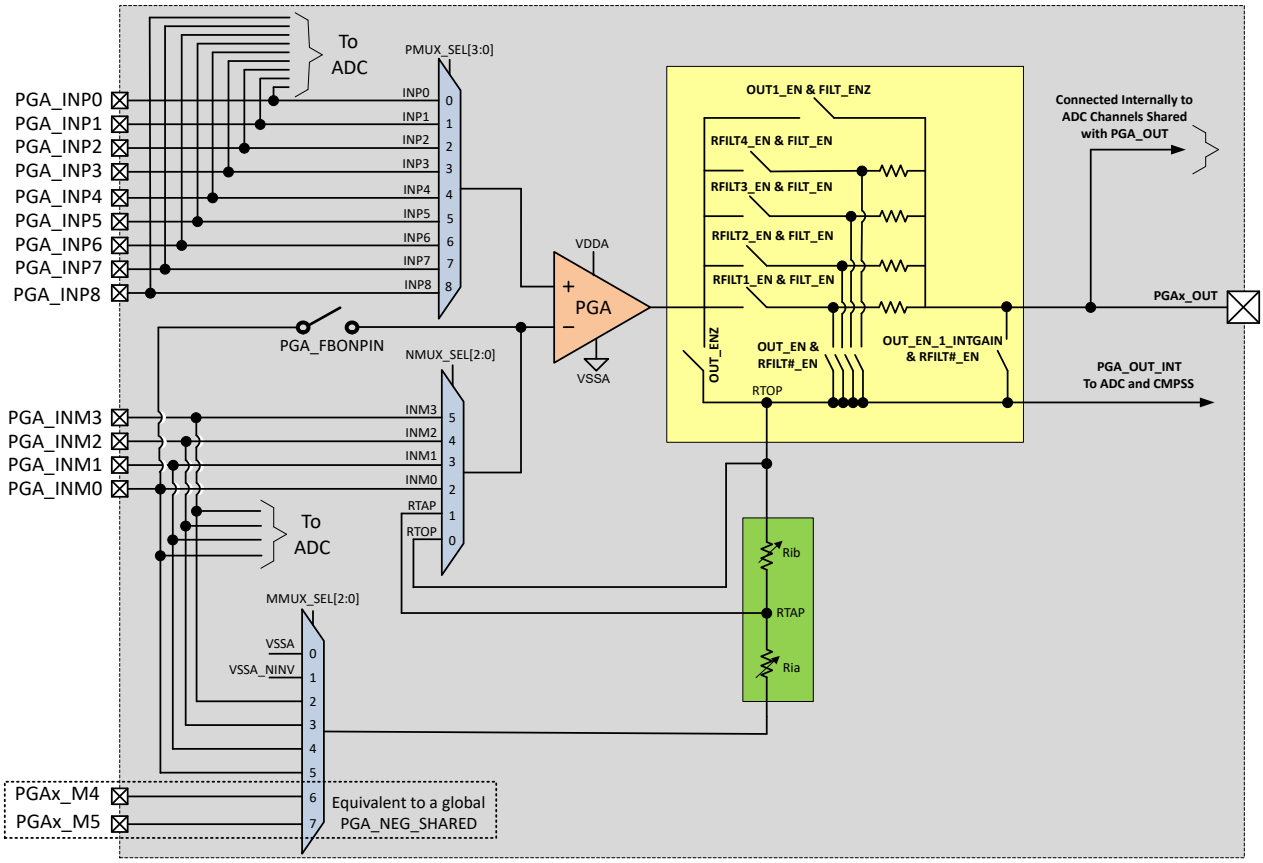


Figure 18-1. PGA Block Diagram

18.1.2.1 PGA Mux Selection Options

Table 18-1. PGA0 Mux Selection Options

PGA0_P	PGA0_N	PGA0_M
PMUX_SEL[3:0]	**NMUX_SEL[1:0]**	**MMUX_SEL[2:0]**
A0_IN9 / A1_IN9 / PGA2_INP1 / PGA0_INP0 / PGA1_INP0	RTAP	VSSA
A0_IN2 / A1_IN2 / CMP0_HP1_LP1 / PGA0_INP1 / PGA1_INM0 / PGA2_INP2	RTOP	VSSA_NINV
A0_IN4 / A2_IN21 / CMP1_HP0_LP0 / PGA0_INM0 / PGA0_INP2 / PGA2_INM0	A0_IN4 / A2_IN21 / CMP1_HP0_LP0 / PGA0_INM0 / PGA0_INP2 / PGA2_INM0	A0_IN4 / A2_IN21 / CMP1_HP0_LP0 / PGA0_INM0 / PGA0_INP2 / PGA2_INM0
A1_IN4 / A2_IN22 / CMP1_HP1_LP1 / PGA0_INP3 / PGA1_INP1	A1_IN11 / A2_IN24 / PGA0_INM1 / PGA1_INM2	A1_IN11 / A2_IN24 / PGA0_INM1 / PGA1_INM2
A0_IN14 / A1_IN14 / CMP1_HP2_LP2 / PGA0_INP4 / PGA2_INP0	A2_IN17 / CMP0_LN1 / PGA2_INM2 / PGA0_INM2	A2_IN17 / CMP0_LN1 / PGA2_INM2 / PGA0_INM2
A2_IN5 / CMP0_HP2_LP2 / PGA2_INP4 / PGA0_INP5 / PGA1_INP4	A0_IN21 / A1_IN21 / PGA0_INM3	A0_IN21 / A1_IN21 / PGA0_INM3
A0_IN27 / A1_IN27 / A2_IN8 / CMP1_HP3_LP3 / PGA0_INP6 / PGA1_INP8 / PGA2_INP8		A1_IN13 / A2_IN19 / CMP2_DACL1 / CMP1_HN1 / PGA1_INM1 (PGAx_INM4)
A0_IN11 / A2_IN30 / CMP2_LN1 / PGA1_INP3 / PGA0_INP7		A1_IN15 / A2_IN28 / CMP1_LN1 / PGA2_INM3 (PGAx_INM5)
A0_IN28 / A1_IN28 / A2_IN9 / CMP0_HP3_LP3 / PGA2_INP6 / PGA0_INP8		

ADVANCE INFORMATION

Table 18-1. PGA0 Mux Selection Options (continued)

PGA0_P	PGA0_N	PGA0_M
PMUX_SEL[3:0]	**NMUX_SEL[1:0]**	**MMUX_SEL[2:0]**
CMP2_DACL		
ADCINCAL0		

Table 18-2. PGA1 Mux Selection Options

PGA1_P	PGA1_N	PGA1_M
PMUX_SEL[3:0]	**NMUX_SEL[1:0]**	**MMUX_SEL[2:0]**
A0_IN9 / A1_IN9 / PGA2_INP1 / PGA0_INP0 / PGA1_INP0	RTAP	VSSA
A1_IN4 / A2_IN22 / CMP1_HP1_LP1 / PGA0_INP3 / PGA1_INP1	RTOP	VSSA_NINV
A0_IN15 / A2_IN12 / CMP3_HP1_LP1 / PGA1_INP2 / PGA2_INP3	A0_IN2 / A1_IN2 / CMP0_HP1_LP1 / PGA0_INP1 / PGA1_INM0 / PGA2_INP2	A0_IN2 / A1_IN2 / CMP0_HP1_LP1 / PGA0_INP1 / PGA1_INM0 / PGA2_INP2
A0_IN11 / A2_IN30 / CMP2_LN1 / PGA1_INP3 / PGA0_INP7	A1_IN13 / A2_IN19 / CMP2_DACL1 / CMP1_HN1 / PGA1_INM1	A1_IN13 / A2_IN19 / CMP2_DACL1 / CMP1_HN1 / PGA1_INM1
A2_IN5 / CMP0_HP2_LP2 / PGA2_INP4 / PGA0_INP5 / PGA1_INP4	A1_IN11 / A2_IN24 / PGA0_INM1 / PGA1_INM2	A1_IN11 / A2_IN24 / PGA0_INM1 / PGA1_INM2
A0_IN5 / A2_IN27 / CMP2_HP3_LP3 / PGA1_INP5 / PGA2_INP5	A0_IN12 / A2_IN1 / CMP0_HP0_LP0 / PGA1_INM3 / PGA2_OUT	A0_IN12 / A2_IN1 / CMP0_HP0_LP0 / PGA1_INM3 / PGA2_OUT
A0_IN22 / A1_IN22 / PGA1_INP6 / PGA2_INP7		A2_IN17 / CMP0_LN1 / PGA2_INM2 / PGA0_INM2
A0_IN30 / A1_IN30 / A2_IN11 / CMP2_HP2_LP2 / PGA1INP7		A1_IN15 / A2_IN28 / CMP1_LN1 / PGA2_INM3
A0_IN27 / A1_IN27 / A2_IN8 / CMP1_HP3_LP3 / PGA0_INP6 / PGA1_INP8 / PGA2_INP8		
CMP2_DACL		
ADCINCAL0		

Table 18-3. PGA2 Mux Selection Options

PGA2_P	PGA2_N	PGA2_M
PMUX_SEL[3:0]	**NMUX_SEL[1:0]**	**MMUX_SEL[2:0]**
A0_IN14 / A1_IN14 / CMP1_HP2_LP2 / PGA0_INP4 / PGA2_INP0	RTAP	VSSA
A0_IN9 / A1_IN9 / PGA2_INP1 / PGA0_INP0 / PGA1_INP0	RTOP	VSSA_NINV
A0_IN2 / A1_IN2 / CMP0_HP1_LP1 / PGA0_INP1 / PGA1_INM0 / PGA2_INP2	A0_IN4 / A2_IN21 / CMP1_HP0_LP0 / PGA0_INM0 / PGA0_INP2 / PGA2_INM0	A0_IN4 / A2_IN21 / CMP1_HP0_LP0 / PGA0_INM0 / PGA0_INP2 / PGA2_INM0
A0_IN15 / A2_IN12 / CMP3_HP1_LP1 / PGA1_INP2 / PGA2_INP3	A1_IN12 / CMP3_HN0_LN0 / PGA2_INM1	A1_IN12 / CMP3_HN0_LN0 / PGA2_INM1
A2_IN5 / CMP0_HP2_LP2 / PGA2_INP4 / PGA0_INP5 / PGA1_INP4	A2_IN17 / CMP0_LN1 / PGA2_INM2 / PGA0_INM2	A2_IN17 / CMP0_LN1 / PGA2_INM2 / PGA0_INM2
A0_IN5 / A2_IN27 / CMP2_HP3_LP3 / PGA1_INP5 / PGA2_INP5	A1_IN15 / A2_IN28 / CMP1_LN1 / PGA2_INM3	A1_IN15 / A2_IN28 / CMP1_LN1 / PGA2_INM3
A0_IN28 / A1_IN28 / A2_IN9 / CMP0_HP3_LP3 / PGA2_INP6 / PGA0_INP8		A1_IN13 / A2_IN19 / CMP2_DACL1 / CMP1_HN1 / PGA1_INM1 (PGAx_INM4)
A0_IN22 / A1_IN22 / PGA1_INP6 / PGA2_INP7		A1_IN11 / A2_IN24 / PGA0_INM1 / PGA1_INM2 (PGAx_INM5)
A0_IN27 / A1_IN27 / A2_IN8 / CMP1_HP3_LP3 / PGA0_INP6 / PGA1_INP8 / PGA2_INP8		
CMP2_DACL		

ADVANCE INFORMATION

Table 18-3. PGA2 Mux Selection Options (continued)

PGA2_P	PGA2_N	PGA2_M
PMUX_SEL[3:0]	NMUX_SEL[1:0]	MMUX_SEL[2:0]
ADCINCAL0		

18.2 Linear Output Range

The absolute output range of the PGA is bounded by the analog VDDA and VSSA supplies – the PGA cannot produce output voltages greater than VDDA or less than VSSA.

Although the PGA can produce full-scale output across the absolute voltage range of VSSA to VDDA, the amplifier output is only linear within a subset of the absolute range. This reduced range is referred to as the linear output range.

The PGA performance specifications in the device data sheet only apply to the linear output range. For best performance, the input signal can be conditioned in such a way that the PGA stays within the linear output range during normal system operation.

Note

The voltage input range required to operate the PGA in the linear output range is unique for each gain mode. See the device data sheet for the linear output range.

18.3 Gain Values

Gain values of PGA are software-selectable using the PGACTL[GAIN] register field. The gain of the PGA in the subtractor and non-inverting modes is determined by a preset ratio between resistors R_{ia} and R_{ib}.

The target values for the gain resistors for subtractor and non-inverting modes are shown in Table 18-4.

Table 18-4. Different Gain Values and Corresponding Resistor Values

PGACTL[GAIN]	Non-inverting Gain	Inverting Gain	R_{ia}	R_{ib}
000	1	NA	256K	0(1)
001	2	-1	16K	16K
010	4	-3	8K	24K
011	8	-7	8K	56K
100	16	-15	8K	120K
101	32	-31	8K	248K
110	64	-63	4K	252K

(1) These values are nominal; tolerance is specified in electrical spec table given in the device data sheet.

Note

Changing the gain mode during normal operation is allowed, but a minimum configuration settling time can be observed when doing so. See the device data sheet for the gain switch settling time.

ADVANCE INFORMATION

18.4 Modes of Operation

PGA can support four different operational modes based on the values in MUXSEL register, which are given in Table 18-5:

1. Buffer mode: PGA works in unity gain mode.
2. Standalone mode: PGA operates as a conventional op-amp.
3. Non-inverting mode: PGA works as a non-inverting op-amp.
4. Subtractor mode: PGA output voltage equals to the subtraction of the two inputs.

Table 18-5. Modes of Operation

Mode	MUXSEL[PMUXSEL]	MUXSEL[NMUXSEL]	MUXSEL[MMUXSEL]
Buffer mode	0	0	0
Standalone mode	0	2	0
Non-inverting mode	0	1	1
Subtractor mode	0	1	2 or 3

18.4.1 Buffer Mode

In this mode, PGA provides a high input impedance, low output impedance, and a voltage gain of approximately one. The block diagram of PGA in this mode is shown in Figure 18-2. This mode is also known as voltage follower, or unity gain mode. Even though this mode does not provide any voltage amplification, this mode is extremely useful because this mode prevents one stage's input impedance from loading the prior stage's output impedance. Not only does buffer mode isolate the load from the signal source, but also offers impedance matching between different parts of a circuit. The buffer op-amp also has a low output impedance that can drive a load without being affected by the load's impedance. Moreover, the low output impedance helps to maintain the signal voltage across the load, preventing voltage drops that can affect the performance of the circuit.

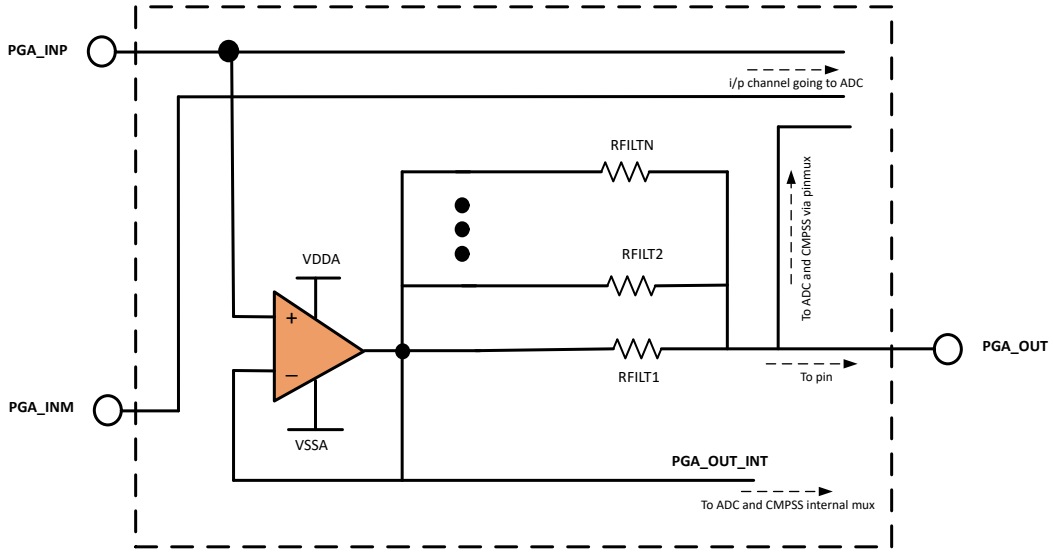


Figure 18-2. Buffer Mode

ADVANCE INFORMATION

18.4.2 Standalone Mode

In this mode, a standalone operational amplifier is available that can support a number of analog applications using a minimum number of external components. Figure 18-3 Shows the block diagram of PGA in this mode. In this mode, the output voltage is determined by the external resistors connected to the inverting and non-inverting pins, as well as the feedback circuitry connected to the output.

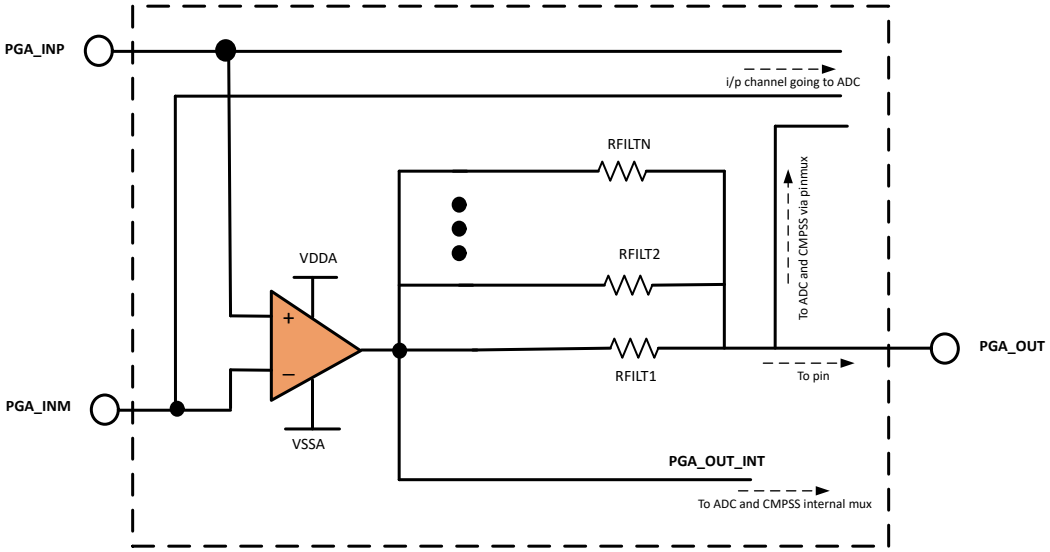


Figure 18-3. Standalone Mode

ADVANCE INFORMATION

18.4.3 Non-inverting Mode

In non-inverting mode, the input voltage signal is applied directly to the non-inverting pin PGA_INP and generates an amplified output signal that is proportional and in phase with the input signal. The block diagram of PGA in this mode is illustrated in Figure 18-4. Feedback control of the non-inverting operational amplifier is achieved by applying a small part of the output voltage signal back to the inverting input terminal by way of a voltage divider network R_{ia} – R_{ib}. The voltage gain in this mode can be calculated using Equation 21:

$$V_{PGA\_OUT} = \left(1 + \frac{R_{ib}}{R_{ia}}\right) V_{PGA\_INP} \quad (21)$$

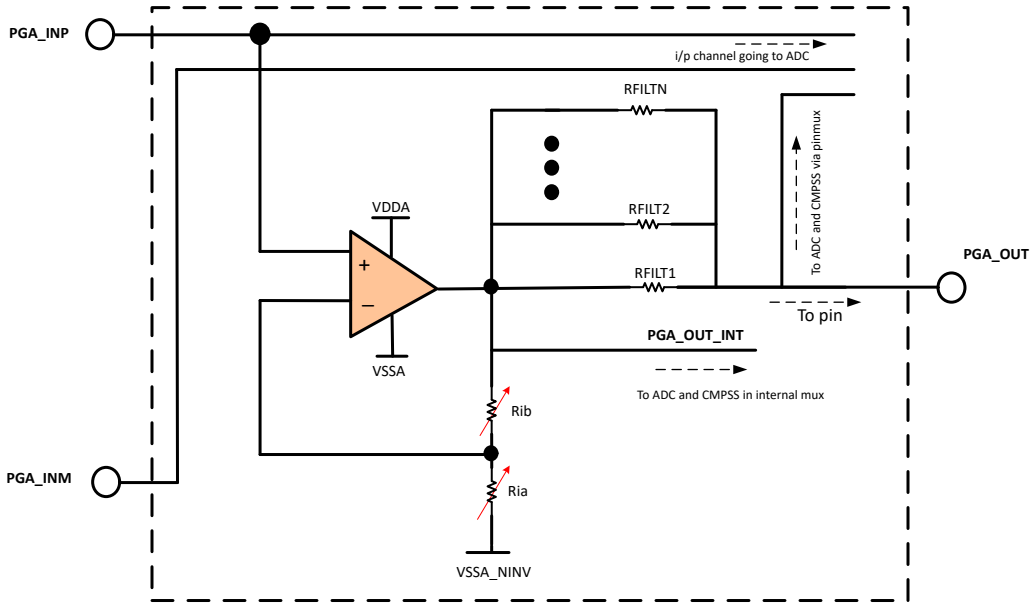


Figure 18-4. Non-inverting Mode

ADVANCE INFORMATION

18.4.4 Subtractor Mode

In subtractor mode, the subtraction of the two input voltages is possible. Superposition is used to calculate the output voltage resulting from each input voltage, and then the two output voltages are subtracted to arrive at the final output voltage. The block diagram of PGA in this mode is demonstrated in Figure 18-5. The voltage gain can be derived as:

$$V_{PGA\_OUT} = \left(1 + \frac{R_{ib}}{R_{ia}}\right)V_{PGA\_INP} - \left(\frac{R_{ib}}{R_{ia}}\right)V_{PGA\_INM} \quad (22)$$

The differential signal is multiplied by the stage gain, so the subtractor mode name is a good choice for the circuit. If a voltage divider circuit with similar values of R_{ia}-R_{ib} is added to the non-inverting pin, this mode only amplifies the differential portion of the input signal; therefore, this mode rejects the common-mode portion of the input signal.

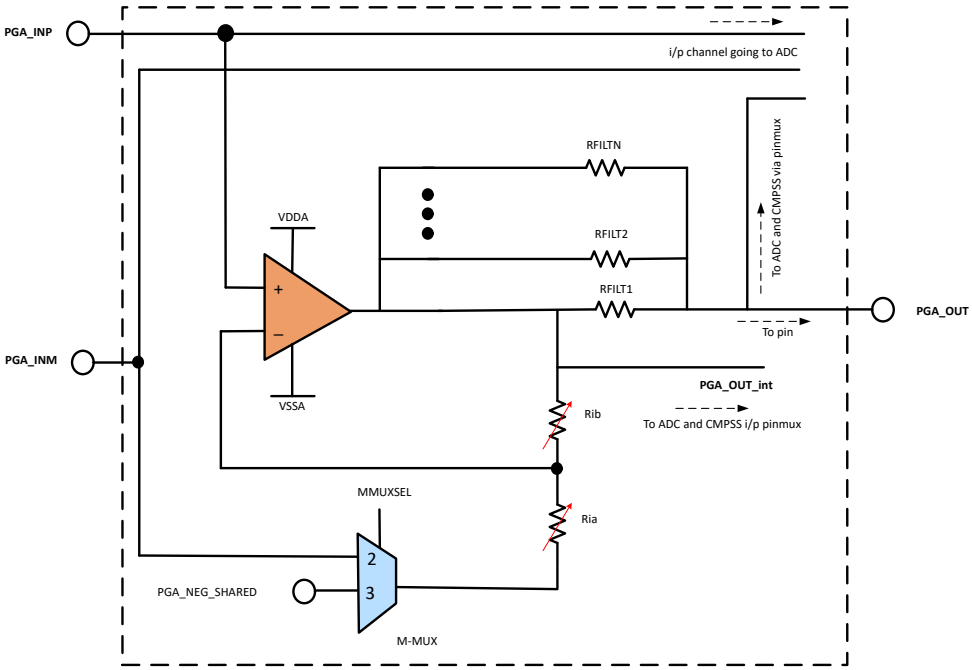


Figure 18-5. Subtractor Mode

ADVANCE INFORMATION

18.5 External Filtering

To remove any unwanted high-frequency noise, two types of low-pass filter can be implemented using internal resistors of PGA module:

1. Low-pass filter using internal filter resistors R_{FILT} and external capacitor
2. Low-pass filter using internal gain resistors R_{ib} and external capacitor

18.5.1 Low-Pass Filter Using Internal Filter Resistor and External Capacitor

The PGA output can be routed to a pin through an embedded series resistor for the purpose of low-pass filtering the amplified signal. The filter resistance is software selectable using the PGACTL[FILT_RES_SEL] register field. The default selection of PGACTL[FILT_RES_SEL] = 0 disables the filter path.

The cutoff frequency can be estimated using the standard low-pass RC given by:

$$f_{cutoff} = \frac{1}{2\pi R_{FILT} C_{FILTER}} \quad (23)$$

Each gain mode requires a minimum amount of series resistance when filtering is enabled. The values are shown in Table 18-6. Also, the external capacitor value C_{FILTER} influences the ADC sampling performance. See Section 18.10.2.2 for more information.

Table 18-6. Minimum Filter Resistance

PGACTL[GAIN]	Minimum R_{FILT} Required PGACTL[FILT_RES_SEL]
0	50Ω
1	50Ω
2	50Ω
3	100Ω
4	100Ω
5	200Ω
6	400Ω

ADVANCE INFORMATION

18.5.2 Single Pole Low-Pass Filter Using Internal Gain Resistor and External Capacitor

Some applications cannot tolerate having an RC network at the output of the amplifier. Amplifier output current flowing through the filter resistor creates a voltage offset that introduces output error. In this case, one can opt to filter the noise spikes by placing a feedback capacitor across the feedback loop. In fact, a simple single-pole low pass filter can be implemented by placing an external capacitance between PGA_INM and PGA_OUT in internal gain mode, which is shown in Figure 18-6.

This form of very simple filter is normally used in instances where a small amount of roll off is required. PGA feedback node RTAP can be brought on pin PGA_INM for external filtering by selecting PMUXSEL = 00, NMUXSEL = 01, and MMUXSEL = 01. Also, to enable this mode, PGA_FB_ON_PIN input/register needs to be set to 1.

The cutoff frequency for this type of filter can be calculated very easily by working out the frequency at which the reactance of the capacitor equals the resistance of the resistor R_{ib}, which is defined by Equation 24:

$$f_{cutoff} = \frac{1}{2\pi R_{ib} C_{Ext}} \quad (24)$$

The gain for this op amp circuit can be calculated in the normal way ignoring the effect of the capacitor:

$$Gain = \left(1 + \frac{R_{ib}}{R_{ia}}\right) \quad (25)$$

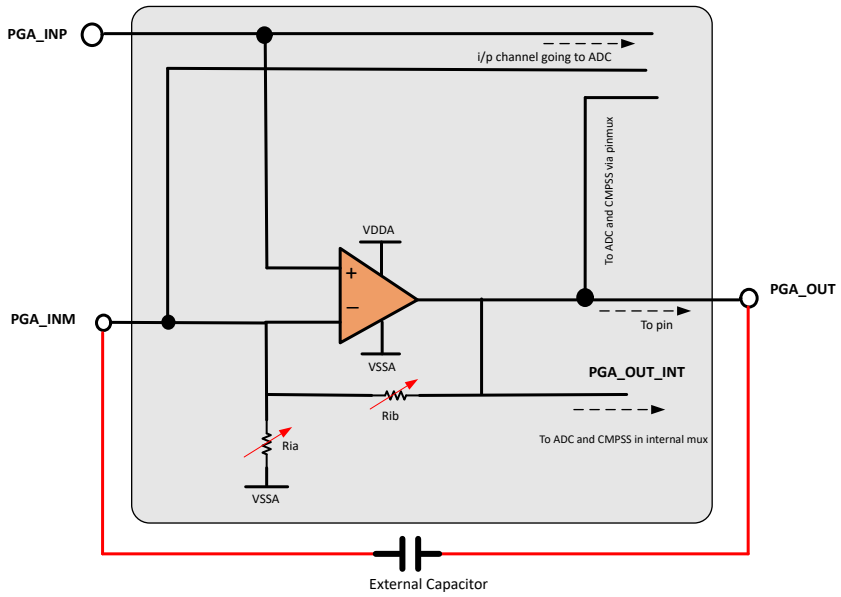


Figure 18-6. Low-Pass Filter Using External Capacitor

ADVANCE INFORMATION

18.6 Error Calibration

Op-Amp error calibration is the process of adjusting the Op-Amp parameters to minimize or eliminate errors. Calibration can be done using different methods, depending on the type of error and the Op-Amp configuration. To reduce inherent offset, factory-generated values are written to the trim registers by calling the Device_cal() function that is located in TI reserved OTP.

18.6.1 Offset Error

This error occurs due to mismatches in the Op-Amp input terminals. The offset error appears as a constant DC offset across the PGA output range, see Figure 18-7. The built-in Device_cal() function reduces the offset error so that the error falls within the data sheet specifications.

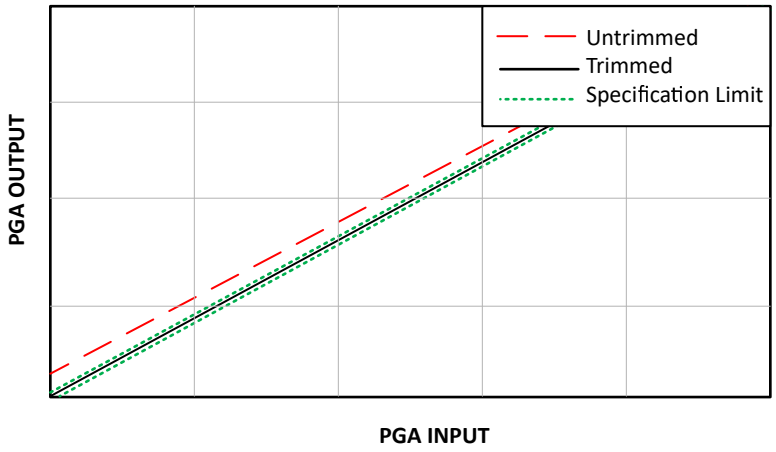


Figure 18-7. PGA Offset Trim

18.6.2 Gain Error

This error occurs due to mismatches in the Op-Amp gain. After the offset error has been removed, the remaining error, the gain error appears as a scaled error that increases in magnitude with increasing PGA output voltage, see Figure 18-8. The hardware trim targets the gain performance so that the hardware trim falls within the data sheet specifications.

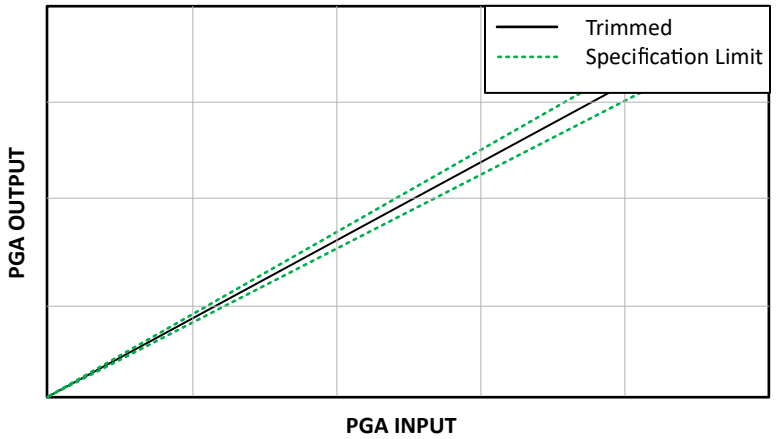


Figure 18-8. PGA Gain Error

ADVANCE INFORMATION

18.7 Chopping Feature

The PGA supports chopping feature, which is a continuous-time modulation technique in which the signal and offset are modulated to different frequencies. The motivation behind adding this feature is primarily to reduce input referred offset to a very small number (in micro volt order). Not only chopping is dynamic technique that continuously reduces offset, but also this technique removes low frequency 1/f noise as well as offset drift over temperature or time.

The chopping feature is demonstrated in Figure 18-9. In broad terms, an input voltage first passes through a chopper driven by a clock at frequency f_{ch}; so, the voltage is converted to a square-wave voltage. Then, the modulated signal along with the offset are amplified by the fully differential amplifier. The second chopper then demodulates the amplified input signal back to DC, while the offset is shifted to the frequency of the clock f_{ch} harmonics. The DC offset can be filtered out by a low-pass filter.

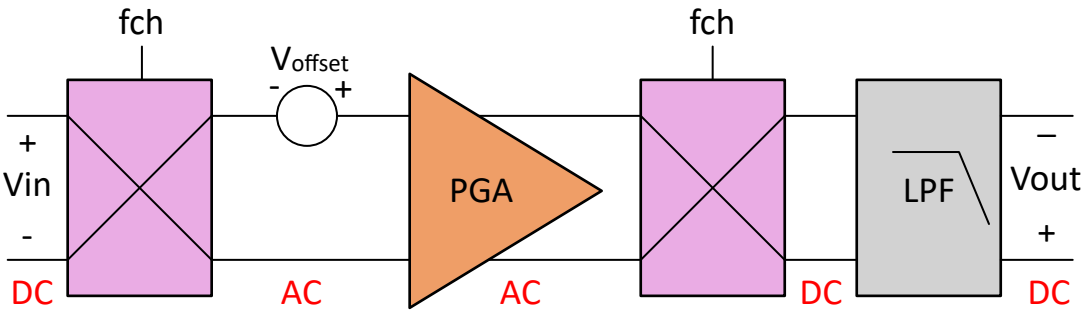


Figure 18-9. General Chopping Technique

This device supports ADC-assisted method. In this method, the chopping control signal comes from the on-chip ADC module. Moreover, Start of Conversion (SOC) in the ADC module can be used to implement chopping. In this method, PGA output is sampled by the on-chip ADC module. Note that the ADC-assisted method relies on ADC module for filtering the output instead of using external low-pass filter. To remove the offset, an average of the output signal can be taken in ADC module. This functionality is provided in post-processing block in ADC module. (See Chapter 16). The ADC-assisted chopping stands out from purely analog chopping methods due to the ability to offer higher accuracy and precision.

Figure 18-10 shows the general diagram of ADC-assisted chopping mode, in which a logic block is used to generate the necessary signal to handle chopping. To configure the ADC-assisted chopping, write to the PGA_CHOP field in the MUXSEL register. To configure control signal for ADC-assisted chopping, write to the CHOP_EXTCTRL field in the PGACTL register.

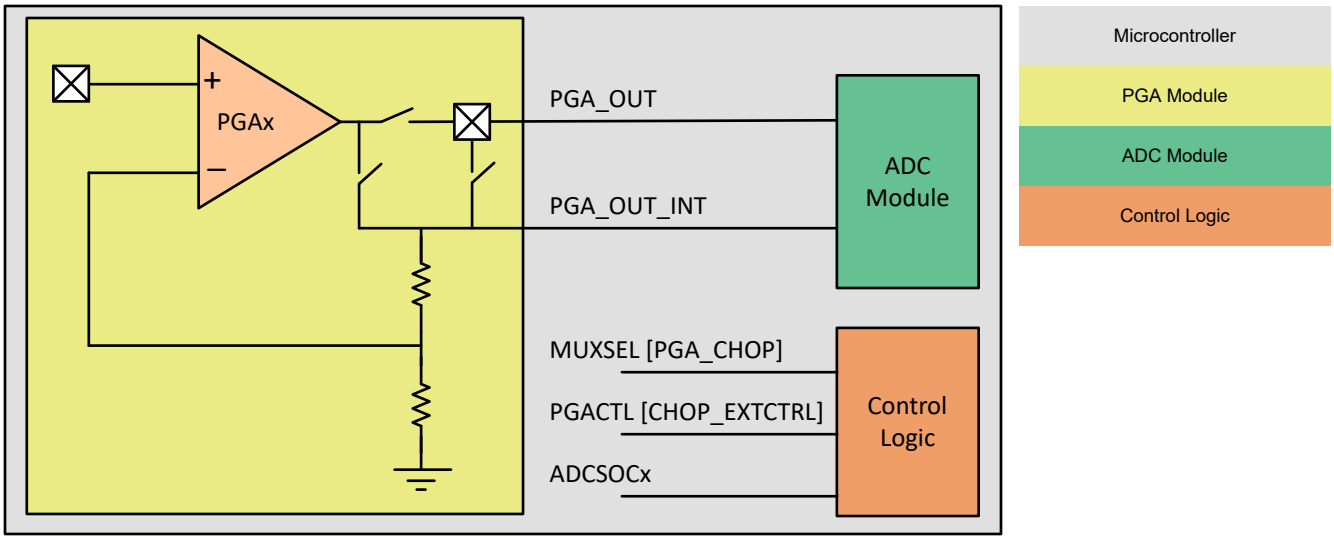


Figure 18-10. ADC-Assisted Chopping Block Diagram

For the PGA, the ADC assisted chopping feature is supported when the ADC is connected to the PGA_OUT_INT of the PGA. shows the connectivity between the PGA and ADC modules.

Table 18-7. PGA and ADC Connection

PGA Connectivity to ADC	ADC
PGA0_OUT_INT	

18.8 Enabling and Disabling the PGA Clock

If the clock to the PGA is disabled while the PGA is outputting a voltage, the output voltage remains unaffected but the PGA registers are no longer updated with register writes. Enabling the clock resumes register writes.

18.9 Lock Register

The PGALOCK register provides the ability to block writes to certain PGA configuration registers. Locking register writes can prevent spurious or malicious code from modifying critical register settings. Once a register has been locked using the PGALOCK register, only a module-level or device-level reset can restore write functionality.

18.10 Analog Front-End Integration

The PGAs operate together with the other embedded analog modules (ADC, CMPSS) creating an analog front-end system.

18.10.1 Buffered DAC

Buffered DACs are commonly used in a variety of applications such as dc-coupled applications,in which the drive amplifier must provide the required gain and offset voltage, to match the signal to the input voltage range of the ADC. As a best practice, the PGA input signal is conditioned so that the PGA output is centered within the linear range. The input signal requires some combination of offset and attenuation to achieve this goal. For example, an external resistor divider can attenuate the input signal while the embedded buffered DAC can provide a positive voltage offset, see Figure 18-11.

Note

While this device does not have a dedicated Buffered DAC module, the same functionality can be implemented with CMPSS DACL Output signal.

ADVANCE INFORMATION

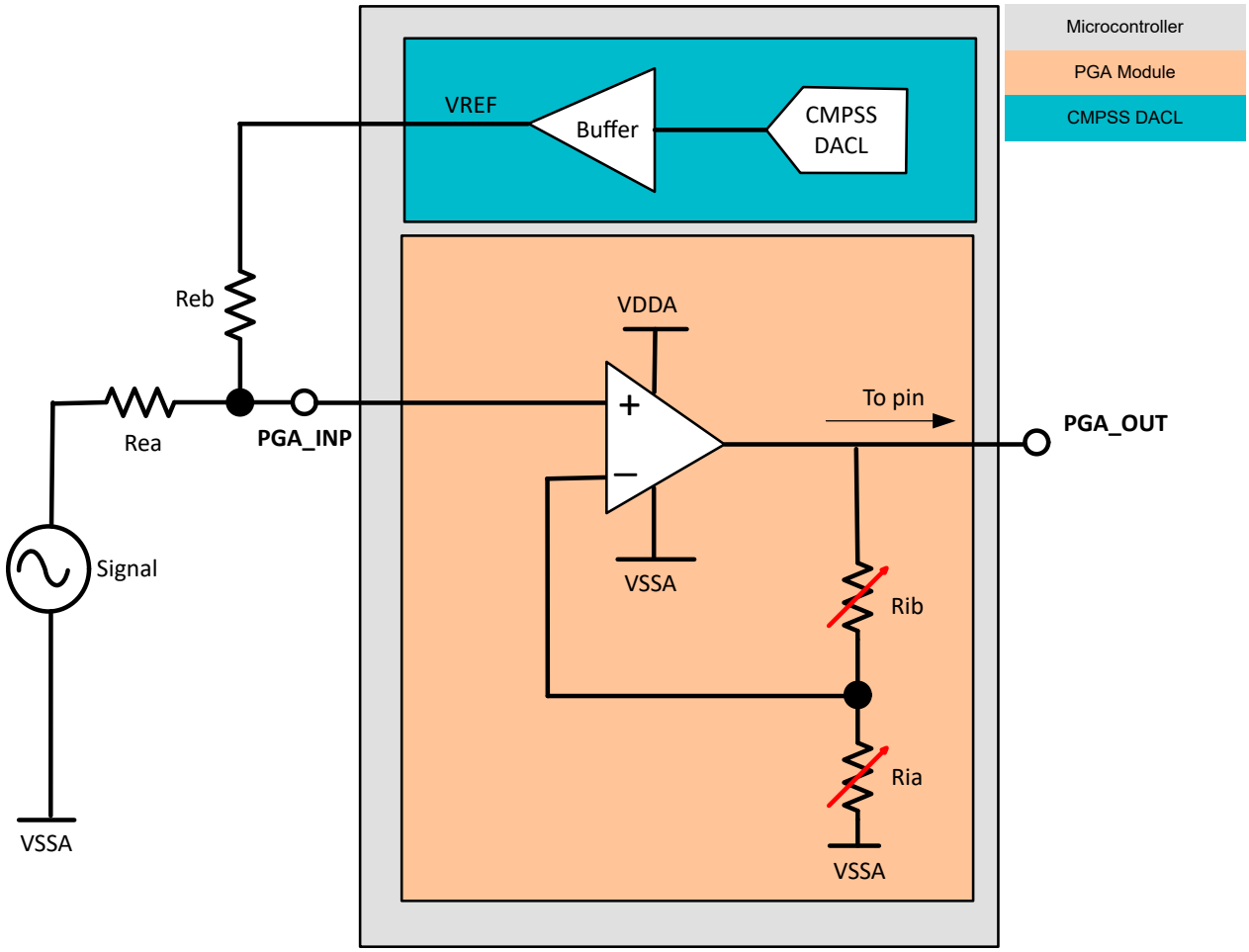


Figure 18-11. Level Shifting Using Internal DAC

With the topology shown in Figure 18-11, the voltage seen at the PGA_INP pin can be calculated as:

$$V_{PGA\_OUT} = \frac{R_{ib} + R_{ia}}{R_{ia}}\left(\frac{R_{eb}}{R_{ea} + R_{eb}}V_{signal} + \frac{R_{ea}}{R_{ea} + R_{eb}}V_{REF}\right) \quad (26)$$

Supplying a V_{DAC_OUT} of 1.65V transforms a bipolar V_{SIGNAL} range of -0.5V to +0.5V into a V_{PGA_INP} range of 0.22V to 0.88V, which is an excellent choice for the 3× gain mode.

18.10.2 Analog-to-Digital Converter (ADC)

In the simplest application, the PGA amplifies small input signals to increase the ADC dynamic range. The PGA also provides the additional benefit of buffering input signals from the ADC sample and hold capacitor, which further reduces sampling error.

Both the filtered and unfiltered PGA output paths are available for ADC sampling. Minimum ADC acquisition windows are recommended when sampling the PGA paths.

18.10.2.1 Unfiltered Acquisition Window

The device data sheet provides a minimum estimated ADC acquisition window for sampling the PGA_OUT signal with one ADC. This estimated value can provide sampling accuracy close to the specified performance parameters of the ADC. A longer acquisition window can be used for better performance.

ADVANCE INFORMATION

18.10.2.2 Filtered Acquisition Window

The minimum ADC acquisition window for sampling the PGA_OF filtered signal with one ADC varies based on the values of R_{FILTER} and C_{FILTER}. To make sure of good performance, choose a C_{FILTER} capacitor that is large enough to satisfy most of the ADC sample and hold capacitor (Ch) charge requirements. The C_{FILTER} value can be sized based on the acceptable amount of ADC sampling error (LSB_{Err}):

$$C_{FILTER} = \frac{C_h \times 4096}{LSB_{Err}} \quad (27)$$

18.10.3 Comparator Subsystem (CMPSS)

The PGA output can be monitored by a CMPSS module for trips above or below a reference voltage. Up to two independent reference thresholds per CMPSS can be used for trip detection.

Both the filtered and unfiltered PGA output paths are available for CMPSS trip monitoring.

18.10.4 PGA_NEG_SHARED Feature

This feature allows sharing PGA_INM pin across three available PGA modules, which helps to save two pins for applications where multiple PGAs are required, but the PGA modules all can have a common negative sense point. To enable this feature, MMUXSEL and NMUXSEL must be set to 3 and 1, respectively. As you can see in Figure 18-12, PGA_NEG_SHARED is shared among three PGA modules internally so that the user can employ there PGA modules with only one inverting pin. In three-phase current measurement, three modules of PGA can be employed to monitor the shunt current by using only four pins. The schematic of three-phase current sensing is shown in Figure 18-13.

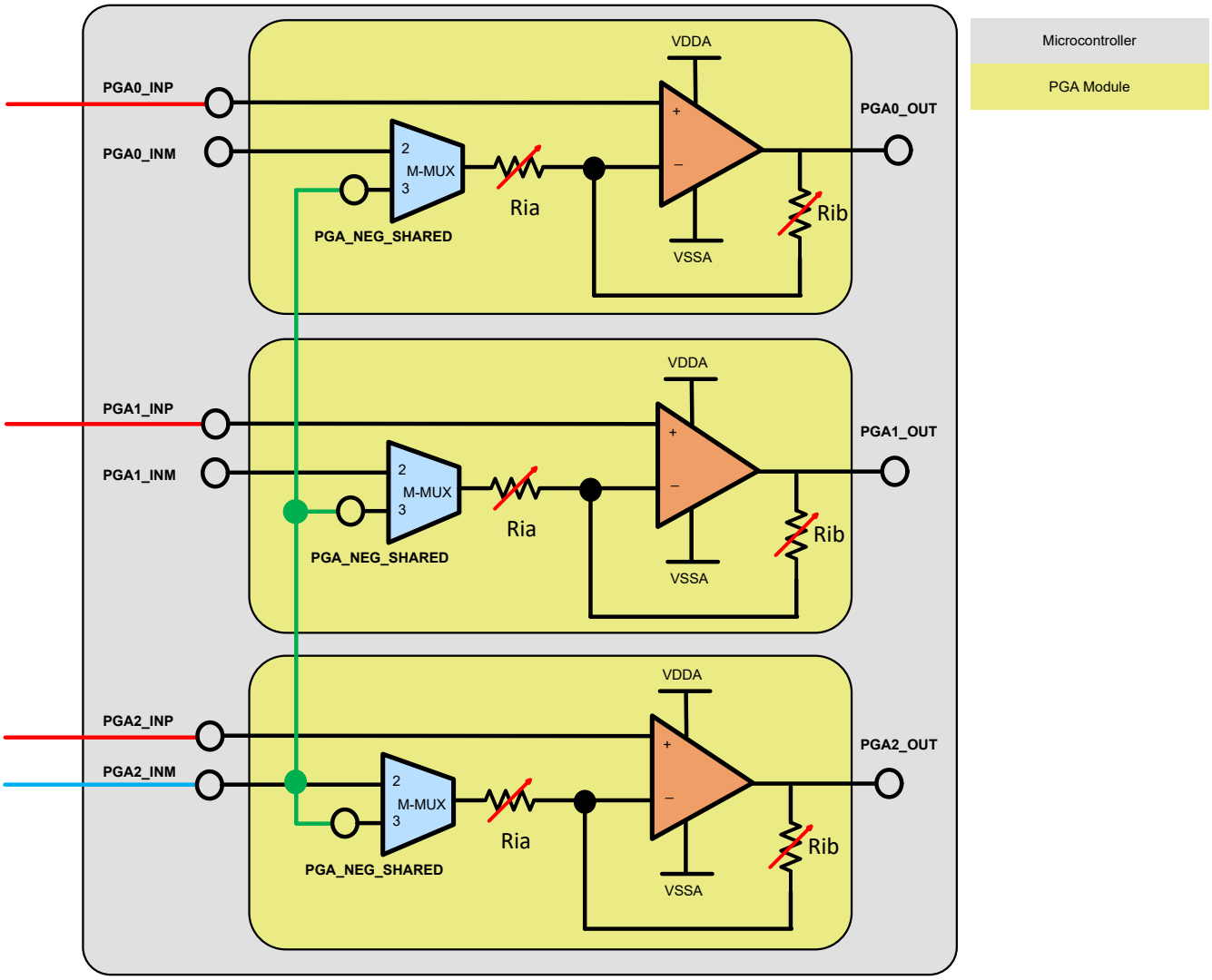


Figure 18-12. Sharing PGA_NEG Pin Between PGA Modules

ADVANCE INFORMATION

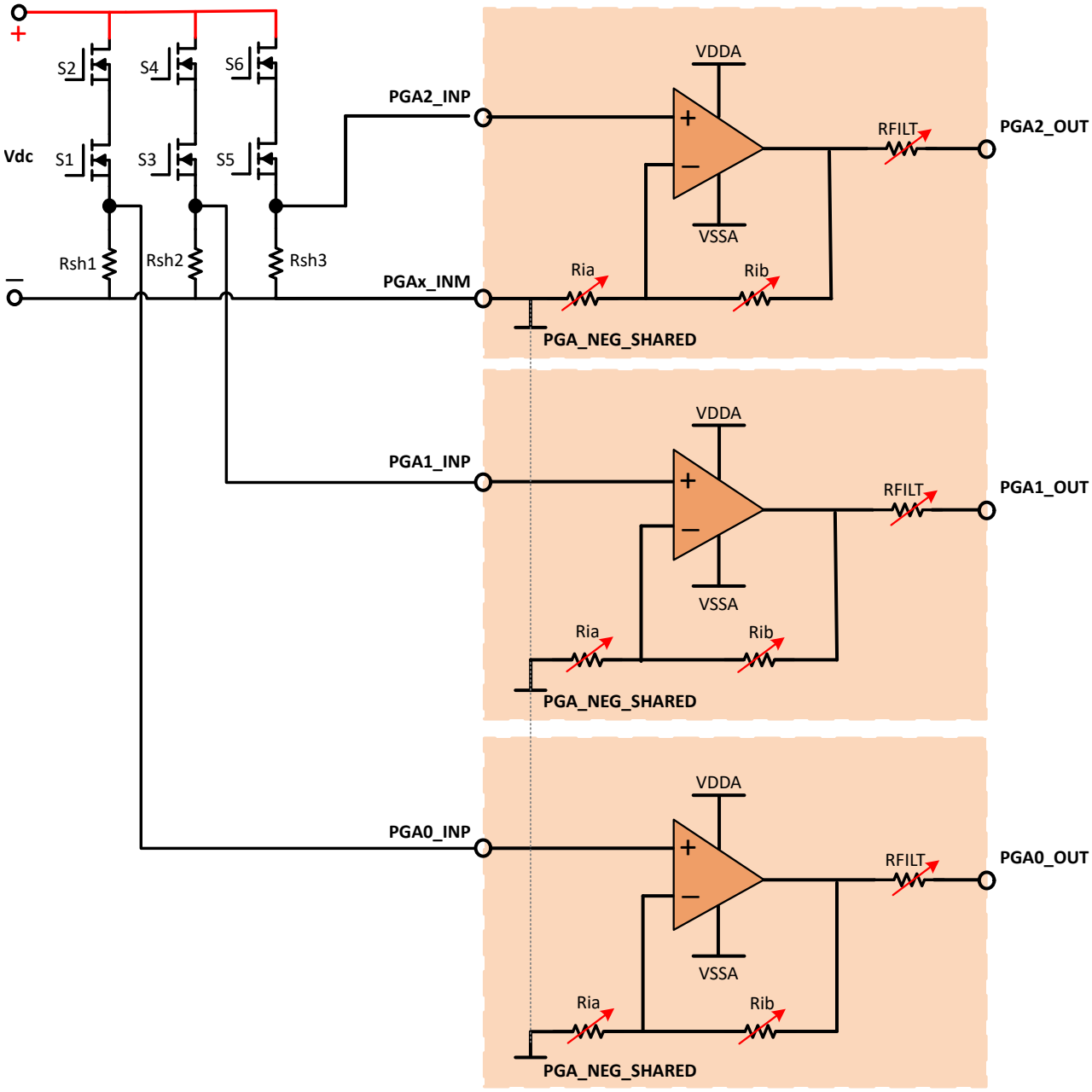


Figure 18-13. Three-phase Current Sensing Using PGA_NEG_SHARED Feature

ADVANCE INFORMATION

18.10.5 Alternate Functions

Each PGA has up to three device terminals for operation: non-inverting pin PGA_INP, inverting pin PGA_INM, and output pin PGA_OUT, see Figure 18-14. Alternate paths at the device level allow the input and output filter terminals to take on alternate functions. This can be especially valuable, if the respective PGA resource is not used for an application. See *Analog Subsystem* for more information.

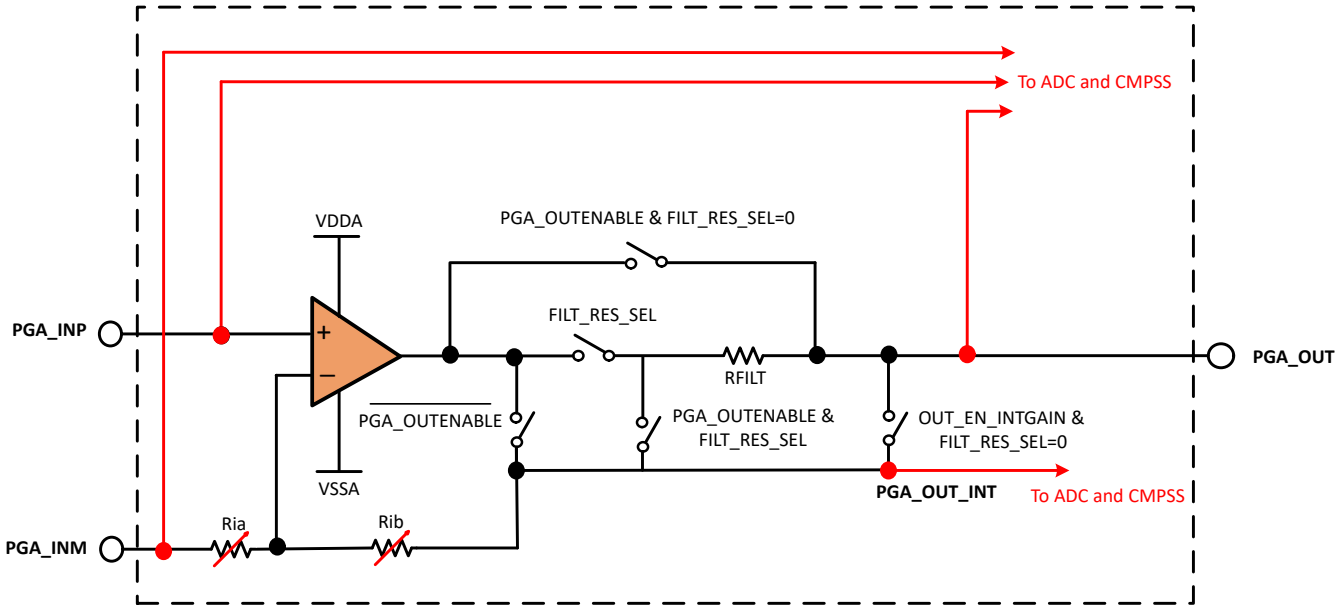


Figure 18-14. PGA Pin Alternate Functions

ADVANCE INFORMATION

18.11 Examples

18.11.1 Non-Inverting Amplifier Using Non-Inverting Mode

Given a small positive signal, the PGA can be used to amplify the signal to increase the dynamic range of ADC sampling and comparator trip monitoring. For example, an input signal with a valid range between 0.25V to 0.75V can be amplified in 4× mode to produce an output signal between 1V to 3V.

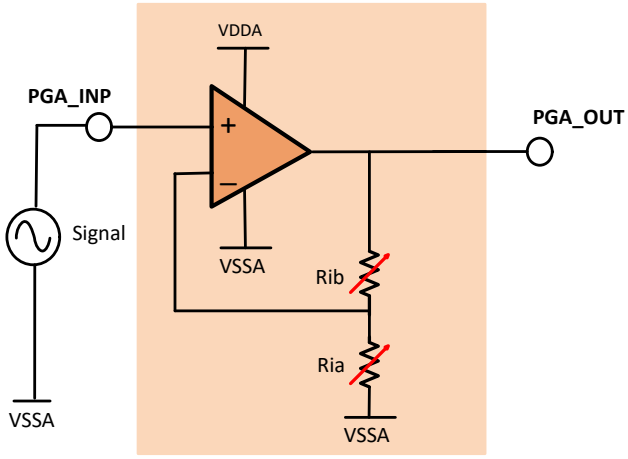


Figure 18-15. Signal Conditioning Using Non-inverting Mode

The amplified output voltage is calculated as:

$$V_{PGA\_OUT} = \left(1 + \frac{R_{ib}}{R_{ib}}\right) V_{Signal} \quad (28)$$

18.11.2 Buffer Mode

Assume that there is an analog sensor that gives a 0 to 12V range signal. To sample the signal using the ADC, a voltage divider along with a unity-gain op-amp is required. In the circuit shown in Figure 18-16, the voltage divider (36kΩ and 100kΩ) brings the 12V sensor voltage down to something less than 3.3V. The 100Ω filter resistor and 1nF capacitor form a low-pass filter with a cut-off frequency of 1.5MHz.

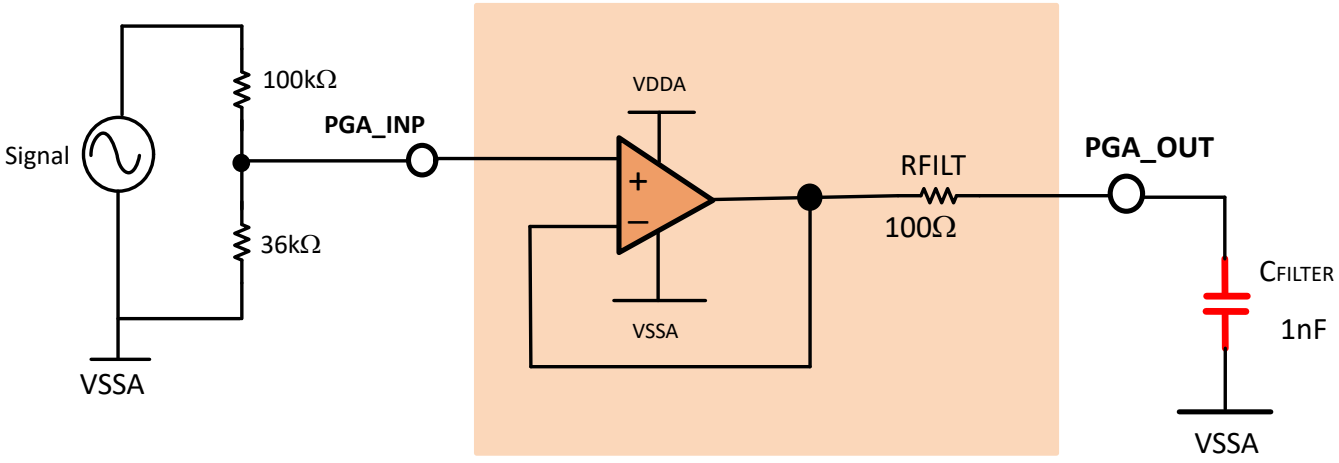


Figure 18-16. Buffer Mode Example

18.11.3 Low-Side Current Sensing

Low-side current sensing connects the sensing resistor between the load and ground. To not adversely affect current flow, the current resistors have a small value that produces a proportionally small voltage across them. The small voltage across the shunt resistor must usually be boosted from tens or hundreds of millivolts to tenths of a volt or volts for upstream conversion by an analog-to-digital converter (ADC). This task can be performed by PGA. Figure 18-17 shows the low-side current sensing using PGA.

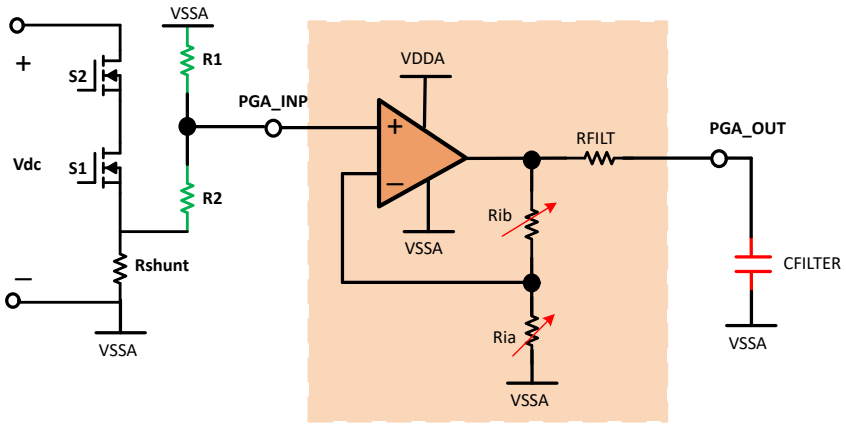


Figure 18-17. PGA Shunt Current Example

In this circuit, the amplifier gain is set by the ratio of R_{ib} divided by R_{ia}. If the divider resistors R_1=R_{ib} and R_2=R_{ia}, then the amplified voltage can be calculated by:

$$V_{PGA\_OUT} = \frac{R_{ib}}{R_{ia}}\left(R_{shunt}I_{load}\right) \quad (29)$$

If low-pass filtering is desired, an external capacitor in conjunction with internal filter resistors R_{FILT} can be used. The filter cutoff frequency is estimated using:

$$f_{cutoff} = \frac{1}{2\pi R_{FILT}C_{FILTER}} \quad (30)$$

ADVANCE INFORMATION

18.11.4 Bidirectional Current Sensing

The PGA senses current flow through a sense resistor in both directions, demonstrated in Figure 18-18. The bidirectional current-sensing capability is achieved by applying a positive voltage at the non-inverting pin to offset the output voltage. A positive differential voltage sensed at the inputs results in an output voltage that is greater than the applied reference voltage; likewise, a negative differential voltage at the inputs results in output voltage that is less than the applied reference voltage. The output voltage of the PGA is given by:

$$V_{PGA\_OUT} = \frac{R_{ib}}{R_{ia}}(R_{sense}I_{load}) + V_{REF} \quad (31)$$

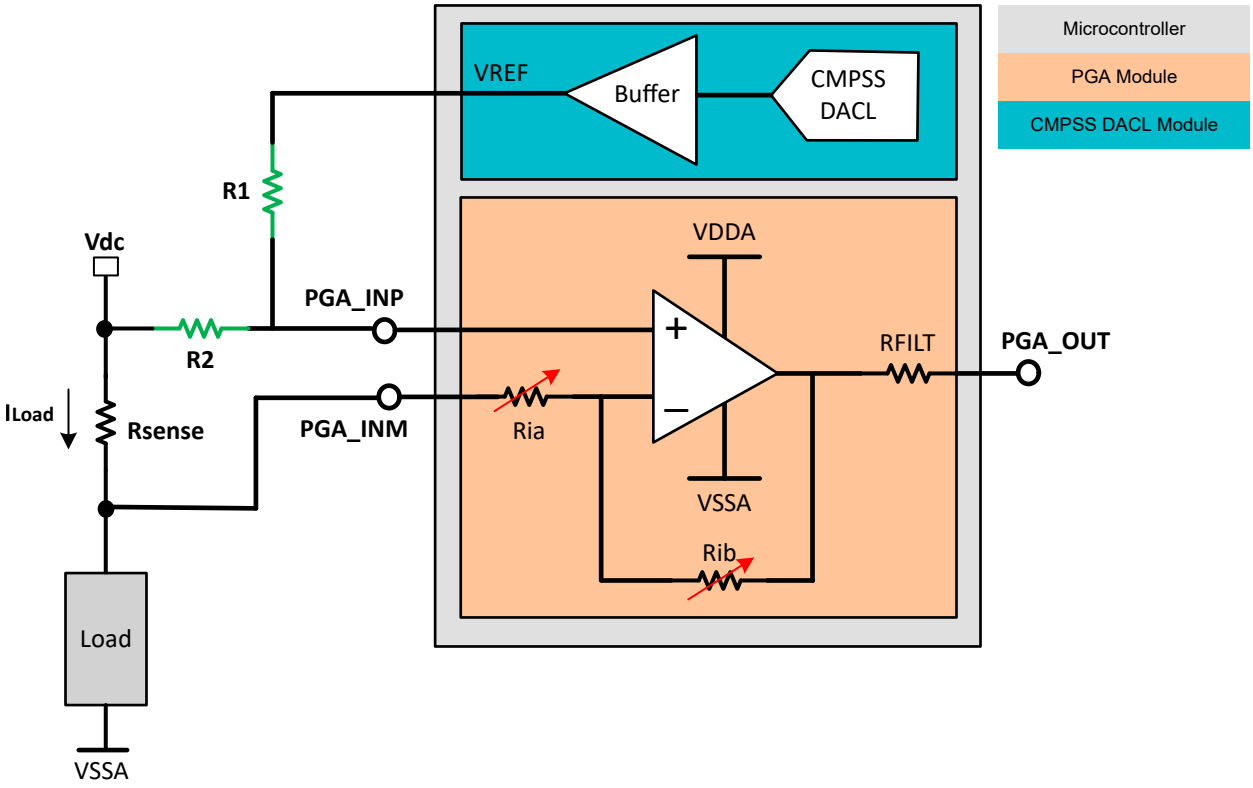


Figure 18-18. Bidirectional Current Sensing

ADVANCE INFORMATION

18.12 PGA Registers

This Section describes the PGA Registers.

18.12.1 PGA Base Address Table

Table 18-8. PGA Base Address Table

Bit Field Name		DriverLib Name	Base Address
Instance	Structure		
Pga0Regs	PGA_REGS	PGA0	0x400F_C000
Pga1Regs	PGA_REGS	PGA1	0x400F_D000
Pga2Regs	PGA_REGS	PGA2	0x400F_E000

18.12.2 PGA_REGS Registers

Table 18-9 lists the memory-mapped registers for the PGA_REGS registers. All register offset addresses not listed in Table 18-9 should be considered as reserved locations and the register contents should not be modified.

Table 18-9. PGA_REGS Registers

Offset	Acronym	Register Name	Write Protection	Section
0h	PGACTL	PGA Control Register	EALLOW,LOCK: PGALOCK_TYP E2.PGACTL	Go
2h	MUXSEL	Mux Selection Register	EALLOW,LOCK: PGALOCK_TYP E2.MUXSEL	Go
4h	OFFSETTRIM	Offset Trim Register	EALLOW,LOCK: PGALOCK_TYP E2.OFFSETTRIM	Go
Ah	PGATYPE	PGA Type Register		Go
Ch	PGALOCK	PGA Lock Register	EALLOW	Go

Complex bit access types are encoded to fit into small table cells. Table 18-10 shows the codes that are used for access types in this section.

Table 18-10. PGA_REGS Access Type Codes

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

1 PGACTL Register (Offset = 0h) [Reset = 0000h]

PGACTL is shown in Figure 18-19 and described in Table 18-11.

Return to the Summary Table.

PGA Control Register

Figure 18-19. PGACTL Register

15	14	13	12	11	10	9	8
CHOP_EXTCTRL	RESERVED					PGA_OUTEN_INTGAIN	PGA_OUTENABLE
R/W-0h	R-0h					R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
GAIN			RESERVED	FILT_RES_SEL			PGAEN
R/W-0h			R-0h	R/W-0h			R/W-0h

Table 18-12. PGACTL Register Field Descriptions

Bit	Field	Type	Reset	Description
15	CHOP_EXTCTRL	R/W	0h	CHOP Signal Control 0 Chop signal is low 1 Chop signal is high Reset type: SYSRSn
14-10	RESERVED	R	0h	Reserved
9	PGA_OUTEN_INTGAIN	R/W	0h	PGA Internal Gain on Pin 0 PGA internal gain path comes to a pin 1 PGA nternal gain path does not come to a pin Reset type: SYSRSn
8	PGA_OUTENABLE	R/W	0h	PGA Output Enable 0 PGA Output does not come to a pin 1 PGA Output does come to a pin Reset type: SYSRSn
7-5	GAIN	R/W	0h	PGA Gain programming-1-64 000 x 1 001 x 2/-1 010 x 4/-3 011 x 8/-7 100 x 16/-15 101 x 32/-31 110 x 64/-63 111 reserved Reset type: SYSRSn
4	RESERVED	R	0h	Reserved
3-1	FILT_RES_SEL	R/W	0h	Filter Resistor Select 000 Filter Disabled 001 50 010 100 011 200 100 400 101 800 110 Filter Disabled 111 Filter Disabled Reset type: SYSRSn

ADVANCE INFORMATION

Table 18-12. PGACTL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
0	PGAEN	R/W	0h	PGA Enable. 0 PGA is disabled and powered down. 1 PGA is enabled. Reset type: SYSRSn

2 MUXSEL Register (Offset = 2h) [Reset = 0000h]

MUXSEL is shown in Figure 18-20 and described in Table 18-12.

Return to the Summary Table.

Mux Selection Register

Figure 18-20. MUXSEL Register

15	14	13	12	11	10	9	8
RESERVED	PGA_CHOP		PGA_FBONPIN	RESERVED	MMUXSEL		
R-0h	R/W-0h		R/W-0h	R-0h	R/W-0h		
7	6	5	4	3	2	1	0
RESERVED	NMUXSEL			RESERVED		PMUXSEL	
R-0h	R/W-0h			R-0h		R/W-0h	

Table 18-14. MUXSEL Register Field Descriptions

Bit	Field	Type	Reset	Description
15	RESERVED	R	0h	Reserved
14-13	PGA_CHOP	R/W	0h	PGA Output Chopping Control 00 Chopping Disabled 01 Reserved 10 ADC assisted chop (using ADCSOC as ctrl) 11 ADC assisted chop (using PGA_CHOP_EXTCTRL register0 Reset type: SYSRSn
12	PGA_FBONPIN	R/W	0h	PGA Feedback to Negative Input Connection 0 PGA_INM and Inverting Input are not connected 1 PGA_INM and Inverting Input are connected Reset type: SYSRSn
11	RESERVED	R	0h	Reserved
10-8	MMUXSEL	R/W	0h	PGA M Mux Select 0x0 Stand alone op-amp or G=1 mode select 0x1 Non-Inverting gain mode select, feeback resistors have path to VSSA 0x2 Inverting gain mode select, feedback resistors have path to local PGA_INM1 pin 0x3 Inverting gain mode select, feedback resistors have path to local PGA_INM2 pin(using PGA_NEG_SHARED input) 0x4 Reserved 0x5 Reserved 0x6 Reserved 0x7 Reserved Reset type: SYSRSn
7	RESERVED	R	0h	Reserved
6-4	NMUXSEL	R/W	0h	PGA Negative Input Mux Select 0x0 Select RTOP (Inter Gain Resistor Tap Point) as inverting input 0x1 Select RTAP (Pre Gain Resistor Tap Point) as inverting input 0x2 Select PGA_INM1 pin as inverting input 0x3 Select PGA_INM2 pin as inverting input 0x4 Reserved 0x5 Reserved 0x6 Reserved 0x7 Reserved Reset type: SYSRSn
3-2	RESERVED	R	0h	Reserved

ADVANCE INFORMATION

Table 18-14. MUXSEL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
1-0	PMUXSEL	R/W	0h	PGA Positive Input Mux Select 0x0 Select PGA_INP1 pin as non inverting input 0x1 Select PGA_INP2 pin as non inverting input 0x2 Select PGA_INP3 pin as non inverting input 0x3 Reserved Reset type: SYSRSn

3 OFFSETTRIM Register (Offset = 4h) [Reset = 00000000h]

OFFSETTRIM is shown in Figure 18-21 and described in Table 18-13.

Return to the Summary Table.

Offset Trim Register

Figure 18-21. OFFSETTRIM Register

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

Table 18-16. OFFSETTRIM Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	RESERVED	R	0h	Reserved
23-16	PGA_OFFSETTRIMP	R/W	0h	PGA internal (analog) offset trim feature for i/p PMOS pair Reset type: SYSRSn
15-8	RESERVED	R	0h	Reserved
7-0	PGA_OFFSETTRIMN	R/W	0h	PGA internal (analog) offset trim feature for i/p NMOS pair Reset type: SYSRSn

ADVANCE INFORMATION

4 PGATYPE Register (Offset = Ah) [Reset = 0201h]

PGATYPE is shown in Figure 18-22 and described in Table 18-14.

Return to the Summary Table.

PGA Type Register

Figure 18-22. PGATYPE Register

15	14	13	12	11	10	9	8
TYPE							
R-2h							
7	6	5	4	3	2	1	0
REV							
R-1h							

Table 18-18. PGATYPE Register Field Descriptions

Bit	Field	Type	Reset	Description
15-8	TYPE	R	2h	PGA Type. Reset type: SYSRSn
7-0	REV	R	1h	PGA Revision. Reset type: SYSRSn

5 PGALOCK Register (Offset = Ch) [Reset = 0000h]

PGALOCK is shown in Figure 18-23 and described in Table 18-15.

Return to the Summary Table.

PGA Lock Register

Figure 18-23. PGALOCK Register

15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED	RESERVED	RESERVED	RESERVED	PGATMCTL	OFFSETTRIM	MUXSEL	PGACTL
WSonce-0h	R-0h	R-0h	R-0h	WSonce-0h	WSonce-0h	WSonce-0h	WSonce-0h

Table 18-20. PGALOCK Register Field Descriptions

Bit	Field	Type	Reset	Description
15-8	RESERVED	R	0h	Reserved
7	RESERVED	WSonce	0h	Reserved
6	RESERVED	R	0h	Reserved
5	RESERVED	R	0h	Reserved
4	RESERVED	R	0h	Reserved
3	PGATMCTL	WSonce	0h	0 Writes to PGATMCTL are enabled. 1 Writes to PGATMCTL are disabled. Reset type: SYSRSn
2	OFFSETTRIM	WSonce	0h	0 Writes to OFFSETTRIM are enabled. 1 Writes to OFFSETTRIM are disabled. Reset type: SYSRSn
1	MUXSEL	WSonce	0h	0 Writes to MUXSEL are enabled. 1 Writes to MUXSEL are disabled. Reset type: SYSRSn
0	PGACTL	WSonce	0h	0 Writes to PGACTL are enabled. 1 Writes to PGACTL are disabled. Reset type: SYSRSn

Chapter 19

Multi-Channel Pulse Width Modulator (MCPWM)

The Multi-channel pulse-width modulator (MCPWM) peripheral is a key element in controlling many of the power electronic systems found in both commercial and industrial equipment. These systems include digital motor control, switch mode power supply control, uninterruptible power supplies (UPS), and other forms of power conversion. The MCPWM peripheral can also perform a digital-to-analog (DAC) function, where the duty cycle is equivalent to a DAC analog value; sometimes referred to as a power DAC.

19.1 Introduction

This chapter includes an overview and information about each submodule:

- Time Base (TB) Submodule
- Counter Compare (CC) Submodule
- Action Qualifier (AQ) Submodule
- Dead-Band Generator (DB) Submodule
- Trip Zone (TZ) Submodule
- Event Trigger (ET) Submodule

The MCPWM module is a reduced version of the type-4 EPWM module featured on other devices. MCPWM has the following enhancements/removals compared to type 4 EPWM:

- **Memory Mapped Active/Shadow Registers:** Additional registers have been added to view the contents of the active register and the shadow register separately for CMPx, TBPRD, DBRED, DBFED, AQCTLA, and AQCTLB.
- **6 Channels per MCPWM module:** In contrast to EPWM, the MCPWM module can feature up to 6 channels on a single module. The 6 channels are treated as 3 pairs of signals, similar to 3 EPWM modules; however, some settings are shared across all 3 channel pairs such as TBPRD, DBRED, DBFED, and TBPHS. This can reduce design flexibility compared to 3 separate EPWM modules.
- **Removed Submodules/feature:** When compared to type-4 EPWM, the following features/submodules have been removed on MCPWM:
  - HRPWM
  - Separate interrupts for TZ events
  - Down-Count mode
  - Digital Compare submodule
  - Chopper module
  - EPWMXLINK
  - T1/T2 action qualifier events
  - Dead-band half-cycle clocking mode

Note

Refer to the *EPWM to MCPWM Migration Guide* for more detail on feature changes and subsequent workarounds when migrating from EPWM to MCPWM.

The MCPWM peripheral is capable of generating complex pulse width waveforms with minimal CPU overhead or intervention. Similar to type-4 EPWM, the MCPWM is split into separate modules each with an independent function. This chapter is divided into individual sections by each MCPWM submodule. For most PWM applications, all modules in MCPWM must be understood and utilized to generate the desired PWM output.

In this document, the letters x and y within a signal or submodule name is used to indicate a generic MCPWM instance and channel pair on a device. For example, output signals MCPWMx_yA and MCPWMx_yB refer to the output signals from the MCPWMx instance and y channel pair (note that there are up to 3 channel pairs per MCPWM instance). Thus, MCPWM1_1A and MCPWM1_1B belong to MCPWM1 channel pair 1 and likewise MCPWM1_3A and MCPWM1_3B belong to MCPWM1 channel pair 3.

19.1.1 PWM Related Collateral

Foundational Materials

- *Real-Time Control Reference Guide*
  - Refer to the PWM section

Getting Started Materials

- *Flexible PWMs Enable Multi-Axis Drives, Multi-Level Inverters Application Report*
- Getting Started with the C2000 ePWM Module (Video)

ADVANCE INFORMATION

- *Using PWM Output as a Digital-to-Analog Converter on a TMS320F280x Digital Signal Control Application Report*
  - Chapters 1 to 6 are Fundamental material, derivations, and explanations that are useful for learning about how PWM can be used to implement a DAC. Subsequent chapters are Getting Started and Expert material for implementing in a system.
- *Using the Enhanced Pulse Width Modulator (ePWM) Module Application Report*

Expert Materials

- C2000 real-time microcontrollers - Reference designs
  - See TI designs related to specific end applications used.

19.1.2 MCPWM Overview

The MCPWM module consists of up to 3 complete PWM channel pairs with the following signal names:

- MCPWMx_1A
- MCPWMx_1B
- MCPWMx_2A
- MCPWMx_2B
- MCPWMx_3A
- MCPWMx_3B

Multiple MCPWM modules can exist on a single device, with each module denoted by a different value of "x".

MCPWM Instance	Number of Channels
MCPWM0	6
MCPWM1	6
MCPWM2	6
MCPWM3	6
MCPWM4	6

Note that not all channels need to be used in 6-channel. For example, to use only 4 channels from a 6-channel MCPWM instance, simply configure the GPIO mux to avoid routing the remaining two channels to any device pin.

The MCPWM modules are chained together by way of a clock synchronization scheme that allows them to operate as a single system when required. Additionally, this synchronization scheme can be extended to the capture peripheral submodules (eCAP).

Each MCPWM module supports the following features:

- Dedicated 16-bit time-base counter with period and frequency control
- Up to 6 PWM outputs(3 channel pairs). Each channel pair can be used in the following configurations:
  - Two independent PWM outputs with single-edge operation
  - Two independent PWM outputs with dual-edge symmetric operation
  - One independent PWM output with dual-edge asymmetric operation
- Asynchronous override control of PWM signals through software.
- Programmable phase-control support for lag or lead operation relative to other MCPWM modules.
- Hardware-locked (synchronized) phase relationship on a cycle-by-cycle basis.
- Dead-band generation with independent rising and falling edge delay control.
- Programmable trip zone allocation of both cycle-by-cycle trip and one-shot trip on fault conditions.
- A trip condition can force either high, low, or high-impedance state logic levels at PWM outputs.
- All events can trigger both CPU interrupts and ADC start of conversion (SOC)
- Programmable event prescaling minimizes CPU overhead on interrupts.

Each MCPWM module consists of 6 submodules and is connected within a system by way of the signals shown in Figure 19-1.

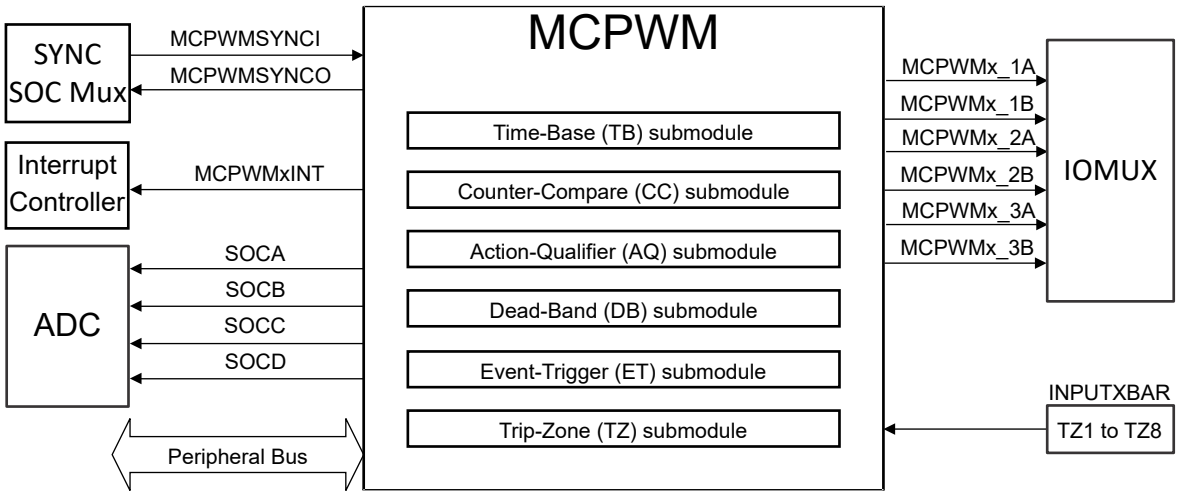


Figure 19-1. Submodules and Signal Connections for an MCPWM Module

Figure 19-2 shows more internal details of a single MCPWM module. The main signals used by the MCPWM module are:

- **MCPWM output signals (MCPWMx_yA and MCPWMx_yB):** The MCPWM output signals are made available external to the device.
- **Trip-zone signals ($\overline{\text{TZ1}}$ to $\overline{\text{TZ8}}$):** These input signals alert the MCPWM module of fault conditions external to the MCPWM module. Each submodule on a device can be configured to either use or ignore any of the trip-zone signals. Each TZx signal is directly connected to the output of one of the 8 MCPWM X-BAR outputs.
- **Time-base synchronization input (MCPWMxSYNCI), output (MCPWMxSYNCO), and peripheral (MCPWMxSYNCPER) signals:** For more information, see Section 19.4.3.3 .

  Each MCPWM module also generates another PWMSYNC signal called MCPWMxSYNCPER. MCPWMxSYNCPER goes to the CMPSS for synchronization purposes. Functionality is configured using the TBCTL register. For more information on how MCPWMxSYNCPER is used by the CMPSS, see the *Comparator Subsystem (CMPSS)* chapter.
- **ADC start-of-conversion signals:** Each MCPWM module has four ADC start of conversion signals(SOCA, SOCB, SOCC, SOCD). Any MCPWM module can trigger a start of conversion. Whichever event triggers the start of conversion is configured in the event-trigger submodule of the MCPWM.
- **Comparator output signals (COMPxOUT):** Output signals from the comparator module can be fed through the MCPWM X-BAR to one or all of the 8 trip inputs to induce one-shot or cycle-by-cycle trip events. Refer to the *Crossbar (X-BAR)* chapter and the Trip-Zone submodule section of this chapter for more information.
- **Peripheral bus:** The peripheral bus is 32-bits wide and allows both 16-bit and 32-bit writes to the MCPWM register file.

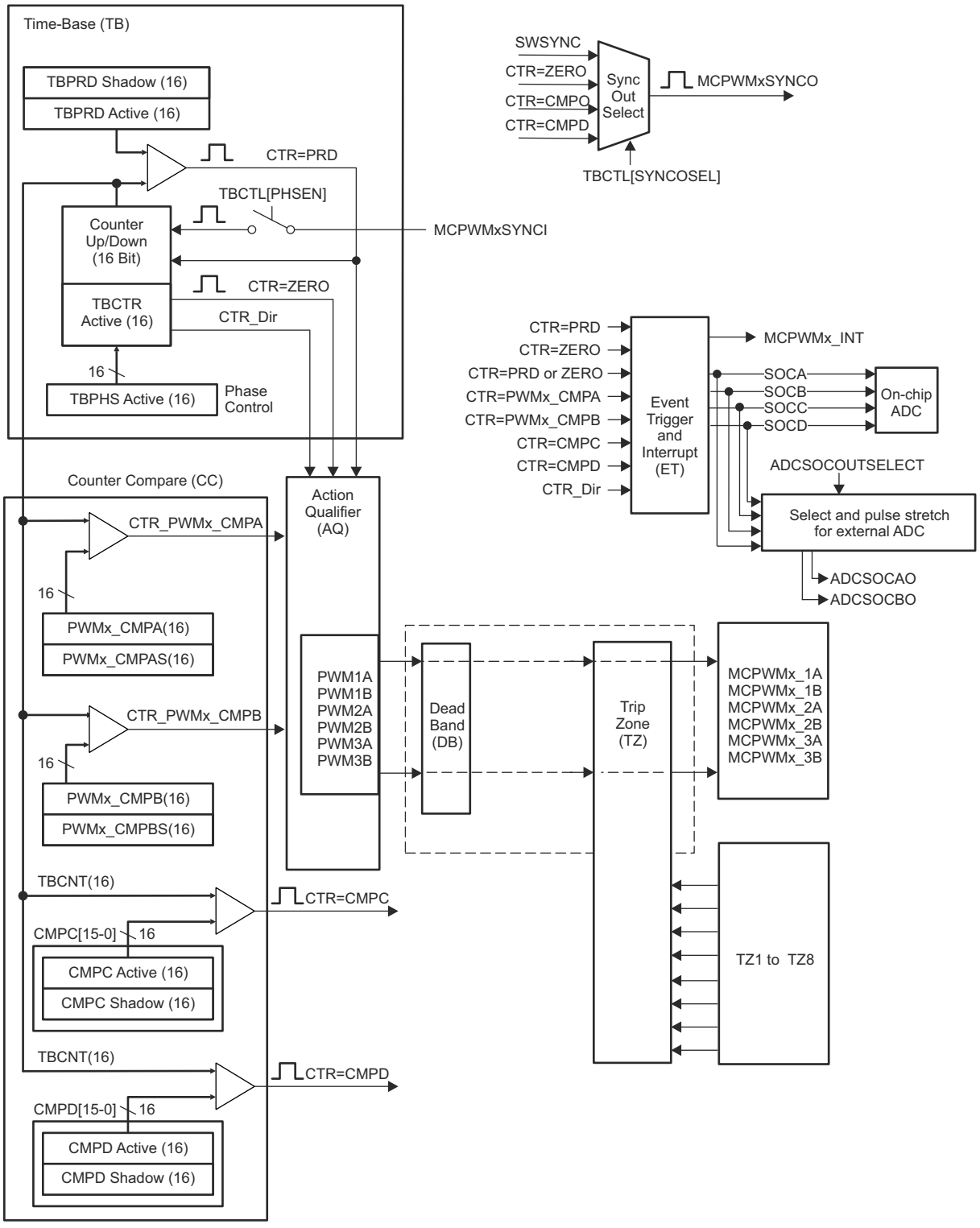


Figure 19-2. MCPWM Modules and Critical Internal Signal Interconnects

ADVANCE INFORMATION

19.2 Configuring Device Pins

To connect the device input pins to the MCPWM module, the Input X-BAR and PWM X-BAR must be used. Some examples of when an external signal can be needed are TZx and MCPWMxSYNCI. Any GPIO on the device can be configured as an input. All trip signals must be routed through the PWM X-BAR, which can take in the Input X-BAR as an input to utilize an external trip signal.

The IOMUX pin function register must be configured for this peripheral. See the *IOMUX* and *XBAR* chapters for additional details.

19.3 MCPWM Modules Overview

The 6 submodules are included in every MCPWM peripheral. Each of these submodules performs specific tasks that can be configured by software.

Table 19-1 lists the key submodules together with a list of the main configuration parameters. For example, if you need to adjust or control the duty cycle of a PWM waveform, see the counter-compare submodule in Section 19.5 for relevant details.

Table 19-1. Submodule Configuration Parameters

Submodule	Configuration Parameter or Option
Time Base (TB)	• Scale the time-base clock (TBCLK) relative to the MCPWM clock (MCPWMCLK). • Configure the PWM time-base counter (TBCTR) frequency or period. • Set the mode for the time-base counter: – count-up mode: used for asymmetric PWM – count-up-and-down mode: used for symmetric PWM • Configure the time-base phase relative to another MCPWM module. • Synchronize the time-base counter between modules through hardware or software. • Configure the direction (up or down) of the time-base counter after a synchronization event. • Configure how the time-base counter behaves when the device is halted by an emulator. • Specify the source for the synchronization output of the MCPWM module. • Configure one shot and global load of registers in this module.
Counter Compare (CC)	• Specify the PWM duty cycle for each outputs MCPWMx_yA and outputs MCPWMx_yB • Specify the time at which switching events occur on the MCPWMx_yA or MCPWMx_yB outputs. • Specify the programmable delay for interrupt and SOC generation with additional comparators • Configure one shot and global load of registers in this module.
Action Qualifier (AQ)	• Specify the type of action taken when a time-base counter-compare – No action taken – Output MCPWMx_yA and MCPWMx_yB switched high – Output MCPWMx_yA and MCPWMx_yB switched low – Output MCPWMx_yA and MCPWMx_yB toggled • Force the PWM output state through software control • Configure and control the PWM dead band through software • Configure one shot and global load of registers in this module.
Dead-Band Generator (DB)	• Control of traditional complementary dead-band relationship between upper and lower switches • Specify the output rising-edge-delay value • Specify the output falling-edge delay value • Bypass the dead-band module entirely. In this case the PWM waveform is passed through without modification. • Allow MCPWMxB phase shifting with respect to the MCPWMxA output. • Configure one shot and global load of registers in this module.

Table 19-1. Submodule Configuration Parameters (continued)

Submodule	Configuration Parameter or Option
Trip Zone (TZ)	• Configure the MCPWM module to react to one, all, or none of the trip-zone signals or digital compare events. • Specify the trip action taken when a fault occurs: – Force outputs high – Force outputs low – Force outputs to a high-impedance state – Configure outputs to ignore any trip condition. • Configure how often the MCPWM reacts to each trip-zone signal: – One-shot – Cycle-by-cycle • Bypass the trip-zone module entirely. • Programmable option for cycle-by-cycle trip clear
Event Trigger (ET)	• Enable the MCPWM events that trigger an interrupt. • Enable MCPWM events that trigger an ADC start-of-conversion event. • Specify the rate at which events cause triggers (every occurrence or every 2nd or up to 15th occurrence) • Poll, set, or clear event flags

19.4 Time-Base (TB) Submodule

Each MCPWM module has a time-base submodule that determines all of the event timing for the MCPWM module. Built-in synchronization logic allows the time-base of multiple MCPWM modules to work together as a single system.

Figure 19-3 illustrates the time-base submodule within the MCPWM.

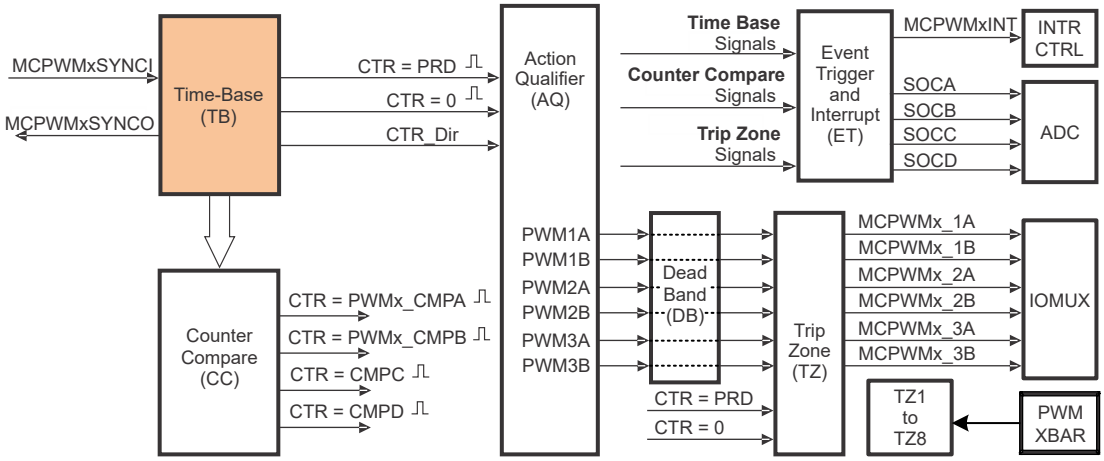


Figure 19-3. Time-Base Submodule

19.4.1 Purpose of the Time-Base Submodule

The time-base submodule can be configured for the following:

- Specify the MCPWM time-base counter (TBCTR) frequency or period to control how often events occur.
- Manage time-base synchronization with other MCPWM modules.
- Maintain a phase relationship with other MCPWM modules.
- Set the time-base counter to count-up, or count-up-and-down mode.
- Generate the following events:
  - CTR = PRD: Time-base counter equal to the specified period (TBCTR = TBPRD).
  - CTR = Zero: Time-base counter equal to zero (TBCTR = 0x00).
- Configure the rate of the time-base clock; a prescaled version of the MCPWM clock (MCPWMCLK). This allows the time-base counter to increment/decrement at a slower rate.

Note

If required by the application code to update the TBCTR value through software while the TBCTR is counting, note that the time-base module needs at least 1 TBCLK cycle for the time-base related events to be realized. Hence, the TBCTR can be written with TBCTR = PRD-1 instead of TBCTR = PRD (in case the counter is counting up) and can be written as TBCTR = 1 instead of TBCTR = 0 (in case the counter is counting down) for the events to be realized.

ADVANCE INFORMATION

19.4.2 Controlling and Monitoring the Time-Base Submodule

The block diagram in Figure 19-4 shows the critical signals and registers of the time-base submodule. Table 19-2 provides descriptions of the key signals associated with the time-base submodule.

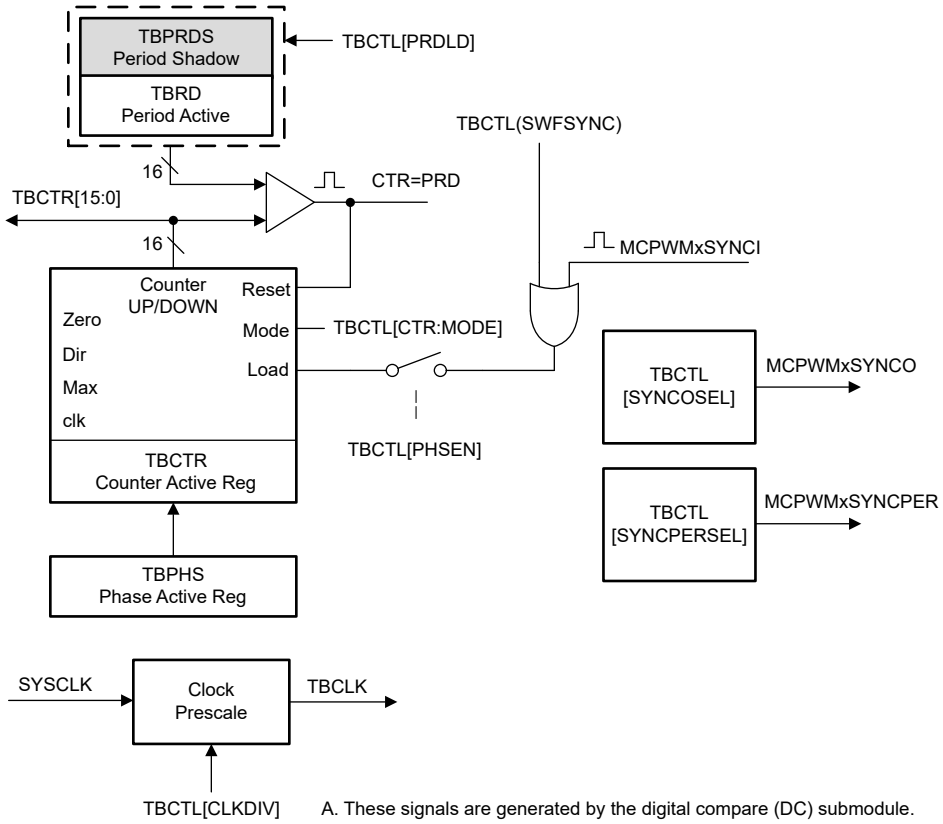


Figure 19-4. Time-Base Submodule Signals and Registers

ADVANCE INFORMATION

Table 19-2. Key Time-Base Signals

Signal	Description
MCPWMxSYNCI	Time-base synchronization input. Input pulse used to synchronize the time-base counter with the counter of other MCPWM modules. For information on the synchronization order of a particular device, see Section 19.4.3.3 .
MCPWMxSYNCO	Time-base synchronization output. This output pulse is used to synchronize the counter of other MCPWM modules. Using the TBCTL.SYNCOSEL bit, the source of the output pulse is selected.
MCPWMxSYNCPER	Time-base peripheral synchronization output. This output signal is used to synchronize the CMPSS to the MCPWM. The output signal can be configured using the TBCTL register.
CTR = PRD	Time-base counter equal to the specified period. This signal is generated whenever the counter value is equal to the active period register value. That is when TBCTR = TBPRD.
CTR = Zero	Time-base counter equal to zero. This signal is generated whenever the counter value is zero. That is when TBCTR equals 0x00.
CTR = PWMx_CMPB	Time-base counter equal to active counter-compare B register (TBCTR = PWMx_CMPB). This event is generated by the counter-compare submodule and used by the synchronization out logic. There is one CMPB register per PWM channel pair, denoted by "x".
CTRDIR	Time-base counter direction. Indicates the current direction of the MCPWM time-base counter. The signal is high when the counter is increasing and the signal is low when the counter is decreasing. The state of this signal can be observed in the TBSTS register.
TBCLK	Time-base clock. This is a prescaled version of the MCPWM clock (MCPWMCLK) and is used by all submodules within the MCPWM. This clock determines the rate at which time-base counter increments or decrements.

19.4.3 Calculating PWM Period and Frequency

The frequency of PWM events is controlled by the time-base period (TBPRD) register and the mode of the time-base counter. Figure 19-5 shows the period (T_{pwm}) and frequency (F_{pwm}) relationships for the up-count and up-down-count time-base counter modes when the period is set to 4 (TBPRD = 4). The time increment for each step is defined by the time-base clock (TBCLK) which is a prescaled version of the MCPWM clock (MCPWMCLK).

The time-base counter has two modes of operation selected by the time-base control register (TBCTL.CTRMODE):

- **Up-Down Count Mode:** In up-down count mode, the time-base counter starts from zero and increments until the period (TBPRD) value is reached. When the period value is reached, the time-base counter then decrements until the counter reaches zero. At this point, the counter repeats the pattern and begins to increment.
- **Up-Count Mode:** In up-count mode, the time-base counter starts from zero and increments until the counter reaches the value in the period register (TBPRD). When the period value is reached, the time-base counter resets to zero and begins to increment once again.

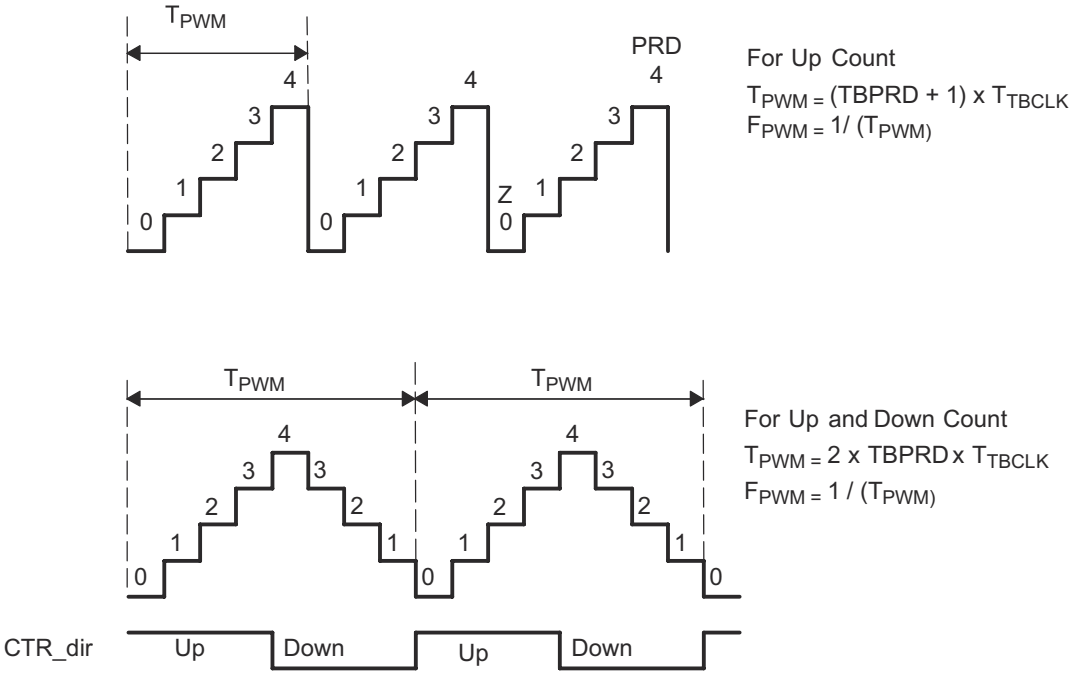


Figure 19-5. Time-Base Frequency and Period

19.4.3.1 Time-Base Period Shadow Register

The time-base period register (TBPRD) has a shadow register. Shadowing allows the register update to be synchronized with the hardware. The following definitions are used to describe all shadow registers in the MCPWM module:

- **Active Register:** The active register controls the hardware and is responsible for actions that the hardware causes or invokes.
- **Shadow Register:** The shadow register buffers provide a temporary holding location for the active register and have no direct effect on any control hardware. At a strategic point in time, the shadow register content is transferred to the active register. This prevents corruption or spurious operation due to the register being asynchronously modified by software.

In contrast to the EPWM module, MCPWM has two separate addresses for the active and shadow register. When shadow mode is enabled, write to the shadow register instead of the active register for shadow loading to occur. Writing to the active register immediately updates TBPRD. When shadowing is disabled, a write to the shadow register does not update the active register. Shadowing is enabled or disabled by the TBCTL[PRDLD] bit. This bit enables and disables the TBPRD shadow register as follows:

- **Time-Base Period Shadow Mode:** The TBPRD shadow register (TBPRDS) is enabled when TBCTL[PRDLD] = 0. When shadow mode is enabled, the value written to the shadow register is loaded to the active register on the next shadow load or global load event. The shadow register contents are transferred to the active register (TBPRD ← TBPRDS) when the time-base counter equals zero (TBCTR = 0x00). The PRDLDSYNC bit is valid only if TBCTL[PRDLD] = 0. A write to the active register (TBPRD) immediately updates TBPRD.

  The global load control mechanism can also be used with the time-base period register by configuring the appropriate bits in the global load configuration register (GLDCTL.GLD). When global load mode is selected the transfer of contents from shadow register to active register, for all registers that have a corresponding shadow register, occurs at the same event as defined by the configuration bits in Global Shadow to Active Load Control Register (GLDCTL.GLDMODE). Global load control mechanism is explained in Section 19.4.6 .
- **Time-Base Period Immediate Load Mode:** If immediate load mode is selected (TBCTL[PRDLD] = 1), then a read from or a write to the TBPRD memory address goes directly to the active register.

19.4.3.2 Time-Base Clock Synchronization

The TBCLKSYNC bit in the peripheral clock enable registers allows all users to globally synchronize all enabled MCPWM modules to the time-base clock (TBCLK). When set, all enabled MCPWM module clocks are started with the first rising edge of TBCLK aligned. For synchronized TBCLKs, the prescalers for each MCPWM module must be set identically.

The proper procedure for enabling MCPWM clocks is as follows:

1. Set TBCLKSYNC = 0
2. Configure MCPWM modules
3. Set TBCLKSYNC = 1

19.4.3.3 Time-Base Counter Synchronization

The MCPWM synchronization scheme allows for increased flexibility of synchronization of the MCPWM modules. Each MCPWM module has a synchronization input (MCPWMxSYNCI), a synchronization output (MCPWMxSYNCO) and a peripheral synchronization output (MCPWMxSYNCPER). Refer to Section 19.4.3.4 for a list of all sync inputs including INPUTXBAR5 and INPUTXBAR6. The TBCTL.SYNCOSEL bit shows the sources that can be used for SYNCO.

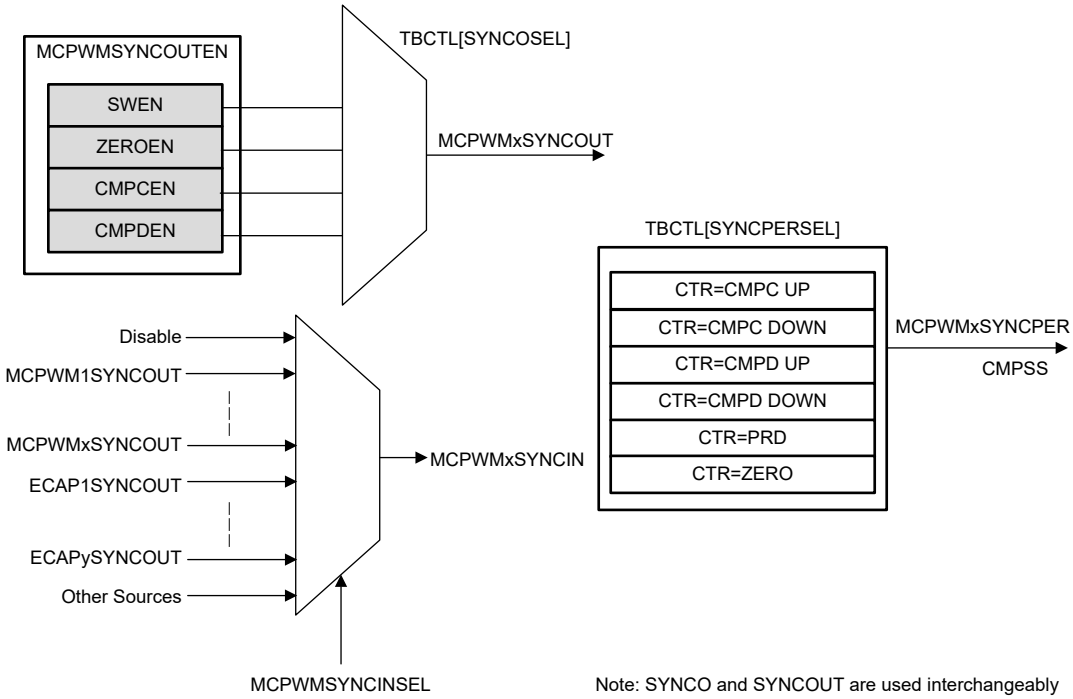
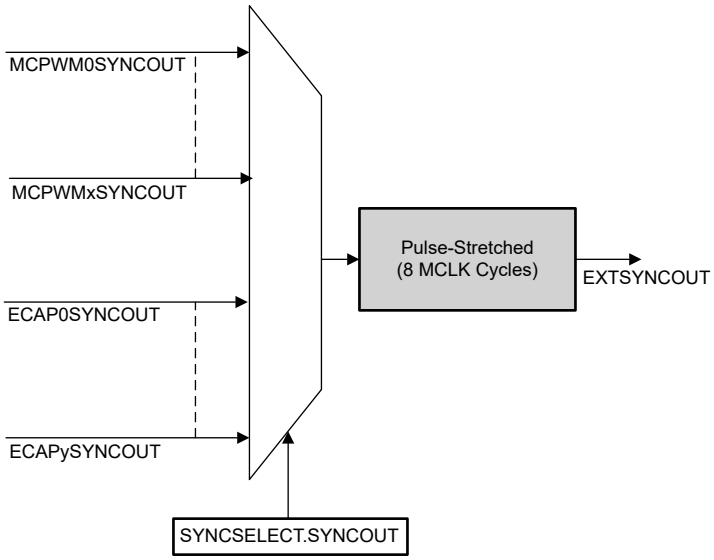


Figure 19-6. Time-Base Counter Synchronization Scheme



MCPWMxSYNCOUT (where x = number of MCPWM modules)
ECAPySYNCOUT (where y = number of ECAP modules)

Figure 19-7. MCPWM External SYNC Output

ADVANCE INFORMATION

Note

See the data sheet for the number of MCPWM and eCAP modules available on your specific device.

Each MCPWM module can be configured to use or ignore the synchronization input. If the TBCTL[PHSEN] bit is set, then the time-base counter (TBCTR) of the MCPWM module is automatically loaded with the phase register (TBPHS) contents when one of the following conditions occur:

- **MCPWMxSYNCI: Synchronization Input Pulse:** The value of the phase register is loaded into the counter register when an input synchronization pulse is detected (TBPHS → TBCTR). This operation occurs on the next valid time-base clock (TBCLK) edge.

  The delay from internal control module to target modules is given by:
  – if (TBCLK = MCPWMCLK): 2 x MCPWMCLK
  – if (TBCLK < MCPWMCLK): 1 x TBCLK
- **Software Forced Synchronization Pulse:** Writing a 1 to the TBCTL[SWFSYNC] control bit invokes a software forced synchronization. This pulse is ORed with the synchronization input signal, and therefore has the same effect as a pulse on MCPWMxSYNCI.

Note

If the MCPWMxSYNCI signal is held high, the sync does not continuously occur. The MCPWMxSYNCI is rising edge activated.

When modifying the TBPHS register during run-time, missed action qualifier events can occur due to sudden jumps in the TBCTR value at the time of the SYNCIN pulse. To recreate the behavior of missed action qualifier events, a CMPB event can be needed if CMPB is not already utilized.

This feature enables the MCPWM module to be automatically synchronized to the time base of another MCPWM module. Lead or lag phase control can be added to the waveforms generated by different MCPWM modules to synchronize them. In up-down-count mode, the TBCTL[PHSDIR] bit configures the direction of the time-base counter immediately after a synchronization event. The new direction is independent of the direction prior to the synchronization event. The PHSDIR bit is ignored in count-up or count-down modes (see Section 19.4.5 for examples).

Clearing the TBCTL[PHSEN] bit configures the MCPWM to ignore the synchronization input pulse.

19.4.3.4 MCPWM SYNC Selection

#unique_494 specifies the sources for the MCPWM SYNC input.

Table 19-3. MCPWM SYNC Selection

PWM*.SYNCINSEL ECAP*.SYNCINSEL	Input Signal
0	Disable (Tie low)
1	INPUTXBAROUT5
2	INPUTXBAROUT6
3	ECAP0SYNCOUT
4	ECAP1SYNCOUT
5	PWM0SYNCOUT
6	PWM1SYNCOUT
7	PWM2SYNCOUT
8	PWM3SYNCOUT
9	PWM4SYNCOUT
10-31	Reserved (Tie low)

Note

The SYNCOUT of a given MCPWMx or ECAPx can not be looped back to the SYNCIN of the same peripheral instance.

19.4.4 Phase Locking the Time-Base Clocks of Multiple MCPWM Modules

The TBCLKSYNC bit can be used to globally synchronize the time-base clocks of all enabled MCPWM modules on a device. This bit is part of the device's clock enable registers and is described in the *System Control and Interrupts* chapter. When TBCLKSYNC = 0, the time-base clock of all MCPWM modules is stopped (default). When TBCLKSYNC = 1, all MCPWM time-base clocks are started with the rising edge of TBCLK aligned. For synchronized TBCLKs, the prescaler bits in the TBCTL register of each MCPWM module must be set identically. The proper procedure for enabling the MCPWM clocks is:

1. Enable the individual MCPWM module clocks. This is described in the *Clock Management Module (CKM)* chapter.
2. Set TBCLKSYNC = 0. This stops the time-base clock within any enabled MCPWM module.
3. Configure the prescaler values and desired MCPWM modes.
4. Set TBCLKSYNC = 1.

19.4.5 Time-Base Counter Modes and Timing Waveforms

The time-base counter operates in one of three modes:

- Up-count mode that is asymmetrical
- Up-down-count that is symmetrical
- Frozen where the time-base counter is held constant at the current value

To illustrate the operation of the first three modes, the following timing diagrams show when events are generated and how the time-base responds to an MCPWMxSYNCI signal.

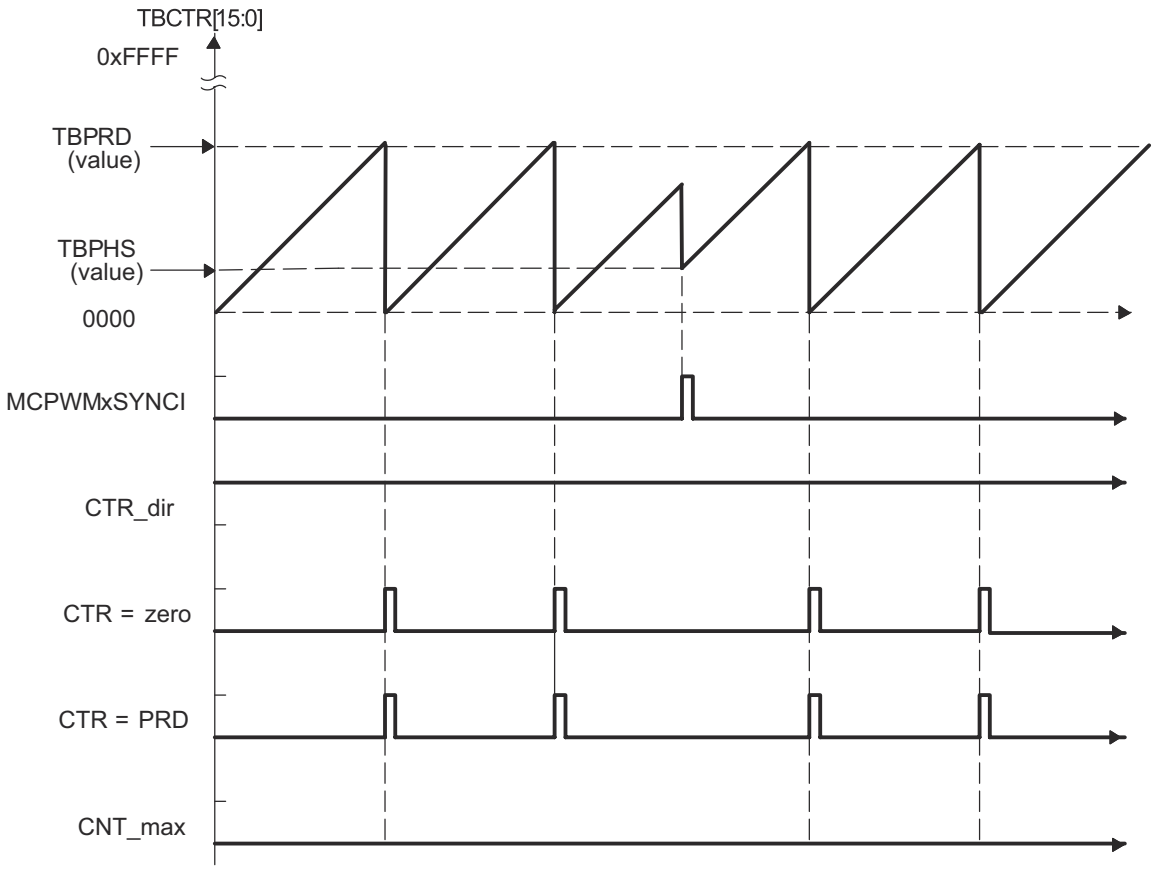


Figure 19-8. Time-Base Up-Count Mode Waveforms

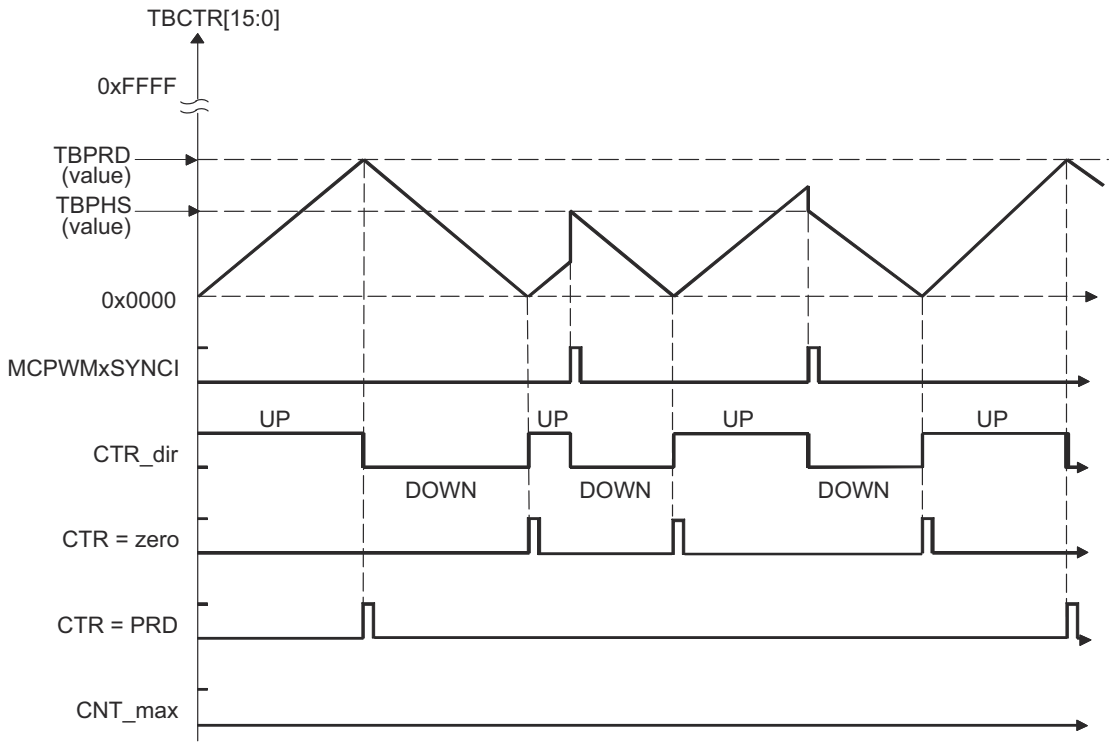


Figure 19-9. Time-Base Up-Down Count Waveforms, TBCTL[PHSDIR = 0] Count Down On Synchronization Event

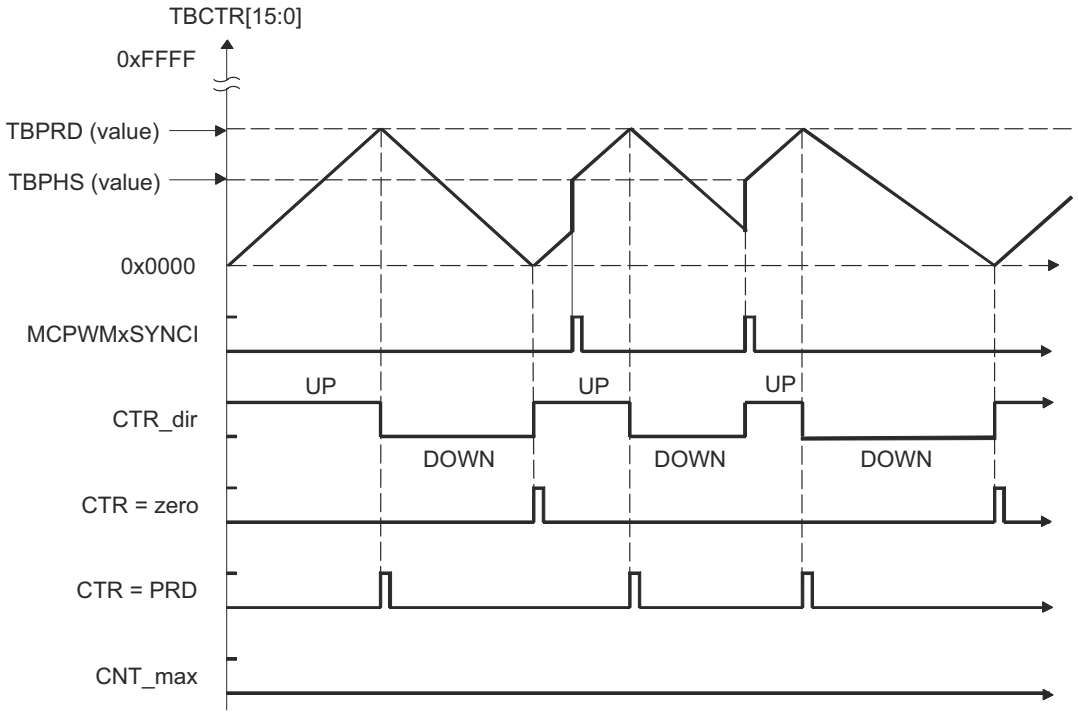


Figure 19-10. Time-Base Up-Down Count Waveforms, TBCTL[PHSDIR = 1] Count Up On Synchronization Event

ADVANCE INFORMATION

19.4.6 Global Load

Figure 19-11 shows the signals and registers associated with the global load feature.

When this feature is enabled, the transfer of contents from the shadow register to the active register, for all registers that have a shadow register, occurs at the same event as defined by the configuration bits in Global Shadow to Active Load Control Register (GLDCTL[GLDMODE]). When GLDCTL[GLD] = 1, shadow to active load event selection bits for individual shadowed registers are ignored and global load mode takes effect for all registers with a corresponding shadow register.

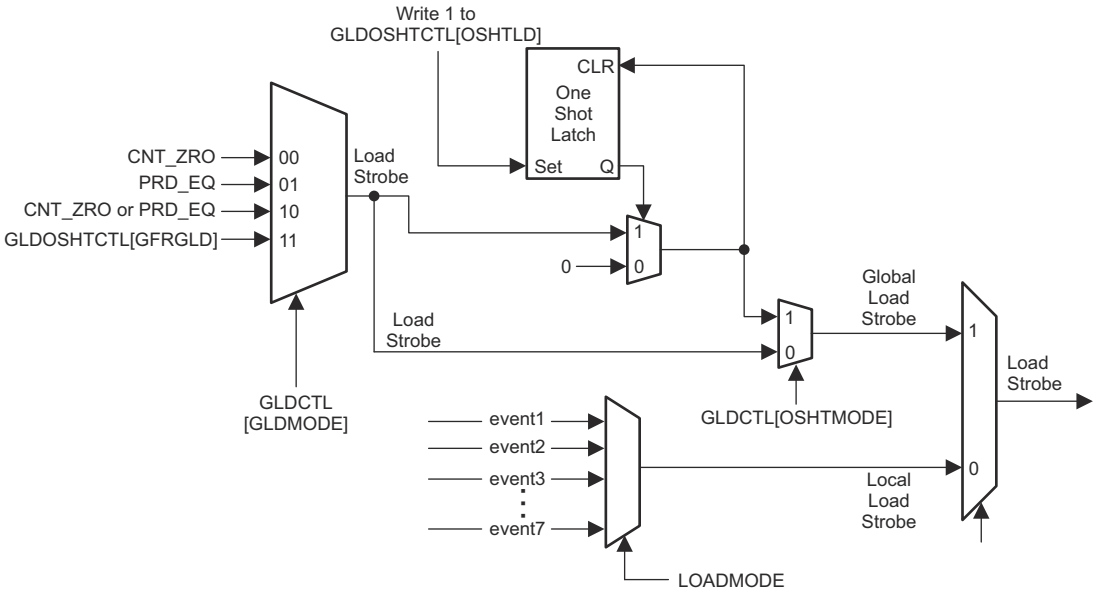


Figure 19-11. Global Load: Signals and Registers

19.4.6.1 One-Shot Load Mode

This feature causes the shadow-register to active-register transfers to occur once. When GLDOSHTCTL[OSHTLD] = 1 and one shot global load is enabled using GLDCTL[OSHTMODE], the shadow-register to active-register transfer takes place on the event selected by GLDCTL[GLDMODE]. Software force loading of contents from the shadow register to an active register is possible by using GLDOSHTCTL[GFRCLD].

ADVANCE INFORMATION

19.5 Counter-Compare (CC) Submodule

Figure 19-12 illustrates the counter-compare submodule within the MCPWM.

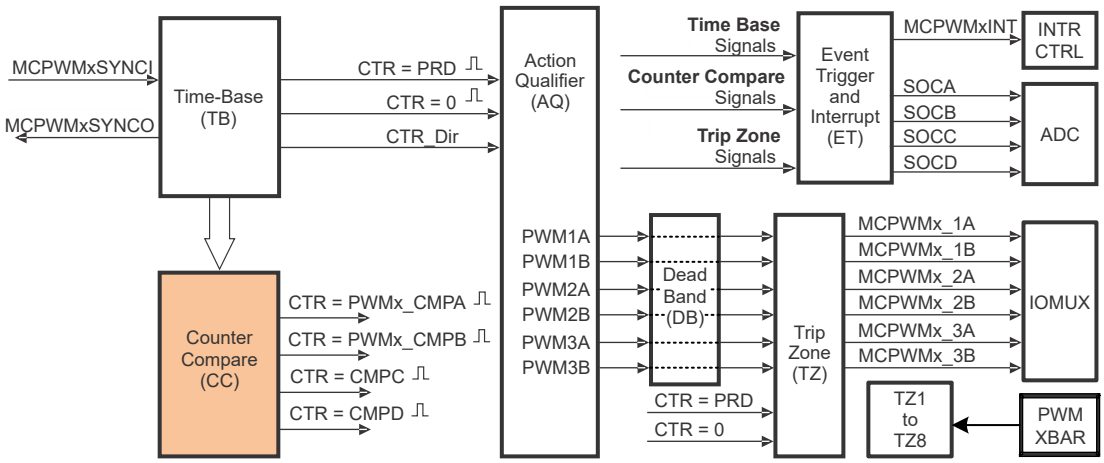


Figure 19-12. Counter-Compare Submodule

19.5.1 Purpose of the Counter-Compare Submodule

The counter-compare submodule takes as input the time-base counter value. This value is continuously compared to the counter-compare A (PWMx_CMPA), counter-compare B (PWMx_CMPB), counter-compare C (CMPC), and counter-compare D (CMPD) registers. When the time-base counter is equal to one of the compare registers, the counter-compare unit generates an appropriate event. Note that each MCPWM module can have up to three PWM pairs, denoted by "PWMx". There is a CMPA and CMPB register for each pair, but only one CMPC and CMPD register for the entire MCPWM module.

The counter-compare:

- Generates events based on programmable time stamps using the CMPA, CMPB, CMPC, and CMPD registers:
  - CTR = PWMx_CMPA: Time-base counter equals counter-compare A register for PWMx (TBCTR = PWMx_CMPA)
  - CTR = PWMx_CMPB: Time-base counter equals counter-compare B register (TBCTR = PWMx_CMPB)
  - CTR = PWMx_CMPC: Time-base counter equals counter-compare C register (TBCTR = CMPC)
  - CTR = PWMx_CMPD: Time-base counter equals counter-compare D register (TBCTR = CMPD)
- Controls the PWM duty cycle, if the action-qualifier submodule is configured appropriately using counter-compare A (PWMx_CMPA) and counter-compare B (PWMx_CMPB)
- Shadows new compare values to prevent corruption or glitches during the active PWM cycle

19.5.2 Controlling and Monitoring the Counter-Compare Submodule

The counter-compare submodule operation is shown in Figure 19-13.

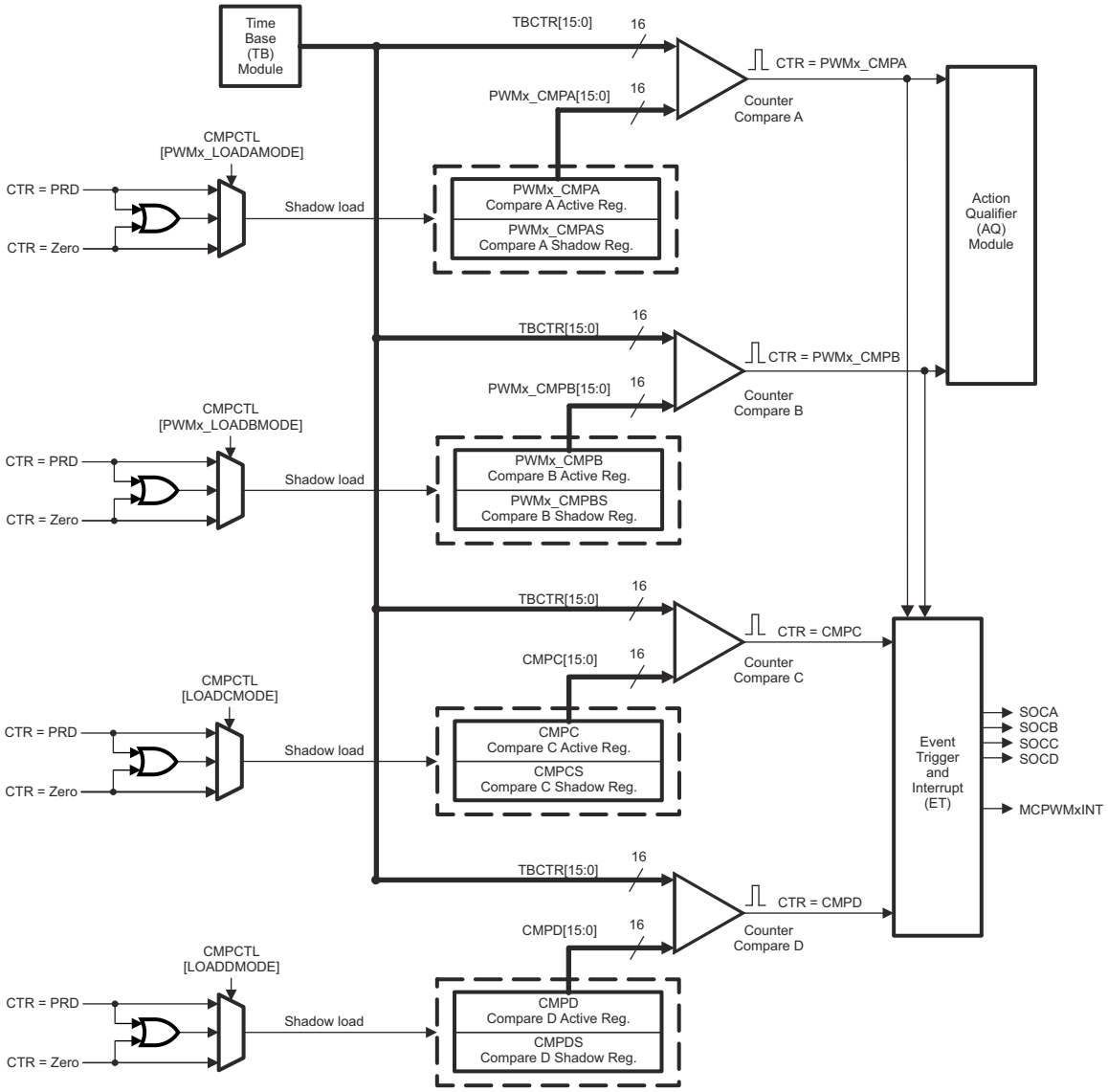


Figure 19-13. Detailed View of the Counter-Compare Submodule

19.5.3 Operational Highlights for the Counter-Compare Submodule

The counter-compare submodule is responsible for generating events that can be used in the action-qualifier and event-trigger submodules:

1. CTR = PWMx_CMPA: Time-base counter equal to counter-compare A register (TBCTR = PWMx_CMPA).
2. CTR = PWMx_CMPB: Time-base counter equal to counter-compare B register (TBCTR = PWMx_CMPB).
3. CTR = CMPC: Time-base counter equal to counter-compare C register (TBCTR = CMPC). This event can be used to generate an event in the event trigger submodule only.
4. CTR = CMPD: Time-base counter equal to counter-compare D register (TBCTR = CMPD). This event can be used to generate an event in the event trigger submodule only.

For up-count mode, each event occurs only once per cycle. For up-down count mode, each event occurs twice per cycle if the compare value is between 0x00-TBPRD; and once per cycle if the compare value is equal to 0x00 or equal to TBPRD. These events are applied to the action-qualifier submodule where the events are qualified by the counter direction and converted into actions if enabled. Refer to Section 19.6.1 for more details.

The counter-compare registers PWMx_CMPA and PWMx_CMPB each have an associated shadow register. Shadowing provides a way to keep updates to the registers synchronized with the hardware. When shadowing is used, updates to the active registers only occur at strategic points. This prevents corruption or spurious operation due to the register being asynchronously modified by software. There is a separate memory address for the active and shadow registers. To utilize the shadow loading feature, the CPU writes to the corresponding shadow register instead of the active register. The behavior of the two load modes are described as follows:

Shadow Mode:

The shadow mode for the PWMx_CMPA and PWMx_CMPB is always enabled, however the user must write to the corresponding shadow register(PWMx_CMPAS or PWMx_CMPBS) for shadow loading to occur.

The content of the shadow register is transferred to the active register on one of the following events as specified by the CMPCTL[PWMx_LOADAMODE] and CMPCTL[PWMx_LOADBMODE] register bits:

- CTR = PRD: Time-base counter equal to the period (TBCTR = TBPRD).
- CTR = Zero: Time-base counter equal to zero (TBCTR = 0x00).
- Both CTR = PRD or CTR = Zero.

Only the active register contents are used by the counter-compare submodule to generate events to be sent to the action-qualifier.

Note

Refer to Section 19.6.5 for valid configurations of CMPA/CMPB and LOADAMODE/LOADBMODE.

Immediate Load Mode:

Immediate load mode occurs when the CPU writes directly to the active register(PWMx_CMPA or PWMx_CMPB).

ADVANCE INFORMATION

Additional Comparators

The counter-compare submodule is responsible for generating two additional independent compare events based on two compare registers, which is fed to Event Trigger submodule:

1. CTR = PWMx_CMPC: Time-base counter equal to counter-compare C register (TBCTR = CMPC).
2. CTR = PWMx_CMPD: Time-base counter equal to counter-compare D register (TBCTR = CMPD).

The counter-compare registers CMPC and CMPD each have an associated shadow register. The memory address of the active register and the shadow register are separate, similar to PWMx_CMPA and PWMx_CMPB. The value in the active CMPC and CMPD register is compared to the time-base counter (TBCTR). When the values are equal, the counter compare module generates an event. The shadow and active mode functionality is identical to the functionality for PWMx_CMPA and PWMx_CMPB described above.

Global Load Support

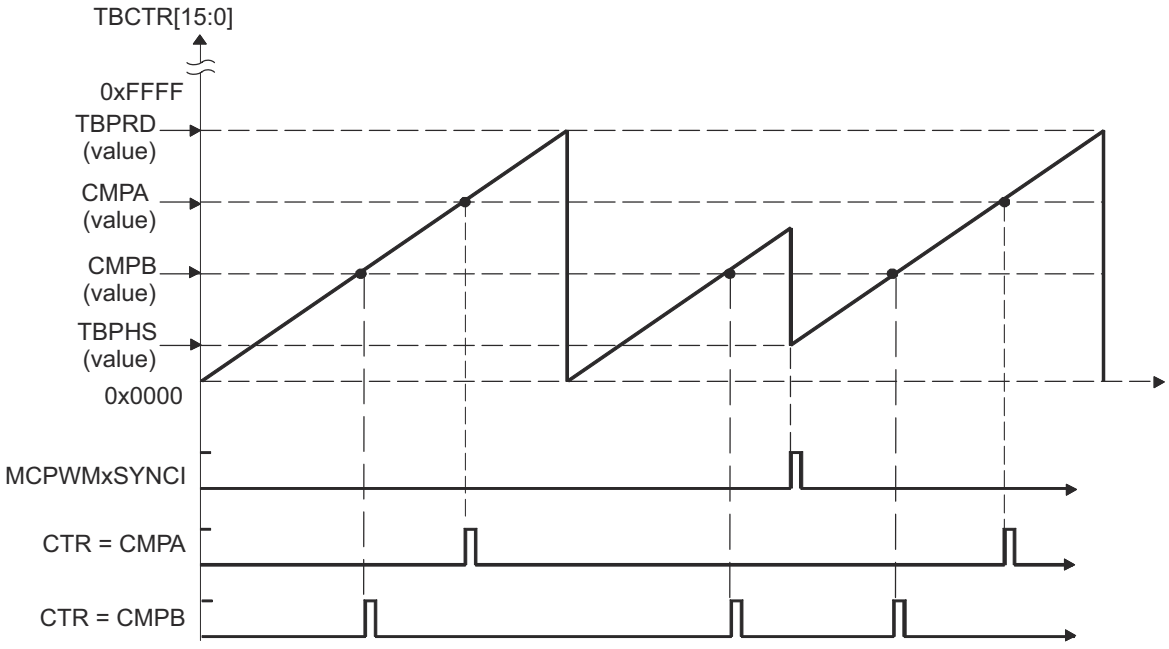
The global load control mechanism can also be used for all counter-compare registers by configuring the appropriate bits in the global load configuration register (GLDCTL). When the global load mode is selected the transfer of contents from shadow register to active register, for all registers that have a corresponding shadow register, occurs at the same event as defined by the configuration bits in the Global Shadow to Active Load Control Register (GLDCTL). The global load control mechanism is explained in Section 19.4.6 .

19.5.4 Count Mode Timing Waveforms

The counter-compare module can generate compare events in both count modes:

- Up-count mode: used to generate an asymmetrical PWM waveform.
- Up-down-count mode: used to generate a symmetrical PWM waveform.

To best illustrate the operation of the two count modes, the timing diagrams in Figure 19-14 through Figure 19-16 show when events are generated and how the MCPWMxSYNCI signal interacts.



Note

An MCPWMxSYNCI external synchronization event can cause a discontinuity in the TBCTR count sequence. This can lead to a compare event being skipped. This skipping is considered normal operation and must be taken into account.

Figure 19-14. Counter-Compare Event Waveforms in Up-Count Mode

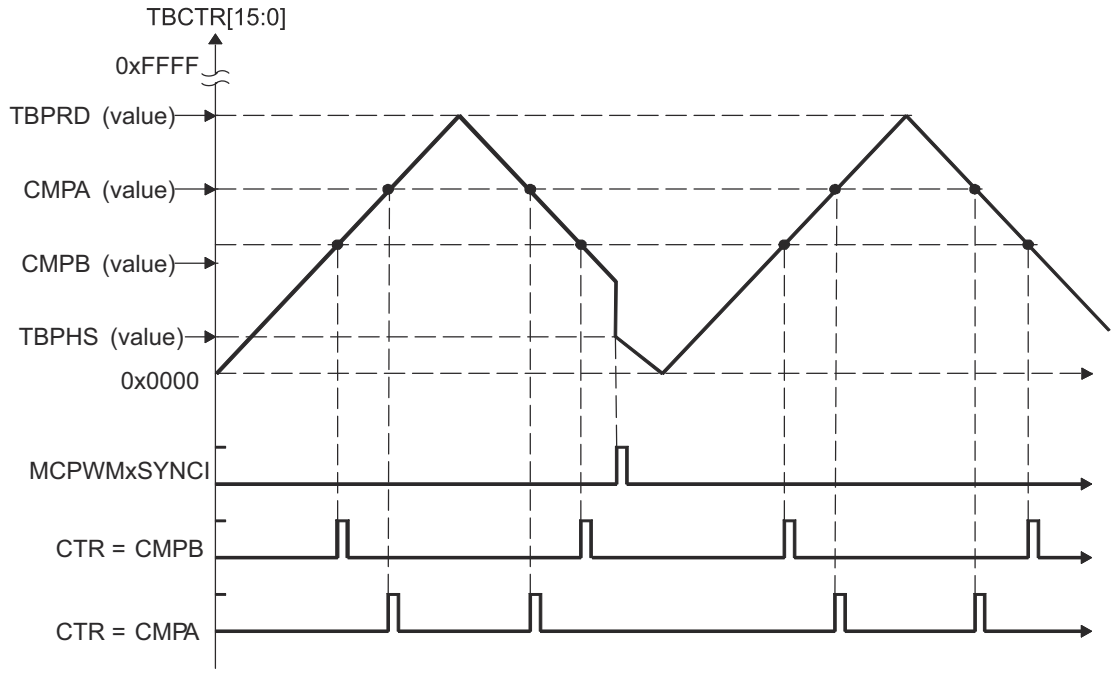


Figure 19-15. Counter-Compare Events In Up-Down Count Mode, TBCTL[PHSDIR = 0] Count Down On Synchronization Event

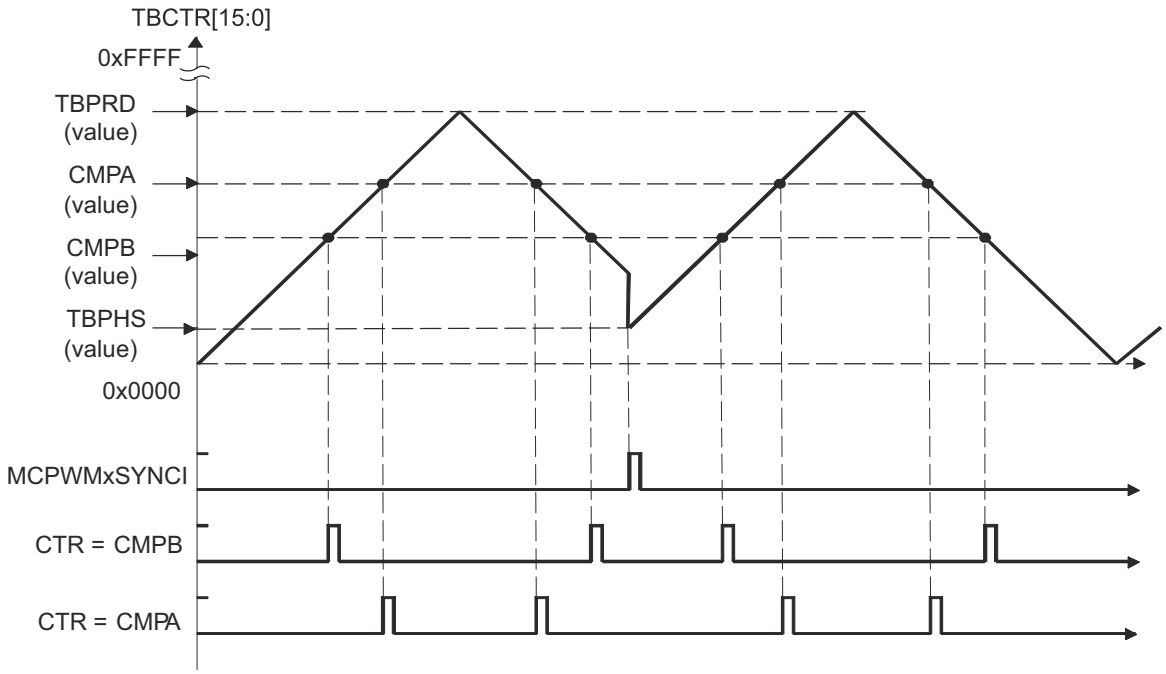


Figure 19-16. Counter-Compare Events In Up-Down Count Mode, TBCTL[PHSDIR = 1] Count Up On Synchronization Event

ADVANCE INFORMATION

19.6 Action-Qualifier (AQ) Submodule

The action-qualifier submodule defines the behavior of MCPWMx_yA and MCPWMx_yB outputs for each MCPWM TBCTR event. This module can trigger a rising or falling edge on the PWM outputs.

Figure 19-17 illustrates the action-qualifier submodule within the MCPWM.

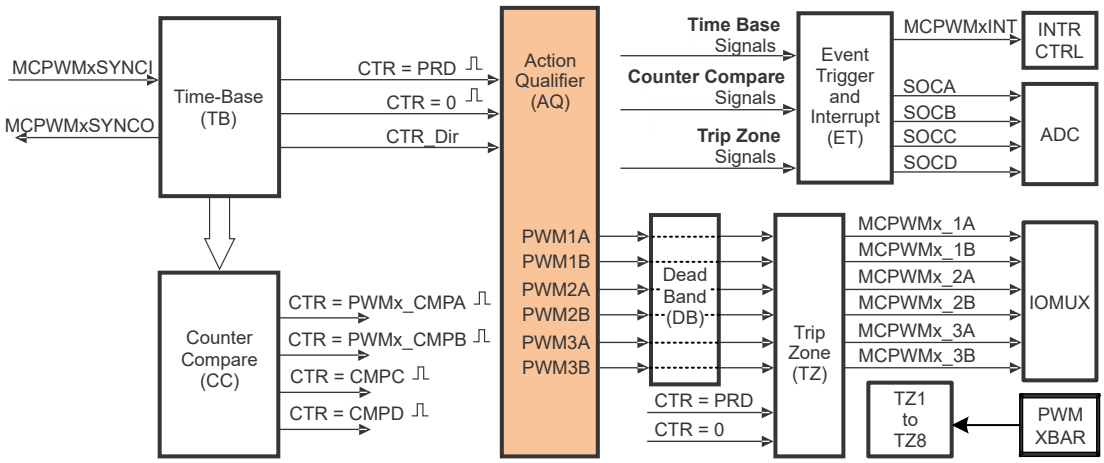


Figure 19-17. Action-Qualifier Submodule

19.6.1 Purpose of the Action-Qualifier Submodule

The action-qualifier submodule is responsible for the following:

- Qualifying and generating actions (set, clear, toggle) based on the following events:
  - CTR = PRD: Time-base counter equal to the period (TBCTR = TBPRD).
  - CTR = Zero: Time-base counter equal to zero (TBCTR = 0x00)
  - CTR = PWMx_CMPA: Time-base counter equal to the counter-compare A register (TBCTR = PWMx_CMPA)
  - CTR = PWMx_CMPB: Time-base counter equal to the counter-compare B register (TBCTR = PWMx_CMPB)
- Managing priority when these events occur concurrently
- Providing independent control of events when the time-base counter is increasing and when the time-base counter is decreasing

ADVANCE INFORMATION

19.6.2 Action-Qualifier Submodule Control and Status Register Definitions

The action-qualifier submodule operation is shown in Figure 19-18 and monitored by way of the registers in *MCPWM Registers*.

For convenience, the possible input events are summarized again in Table 19-4.

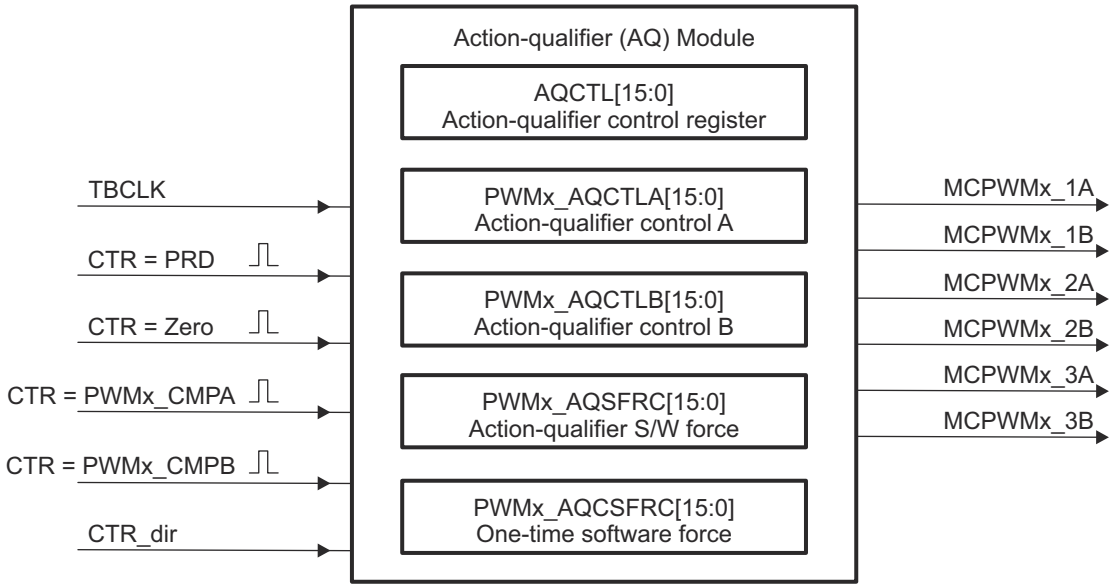


Figure 19-18. Action-Qualifier Submodule Inputs and Outputs

Table 19-4. Action-Qualifier Submodule Possible Input Events

Signal	Description	Registers Compared
CTR = PRD	Time-base counter equal to the period value	TBCTR = TBPRD
CTR = Zero	Time-base counter equal to 0	TBCTR = 0x00
CTR = CMPA	Time-base counter equal to the counter-compare A	TBCTR = PWMx_CMPA
CTR = CMPB	Time-base counter equal to the counter-compare B	TBCTR = PWMx_CMPB
Software forced event	Asynchronous event initiated by software	

The software forced action is a useful asynchronous event. This control is handled by the PWMx_AQSFRC and PWMx_AQOTSFRC register.

The action-qualifier submodule controls how the outputs MCPWMx_yA and MCPWMx_yB behave when a particular event occurs. The event inputs to the action-qualifier submodule are further qualified by the counter direction (up or down). This allows for independent action on outputs on both the count-up and count-down phases.

The possible actions imposed on outputs MCPWMx_yA and MCPWMx_yB are:

- **Set High:** Set output MCPWMx_yA or MCPWMx_yB to a high level.
- **Clear Low:** Set output MCPWMx_yA or MCPWMx_yB to a low level.
- **Toggle:** If MCPWMx_yA or MCPWMx_yB is currently pulled high, then pull the output low. If MCPWMx_yA or MCPWMx_yB is currently pulled low, then pull the output high.
- **Do Nothing:** Keep outputs MCPWMx_yA and MCPWMx_yB at same level as currently set. Although the "Do Nothing" option prevents an event from causing an action on the MCPWMx_yA and MCPWMx_yB outputs, this event can still trigger interrupts and ADC start of conversion. See the description in Section 19.9 for details.

Actions are specified independently for either output (MCPWMx_yA or MCPWMx_yB). Any or all events can be configured to generate actions on a given output. For example, both CTR = PWMx_CMPA and CTR = PWMx_CMPB can operate on output MCPWMx_yA. For each PWM pair "x", independent action qualifier events are specified. All qualifier actions are configured using the control registers found in *MCPWM Registers*.

For clarity, the illustrations in this chapter use a set of symbolic actions. These symbols are summarized in Figure 19-19. Each symbol represents an action as a marker in time. Some actions are fixed in time (zero and period) while the CMPA and CMPB actions are moveable and the time positions are programmed by way of the counter-compare A and B registers, respectively. To turn off or disable an action, use the "Do Nothing option"(the default at reset).

S/W force	TB Counter equals				Actions
	Zero	Comp A	Comp B	Period	
SW	Z	CA	CB	P	Do Nothing
SW	Z	CA	CB	P	Clear Lo
SW	Z	CA	CB	P	Set Hi
SW	Z	CA	CB	P	Toggle

Figure 19-19. Possible Action-Qualifier Actions for MCPWMx_yA and MCPWMx_yB Outputs

ADVANCE INFORMATION

19.6.3 Action-Qualifier Event Priority

It is possible for the MCPWM action qualifier to receive more than one event at the same time. In this case, events are assigned a priority by the hardware. The general rule is events occurring later in time have a higher priority and software forced events always have the highest priority. The event priority levels for up-down count mode are shown in Table 19-5. A priority level of 1 is the highest priority and level 10 is the lowest. The priority changes slightly depending on the direction of TBCTR.

Table 19-5. Action-Qualifier Event Priority for Up-Down Count Mode

Priority Level	Event If TBCTR is Incrementing TBCTR = Zero up to TBCTR = TBPRD	Event If TBCTR is Decrementing TBCTR = TBPRD down to TBCTR = 1
1 (Highest)	Software forced event	Software forced event
2	Counter equals CMPB on up-count (CBU)	Counter equals CMPB on down-count (CBD)
3	Counter equals CMPA on up-count (CAU)	Counter equals CMPA on down-count (CAD)
4 (Lowest)	Counter equals zero	Counter equals period (TBPRD)

Table 19-6 shows the action-qualifier priority for up-count mode. In this case, the counter direction is always defined as up; therefore, down-count events never are taken.

Table 19-6. Action-Qualifier Event Priority for Up-Count Mode

Priority Level	Event
1 (Highest)	Software forced event
2	Counter equal to period (TBPRD)
3	Counter equal to CMPB on up-count (CBU)
4	Counter equal to CMPA on up-count (CAU)
5 (Lowest)	Counter equal to Zero

It is possible to set the compare value greater than the period. In this case, the action takes place as shown in Table 19-7.

Table 19-7. Behavior if CMPA/CMPB is Greater than the Period

Counter Mode	Compare on Up-Count Event CAD/CBD	Compare on Down-Count Event CAD/CBD
Up-Count Mode	If CMPA/CMPB ≤ TBPRD period, then the event occurs on a compare match (TBCTR = CMPA or CMPB). If CMPA/CMPB > TBPRD, then the event does not occur.	Never occurs.
Up-Down Count Mode	If CMPA/CMPB < TBPRD and the counter is incrementing, the event occurs on a compare match (TBCTR = CMPA or CMPB). If CMPA/CMPB is ≥ TBPRD, the event occurs on a period match (TBCTR = TBPRD).	If CMPA/CMPB < TBPRD and the counter is decrementing, the event occurs on a compare match (TBCTR = CMPA or CMPB). If CMPA/CMPB ≥ TBPRD, the event occurs on a period match (TBCTR = TBPRD).

19.6.4 AQCTLA and AQCTLB Shadow Mode Operations

Shadow Mode:

Shadowing is always enabled for PWMx_AQCTLA and PWMx_AQCTLB. To utilize shadowing, the user must write to the corresponding shadow registers (PWMx_AQCTLAS and PWMx_AQCTLBS).

The content of the shadow register is transferred to the active register on one of the following events as specified by the AQCTL[PWMx_LDAQAMODE] and AQCTL[PWMx_LDAQBMODE] register bits:

- CTR = PRD: Time-base counter equal to the period (TBCTR = TBPRD).
- CTR = Zero: Time-base counter equal to zero (TBCTR = 0x00)
- Both CTR = PRD and CTR = Zero

Global Load Support

Global load control mechanism can also be used for PWMx_AQCTLA and PWMx_AQCTLB registers when global load is enabled via the global load configuration register (GLDCTL). When global load mode is selected, the transfer of contents from the shadow registers to active registers occurs at the same event as defined by the configuration bits in the Global Shadow to Active Load Control Register (GLDCTL). The global load control mechanism is explained in Section 19.4.6 .

Immediate Load Mode:

To utilize immediate load mode, global load must be disabled and the user must write to the active register instead of the shadow register. See Figure 19-20 and Figure 19-21.

Note

Shadow to Active Load of Action Qualifier Output A/B Control Register [PWMx_AQCTLA and PWMx_AQCTLB] on CMPA = 0 or CMPB = 0 boundary

If the Counter-Compare A Register (PWMx_CMPA) or Counter-Compare B Register (PWMx_CMPB) is set to a value of 0 and the action qualifier action on PWMx_AQCTLA and PWMx_AQCTLB is configured to occur in the same instant as a shadow to active load (that is, CMPA = 0 and AQCTLA shadow to active load on TBCTR = 0 using AQCTL register LDAQAMODE and LDAQAMODE bits), then both events can enter contention depending on the timing of the write to PWMx_CMPA or PWMx_CMPB, resulting in a missed action qualifier event for one MCPWM period. It is recommended to use a Non-Zero Counter-Compare when using Shadow to Active Load of Action Qualifier Output A/B Control Register on TBCTR = 0 boundary.

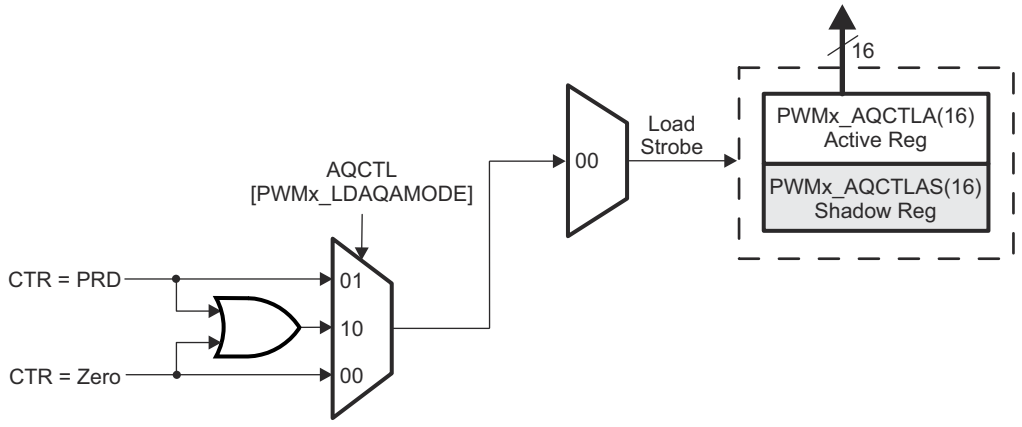


Figure 19-20. AQCTL[PWMx_LDAQAMODE]

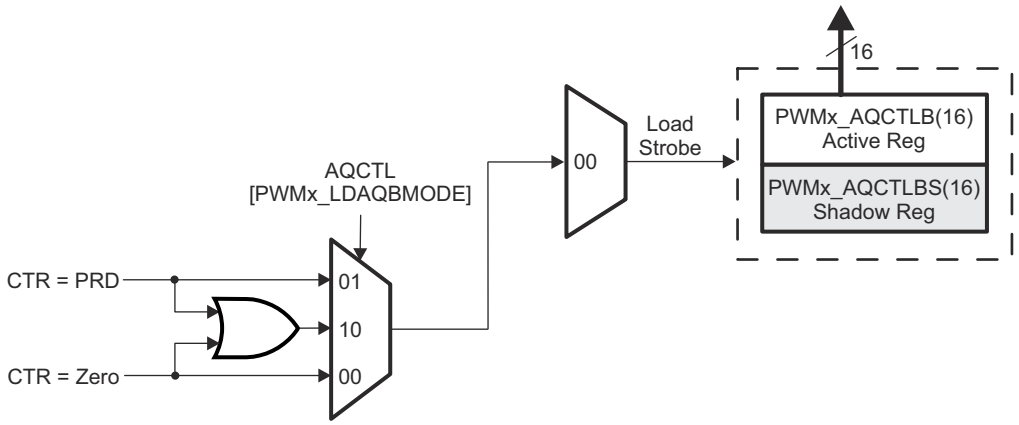


Figure 19-21. AQCTL[PWMx_LDAQBMODE]

ADVANCE INFORMATION

19.6.5 Configuration Requirements for Common Waveforms

Note

The waveforms in this chapter show the behavior of the MCPWMs for a static compare register value. In a running system, the active compare registers (PWMx_CMPA and PWMx_CMPB) are typically updated from the respective shadow registers (PWMx_CMPAS and PWMx_CMPBS) once every period. Specify when the update takes place: either when the time-base counter reaches zero or when the time-base counter reaches the period. There are some cases when the action based on the new value can be delayed by one period or the action based on the old value can take effect for an extra period. Some PWM configurations avoid this situation. These include, but are not limited to, the following:

Use up-down count mode to generate a symmetric PWM:

- If loading PWMx_CMPA/PWMx_CMPB on zero, then use PWMx_CMPA/PWMx_CMPB values greater than or equal to 1.
- If loading PWMx_CMPA/PWMx_CMPB on period, then use PWMx_CMPA/PWMx_CMPB values less than or equal to TBPRD - 1.

  This means there is always a pulse of at least 1 TBCLK cycle in a PWM period which, when very short, tend to be ignored by the system.

Use up-down count mode to generate an asymmetric PWM:

- To achieve 50%-0% asymmetric PWM, use the following configuration: load PWMx_CMPA/PWMx_CMPB on period and use the period action to clear the PWM and a compare-up action to set the PWM. Modulate the compare value from 0 to TBPRD to achieve 50%-0% PWM duty.

When using up-count mode to generate an asymmetric PWM:

- To achieve 0-100% asymmetric PWM, load PWMx_CMPA/PWMx_CMPB on TBPRD. When PWMx_CMPA/PWMx_CMPB is not loaded on TBCTR = PRD, boundary conditions can occur depending on the timing of the write and the value written to PWMx_CMPA/PWMx_CMPB. Use the Zero action to set the PWM and a compare-up action to clear the PWM. Modulate the compare value from 0 to TBPRD + 1 to achieve 0-100% PWM duty.

When using up-count mode to generate an asymmetric PWM with dead-band enabled:

- To achieve 0%-100% PWM, use the following configuration: When the PWMx_CMPA value is too close to 0 or PRD such that the following conditions are met (CMPX < Deadband) or (CMPX > PRD – Deadband), the actions specified by the AQCTL register for CMPX do not take effect. To avoid this, the AQCTL settings must be altered under these conditions only to generate either high or low pulses for both CAU or CAD events (both set or both clear). Make sure that this software update is occurring synchronous to the PWM carrier cycle, and shadow mode is enabled.

When using up-down count mode to generate an asymmetric PWM with dead-band enabled:

- To achieve 0%-100% PWM, use the following configuration: When the PWMx_CMPA value is too close to 0 or PRD such that the following conditions are met (CMPX < Deadband/2) or (CMPX > PRD – (Deadband)/2), the actions specified by the AQCTL register for CMPX do not take effect. To avoid this, the AQCTL settings must be altered under these conditions only to generate either high or low pulses for both CAU or CAD events (both set or both clear). Make sure that this software update is occurring synchronous to the PWM carrier cycle, and shadow mode is enabled.

See *Using Enhanced Pulse Width Modulator (ePWM) Module for 0-100% Duty Cycle Control*.

Figure 19-22 shows how a symmetric PWM waveform can be generated using the up-down-count mode of the TBCTR. In this mode, 0%-100% DC modulation is achieved by using equal compare matches on the up count and down count portions of the waveform. In the example shown, PWMx_CMPA is used to make the comparison. When the counter is incrementing, the PWMx_CMPA match pulls the PWM output high. Likewise when the counter is decrementing, the compare match pulls the PWM signal low. When PWMx_CMPA = 0, the

PWM signal is high for the entire period giving a 100% duty waveform. When PWMx_CMPA = TBPRD, the PWM signal is low achieving 0% duty.

When using this configuration in practice, if loading PWMx_CMPA/PWMx_CMPB on zero, then use PWMx_CMPA/PWMx_CMPB values greater than or equal to 1. If loading PWMx_CMPA/PWMx_CMPB on period, then use PWMx_CMPA/PWMx_CMPB values less than or equal to TBPRD - 1. This means there is always a pulse of at least 1 TBCLK cycle in a PWM period which, when very short, tend to be ignored by the system.

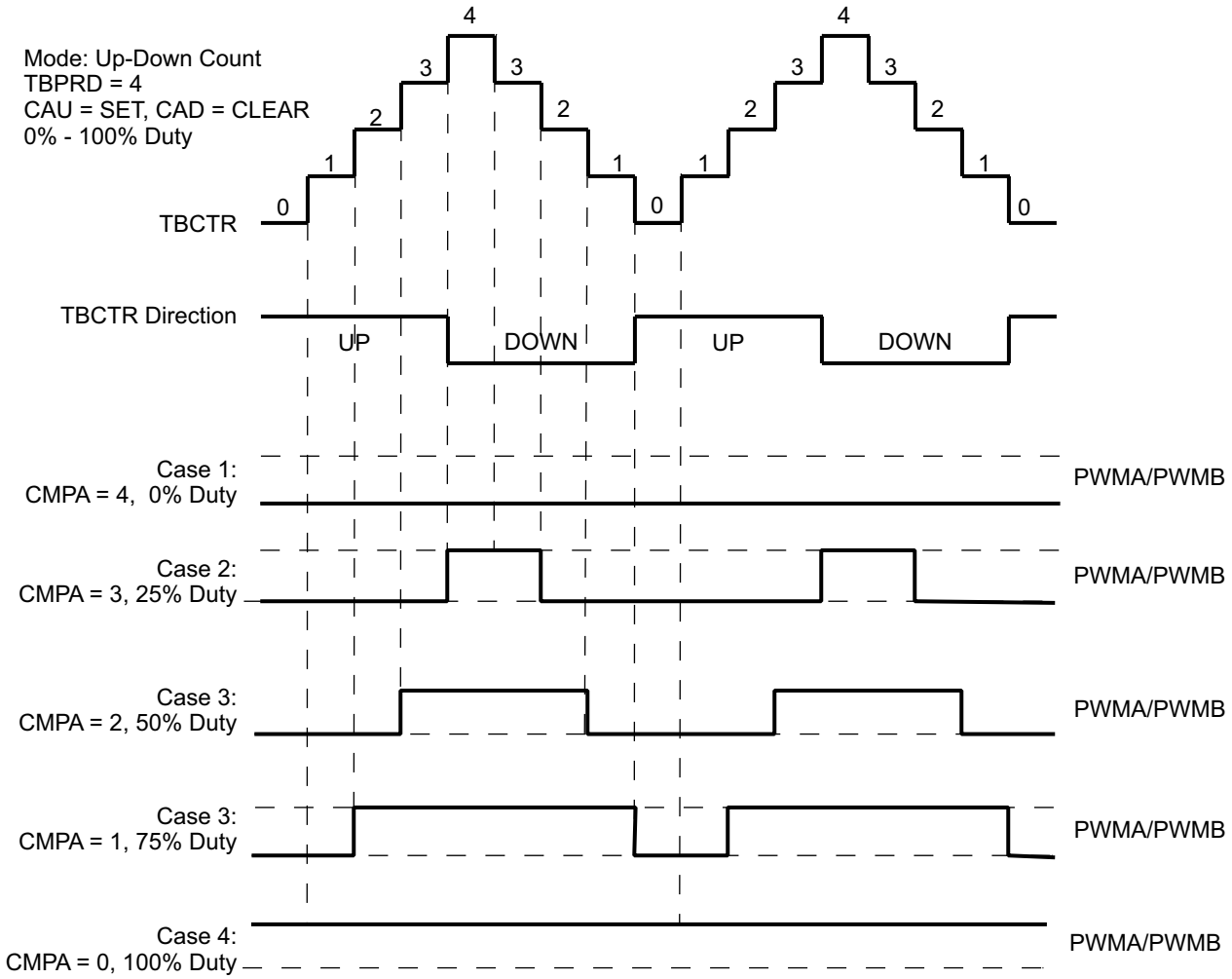
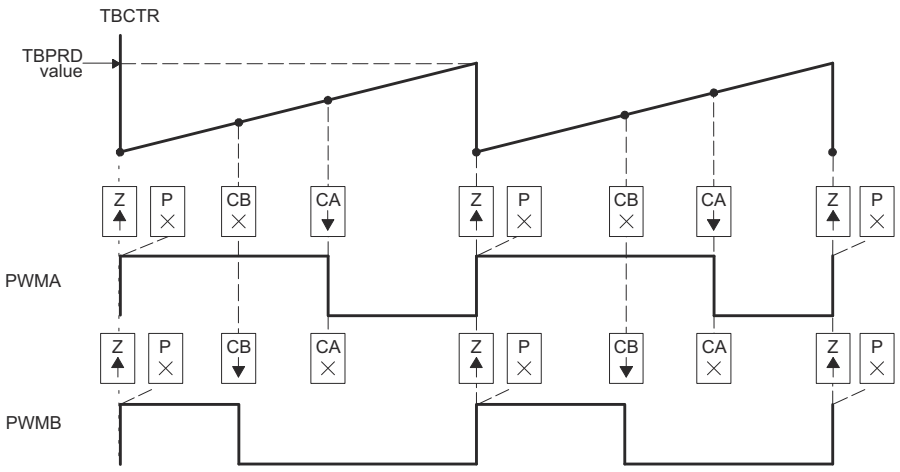


Figure 19-22. Up-Down Count Mode Symmetrical Waveform

ADVANCE INFORMATION

The following PWM waveforms show some common action-qualifier configurations. Some conventions used in the figures and examples are as follows:

- TBPRD, PWMx_CMPA, and PWMx_CMPB refer to the value written in the respective registers. The active register, not the shadow register, is used by the hardware.
- CMPx, refers to either PWMx_CMPA or PWMx_CMPB.
- MCPWMx_yA and MCPWMx_yB refer to the output signals from MCPWMx, Channel pair y.
- Up-Down means count-up and count-down mode, Up means up-count mode.
- Sym = Symmetric, Asym = Asymmetric.



Note

PWM period = (TBPRD + 1) × T_{TBCLK}

Note

Duty modulation for MCPWMx_yA is set by PWMy_CMPA, and is active high (that is, high time duty proportional to PWMy_CMPA).

Note

Duty modulation for MCPWMx_yB is set by PWMy_CMPB and is active high (that is, high time duty proportional to PWMy_CMPB).

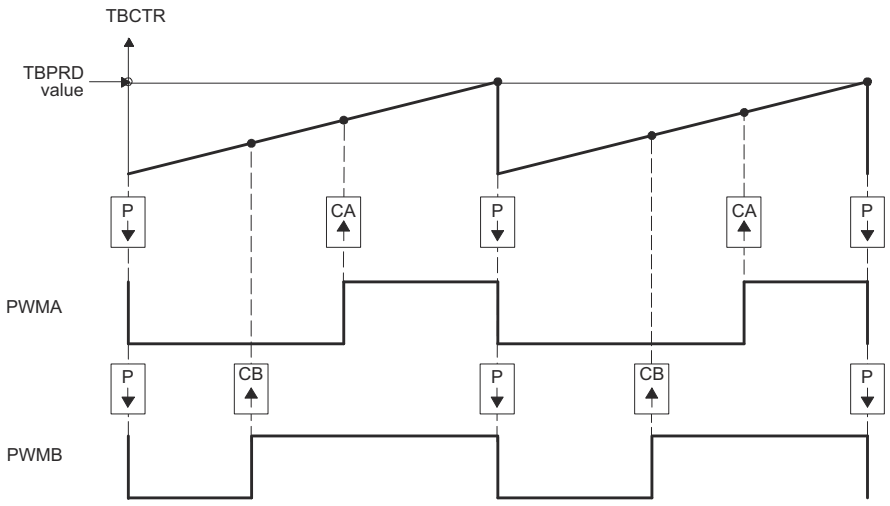
Note

The "Do Nothing" actions (X) are shown for completeness, but are not shown on subsequent diagrams.

Note

Actions at zero and period, although appearing to occur concurrently, are actually separated by one TBCLK period. TBCTR wraps from period to 0000.

Figure 19-23. Up, Single Edge Asymmetric Waveform, with Independent Modulation on PWMxA and PWMxB—Active High



Note

PWM period = (TBPRD + 1) × T_{TBCLK}

Note

Duty modulation for MCPWMx_yA is set by PWMy_CMPA, and is active low (that is, the low time duty is proportional to PWMy_CMPA).

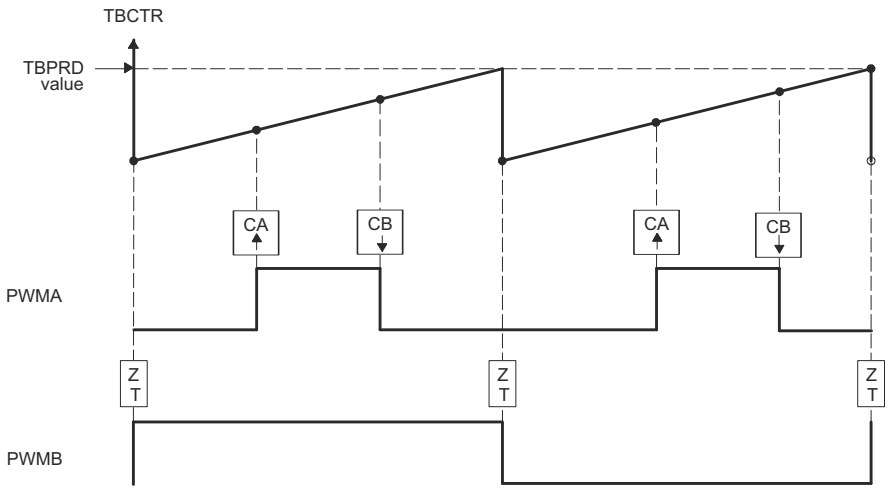
Note

Duty modulation for MCPWMx_yB is set by PWMy_CMPB and is active low (that is, the low time duty is proportional to PWMy_CMPB).

Note

Actions at zero and period, although appearing to occur concurrently, are actually separated by one TBCLK period. TBCTR wraps from period to 0000.

Figure 19-24. Up, Single Edge Asymmetric Waveform with Independent Modulation on PWMxA and PWMxB—Active Low



Note

PWM frequency = 1/((TBPRD + 1) × T_{TBCLK})

Note

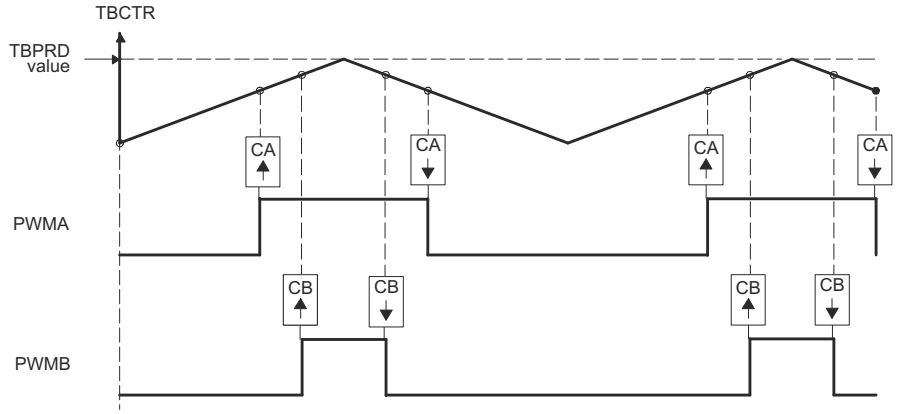
Pulse can be placed anywhere within the PWM cycle (0000 - TBPRD)

Note

High time duty proportional to (CMPB - CMPA)

Figure 19-25. Up-Count, Pulse Placement Asymmetric Waveform With Independent Modulation on PWMxA

ADVANCE INFORMATION



Note

PWM period = 2 x TBPRD × T_{TBCLK}

Note

Duty modulation for MCPWMx_yA is set by PWMy_CMPA, and is active low (that is, the low time duty is proportional to PWMy_CMPA).

Note

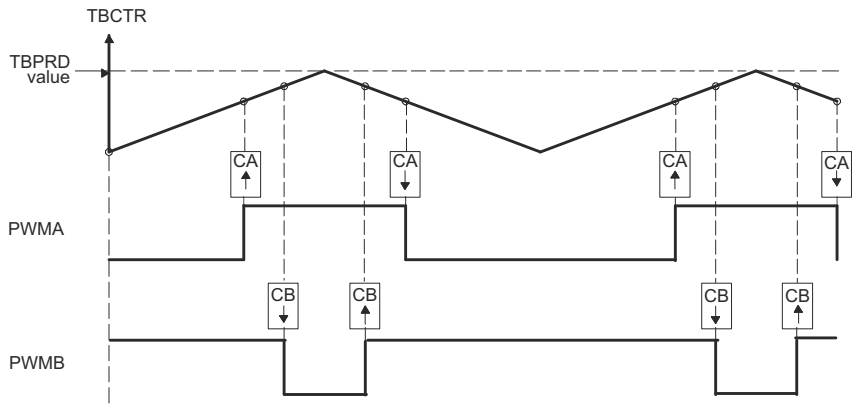
Duty modulation for MCPWMx_yB is set by PWMy_CMPB and is active low (that is, the low time duty is proportional to PWMy_CMPB).

Note

Outputs MCPWMx_yA and MCPWMx_yB can drive independent power switches.

Figure 19-26. Up-Down Count, Dual-Edge Symmetric Waveform, with Independent Modulation on PWMxA and PWMxB — Active Low

ADVANCE INFORMATION



Note

PWM period = 2 × TBPRD × T_{TBCLK}

Note

Duty modulation for MCPWMx_yA is set by PWMy_CMPA, and is active low, that is, low time duty proportional to PWMy_CMPA.

Note

Duty modulation for MCPWMx_yB is set by PWMy_CMPB and is active high, that is, high time duty proportional to PWMy_CMPB.

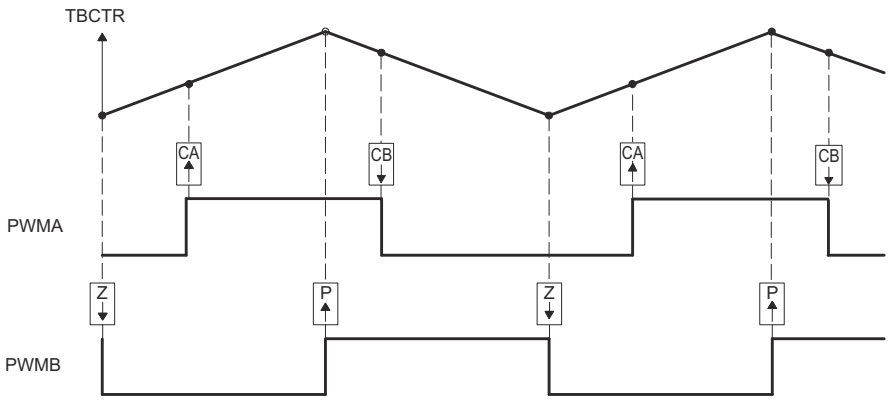
Note

Outputs MCPWMx can drive upper/lower (complementary) power switches.

Note

Dead-band = CMPB - CMPA (fully programmable edge placement by software). Note the dead-band module is also available if the more classical edge delay method is required.

Figure 19-27. Up-Down Count, Dual-Edge Symmetric Waveform, with Independent Modulation on PWMxA and PWMxB — Complementary



Note

PWM period = 2 × TBPRD × TBCLK

Note

Rising edge and falling edge can be asymmetrically positioned within a PWM cycle. This allows for pulse placement techniques.

Note

Duty modulation for MCPWMx_yA is set by PWMy_CMPA and PWMy_CMPB.

Note

Low time duty for MCPWMx_yA is proportional to (PWMy_CMPA + PWMy_CMPB).

Note

To change this example to active high, PWMy_CMPA and PWMy_CMPB actions need to be inverted (that is, Clear on PWMy_CMPA, Set on PWMy_CMPB).

Note

Duty modulation for MCPWMx_yB is fixed at 50% (utilizes spare action resources for MCPWMx_yB).

Figure 19-28. Up-Down Count, Dual-Edge Asymmetric Waveform, with Independent Modulation on PWMxA—Active Low

19.7 Dead-Band Generator (DB) Submodule

Figure 19-29 illustrates the dead-band submodule within the MCPWM.

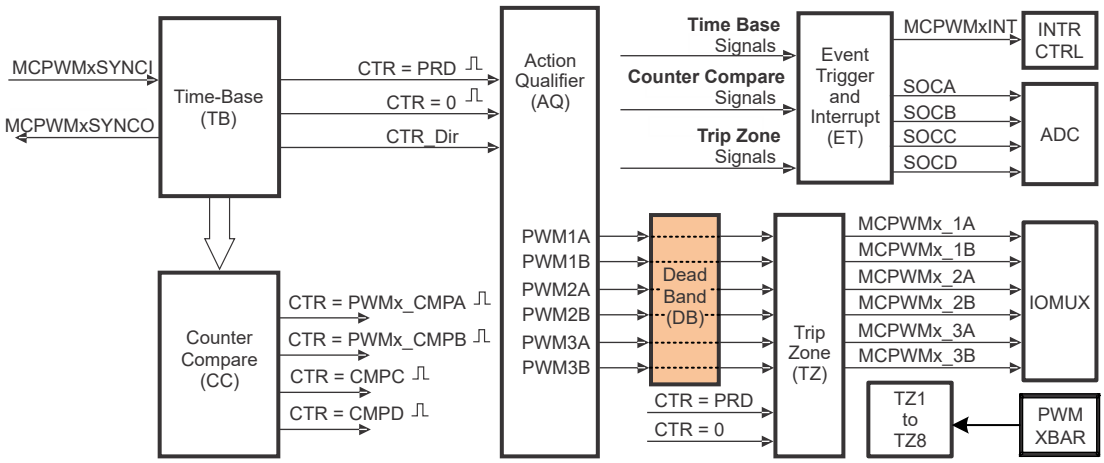


Figure 19-29. Dead-band Generator Submodule

19.7.1 Purpose of the Dead-Band Submodule

The action-qualifier (AQ) module section discussed how the AQ module can generate the required dead band by having full control over edge placement using both the PWMx_CMPA and PWMx_CMPB resources of the MCPWM module. However, if the more classical edge delay-based dead band with polarity control is required, then the dead-band submodule described here must be used.

The key functions of the dead-band module are:

- Generating appropriate signal pairs (MCPWMx_yA and MCPWMx_yB) with dead-band relationship from a single MCPWMx_yA input
- Programming signal pairs for:
  - Active high (AH)
  - Active low (AL)
  - Active high complementary (AHC)
  - Active low complementary (ALC)
- Adding programmable delay to rising edges (RED)
- Adding programmable delay to falling edges (FED)
- Can be totally bypassed from the signal path (note dotted lines in diagram)

ADVANCE INFORMATION

19.7.2 Dead-Band Submodule Additional Operating Modes

On the type-1 ePWM, RED can appear on one channel output and FED can appear on the other channel output.

The dead-band module for MCPWM is unchanged from a type-4 ePWM with the following exceptions:

- Addition of memory-mapped shadow registers for DBCTL, DBFED, and DBRED.
- High-resolution dead-band and half-cycle clocking mode are removed.
- The shadow register for DBCTL is removed.

The following list shows the distinct difference between type-1 and type-4 ePWM modules with respect to dead-band operating modes:

- By adding S6, S7, and S8 in Figure 19-30, RED and FED can appear on both the A-channel and B-channel outputs. Additionally, both RED and FED together can be applied to either the A-channel or B-channel outputs to allow B-channel phase shifting with respect to the A-channel.

Note

Phase shifting B-channel with respect to the A-channel using the dead-band submodule additional operating modes has limitations with respect to the choice of RED and FED delay with respect to the operating duty cycle of the PWMxA and PWMxB outputs.

- The dead-band counters have also been increased to 14 bits
- Dead-band and dead-band high-resolution registers are now shadowed

Note

Cannot have both RED and FED together applied to both PWMxA and PWMxB. RED and FED together can be applied only to either OutA OR OutB.

Phase shifting B-channel with respect to the A-channel: When PWMxB is derived from PWMxA using the DEDB_MODE bit and by delaying rising edge and falling edge by the phase shift amount. When the duty cycle value on PWMxA is less than this phase shift amount, PWMxA's falling edge has precedence over the delayed rising edge for PWMxB. Make sure the duty cycle value of the current waveform applied to the dead-band module is greater than the required phase shift amount.

The type-4 action qualifier and dead-band outputs of the ePWM module are delayed by one TBCLK cycle in comparison to the type-2 ePWM module, although the type-4 ePWM behavior is the same as the type-3 ePWM. Both PWMA and PWMB signals are delayed under all circumstances.

Shadow Mode:

The shadow mode for the DBRED and DBFED registers is always enabled; however, write to the memory-mapped shadow registers (DBREDS, DBFEDS) to utilize shadow loading. Writing to the active register results in an immediate load.

If writing to the shadow register, the content of the shadow register is transferred to the active register on one of the following events as specified by the DBCTL [LOADREDMODE] and DBCTL [LOADFEDMODE] register bits:

- CTR = PRD: Time-base counter equal to the period (TBCTR = TBPRD).
- CTR = Zero: Time-base counter equal to zero (TBCTR = 0x00)
- Both CTR = PRD and CTR = Zero

Global Load Support

Global load control mechanism can also be used for DBRED and DBFED registers by configuring the appropriate bits in the global load configuration register (GLDCTL). When global load mode is selected the transfer of contents from shadow register to active register, for all registers that have a shadow register, occurs at the same event as defined by the configuration bits in the Global Shadow to Active Load Control Register (GLDCTL). The Global load control mechanism is explained in Section 19.4.6 .

Note

When DBRED/DBFED active is loaded with a new shadow value while DB counters are counting, the new DBRED/DBFED value only affects the NEXT PWMx edge and not the current edge.

A dead-band value of zero cannot be used when the Global Shadow to Active Load is set to occur at CTR = ZERO. Similarly, a dead-band value of PRD cannot be used when the Global Shadow to Active Load is set to occur at CTR = PRD.

19.7.3 Operational Highlights for the Dead-Band Submodule

The configuration options for the dead-band submodule are shown in Figure 19-30.

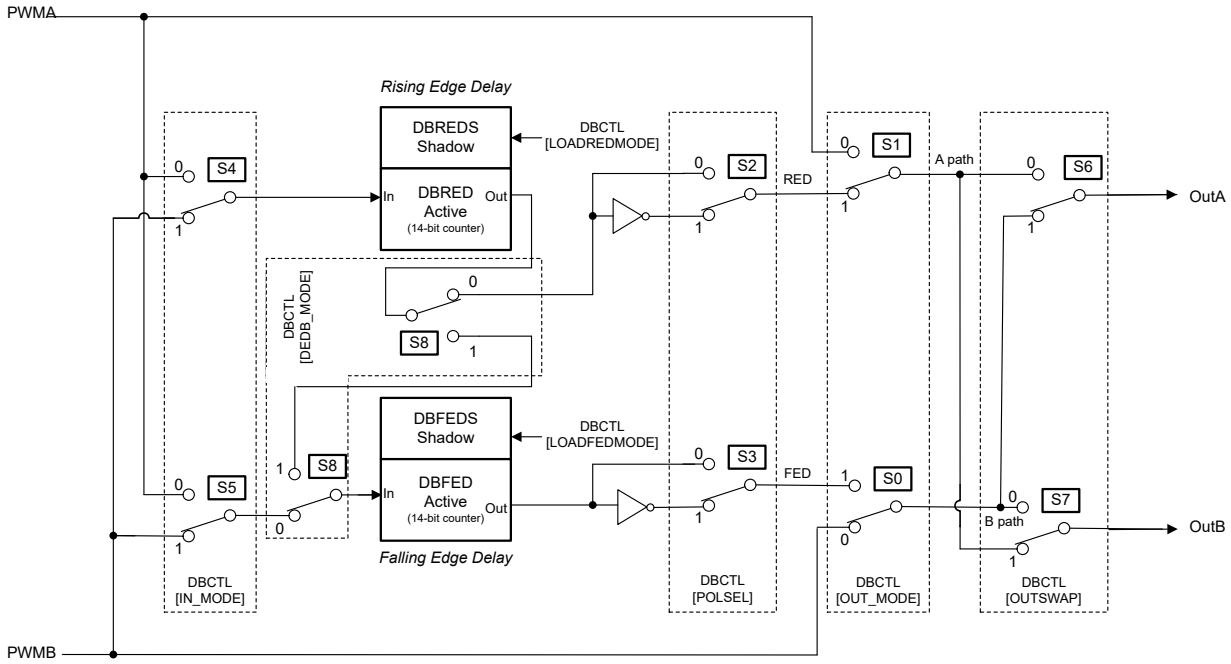


Figure 19-30. Configuration Options for the Dead-Band Submodule

Note

The Dead-Band (DB) submodule is disabled

Note

Dead-band settings are configured by the same register bits for all PWM pairs. Separate dead-band settings for individual PWM pairs within a single MCPWM module is **not supported**.

Although all combinations are supported, not all are typical usage modes. Table 19-8 documents some classical dead-band configurations. These modes assume that the DBCTL[IN_MODE] is configured such that MCPWMx_yA In is the source for both falling-edge and rising-edge delay. Enhanced, or non-traditional modes can be achieved by changing the input signal source. The modes shown in Table 19-8 fall into the following categories:

- **Mode 1: Bypass both falling-edge delay (FED) and rising-edge delay (RED):** Allows the user to fully disable the dead-band submodule from the PWM signal path. **(DEFAULT)**
- **Mode 2-5: Classical Dead-Band Polarity Settings:** These represent typical polarity configurations that can address all the active-high and active-low modes required by available industry power switch gate drivers. The waveforms for these typical cases are shown in Figure 19-31. Note that to generate equivalent waveforms to Figure 19-31, configure the action-qualifier submodule to generate the signal as shown for MCPWMx_yA.
- **Mode 6: Bypass rising-edge delay (RED) and Mode 7: Bypass falling-edge delay (FED):** Finally the last two entries in Table 19-8 show combinations where either the falling-edge delay (FED) or rising-edge delay (RED) blocks are bypassed.

Figure 19-31 shows waveforms for typical cases where 0% < duty < 100%.

ADVANCE INFORMATION

Table 19-8. Classical Dead-Band Operating Modes

Mode	Mode Description	DBCTL[POLSEL]		DBCTL[OUT_MODE]	
		S3	S2	S1	S0
1	MCPWMx_yA and MCPWMx_yB Passed Through (No Delay)	X	X	0	0
2	Active High Complementary (AHC)	1	0	1	1
3	Active Low Complementary (ALC)	0	1	1	1
4	Active High (AH)	0	0	1	1
5	Active Low (AL)	1	1	1	1
6	MCPWMx_yA Out = MCPWMx_yA In (No Delay) MCPWMx_yB Out = MCPWMx_yA In with Falling-Edge Delay	0 or 1	0 or 1	0	1
7	MCPWMx_yA Out = MCPWMx_yA In with Rising-Edge Delay MCPWMx_yB Out = MCPWMx_yB In with No Delay	0 or 1	0 or 1	1	0

Table 19-9. Additional Dead-Band Operating Modes

Mode Description	DBCTL[DEDB-MODE]	DBCTL[OUTSWAP]	
	S8	S6	S7
MCPWMx_yA and MCPWMx_yB signals are as defined by OUT-MODE bits.	0	0	0
MCPWMx_yA = A-path as defined by OUT-MODE bits. MCPWMx_yB = A-path as defined by OUT-MODE bits (rising-edge delay or delay-bypassed A-signal path)	0	0	1
MCPWMx_yA = B-path as defined by OUT-MODE bits (falling-edge delay or delay-bypassed B-signal path) MCPWMx_yB = B-path as defined by OUT-MODE bits	0	1	0
MCPWMx_yA = B-path as defined by OUT-MODE bits (falling-edge delay or delay-bypassed B-signal path) MCPWMx_yB = A-path as defined by OUT-MODE bits (rising-edge delay or delay-bypassed A-signal path)	0	1	1
Rising-edge delay applied to MCPWMx_yA / MCPWMx_yB as selected by S4 switch (IN-MODE bits) on A signal path only. Falling-edge delay applied to MCPWMx_yA / MCPWMx_yB as selected by S5 switch (IN-MODE bits) on B signal path only.	0	X	X
Rising-edge delay and falling-edge delay applied to source selected by S4 switch (IN-MODE bits) and output to B signal path only.(1)	1	X	X

(1) When this bit is set to 1, the user can always either set OUT_MODE bits such that Apath = InA or set OUTSWAP bits such that MCPWMx_yA=Bpath. Otherwise, MCPWMx_yA is invalid.

ADVANCE INFORMATION

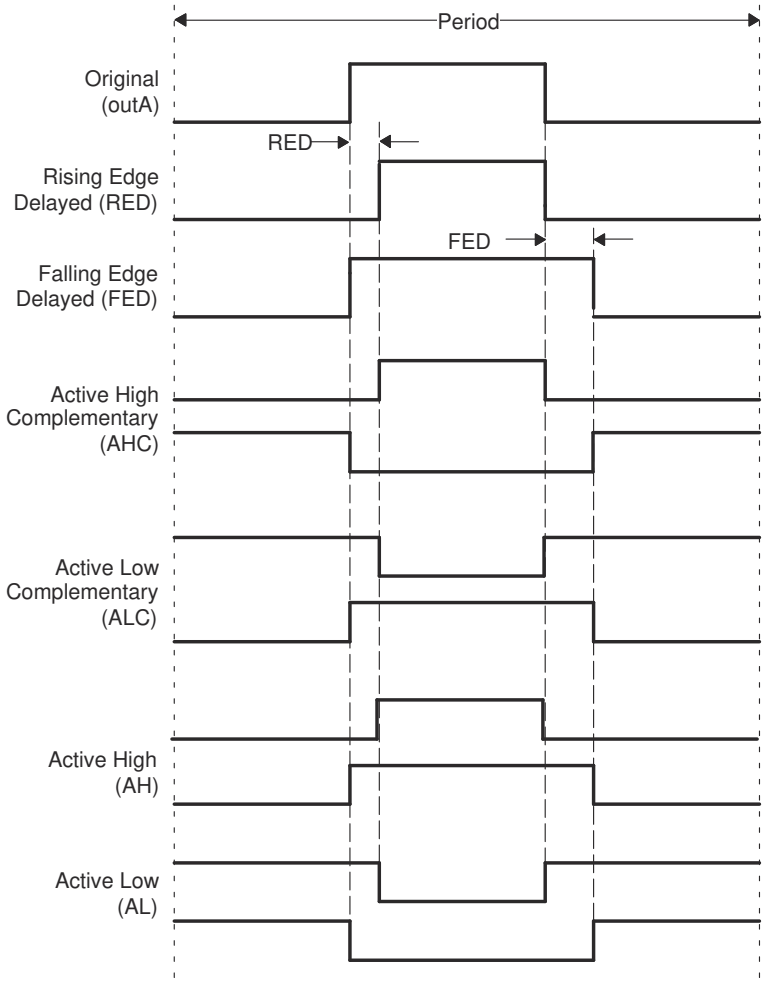


Figure 19-31. Dead-Band Waveforms for Typical Cases (0% < Duty < 100%)

The dead-band submodule supports independent values for rising-edge (RED) and falling-edge (FED) delays. The amount of delay is programmed using the DBRED and DBFED registers. These are 10-bit registers and the value represents the number of TBCLK (time-base clock) pulses by which a signal edge is delayed. For example, the formula to calculate falling-edge-delay and rising-edge-delay is:

FED = DBFED × T_{TBCLK}

RED = DBRED × T_{TBCLK}

Where T_{TBCLK} is the period of TBCLK, the prescaled version of PWMCLK. The TBCLK.CLKDIV register bitfield supports divider options from /1 to /32768.

Table 19-10. TBCLK Prescale Options

TBCTL.CLKDIV Binary Value (Decimal)	Prescale/Divider Value
0000 (0) **(Default)**	TBCLK = PWMCLK/1
0001 (1)	TBCLK = PWMCLK/2
0010 (2)	TBCLK = PWMCLK/4
0011 (3)	TBCLK = PWMCLK/8
0100 (4)	TBCLK = PWMCLK/16

Table 19-10. TBCLK Prescale Options (continued)

TBCTL.CLKDIV Binary Value (Decimal)	Prescale/Divider Value
0101 (5)	TBCLK = PWMCLK/32
0110 (6)	TBCLK = PWMCLK/64
0111 (7)	TBCLK = PWMCLK/128
1000 (8)	TBCLK = PWMCLK/256
1001 (9)	TBCLK = PWMCLK/512
1010 (10)	TBCLK = PWMCLK/1024
1011 (11)	TBCLK = PWMCLK/2048
1100 (12)	TBCLK = PWMCLK/4096
1101 (13)	TBCLK = PWMCLK/8192
1110 (14)	TBCLK = PWMCLK/16384
1111 (15)	TBCLK = PWMCLK/32768

The final dead-band delay values for some common TBCLK options are shown in the table below for convenience.

Table 19-11. Dead-Band Delay Values in µs as a Function of DBFED and BRED (PWMCLK = 200MHz)

Dead-Band Value	Dead-Band Delay (µs)		
DBFED, DBRED	TBCLK = PWMCLK/1	TBCLK = PWMCLK /2	TBCLK = PWMCLK/4
1	0.005	0.01	0.02
5	0.025	0.05	0.10
10	0.050	0.10	0.20
100	0.500	1.00	2.00
200	1.00	2.00	4.00
400	2.00	4.00	8.00
500	2.50	5.00	10.00
600	3.00	6.00	12.00
700	3.50	7.00	14.00
800	4.00	8.00	16.00
900	4.50	9.00	18.00
1000	5.00	10.00	20.00

19.8 Trip-Zone (TZ) Submodule

Each MCPWM module is connected to eight $\overline{TZn}$ signals ($\overline{TZ1}$ to $\overline{TZ8}$). All eight $\overline{TZn}$ signals are sourced from the PWM X-BAR inherited from the type-4 ePWM. Refer to the *Crossbar (X-BAR)* chapter to see what signals can be routed to the MCPWM module to be used as a trip source. These signals indicate external fault or trip conditions, and the MCPWM outputs can be programmed to respond accordingly when faults occur.

Figure 19-32 illustrates the trip-zone submodule within the MCPWM.

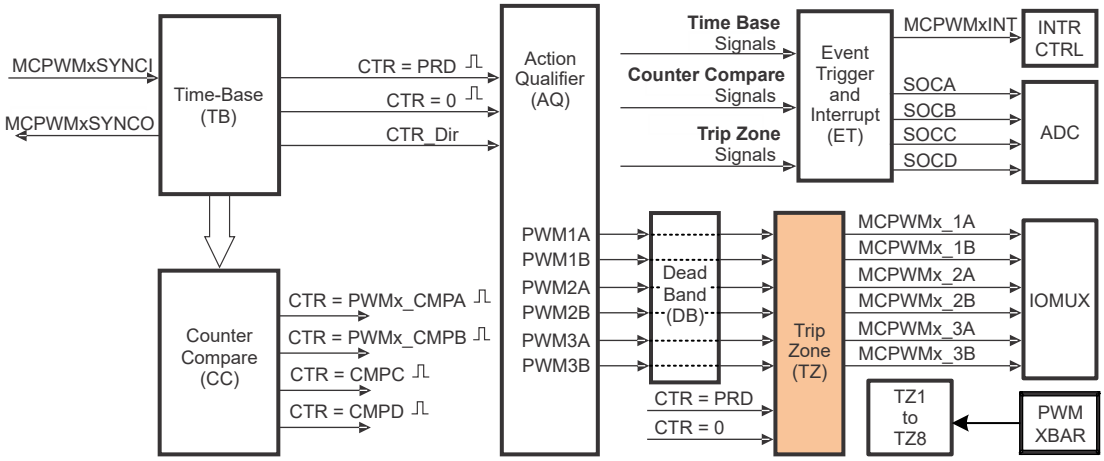


Figure 19-32. Trip-Zone Submodule

19.8.1 Purpose of the Trip-Zone Submodule

The key functions of the trip-zone submodule are:

- Trip inputs $\overline{TZ1}$ to $\overline{TZ8}$ can be flexibly mapped to any MCPWM module.
- Upon a fault condition, outputs MCPWMx_yA and MCPWMx_yB can be forced to one of the following:
  - High
  - Low
  - High-impedance
  - No action taken
- Support for one-shot trip (OST) for major short circuits or over-current conditions.
- Support for cycle-by-cycle tripping (CBC) for current limiting operation.
- Each trip-zone input can be allocated to either one-shot or cycle-by-cycle operation.
- Interrupt generation is possible on any trip-zone input.
- The trip-zone submodule can be fully bypassed if the trip-zone submodule is not required.

19.8.2 Operational Highlights for the Trip-Zone Submodule

The following sections describe the operational highlights and configuration options for the trip-zone submodule.

The trip-zone signals $\overline{TZ1}$ to $\overline{TZ8}$ (also collectively referred to as $\overline{TZn}$) are active-low input signals. When one of these signals goes low, the indication is that a trip event has occurred. Each MCPWM module can be individually configured to ignore or use each of the trip-zone signals. Note that the trip zone settings for all PWM pairs are configured by the same register bits, separate trip-zone settings for separate PWM pairs within a single MCPWM module is not supported. The trip-zone signals used by a particular MCPWM module is configured by the TZSEL register for that specific MCPWM module. The trip-zone signals can be digitally filtered within the GPIO MUX block. A minimum of 3 × TBCLK low pulse width on $\overline{TZn}$ inputs is sufficient to trigger a fault condition on the MCPWM module. If the pulse width is less than this, the trip condition cannot be latched by CBC or OST latches. The asynchronous trip makes sure that if clocks are missing for any reason, the outputs can still be tripped by a valid event present on $\overline{TZn}$ inputs. The GPIOs or peripherals must be appropriately configured.

Each $\overline{TZn}$ input can be individually configured to provide either a cycle-by-cycle or one-shot trip event for an MCPWM module. This configuration is determined by the TZSEL[CBCn], and TZSEL[OSTn] control bits (where n corresponds to the trip input), respectively.

- **Cycle-by-Cycle (CBC):** When a cycle-by-cycle trip event occurs, the action specified in the TZCTL[TZA] and TZCTL[TZB] bits is carried out immediately on the MCPWMx_yA and MCPWMx_yB outputs. The same trip actions are taken on all PWM pairs, denoted by "y". Table 19-12 lists some of the possible actions.

  Additionally, when a cycle-by-cycle trip event occurs, the cycle-by-cycle trip event flag (INTFLAG[CBC]) is set and a PWMx interrupt is generated if the INTEN[CBC] bit is set. A corresponding flag for the specific event that caused the CBC interrupt is also set in the TZCBCOSTFLAG register, which can be cleared by writing to the corresponding flag in TZCBCOSTCLR.

  The specified condition on the inputs is automatically cleared based on the selection made with INTCLR[CBC] if the trip event is no longer present. Therefore, in this mode, the trip event is cleared or reset every PWM cycle. The INTFLG[CBC] and TZCBCOSTFLAG flag bits remain set until the flag bits are manually cleared by writing to the INTCLR[CBC] and TZCBCOSTCLR flag bits. If the cycle-by-cycle trip event is still present when the INTFLG[CBC] and TZCBCOSTFLAG register bits are cleared, then these bits are again immediately set.
- **One-Shot (OST):** When a one-shot trip event occurs, the action specified in the TZCTL[TZA] and TZCTL[TZB] bits is carried out immediately on the MCPWMx_yA and MCPWMx_yB output. The same trip actions are taken on all PWM pairs, denoted by "y". Table 19-12 lists some of the possible actions.

  Additionally, when a one-shot trip event occurs, the one-shot trip event flag (INTFLG[OST]) is set and a PWMx interrupt is generated if the INTEN[OST] bit is set.. A corresponding flag for the event that caused the OST event is also set in register TZCBCOSTFLAG. The one-shot trip condition must be cleared manually by writing to the INTCLR[OST] bit. If desired, the TZCBCOSTFLAG register bit can also be cleared by manually writing to the corresponding bit in the TZCBCOSTCLR register.

Note

Clear the INTFLAG and TZCBCOSTFLAG flags after making sure that the TRIPIN source of the OST has become inactive. Otherwise, if interrupts are enabled, depending on when the flags are cleared, an OST interrupt can occur where the OST flags are zero.

The action taken when a trip event occurs can be configured individually for the A outputs and B outputs of the MCPWM module, but the action is the same across all A outputs and the same across all B outputs. Trip actions are specified by the TZCTL register. Some of the possible actions, shown in Table 19-12, can be taken on a trip event.

Table 19-12. Possible Actions On a Trip Event

TZCTL[TZx] Register Bitfield Settings	MCPWMx_yA and MCPWMx_yB	Comment
0,0	High-Impedance	Tripped
0,1	Force to High State	Tripped
1,0	Force to Low State	Tripped
1,1	No Change	Do Nothing. No change is made to the output.

Trip-Zone Configurations

Scenario A:
A one-shot trip event on $\overline{TZ1}$ pulls the MCPWM1_yA, MCPWM1_yB outputs low.

- Configure the MCPWM1 registers as follows:
  - TZSEL[OST1] = 1: enables $\overline{TZ1}$ as a one-shot event source for MCPWM1
  - TZCTL[TZA] = 2: MCPWM1_yA is forced low on a trip event.
  - TZCTL[TZB] = 2: MCPWM1_yB is forced low on a trip event.

Scenario B:
A cycle-by-cycle event on $\overline{TZ1}$ or $\overline{TZ6}$ puts MCPWM1_yA into a high-impedance state, MCPWM1_yB is unaffected.

- Configure the MCPWM1 registers as follows:
  - TZSEL[CBC1] = 1: enables $\overline{TZ1}$ as a cycle-by-cycle event source for MCPWM1
  - TZSEL[CBC6] = 1: enables $\overline{TZ6}$ as a cycle-by-cycle event source for MCPWM1
  - TZCTL[TZA] = 0: MCPWM1_yA is put into a high-impedance state on a trip event.
  - TZCTL[TZB] = 3: MCPWM1_yB ignores the trip event.

Note

When configuring the GPIOs and INPUT X-BAR/MCPWM X-BAR options, be aware that a change in the X-BAR input selections can cause an unwanted event. Therefore, set up the GPIO and X-BAR input configurations before enabling the MCPWM Trip-Zone. If a requirement is to change the GPIO/X-BAR configurations while the MCPWM Trip-Zone is enabled, the user can turn off the TRIPs by clearing the TZSEL register and reconfiguring the TRIP selection (TZSEL) after the INPUT X-BAR selection is changed.

19.8.3 Generating Trip Event Interrupts

Figure 19-33 illustrates the trip-zone submodule control logic.

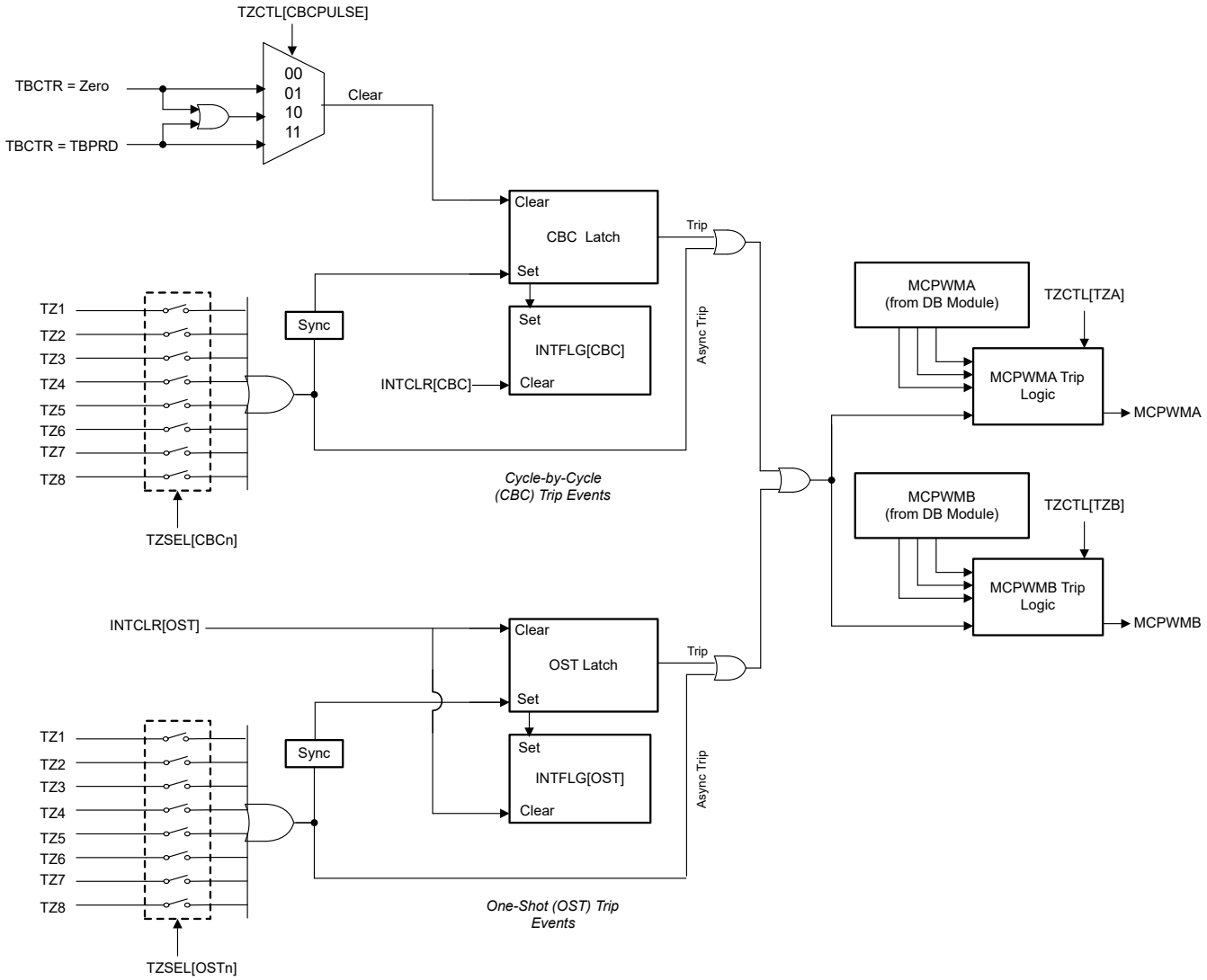


Figure 19-33. Trip-Zone Submodule Mode Control Logic

19.9 Event-Trigger (ET) Submodule

The key functions of the event-trigger submodule are:

- Receives event inputs generated by the time-base and counter-compare submodules
- Uses the time-base direction information for up/down event qualification
- Uses prescaling logic to issue interrupt requests and ADC start of conversion at:
  - Every event
  - Every second event
  - ...
  - Up to every seventh event
- Uses TZ signals to interrupt the CPU on a trip zone event
- Provides full visibility of event generation using event counters and flags
- Allows software forcing of interrupts

The event-trigger submodule manages the events generated by the time-base submodule, counter-compare submodule, and trip-zone submodule to generate an interrupt to the CPU or a start of conversion pulse to the ADC when a selected event occurs.

Figure 19-34 illustrates the event-trigger submodule within the MCPWM.

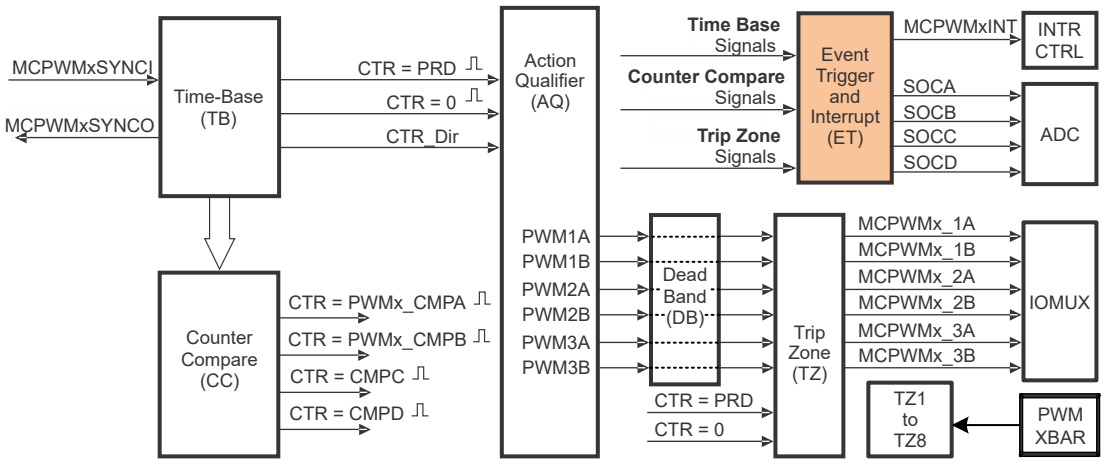


Figure 19-34. Event-Trigger Submodule

19.9.1 Operational Overview of the MCPWM Event-Trigger Submodule

The event-trigger submodule monitors various event conditions (shown as inputs on the left side of Figure 19-35) and can be configured to prescale these events before issuing an Interrupt request or an ADC start of conversion. The event-trigger prescaling logic can issue interrupt requests and ADC start-of-conversion at:

- Every event
- Every second event
- Up to every seventh event

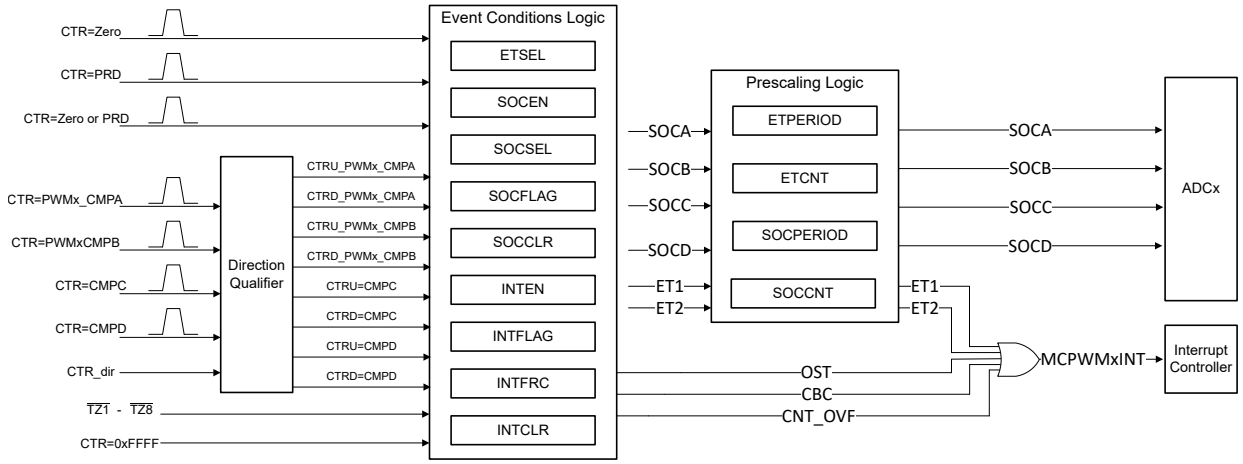


Figure 19-35. Event-Trigger Submodule Showing Event Inputs and Prescaled Outputs

- ETSEL: This selects which of the possible events trigger events ET1 or ET2.
- ETPERIOD: This programs the event prescaling options mentioned above.
- ETCNT: This indicates how many events have occurred if the prescaling feature is used.
- INTEN: This enables interrupt sources for the MCPWM.
- INTFLAG: These are flag bits indicating whether an E1x or trip event has occurred.
- INTCLR: These bits allow clearing the flag bits in the INTFLAG register using software.
- INTFRC: These bits allow software forcing of an event. Useful for debugging or software intervention.
- SOCEN: This enables ADC start of conversion events.
- SOCSEL: This selects the trigger sources for the ADC start of conversion events.
- SOCPERIOD: This programs the event prescaling options for SOC events.
- SOCCNT: This indicates how many SOC events have been generated when using the prescaling feature.
- SOCFLAG: This indicates which SOC events have been generated.
- SOCCLR: This clears the SOCFLAG bits when written to using software.

A more detailed look at how the various register bits interact with the interrupt and ADC start-of-conversion logic are shown in Figure 19-36 and Figure 19-37.

Figure 19-36 shows the event-trigger interrupt generation logic. The interrupt-period (ETPERIOD[ETx_PERIOD]) bits specify the number of events required to cause an interrupt pulse to be generated. The choices available are:

- Do not generate an interrupt.
- Generate an interrupt on every event.
- Generate an interrupt on every second event.
- Generate an interrupt on up to every seventh event.

The event that can cause an interrupt is configured by the interrupt enable(INTEN) register and event trigger selection (ETSEL) register. The event can be one of the following:

- Time-base counter equal to zero (TBCTR = 0x00).

- Time-base counter equal to period (TBCTR = TBPRD).
- Time-base counter equal to zero or period (TBCTR = 0x00 || TBCTR = TBPRD).
- Time-base counter equal to the compare A register (CMPA) when the timer is incrementing.
- Time-base counter equal to the compare A register (CMPA) when the timer is decrementing.
- Time-base counter equal to the compare B register (CMPB) when the timer is incrementing.
- Time-base counter equal to the compare B register (CMPB) when the timer is decrementing.
- Time-base counter equal to the compare C register (CMPC) when the timer is incrementing.
- Time-base counter equal to the compare C register (CMPC) when the timer is decrementing.
- Time-base counter equal to the compare D register (CMPD) when the timer is incrementing.
- Time-base counter equal to the compare D register (CMPD) when the timer is decrementing.
- One-shot trip event.
- Cycle-by-cycle trip event.

The number of events that have occurred can be read from the interrupt event counter ETCNT[ETx_CNT] register bits based off of the selection made using ETSEL. The specified event increments the ETCNT[ETx_CNT] bits until the bits reach the value specified by ETPERIOD[ETx_PERIOD]. When ETPS[INTCNT] = ETPS[INTPRD], the counter stops counting and the counter output is set. The counter is cleared when an INTFLAG[ETx] event is generated.

When ETCNT[ETx_CNT] reaches ETPERIOD[ETx_PERIOD], the following behavior occurs:

If interrupts are enabled, INTEN[ETx] = 1 and the interrupt flag is clear,INTFLAG[ETx] = 0, then an interrupt pulse is generated and the interrupt flag is set, INTFLAG[ETx] = 1. The counter is reset and begins counting events again

- If interrupts are disabled, INTEN[ETx] = 0, or the interrupt flag is set, INTFLAG[ETx] = 1, the counter stops counting events when the counter reaches the period value ETCNT[ETx_CNT] = ETPERIOD[ETx_PERIOD].
- If interrupts are enabled, but the interrupt flag is already set, then the counter holds the output high until the INTFLAG[ETx] flag is cleared. This allows for one interrupt to be pending while one is serviced.

Writing a 0 or a value that is less than the current ETCNT value to the ETPERIOD bits results in the counter going to an undefined state.

The previous definition means that an interrupt on every event up to 7 events if using the ETCNT and ETPERIOD can be generated.

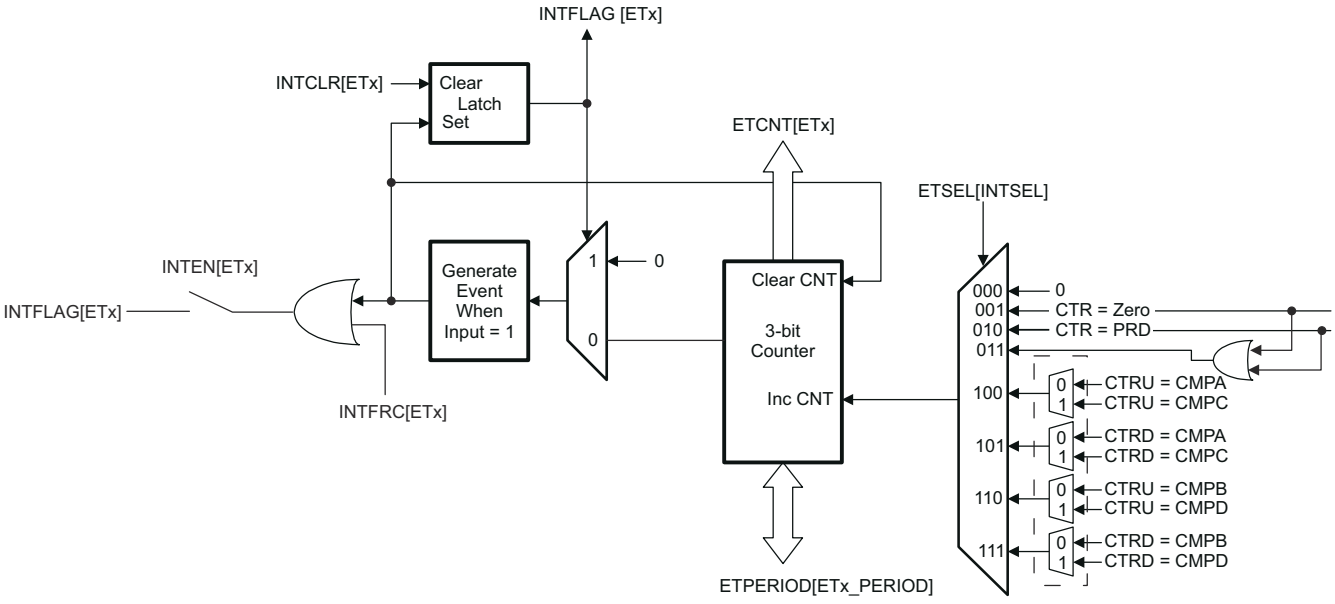


Figure 19-36. Event-Trigger Interrupt Generator

Figure 19-37 shows the operation of the event-trigger's start-of-conversion (SOCx) pulse generator. The SOCCNT[SOCx_CNT] counters and SOCPERIOD[SOCx_PERIOD] period values behave similarly to the interrupt generator except that the pulses are continuously generated. That is, the pulse flag SOCFLAG[SOCx] is latched when a pulse is generated, but the interrupt generator does not stop further pulse generation. The enable and disable bit SOCEN[SOCx_ENABLE] stops pulse generation, but input events can still be counted until the period value is reached as with the interrupt generation logic. The event that triggers an SOCx pulse can be configured separately in the SOCSEL[SOCx_SEL] bits. The possible events are the same events that can be specified for the interrupt generation logic.

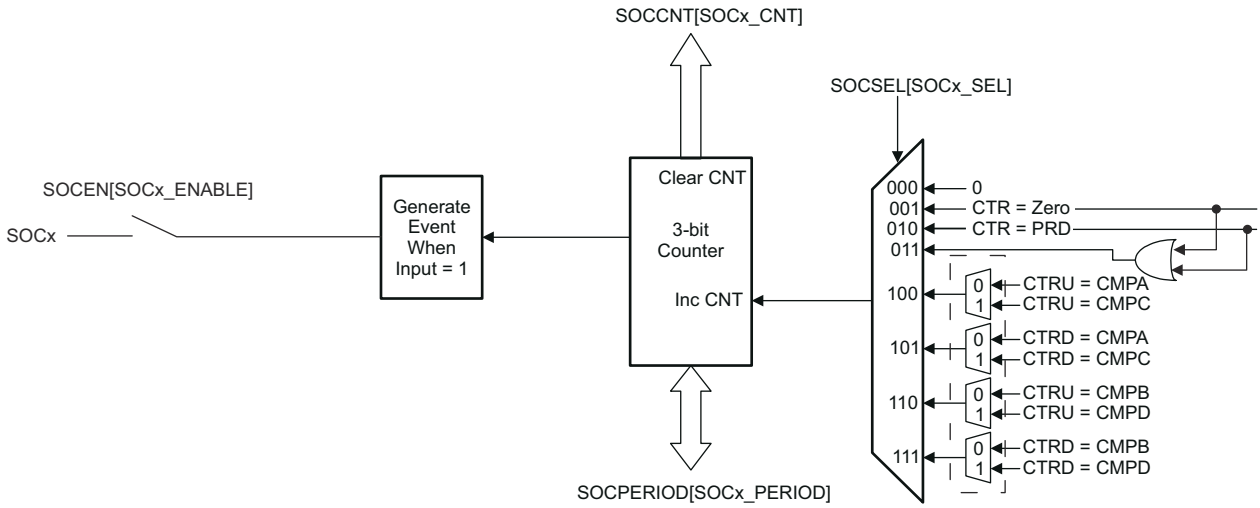


Figure 19-37. Event-Trigger Start-of-Conversion (SOC) Pulse Generator

19.10 PWM Crossbar (X-BAR)

Figure 19-38 shows the architecture of the PWM Crossbar (X-BAR). This module enables selection of various trigger sources into any of the dedicated MCPWM trips inputs.

Note

Refer to the *Crossbar (X-BAR)* chapter for more information on the X-BAR modules, including X-BAR flags.

ADVANCE INFORMATION

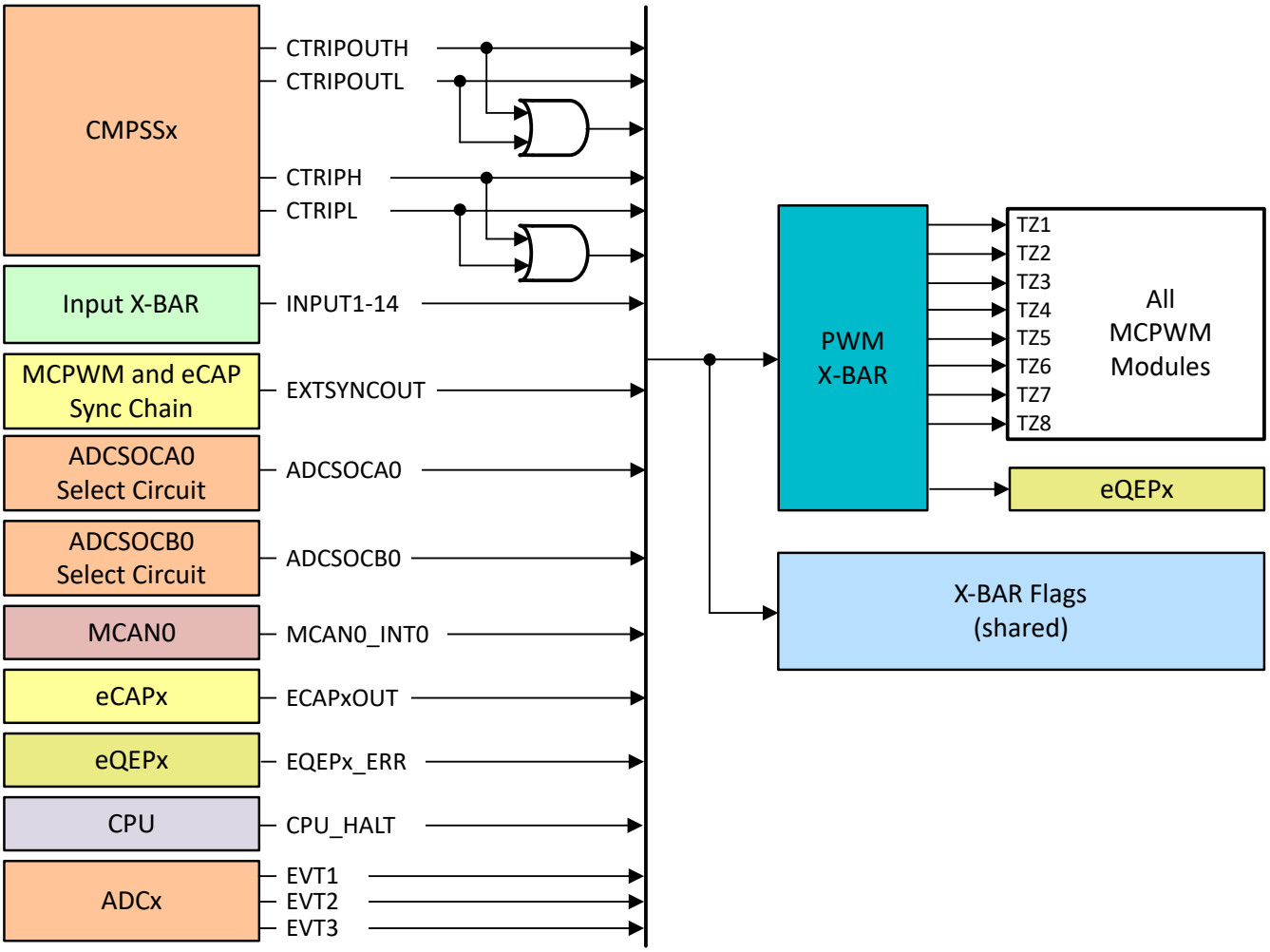


Figure 19-38. PWM X-BAR

The PWM X-BAR has eight outputs that are routed to each MCPWM module. Figure 19-39 represents the architecture of a single output but the output is identical to the architecture of all of the other outputs.

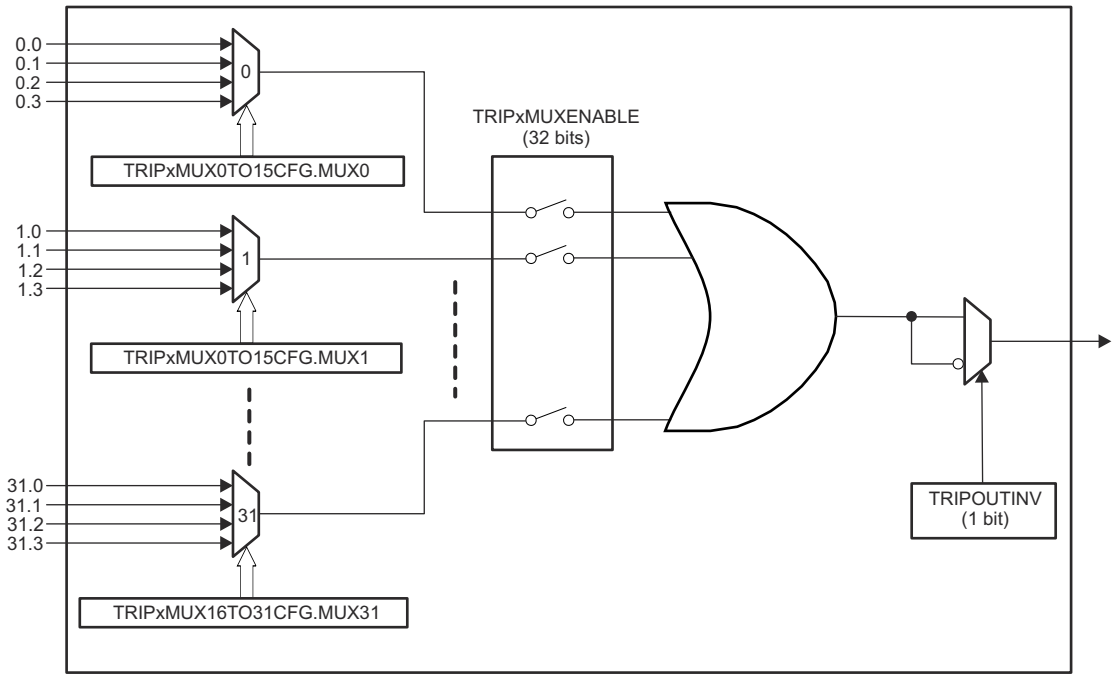


Figure 19-39. PWM X-BAR Architecture - Single Output

First, determine the signals that can be passed to the MCPWM by referencing the PWM X-Bar Mux Configuration Table (see Section 22.2.1.1). Select up to one signal per mux (32 total muxes) for each TRIPx output. Select the inputs to each mux with the TRIPxMUX0TO15CFG and TRIPxMUX16TO31CFG registers. To pass any signal through to the MCPWM, enable the mux in the TRIPxMUXENABLE register. All muxes that are enabled are logically ORed before being passed on to the respective TRIPx signal on the MCPWM. To optionally invert the signal, use the TRIPOUTINV register.

Note

All unused and reserved positions are tied to 0.

19.11 MCPWM Registers

This Section describes the MCPWM Registers.

19.11.1 MCPWM Base Address Table

Table 19-13. MCPWM Base Address Table

Bit Field Name		DriverLib Name	Base Address
Instance	Structure		
Mcpwm0Regs	MCPWM_6CH_REGS	MCPWM0	0x4001_0000
Mcpwm1Regs	MCPWM_6CH_REGS	MCPWM1	0x4001_1000
Mcpwm2Regs	MCPWM_6CH_REGS	MCPWM2	0x4001_2000
Mcpwm3Regs	MCPWM_6CH_REGS	MCPWM3	0x4001_3000
Mcpwm4Regs	MCPWM_6CH_REGS	MCPWM4	0x4001_4000

ADVANCE INFORMATION

19.11.2 MCPWM_6CH_REGS Registers

Table 19-14 lists the memory-mapped registers for the MCPWM_6CH_REGS registers. All register offset addresses not listed in Table 19-14 should be considered as reserved locations and the register contents should not be modified.

Table 19-14. MCPWM_6CH_REGS Registers

Offset	Acronym	Register Name	Write Protection	Section
0h	REVISION	IP revision id register		Go
10h	TBCTL	Time base control register		Go
14h	TBPRD	Time base period register		Go
18h	TBPRDS	Time base period shadow register		Go
1Ch	TBPHS	Time base phase offset register		Go
20h	TBSTS	Time base status register		Go
24h	TBSTSCLR	Time base status clear register		Go
28h	TBCTR	Time base counter register		Go
30h	CMPCTL	Counter compare control register		Go
40h	CMPC	Counter compare C register		Go
44h	CMPD	Counter compare D register		Go
48h	CMPCS	Counter compare C shadow register		Go
4Ch	CMPDS	Counter compare D shadow register		Go
50h	AQCTL	Action qualifier control register		Go
60h	SOCEN	Start of conversion enable		Go
64h	SOCSEL	Start of conversion selection		Go
68h	SOCPERIOD	Start of conversion period		Go
6Ch	SOCCNT	Start of conversion count		Go
70h	SOCFLAG	Start of conversion flag		Go
74h	SOCCLR	Start of conversion clear		Go
80h	ETSEL	Event trigger selection		Go
84h	ETPERIOD	Event trigger period		Go
88h	ETCNT	Event trigger count		Go
90h	INTEN	Interrupt Enable Register	EALLOW	Go
94h	INTFLAG	Interrupt Flag Register		Go
98h	INTCLR	Interrupt Clear Register	EALLOW	Go
9Ch	INTFRC	Interrupt Force Register	EALLOW	Go
A0h	TZSEL	Trip Zone Selection	EALLOW	Go
ACh	TZCTL	Trip Zone control	EALLOW	Go
B0h	TZCBCOSTFLAG	Trip Zone CBC and OST Flag Register		Go
B4h	TZCBCOSTCLR	Trip Zone CBC and OST Clear Register	EALLOW	Go
C0h	DBCTL	Dead band control register		Go
D0h	DBFED	Dead band fall edge delay		Go
D4h	DBRED	Dead band rise edge delay		Go
D8h	DBFEDS	Dead band fall edge delay shadow register		Go
DCh	DBREDS	Dead band rise edge delay shadow register		Go
F0h	GLDCTL	Global load control register	EALLOW	Go
F4h	GLDOSHTCTL	Global load one shot control register		Go
F8h	GLDOSHTSTS	Global load one shot status register		Go
100h	PWM1_CMPA	PWM1 counter compare A register		Go

Table 19-14. MCPWM_6CH_REGS Registers (continued)

Offset	Acronym	Register Name	Write Protection	Section
104h	PWM1_CMPAS	PWM1 counter compare A shadow register		Go
108h	PWM1_CMPB	PWM1 counter compare B register		Go
10Ch	PWM1_CMPBS	PWM1 counter compare B shadow register		Go
120h	PWM1_AQCTLA	PWM1 action qualifier A register		Go
124h	PWM1_AQCTLAS	PWM1 action qualifier A shadow register		Go
128h	PWM1_AQCTLB	PWM1 action qualifier B register		Go
12Ch	PWM1_AQCTLBS	PWM1 action qualifier B shadow register		Go
130h	PWM1_AQSFRC	PWM1 action qualifier software force		Go
134h	PWM1_AQOTSFRC	PWM1 action qualifier one time software force		Go
300h	PWM2_CMPA	PWM2 counter compare A register		Go
304h	PWM2_CMPAS	PWM2 counter compare A shadow register		Go
308h	PWM2_CMPB	PWM2 counter compare B register		Go
30Ch	PWM2_CMPBS	PWM2 counter compare B shadow register		Go
320h	PWM2_AQCTLA	PWM2 action qualifier A register		Go
324h	PWM2_AQCTLAS	PWM2 action qualifier A shadow register		Go
328h	PWM2_AQCTLB	PWM2 action qualifier B register		Go
32Ch	PWM2_AQCTLBS	PWM2 action qualifier B shadow register		Go
330h	PWM2_AQSFRC	PWM2 action qualifier software force		Go
334h	PWM2_AQOTSFRC	PWM2 action qualifier one time software force		Go
500h	PWM3_CMPA	PWM3 counter compare A register		Go
504h	PWM3_CMPAS	PWM3 counter compare A shadow register		Go
508h	PWM3_CMPB	PWM3 counter compare B register		Go
50Ch	PWM3_CMPBS	PWM3 counter compare B shadow register		Go
520h	PWM3_AQCTLA	PWM3 action qualifier A register		Go
524h	PWM3_AQCTLAS	PWM3 action qualifier A shadow register		Go
528h	PWM3_AQCTLB	PWM3 action qualifier B register		Go
52Ch	PWM3_AQCTLBS	PWM3 action qualifier B shadow register		Go
530h	PWM3_AQSFRC	PWM3 action qualifier software force		Go
534h	PWM3_AQOTSFRC	PWM3 action qualifier one time software force		Go

Complex bit access types are encoded to fit into small table cells. Table 19-15 shows the codes that are used for access types in this section.

Table 19-15. MCPWM_6CH_REGS Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
R-0	R -0	Read Returns 0s
Write Type		
W	W	Write
W1S	W 1S	Write 1 to set
Reset or Default Value		
-n		Value after reset or the default value
Register Array Variables		

Table 19-15. MCPWM_6CH_REGS Access Type Codes (continued)

Access Type	Code	Description
i,j,k,l,m,n		When these variables are used in a register name, an offset, or an address, they refer to the value of a register array where the register is part of a group of repeating registers. The register groups form a hierarchical structure and the array is represented with a formula.
y		When this variable is used in a register name, an offset, or an address it refers to the value of a register array.

1 REVISION Register (Offset = 0h) [Reset = 00000007h]

REVISION is shown in Figure 19-40 and described in Table 19-16.

Return to the Summary Table.

IP revision id register

Figure 19-40. REVISION Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED				CAP_PRESENT	CMPCD_PRESENT	PWM3_PRESENT	PWM2_PRESENT
R-0-0h				R-0h	R-1h	R-1h	R-1h

Table 19-17. REVISION Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R-0	0h	Reserved
3	CAP_PRESENT	R	0h	This hardcoded field defines the presence of Capture mode feature. Reset type: SYSRSn
2	CMPCD_PRESENT	R	1h	This hardcoded field defines the presence of Compare C and D registers. Reset type: SYSRSn
1	PWM3_PRESENT	R	1h	This hardcoded field defines the presence of PWM3 channels. Reset type: SYSRSn
0	PWM2_PRESENT	R	1h	This hardcoded field defines the presence of PWM2 channels. Reset type: SYSRSn

2 TBCTL Register (Offset = 10h) [Reset = 00000002h]

TBCTL is shown in Figure 19-41 and described in Table 19-17.

Return to the Summary Table.

Time base control register

Figure 19-41. TBCTL Register

31	30	29	28	27	26	25	24
RESERVED		SYNCISEL					SYNCPERSEL
R-0-0h		R/W-0h					R/W-0h
23	22	21	20	19	18	17	16
SYNCPERSEL		FREE_SOFT		RESERVED	SYNCOSEL		
R/W-0h		R/W-0h		R-0-0h	R/W-0h		
15	14	13	12	11	10	9	8
SWSYNC	RESERVED			PHSDIR	PHSEN	RESERVED	PRDLD
R-0/W1S-0h	R-0-0h			R/W-0h	R/W-0h	R-0-0h	R/W-0h
7	6	5	4	3	2	1	0
RESERVED		CLKDIV				CTRMODE	
R-0-0h		R/W-0h				R/W-2h	

Table 19-19. TBCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	RESERVED	R-0	0h	Reserved
29-25	SYNCISEL	R/W	0h	These bits determines the source of SYNCIN signal. 0x0 : Disabled using SOC tieoff. 0x1-0x1F : Refer to PWM chapter of TRM. Reset type: SYSRSn
24-22	SYNCPERSEL	R/W	0h	Sync peripheral Select 000: Reserved (Disabled) 001: Reserved (Disabled) 010: CTR = PRD 011: CTR = 0 100: CTR = CMPC, Count direction Up 101: CTR = CMPC, Count direction Down 110: CTR = CMPD, Count direction Up 111: CTR = CMPD, Count direction Down Reset type: SYSRSn
21-20	FREE_SOFT	R/W	0h	Emulation Mode Bits. These bits select the behavior of the time-base counter during emulation events 00: Stop after the next time-base counter increment or decrement 01: Stop when counter completes a whole cycle: - Up-count mode: stop when the time-base counter = period (TBCTR = TBPRD) - Up-down-count mode: stop when the time-base counter = 0x00 (TBCTR = 0x00) 1x: Free run Reset type: SYSRSn
19	RESERVED	R-0	0h	Reserved

ADVANCE INFORMATION

Table 19-19. TBCTL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
18-16	SYNCOSEL	R/W	0h	Sync Output Select 000: SWFSYNC 001: CTR = zero: Time-base counter equal to zero (TBCTR = 0x00) 010: CTR = CMPC : Time-base counter equal to counter-compare C (TBCTR = CMPC) 011: CTR = CMPD : Time-base counter equal to counter-compare D (TBCTR = CMPD) 1xx: Disabled MCPWMxSYNCO sync signal Reset type: SYSRSn
15	SWSYNC	R-0/W1S	0h	Software Forced Sync Pulse 0: Writing a 0 has no effect and reads always return a 0. 1: Writing a 1 forces a one-time synchronization pulse to be generated. Reset type: SYSRSn
14-12	RESERVED	R-0	0h	Reserved
11	PHSDIR	R/W	0h	Phase Direction Bit. This bit is only used when the time-base counter is configured in the up-down-count mode. In the up-count mode, this bit is ignored. The bit indicates the direction of the time-base counter (TBCTR) after a sync event occurs and a new phase value is loaded from the phase (TBPHS) register. 0: Count down after the sync event. 1: Count up after the sync event. Reset type: SYSRSn
10	PHSEN	R/W	0h	Load Phase register to time-base counter(TBCNTR) 0: Do not load the time-base counter (TBCTR) from the time-base phase register (TBPHS). 1: Allow Counter to be loaded from the Phase register (TBPHS) when an MCPWMxSYNCI input signal occurs or a software-forced sync (SWSYNC). Reset type: SYSRSn
9	RESERVED	R-0	0h	Reserved
8	PRDLD	R/W	0h	Shadow to Active load of TBPRD register 0: Shadow to Active Load of TBPRD occurs when TBCTR = 0 1 : Disabled shadow to active load of TBPRD Reset type: SYSRSn
7-6	RESERVED	R-0	0h	Reserved

Table 19-19. TBCTL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
5-2	CLKDIV	R/W	0h	Time Base Clock Pre-Scale Bits These bits select the time base clock pre-scale value (TBCLK = MCPWMCLK/CLKDIV): 0000: /1 (default on reset) 0001: /2 0010: /4 0011: /8 0100: /16 0101: /32 0110: /64 0111: /128 1000: /256 1001: /512 1010: /1024 1011: /2048 1100: /4096 1101: /8192 1110: /16384 1111: /32768 Reset type: SYSRSn
1-0	CTRMODE	R/W	2h	The time-base counter mode is normally configured once and not changed during normal operation. If you change the mode of the counter, the change will take effect at the next TBCLK edge and the current counter value shall increment or decrement from the value before the mode change. These bits set the time-base counter mode of operation as follows: 00: Up-count mode 01: Up-down count mode 1x: Freeze counter operation (default on reset) Reset type: SYSRSn

3 TBPRD Register (Offset = 14h) [Reset = 00000000h]

TBPRD is shown in Figure 19-42 and described in Table 19-18.

Return to the Summary Table.

Time base period register

Figure 19-42. TBPRD Register

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

Table 19-21. TBPRD Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R-0	0h	Reserved
15-0	TBPRD	R/W	0h	Time Base Period Register These bits determine the period of the time-base counter. This sets the PWM frequency. Shadowing of this register is enabled and disabled by the TBCTL[PRDLD] bit. By default this register is shadowed. - If TBCTL[PRDLD] = 0, then the shadow is enabled. This register will also loaded from the shadow register when the time-base counter equals zero. - If TBCTL[PRDLD] = 1, then the shadow is disabled. Reset type: SYSRSn

4 TBPRDS Register (Offset = 18h) [Reset = 00000000h]

TBPRDS is shown in Figure 19-43 and described in Table 19-19.

Return to the Summary Table.

Time base period shadow register

Figure 19-43. TBPRDS Register

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

Table 19-23. TBPRDS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R-0	0h	Reserved
15-0	TBPRDS	R/W	0h	Time Base Period Shadow Register The value in the TBPRDS register is loaded into TBPRD register when shadow to active load occurs. Reset type: SYSRSn

ADVANCE INFORMATION

5 TBPHS Register (Offset = 1Ch) [Reset = 00000000h]

TBPHS is shown in Figure 19-44 and described in Table 19-20.

Return to the Summary Table.

Time base phase offset register

Figure 19-44. TBPHS Register

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

Table 19-25. TBPHS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R-0	0h	Reserved
15-0	TBPHS	R/W	0h	Phase Offset Register These bits set time-base counter phase of the PWM relative to the sync (MCPWMxSYNCI / SWFSYNC) - If TBCTL[PHSEN] = 0, then the sync event is ignored. - If TBCTL[PHSEN] = 1, then the time-base counter (TBCTR) will be loaded with the phase (TBPHS) when a sync event occurs. The sync event can be initiated by the input sync signal (MCPWMxSYNCI) or by a software forced sync. Reset type: SYSRSn

6 TBSTS Register (Offset = 20h) [Reset = 00000000h]

TBSTS is shown in Figure 19-45 and described in Table 19-21.

Return to the Summary Table.

Time base status register

Figure 19-45. TBSTS Register

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

Table 19-27. TBSTS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R-0	0h	Reserved
1	SYNCI	R	0h	Input Synchronization Latched Status Bit 0: No external sync event has occurred. 1: External sync event has occurred (MCPWMxSYNCI). Reset type: SYSRSn
0	CTRDIR	R	0h	Time Base Counter Direction Status Bit 0: Time-Base Counter is currently counting up. 1: Time-Base Counter is currently counting down. Note: This bit is only valid when the counter is not frozen. Reset type: SYSRSn

7 TBSTSCLR Register (Offset = 24h) [Reset = 00000000h]

TBSTSCLR is shown in Figure 19-46 and described in Table 19-22.

Return to the Summary Table.

Time base status clear register

Figure 19-46. TBSTSCLR Register

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

Table 19-29. TBSTSCLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R-0	0h	Reserved
1	SYNCI	R-0/W1S	0h	Input Synchronization Latched Status Clear 0: Writing a 0 has no effect. 1: Writing a 1 will clear the TBSTS[SYNCI] bit. Reset type: SYSRSn
0	RESERVED	R-0	0h	Reserved

ADVANCE INFORMATION

8 TBCTR Register (Offset = 28h) [Reset = 00000000h]

TBCTR is shown in Figure 19-47 and described in Table 19-23.

Return to the Summary Table.

Time base counter register

Figure 19-47. TBCTR Register

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

Table 19-31. TBCTR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R-0	0h	Reserved
15-0	TBCTR	R/W	0h	Time Base Counter Register Reset type: SYSRSn

9 CMPCTL Register (Offset = 30h) [Reset = 00000000h]

CMPCTL is shown in Figure 19-48 and described in Table 19-24.

Return to the Summary Table.

Counter compare control register

Figure 19-48. CMPCTL Register

31	30	29	28	27	26	25	24
RESERVED				LOADDMODE		LOADCMODE	
R-0-0h				R/W-0h		R/W-0h	
23	22	21	20	19	18	17	16
RESERVED				PWM3_LOADBMODE		PWM3_LOADAMODE	
R-0-0h				R/W-0h		R/W-0h	
15	14	13	12	11	10	9	8
RESERVED				PWM2_LOADBMODE		PWM2_LOADAMODE	
R-0-0h				R/W-0h		R/W-0h	
7	6	5	4	3	2	1	0
RESERVED				PWM1_LOADBMODE		PWM1_LOADAMODE	
R-0-0h				R/W-0h		R/W-0h	

Table 19-33. CMPCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	RESERVED	R-0	0h	Reserved
27-26	LOADDMODE	R/W	0h	Shadow to Active load of CMPD register 00: Load on CTR = Zero: Time-base counter equal to zero (TBCTR = 0x0000) 01: Load on CTR = PRD: Time-base counter equal to period (TBCTR = TBPRD) 10: Load on either CTR = Zero or CTR = PRD 11: Freeze (no loads possible) Note: This field can be present only when the configuration parameter CCM_CD_PRESENT = 1 Reset type: SYSRSn
25-24	LOADCMODE	R/W	0h	Shadow to Active load of CMPC register 00: Load on CTR = Zero: Time-base counter equal to zero (TBCTR = 0x0000) 01: Load on CTR = PRD: Time-base counter equal to period (TBCTR = TBPRD) 10: Load on either CTR = Zero or CTR = PRD 11: Freeze (no loads possible) Note: This field can be present only when the configuration parameter CCM_CD_PRESENT = 1 Reset type: SYSRSn
23-20	RESERVED	R-0	0h	Reserved

Table 19-33. CMPCTL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
19-18	PWM3_LOADBMODE	R/W	0h	Shadow to Active load of PWM3_CMPB register 00: Load on CTR = Zero: Time-base counter equal to zero (TBCTR = 0x0000) 01: Load on CTR = PRD: Time-base counter equal to period (TBCTR = TBPRD) 10: Load on either CTR = Zero or CTR = PRD 11: Freeze (no loads possible) Note: This field can be present only when the configuration parameter PWM3_PRESENT = 1 Reset type: SYSRSn
17-16	PWM3_LOADAMODE	R/W	0h	Shadow to Active load of PWM3_CMPA register 00: Load on CTR = Zero: Time-base counter equal to zero (TBCTR = 0x0000) 01: Load on CTR = PRD: Time-base counter equal to period (TBCTR = TBPRD) 10: Load on either CTR = Zero or CTR = PRD 11: Freeze (no loads possible) Note: This field can be present only when the configuration parameter PWM3_PRESENT = 1 Reset type: SYSRSn
15-12	RESERVED	R-0	0h	Reserved
11-10	PWM2_LOADBMODE	R/W	0h	Shadow to Active load of PWM2_CMPB register 00: Load on CTR = Zero: Time-base counter equal to zero (TBCTR = 0x0000) 01: Load on CTR = PRD: Time-base counter equal to period (TBCTR = TBPRD) 10: Load on either CTR = Zero or CTR = PRD 11: Freeze (no loads possible) Note: This field can be present only when the configuration parameter PWM2_PRESENT = 1 Reset type: SYSRSn
9-8	PWM2_LOADAMODE	R/W	0h	Shadow to Active load of PWM2_CMPA register 00: Load on CTR = Zero: Time-base counter equal to zero (TBCTR = 0x0000) 01: Load on CTR = PRD: Time-base counter equal to period (TBCTR = TBPRD) 10: Load on either CTR = Zero or CTR = PRD 11: Freeze (no loads possible) Note: This field can be present only when the configuration parameter PWM2_PRESENT = 1 Reset type: SYSRSn
7-4	RESERVED	R-0	0h	Reserved
3-2	PWM1_LOADBMODE	R/W	0h	Shadow to Active load of PWM1_CMPB register 00: Load on CTR = Zero: Time-base counter equal to zero (TBCTR = 0x0000) 01: Load on CTR = PRD: Time-base counter equal to period (TBCTR = TBPRD) 10: Load on either CTR = Zero or CTR = PRD 11: Freeze (no loads possible) Reset type: SYSRSn

Table 19-33. CMPCTL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
1-0	PWM1_LOADAMODE	R/W	0h	Shadow to Active load of PWM1_CMPA register 00: Load on CTR = Zero: Time-base counter equal to zero (TBCTR = 0x0000) 01: Load on CTR = PRD: Time-base counter equal to period (TBCTR = TBPRD) 10: Load on either CTR = Zero or CTR = PRD 11: Freeze (no loads possible) Reset type: SYSRSn

10 CMPC Register (Offset = 40h) [Reset = 00000000h]

CMPC is shown in Figure 19-49 and described in Table 19-25.

Return to the Summary Table.

Counter compare C register

Figure 19-49. CMPC Register

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

Table 19-35. CMPC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R-0	0h	Reserved
15-0	CMPC	R/W	0h	Compare C register The value in the CMPC register is continuously compared to the time-base counter (TBCTR). When the values are equal, the counter-compare module generates a 'time-base counter equal to counter compare C' event. Shadowing of this register is enabled and disabled by the CMPCTL[LOADCMODE] field. By default this register is shadowed. Reset type: SYSRSn

11 CMPD Register (Offset = 44h) [Reset = 00000000h]

CMPD is shown in Figure 19-50 and described in Table 19-26.

Return to the Summary Table.

Counter compare D register

Figure 19-50. CMPD Register

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

Table 19-37. CMPD Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R-0	0h	Reserved
15-0	CMPD	R/W	0h	Compare D register The value in the CMPD register is continuously compared to the time-base counter (TBCTR). When the values are equal, the counter-compare module generates a 'time-base counter equal to counter compare D' event. Shadowing of this register is enabled and disabled by the CMPCTL[LOADDMODE] field. By default this register is shadowed. Reset type: SYSRSn

12 CMPCS Register (Offset = 48h) [Reset = 00000000h]

CMPCS is shown in Figure 19-51 and described in Table 19-27.

Return to the Summary Table.

Counter compare C shadow register

Figure 19-51. CMPCS Register

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

Table 19-39. CMPCS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R-0	0h	Reserved
15-0	CMPCS	R/W	0h	Compare C Shadow Register The value in the CMPCS register is loaded into CMPC register when shadow to active load occurs. Reset type: SYSRSn

ADVANCE INFORMATION

13 CMPDS Register (Offset = 4Ch) [Reset = 00000000h]

CMPDS is shown in Figure 19-52 and described in Table 19-28.

Return to the Summary Table.

Counter compare D shadow register

Figure 19-52. CMPDS Register

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

Table 19-41. CMPDS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R-0	0h	Reserved
15-0	CMPDS	R/W	0h	Compare D Shadow Register The value in the CMPDS register is loaded into CMPD register when shadow to active load occurs. Reset type: SYSRSn

ADVANCE INFORMATION

14 AQCTL Register (Offset = 50h) [Reset = 00000000h]

AQCTL is shown in Figure 19-53 and described in Table 19-29.

Return to the Summary Table.

Action qualifier control register

Figure 19-53. AQCTL Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED				PWM3_LDAQBMODE		PWM3_LDAQAMODE	
R-0-0h				R/W-0h		R/W-0h	
15	14	13	12	11	10	9	8
RESERVED				PWM2_LDAQBMODE		PWM2_LDAQAMODE	
R-0-0h				R/W-0h		R/W-0h	
7	6	5	4	3	2	1	0
RESERVED				PWM1_LDAQBMODE		PWM1_LDAQAMODE	
R-0-0h				R/W-0h		R/W-0h	

Table 19-43. AQCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-20	RESERVED	R-0	0h	Reserved
19-18	PWM3_LDAQBMODE	R/W	0h	Shadow to Active load of PWM3_AQCTLB register 00: Load on CTR = Zero: Time-base counter equal to zero (TBCTR = 0x0000) 01: Load on CTR = PRD: Time-base counter equal to period (TBCTR = TBPRD) 10: Load on either CTR = Zero or CTR = PRD 11: Freeze (no loads possible) Note: This field can be present only when the configuration parameter PWM3_PRESENT = 1 Reset type: SYSRSn
17-16	PWM3_LDAQAMODE	R/W	0h	Shadow to Active load of PWM3_AQCTLA register 00: Load on CTR = Zero: Time-base counter equal to zero (TBCTR = 0x0000) 01: Load on CTR = PRD: Time-base counter equal to period (TBCTR = TBPRD) 10: Load on either CTR = Zero or CTR = PRD 11: Freeze (no loads possible) Note: This field can be present only when the configuration parameter PWM3_PRESENT = 1 Reset type: SYSRSn
15-12	RESERVED	R-0	0h	Reserved

Table 19-43. AQCTL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
11-10	PWM2_LDAQBMODE	R/W	0h	Shadow to Active load of PWM2_AQCTLB register 00: Load on CTR = Zero: Time-base counter equal to zero (TBCTR = 0x0000) 01: Load on CTR = PRD: Time-base counter equal to period (TBCTR = TBPRD) 10: Load on either CTR = Zero or CTR = PRD 11: Freeze (no loads possible) Note: This field can be present only when the configuration parameter PWM2_PRESENT = 1 Reset type: SYSRSn
9-8	PWM2_LDAQAMODE	R/W	0h	Shadow to Active load of PWM2_AQCTLA register 00: Load on CTR = Zero: Time-base counter equal to zero (TBCTR = 0x0000) 01: Load on CTR = PRD: Time-base counter equal to period (TBCTR = TBPRD) 10: Load on either CTR = Zero or CTR = PRD 11: Freeze (no loads possible) Note: This field can be present only when the configuration parameter PWM2_PRESENT = 1 Reset type: SYSRSn
7-4	RESERVED	R-0	0h	Reserved
3-2	PWM1_LDAQBMODE	R/W	0h	Shadow to Active load of PWM1_AQCTLB register 00: Load on CTR = Zero: Time-base counter equal to zero (TBCTR = 0x0000) 01: Load on CTR = PRD: Time-base counter equal to period (TBCTR = TBPRD) 10: Load on either CTR = Zero or CTR = PRD 11: Freeze (no loads possible) Reset type: SYSRSn
1-0	PWM1_LDAQAMODE	R/W	0h	Shadow to Active load of PWM1_AQCTLA register 00: Load on CTR = Zero: Time-base counter equal to zero (TBCTR = 0x0000) 01: Load on CTR = PRD: Time-base counter equal to period (TBCTR = TBPRD) 10: Load on either CTR = Zero or CTR = PRD 11: Freeze (no loads possible) Reset type: SYSRSn

15 SOCEN Register (Offset = 60h) [Reset = 00000000h]

SOCEN is shown in Figure 19-54 and described in Table 19-30.

Return to the Summary Table.

Start of conversion enable

Figure 19-54. SOCEN Register

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

Table 19-45. SOCEN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R-0	0h	Reserved
3	SOCD_ENABLE	R/W	0h	SOCD Selection Enable 0 - SOCD disable 1- SOCD enable Reset type: SYSRSn
2	SOCC_ENABLE	R/W	0h	SOCC Selection Enable 0 - SOCC disable 1- SOCC enable Reset type: SYSRSn
1	SOCB_ENABLE	R/W	0h	SOCB Selection Enable 0 - SOCB disable 1- SOCB enable Reset type: SYSRSn
0	SOCA_ENABLE	R/W	0h	SOCA Selection Enable 0 - SOCA disable 1- SOCA enable Reset type: SYSRSn

16 SOCSEL Register (Offset = 64h) [Reset = 00000000h]

SOCSEL is shown in Figure 19-55 and described in Table 19-31.

Return to the Summary Table.

Start of conversion selection

Figure 19-55. SOCSEL Register

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

Table 19-47. SOCSEL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R-0	0h	Reserved

ADVANCE INFORMATION

Table 19-47. SOCSEL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
28-24	SOCD_SEL	R/W	0h	SOCD Selection Options These bits determine when SOCD pulse will be generated. 00000: Reserved (SOC D disabled) 00001: Reserved (SOC D disabled) 00010: Enable event time-base counter equal to zero (TBCTR = 0x0000) 00011: Enable event time-base counter equal to period (TBCTR = TBPRD) 00100: Enable event time-base counter equal to zero or period (TBCTR = 0x00 or TBCTR = TBPRD) 00101 - 00111: Reserved (SOC D disabled) 01000: Enable event time-base counter equal to CMPC when the timer is incrementing 01001: Enable event time-base counter equal to CMPD when the timer is incrementing 01010: Enable event time-base counter equal to PWM1_CMPA when the timer is incrementing 01011: Enable event time-base counter equal to PWM1_CMPB when the timer is incrementing 01100: Enable event time-base counter equal to PWM2_CMPA when the timer is incrementing 01101: Enable event time-base counter equal to PWM2_CMPB when the timer is incrementing 01110: Enable event time-base counter equal to PWM3_CMPA when the timer is incrementing 01111: Enable event time-base counter equal to PWM3_CMPB when the timer is incrementing 10000: Enable event time-base counter equal to CMPC when the timer is decrementing 10001: Enable event time-base counter equal to CMPD when the timer is decrementing 10010: Enable event time-base counter equal to PWM1_CMPA when the timer is decrementing 10011: Enable event time-base counter equal to PWM1_CMPB when the timer is decrementing 10100: Enable event time-base counter equal to PWM2_CMPA when the timer is decrementing 10101: Enable event time-base counter equal to PWM2_CMPB when the timer is decrementing 10110: Enable event time-base counter equal to PWM3_CMPA when the timer is decrementing 10111: Enable event time-base counter equal to PWM3_CMPB when the timer is decrementing 11000 - 11111: Reserved (SOC D disabled) Reset type: SYSRSn
23-21	RESERVED	R-0	0h	Reserved

ADVANCE INFORMATION

Table 19-47. SOCSEL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
20-16	SOCC_SEL	R/W	0h	SOCC Selection Options These bits determine when SOCC pulse will be generated. 00000: Reserved (SOC C disabled) 00001: Reserved (SOC C disabled) 00010: Enable event time-base counter equal to zero (TBCTR = 0x0000) 00011: Enable event time-base counter equal to period (TBCTR = TBPRD) 00100: Enable event time-base counter equal to zero or period (TBCTR = 0x00 or TBCTR = TBPRD) 00101 - 00111: Reserved (SOC C disabled) 01000: Enable event time-base counter equal to CMPC when the timer is incrementing 01001: Enable event time-base counter equal to CMPD when the timer is incrementing 01010: Enable event time-base counter equal to PWM1_CMPA when the timer is incrementing 01011: Enable event time-base counter equal to PWM1_CMPB when the timer is incrementing 01100: Enable event time-base counter equal to PWM2_CMPA when the timer is incrementing 01101: Enable event time-base counter equal to PWM2_CMPB when the timer is incrementing 01110: Enable event time-base counter equal to PWM3_CMPA when the timer is incrementing 01111: Enable event time-base counter equal to PWM3_CMPB when the timer is incrementing 10000: Enable event time-base counter equal to CMPC when the timer is decrementing 10001: Enable event time-base counter equal to CMPD when the timer is decrementing 10010: Enable event time-base counter equal to PWM1_CMPA when the timer is decrementing 10011: Enable event time-base counter equal to PWM1_CMPB when the timer is decrementing 10100: Enable event time-base counter equal to PWM2_CMPA when the timer is decrementing 10101: Enable event time-base counter equal to PWM2_CMPB when the timer is decrementing 10110: Enable event time-base counter equal to PWM3_CMPA when the timer is decrementing 10111: Enable event time-base counter equal to PWM3_CMPB when the timer is decrementing 11000 - 11111: Reserved (SOC C disabled) Reset type: SYSRSn
15-13	RESERVED	R-0	0h	Reserved

ADVANCE INFORMATION

Table 19-47. SOCSEL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
12-8	SOCB_SEL	R/W	0h	SOCB Selection Options These bits determine when SOCB pulse will be generated. 00000: Reserved (SOC B disabled) 00001: Reserved (SOC B disabled) 00010: Enable event time-base counter equal to zero (TBCTR = 0x0000) 00011: Enable event time-base counter equal to period (TBCTR = TBPRD) 00100: Enable event time-base counter equal to zero or period (TBCTR = 0x00 or TBCTR = TBPRD) 00101 - 00111: Reserved (SOC B disabled) 01000: Enable event time-base counter equal to CMPC when the timer is incrementing 01001: Enable event time-base counter equal to CMPD when the timer is incrementing 01010: Enable event time-base counter equal to PWM1_CMPA when the timer is incrementing 01011: Enable event time-base counter equal to PWM1_CMPB when the timer is incrementing 01100: Enable event time-base counter equal to PWM2_CMPA when the timer is incrementing 01101: Enable event time-base counter equal to PWM2_CMPB when the timer is incrementing 01110: Enable event time-base counter equal to PWM3_CMPA when the timer is incrementing 01111: Enable event time-base counter equal to PWM3_CMPB when the timer is incrementing 10000: Enable event time-base counter equal to CMPC when the timer is decrementing 10001: Enable event time-base counter equal to CMPD when the timer is decrementing 10010: Enable event time-base counter equal to PWM1_CMPA when the timer is decrementing 10011: Enable event time-base counter equal to PWM1_CMPB when the timer is decrementing 10100: Enable event time-base counter equal to PWM2_CMPA when the timer is decrementing 10101: Enable event time-base counter equal to PWM2_CMPB when the timer is decrementing 10110: Enable event time-base counter equal to PWM3_CMPA when the timer is decrementing 10111: Enable event time-base counter equal to PWM3_CMPB when the timer is decrementing 11000 - 11111: Reserved (SOC B disabled) Reset type: SYSRSn
7-5	RESERVED	R-0	0h	Reserved

ADVANCE INFORMATION

Table 19-47. SOCSEL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
4-0	SOCA_SEL	R/W	0h	SOCA Selection Options These bits determine when SOCA pulse will be generated. 00000: Reserved (SOC A disabled) 00001: Reserved (SOC A disabled) 00010: Enable event time-base counter equal to zero (TBCTR = 0x0000) 00011: Enable event time-base counter equal to period (TBCTR = TBPRD) 00100: Enable event time-base counter equal to zero or period (TBCTR = 0x00 or TBCTR = TBPRD) 00101 - 00111: Reserved (SOC A disabled) 01000: Enable event time-base counter equal to CMPC when the timer is incrementing 01001: Enable event time-base counter equal to CMPD when the timer is incrementing 01010: Enable event time-base counter equal to PWM1_CMPA when the timer is incrementing 01011: Enable event time-base counter equal to PWM1_CMPB when the timer is incrementing 01100: Enable event time-base counter equal to PWM2_CMPA when the timer is incrementing 01101: Enable event time-base counter equal to PWM2_CMPB when the timer is incrementing 01110: Enable event time-base counter equal to PWM3_CMPA when the timer is incrementing 01111: Enable event time-base counter equal to PWM3_CMPB when the timer is incrementing 10000: Enable event time-base counter equal to CMPC when the timer is decrementing 10001: Enable event time-base counter equal to CMPD when the timer is decrementing 10010: Enable event time-base counter equal to PWM1_CMPA when the timer is decrementing 10011: Enable event time-base counter equal to PWM1_CMPB when the timer is decrementing 10100: Enable event time-base counter equal to PWM2_CMPA when the timer is decrementing 10101: Enable event time-base counter equal to PWM2_CMPB when the timer is decrementing 10110: Enable event time-base counter equal to PWM3_CMPA when the timer is decrementing 10111: Enable event time-base counter equal to PWM3_CMPB when the timer is decrementing 11000 - 11111: Reserved (SOC A disabled) Reset type: SYSRSn

ADVANCE INFORMATION

17 SOCPERIOD Register (Offset = 68h) [Reset = 00000000h]

SOCPERIOD is shown in Figure 19-56 and described in Table 19-32.

Return to the Summary Table.

Start of conversion period

Figure 19-56. SOCPERIOD Register

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

Table 19-49. SOCPERIOD Register Field Descriptions

Bit	Field	Type	Reset	Description
31-27	RESERVED	R-0	0h	Reserved
26-24	SOCD_PERIOD	R/W	0h	SOCD Period These bits determine how many selected SOCSEL[SOCD_SEL] events need to occur before an SOCD pulse is generated. To be generated, the pulse must be enabled (SOCEN[SOCD_ENABLE] = 1). The SOCD pulse will be generated even if the status flag is set from a previous start of conversion (SOCFLAG[SOCD] = 1). Once the SOCD pulse is generated, the SOCCNT[SOCD_CNT] bits will automatically be cleared. Reset type: SYSRSn
23-19	RESERVED	R-0	0h	Reserved
18-16	SOCC_PERIOD	R/W	0h	SOCC Period These bits determine how many selected SOCSEL[SOCC_SEL] events need to occur before an SOCC pulse is generated. To be generated, the pulse must be enabled (SOCEN[SOCC_ENABLE] = 1). The SOCC pulse will be generated even if the status flag is set from a previous start of conversion (SOCFLAG[SOCC] = 1). Once the SOCC pulse is generated, the SOCCNT[SOCC_CNT] bits will automatically be cleared. Reset type: SYSRSn
15-11	RESERVED	R-0	0h	Reserved
10-8	SOCB_PERIOD	R/W	0h	SOCB Period These bits determine how many selected SOCSEL[SOCB_SEL] events need to occur before an SOCB pulse is generated. To be generated, the pulse must be enabled (SOCEN[SOCB_ENABLE] = 1). The SOCB pulse will be generated even if the status flag is set from a previous start of conversion (SOCFLAG[SOCB] = 1). Once the SOCB pulse is generated, the SOCCNT[SOCB_CNT] bits will automatically be cleared. Reset type: SYSRSn

Table 19-49. SOCPERIOD Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
7-3	RESERVED	R-0	0h	Reserved
2-0	SOCA_PERIOD	R/W	0h	SOCA Period These bits determine how many selected SOCSEL[SOCA_SEL] events need to occur before an SOCA pulse is generated. To be generated, the pulse must be enabled (SOCEN[SOCA_ENABLE] = 1). The SOCA pulse will be generated even if the status flag is set from a previous start of conversion (SOCFLAG[SOCA] = 1). Once the SOCA pulse is generated, the SOCCNT[SOCA_CNT] bits will automatically be cleared. Reset type: SYSRSn

18 SOCCNT Register (Offset = 6Ch) [Reset = 00000000h]

SOCCNT is shown in Figure 19-57 and described in Table 19-33.

Return to the Summary Table.

Start of conversion count

Figure 19-57. SOCCNT Register

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

Table 19-51. SOCCNT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-27	RESERVED	R-0	0h	Reserved
26-24	SOCD_CNT	R	0h	SOC D Counter Register These bits indicate how many selected SOCSEL[SOCD_SEL] events have occurred. These bits are automatically cleared when a SOCD pulse is generated. Reset type: SYSRSn
23-19	RESERVED	R-0	0h	Reserved
18-16	SOCC_CNT	R	0h	SOC C Counter Register These bits indicate how many selected SOCSEL[SOCC_SEL] events have occurred. These bits are automatically cleared when a SOCC pulse is generated. Reset type: SYSRSn
15-11	RESERVED	R-0	0h	Reserved
10-8	SOCB_CNT	R	0h	SOC B Counter Register These bits indicate how many selected SOCSEL[SOCB_SEL] events have occurred. These bits are automatically cleared when a SOCB pulse is generated. Reset type: SYSRSn
7-3	RESERVED	R-0	0h	Reserved
2-0	SOCA_CNT	R	0h	SOC A Counter Register These bits indicate how many selected SOCSEL[SOCA_SEL] events have occurred. These bits are automatically cleared when a SOCA pulse is generated. Reset type: SYSRSn

19 SOCFLAG Register (Offset = 70h) [Reset = 00000000h]

SOCFLAG is shown in Figure 19-58 and described in Table 19-34.

Return to the Summary Table.

Start of conversion flag

Figure 19-58. SOCFLAG Register

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

Table 19-53. SOCFLAG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R-0	0h	Reserved
3	SOCD	R	0h	Latched SOC D Status Flag SOCD output will continue to pulse even if the flag bit is set. 0: Indicates no event occurred 1: Indicates that a start of conversion pulse was generated on SOCD. The SOCD output will continue to be generated even if the flag bit is set. Reset type: SYSRSn
2	SOCC	R	0h	Latched SOC C Status Flag SOCC output will continue to pulse even if the flag bit is set. 0: Indicates no event occurred 1: Indicates that a start of conversion pulse was generated on SOCC. The SOCC output will continue to be generated even if the flag bit is set. Reset type: SYSRSn
1	SOCB	R	0h	Latched SOC B Status Flag SOCB output will continue to pulse even if the flag bit is set. 0: Indicates no event occurred 1: Indicates that a start of conversion pulse was generated on SOCB. The SOCB output will continue to be generated even if the flag bit is set. Reset type: SYSRSn
0	SOCA	R	0h	Latched SOC A Status Flag SOCA output will continue to pulse even if the flag bit is set. 0: Indicates no event occurred 1: Indicates that a start of conversion pulse was generated on SOCA. The SOCA output will continue to be generated even if the flag bit is set. Reset type: SYSRSn

20 SOCCLR Register (Offset = 74h) [Reset = 00000000h]

SOCCLR is shown in Figure 19-59 and described in Table 19-35.

Return to the Summary Table.

Start of conversion clear

Figure 19-59. SOCCLR Register

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

Table 19-55. SOCCLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R-0	0h	Reserved
3	SOCD	R-0/W1S	0h	Clear SOC D Status Flag 0: Writing a 0 has no effect. 1: Writing a 1 will clear the SOCFLAG[SOCD] bit. Reset type: SYSRSn
2	SOCC	R-0/W1S	0h	Clear SOC C Status Flag 0: Writing a 0 has no effect. 1: Writing a 1 will clear the SOCFLAG[SOCC] bit. Reset type: SYSRSn
1	SOCB	R-0/W1S	0h	Clear SOC B Status Flag 0: Writing a 0 has no effect. 1: Writing a 1 will clear the SOCFLAG[SOCB] bit. Reset type: SYSRSn
0	SOCA	R-0/W1S	0h	Clear SOC A Status Flag 0: Writing a 0 has no effect. 1: Writing a 1 will clear the SOCFLAG[SOCA] bit. Reset type: SYSRSn

21 ETSEL Register (Offset = 80h) [Reset = 00000000h]

ETSEL is shown in Figure 19-60 and described in Table 19-36.

Return to the Summary Table.

Event trigger selection

Figure 19-60. ETSEL Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
R-0-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED			ET2_SEL					RESERVED			ET1_SEL				
R-0-0h			R/W-0h					R-0-0h			R/W-0h				

Table 19-57. ETSEL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-13	RESERVED	R-0	0h	Reserved

Table 19-57. ETSEL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
12-8	ET2_SEL	R/W	0h	Event trigger2 Selection Options These bits determine when event trigger pulse will be generated. 00000: Reserved (ET2 Disabled) 00001: Reserved (ET2 Disabled) 00010: Enable event time-base counter equal to zero (TBCTR = 0x0000) 00011: Enable event time-base counter equal to period (TBCTR = TBPRD) 00100: Enable event time-base counter equal to zero or period (TBCTR = 0x00 or TBCTR = TBPRD) 00101 - 00111: Reserved (ET2 Disabled) 01000: Enable event time-base counter equal to CMPC when the timer is incrementing 01001: Enable event time-base counter equal to CMPD when the timer is incrementing 01010: Enable event time-base counter equal to PWM1_CMPA when the timer is incrementing 01011: Enable event time-base counter equal to PWM1_CMPB when the timer is incrementing 01100: Enable event time-base counter equal to PWM2_CMPA when the timer is incrementing 01101: Enable event time-base counter equal to PWM2_CMPB when the timer is incrementing 01110: Enable event time-base counter equal to PWM3_CMPA when the timer is incrementing 01111: Enable event time-base counter equal to PWM3_CMPB when the timer is incrementing 10000: Enable event time-base counter equal to CMPC when the timer is decrementing 10001: Enable event time-base counter equal to CMPD when the timer is decrementing 10010: Enable event time-base counter equal to PWM1_CMPA when the timer is decrementing 10011: Enable event time-base counter equal to PWM1_CMPB when the timer is decrementing 10100: Enable event time-base counter equal to PWM2_CMPA when the timer is decrementing 10101: Enable event time-base counter equal to PWM2_CMPB when the timer is decrementing 10110: Enable event time-base counter equal to PWM3_CMPA when the timer is decrementing 10111: Enable event time-base counter equal to PWM3_CMPB when the timer is decrementing 11000 - 11111: Reserved (ET2 Disabled) Reset type: SYSRSn
7-5	RESERVED	R-0	0h	Reserved

ADVANCE INFORMATION

Table 19-57. ETSEL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
4-0	ET1_SEL	R/W	0h	Event trigger1 Selection Options These bits determine when event trigger pulse will be generated. 00000: Reserved (ET1 Disabled) 00001: Reserved (ET1 Disabled) 00010: Enable event time-base counter equal to zero (TBCTR = 0x0000) 00011: Enable event time-base counter equal to period (TBCTR = TBPRD) 00100: Enable event time-base counter equal to zero or period (TBCTR = 0x00 or TBCTR = TBPRD) 00101 - 00111: Reserved (ET1 Disabled) 01000: Enable event time-base counter equal to CMPC when the timer is incrementing 01001: Enable event time-base counter equal to CMPD when the timer is incrementing 01010: Enable event time-base counter equal to PWM1_CMPA when the timer is incrementing 01011: Enable event time-base counter equal to PWM1_CMPB when the timer is incrementing 01100: Enable event time-base counter equal to PWM2_CMPA when the timer is incrementing 01101: Enable event time-base counter equal to PWM2_CMPB when the timer is incrementing 01110: Enable event time-base counter equal to PWM3_CMPA when the timer is incrementing 01111: Enable event time-base counter equal to PWM3_CMPB when the timer is incrementing 10000: Enable event time-base counter equal to CMPC when the timer is decrementing 10001: Enable event time-base counter equal to CMPD when the timer is decrementing 10010: Enable event time-base counter equal to PWM1_CMPA when the timer is decrementing 10011: Enable event time-base counter equal to PWM1_CMPB when the timer is decrementing 10100: Enable event time-base counter equal to PWM2_CMPA when the timer is decrementing 10101: Enable event time-base counter equal to PWM2_CMPB when the timer is decrementing 10110: Enable event time-base counter equal to PWM3_CMPA when the timer is decrementing 10111: Enable event time-base counter equal to PWM3_CMPB when the timer is decrementing 11000 - 11111: Reserved (ET1 Disabled) Reset type: SYSRSn

ADVANCE INFORMATION

22 ETPERIOD Register (Offset = 84h) [Reset = 00000000h]

ETPERIOD is shown in Figure 19-61 and described in Table 19-37.

Return to the Summary Table.

Event trigger period

Figure 19-61. ETPERIOD Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED					ET2_PERIOD		
R-0-0h					R/W-0h		
7	6	5	4	3	2	1	0
RESERVED					ET1_PERIOD		
R-0-0h					R/W-0h		

Table 19-59. ETPERIOD Register Field Descriptions

Bit	Field	Type	Reset	Description
31-11	RESERVED	R-0	0h	Reserved
10-8	ET2_PERIOD	R/W	0h	These bits determine how many selected ETSEL[ET2_SEL] events need to occur before an interrupt is generated. If the interrupt status flag is set from a previous interrupt (INTFLG[ET2] = 1) then no interrupt will be generated until the flag is cleared via the INTCLR[ET2] bit. This allows for one interrupt to be pending while another is still being serviced. Once the interrupt is generated, the ETCNT[ET2_CNT] bits will automatically be cleared. Writing a PERIOD value that is the same as the current counter value will trigger an interrupt if it is enabled and the status flag is clear. Writing a PERIOD value that is less than the current counter value will result in an undefined state. If a counter event occurs at the same instant as a new zero or non-zero PERIOD value is written, the counter is incremented. Reset type: SYSRSn
7-3	RESERVED	R-0	0h	Reserved
2-0	ET1_PERIOD	R/W	0h	These bits determine how many selected ETSEL[ET1_SEL] events need to occur before an interrupt is generated. If the interrupt status flag is set from a previous interrupt (INTFLG[ET1] = 1) then no interrupt will be generated until the flag is cleared via the INTCLR[ET1] bit. This allows for one interrupt to be pending while another is still being serviced. Once the interrupt is generated, the ETCNT[ET1_CNT] bits will automatically be cleared. Writing a PERIOD value that is the same as the current counter value will trigger an interrupt if it is enabled and the status flag is clear. Writing a PERIOD value that is less than the current counter value will result in an undefined state. If a counter event occurs at the same instant as a new zero or non-zero PERIOD value is written, the counter is incremented. Reset type: SYSRSn

23 ETCNT Register (Offset = 88h) [Reset = 00000000h]

ETCNT is shown in Figure 19-62 and described in Table 19-38.

Return to the Summary Table.

Event trigger count

Figure 19-62. ETCNT Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
R-0-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED					ET2_CNT			RESERVED					ET1_CNT		
R-0-0h					R-0h			R-0-0h					R-0h		

Table 19-61. ETCNT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-11	RESERVED	R-0	0h	Reserved
10-8	ET2_CNT	R	0h	Event trigger2 Counter Register These bits indicate how many selected ET_SEL[ET2_SEL] events have occurred. These bits are automatically cleared once INTFLAG.ET2 is generated (irrespective of INTEN.ET2 configuration). Reset type: SYSRSn
7-3	RESERVED	R-0	0h	Reserved
2-0	ET1_CNT	R	0h	Event trigger1 Counter Register These bits indicate how many selected ET_SEL[ET1_SEL] events have occurred. These bits are automatically cleared once INTFLAG.ET1 is generated (irrespective of INTEN.ET1 configuration). Reset type: SYSRSn

24 INTEN Register (Offset = 90h) [Reset = 00000000h]

INTEN is shown in Figure 19-63 and described in Table 19-39.

Return to the Summary Table.

Interrupt Enable Register

Figure 19-63. INTEN Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED		CNT_OVF	ET2	ET1	OST	CBC	RESERVED
R-0-0h		R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R-0h

Table 19-63. INTEN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-6	RESERVED	R-0	0h	Reserved
5	CNT_OVF	R/W	0h	Counter Overflow Interrupt Enable 1 : Enables CNT_OVF interrupt 0 : Disable CNT_OVF interrupt Reset type: SYSRSn
4	ET2	R/W	0h	Event Trigger 2 Interrupt Enable 1 : Enables ET2 interrupt 0 : Disable ET2 interrupt Reset type: SYSRSn
3	ET1	R/W	0h	Event Trigger 1 Interrupt Enable 1 : Enables ET1 interrupt 0 : Disable ET1 interrupt Reset type: SYSRSn
2	OST	R/W	0h	Trip-zone One-Shot Interrupt Enable 1 : Enables OST interrupt 0 : Disable OST interrupt Reset type: SYSRSn
1	CBC	R/W	0h	Trip-zone Cycle-by-Cycle Interrupt Enable 1 : Enables CBC interrupt 0 : Disable CBC interrupt Reset type: SYSRSn
0	RESERVED	R	0h	Reserved

25 INTFLAG Register (Offset = 94h) [Reset = 00000000h]

INTFLAG is shown in Figure 19-64 and described in Table 19-40.

Return to the Summary Table.

Interrupt Flag Register

Figure 19-64. INTFLAG Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED		CNT_OVF	ET2	ET1	OST	CBC	INT
R-0-0h		R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

Table 19-65. INTFLAG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-6	RESERVED	R-0	0h	Reserved
5	CNT_OVF	R	0h	Latched Status Flag for A Counter Overflow Event 1 : CNT_OVF flag is set 0 : CNT_OVF flag is not set Reset type: SYSRSn
4	ET2	R	0h	Latched Status Flag for A Event Trigger 2 Event 1: ET2 flag is set 0: ET2 flag is not set Reset type: SYSRSn
3	ET1	R	0h	Latched Status Flag for A Event Trigger 1 Event 1: ET1 flag is set 0: ET1 flag is not set Reset type: SYSRSn
2	OST	R	0h	Latched Status Flag for A One-Shot Trip Event 1: OST flag is set 0: OST flag is not set Reset type: SYSRSn
1	CBC	R	0h	Latched Status Flag for Cycle-By-Cycle Trip Event 1: CBC flag is set 0: CBC flag is not set Reset type: SYSRSn
0	INT	R	0h	Global Interrupt Status Flag 1: Global flag is set 0: Global flag is not set Reset type: SYSRSn

26 INTCLR Register (Offset = 98h) [Reset = 00000000h]

INTCLR is shown in Figure 19-65 and described in Table 19-41.

Return to the Summary Table.

Interrupt Clear Register

Figure 19-65. INTCLR Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED		CNT_OVF	ET2	ET1	OST	CBC	INT
R-0-0h		R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h

Table 19-67. INTCLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-6	RESERVED	R-0	0h	Reserved
5	CNT_OVF	R-0/W1S	0h	Clear Counter Overflow Flag Writing '1' will clear INTFLAG[CNT_OVF] register. Reset type: SYSRSn
4	ET2	R-0/W1S	0h	Clear Event Trigger 2 Flag Writing '1' will clear INTFLAG[ET2] register. Reset type: SYSRSn
3	ET1	R-0/W1S	0h	Clear Event Trigger 1 Flag Writing '1' will clear INTFLAG[ET1] register. Reset type: SYSRSn
2	OST	R-0/W1S	0h	Clear One-Shot Trip Latch Writing '1' will clear INTFLAG[OST] register. Reset type: SYSRSn
1	CBC	R-0/W1S	0h	Clear Cycle-by-Cycle Trip Latch Writing '1' will clear INTFLAG[CBC] register. Reset type: SYSRSn
0	INT	R-0/W1S	0h	Clear Global Interrupt Flag Writing '1' will clear INTFLAG[INT] register. Reset type: SYSRSn

27 INTFRC Register (Offset = 9Ch) [Reset = 00000000h]

INTFRC is shown in Figure 19-66 and described in Table 19-42.

Return to the Summary Table.

Interrupt Force Register

Figure 19-66. INTFRC Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED		CNT_OVF	ET2	ET1	OST	CBC	RESERVED
R-0-0h		R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0h

Table 19-69. INTFRC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-6	RESERVED	R-0	0h	Reserved
5	CNT_OVF	R-0/W1S	0h	Force Counter Overflow Interrupt Writing '1' will set INTFLAG[CNT_OVF] register. Reset type: SYSRSn
4	ET2	R-0/W1S	0h	Force Event Trigger 2 Interrupt Writing '1' will set INTFLAG[ET2] register. Reset type: SYSRSn
3	ET1	R-0/W1S	0h	Force Event Trigger 1 Interrupt Writing '1' will set INTFLAG[ET1] register. Reset type: SYSRSn
2	OST	R-0/W1S	0h	Force One-Shot Trip Interrupt Writing '1' will set INTFLAG[OST] register. Reset type: SYSRSn
1	CBC	R-0/W1S	0h	Force Cycle-by-Cycle Trip Interrupt Writing '1' will set INTFLAG[CBC] register. Reset type: SYSRSn
0	RESERVED	R	0h	Reserved

28 TZSEL Register (Offset = A0h) [Reset = 00000000h]

TZSEL is shown in Figure 19-67 and described in Table 19-43.

Return to the Summary Table.

Trip Zone Selection

Figure 19-67. TZSEL Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
OST8	OST7	OST6	OST5	OST4	OST3	OST2	OST1
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
CBC8	CBC7	CBC6	CBC5	CBC4	CBC3	CBC2	CBC1
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 19-71. TZSEL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	RESERVED	R-0	0h	Reserved
23	OST8	R/W	0h	Select Trip-zone 8 (TZ8) for OST generation 0: Disable TZ8 as a OST trip source for this MCPWM module 1: Enable TZ8 as a OST trip source for this MCPWM module Reset type: SYSRSn
22	OST7	R/W	0h	Select Trip-zone 7 (TZ7) for OST generation 0: Disable TZ7 as a OST trip source for this MCPWM module 1: Enable TZ7 as a OST trip source for this MCPWM module Reset type: SYSRSn
21	OST6	R/W	0h	Select Trip-zone 6 (TZ6) for OST generation 0: Disable TZ6 as a OST trip source for this MCPWM module 1: Enable TZ6 as a OST trip source for this MCPWM module Reset type: SYSRSn
20	OST5	R/W	0h	Select Trip-zone 5 (TZ5) for OST generation 0: Disable TZ5 as a OST trip source for this MCPWM module 1: Enable TZ5 as a OST trip source for this MCPWM module Reset type: SYSRSn
19	OST4	R/W	0h	Select Trip-zone 4 (TZ4) for OST generation 0: Disable TZ4 as a OST trip source for this MCPWM module 1: Enable TZ4 as a OST trip source for this MCPWM module Reset type: SYSRSn
18	OST3	R/W	0h	Select Trip-zone 3 (TZ3) for OST generation 0: Disable TZ3 as a OST trip source for this MCPWM module 1: Enable TZ3 as a OST trip source for this MCPWM module Reset type: SYSRSn

Table 19-71. TZSEL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
17	OST2	R/W	0h	Select Trip-zone 2 (TZ2) for OST generation 0: Disable TZ2 as a OST trip source for this MCPWM module 1: Enable TZ2 as a OST trip source for this MCPWM module Reset type: SYSRSn
16	OST1	R/W	0h	Select Trip-zone 1 (TZ1) for OST generation 0: Disable TZ1 as a OST trip source for this MCPWM module 1: Enable TZ1 as a OST trip source for this MCPWM module Reset type: SYSRSn
15-8	RESERVED	R-0	0h	Reserved
7	CBC8	R/W	0h	Select Trip-zone 8 (TZ8) for CBC generation 0: Disable TZ8 as a CBC trip source for this MCPWM module 1: Enable TZ8 as a CBC trip source for this MCPWM module Reset type: SYSRSn
6	CBC7	R/W	0h	Select Trip-zone 7 (TZ7) for CBC generation 0: Disable TZ7 as a CBC trip source for this MCPWM module 1: Enable TZ7 as a CBC trip source for this MCPWM module Reset type: SYSRSn
5	CBC6	R/W	0h	Select Trip-zone 6 (TZ6) for CBC generation 0: Disable TZ6 as a CBC trip source for this MCPWM module 1: Enable TZ6 as a CBC trip source for this MCPWM module Reset type: SYSRSn
4	CBC5	R/W	0h	Select Trip-zone 5 (TZ5) for CBC generation 0: Disable TZ5 as a CBC trip source for this MCPWM module 1: Enable TZ5 as a CBC trip source for this MCPWM module Reset type: SYSRSn
3	CBC4	R/W	0h	Select Trip-zone 4 (TZ4) for CBC generation 0: Disable TZ4 as a CBC trip source for this MCPWM module 1: Enable TZ4 as a CBC trip source for this MCPWM module Reset type: SYSRSn
2	CBC3	R/W	0h	Select Trip-zone 3 (TZ3) for CBC generation 0: Disable TZ3 as a CBC trip source for this MCPWM module 1: Enable TZ3 as a CBC trip source for this MCPWM module Reset type: SYSRSn
1	CBC2	R/W	0h	Select Trip-zone 2 (TZ2) for CBC generation 0: Disable TZ2 as a CBC trip source for this MCPWM module 1: Enable TZ2 as a CBC trip source for this MCPWM module Reset type: SYSRSn
0	CBC1	R/W	0h	Select Trip-zone 1 (TZ1) for CBC generation 0: Disable TZ1 as a CBC trip source for this MCPWM module 1: Enable TZ1 as a CBC trip source for this MCPWM module Reset type: SYSRSn

29 TZCTL Register (Offset = ACh) [Reset = 00000010h]

TZCTL is shown in Figure 19-68 and described in Table 19-44.

Return to the Summary Table.

Trip Zone control

Figure 19-68. TZCTL Register

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

Table 19-73. TZCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-6	RESERVED	R-0	0h	Reserved
5-4	CBCPULSE	R/W	1h	Clear Pulse for Cycle-By-Cycle (CBC) Trip Latch This bit field determines which pulse clears the CBC trip latch. 00: CBC trip latch is not cleared 01: CTR = ZERO pulse clears CBC trip latch 10: CTR = PRD pulse clears CBC trip latch 11: CTR = ZERO or CTR = PRD pulse clears CBC trip latch Reset type: SYSRSn
3-2	TZB	R/W	0h	Trip action on PWMxB 00: High-impedance (PWMxB = High-impedance state) 01: Force PWMxB to a high state 10: Force PWMxB to a low state 11: Do nothing, no action is taken on PWMxB. Reset type: SYSRSn
1-0	TZA	R/W	0h	Trip action on PWMxA 00: High-impedance (PWMxA = High-impedance state) 01: Force PWMxA to a high state 10: Force PWMxA to a low state 11: Do nothing, no action is taken on PWMxA. Reset type: SYSRSn

30 TZCBCOSTFLAG Register (Offset = B0h) [Reset = 00000000h]

TZCBCOSTFLAG is shown in Figure 19-69 and described in Table 19-45.

Return to the Summary Table.

Trip Zone CBC and OST Flag Register

Figure 19-69. TZCBCOSTFLAG Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
OST8	OST7	OST6	OST5	OST4	OST3	OST2	OST1
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
CBC8	CBC7	CBC6	CBC5	CBC4	CBC3	CBC2	CBC1
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

Table 19-75. TZCBCOSTFLAG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	RESERVED	R-0	0h	Reserved
23	OST8	R	0h	Latched Status Flag for OST TZ8 Trip Latch 0: Reading a 0 indicates that no OST trip has occurred by TZ8. 1: Reading a 1 indicates a OST trip has occured by TZ8. Reset type: SYSRSn
22	OST7	R	0h	Latched Status Flag for OST TZ7 Trip Latch 0: Reading a 0 indicates that no OST trip has occurred by TZ7. 1: Reading a 1 indicates a OST trip has occured by TZ7. Reset type: SYSRSn
21	OST6	R	0h	Latched Status Flag for OST TZ6 Trip Latch 0: Reading a 0 indicates that no OST trip has occurred by TZ6. 1: Reading a 1 indicates a OST trip has occured by TZ6. Reset type: SYSRSn
20	OST5	R	0h	Latched Status Flag for OST TZ5 Trip Latch 0: Reading a 0 indicates that no OST trip has occurred by TZ5. 1: Reading a 1 indicates a OST trip has occured by TZ5. Reset type: SYSRSn
19	OST4	R	0h	Latched Status Flag for OST TZ4 Trip Latch 0: Reading a 0 indicates that no OST trip has occurred by TZ4. 1: Reading a 1 indicates a OST trip has occured by TZ4. Reset type: SYSRSn
18	OST3	R	0h	Latched Status Flag for OST TZ3 Trip Latch 0: Reading a 0 indicates that no OST trip has occurred by TZ3. 1: Reading a 1 indicates a OST trip has occured by TZ3. Reset type: SYSRSn

Table 19-75. TZCBCOSTFLAG Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
17	OST2	R	0h	Latched Status Flag for OST TZ2 Trip Latch 0: Reading a 0 indicates that no OST trip has occurred by TZ2. 1: Reading a 1 indicates a OST trip has occured by TZ2. Reset type: SYSRSn
16	OST1	R	0h	Latched Status Flag for OST TZ1 Trip Latch 0: Reading a 0 indicates that no OST trip has occurred by TZ1. 1: Reading a 1 indicates a OST trip has occured by TZ1. Reset type: SYSRSn
15-8	RESERVED	R-0	0h	Reserved
7	CBC8	R	0h	Latched Status Flag for CBC TZ8 Trip Latch 0: Reading a 0 indicates that no CBC trip has occurred by TZ8. 1: Reading a 1 indicates a CBC trip has occured by TZ8. Reset type: SYSRSn
6	CBC7	R	0h	Latched Status Flag for CBC TZ7 Trip Latch 0: Reading a 0 indicates that no CBC trip has occurred by TZ7. 1: Reading a 1 indicates a CBC trip has occured by TZ7. Reset type: SYSRSn
5	CBC6	R	0h	Latched Status Flag for CBC TZ6 Trip Latch 0: Reading a 0 indicates that no CBC trip has occurred by TZ6. 1: Reading a 1 indicates a CBC trip has occured by TZ6. Reset type: SYSRSn
4	CBC5	R	0h	Latched Status Flag for CBC TZ5 Trip Latch 0: Reading a 0 indicates that no CBC trip has occurred by TZ5. 1: Reading a 1 indicates a CBC trip has occured by TZ5. Reset type: SYSRSn
3	CBC4	R	0h	Latched Status Flag for CBC TZ4 Trip Latch 0: Reading a 0 indicates that no CBC trip has occurred by TZ4. 1: Reading a 1 indicates a CBC trip has occured by TZ4. Reset type: SYSRSn
2	CBC3	R	0h	Latched Status Flag for CBC TZ3 Trip Latch 0: Reading a 0 indicates that no CBC trip has occurred by TZ3. 1: Reading a 1 indicates a CBC trip has occured by TZ3. Reset type: SYSRSn
1	CBC2	R	0h	Latched Status Flag for CBC TZ2 Trip Latch 0: Reading a 0 indicates that no CBC trip has occurred by TZ2. 1: Reading a 1 indicates a CBC trip has occured by TZ2. Reset type: SYSRSn
0	CBC1	R	0h	Latched Status Flag for CBC TZ1 Trip Latch 0: Reading a 0 indicates that no CBC trip has occurred by TZ1. 1: Reading a 1 indicates a CBC trip has occured by TZ1. Reset type: SYSRSn

31 TZCBCOSTCLR Register (Offset = B4h) [Reset = 00000000h]

TZCBCOSTCLR is shown in Figure 19-70 and described in Table 19-46.

Return to the Summary Table.

Trip Zone CBC and OST Clear Register

Figure 19-70. TZCBCOSTCLR Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
OST8	OST7	OST6	OST5	OST4	OST3	OST2	OST1
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
CBC8	CBC7	CBC6	CBC5	CBC4	CBC3	CBC2	CBC1
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h

Table 19-77. TZCBCOSTCLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	RESERVED	R-0	0h	Reserved
23	OST8	R-0/W1S	0h	Clear OST TZ8 Status Flag 0: Writing a 0 has no effect. 1: Writing a 1 will clear the TZCBCOSTFLAG[OST8] bit. Reset type: SYSRSn
22	OST7	R-0/W1S	0h	Clear OST TZ7 Status Flag 0: Writing a 0 has no effect. 1: Writing a 1 will clear the TZCBCOSTFLAG[OST7] bit. Reset type: SYSRSn
21	OST6	R-0/W1S	0h	Clear OST TZ6 Status Flag 0: Writing a 0 has no effect. 1: Writing a 1 will clear the TZCBCOSTFLAG[OST6] bit. Reset type: SYSRSn
20	OST5	R-0/W1S	0h	Clear OST TZ5 Status Flag 0: Writing a 0 has no effect. 1: Writing a 1 will clear the TZCBCOSTFLAG[OST5] bit. Reset type: SYSRSn
19	OST4	R-0/W1S	0h	Clear OST TZ4 Status Flag 0: Writing a 0 has no effect. 1: Writing a 1 will clear the TZCBCOSTFLAG[OST4] bit. Reset type: SYSRSn
18	OST3	R-0/W1S	0h	Clear OST TZ3 Status Flag 0: Writing a 0 has no effect. 1: Writing a 1 will clear the TZCBCOSTFLAG[OST3] bit. Reset type: SYSRSn

Table 19-77. TZCBCOSTCLR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
17	OST2	R-0/W1S	0h	Clear OST TZ2 Status Flag 0: Writing a 0 has no effect. 1: Writing a 1 will clear the TZCBCOSTFLAG[OST2] bit. Reset type: SYSRSn
16	OST1	R-0/W1S	0h	Clear OST TZ1 Status Flag 0: Writing a 0 has no effect. 1: Writing a 1 will clear the TZCBCOSTFLAG[OST1] bit. Reset type: SYSRSn
15-8	RESERVED	R-0	0h	Reserved
7	CBC8	R-0/W1S	0h	Clear CBC TZ8 Status Flag 0: Writing a 0 has no effect. 1: Writing a 1 will clear the TZCBCOSTFLAG[CBC8] bit. Reset type: SYSRSn
6	CBC7	R-0/W1S	0h	Clear CBC TZ7 Status Flag 0: Writing a 0 has no effect. 1: Writing a 1 will clear the TZCBCOSTFLAG[CBC7] bit. Reset type: SYSRSn
5	CBC6	R-0/W1S	0h	Clear CBC TZ6 Status Flag 0: Writing a 0 has no effect. 1: Writing a 1 will clear the TZCBCOSTFLAG[CBC6] bit. Reset type: SYSRSn
4	CBC5	R-0/W1S	0h	Clear CBC TZ5 Status Flag 0: Writing a 0 has no effect. 1: Writing a 1 will clear the TZCBCOSTFLAG[CBC5] bit. Reset type: SYSRSn
3	CBC4	R-0/W1S	0h	Clear CBC TZ4 Status Flag 0: Writing a 0 has no effect. 1: Writing a 1 will clear the TZCBCOSTFLAG[CBC4] bit. Reset type: SYSRSn
2	CBC3	R-0/W1S	0h	Clear CBC TZ3 Status Flag 0: Writing a 0 has no effect. 1: Writing a 1 will clear the TZCBCOSTFLAG[CBC3] bit. Reset type: SYSRSn
1	CBC2	R-0/W1S	0h	Clear CBC TZ2 Status Flag 0: Writing a 0 has no effect. 1: Writing a 1 will clear the TZCBCOSTFLAG[CBC2] bit. Reset type: SYSRSn
0	CBC1	R-0/W1S	0h	Clear CBC TZ1 Status Flag 0: Writing a 0 has no effect. 1: Writing a 1 will clear the TZCBCOSTFLAG[CBC1] bit. Reset type: SYSRSn

32 DBCTL Register (Offset = C0h) [Reset = 00000000h]

DBCTL is shown in Figure 19-71 and described in Table 19-47.

Return to the Summary Table.

Dead band control register

Figure 19-71. DBCTL Register

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

Table 19-79. DBCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-20	RESERVED	R-0	0h	Reserved
19-18	LOADREDMODE	R/W	0h	Shadow to Active load of DBRED register 00: Load on CTR = Zero: Time-base counter equal to zero (TBCTR = 0x0000) 01: Load on CTR = PRD: Time-base counter equal to period (TBCTR = TBPRD) 10: Load on either CTR = Zero or CTR = PRD 11: Freeze (no loads possible) Reset type: SYSRSn
17-16	LOADFEDMODE	R/W	0h	Shadow to Active load of DBFED register 00: Load on CTR = Zero: Time-base counter equal to zero (TBCTR = 0x0000) 01: Load on CTR = PRD: Time-base counter equal to period (TBCTR = TBPRD) 10: Load on either CTR = Zero or CTR = PRD 11: Freeze (no loads possible) Reset type: SYSRSn
15-9	RESERVED	R-0	0h	Reserved
8	DEDB_MODE	R/W	0h	Dead Band Dual-Edge B Mode Control (S8 switch) 0: Rising edge delay applied to InA/InB as selected by S4 switch (IN-MODE bits) on A signal path only. Falling edge delay applied to InA/InB as selected by S5 switch (INMODE bits) on B signal path only. 1: Rising edge delay and falling edge delay applied to source selected by S4 switch (INMODE bits) and output to B signal path only. Note: When this bit is set to 1, user should always either set OUT_MODE bits such that Apath = InA OR OUTSWAP bits such that OutA=Bpath otherwise, OutA will be invalid. Reset type: SYSRSn

ADVANCE INFORMATION

Table 19-79. DBCTL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
7-6	OUTSWAP	R/W	0h	Dead Band Output Swap Control Bit 7 controls the S6 switch and bit 6 controls the S7 switch. 00: OutA and OutB signals are as defined by OUT-MODE bits. 01: OutA = A-path as defined by OUT-MODE bits. OutB = A-path as defined by OUT-MODE bits (rising edge delay or delay-bypassed A signal path). 10: OutA = B-path as defined by OUT-MODE bits (falling edge delay or delay-bypassed B signal path). OutB = B-path as defined by OUT-MODE bits. 11: OutA = B-path as defined by OUT-MODE bits (falling edge delay or delay-bypassed B signal path). OutB = A-path as defined by OUT-MODE bits (rising edge delay or delay-bypassed A signal path). Reset type: SYSRSn
5-4	IN_MODE	R/W	0h	Dead-Band Input Mode Control Bit 5 controls the S5 switch and bit 4 controls the S4 switch shown. This allows you to select the input source to the falling-edge and rising-edge delay. To produce classical dead-band waveforms the default is PWMxA In is the source for both falling and rising-edge delays. 00: PWMxA In (from the action-qualifier) is the source for both falling-edge and rising-edge delay. 01: PWMxB In (from the action-qualifier) is the source for rising-edge delayed signal. PWMxA In (from the action-qualifier) is the source for falling-edge delayed signal. 10: PWMxA In (from the action-qualifier) is the source for rising-edge delayed signal. PWMxB In (from the action-qualifier) is the source for falling-edge delayed signal. 11: PWMxB In (from the action-qualifier) is the source for both rising-edge delay and falling-edge delayed signal. Reset type: SYSRSn
3-2	POLSEL	R/W	0h	Polarity Select Control Bit 3 controls the S3 switch and bit 2 controls the S2 switch. This allows you to selectively invert one of the delayed signals before it is sent out of the dead-band submodule. The following descriptions correspond to classical upper/lower switch control as found in one leg of a digital motor control inverter. These assume that DBCTL[OUT_MODE] = 1,1 and DBCTL[IN_MODE] = 0x0. Other enhanced modes are also possible, but not regarded as typical usage modes. 00: Active high (AH) mode. Neither PWMxA nor PWMxB is inverted (default). 01: Active low complementary (ALC) mode. PWMxA is inverted. 10: Active high complementary (AHC). PWMxB is inverted. 11: Active low (AL) mode. Both PWMxA and PWMxB are inverted. Reset type: SYSRSn
1-0	OUT_MODE	R/W	0h	Dead-Band Output Mode Control Bit 1 controls the S1 switch and bit 0 controls the S0 switch. 00: DBM is fully disabled or by-passed. In this mode the POLSEL and IN-MODE bits have no effect. 01: Apath = InA (delay is by-passed for A signal path) Bpath = FED (Falling Edge Delay in B signal path) 10: Apath = RED (Rising Edge Delay in A signal path) Bpath = InB (delay is by-passed for B signal path) 11: DBM is fully enabled (i.e. both RED and FED active) Reset type: SYSRSn

33 DBFED Register (Offset = D0h) [Reset = 00000000h]

DBFED is shown in Figure 19-72 and described in Table 19-48.

Return to the Summary Table.

Dead band fall edge delay

Figure 19-72. DBFED Register

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

Table 19-81. DBFED Register Field Descriptions

Bit	Field	Type	Reset	Description
31-14	RESERVED	R-0	0h	Reserved
13-0	DBFED	R/W	0h	Falling Edge Delay Count 14-bit counter Reset type: SYSRSn

ADVANCE INFORMATION

34 DBRED Register (Offset = D4h) [Reset = 00000000h]

DBRED is shown in Figure 19-73 and described in Table 19-49.

Return to the Summary Table.

Dead band rise edge delay

Figure 19-73. DBRED Register

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

Table 19-83. DBRED Register Field Descriptions

Bit	Field	Type	Reset	Description
31-14	RESERVED	R-0	0h	Reserved
13-0	DBRED	R/W	0h	Rising Edge Delay Count 14-bit counter Reset type: SYSRSn

ADVANCE INFORMATION

35 DBFEDS Register (Offset = D8h) [Reset = 00000000h]

DBFEDS is shown in Figure 19-74 and described in Table 19-50.

Return to the Summary Table.

Dead band fall edge delay shadow register

Figure 19-74. DBFEDS Register

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

Table 19-85. DBFEDS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-14	RESERVED	R-0	0h	Reserved
13-0	DBFEDS	R/W	0h	DBFED Shadow Register The value in the DBFEDS register is loaded into DBFED register when shadow to active load occurs. Reset type: SYSRSn

ADVANCE INFORMATION

36 DBREDS Register (Offset = DCh) [Reset = 00000000h]

DBREDS is shown in Figure 19-75 and described in Table 19-51.

Return to the Summary Table.

Dead band rise edge delay shadow register

Figure 19-75. DBREDS Register

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

Table 19-87. DBREDS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-14	RESERVED	R-0	0h	Reserved
13-0	DBREDS	R/W	0h	DBRED Shadow Register The value in the DBREDS register is loaded into DBRED register when shadow to active load occurs. Reset type: SYSRSn

ADVANCE INFORMATION

37 GLDCTL Register (Offset = F0h) [Reset = 00000000h]

GLDCTL is shown in Figure 19-76 and described in Table 19-52.

Return to the Summary Table.

Global load control register

Figure 19-76. GLDCTL Register

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

Table 19-89. GLDCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-6	RESERVED	R-0	0h	Reserved
5-4	GLDMODE	R/W	0h	Select global load event 00: CTR = Zero: Time-base counter equal to zero (TBCTR = 0x0000) 01: CTR = PRD: Time-base counter equal to period (TBCTR = TBPRD) 10: CTR = Zero or CTR = PRD 11: GLDOSHTCTL[GFRCLD] - Softwrae load Reset type: SYSRSn
3-2	RESERVED	R-0	0h	Reserved
1	OSHTMODE	R/W	0h	Global load one-shot enable 0: Disable global load one-shot 1: Enable global load one-shot Reset type: SYSRSn
0	GLD	R/W	0h	Global load enable 0: Disable global load 1: Enable global load Reset type: SYSRSn

38 GLDOSHTCTL Register (Offset = F4h) [Reset = 00000000h]

GLDOSHTCTL is shown in Figure 19-77 and described in Table 19-53.

Return to the Summary Table.

Global load one shot control register

Figure 19-77. GLDOSHTCTL Register

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

Table 19-91. GLDOSHTCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R-0	0h	Reserved
2	GFRCLD	R-0/W1S	0h	Force Load Event in One Shot Mode 0: Writing of 0 will be ignored. Always reads back a 0. 1: Force load event. This bit is intended to be used for testing and/or software force loading of the events in global load mode. Reset type: SYSRSn
1	OSHTCLR	R-0/W1S	0h	Clear One Shot latch 0: Writing of 0 will be ignored. Always reads back a 0. 1: Turns the one shot latch condition OFF. Reset type: SYSRSn
0	OSHTLD	R-0/W1S	0h	Enable Reload Event in One Shot Mode 0: Writing of 0 will be ignored. Always reads back a 0. 1: Turns the one shot latch condition ON. Upon occurrence of a chosen load strobe, one shadow to active reload occurs and the latch will be cleared. Hence writing 1 to this bit would allow one load strobe event to pass through and block further strobe events. Reset type: SYSRSn

39 GLDOSHTSTS Register (Offset = F8h) [Reset = 00000000h]

GLDOSHTSTS is shown in Figure 19-78 and described in Table 19-54.

Return to the Summary Table.

Global load one shot status register

Figure 19-78. GLDOSHTSTS Register

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

Table 19-93. GLDOSHTSTS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R-0	0h	Reserved
0	OSHTLATCH	R	0h	One shot latch output 0: one shot latch condition is OFF. 1: one shot latch condition is ON. Reset type: SYSRSn

ADVANCE INFORMATION

40 PWM1_CMPA Register (Offset = 100h) [Reset = 00000000h]

PWM1_CMPA is shown in Figure 19-79 and described in Table 19-55.

Return to the Summary Table.

PWM1 counter compare A register

Figure 19-79. PWM1_CMPA Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																PWM1_CMPA															
R-0-0h																R/W-0h															

Table 19-95. PWM1_CMPA Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R-0	0h	Reserved
15-0	PWM1_CMPA	R/W	0h	PWM1 Compare A register The value in the PWM1_CMPA register is continuously compared to the time-base counter (TBCTR). When the values are equal, the PWM1 counter-compare module generates a 'time-base counter equal to CMP A' event. This event is sent to the action-qualifier where it is qualified and converted it into one or more actions. These actions can be applied to either the PWM1A or the PWM1B output depending on the configuration of the PWM1_AQCTLA and PWM1_AQCTLB registers Shadowing of this register is enabled and disabled by the CMPCTL[PWM1_LOADAMODE] field. By default this register is shadowed. Reset type: SYSRSn

41 PWM1_CMPAS Register (Offset = 104h) [Reset = 00000000h]

PWM1_CMPAS is shown in Figure 19-80 and described in Table 19-56.

Return to the Summary Table.

PWM1 counter compare A shadow register

Figure 19-80. PWM1_CMPAS Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																PWM1_CMPAS															
R-0-0h																R/W-0h															

Table 19-97. PWM1_CMPAS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R-0	0h	Reserved
15-0	PWM1_CMPAS	R/W	0h	PWM1 Compare A Shadow Register The value in the PWM1_CMPAS register is loaded into PWM!_CMPA register when shadow to active load occurs. Reset type: SYSRSn

42 PWM1_CMPB Register (Offset = 108h) [Reset = 00000000h]

PWM1_CMPB is shown in Figure 19-81 and described in Table 19-57.

Return to the Summary Table.

PWM1 counter compare B register

Figure 19-81. PWM1_CMPB Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																PWM1_CMPB															
R-0-0h																R/W-0h															

Table 19-99. PWM1_CMPB Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R-0	0h	Reserved
15-0	PWM1_CMPB	R/W	0h	PWM1 Compare B register The value in the PWM1_CMPB register is continuously compared to the time-base counter (TBCTR). When the values are equal, the PWM1 counter-compare module generates a 'time-base counter equal to CMP B' event. This event is sent to the action-qualifier where it is qualified and converted it into one or more actions. These actions can be applied to either the PWM1A or the PWM1B output depending on the configuration of the PWM1_AQCTLA and PWM1_AQCTLB registers Shadowing of this register is enabled and disabled by the CMPCTL[PWM1_LOADBMODE] field. By default this register is shadowed. Reset type: SYSRSn

43 PWM1_CMPBS Register (Offset = 10Ch) [Reset = 00000000h]

PWM1_CMPBS is shown in Figure 19-82 and described in Table 19-58.

Return to the Summary Table.

PWM1 counter compare B shadow register

Figure 19-82. PWM1_CMPBS Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																PWM1_CMPBS															
R-0-0h																R/W-0h															

Table 19-101. PWM1_CMPBS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R-0	0h	Reserved
15-0	PWM1_CMPBS	R/W	0h	PWM1 Compare B Shadow Register The value in the PWM1 CMPBS register is loaded into PWM1 CMPB register when shadow to active load occurs. Reset type: SYSRSn

44 PWM1_AQCTLA Register (Offset = 120h) [Reset = 00000000h]

PWM1_AQCTLA is shown in Figure 19-83 and described in Table 19-59.

Return to the Summary Table.

PWM1 action qualifier A register

Figure 19-83. PWM1_AQCTLA Register

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

Table 19-103. PWM1_AQCTLA Register Field Descriptions

Bit	Field	Type	Reset	Description
31-12	RESERVED	R-0	0h	Reserved
11-10	CBD	R/W	0h	Action When TBCTR = PWM1_CMPB on Down Count Note: By definition, in count up-down mode when the counter equals 0 the direction is defined as 1 or counting up. 00: Do nothing (action disabled) 01: Clear: force PWM1A output low. 10: Set: force PWM1A output high. 11: Toggle PWM1A output: low output signal will be forced high, and a high signal will be forced low. Reset type: SYSRSn
9-8	CBU	R/W	0h	Action When TBCTR = PWM1_CMPB on Up Count Note: By definition, in count up-down mode when the counter equals 0 the direction is defined as 1 or counting up. 00: Do nothing (action disabled) 01: Clear: force PWM1A output low. 10: Set: force PWM1A output high. 11: Toggle PWM1A output: low output signal will be forced high, and a high signal will be forced low. Reset type: SYSRSn
7-6	CAD	R/W	0h	Action When TBCTR = PWM1_CMPA on Down Count Note: By definition, in count up-down mode when the counter equals 0 the direction is defined as 1 or counting up. 00: Do nothing (action disabled) 01: Clear: force PWM1A output low. 10: Set: force PWM1A output high. 11: Toggle PWM1A output: low output signal will be forced high, and a high signal will be forced low. Reset type: SYSRSn
5-4	CAU	R/W	0h	Action When TBCTR = PWM1_CMPA on Up Count Note: By definition, in count up-down mode when the counter equals 0 the direction is defined as 1 or counting up. 00: Do nothing (action disabled) 01: Clear: force PWM1A output low. 10: Set: force PWM1A output high. 11: Toggle PWM1A output: low output signal will be forced high, and a high signal will be forced low. Reset type: SYSRSn

Table 19-103. PWM1_AQCTLA Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3-2	PRD	R/W	0h	Action When TBCTR = TBPRD 00: Do nothing (action disabled) 01: Clear: force PWM1A output low. 10: Set: force PWM1A output high. 11: Toggle PWM1A output: low output signal will be forced high, and a high signal will be forced low. Reset type: SYSRSn
1-0	ZRO	R/W	0h	Action When TBCTR = 0 00: Do nothing (action disabled) 01: Clear: force PWM1A output low. 10: Set: force PWM1A output high. 11: Toggle PWM1A output: low output signal will be forced high, and a high signal will be forced low. Reset type: SYSRSn

45 PWM1_AQCTLAS Register (Offset = 124h) [Reset = 00000000h]

PWM1_AQCTLAS is shown in Figure 19-84 and described in Table 19-60.

Return to the Summary Table.

PWM1 action qualifier A shadow register

Figure 19-84. PWM1_AQCTLAS Register

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

Table 19-105. PWM1_AQCTLAS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-12	RESERVED	R-0	0h	Reserved
11-10	CBD	R/W	0h	Action When TBCTR = PWM1_CMPB on Down Count Note: By definition, in count up-down mode when the counter equals 0 the direction is defined as 1 or counting up. 00: Do nothing (action disabled) 01: Clear: force PWM1A output low. 10: Set: force PWM1A output high. 11: Toggle PWM1A output: low output signal will be forced high, and a high signal will be forced low. Reset type: SYSRSn
9-8	CBU	R/W	0h	Action When TBCTR = PWM1_CMPB on Up Count Note: By definition, in count up-down mode when the counter equals 0 the direction is defined as 1 or counting up. 00: Do nothing (action disabled) 01: Clear: force PWM1A output low. 10: Set: force PWM1A output high. 11: Toggle PWM1A output: low output signal will be forced high, and a high signal will be forced low. Reset type: SYSRSn
7-6	CAD	R/W	0h	Action When TBCTR = PWM1_CMPA on Down Count Note: By definition, in count up-down mode when the counter equals 0 the direction is defined as 1 or counting up. 00: Do nothing (action disabled) 01: Clear: force PWM1A output low. 10: Set: force PWM1A output high. 11: Toggle PWM1A output: low output signal will be forced high, and a high signal will be forced low. Reset type: SYSRSn
5-4	CAU	R/W	0h	Action When TBCTR = PWM1_CMPA on Up Count Note: By definition, in count up-down mode when the counter equals 0 the direction is defined as 1 or counting up. 00: Do nothing (action disabled) 01: Clear: force PWM1A output low. 10: Set: force PWM1A output high. 11: Toggle PWM1A output: low output signal will be forced high, and a high signal will be forced low. Reset type: SYSRSn

Table 19-105. PWM1_AQCTLAS Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3-2	PRD	R/W	0h	Action When TBCTR = TBPRD 00: Do nothing (action disabled) 01: Clear: force PWM1A output low. 10: Set: force PWM1A output high. 11: Toggle PWM1A output: low output signal will be forced high, and a high signal will be forced low. Reset type: SYSRSn
1-0	ZRO	R/W	0h	Action When TBCTR = 0 00: Do nothing (action disabled) 01: Clear: force PWM1A output low. 10: Set: force PWM1A output high. 11: Toggle PWM1A output: low output signal will be forced high, and a high signal will be forced low. Reset type: SYSRSn

46 PWM1_AQCTLB Register (Offset = 128h) [Reset = 00000000h]

PWM1_AQCTLB is shown in Figure 19-85 and described in Table 19-61.

Return to the Summary Table.

PWM1 action qualifier B register

Figure 19-85. PWM1_AQCTLB Register

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

Table 19-107. PWM1_AQCTLB Register Field Descriptions

Bit	Field	Type	Reset	Description
31-12	RESERVED	R-0	0h	Reserved
11-10	CBD	R/W	0h	Action When TBCTR = PWM1_CMPB on Down Count Note: By definition, in count up-down mode when the counter equals 0 the direction is defined as 1 or counting up. 00: Do nothing (action disabled) 01: Clear: force PWM1B output low. 10: Set: force PWM1B output high. 11: Toggle PWM1B output: low output signal will be forced high, and a high signal will be forced low. Reset type: SYSRSn
9-8	CBU	R/W	0h	Action When TBCTR = PWM1_CMPB on Up Count Note: By definition, in count up-down mode when the counter equals 0 the direction is defined as 1 or counting up. 00: Do nothing (action disabled) 01: Clear: force PWM1B output low. 10: Set: force PWM1B output high. 11: Toggle PWM1B output: low output signal will be forced high, and a high signal will be forced low. Reset type: SYSRSn
7-6	CAD	R/W	0h	Action When TBCTR = PWM1_CMPA on Down Count Note: By definition, in count up-down mode when the counter equals 0 the direction is defined as 1 or counting up. 00: Do nothing (action disabled) 01: Clear: force PWM1B output low. 10: Set: force PWM1B output high. 11: Toggle PWM1B output: low output signal will be forced high, and a high signal will be forced low. Reset type: SYSRSn
5-4	CAU	R/W	0h	Action When TBCTR = PWM1_CMPA on Up Count Note: By definition, in count up-down mode when the counter equals 0 the direction is defined as 1 or counting up. 00: Do nothing (action disabled) 01: Clear: force PWM1B output low. 10: Set: force PWM1B output high. 11: Toggle PWM1B output: low output signal will be forced high, and a high signal will be forced low. Reset type: SYSRSn

Table 19-107. PWM1_AQCTLB Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3-2	PRD	R/W	0h	Action When TBCTR = TBPRD 00: Do nothing (action disabled) 01: Clear: force PWM1B output low. 10: Set: force PWM1B output high. 11: Toggle PWM1B output: low output signal will be forced high, and a high signal will be forced low. Reset type: SYSRSn
1-0	ZRO	R/W	0h	Action When TBCTR = 0 00: Do nothing (action disabled) 01: Clear: force PWM1B output low. 10: Set: force PWM1B output high. 11: Toggle PWM1B output: low output signal will be forced high, and a high signal will be forced low. Reset type: SYSRSn

47 PWM1_AQCTLBS Register (Offset = 12Ch) [Reset = 00000000h]

PWM1_AQCTLBS is shown in Figure 19-86 and described in Table 19-62.

Return to the Summary Table.

PWM1 action qualifier B shadow register

Figure 19-86. PWM1_AQCTLBS Register

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

Table 19-109. PWM1_AQCTLBS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-12	RESERVED	R-0	0h	Reserved
11-10	CBD	R/W	0h	Action When TBCTR = PWM1_CMPB on Down Count Note: By definition, in count up-down mode when the counter equals 0 the direction is defined as 1 or counting up. 00: Do nothing (action disabled) 01: Clear: force PWM1B output low. 10: Set: force PWM1B output high. 11: Toggle PWM1B output: low output signal will be forced high, and a high signal will be forced low. Reset type: SYSRSn
9-8	CBU	R/W	0h	Action When TBCTR = PWM1_CMPB on Up Count Note: By definition, in count up-down mode when the counter equals 0 the direction is defined as 1 or counting up. 00: Do nothing (action disabled) 01: Clear: force PWM1B output low. 10: Set: force PWM1B output high. 11: Toggle PWM1B output: low output signal will be forced high, and a high signal will be forced low. Reset type: SYSRSn
7-6	CAD	R/W	0h	Action When TBCTR = PWM1_CMPA on Down Count Note: By definition, in count up-down mode when the counter equals 0 the direction is defined as 1 or counting up. 00: Do nothing (action disabled) 01: Clear: force PWM1B output low. 10: Set: force PWM1B output high. 11: Toggle PWM1B output: low output signal will be forced high, and a high signal will be forced low. Reset type: SYSRSn
5-4	CAU	R/W	0h	Action When TBCTR = PWM1_CMPA on Up Count Note: By definition, in count up-down mode when the counter equals 0 the direction is defined as 1 or counting up. 00: Do nothing (action disabled) 01: Clear: force PWM1B output low. 10: Set: force PWM1B output high. 11: Toggle PWM1B output: low output signal will be forced high, and a high signal will be forced low. Reset type: SYSRSn

Table 19-109. PWM1_AQCTLBS Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3-2	PRD	R/W	0h	Action When TBCTR = TBPRD 00: Do nothing (action disabled) 01: Clear: force PWM1B output low. 10: Set: force PWM1B output high. 11: Toggle PWM1B output: low output signal will be forced high, and a high signal will be forced low. Reset type: SYSRSn
1-0	ZRO	R/W	0h	Action When TBCTR = 0 00: Do nothing (action disabled) 01: Clear: force PWM1B output low. 10: Set: force PWM1B output high. 11: Toggle PWM1B output: low output signal will be forced high, and a high signal will be forced low. Reset type: SYSRSn

48 PWM1_AQSFRC Register (Offset = 130h) [Reset = 00000000h]

PWM1_AQSFRC is shown in Figure 19-87 and described in Table 19-63.

Return to the Summary Table.

PWM1 action qualifier software force

Figure 19-87. PWM1_AQSFRC Register

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

Table 19-111. PWM1_AQSFRC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-7	RESERVED	R-0	0h	Reserved
6-4	PWMB	R/W	0h	Action qualifier software force on PWMB 000: Does nothing (softwrae force disabled) 001: Forces a continuous low on output B 010: Forces a continuous high on output B 011: Does nothing (softwrae force disabled) 100: Does nothing (softwrae force disabled) 101: Clear (low) when PWM1_AQOTSFRC[PWMB] = '1'. 110: Set (high) when PWM1_AQOTSFRC[PWMB] = '1'. 111: Toggle (Low -> High, High -> Low) when PWM1_AQOTSFRC[PWMB] = '1'. Reset type: SYSRSn
3	RESERVED	R-0	0h	Reserved
2-0	PWMA	R/W	0h	Action qualifier software force on PWMA 000: Does nothing (softwrae force disabled) 001: Forces a continuous low on output A 010: Forces a continuous high on output A 011: Does nothing (softwrae force disabled) 100: Does nothing (softwrae force disabled) 101: Clear (low) when PWM1_AQOTSFRC[PWMA] = '1'. 110: Set (high) when PWM1_AQOTSFRC[PWMA] = '1'. 111: Toggle (Low -> High, High -> Low) when PWM1_AQOTSFRC[PWMA] = '1'. Reset type: SYSRSn

49 PWM1_AQOTSFRC Register (Offset = 134h) [Reset = 00000000h]

PWM1_AQOTSFRC is shown in Figure 19-88 and described in Table 19-64.

Return to the Summary Table.

PWM1 action qualifier one time software force

Figure 19-88. PWM1_AQOTSFRC Register

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

Table 19-113. PWM1_AQOTSFRC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-5	RESERVED	R-0	0h	Reserved
4	PWMB	R-0/W1S	0h	Action qualifier one time software force on PWMB 0: Writing a 0 has no effect. Always reads back a 0. This bit is auto cleared once a write to this register is complete (i.e., a forced event is initiated.). This is a one-shot forced event. It can be overridden by another subsequent event on PWMB. 1: Initiates a single software forced event Reset type: SYSRSn
3-1	RESERVED	R-0	0h	Reserved
0	PWMA	R-0/W1S	0h	Action qualifier one time software force on PWMA 0: Writing a 0 has no effect. Always reads back a 0. This bit is auto cleared once a write to this register is complete (i.e., a forced event is initiated.). This is a one-shot forced event. It can be overridden by another subsequent event on PWMA. 1: Initiates a single software forced event Reset type: SYSRSn

50 PWM2_CMPA Register (Offset = 300h) [Reset = 00000000h]

PWM2_CMPA is shown in Figure 19-89 and described in Table 19-65.

Return to the Summary Table.

PWM2 counter compare A register

Figure 19-89. PWM2_CMPA Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																PWM2_CMPA															
R-0-0h																R/W-0h															

Table 19-115. PWM2_CMPA Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R-0	0h	Reserved
15-0	PWM2_CMPA	R/W	0h	PWM2 Compare A register The value in the PWM2_CMPA register is continuously compared to the time-base counter (TBCTR). When the values are equal, the PWM2 counter-compare module generates a 'time-base counter equal to CMP A' event. This event is sent to the action-qualifier where it is qualified and converted it into one or more actions. These actions can be applied to either the PWM2A or the PWM2B output depending on the configuration of the PWM2_AQCTLA and PWM2_AQCTLB registers Shadowing of this register is enabled and disabled by the CMPCTL[PWM2_LOADAMODE] field. By default this register is shadowed. Reset type: SYSRSn

51 PWM2_CMPAS Register (Offset = 304h) [Reset = 00000000h]

PWM2_CMPAS is shown in Figure 19-90 and described in Table 19-66.

Return to the Summary Table.

PWM2 counter compare A shadow register

Figure 19-90. PWM2_CMPAS Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																PWM2_CMPAS															
R-0-0h																R/W-0h															

Table 19-117. PWM2_CMPAS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R-0	0h	Reserved
15-0	PWM2_CMPAS	R/W	0h	PWM2 Compare A Shadow Register The value in the PWM2_CMPAS register is loaded into PWM2_CMPA register when shadow to active load occurs. Reset type: SYSRSn

52 PWM2_CMPB Register (Offset = 308h) [Reset = 00000000h]

PWM2_CMPB is shown in Figure 19-91 and described in Table 19-67.

Return to the Summary Table.

PWM2 counter compare B register

Figure 19-91. PWM2_CMPB Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																PWM2_CMPB															
R-0-0h																R/W-0h															

Table 19-119. PWM2_CMPB Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R-0	0h	Reserved
15-0	PWM2_CMPB	R/W	0h	PWM2 Compare B register The value in the PWM2_CMPB register is continuously compared to the time-base counter (TBCTR). When the values are equal, the PWM2 counter-compare module generates a 'time-base counter equal to CMP B' event. This event is sent to the action-qualifier where it is qualified and converted it into one or more actions. These actions can be applied to either the PWM2A or the PWM2B output depending on the configuration of the PWM2_AQCTLA and PWM2_AQCTLB registers Shadowing of this register is enabled and disabled by the CMPCTL[PWM2_LOADBMODE] field. By default this register is shadowed. Reset type: SYSRSn

53 PWM2_CMPBS Register (Offset = 30Ch) [Reset = 00000000h]

PWM2_CMPBS is shown in Figure 19-92 and described in Table 19-68.

Return to the Summary Table.

PWM2 counter compare B shadow register

Figure 19-92. PWM2_CMPBS Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																PWM2_CMPBS															
R-0-0h																R/W-0h															

Table 19-121. PWM2_CMPBS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R-0	0h	Reserved
15-0	PWM2_CMPBS	R/W	0h	PWM2 Compare B Shadow Register The value in the PWM2_CMPBS register is loaded into PWM2_CMPB register when shadow to active load occurs. Reset type: SYSRSn

54 PWM2_AQCTLA Register (Offset = 320h) [Reset = 00000000h]

PWM2_AQCTLA is shown in Figure 19-93 and described in Table 19-69.

Return to the Summary Table.

PWM2 action qualifier A register

Figure 19-93. PWM2_AQCTLA Register

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

Table 19-123. PWM2_AQCTLA Register Field Descriptions

Bit	Field	Type	Reset	Description
31-12	RESERVED	R-0	0h	Reserved
11-10	CBD	R/W	0h	Action When TBCTR = PWM2_CMPB on Down Count Note: By definition, in count up-down mode when the counter equals 0 the direction is defined as 1 or counting up. 00: Do nothing (action disabled) 01: Clear: force PWM2A output low. 10: Set: force PWM2A output high. 11: Toggle PWM2A output: low output signal will be forced high, and a high signal will be forced low. Reset type: SYSRSn
9-8	CBU	R/W	0h	Action When TBCTR = PWM2_CMPB on Up Count Note: By definition, in count up-down mode when the counter equals 0 the direction is defined as 1 or counting up. 00: Do nothing (action disabled) 01: Clear: force PWM2A output low. 10: Set: force PWM2A output high. 11: Toggle PWM2A output: low output signal will be forced high, and a high signal will be forced low. Reset type: SYSRSn
7-6	CAD	R/W	0h	Action When TBCTR = PWM2_CMPA on Down Count Note: By definition, in count up-down mode when the counter equals 0 the direction is defined as 1 or counting up. 00: Do nothing (action disabled) 01: Clear: force PWM2A output low. 10: Set: force PWM2A output high. 11: Toggle PWM2A output: low output signal will be forced high, and a high signal will be forced low. Reset type: SYSRSn
5-4	CAU	R/W	0h	Action When TBCTR = PWM2_CMPA on Up Count Note: By definition, in count up-down mode when the counter equals 0 the direction is defined as 1 or counting up. 00: Do nothing (action disabled) 01: Clear: force PWM2A output low. 10: Set: force PWM2A output high. 11: Toggle PWM2A output: low output signal will be forced high, and a high signal will be forced low. Reset type: SYSRSn

Table 19-123. PWM2_AQCTLA Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3-2	PRD	R/W	0h	Action When TBCTR = TBPRD 00: Do nothing (action disabled) 01: Clear: force PWM2A output low. 10: Set: force PWM2A output high. 11: Toggle PWM2A output: low output signal will be forced high, and a high signal will be forced low. Reset type: SYSRSn
1-0	ZRO	R/W	0h	Action When TBCTR = 0 00: Do nothing (action disabled) 01: Clear: force PWM2A output low. 10: Set: force PWM2A output high. 11: Toggle PWM2A output: low output signal will be forced high, and a high signal will be forced low. Reset type: SYSRSn

55 PWM2_AQCTLAS Register (Offset = 324h) [Reset = 00000000h]

PWM2_AQCTLAS is shown in Figure 19-94 and described in Table 19-70.

Return to the Summary Table.

PWM2 action qualifier A shadow register

Figure 19-94. PWM2_AQCTLAS Register

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

Table 19-125. PWM2_AQCTLAS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-12	RESERVED	R-0	0h	Reserved
11-10	CBD	R/W	0h	Action When TBCTR = PWM2_CMPB on Down Count Note: By definition, in count up-down mode when the counter equals 0 the direction is defined as 1 or counting up. 00: Do nothing (action disabled) 01: Clear: force PWM2A output low. 10: Set: force PWM2A output high. 11: Toggle PWM2A output: low output signal will be forced high, and a high signal will be forced low. Reset type: SYSRSn
9-8	CBU	R/W	0h	Action When TBCTR = PWM2_CMPB on Up Count Note: By definition, in count up-down mode when the counter equals 0 the direction is defined as 1 or counting up. 00: Do nothing (action disabled) 01: Clear: force PWM2A output low. 10: Set: force PWM2A output high. 11: Toggle PWM2A output: low output signal will be forced high, and a high signal will be forced low. Reset type: SYSRSn
7-6	CAD	R/W	0h	Action When TBCTR = PWM2_CMPA on Down Count Note: By definition, in count up-down mode when the counter equals 0 the direction is defined as 1 or counting up. 00: Do nothing (action disabled) 01: Clear: force PWM2A output low. 10: Set: force PWM2A output high. 11: Toggle PWM2A output: low output signal will be forced high, and a high signal will be forced low. Reset type: SYSRSn
5-4	CAU	R/W	0h	Action When TBCTR = PWM2_CMPA on Up Count Note: By definition, in count up-down mode when the counter equals 0 the direction is defined as 1 or counting up. 00: Do nothing (action disabled) 01: Clear: force PWM2A output low. 10: Set: force PWM2A output high. 11: Toggle PWM2A output: low output signal will be forced high, and a high signal will be forced low. Reset type: SYSRSn

Table 19-125. PWM2_AQCTLAS Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3-2	PRD	R/W	0h	Action When TBCTR = TBPRD 00: Do nothing (action disabled) 01: Clear: force PWM2A output low. 10: Set: force PWM2A output high. 11: Toggle PWM2A output: low output signal will be forced high, and a high signal will be forced low. Reset type: SYSRSn
1-0	ZRO	R/W	0h	Action When TBCTR = 0 00: Do nothing (action disabled) 01: Clear: force PWM2A output low. 10: Set: force PWM2A output high. 11: Toggle PWM2A output: low output signal will be forced high, and a high signal will be forced low. Reset type: SYSRSn

56 PWM2_AQCTLB Register (Offset = 328h) [Reset = 00000000h]

PWM2_AQCTLB is shown in Figure 19-95 and described in Table 19-71.

Return to the Summary Table.

PWM2 action qualifier B register

Figure 19-95. PWM2_AQCTLB Register

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

Table 19-127. PWM2_AQCTLB Register Field Descriptions

Bit	Field	Type	Reset	Description
31-12	RESERVED	R-0	0h	Reserved
11-10	CBD	R/W	0h	Action When TBCTR = PWM2_CMPB on Down Count Note: By definition, in count up-down mode when the counter equals 0 the direction is defined as 1 or counting up. 00: Do nothing (action disabled) 01: Clear: force PWM2B output low. 10: Set: force PWM2B output high. 11: Toggle PWM2B output: low output signal will be forced high, and a high signal will be forced low. Reset type: SYSRSn
9-8	CBU	R/W	0h	Action When TBCTR = PWM2_CMPB on Up Count Note: By definition, in count up-down mode when the counter equals 0 the direction is defined as 1 or counting up. 00: Do nothing (action disabled) 01: Clear: force PWM2B output low. 10: Set: force PWM2B output high. 11: Toggle PWM2B output: low output signal will be forced high, and a high signal will be forced low. Reset type: SYSRSn
7-6	CAD	R/W	0h	Action When TBCTR = PWM2_CMPA on Down Count Note: By definition, in count up-down mode when the counter equals 0 the direction is defined as 1 or counting up. 00: Do nothing (action disabled) 01: Clear: force PWM2B output low. 10: Set: force PWM2B output high. 11: Toggle PWM2B output: low output signal will be forced high, and a high signal will be forced low. Reset type: SYSRSn
5-4	CAU	R/W	0h	Action When TBCTR = PWM2_CMPA on Up Count Note: By definition, in count up-down mode when the counter equals 0 the direction is defined as 1 or counting up. 00: Do nothing (action disabled) 01: Clear: force PWM2B output low. 10: Set: force PWM2B output high. 11: Toggle PWM2B output: low output signal will be forced high, and a high signal will be forced low. Reset type: SYSRSn

Table 19-127. PWM2_AQCTLB Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3-2	PRD	R/W	0h	Action When TBCTR = TBPRD 00: Do nothing (action disabled) 01: Clear: force PWM2B output low. 10: Set: force PWM2B output high. 11: Toggle PWM2B output: low output signal will be forced high, and a high signal will be forced low. Reset type: SYSRSn
1-0	ZRO	R/W	0h	Action When TBCTR = 0 00: Do nothing (action disabled) 01: Clear: force PWM2B output low. 10: Set: force PWM2B output high. 11: Toggle PWM2B output: low output signal will be forced high, and a high signal will be forced low. Reset type: SYSRSn

57 PWM2_AQCTLBS Register (Offset = 32Ch) [Reset = 00000000h]

PWM2_AQCTLBS is shown in Figure 19-96 and described in Table 19-72.

Return to the Summary Table.

PWM2 action qualifier B shadow register

Figure 19-96. PWM2_AQCTLBS Register

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

Table 19-129. PWM2_AQCTLBS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-12	RESERVED	R-0	0h	Reserved
11-10	CBD	R/W	0h	Action When TBCTR = PWM2_CMPB on Down Count Note: By definition, in count up-down mode when the counter equals 0 the direction is defined as 1 or counting up. 00: Do nothing (action disabled) 01: Clear: force PWM2B output low. 10: Set: force PWM2B output high. 11: Toggle PWM2B output: low output signal will be forced high, and a high signal will be forced low. Reset type: SYSRSn
9-8	CBU	R/W	0h	Action When TBCTR = PWM2_CMPB on Up Count Note: By definition, in count up-down mode when the counter equals 0 the direction is defined as 1 or counting up. 00: Do nothing (action disabled) 01: Clear: force PWM2B output low. 10: Set: force PWM2B output high. 11: Toggle PWM2B output: low output signal will be forced high, and a high signal will be forced low. Reset type: SYSRSn
7-6	CAD	R/W	0h	Action When TBCTR = PWM2_CMPA on Down Count Note: By definition, in count up-down mode when the counter equals 0 the direction is defined as 1 or counting up. 00: Do nothing (action disabled) 01: Clear: force PWM2B output low. 10: Set: force PWM2B output high. 11: Toggle PWM2B output: low output signal will be forced high, and a high signal will be forced low. Reset type: SYSRSn
5-4	CAU	R/W	0h	Action When TBCTR = PWM2_CMPA on Up Count Note: By definition, in count up-down mode when the counter equals 0 the direction is defined as 1 or counting up. 00: Do nothing (action disabled) 01: Clear: force PWM2B output low. 10: Set: force PWM2B output high. 11: Toggle PWM2B output: low output signal will be forced high, and a high signal will be forced low. Reset type: SYSRSn

Table 19-129. PWM2_AQCTLBS Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3-2	PRD	R/W	0h	Action When TBCTR = TBPRD 00: Do nothing (action disabled) 01: Clear: force PWM2B output low. 10: Set: force PWM2B output high. 11: Toggle PWM2B output: low output signal will be forced high, and a high signal will be forced low. Reset type: SYSRSn
1-0	ZRO	R/W	0h	Action When TBCTR = 0 00: Do nothing (action disabled) 01: Clear: force PWM2B output low. 10: Set: force PWM2B output high. 11: Toggle PWM2B output: low output signal will be forced high, and a high signal will be forced low. Reset type: SYSRSn

58 PWM2_AQSFRC Register (Offset = 330h) [Reset = 00000000h]

PWM2_AQSFRC is shown in Figure 19-97 and described in Table 19-73.

Return to the Summary Table.

PWM2 action qualifier software force

Figure 19-97. PWM2_AQSFRC Register

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

Table 19-131. PWM2_AQSFRC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-7	RESERVED	R-0	0h	Reserved
6-4	PWMB	R/W	0h	Action qualifier software force on PWMB 000: Does nothing (softwrae force disabled) 001: Forces a continuous low on output B 010: Forces a continuous high on output B 011: Does nothing (softwrae force disabled) 100: Does nothing (softwrae force disabled) 101: Clear (low) when PWM2_AQOTSFRC[PWMB] = '1'. 110: Set (high) when PWM2_AQOTSFRC[PWMB] = '1'. 111: Toggle (Low -> High, High -> Low) when PWM2_AQOTSFRC[PWMB] = '1'. Reset type: SYSRSn
3	RESERVED	R-0	0h	Reserved
2-0	PWMA	R/W	0h	Action qualifier software force on PWMA 000: Does nothing (softwrae force disabled) 001: Forces a continuous low on output A 010: Forces a continuous high on output A 011: Does nothing (softwrae force disabled) 100: Does nothing (softwrae force disabled) 101: Clear (low) when PWM2_AQOTSFRC[PWMA] = '1'. 110: Set (high) when PWM2_AQOTSFRC[PWMA] = '1'. 111: Toggle (Low -> High, High -> Low) when PWM2_AQOTSFRC[PWMA] = '1'. Reset type: SYSRSn

59 PWM2_AQOTSFRC Register (Offset = 334h) [Reset = 00000000h]

PWM2_AQOTSFRC is shown in Figure 19-98 and described in Table 19-74.

Return to the Summary Table.

PWM2 action qualifier one time software force

Figure 19-98. PWM2_AQOTSFRC Register

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

Table 19-133. PWM2_AQOTSFRC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-5	RESERVED	R-0	0h	Reserved
4	PWMB	R-0/W1S	0h	Action qualifier one time software force on PWMB 0: Writing a 0 has no effect. Always reads back a 0. This bit is auto cleared once a write to this register is complete (i.e., a forced event is initiated.). This is a one-shot forced event. It can be overridden by another subsequent event on PWMB. 1: Initiates a single software forced event Reset type: SYSRSn
3-1	RESERVED	R-0	0h	Reserved
0	PWMA	R-0/W1S	0h	Action qualifier one time software force on PWMA 0: Writing a 0 has no effect. Always reads back a 0. This bit is auto cleared once a write to this register is complete (i.e., a forced event is initiated.). This is a one-shot forced event. It can be overridden by another subsequent event on PWMA. 1: Initiates a single software forced event Reset type: SYSRSn

60 PWM3_CMPA Register (Offset = 500h) [Reset = 00000000h]

PWM3_CMPA is shown in Figure 19-99 and described in Table 19-75.

Return to the Summary Table.

PWM3 counter compare A register

Figure 19-99. PWM3_CMPA Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																PWM3_CMPA															
R-0-0h																R/W-0h															

Table 19-135. PWM3_CMPA Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R-0	0h	Reserved
15-0	PWM3_CMPA	R/W	0h	PWM3 Compare A register The value in the PWM3_CMPA register is continuously compared to the time-base counter (TBCTR). When the values are equal, the PWM3 counter-compare module generates a 'time-base counter equal to CMP A' event. This event is sent to the action-qualifier where it is qualified and converted it into one or more actions. These actions can be applied to either the PWM3A or the PWM3B output depending on the configuration of the PWM3_AQCTLA and PWM3_AQCTLB registers Shadowing of this register is enabled and disabled by the CMPCTL[PWM3_LOADAMODE] field. By default this register is shadowed. Reset type: SYSRSn

61 PWM3_CMPAS Register (Offset = 504h) [Reset = 00000000h]

PWM3_CMPAS is shown in Figure 19-100 and described in Table 19-76.

Return to the Summary Table.

PWM3 counter compare A shadow register

Figure 19-100. PWM3_CMPAS Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																PWM3_CMPAS															
R-0-0h																R/W-0h															

Table 19-137. PWM3_CMPAS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R-0	0h	Reserved
15-0	PWM3_CMPAS	R/W	0h	PWM3 Compare A Shadow Register The value in the PWM3_CMPAS register is loaded into PWM3_CMPA register when shadow to active load occurs. Reset type: SYSRSn

62 PWM3_CMPB Register (Offset = 508h) [Reset = 00000000h]

PWM3_CMPB is shown in Figure 19-101 and described in Table 19-77.

Return to the Summary Table.

PWM3 counter compare B register

Figure 19-101. PWM3_CMPB Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																PWM3_CMPB															
R-0-0h																R/W-0h															

Table 19-139. PWM3_CMPB Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R-0	0h	Reserved
15-0	PWM3_CMPB	R/W	0h	PWM3 Compare B register The value in the PWM3_CMPB register is continuously compared to the time-base counter (TBCTR). When the values are equal, the PWM3 counter-compare module generates a 'time-base counter equal to CMP B' event. This event is sent to the action-qualifier where it is qualified and converted it into one or more actions. These actions can be applied to either the PWM3A or the PWM3B output depending on the configuration of the PWM3_AQCTLA and PWM3_AQCTLB registers Shadowing of this register is enabled and disabled by the CMPCTL[PWM3_LOADBMODE] field. By default this register is shadowed. Reset type: SYSRSn

63 PWM3_CMPBS Register (Offset = 50Ch) [Reset = 00000000h]

PWM3_CMPBS is shown in Figure 19-102 and described in Table 19-78.

Return to the Summary Table.

PWM3 counter compare B shadow register

Figure 19-102. PWM3_CMPBS Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																PWM3_CMPBS															
R-0-0h																R/W-0h															

Table 19-141. PWM3_CMPBS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R-0	0h	Reserved
15-0	PWM3_CMPBS	R/W	0h	PWM3 Compare B Shadow Register The value in the PWM3 CMPBS register is loaded into PWM3 CMPB register when shadow to active load occurs. Reset type: SYSRSn

64 PWM3_AQCTLA Register (Offset = 520h) [Reset = 00000000h]

PWM3_AQCTLA is shown in Figure 19-103 and described in Table 19-79.

Return to the Summary Table.

PWM3 action qualifier A register

Figure 19-103. PWM3_AQCTLA Register

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

Table 19-143. PWM3_AQCTLA Register Field Descriptions

Bit	Field	Type	Reset	Description
31-12	RESERVED	R-0	0h	Reserved
11-10	CBD	R/W	0h	Action When TBCTR = PWM3_CMPB on Down Count Note: By definition, in count up-down mode when the counter equals 0 the direction is defined as 1 or counting up. 00: Do nothing (action disabled) 01: Clear: force PWM3A output low. 10: Set: force PWM3A output high. 11: Toggle PWM3A output: low output signal will be forced high, and a high signal will be forced low. Reset type: SYSRSn
9-8	CBU	R/W	0h	Action When TBCTR = PWM3_CMPB on Up Count Note: By definition, in count up-down mode when the counter equals 0 the direction is defined as 1 or counting up. 00: Do nothing (action disabled) 01: Clear: force PWM3A output low. 10: Set: force PWM3A output high. 11: Toggle PWM3A output: low output signal will be forced high, and a high signal will be forced low. Reset type: SYSRSn
7-6	CAD	R/W	0h	Action When TBCTR = PWM3_CMPA on Down Count Note: By definition, in count up-down mode when the counter equals 0 the direction is defined as 1 or counting up. 00: Do nothing (action disabled) 01: Clear: force PWM3A output low. 10: Set: force PWM3A output high. 11: Toggle PWM3A output: low output signal will be forced high, and a high signal will be forced low. Reset type: SYSRSn
5-4	CAU	R/W	0h	Action When TBCTR = PWM3_CMPA on Up Count Note: By definition, in count up-down mode when the counter equals 0 the direction is defined as 1 or counting up. 00: Do nothing (action disabled) 01: Clear: force PWM3A output low. 10: Set: force PWM3A output high. 11: Toggle PWM3A output: low output signal will be forced high, and a high signal will be forced low. Reset type: SYSRSn

Table 19-143. PWM3_AQCTLA Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3-2	PRD	R/W	0h	Action When TBCTR = TBPRD 00: Do nothing (action disabled) 01: Clear: force PWM3A output low. 10: Set: force PWM3A output high. 11: Toggle PWM3A output: low output signal will be forced high, and a high signal will be forced low. Reset type: SYSRSn
1-0	ZRO	R/W	0h	Action When TBCTR = 0 00: Do nothing (action disabled) 01: Clear: force PWM3A output low. 10: Set: force PWM3A output high. 11: Toggle PWM3A output: low output signal will be forced high, and a high signal will be forced low. Reset type: SYSRSn

65 PWM3_AQCTLAS Register (Offset = 524h) [Reset = 00000000h]

PWM3_AQCTLAS is shown in Figure 19-104 and described in Table 19-80.

Return to the Summary Table.

PWM3 action qualifier A shadow register

Figure 19-104. PWM3_AQCTLAS Register

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

Table 19-145. PWM3_AQCTLAS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-12	RESERVED	R-0	0h	Reserved
11-10	CBD	R/W	0h	Action When TBCTR = PWM3_CMPB on Down Count Note: By definition, in count up-down mode when the counter equals 0 the direction is defined as 1 or counting up. 00: Do nothing (action disabled) 01: Clear: force PWM3A output low. 10: Set: force PWM3A output high. 11: Toggle PWM3A output: low output signal will be forced high, and a high signal will be forced low. Reset type: SYSRSn
9-8	CBU	R/W	0h	Action When TBCTR = PWM3_CMPB on Up Count Note: By definition, in count up-down mode when the counter equals 0 the direction is defined as 1 or counting up. 00: Do nothing (action disabled) 01: Clear: force PWM3A output low. 10: Set: force PWM3A output high. 11: Toggle PWM3A output: low output signal will be forced high, and a high signal will be forced low. Reset type: SYSRSn
7-6	CAD	R/W	0h	Action When TBCTR = PWM3_CMPA on Down Count Note: By definition, in count up-down mode when the counter equals 0 the direction is defined as 1 or counting up. 00: Do nothing (action disabled) 01: Clear: force PWM3A output low. 10: Set: force PWM3A output high. 11: Toggle PWM3A output: low output signal will be forced high, and a high signal will be forced low. Reset type: SYSRSn
5-4	CAU	R/W	0h	Action When TBCTR = PWM3_CMPA on Up Count Note: By definition, in count up-down mode when the counter equals 0 the direction is defined as 1 or counting up. 00: Do nothing (action disabled) 01: Clear: force PWM3A output low. 10: Set: force PWM3A output high. 11: Toggle PWM3A output: low output signal will be forced high, and a high signal will be forced low. Reset type: SYSRSn

Table 19-145. PWM3_AQCTLAS Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3-2	PRD	R/W	0h	Action When TBCTR = TBPRD 00: Do nothing (action disabled) 01: Clear: force PWM3A output low. 10: Set: force PWM3A output high. 11: Toggle PWM3A output: low output signal will be forced high, and a high signal will be forced low. Reset type: SYSRSn
1-0	ZRO	R/W	0h	Action When TBCTR = 0 00: Do nothing (action disabled) 01: Clear: force PWM3A output low. 10: Set: force PWM3A output high. 11: Toggle PWM3A output: low output signal will be forced high, and a high signal will be forced low. Reset type: SYSRSn

66 PWM3_AQCTLB Register (Offset = 528h) [Reset = 00000000h]

PWM3_AQCTLB is shown in Figure 19-105 and described in Table 19-81.

Return to the Summary Table.

PWM3 action qualifier B register

Figure 19-105. PWM3_AQCTLB Register

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

Table 19-147. PWM3_AQCTLB Register Field Descriptions

Bit	Field	Type	Reset	Description
31-12	RESERVED	R-0	0h	Reserved
11-10	CBD	R/W	0h	Action When TBCTR = PWM3_CMPB on Down Count Note: By definition, in count up-down mode when the counter equals 0 the direction is defined as 1 or counting up. 00: Do nothing (action disabled) 01: Clear: force PWM3B output low. 10: Set: force PWM3B output high. 11: Toggle PWM3B output: low output signal will be forced high, and a high signal will be forced low. Reset type: SYSRSn
9-8	CBU	R/W	0h	Action When TBCTR = PWM3_CMPB on Up Count Note: By definition, in count up-down mode when the counter equals 0 the direction is defined as 1 or counting up. 00: Do nothing (action disabled) 01: Clear: force PWM3B output low. 10: Set: force PWM3B output high. 11: Toggle PWM3B output: low output signal will be forced high, and a high signal will be forced low. Reset type: SYSRSn
7-6	CAD	R/W	0h	Action When TBCTR = PWM3_CMPA on Down Count Note: By definition, in count up-down mode when the counter equals 0 the direction is defined as 1 or counting up. 00: Do nothing (action disabled) 01: Clear: force PWM3B output low. 10: Set: force PWM3B output high. 11: Toggle PWM3B output: low output signal will be forced high, and a high signal will be forced low. Reset type: SYSRSn
5-4	CAU	R/W	0h	Action When TBCTR = PWM3_CMPA on Up Count Note: By definition, in count up-down mode when the counter equals 0 the direction is defined as 1 or counting up. 00: Do nothing (action disabled) 01: Clear: force PWM3B output low. 10: Set: force PWM3B output high. 11: Toggle PWM3B output: low output signal will be forced high, and a high signal will be forced low. Reset type: SYSRSn

Table 19-147. PWM3_AQCTLB Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3-2	PRD	R/W	0h	Action When TBCTR = TBPRD 00: Do nothing (action disabled) 01: Clear: force PWM3B output low. 10: Set: force PWM3B output high. 11: Toggle PWM3B output: low output signal will be forced high, and a high signal will be forced low. Reset type: SYSRSn
1-0	ZRO	R/W	0h	Action When TBCTR = 0 00: Do nothing (action disabled) 01: Clear: force PWM3B output low. 10: Set: force PWM3B output high. 11: Toggle PWM3B output: low output signal will be forced high, and a high signal will be forced low. Reset type: SYSRSn

67 PWM3_AQCTLBS Register (Offset = 52Ch) [Reset = 00000000h]

PWM3_AQCTLBS is shown in Figure 19-106 and described in Table 19-82.

Return to the Summary Table.

PWM3 action qualifier B shadow register

Figure 19-106. PWM3_AQCTLBS Register

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

Table 19-149. PWM3_AQCTLBS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-12	RESERVED	R-0	0h	Reserved
11-10	CBD	R/W	0h	Action When TBCTR = PWM3_CMPB on Down Count Note: By definition, in count up-down mode when the counter equals 0 the direction is defined as 1 or counting up. 00: Do nothing (action disabled) 01: Clear: force PWM3B output low. 10: Set: force PWM3B output high. 11: Toggle PWM3B output: low output signal will be forced high, and a high signal will be forced low. Reset type: SYSRSn
9-8	CBU	R/W	0h	Action When TBCTR = PWM3_CMPB on Up Count Note: By definition, in count up-down mode when the counter equals 0 the direction is defined as 1 or counting up. 00: Do nothing (action disabled) 01: Clear: force PWM3B output low. 10: Set: force PWM3B output high. 11: Toggle PWM3B output: low output signal will be forced high, and a high signal will be forced low. Reset type: SYSRSn
7-6	CAD	R/W	0h	Action When TBCTR = PWM3_CMPA on Down Count Note: By definition, in count up-down mode when the counter equals 0 the direction is defined as 1 or counting up. 00: Do nothing (action disabled) 01: Clear: force PWM3B output low. 10: Set: force PWM3B output high. 11: Toggle PWM3B output: low output signal will be forced high, and a high signal will be forced low. Reset type: SYSRSn
5-4	CAU	R/W	0h	Action When TBCTR = PWM3_CMPA on Up Count Note: By definition, in count up-down mode when the counter equals 0 the direction is defined as 1 or counting up. 00: Do nothing (action disabled) 01: Clear: force PWM3B output low. 10: Set: force PWM3B output high. 11: Toggle PWM3B output: low output signal will be forced high, and a high signal will be forced low. Reset type: SYSRSn

Table 19-149. PWM3_AQCTLBS Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3-2	PRD	R/W	0h	Action When TBCTR = TBPRD 00: Do nothing (action disabled) 01: Clear: force PWM3B output low. 10: Set: force PWM3B output high. 11: Toggle PWM3B output: low output signal will be forced high, and a high signal will be forced low. Reset type: SYSRSn
1-0	ZRO	R/W	0h	Action When TBCTR = 0 00: Do nothing (action disabled) 01: Clear: force PWM3B output low. 10: Set: force PWM3B output high. 11: Toggle PWM3B output: low output signal will be forced high, and a high signal will be forced low. Reset type: SYSRSn

68 PWM3_AQSFRC Register (Offset = 530h) [Reset = 00000000h]

PWM3_AQSFRC is shown in Figure 19-107 and described in Table 19-83.

Return to the Summary Table.

PWM3 action qualifier software force

Figure 19-107. PWM3_AQSFRC Register

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

Table 19-151. PWM3_AQSFRC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-7	RESERVED	R-0	0h	Reserved
6-4	PWMB	R/W	0h	Action qualifier software force on PWMB 000: Does nothing (softwrae force disabled) 001: Forces a continuous low on output B 010: Forces a continuous high on output B 011: Does nothing (softwrae force disabled) 100: Does nothing (softwrae force disabled) 101: Clear (low) when PWM3_AQOTSFRC[PWMB] = '1'. 110: Set (high) when PWM3_AQOTSFRC[PWMB] = '1'. 111: Toggle (Low -> High, High -> Low) when PWM3_AQOTSFRC[PWMB] = '1'. Reset type: SYSRSn
3	RESERVED	R-0	0h	Reserved
2-0	PWMA	R/W	0h	Action qualifier software force on PWMA 000: Does nothing (softwrae force disabled) 001: Forces a continuous low on output A 010: Forces a continuous high on output A 011: Does nothing (softwrae force disabled) 100: Does nothing (softwrae force disabled) 101: Clear (low) when PWM3_AQOTSFRC[PWMA] = '1'. 110: Set (high) when PWM3_AQOTSFRC[PWMA] = '1'. 111: Toggle (Low -> High, High -> Low) when PWM3_AQOTSFRC[PWMA] = '1'. Reset type: SYSRSn

69 PWM3_AQOTSFRC Register (Offset = 534h) [Reset = 00000000h]

PWM3_AQOTSFRC is shown in Figure 19-108 and described in Table 19-84.

Return to the Summary Table.

PWM3 action qualifier one time software force

Figure 19-108. PWM3_AQOTSFRC Register

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

Table 19-153. PWM3_AQOTSFRC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-5	RESERVED	R-0	0h	Reserved
4	PWMB	R-0/W1S	0h	Action qualifier one time software force on PWMB 0: Writing a 0 has no effect. Always reads back a 0. This bit is auto cleared once a write to this register is complete (i.e., a forced event is initiated.). This is a one-shot forced event. It can be overridden by another subsequent event on PWMB. 1: Initiates a single software forced event Reset type: SYSRSn
3-1	RESERVED	R-0	0h	Reserved
0	PWMA	R-0/W1S	0h	Action qualifier one time software force on PWMA 0: Writing a 0 has no effect. Always reads back a 0. This bit is auto cleared once a write to this register is complete (i.e., a forced event is initiated.). This is a one-shot forced event. It can be overridden by another subsequent event on PWMA. 1: Initiates a single software forced event Reset type: SYSRSn

Chapter 20

Enhanced Capture (eCAP)

This chapter describes the enhanced capture (eCAP) module, which is used in systems where accurate timing of external events is important.

The enhanced capture (eCAP) module is a eCAP.

ADVANCE INFORMATION

20.1 Introduction

20.1.1 Features

The features of the eCAP module include:

- Speed measurements of rotating machinery (for example, toothed sprockets sensed by way of Hall sensors)
- Elapsed time measurements between position sensor pulses
- Period and duty cycle measurements of pulse train signals
- Decoding current or voltage amplitude derived from duty cycle encoded current/voltage sensors

The eCAP module features described in this chapter include:

- 4-event time-stamp registers (each 32 bits)
- Edge polarity selection for up to four sequenced time-stamp capture events
- Interrupt on either of the four events
- Single-shot capture of up to four event time-stamps
- Continuous mode capture of time stamps in a four-deep circular buffer
- Absolute time-stamp capture
- Difference (Delta) mode time-stamp capture
- When not used in capture mode, the eCAP module can be configured as a single-channel PWM output

The capture functionality of the Type 1 eCAP is enhanced from the Type 0 eCAP with the following added features:

- Event filter reset bit:
  - Writing a 1 to ECCTL2[CTRFILTRESET] clears the event filter, the modulo counter, and any pending interrupts flags. Resetting the bit is useful for initialization and debug.
- Modulo counter status bits:
  - The modulo counter (ECCTL2 [MODCNTRSTS]) indicates which capture register is loaded next. In the Type 0 eCAP, to know the current state of the modulo counter was not possible
- Input multiplexer:
  - ECCTL0 [INPUTSEL] selects one of 128 input signals, which are detailed in Section 20.3 .
- EALLOW protection:
  - EALLOW protection was added to critical registers. To maintain software compatibility with Type-0, configure DEV_CFG_REGS.ECAPTYPE to make these registers unprotected.

The capture functionality of the Type 2 eCAP is enhanced from the Type 1 eCAP with the following added features:

- Added ECAPxSYNCINSEL register:
  - ECAPxSYNCINSEL register is added for each eCAP to select an external SYNCIN. Every eCAP can have a separate SYNCIN signal.

20.1.2 ECAP Related Collateral

Getting Started Materials

- *Leveraging High Resolution Capture (HRCAP) for Single Wire Data Transfer Application Report*

20.2 Description

The eCAP module represents one complete capture channel that can be instantiated multiple times, depending on the target device. In the context of this guide, one eCAP channel has the following independent key resources:

- Capture inputs can be connected using the Input X-BAR
- 128:1 input multiplexer
- Output X-BAR is used to configure output in APWM mode
- 32-bit time base (counter)

- 4 × 32-bit time-stamp capture registers (CAP1-CAP4)
- Four-stage sequencer (modulo4 counter) that is synchronized to external events, eCAP pin rising/falling edges.
- Modulo counter status register (MODCNTRSTS) to indicate sequencer state
- Independent edge polarity (rising/falling edge) selection for all four events
- Input capture signal prescaling (from 2-62 or bypass)
- One-shot compare register (two bits) to freeze captures after 1-4 time-stamp events
- Control for continuous time-stamp captures using a four-deep circular buffer (CAP1-CAP4) scheme
- Ability to reset event filter, modulo counter, and interrupt flags
- Interrupt capabilities on any of the four capture events
- EALLOW protection to control registers

20.3 Configuring Device Pins for the eCAP

The Input X-BAR connects the device pins to the module as input. Any GPIO on the device can be configured as an input. The GPIO input qualification can be set to synchronous or asynchronous mode. Using synchronized inputs can help with noise immunity but affects the eCAP accuracy by ±2 cycles.

A 128:1 input multiplexer must also be configured (see Figure 20-3). This multiplexer can select a variety of inputs by configuring ECCTL0.INPUTSEL.

Table 20-1. eCAP Input Selection

Input Signal	ECAP0 INDEX	ECAP1 INDEX
INPUTXBAR1	0	0
INPUTXBAR2	1	1
INPUTXBAR3	2	2
INPUTXBAR4	3	3
INPUTXBAR5	4	4
INPUTXBAR6	5	5
INPUTXBAR7	6	6
INPUTXBAR8	7	7
INPUTXBAR9	8	8
INPUTXBAR10	9	9
INPUTXBAR11	10	10
INPUTXBAR12	11	11
INPUTXBAR13	12	12
INPUTXBAR14	13	13
INPUTXBAR15	14	14
INPUTXBAR16	15	15
MCAN_INT0	16	16
OUTPUTXBAR1	17	17
OUTPUTXBAR2	18	18
OUTPUTXBAR3	19	19
OUTPUTXBAR4	20	20
OUTPUTXBAR5	21	21
OUTPUTXBAR6	22	22

Table 20-1. eCAP Input Selection (continued)

Input Signal	ECAP0 INDEX	ECAP1 INDEX
OUTPUTXBAR7	23	23
OUTPUTXBAR8	24	24
Reserved (Tie low)	25	25
Reserved (Tie low)	26	26
Reserved (Tie low)	27	27
Reserved (Tie low)	28	28
ADC0EVT1	29	29
ADC0EVT2	30	30
ADC0EVT3	31	31
ADC0EVT4	32	32
ADC1EVT1	33	33
ADC1EVT2	34	34
ADC1EVT3	35	35
ADC1EVT4	36	36
ADC2EVT1	37	37
ADC2EVT2	38	38
ADC2EVT3	39	39
ADC2EVT4	40	40
CMPSS0_CTRIPL	41	41
CMPSS1_CTRIPL	42	42
CMPSS2_CTRIPL	43	43
CMPSS3_CTRIPL	44	44
CMPSS0_CTRIPH	45	45
CMPSS1_CTRIPH	46	46
CMPSS2_CTRIPH	47	47
CMPSS3_CTRIPH	48	48
CMPSS0_CTRIPH_OR_CTRIPL	49	49
CMPSS1_CTRIPH_OR_CTRIPL	50	50
CMPSS2_CTRIPH_OR_CTRIPL	51	51
CMPSS3_CTRIPH_OR_CTRIPL	52	52
INPUTXBAR7	53	Reserved
INPUTXBAR8	Reserved	53
Reserved (Tie low)	54-127	54-127

Note

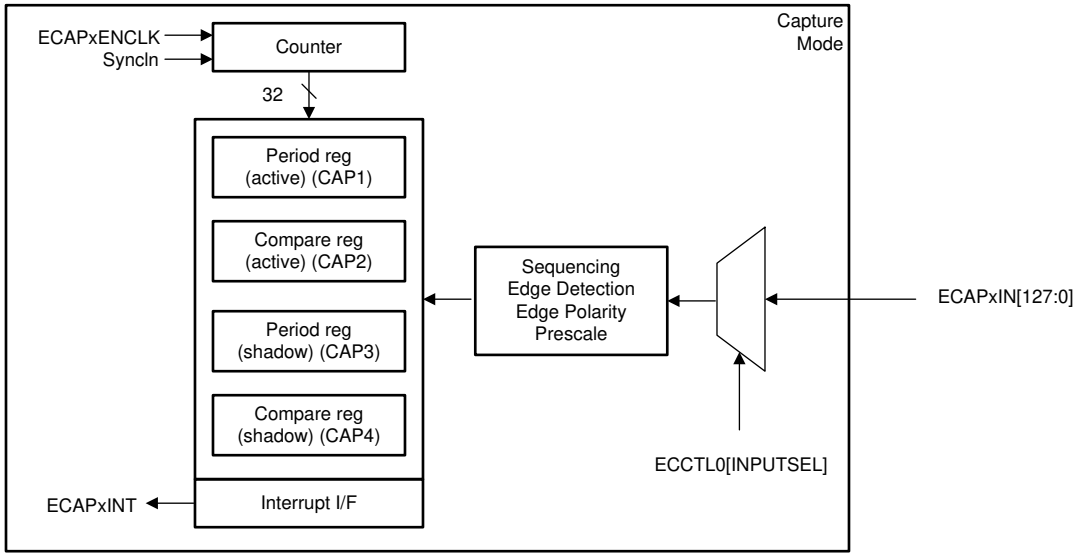
ECAPxIN has to be at least 2 * MCLK-cycles wide to be properly captured by the eCAP module; otherwise, the input pulse can get missed from sampling by the MCLK .

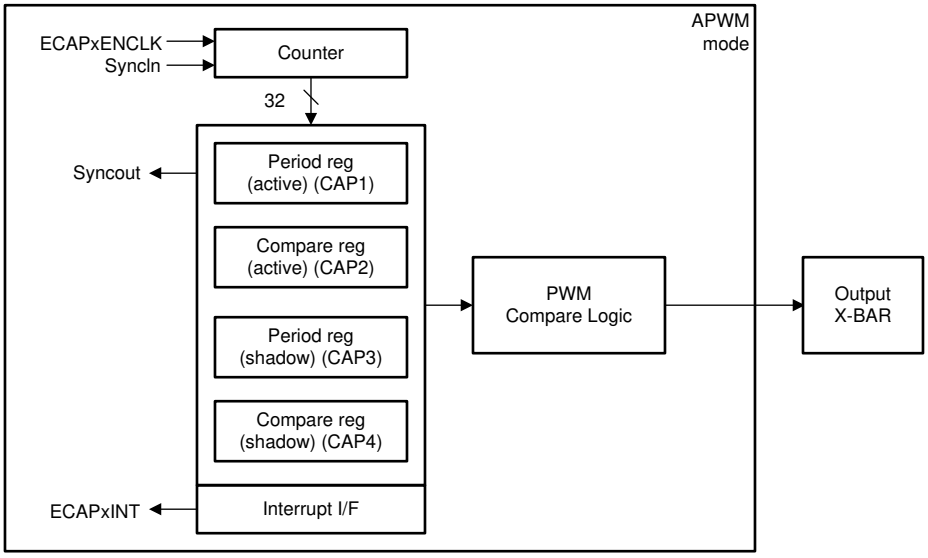
ADVANCE INFORMATION

20.4 Capture and APWM Operating Mode

Use the eCAP module resources to implement a single-channel PWM generator (with 32-bit capabilities) when the eCAP module is not being used for input captures. The counter operates in count-up mode, providing a time-base for asymmetrical pulse width modulation (PWM) waveforms. The CAP1 and CAP2 registers become the active period and compare registers, respectively, while CAP3 and CAP4 registers become the period and compare shadow registers, respectively. Figure 20-1 is a high-level view of both the capture and auxiliary pulse-width modulator (APWM) modes of operation.

Figure 20-2 further descries the output of the eCAP in APWM mode based on the CMP and PRD values.





Note

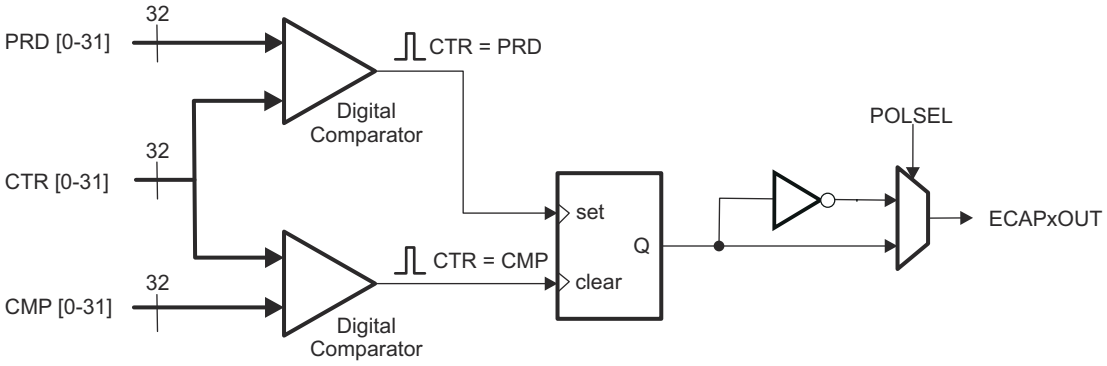
A single pin is shared between CAP and APWM functions. In capture mode, the pin is an input; in APWM mode, the pin is an output.

ADVANCE INFORMATION

Note

In APWM mode, writing any value to CAP1/CAP2 active registers also writes the same value to the corresponding shadow registers CAP3/CAP4. This emulates immediate mode. Writing to the shadow registers CAP3/CAP4 invokes the shadow mode.

Figure 20-1. Capture and APWM Modes of Operation



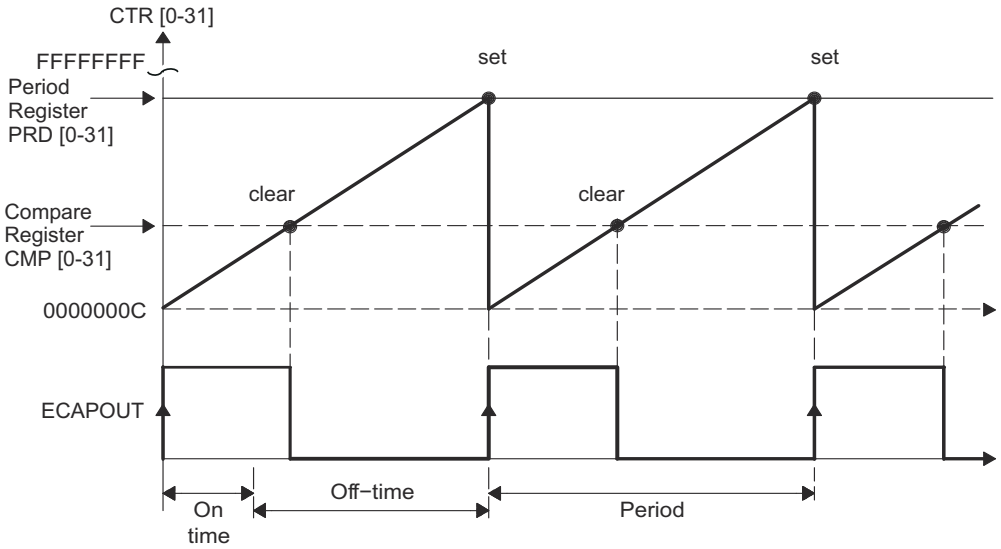


Figure 20-2. Counter Compare and PRD Effects on the eCAP Output in APWM Mode

ADVANCE INFORMATION

20.5 Capture Mode Description

Figure 20-3 shows the various components that implement the capture function.

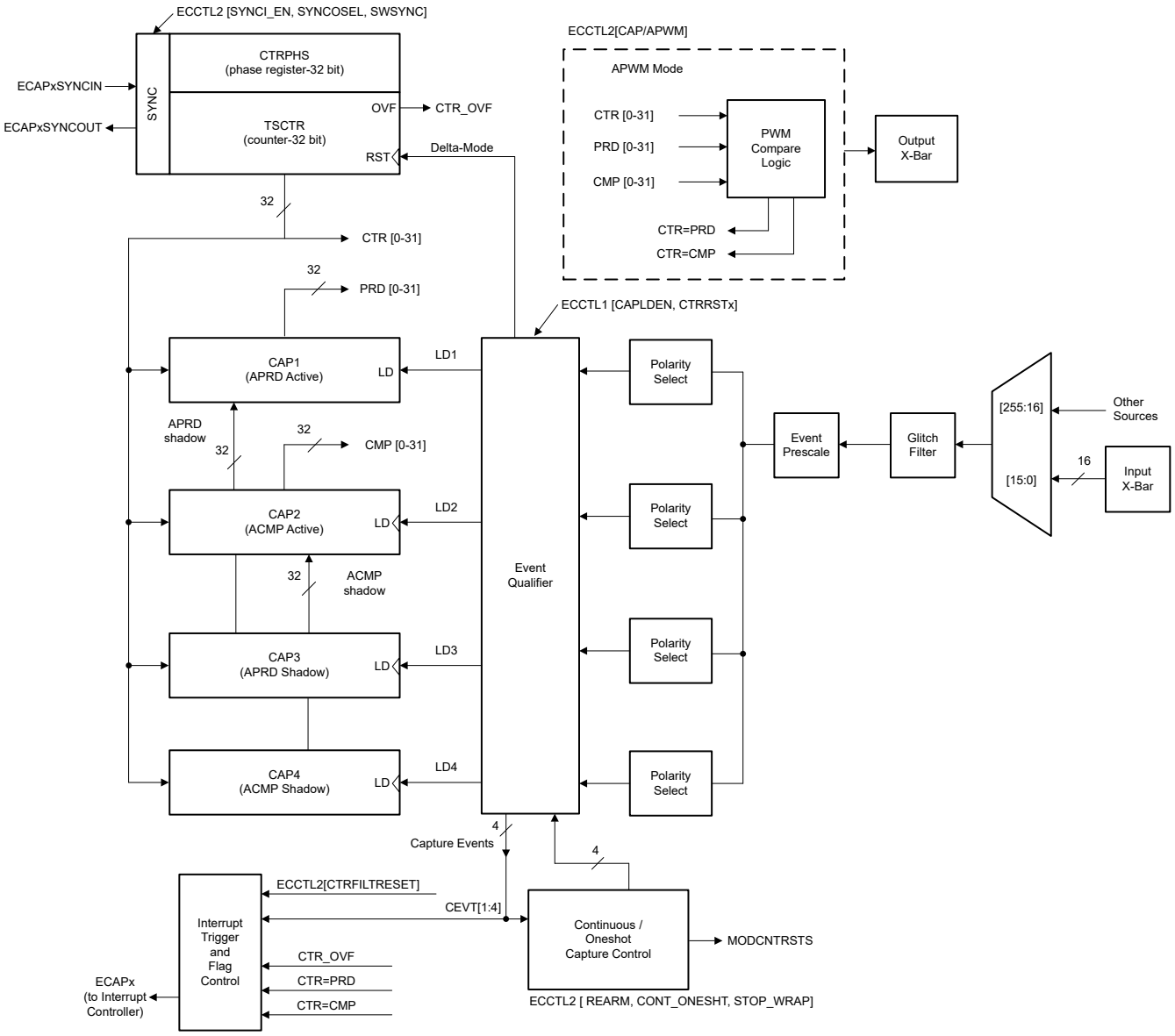
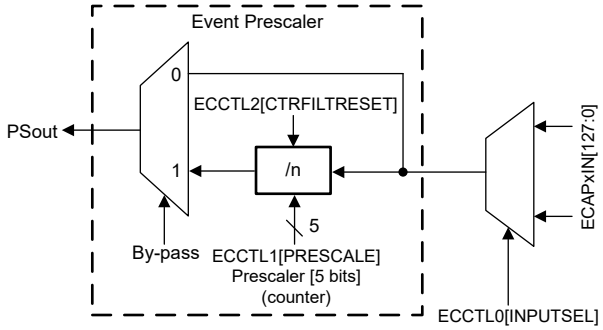


Figure 20-3. eCAP Block Diagram

20.5.1 Event Prescaler

An input capture signal (pulse train) can be prescaled by N = 2-62 (in multiples of 2) or can bypass the prescaler. This is useful when very high frequency signals are used as inputs. Figure 20-4 shows a functional diagram and Figure 20-5 shows the operation of the prescale function. The event prescaler can be reset by setting the ECCTL2.CTRFILTRESET register bit.



Note

When a prescale value of 1 is chosen (ECCTL1[13:9] = 0,0,0,0,0), the input capture signal bypasses the prescale logic completely.

Note

The first Rise edge after Prescale configuration change is not passed to Capture logic, prescaler value takes into effect on the second rising edge after the configuration.

Figure 20-4. Event Prescale Control

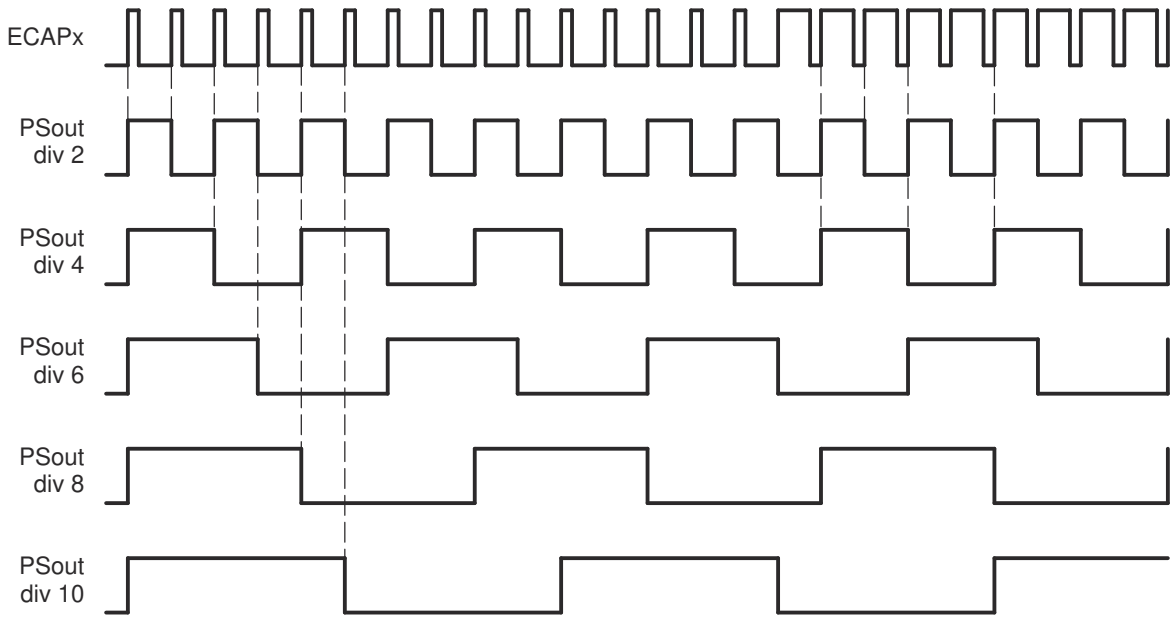


Figure 20-5. Prescale Function Waveforms

ADVANCE INFORMATION

20.5.2 Edge Polarity Select and Qualifier

Functionality and features include:

- Four independent edge polarity (rising edge/falling edge) selection muxes are used, one for each capture event.
- Each edge (up to 4) is event qualified by the Modulo4 sequencer.
- The edge event is gated to the respective CAPx register by the Mod4 counter. The CAPx register is loaded on the falling edge.

20.5.3 Continuous/One-Shot Control

Operation of eCAP in Continuous/One-Shot mode:

- The Mod4 (2-bit) counter is incremented using edge qualified events (CEVT1-CEVT4).
- The Mod4 counter continues counting (0->1->2->3->0) and wraps around unless stopped.
- During one-shot operation, a 2-bit stop register (STOP_WRAP) is used to compare the Mod4 counter output, and when equal, stops the Mod4 counter and inhibits further loads of the CAP1-CAP4 registers. In this mode, if TSCCTR counter is configured to reset on capture event (CEVTx) by configuring ECCTL1.CTRRSTx bit, the operation still keeps resetting the TSCCTR counter on capture event (CEVTx) after the STOP_WRAP value is reached and re-arm (REARM) has not occurred.

The continuous/one-shot block controls the start, stop and reset (zero) functions of the Mod4 counter, using a mono-shot type of action that can be triggered by the stop-value comparator and re-armed using software control.

Once armed, the eCAP module waits for 1-4 (defined by stop-value) capture events before freezing both the Mod4 counter and contents of CAP1-4 registers (time stamps).

Re-arming prepares the eCAP module for another capture sequence. Also, re-arming clears (to zero) the Mod4 counter and permits loading of CAP1-4 registers again, providing the CAPLDEN bit is set.

In continuous mode, the Mod4 counter continues to run (0->1->2->3->0, the one-shot action is ignored, and capture values continue to be written to CAP1-4 in a circular buffer sequence.

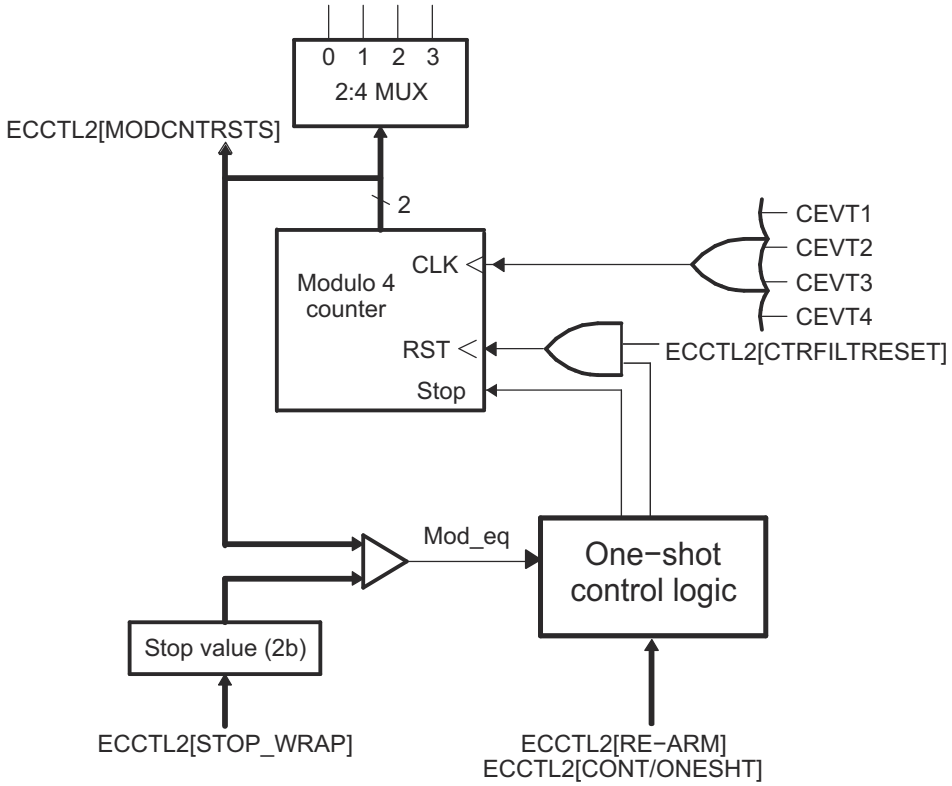


Figure 20-6. Details of the Continuous/One-shot Block

ADVANCE INFORMATION

20.5.4 32-Bit Counter and Phase Control

This counter provides the time-base for event captures, and is clocked using the system clock.

A phase register is provided to achieve synchronization with other counters using a hardware and software forced sync. This is useful in APWM mode when a phase offset between modules is needed.

On any of the four event loads, an option to reset the 32-bit counter is given. This is useful for time difference capture. The 32-bit counter value is captured first, then the counter value is reset to 0 by any of the LD1-LD4 signals.

20.5.5 CAP1-CAP4 Registers

These 32-bit registers are supplied by the 32-bit counter timer bus, CTR[0-31], and are loaded (capture a time-stamp) when the respective LD inputs are strobed.

Control bit CAPLDEN can inhibit loading of the capture registers. During one-shot operation, this bit is cleared (loading is inhibited) automatically when a stop condition occurs, StopValue = Mod4.

CAP1 and CAP2 registers become the active period and compare registers, respectively, in APWM mode.

CAP3 and CAP4 registers become the respective shadow registers (APRD and ACMP) for CAP1 and CAP2 during APWM operation.

20.5.6 eCAP Synchronization

eCAP modules can be synchronized with each other by selecting a common SYNCIN source. SYNCIN source for eCAP can be either software sync-in or external sync-in. The external sync-in signal can come from MCPWM. The SWSYNC of the eCAP module is logical ORed with the SYNC signal as shown in Figure 20-7. The SYNC signal is defined by the selection of ECAPxSYNCINSEL[SEL] as shown in Figure 20-8.

Note

ECAPxSYNCOUT going to the ECAPSYNCIN multiplexer is disabled. For example, ECAP1SYNCOUT cannot be a sync in to ECAP1, but ECAP1SYNCOUT can be a sync in to ECAP2, ECAP3, and so on.

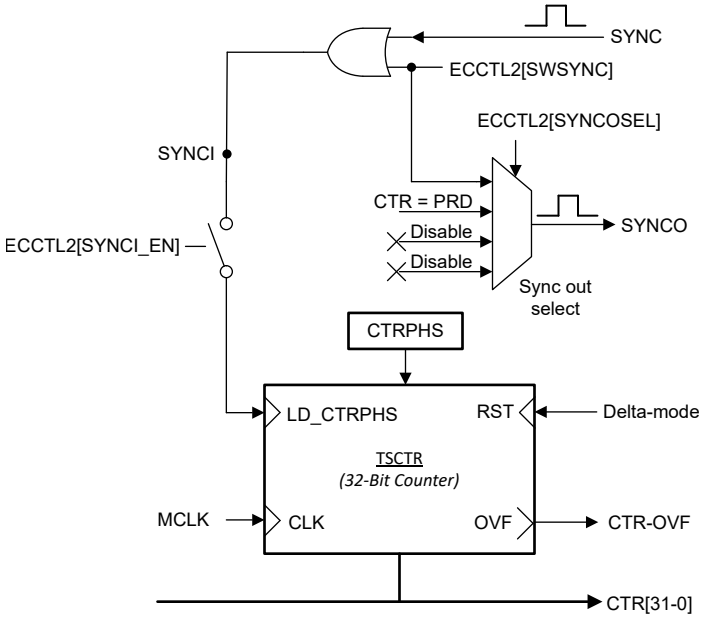


Figure 20-7. ECAP Counter and Synchronization Block

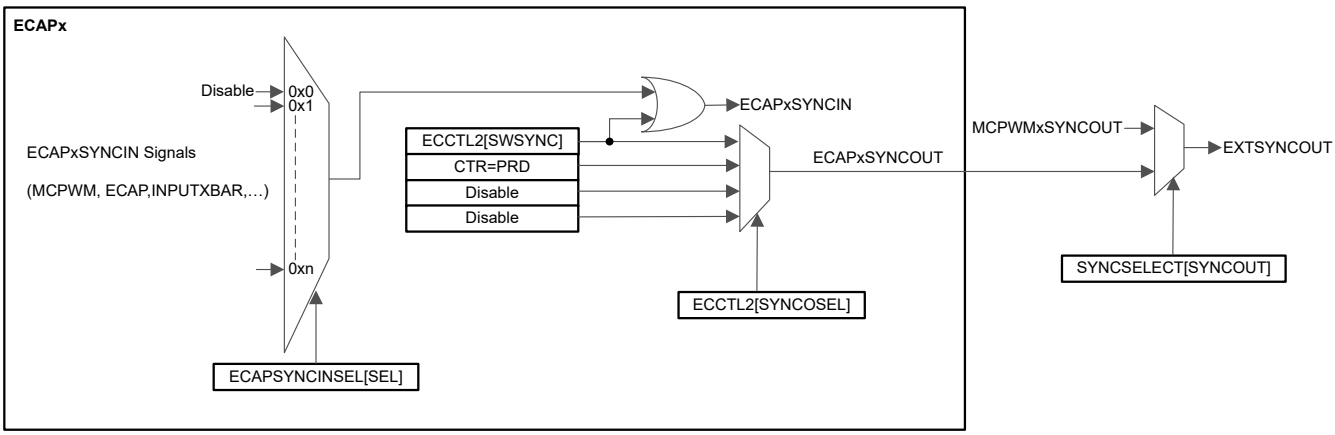


Figure 20-8. eCAP Synchronization Scheme

20.5.6.1 Example 1 - Using SWSYNC with ECAP Module

Implement the following steps to use SWSYNC with ECAP1 and ECAP2.

- Configure ECAP[1..2].ECAPSYNCINSEL.SEL = 0x0 to disable external SYNCIN coming to eCAP1.
- Configure ECAP[1..2].ECCTL2.SWSYNC = 0x1, to force Software Synchronization of the TSCTR counter.

To use SWSYNC with other eCAP modules, make sure that the previous eCAP chain is not generating a SYNCOUT signal that interferes with the software synchronization.

20.5.7 Interrupt Control

Operation and features of the eCAP interrupt control include (see Figure 20-9):

- An interrupt can be generated on capture events (CEVT1-CEVT4, CTROVF) or APWM events (CTR = PRD, CTR = CMP).

ADVANCE INFORMATION

- A counter overflow event (FFFFFFFF->00000000) is also provided as an interrupt source (CTROVF).
- The capture events are edge and sequencer-qualified (ordered in time) by the polarity select and Mod4 gating, respectively.
- One of these events can be selected as the interrupt source (from the eCAPx module) going to the Interrupt Controller.
- Seven interrupt events (CEVT1, CEVT2, CEVT3, CEVT4, CNTOVF, CTR=PRD, CTR=CMP) can be generated.
- The interrupt enable register (ECEINT) is used to enable/disable individual interrupt event sources. The interrupt flag register (ECFLG) indicates if any interrupt event has been latched and contains the global interrupt flag bit (INT). An interrupt pulse is generated to the interrupt controller only if any of the interrupt events are enabled, the flag bit is 1, and the INT flag bit is 0. The interrupt service routine must clear the global interrupt flag bit and the serviced event using the interrupt clear register (ECCLR) before any other interrupt pulses are generated. All interrupt flags are cleared upon an event filter reset by writing a 1 to ECCTL2[CLRFILTRESET]. To force an interrupt event, use the interrupt force register (ECFRC). Forced interrupts can be used for test purposes.

Note

The CEVT1, CEVT2, CEVT3, CEVT4 flags are only active in capture mode (ECCTL2[CAP/APWM == 0]). The CTR=PRD, CTR=CMP flags are only valid in APWM mode (ECCTL2[CAP/APWM == 1]). CNTOVF flag is valid in both modes.

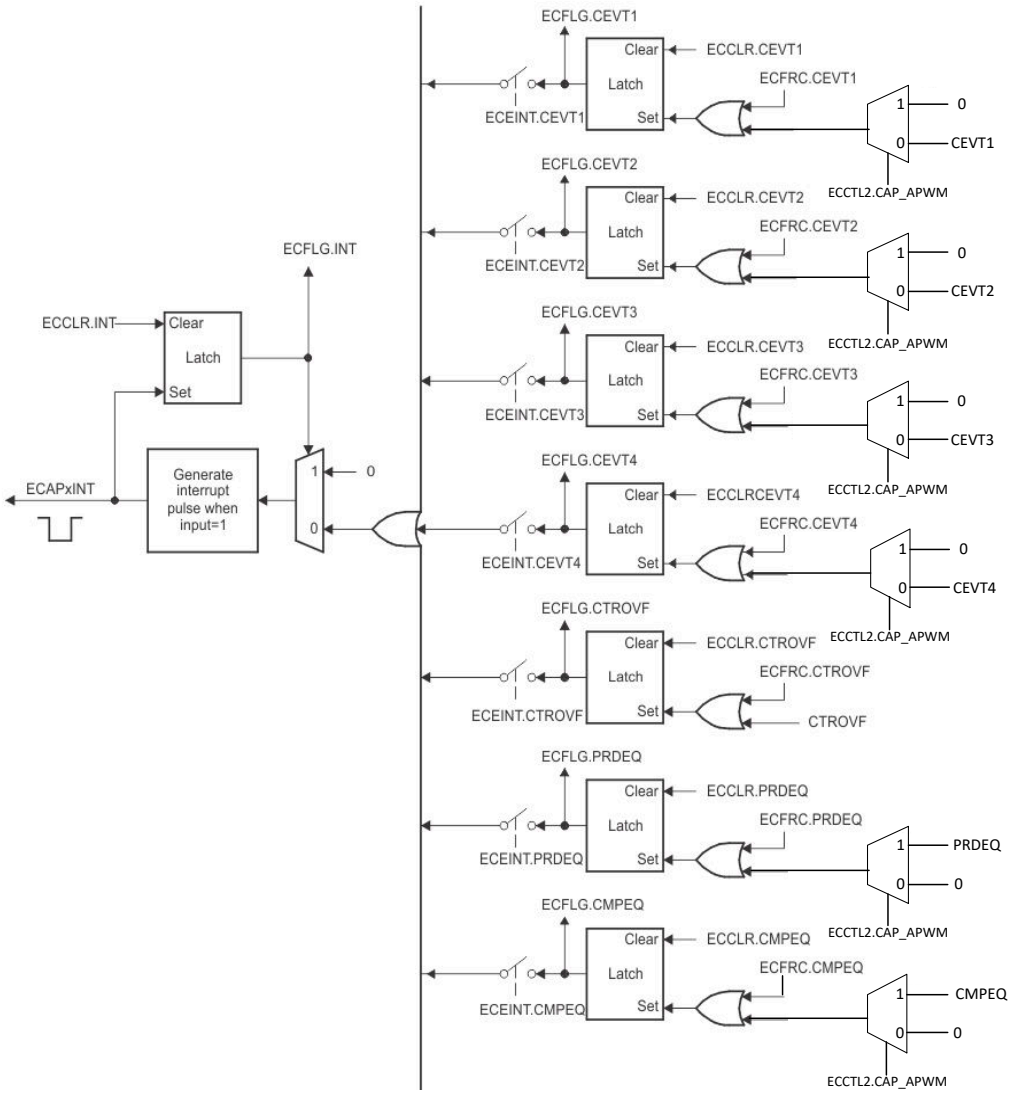


Figure 20-9. Interrupts in eCAP Module

ADVANCE INFORMATION

20.5.8 Shadow Load and Lockout Control

In capture mode, this logic inhibits (locks out) any shadow loading of CAP1 or CAP2 from APRD and ACMP registers, respectively.

In APWM mode, shadow loading is active and two choices are permitted:

- Immediate - APRD or ACMP are transferred to CAP1 or CAP2 immediately upon writing a new value.
- On period equal, CTR[31:0] = PRD[31:0].

20.5.9 APWM Mode Operation

Main operating highlights of the APWM section:

- The time-stamp counter bus is made available for comparison by way of 2 digital (32-bit) comparators.
- When CAP1/2 registers are not used in capture mode, the contents can be used as Period and Compare values in APWM mode.
- Double buffering is achieved using shadow registers APRD and ACMP (CAP3/4). The shadow register contents are transferred over to CAP1/2 registers, either immediately upon a write, or on a CTR = PRD trigger.
- In APWM mode, writing to CAP1/CAP2 active registers also writes the same value to the corresponding shadow registers CAP3/CAP4. This emulates immediate mode. Writing to the shadow registers CAP3/CAP4 invokes the shadow mode.
- During initialization, write to the active registers for both period and compare. This automatically copies the initial values into the shadow values. For subsequent compare updates during run-time, use the shadow registers.

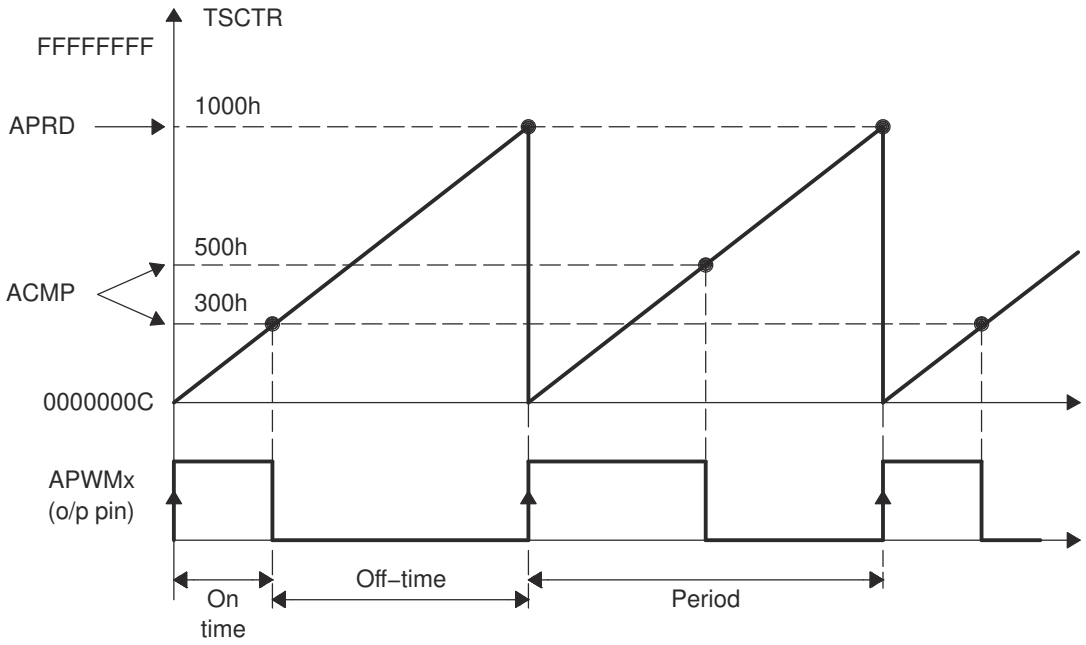


Figure 20-10. PWM Waveform Details Of APWM Mode Operation

The behavior of APWM active high mode (APWMPOL == 0) is as follows:

```
CMP = 0x00000000, output low for duration of period (0% duty)
```

```
CMP = 0x00000001, output high 1 cycle
```

```
CMP = 0x00000002, output high 2 cycles
```

```
CMP = PERIOD, output high except for 1 cycle (<100% duty)
```

```
CMP = PERIOD+1, output high for complete period (100% duty)
```

```
CMP > PERIOD+1, output high for complete period
```

The behavior of APWM active low mode (APWMPOL == 1) is as follows:

```
CMP = 0x00000000, output high for duration of period (0% duty)
```

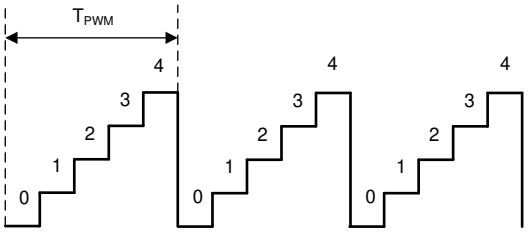
```
CMP = 0x00000001, output low 1 cycle
```

```
CMP = 0x00000002, output low 2 cycles
```

```
CMP = PERIOD, output low except for 1 cycle (<100% duty)
```

```
CMP = PERIOD+1, output low for complete period (100% duty)
```

```
CMP > PERIOD+1, output low for complete period
```



$$T_{PWM} = (CAP1 + 1) \times T_{TSCTR}$$

$$F_{PWM} = \frac{1}{T_{PWM}}$$

Figure 20-11. Time-Base Frequency and Period Calculation

ADVANCE INFORMATION

20.6 Application of the eCAP Module

The following sections provide applications examples to show how to operate the eCAP module.

20.6.1 Example 1 - Absolute Time-Stamp Operation Rising-Edge Trigger

Figure 20-12 shows an example of continuous capture operation (Mod4 counter wraps around). In this figure, TSCTR counts-up without resetting and capture events are qualified on the rising edge only, this gives period (and frequency) information.

On an event, the TSCTR contents (time-stamp) is first captured, then Mod4 counter is incremented to the next state. When the TSCTR reaches FFFFFFFF (maximum value), the Mod4 counter wraps around to 00000000 (not shown in Figure 20-12), if this occurs, the CTROVF (counter overflow) flag is set, and an interrupt (if enabled) occurs. Captured Time-stamps are valid at the point indicated by the diagram (after the fourth event); hence, event CEVT4 can conveniently be used to trigger an interrupt and the CPU can read data from the CAPx registers.

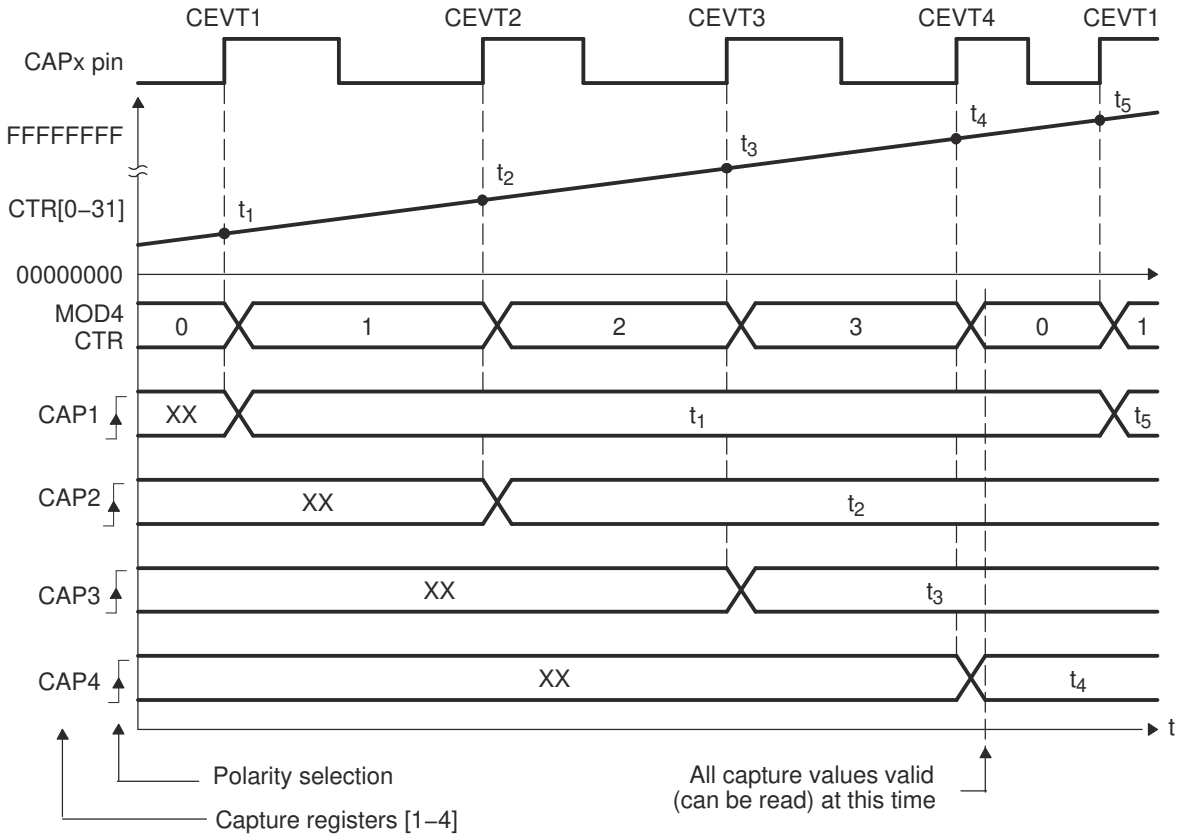


Figure 20-12. Capture Sequence for Absolute Time-stamp and Rising-Edge Detect

20.6.2 Example 2 - Absolute Time-Stamp Operation Rising- and Falling-Edge Trigger

In Figure 20-13, the eCAP operating mode is almost the same as in the previous section except capture events are qualified as either rising or falling edge, this now gives both period and duty cycle information, that is: Period1 = $t_3 - t_1$, Period2 = $t_5 - t_3$, …and so on. Duty Cycle1 (on-time %) = $(t_2 - t_1)$ / Period1 x 100%, and so on. Duty Cycle1 (off-time %) = $(t_3 - t_2)$ / Period1 x 100%, and so on.

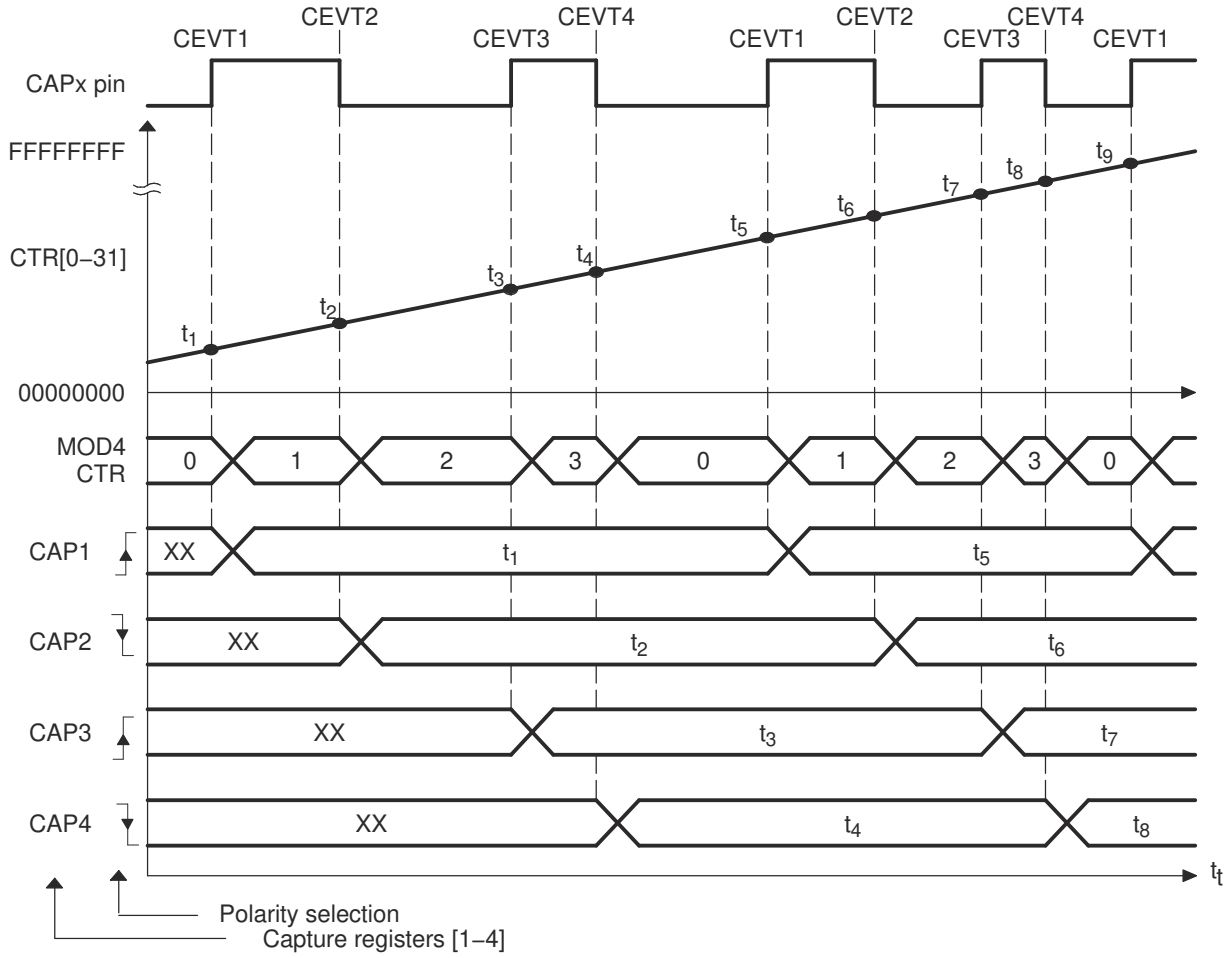


Figure 20-13. Capture Sequence for Absolute Time-stamp with Rising- and Falling-Edge Detect

ADVANCE INFORMATION

20.6.3 Example 3 - Time Difference (Delta) Operation Rising-Edge Trigger

Figure 20-14 shows how the eCAP module can be used to collect delta timing data from pulse train waveforms. Here Continuous Capture mode (TSCTR counts-up without resetting, and Mod4 counter wraps around) is used. In Delta-time mode, TSCTR is reset back to zero on every valid event. Here capture events are qualified as rising edge only. On an event, TSCTR contents (Time-Stamp) is captured first, and then TSCTR is reset to zero. The Mod4 counter then increments to the next state. If TSCTR reaches FFFFFFFF (maximum value), before the next event, the Mod4 counter wraps around to 00000000 and continues, a CNTOVF (counter overflow) flag is set, and an interrupt (if enabled) occurs. The advantage of Delta-time mode is that the CAPx contents directly give timing data without the need for CPU calculations, that is, Period1 = T_1, Period2 = T_2, and so on. As shown in Figure 20-14, the CEVT1 event is a good trigger point to read the timing data, T_1, T_2, T_3, T_4 are all valid here.

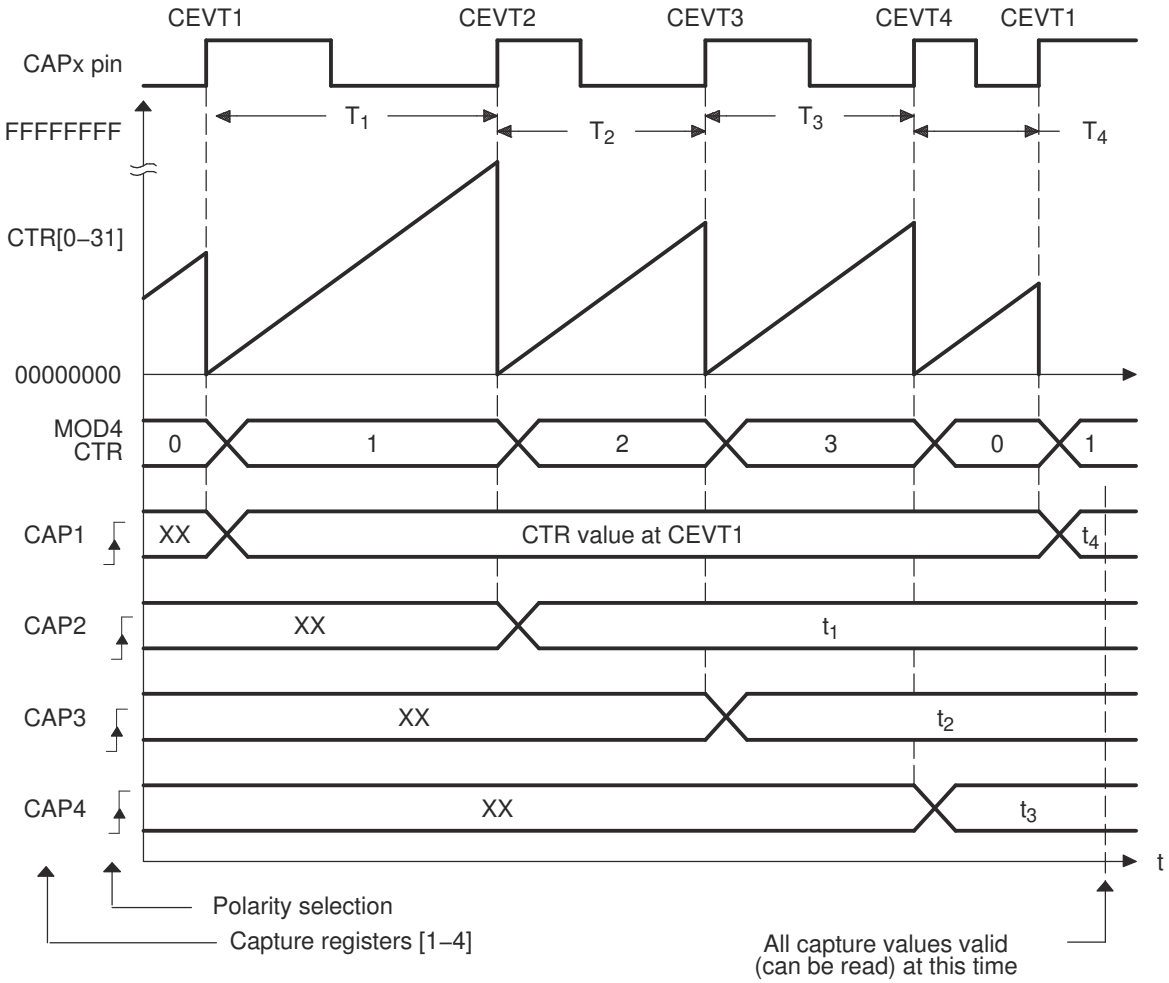


Figure 20-14. Capture Sequence for Delta Mode Time-stamp and Rising Edge Detect

ADVANCE INFORMATION

20.6.4 Example 4 - Time Difference (Delta) Operation Rising- and Falling-Edge Trigger

In Figure 20-15, the eCAP operating mode is almost the same as in previous section except capture events are qualified as either rising or falling edge, this now gives both period and duty cycle information, that is: Period1 = T_1+T_2, Period2 = T_3+T_4, and so on. Duty Cycle1 (on-time %) = T_1 / Period1 x 100%, Duty Cycle1 (off-time %) = T_2 / Period1 x 100%, and so on.

During initialization, write to the active registers for both period and compare. This action automatically copies the init values into the shadow values. For subsequent compare updates during run-time, the shadow registers must be used.

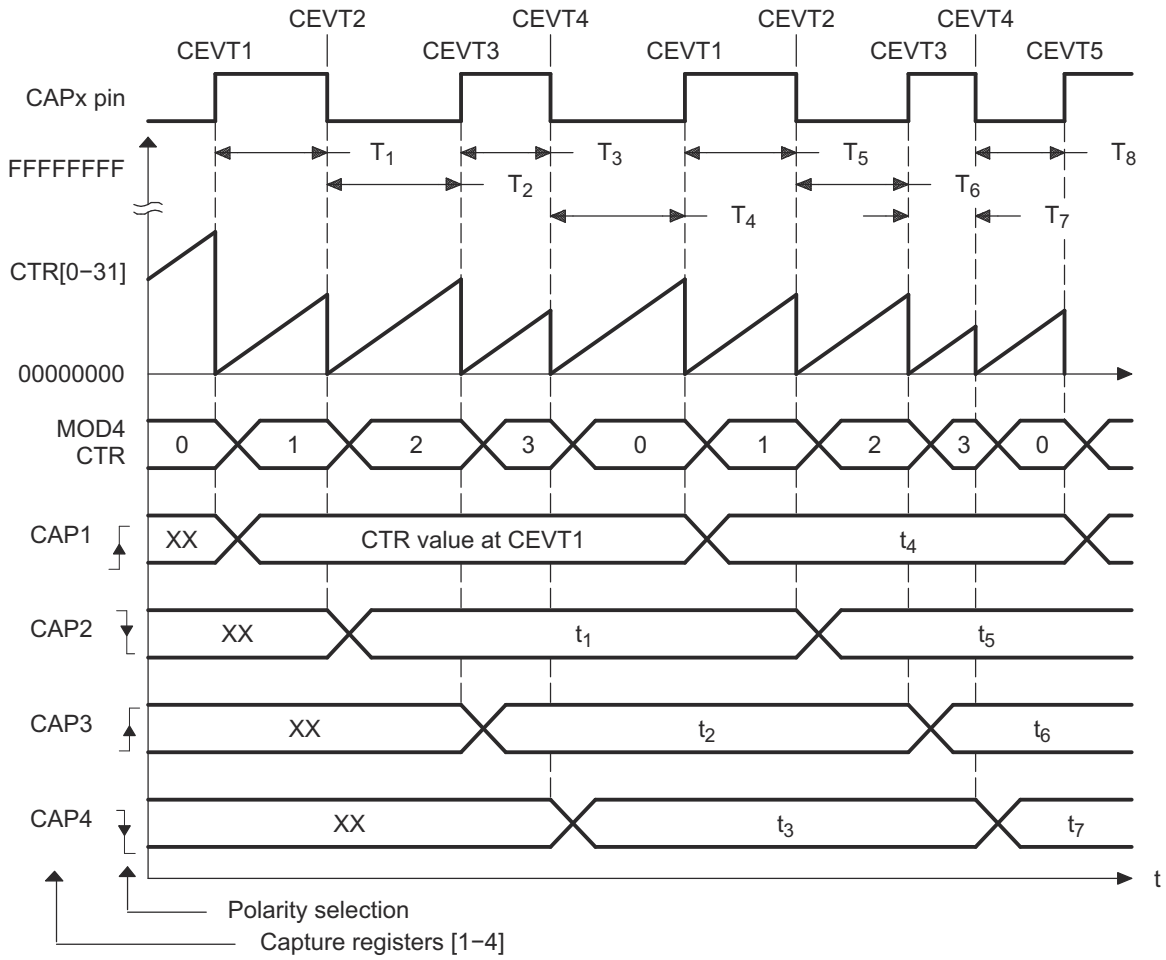


Figure 20-15. Capture Sequence for Delta Mode Time-stamp with Rising- and Falling-Edge Detect

ADVANCE INFORMATION

20.7 Application of the APWM Mode

In this example, the eCAP module is configured to operate as a PWM generator. Here, a very simple single-channel PWM waveform is generated from the APWMx output pin. The PWM polarity is active high, which means that the compare value (CAP2 reg is now a compare register) represents the on-time (high level) of the period. Alternatively, if the APWMPOL bit is configured for active low, then the compare value represents the off-time.

20.7.1 Example 1 - Simple PWM Generation (Independent Channels)

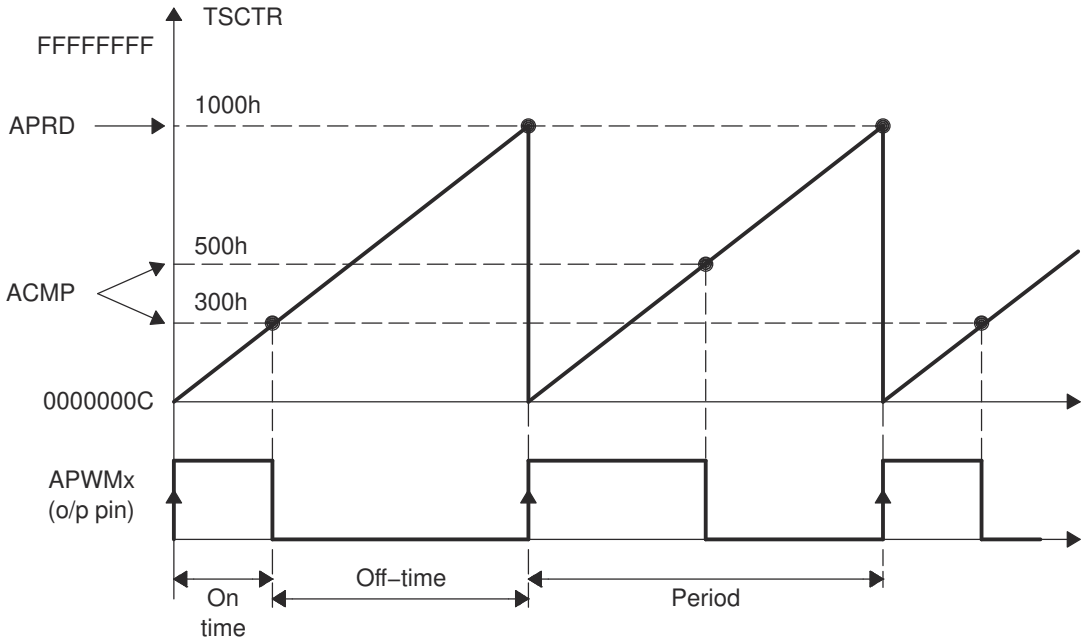


Figure 20-16. PWM Waveform Details of APWM Mode Operation

20.8 ECAP Registers

This Section describes the ECAP Registers.

20.8.1 ECAP Base Address Table

Table 20-2. ECAP Base Address Table

Bit Field Name		DriverLib Name	Base Address
Instance	Structure		
ECap0Regs	ECAP_REGS	ECAP0	0x4044_0000
ECap1Regs	ECAP_REGS	ECAP1	0x4044_1000

ADVANCE INFORMATION

20.8.2 ECAP_REGS Registers

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

Table 20-3. ECAP_REGS Registers

Offset	Acronym	Register Name	Write Protection	Section
0h	TSCTR	Time-Stamp Counter		Go
4h	CTRPHS	Counter Phase Offset Value Register		Go
8h	CAP1	Capture 1 Register		Go
Ch	CAP2	Capture 2 Register		Go
10h	CAP3	Capture 3 Register		Go
14h	CAP4	Capture 4 Register		Go
24h	ECCTL0	Capture Control Register 0	EALLOW	Go
28h	ECCTL1	Capture Control Register 1	EALLOW	Go
2Ah	ECCTL2	Capture Control Register 2	EALLOW	Go
2Ch	ECEINT	Capture Interrupt Enable Register	EALLOW	Go
2Eh	ECFLG	Capture Interrupt Flag Register		Go
30h	ECCLR	Capture Interrupt Clear Register		Go
32h	ECFRC	Capture Interrupt Force Register	EALLOW	Go
3Ch	ECAPSYNCINSEL	SYNC source select register	EALLOW	Go

Complex bit access types are encoded to fit into small table cells. Table 20-4 shows the codes that are used for access types in this section.

Table 20-4. ECAP_REGS Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
R-0	R -0	Read Returns 0s
Write Type		
W	W	Write
W1C	W 1C	Write 1 to clear
W1S	W 1S	Write 1 to set
Reset or Default Value		
-n		Value after reset or the default value

ADVANCE INFORMATION

1 TSCTR Register (Offset = 0h) [Reset = 00000000h]

TSCTR is shown in Figure 20-17 and described in Table 20-5.

Return to the Summary Table.

Time-Stamp Counter

Figure 20-17. TSCTR Register

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

Table 20-6. TSCTR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TSCTR	R/W	0h	Active 32-bit counter register that is used as the capture time-base HR mode : 1) This register reads HRCOUNTER value and is not writable 2) can be reset using CTRFILTRESET 3) Its not synchronized to SYSCLK domain so reads may not be accurate Reset type: SYSRSn

2 CTRPHS Register (Offset = 4h) [Reset = 00000000h]

CTRPHS is shown in Figure 20-18 and described in Table 20-6.

Return to the Summary Table.

Counter Phase Offset Value Register

Figure 20-18. CTRPHS Register

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

Table 20-8. CTRPHS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CTRPHS	R/W	0h	Counter phase value register that can be programmed for phase lag/lead. This register CTRPHS is loaded into TSCTR upon either a SYNCI event or S/W force via a control bit. Used to achieve phase control synchronization with respect to other eCAP and PWM time-bases. This register is not applicable in HR mode. Reset type: SYSRSn

3 CAP1 Register (Offset = 8h) [Reset = 00000000h]

CAP1 is shown in Figure 20-19 and described in Table 20-7.

Return to the Summary Table.

Capture 1 Register

Figure 20-19. CAP1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CAP1																															
R/W-0h																															

Table 20-10. CAP1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CAP1	R/W	0h	This register can be loaded (written) by: - Time-Stamp counter value (TSCTR) during a capture event - Software - may be useful for test purposes or initialization - ARPD shadow register (CAP3) when used in APWM mode Reset type: SYSRSn

4 CAP2 Register (Offset = Ch) [Reset = 00000000h]

CAP2 is shown in Figure 20-20 and described in Table 20-8.

Return to the Summary Table.

Capture 2 Register

Figure 20-20. CAP2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CAP2																															
R/W-0h																															

Table 20-12. CAP2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CAP2	R/W	0h	This register can be loaded (written) by: - Time-Stamp (counter value) during a capture event - Software - may be useful for test purposes - ACMP shadow register (CAP4) when used in APWM mode Reset type: SYSRSn

5 CAP3 Register (Offset = 10h) [Reset = 00000000h]

CAP3 is shown in Figure 20-21 and described in Table 20-9.

Return to the Summary Table.

Capture 3 Register

Figure 20-21. CAP3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CAP3																															
R/W-0h																															

Table 20-14. CAP3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CAP3	R/W	0h	In CMP mode, this is a time-stamp capture register. In APWM mode, this is the period shadow (APRD) register. You can update the PWM period value through this register. CAP3 (APRD) shadows CAP1 in this mode. Reset type: SYSRSn

6 CAP4 Register (Offset = 14h) [Reset = 00000000h]

CAP4 is shown in Figure 20-22 and described in Table 20-10.

Return to the Summary Table.

Capture 4 Register

Figure 20-22. CAP4 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CAP4																															
R/W-0h																															

Table 20-16. CAP4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CAP4	R/W	0h	In CMP mode, this is a time-stamp capture register. In APWM mode, this is the compare shadow (ACMP) register. You can update the PWM compare value via this register. CAP4 (ACMP) shadows CAP2 in this mode. Reset type: SYSRSn

7 ECCTL0 Register (Offset = 24h) [Reset = 0000007Fh]

ECCTL0 is shown in Figure 20-23 and described in Table 20-11.

Return to the Summary Table.

Capture Control Register 0

Figure 20-23. ECCTL0 Register

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

Table 20-18. ECCTL0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-7	RESERVED	R-0	0h	Reserved
6-0	INPUTSEL	R/W	7Fh	Capture input source select bits 0000000 capture input is ECAPxINPUT[0] 0000001 capture input is ECAPxINPUT[1] 0000010 capture input is ECAPxINPUT[2] ... 1111111 capture input is ECAPxINPUT[127] Reset type: CPU1.SYSRSn

ADVANCE INFORMATION

8 ECCTL1 Register (Offset = 28h) [Reset = 0000h]

ECCTL1 is shown in Figure 20-24 and described in Table 20-12.

Return to the Summary Table.

Capture Control Register 1

Figure 20-24. ECCTL1 Register

15	14	13	12	11	10	9	8
FREE_SOFT		PRESCALE					CAPLDEN
R/W-0h		R/W-0h					R/W-0h
7	6	5	4	3	2	1	0
CTRRST4	CAP4POL	CTRRST3	CAP3POL	CTRRST2	CAP2POL	CTRRST1	CAP1POL
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 20-20. ECCTL1 Register Field Descriptions

Bit	Field	Type	Reset	Description
15-14	FREE_SOFT	R/W	0h	Emulation Control Reset type: SYSRSn 0h (R/W) = TSCTR counter stops immediately on emulation suspend 1h (R/W) = TSCTR counter runs until = 0 2h (R/W) = TSCTR counter is unaffected by emulation suspend (Run Free) 3h (R/W) = TSCTR counter is unaffected by emulation suspend (Run Free)
13-9	PRESCALE	R/W	0h	Event Filter prescale select Reset type: SYSRSn 0h (R/W) = Divide by 1 (i.e,. no prescale, by-pass the prescaler) 1h (R/W) = Divide by 2 2h (R/W) = Divide by 4 3h (R/W) = Divide by 6 4h (R/W) = Divide by 8 5h (R/W) = Divide by 10 1Eh (R/W) = Divide by 60 1Fh (R/W) = Divide by 62
8	CAPLDEN	R/W	0h	Enable Loading of CAP1-4 registers on a capture event. Note that this bit does not disable CEVTn events from being generated. Reset type: SYSRSn 0h (R/W) = Disable CAP1-4 register loads at capture event time. 1h (R/W) = Enable CAP1-4 register loads at capture event time.
7	CTRRST4	R/W	0h	Counter Reset on Capture Event 4 Reset type: SYSRSn 0h (R/W) = Do not reset counter on Capture Event 4 (absolute time stamp operation) 1h (R/W) = Reset counter after Capture Event 4 time-stamp has been captured (used in difference mode operation)
6	CAP4POL	R/W	0h	Capture Event 4 Polarity select Reset type: SYSRSn 0h (R/W) = Capture Event 4 triggered on a rising edge (RE) 1h (R/W) = Capture Event 4 triggered on a falling edge (FE)

Table 20-20. ECCTL1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
5	CTRRST3	R/W	0h	Counter Reset on Capture Event 3 Reset type: SYSRSn 0h (R/W) = Do not reset counter on Capture Event 3 (absolute time stamp) 1h (R/W) = Reset counter after Event 3 time-stamp has been captured (used in difference mode operation)
4	CAP3POL	R/W	0h	Capture Event 3 Polarity select Reset type: SYSRSn 0h (R/W) = Capture Event 3 triggered on a rising edge (RE) 1h (R/W) = Capture Event 3 triggered on a falling edge (FE)
3	CTRRST2	R/W	0h	Counter Reset on Capture Event 2 Reset type: SYSRSn 0h (R/W) = Do not reset counter on Capture Event 2 (absolute time stamp) 1h (R/W) = Reset counter after Event 2 time-stamp has been captured (used in difference mode operation)
2	CAP2POL	R/W	0h	Capture Event 2 Polarity select Reset type: SYSRSn 0h (R/W) = Capture Event 2 triggered on a rising edge (RE) 1h (R/W) = Capture Event 2 triggered on a falling edge (FE)
1	CTRRST1	R/W	0h	Counter Reset on Capture Event 1 Reset type: SYSRSn 0h (R/W) = Do not reset counter on Capture Event 1 (absolute time stamp) 1h (R/W) = Reset counter after Event 1 time-stamp has been captured (used in difference mode operation)
0	CAP1POL	R/W	0h	Capture Event 1 Polarity select Reset type: SYSRSn 0h (R/W) = Capture Event 1 triggered on a rising edge (RE) 1h (R/W) = Capture Event 1 triggered on a falling edge (FE)

9 ECCTL2 Register (Offset = 2Ah) [Reset = 0006h]

ECCTL2 is shown in Figure 20-25 and described in Table 20-13.

Return to the Summary Table.

Capture Control Register 2

Figure 20-25. ECCTL2 Register

15	14	13	12	11	10	9	8
MODCNTRSTS		DMAEVTSEL		CTRFILTRESET	APWMPOL	CAP_APWM	SWSYNC
R-0h		R/W-0h		R-0/W1C-0h	R/W-0h	R/W-0h	R-0/W1S-0h
7	6	5	4	3	2	1	0
SYNCO_SEL		SYNCI_EN	TSCTRSTOP	REARM	STOP_WRAP		CONT_ONESHT
R/W-0h		R/W-0h	R/W-0h	R-0/W1S-0h	R/W-3h		R/W-0h

Table 20-22. ECCTL2 Register Field Descriptions

Bit	Field	Type	Reset	Description
15-14	MODCNTRSTS	R	0h	This bit field reads current status on modulo counter 00b (R) = CAP1 register gets loaded on next capture event. 01b (R) = CAP2 register gets loaded on next capture event. 10b (R) = CAP3 register gets loaded on next capture event. 11b (R) = CAP4 register gets loaded on next capture event. Reset type: CPU1.SYSRSn
13-12	DMAEVTSEL	R/W	0h	DMA event select 00b (R/W) = DMA interrupt source is CEVT1 01b (R/W) = DMA interrupt source is CEVT2 10b (R/W) = DMA interrupt source is CEVT3 11b (R/W) = DMA interrupt source is CEVT4 Note: ECCTL1.CAPLDEN also needs to be set to '1' for ECAPxDMA_INT to be generated Reset type: CPU1.SYSRSn
11	CTRFILTRESET	R-0/W1C	0h	Reset Bit 0h (R) = No effect 1h (W) = Resets event filter, counter, modulo counter and CEVT[1,2,3,4] and CNTOVF , HRERROR flags Note: This provides an ability start capture module from known state in case spurious inputs are captured while ECAP is configured. Reset type: CPU1.SYSRSn
10	APWMPOL	R/W	0h	APWM output polarity select. This is applicable only in APWM operating mode. Reset type: SYSRSn 0h (R/W) = Output is active high (Compare value defines high time) 1h (R/W) = Output is active low (Compare value defines low time)

Table 20-22. ECCTL2 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
9	CAP_APWM	R/W	0h	CAP/APWM operating mode select Reset type: SYSRSn 0h (R/W) = ECAP module operates in capture mode. This mode forces the following configuration: - Inhibits TSCTR resets via CTR = PRD event - Inhibits shadow loads on CAP1 and 2 registers - Permits user to enable CAP1-4 register load - CAPx/APWMx pin operates as a capture input 1h (R/W) = ECAP module operates in APWM mode. This mode forces the following configuration: - Resets TSCTR on CTR = PRD event (period boundary - Permits shadow loading on CAP1 and 2 registers - Disables loading of time-stamps into CAP1-4 registers - CAPx/APWMx pin operates as a APWM output
8	SWSYNC	R-0/W1S	0h	Software-forced Counter (TSCTR) Synchronizer. This provides the user a method to generate a synchronization pulse through software. In APWM mode, the synchronization pulse can also be sourced from the CTR = PRD event. Reset type: SYSRSn 0h (R/W) = Writing a zero has no effect. Reading always returns a zero 1h (R/W) = Writing a one forces a TSCTR shadow load of current ECAP module and any ECAP modules down-stream providing the SYNCO_SEL bits are 0,0. After writing a 1, this bit returns to a zero.
7-6	SYNCO_SEL	R/W	0h	Sync-Out Select Reset type: SYSRSn 0h (R/W) = sync out signal is SWSYNC 1h (R/W) = Select CTR = PRD event to be the sync-out signal. Note: Selection CTR = PRD is meaningful only in APWM mode 2h (R/W) = Disable sync out signal 3h (R/W) = Disable sync out signal
5	SYNCI_EN	R/W	0h	Counter (TSCTR) Sync-In select mode Reset type: SYSRSn 0h (R/W) = Disable sync-in option 1h (R/W) = Enable counter (TSCTR) to be loaded from CTRPHS register upon either a SYNCI signal or a S/W force event.
4	TSCTRSTOP	R/W	0h	Time Stamp (TSCTR) Counter Stop (freeze) Control Reset type: SYSRSn 0h (R/W) = TSCTR stopped 1h (R/W) = TSCTR free-running
3	REARM	R-0/W1S	0h	Re-Arming Control. Note: The re-arm function is valid in one shot or continuous mode Reset type: SYSRSn 0h (R/W) = Has no effect (reading always returns a 0) 1h (R/W) = Arms the one-shot sequence as follows: 1) Resets the Mod4 counter to zero 2) Unfreezes the Mod4 counter 3) Enables capture register loads

Table 20-22. ECCTL2 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
2-1	STOP_WRAP	R/W	3h	Stop value for one-shot mode. This is the number (between 1-4) of captures allowed to occur before the CAP(1-4) registers are frozen, that is, capture sequence is stopped. Wrap value for continuous mode. This is the number (between 1-4) of the capture register in which the circular buffer wraps around and starts again. Notes: STOP_WRAP is compared to Mod4 counter and, when equal, 2 actions occur: - Mod4 counter is stopped (frozen) - Capture register loads are inhibited In one-shot mode, further interrupt events are blocked until re-armed. Reset type: SYSRSn 0h (R/W) = Stop after Capture Event 1 in one-shot mode Wrap after Capture Event 1 in continuous mode. 1h (R/W) = Stop after Capture Event 2 in one-shot mode Wrap after Capture Event 2 in continuous mode. 2h (R/W) = Stop after Capture Event 3 in one-shot mode Wrap after Capture Event 3 in continuous mode. 3h (R/W) = Stop after Capture Event 4 in one-shot mode Wrap after Capture Event 4 in continuous mode.
0	CONT_ONESHT	R/W	0h	Continuous or one-shot mode control (applicable only in capture mode) Reset type: SYSRSn 0h (R/W) = Operate in continuous mode 1h (R/W) = Operate in one-Shot mode

10 ECEINT Register (Offset = 2Ch) [Reset = 0000h]

ECEINT is shown in Figure 20-26 and described in Table 20-14.

Return to the Summary Table.

The interrupt enable bits (CEVT1, ...) block any of the selected events from generating an interrupt. Events will still be latched into the flag bit (ECFLG register) and can be forced/cleared via the ECFRC/ECCLR registers. The proper procedure for configuring peripheral modes and interrupts is as follows:
- Disable global interrupts
- Stop eCAP counter
- Disable eCAP interrupts
- Configure peripheral registers
- Clear spurious eCAP interrupt flags
- Enable eCAP interrupts
- Start eCAP counter
- Enable global interrupts

Figure 20-26. ECEINT Register

15	14	13	12	11	10	9	8
RESERVED							RESERVED
R-0h							R/W-0h
7	6	5	4	3	2	1	0
CTR_EQ_CMP	CTR_EQ_PRD	CTROVF	CEVT4	CEVT3	CEVT2	CEVT1	RESERVED
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R-0h

Table 20-24. ECEINT Register Field Descriptions

Bit	Field	Type	Reset	Description
15-9	RESERVED	R	0h	Reserved
8	RESERVED	R/W	0h	Reserved
7	CTR_EQ_CMP	R/W	0h	Counter Equal Compare Interrupt Enable Reset type: SYSRSn 0h (R/W) = Disable Compare Equal as an Interrupt source 1h (R/W) = Enable Compare Equal as an Interrupt source
6	CTR_EQ_PRD	R/W	0h	Counter Equal Period Interrupt Enable Reset type: SYSRSn 0h (R/W) = Disable Period Equal as an Interrupt source 1h (R/W) = Enable Period Equal as an Interrupt source
5	CTROVF	R/W	0h	Counter Overflow Interrupt Enable Reset type: SYSRSn 0h (R/W) = Disabled counter Overflow as an Interrupt source 1h (R/W) = Enable counter Overflow as an Interrupt source
4	CEVT4	R/W	0h	Capture Event 4 Interrupt Enable Reset type: SYSRSn 0h (R/W) = Disable Capture Event 4 as an Interrupt source 1h (R/W) = Capture Event 4 Interrupt Enable
3	CEVT3	R/W	0h	Capture Event 3 Interrupt Enable Reset type: SYSRSn 0h (R/W) = Disable Capture Event 3 as an Interrupt source 1h (R/W) = Enable Capture Event 3 as an Interrupt source

ADVANCE INFORMATION

Table 20-24. ECEINT Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
2	CEVT2	R/W	0h	Capture Event 2 Interrupt Enable Reset type: SYSRSn 0h (R/W) = Disable Capture Event 2 as an Interrupt source 1h (R/W) = Enable Capture Event 2 as an Interrupt source
1	CEVT1	R/W	0h	Capture Event 1 Interrupt Enable Reset type: SYSRSn 0h (R/W) = Disable Capture Event 1 as an Interrupt source 1h (R/W) = Enable Capture Event 1 as an Interrupt source
0	RESERVED	R	0h	Reserved

11 ECFLG Register (Offset = 2Eh) [Reset = 0000h]

ECFLG is shown in Figure 20-27 and described in Table 20-15.

Return to the Summary Table.

Capture Interrupt Flag Register

Figure 20-27. ECFLG Register

15	14	13	12	11	10	9	8
RESERVED							RESERVED
R-0h							R-0h
7	6	5	4	3	2	1	0
CTR_CMP	CTR_PRD	CTROVF	CEVT4	CEVT3	CEVT2	CEVT1	INT
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

Table 20-26. ECFLG Register Field Descriptions

Bit	Field	Type	Reset	Description
15-9	RESERVED	R	0h	Reserved
8	RESERVED	R	0h	Reserved
7	CTR_CMP	R	0h	Compare Equal Compare Status Flag. This flag is active only in APWM mode. Reset type: SYSRSn 0h (R/W) = Indicates no event occurred 1h (R/W) = Indicates the counter (TSCTR) reached the compare register value (ACMP)
6	CTR_PRD	R	0h	Counter Equal Period Status Flag. This flag is only active in APWM mode. Reset type: SYSRSn 0h (R/W) = Indicates no event occurred 1h (R/W) = Indicates the counter (TSCTR) reached the period register value (APRD) and was reset.
5	CTROVF	R	0h	Counter Overflow Status Flag. This flag is active in CAP and APWM mode. Reset type: SYSRSn 0h (R/W) = Indicates no event occurred 1h (R/W) = Indicates the counter (TSCTR) has made the transition from FFFFFFFF to 00000000
4	CEVT4	R	0h	Capture Event 4 Status Flag This flag is only active in CAP mode. Reset type: SYSRSn 0h (R/W) = Indicates no event occurred 1h (R/W) = Indicates the fourth event occurred at ECAPx pin
3	CEVT3	R	0h	Capture Event 3 Status Flag. This flag is active only in CAP mode. Reset type: SYSRSn 0h (R/W) = Indicates no event occurred 1h (R/W) = Indicates the third event occurred at ECAPx pin.
2	CEVT2	R	0h	Capture Event 2 Status Flag. This flag is only active in CAP mode. Reset type: SYSRSn 0h (R/W) = Indicates no event occurred 1h (R/W) = Indicates the second event occurred at ECAPx pin.

Table 20-26. ECFLG Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
1	CEVT1	R	0h	Capture Event 1 Status Flag. This flag is only active in CAP mode. Reset type: SYSRSn 0h (R/W) = Indicates no event occurred 1h (R/W) = Indicates the first event occurred at ECAPx pin.
0	INT	R	0h	Global Interrupt Status Flag Reset type: SYSRSn 0h (R/W) = Indicates no event occurred 1h (R/W) = Indicates that an interrupt was generated.

12 ECCLR Register (Offset = 30h) [Reset = 0000h]

ECCLR is shown in Figure 20-28 and described in Table 20-16.

Return to the Summary Table.

Capture Interrupt Clear Register

Figure 20-28. ECCLR Register

15	14	13	12	11	10	9	8
RESERVED							RESERVED
R-0h							R-0/W1C-0h
7	6	5	4	3	2	1	0
CTR_CMP	CTR_PRD	CTROVF	CEVT4	CEVT3	CEVT2	CEVT1	INT
R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h

Table 20-28. ECCLR Register Field Descriptions

Bit	Field	Type	Reset	Description
15-9	RESERVED	R	0h	Reserved
8	RESERVED	R-0/W1C	0h	Reserved
7	CTR_CMP	R-0/W1C	0h	Counter Equal Compare Status Clear Reset type: SYSRSn 0h (R/W) = Writing a 0 has no effect. Always reads back a 0 1h (R/W) = Writing a 1 clears the CTR=CMP flag.
6	CTR_PRD	R-0/W1C	0h	Counter Equal Period Status Clear Reset type: SYSRSn 0h (R/W) = Writing a 0 has no effect. Always reads back a 0 1h (R/W) = Writing a 1 clears the CTR=PRD flag.
5	CTROVF	R-0/W1C	0h	Counter Overflow Status Clear Reset type: SYSRSn 0h (R/W) = Writing a 0 has no effect. Always reads back a 0 1h (R/W) = Writing a 1 clears the CTROVF flag.
4	CEVT4	R-0/W1C	0h	Capture Event 4 Status Clear Reset type: SYSRSn 0h (R/W) = Writing a 0 has no effect. Always reads back a 0 1h (R/W) = Writing a 1 clears the CEVT4 flag.
3	CEVT3	R-0/W1C	0h	Capture Event 3 Status Clear Reset type: SYSRSn 0h (R/W) = Writing a 0 has no effect. Always reads back a 0 1h (R/W) = Writing a 1 clears the CEVT3 flag.
2	CEVT2	R-0/W1C	0h	Capture Event 2 Status Clear Reset type: SYSRSn 0h (R/W) = Writing a 0 has no effect. Always reads back a 0 1h (R/W) = Writing a 1 clears the CEVT2 flag.
1	CEVT1	R-0/W1C	0h	Capture Event 1 Status Clear Reset type: SYSRSn 0h (R/W) = Writing a 0 has no effect. Always reads back a 0 1h (R/W) = Writing a 1 clears the CEVT1 flag.

Table 20-28. ECCLR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
0	INT	R-0/W1C	0h	ECAP Global Interrupt Status Clear Reset type: SYSRSn 0h (R/W) = Writing a 0 has no effect. Always reads back a 0 1h (R/W) = Writing a 1 clears the INT flag and enable further interrupts to be generated if any of the event flags are set to 1

13 ECFRC Register (Offset = 32h) [Reset = 0000h]

ECFRC is shown in Figure 20-29 and described in Table 20-17.

Return to the Summary Table.

Capture Interrupt Force Register

Figure 20-29. ECFRC Register

15	14	13	12	11	10	9	8
RESERVED							RESERVED
R-0h							R-0/W1S-0h
7	6	5	4	3	2	1	0
CTR_CMP	CTR_PRD	CTROVF	CEVT4	CEVT3	CEVT2	CEVT1	RESERVED
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0h

Table 20-30. ECFRC Register Field Descriptions

Bit	Field	Type	Reset	Description
15-9	RESERVED	R	0h	Reserved
8	RESERVED	R-0/W1S	0h	Reserved
7	CTR_CMP	R-0/W1S	0h	Force Counter Equal Compare Interrupt. This event is only active in APWM mode. Reset type: SYSRSn 0h (R/W) = No effect. Always reads back a 0. 1h (R/W) = Writing a 1 sets the CTR_CMP flag.
6	CTR_PRD	R-0/W1S	0h	Force Counter Equal Period Interrupt. This event is only active in APWM mode. Reset type: SYSRSn 0h (R/W) = No effect. Always reads back a 0. 1h (R/W) = Writing a 1 sets the CTR_PRD flag.
5	CTROVF	R-0/W1S	0h	Force Counter Overflow Reset type: SYSRSn 0h (R/W) = No effect. Always reads back a 0. 1h (R/W) = Writing a 1 to this bit sets the CTROVF flag.
4	CEVT4	R-0/W1S	0h	Force Capture Event 4. This event is only active in CAP mode. Reset type: SYSRSn 0h (R/W) = No effect. Always reads back a 0. 1h (R/W) = Writing a 1 sets the CEVT4 flag.
3	CEVT3	R-0/W1S	0h	Force Capture Event 3. This event is only active in CAP mode. Reset type: SYSRSn 0h (R/W) = No effect. Always reads back a 0. 1h (R/W) = Writing a 1 sets the CEVT3 flag.
2	CEVT2	R-0/W1S	0h	Force Capture Event 2. This event is only active in CAP mode. Reset type: SYSRSn 0h (R/W) = No effect. Always reads back a 0. 1h (R/W) = Writing a 1 sets the CEVT2 flag.
1	CEVT1	R-0/W1S	0h	Force Capture Event 1. This event is only active in CAP mode. Reset type: SYSRSn 0h (R/W) = No effect. Always reads back a 0. 1h (R/W) = Sets the CEVT1 flag.

Table 20-30. ECFRC Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
0	RESERVED	R	0h	Reserved

14 ECAPSYNCINSEL Register (Offset = 3Ch) [Reset = 00000001h]

ECAPSYNCINSEL is shown in Figure 20-30 and described in Table 20-18.

Return to the Summary Table.

SYNC source select register

Figure 20-30. ECAPSYNCINSEL Register

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

Table 20-32. ECAPSYNCINSEL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-5	RESERVED	R	0h	Reserved
4-0	SEL	R/W	1h	These bits determines the source of SYNCIN signal. 0x0 : Disabled using SOC tieoff. 0x1-0x7F : Refer to MCPWM SYNCIN sources in MCPWM TRM chapter. Reset type: SYSRSn

Chapter 21

Enhanced Quadrature Encoder Pulse (eQEP)

The enhanced Quadrature Encoder Pulse (eQEP) module described here is a Type 2 eQEP.

The enhanced quadrature encoder pulse (eQEP) module is used for direct interface with a linear or rotary incremental encoder to get position, direction, and speed information from a rotating machine for use in a high-performance motion and position-control system.

ADVANCE INFORMATION

21.1 Introduction

An incremental encoder disk is patterned with a track of slots along the periphery, as shown in Figure 21-1. These slots create an alternating pattern of dark and light lines. The disk count is defined as the number of dark and light line pairs that occur per revolution (lines per revolution). As a rule, a second track is added to generate a signal that occurs once per revolution (index signal: QEPI), which can be used to indicate an absolute position. Encoder manufacturers identify the index pulse using different terms such as index, marker, home position, and zero reference

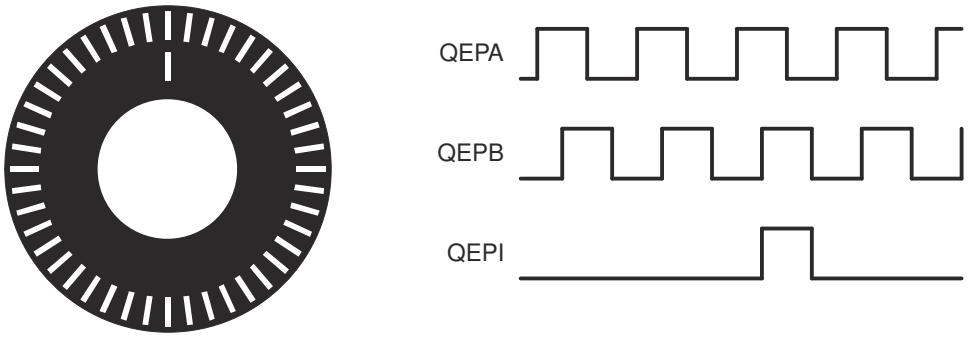
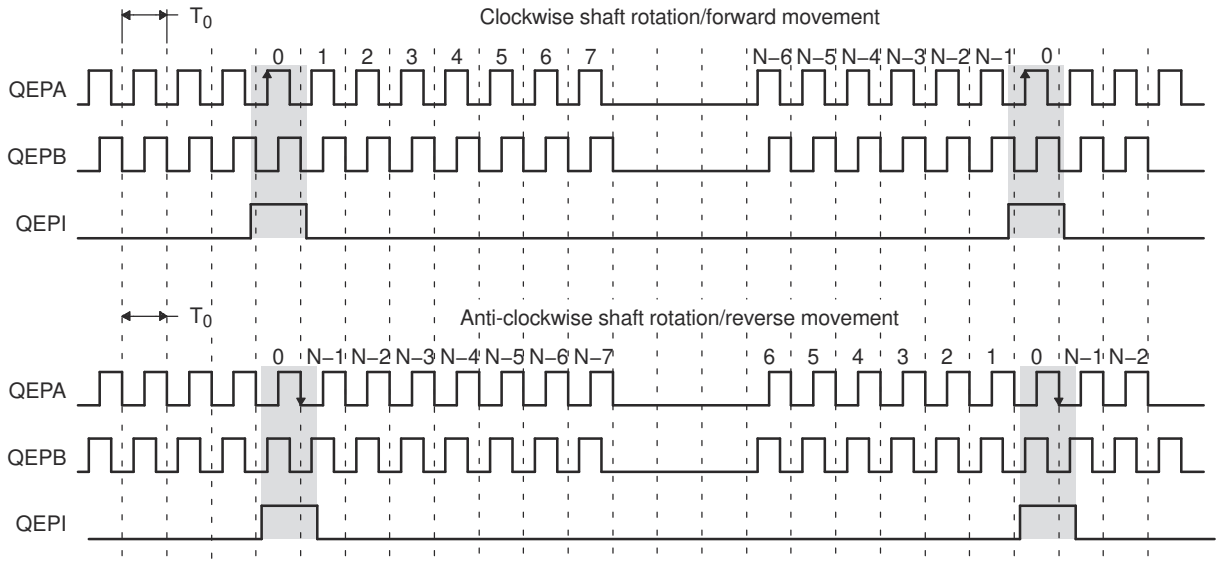


Figure 21-1. Optical Encoder Disk

To derive direction information, the lines on the disk are read out by two different photo-elements that "look" at the disk pattern with a mechanical shift of 1/4 the pitch of a line pair between them. This shift is detected with a reticle or mask that restricts the view of the photo-element to the desired part of the disk lines. As the disk rotates, the two photo-elements generate signals that are shifted 90° out of phase from each other. These are commonly called the quadrature QEPA and QEPB signals. The clockwise direction for most encoders is defined as the QEPA channel going positive before the QEPB channel and conversely, as shown in Figure 21-2.



Legend: N = lines per revolution

Figure 21-2. QEP Encoder Output Signal for Forward/Reverse Movement

The encoder wheel typically makes one revolution for every revolution of the motor, or the wheel can be at a geared rotation ratio with respect to the motor. Therefore, the frequency of the digital signal coming from the QEPA and QEPB outputs varies proportionally with the velocity of the motor. For example, a 2000-line encoder

ADVANCE INFORMATION

directly coupled to a motor running at 5000 revolutions-per-minute (rpm) results in a frequency of 166.6kHz, so by measuring the frequency of either the QEPA or QEPB output, the processor can determine the velocity of the motor.

Quadrature encoders from different manufacturers come with two forms of index pulse (gated index pulse or ungated index pulse) as shown in Figure 21-3. A nonstandard form of index pulse is ungated. In the ungated configuration, the index edges are not necessarily coincident with A and B signals. The gated index pulse is aligned to any of the four quadrature edges and width of the index pulse and can be equal to a quarter, half, or full period of the quadrature signal.

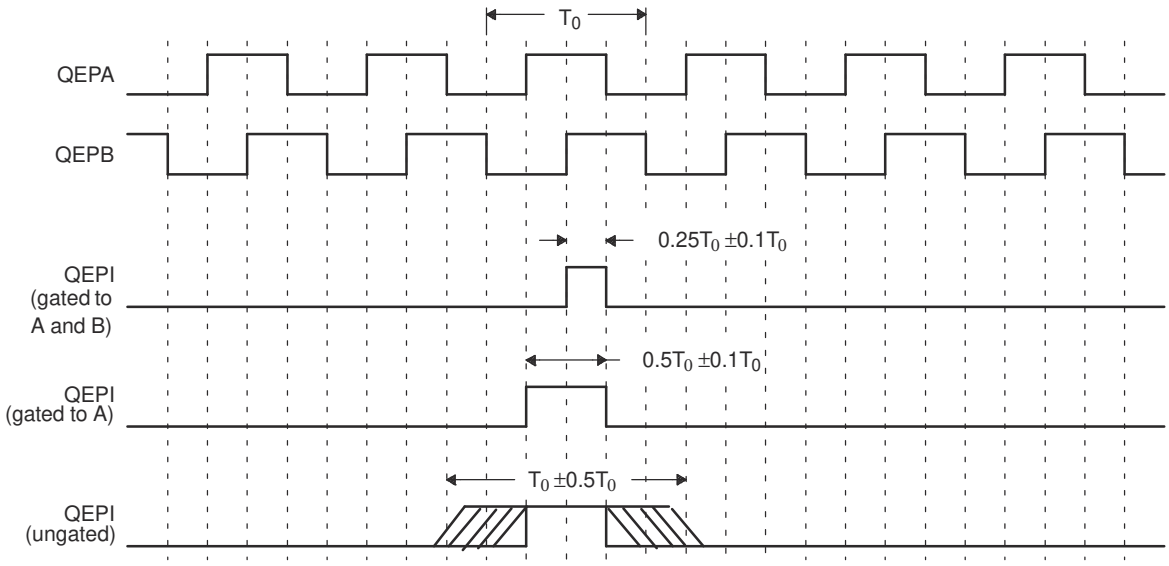


Figure 21-3. Index Pulse Example

Some typical applications of shaft encoders include robotics and computer input in the form of a mouse. Inside your mouse you can see where the mouse ball spins a pair of axles (a left/right, and an up/down axle). These axles are connected to optical shaft encoders that effectively tell the computer how fast and in what direction the mouse is moving.

General Issues: Estimating velocity from a digital position sensor is a cost-effective strategy in motor control. Two different first order approximations for velocity can be written as:

$$v(k) \approx \frac{x(k) - x(k-1)}{T} = \frac{\Delta X}{T} \quad (32)$$

$$v(k) \approx \frac{X}{t(k) - t(k-1)} = \frac{X}{\Delta T} \quad (33)$$

where:

- v(k) = Velocity at time instant k
- x(k) = Position at time instant k
- x(k-1) = Position at time instant k-1
- T = Fixed unit time or inverse of velocity calculation rate
- ΔX = Incremental position movement in unit time
- t(k) = Time instant "k"
- t(k-1) = Time instant "k-1"
- X = Fixed unit position
- ΔT = Incremental time elapsed for unit position movement

ADVANCE INFORMATION

Equation 32 is the conventional approach to velocity estimation and requires a time base to provide a unit time event for velocity calculation. Unit time is basically the inverse of the velocity calculation rate.

The encoder count (position) is read once during each unit time event. The quantity [x(k) - x(k-1)] is formed by subtracting the previous reading from the current reading. Then the velocity estimate is computed by multiplying by the known constant 1/T (where T is the constant time between unit time events and is known in advance).

Estimation based on Equation 32 has an inherent accuracy limit directly related to the resolution of the position sensor and the unit time period T. For example, consider a 500 line-per-revolution quadrature encoder with a velocity calculation rate of 400Hz. When used for position, the quadrature encoder gives a four-fold increase in resolution; in this case, 2000 counts-per-revolution. The minimum rotation that can be detected is, therefore, 0.0005 revolutions, which gives a velocity resolution of 12rpm when sampled at 400Hz. While this resolution can be satisfactory at moderate or high speeds, for example 1% error at 1200rpm, this resolution clearly proves inadequate at low speeds. In fact, at speeds below 12rpm, the speed estimate is erroneously zero much of the time.

At low speed, Equation 33 provides a more accurate approach. This method requires a position sensor that outputs a fixed interval pulse train, such as the aforementioned quadrature encoder. The width of each pulse is defined by motor speed for a given sensor resolution. Equation 33 can be used to calculate motor speed by measuring the elapsed time between successive quadrature pulse edges. However, this method suffers from the opposite limitation, as does Equation 32. A combination of relatively large motor speeds and high sensor resolution makes the time interval ΔT small, and thus more greatly influenced by the timer resolution. This can introduce considerable error into high-speed estimates.

For systems with a large speed range (that is, speed estimation is needed at both low and high speeds), one approach is to use Equation 33 at low speed and have the DSP software switch over to Equation 32 when the motor speed rises above some specified threshold.

21.1.1 EQEP Related Collateral

Foundational Materials

- Interfacing with Quadrature Encoders (Video)
- *Real-Time Control Reference Guide*
  - Refer to the Encoders section

Getting Started Materials

- *C2000™ Position Manager PTO API Reference Guide Application Report*

Expert Materials

- *CW/CCW Support on the C2000 eQEP Module Application Report*

21.2 Configuring Device Pins

There are no dedicated pinmux mode options to connect EQEP signals to the device pins. Instead, any GPIO can be used configured to supply the EQEP inputs via the Input XBAR.

See the Crossbar (XBAR) chapter for more details on the INPUTXBAR mux and settings.

21.3 Description

This section provides the eQEP inputs, memory map, and functional description.

21.3.1 EQEP Inputs

The eQEP inputs include two pins for quadrature-clock mode or direction-count mode, an index (or 0 marker), and a strobe input. The eQEP module requires that the QEPA, QEPB, and QEPI inputs are synchronized to MCLK prior to entering the module.

- **QEPA/XCLK and QEPB/XDIR:** These two pins can be used in quadrature-clock mode or direction-count mode.
  - Quadrature-clock Mode: The eQEP encoders provide two square wave signals (A and B) 90 electrical degrees out of phase. This phase relationship is used to determine the direction of rotation of the input shaft and number of eQEP pulses from the index position to derive the relative position information. For forward or clockwise rotation, QEPA signal leads QEPB signal and conversely. The quadrature decoder uses these two inputs to generate quadrature-clock and direction signals.
  - Direction-count Mode: In direction-count mode, direction and clock signals are provided directly from the external source. Some position encoders have this type of output instead of quadrature output. The QEPA pin provides the clock input and the QEPB pin provides the direction input.
- **QEPI: Index or Zero Marker:** The eQEP encoder uses an index signal to assign an absolute start position from which position information is incrementally encoded using quadrature pulses. This pin is connected to the index output of the eQEP encoder to optionally reset the position counter for each revolution. This signal can be used to initialize or latch the position counter on the occurrence of a desired event on the index pin.
- **QEPS: Strobe Input:** This general-purpose strobe signal can initialize or latch the position counter on the occurrence of a desired event on the strobe pin. This signal is typically connected to a sensor or limit switch to notify that the motor has reached a defined position.

Input signals to the eQEP (QEPA, QEPB, QEPI and QEPS) can come from multiple sources; that is, device pin, CMPSSx, Input X-BAR, or PWMXBARx. One typical use case is if SinCos transducers are used in the motor control system to estimate the position of motor shaft and Index signal is coming from traditional rotary encoder, source of the eQEP signals (QEPA, QEPB and QEPI) can be configured as output of CMPSSx that decodes the Sin, Cos, and Index signals. Figure 21-4 illustrates this.

Selection of the source of Input signals (QEPA, QEPB, and QEPI) is user-configurable through the QEPSRCSEL register as shown in Table 21-1.

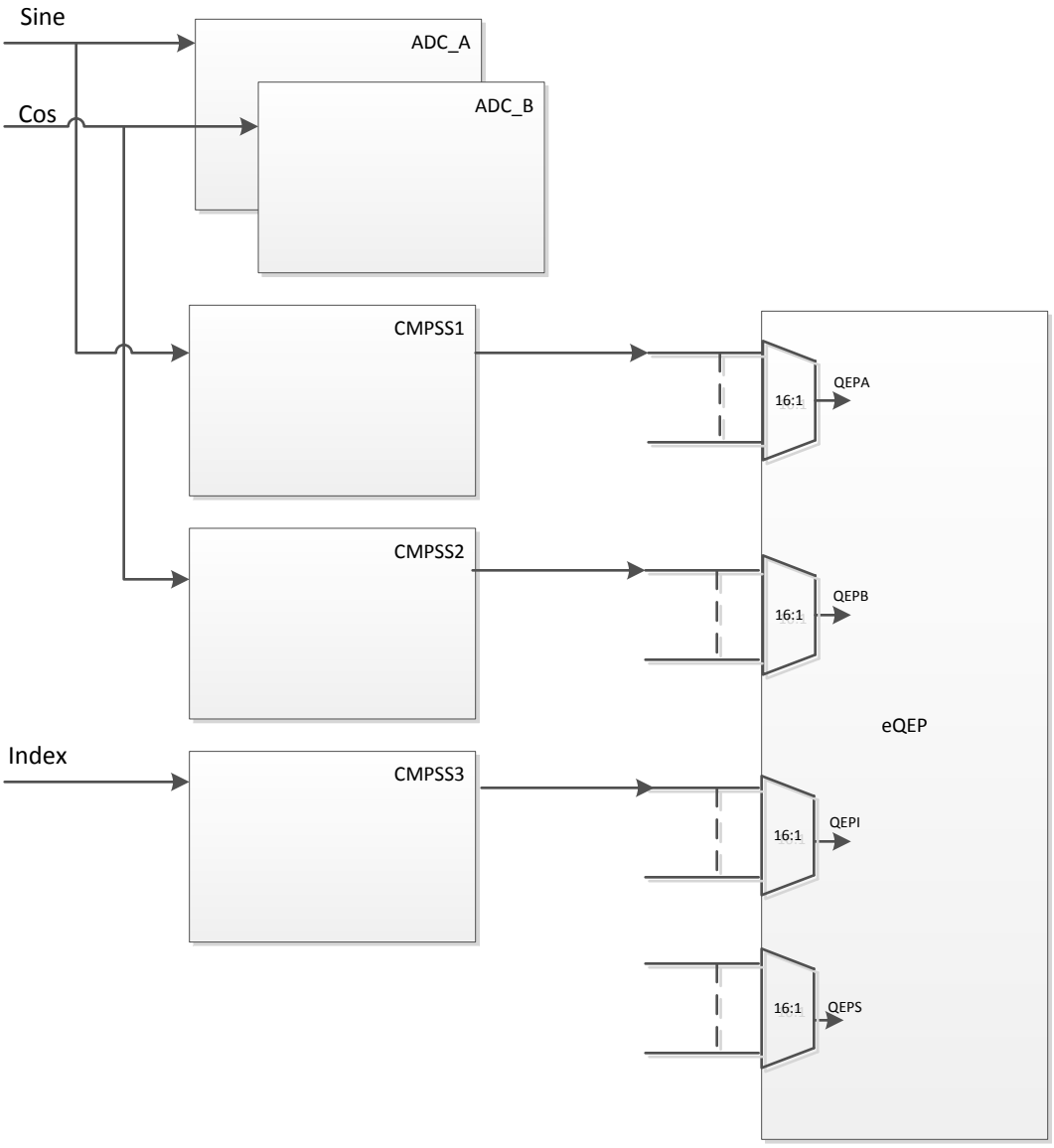


Figure 21-4. Using eQEP to Decode Signals from SinCos Transducer

Note

Configuration of QEPSRCSEL register to select the source of QEPA, QEPB, and QEPI signals can lead to unexpected transition on these signals, which can cause an undesirable outcome if eQEP is already running. Make sure that the eQEP is disabled before configuring the QEPSRCSEL register for input signals.

21.3.2 Functional Description

The eQEP peripheral contains the following major functional units (as shown in Figure 21-5):

- Programmable input qualification for each pin (part of the Input X-BAR)
- Quadrature decoder unit (QDU)
- Position counter and control unit for position measurement (PCCU)
- Quadrature edge-capture unit for low-speed measurement (QCAP)
- Unit time base for speed/frequency measurement (UTIME)
- Watchdog timer for detecting stalls (QWDOG)
- Quadrature Mode Adapter (QMA)

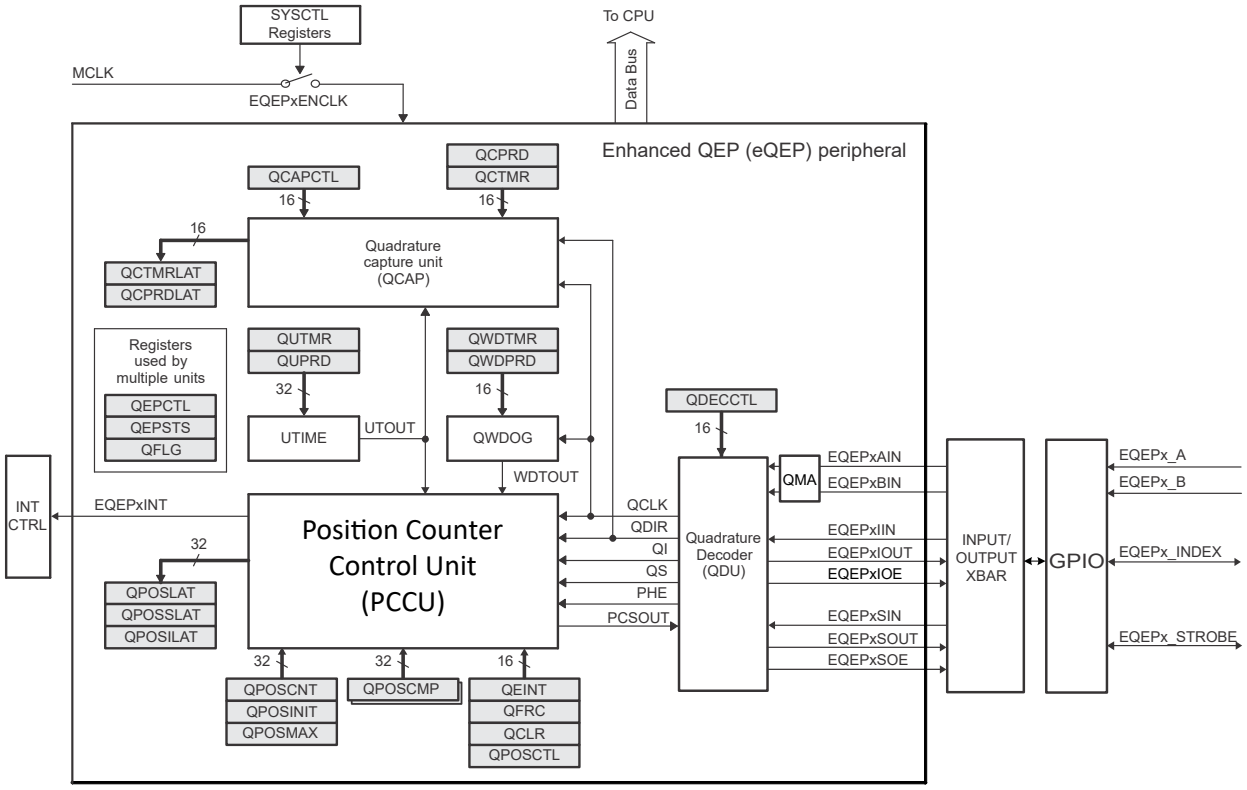


Figure 21-5. Functional Block Diagram of the eQEP Peripheral

ADVANCE INFORMATION

21.3.3 eQEP Memory Map

Table 21-2 lists the registers with the memory locations, sizes, and reset values.

Table 21-2. EQEP Memory Map

Name	Offset	Size(x16)/ #shadow	Reset	Register Description
QPOSCNT	0x00	2/0	0x0000 0000	eQEP Position Counter
QPOSINIT	0x02	2/0	0x0000 0000	eQEP Initialization Position Count
QPOSMAX	0x04	2/0	0x0000 0000	eQEP Maximum Position Count
QPOSCMP	0x06	2/1	0x0000 0000	eQEP Position-compare
QPOSILAT	0x08	2/0	0x0000 0000	eQEP Index Position Latch
QPOSSLAT	0x0A	2/0	0x0000 0000	eQEP Strobe Position Latch
QPOSLAT	0x0C	2/0	0x0000 0000	eQEP Position Latch
QUTMR	0x0E	2/0	0x0000 0000	eQEP Unit Timer
QUPRD	0x10	2/0	0x0000 0000	eQEP Unit Period Register
QWDTMR	0x12	1/0	0x0000	eQEP Watchdog Timer
QWDPRD	0x13	1/0	0x0000	eQEP Watchdog Period Register
QDECCTL	0x14	1/0	0x0000	eQEP Decoder Control Register
QEPCTL	0x15	1/0	0x0000	eQEP Control Register
QCAPCTL	0x16	1/0	0x0000	eQEP Capture Control Register
QPOSCTL	0x17	1/0	0x0000	eQEP Position-compare Control Register
QEINT	0x18	1/0	0x0000	eQEP Interrupt Enable Register
QFLG	0x19	1/0	0x0000	eQEP Interrupt Flag Register
QCLR	0x1A	1/0	0x0000	eQEP Interrupt Clear Register
QFRC	0x1B	1/0	0x0000	eQEP Interrupt Force Register
QEPSTS	0x1C	1/0	0x0000	eQEP Status Register
QCTMR	0x1D	1/0	0x0000	eQEP Capture Timer
QCPRD	0x1E	1/0	0x0000	eQEP Capture Period Register
QCTMRLAT	0x1F	1/0	0x0000	eQEP Capture Timer Latch
QCPRDLAT	0x20	1/0	0x0000	eQEP Capture Period Latch
Reserved	0x21 to 0x2F	15/0		
REV	0x30	2/0	0x0000	eQEP Revision Number
QEPSTROBESEL	0x32	2/0	0x0000	eQEP Strobe select register
QMACTRL	0x34	2/0	0x0000	eQEP QMA Control register
QEPSRCSEL	0x36	2/0	0x0000	eQEP Source Select Register
Reserved	0x38 to 0x3F	8/0		

ADVANCE INFORMATION

21.4 Quadrature Decoder Unit (QDU)

Figure 21-6 shows a functional block diagram of the QDU.

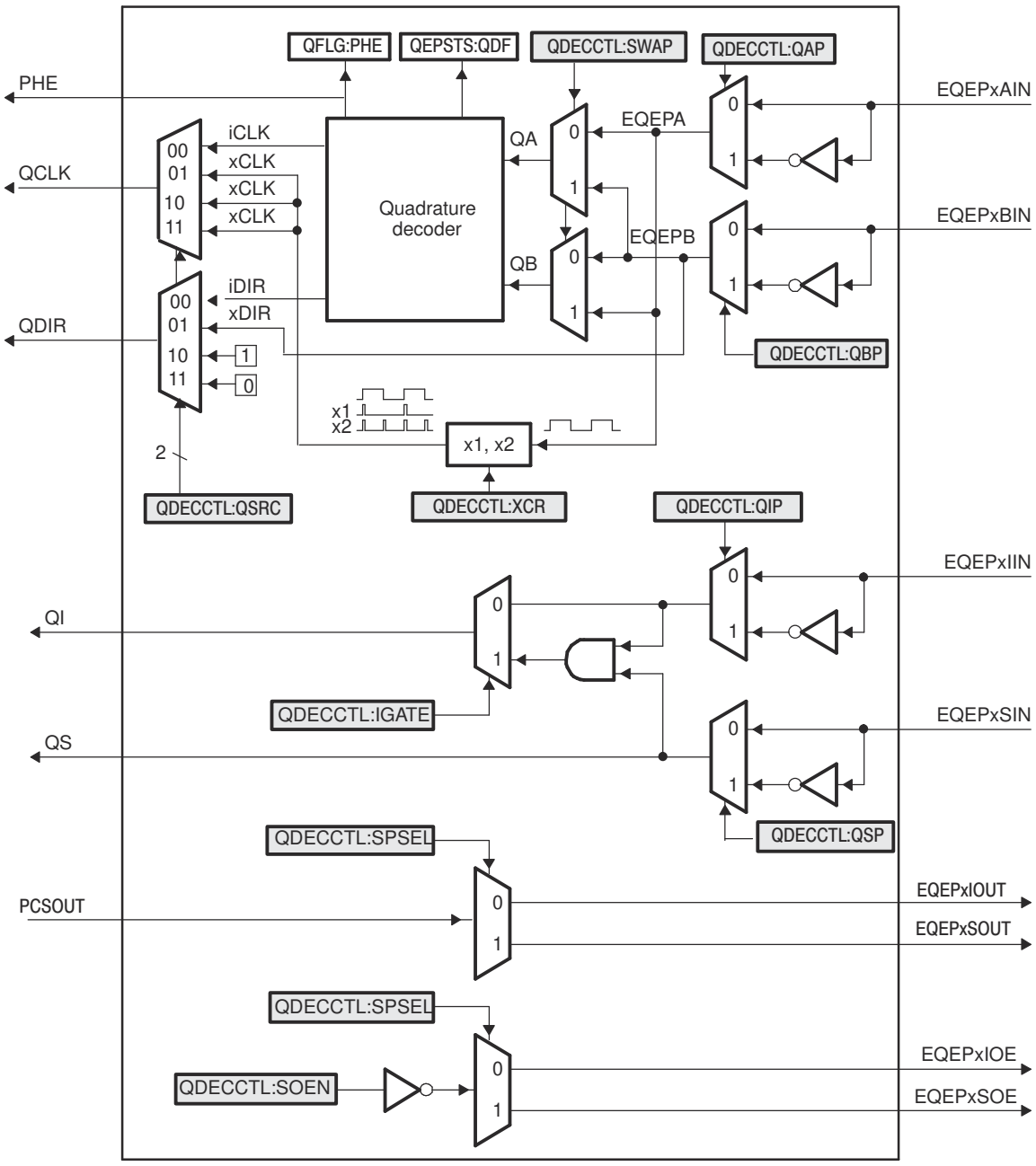


Figure 21-6. Functional Block Diagram of Decoder Unit

21.4.1 Position Counter Input Modes

Clock and direction input to the position counter is selected using QDECCTL[QSRC] bits, based on interface input requirement as follows:

- Quadrature-count mode
- Direction-count mode
- UP-count mode
- DOWN-count mode

ADVANCE INFORMATION

21.4.1.1 Quadrature Count Mode

The quadrature decoder generates the direction and clock to the position counter in quadrature count mode.

Direction Decoding

The direction decoding logic of the eQEP circuit determines which one of the sequences (QEPA, QEPB) is the leading sequence and accordingly updates the direction information in the QEPSTS[QDF] bit. Table 21-3 and Figure 21-7 show the direction decoding logic in truth table and state machine form. Both edges of the QEPA and QEPB signals are sensed to generate count pulses for the position counter. Therefore, the frequency of the clock generated by the eQEP logic is four times that of each input sequence. Figure 21-8 shows the direction decoding and clock generation from the eQEP input signals.

Table 21-3. Quadrature Decoder Truth Table

Previous Edge	Present Edge	QDIR	QPOSCNT
QA↑	QB↑	UP	Increment
	QB↓	DOWN	Decrement
	QA↓	TOGGLE	Increment or Decrement
QA↓	QB↓	UP	Increment
	QB↑	DOWN	Decrement
	QA↑	TOGGLE	Increment or Decrement
QB↑	QA↑	DOWN	Decrement
	QA↓	UP	Increment
	QB↓	TOGGLE	Increment or Decrement
QB↓	QA↓	DOWN	Decrement
	QA↑	UP	Increment
	QB↑	TOGGLE	Increment or Decrement

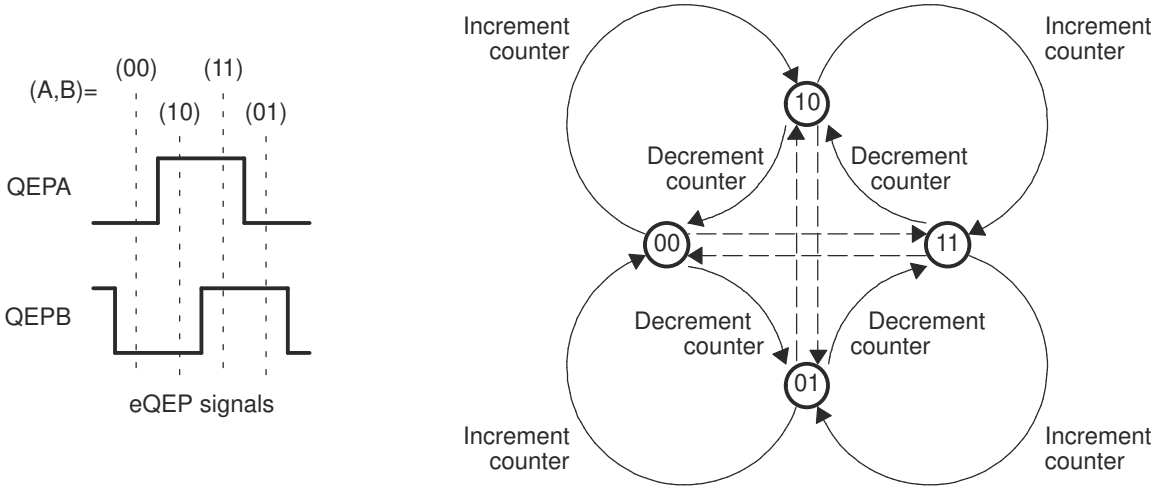


Figure 21-7. Quadrature Decoder State Machine

ADVANCE INFORMATION

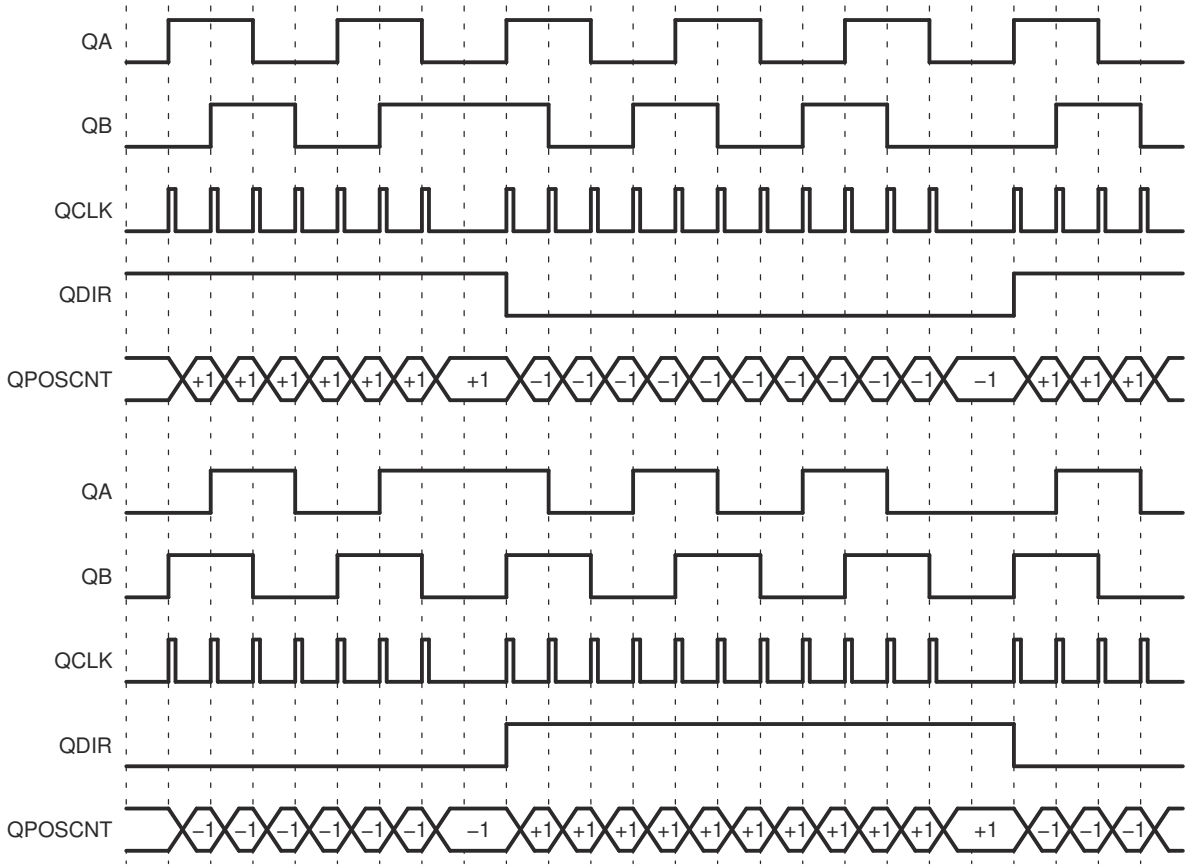


Figure 21-8. Quadrature-clock and Direction Decoding

Phase Error Flag In normal operating conditions, quadrature inputs QEPA and QEPB is 90 degrees out of phase. The phase error flag (PHE) is set in the QFLG register and the QPOSCNT value can be incorrect and offset by multiples of 1 or 3. That is, when edge transition is detected simultaneously on the QEPA and QEPB signals to optionally generate interrupts. State transitions marked by dashed lines in Figure 21-7 are invalid transitions that generate a phase error.

Count Multiplication The eQEP position counter provides 4x times the resolution of an input clock by generating a quadrature-clock (QCLK) on the rising/falling edges of both eQEP input clocks (QEPA and QEPB) as shown in Figure 21-8.

Reverse Count In normal quadrature count operation, QEPA input is applied to the QA input of the quadrature decoder and the QEPB input is applied to the QB input of the quadrature decoder. Reverse counting is enabled by setting the SWAP bit in the QDECCTL register. This swaps the input to the quadrature decoder; thereby, reversing the counting direction.

21.4.1.2 Direction-Count Mode

Some position encoders provide direction and clock outputs, instead of quadrature outputs. In such cases, direction-count mode can be used. The QEPA input provides the clock for the position counter and the QEPB input has the direction information. The position counter is incremented on every rising edge of a QEPA input when the direction input is high, and decremented when the direction input is low.

ADVANCE INFORMATION

21.4.1.3 Up-Count Mode

The counter direction signal is hard-wired for up-count and the position counter is used to measure the frequency of the QEPA input. Clearing the QDECCTL[XCR] bit enables clock generation to the position counter on both edges of the QEPA input; thereby, increasing the measurement resolution by a factor of 2x. In up-count mode, we recommend that the application not configure QEPB as a GPIO mux option, or make sure that a signal edge is not generated on the QEPB input.

21.4.1.4 Down-Count Mode

The counter direction signal is hardwired for a down-count and the position counter is used to measure the frequency of the QEPA input. Clearing the QDECCTL[XCR] bit enables clock generation to the position counter on both edges of a QEPA input, thereby increasing the measurement resolution by a factor of 2x. In down-count mode, the application must not configure QEPB as a GPIO mux option or make sure that a signal edge is not generated on the QEPB input.

21.4.2 eQEP Input Polarity Selection

Each eQEP input can be inverted using QDECCTL[8:5] control bits. As an example, setting the QDECCTL[QIP] bit inverts the index input.

21.4.3 Position-Compare Sync Output

The enhanced eQEP peripheral includes a position-compare unit that is used to generate the position-compare sync signal on compare match between the position-counter register (QPOSCNT) and the position- compare register (QPOSCMP). This sync signal can be output using an index pin or strobe pin of the EQEP peripheral.

Setting the QDECCTL[SOEN] bit enables the position-compare sync output and the QDECCTL[SPSEL] bit selects either an eQEP index pin or an eQEP strobe pin.

21.5 Position Counter and Control Unit (PCCU)

The position-counter and control unit provides two configuration registers (QEPCTL and QPOSCTL) for setting up position-counter operational modes, position-counter initialization/latch modes and position-compare logic for sync signal generation.

21.5.1 Position Counter Operating Modes

Position-counter data can be captured in different manners. In some systems, the position counter is accumulated continuously for multiple revolutions and the position-counter value provides the position information with respect to the known reference. An example of this is the quadrature encoder mounted on the motor controlling the print head in the printer. Here the position counter is reset by moving the print head to the home position and then the position counter provides absolute position information with respect to home position.

In other systems, the position counter is reset on every revolution using index pulse, and the position counter provides a rotor angle with respect to the index pulse position.

The position counter can be configured to operate in following four modes

- Position-Counter Reset on Index Event
- Position-Counter Reset on Maximum Position
- Position-Counter Reset on the first Index Event
- Position-Counter Reset on Unit Time Out Event (Frequency Measurement)

In all the above operating modes, the position counter is reset to 0 on overflow and to the QPOSMAX register value on underflow. Overflow occurs when the position counter counts up after the QPOSMAX value. Underflow occurs when the position counter counts down after 0. The Interrupt flag is set to indicate overflow/underflow in QFLG register.

21.5.1.1 Position Counter Reset on Index Event (QEPCTL[PCRM] = 00)

If the index event occurs during the forward movement, then the position counter is reset to 0 on the next eQEP clock. If the index event occurs during the reverse movement, then the position counter is reset to the value in the QPOSMAX register on the next eQEP clock.

The first index marker is defined as the quadrature edge following the first index edge. The eQEP peripheral records the occurrence of the first index marker (QEPSTS[FIMF]) and direction on the first index event marker (QEPSTS[FIDF]) in QEPSTS registers, the eQEP peripheral also remembers the quadrature edge on the first index marker so that same relative quadrature transition is used for index event reset operation.

For example, if the first reset operation occurs on the falling edge of QEPB during the forward direction, then all the subsequent reset must be aligned with the falling edge of QEPB for the forward rotation and on the rising edge of QEPB for the reverse rotation as shown in Figure 21-9.

The position-counter value is latched to the QPOSILAT register and direction information is recorded in the QEPSTS[QDLF] bit on every index event marker. The position-counter error flag (QEPSTS[PCEF]) and error interrupt flag (QFLG[PCE]) are set if the latched value is not equal to 0 or QPOSMAX. The position-counter error flag (QEPSTS[PCEF]) is updated on every index event marker and an interrupt flag (QFLG[PCE]) is set on error that can be cleared only through software.

The index event latch configuration QEPCTL[IEL] must be configured to 00 or 11 when PRCM = 0 and the position counter error flag/interrupt flag are generated only in index event reset mode. The position counter value is latched into the IPOSLAT register on every index marker.

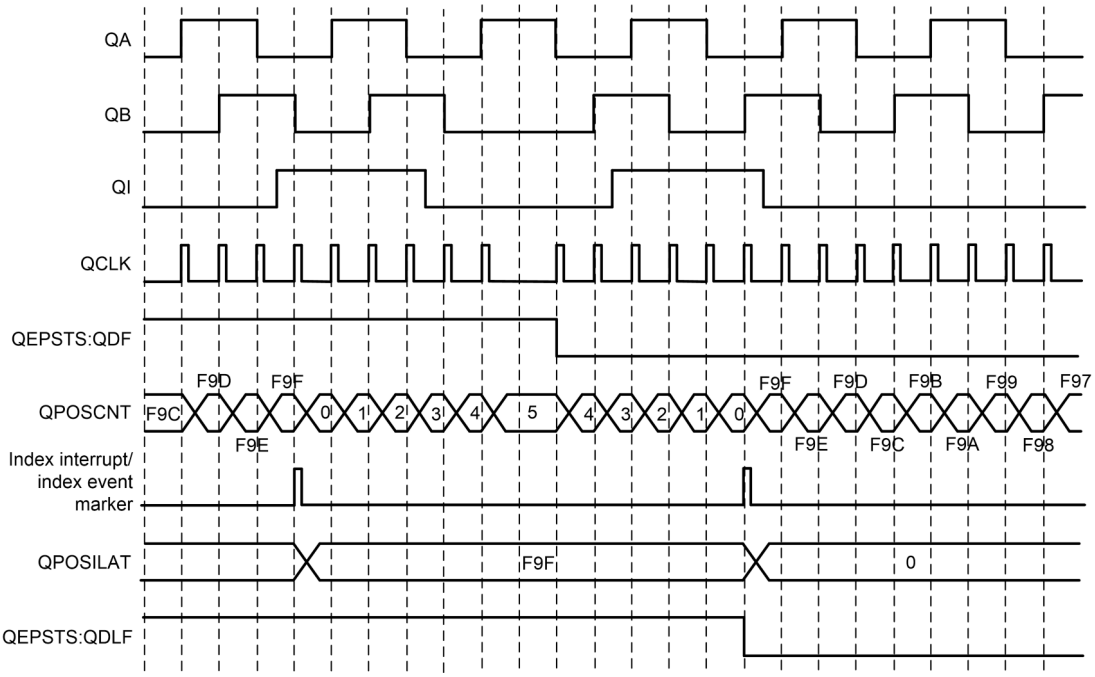


Figure 21-9. Position Counter Reset by Index Pulse for 1000-Line Encoder (QPOSMAX = 3999 or 0xF9F)

Note

In case of a boundary condition where the time period between the Index Event and the previous QCLK edge is less than MCLK period, then QPOSCNT gets reset to zero or QPOSMAX in the same MCLK cycle and does not wait for the next QCLK edge to occur.

21.5.1.2 Position Counter Reset on Maximum Position (QEPCTL[PCRM] = 01)

If the position counter is equal to QPOSMAX, then the position counter is reset to 0 on the next eQEP clock for forward movement and position counter overflow flag is set. If the position counter is equal to ZERO, then the position counter is reset to QPOSMAX on the next QEP clock for reverse movement and position-counter underflow flag is set. Figure 21-10 shows the position-counter reset operation in this mode.

The first index marker fields (QEPSTS[FIDF] and QEPSTS[FIMF]) are not applicable in this mode.

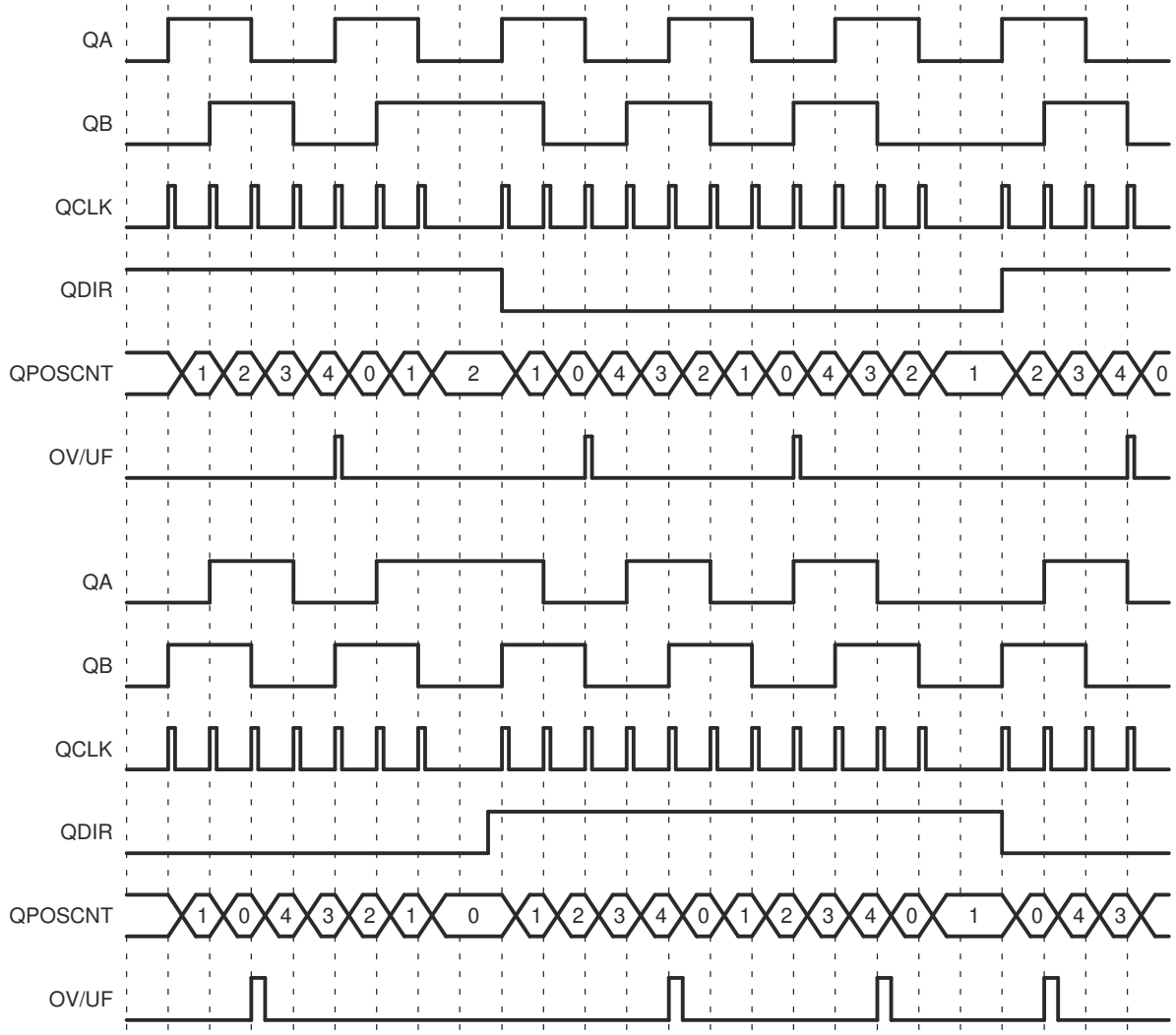


Figure 21-10. Position Counter Underflow/Overflow (QPOSMAX = 4)

21.5.1.3 Position Counter Reset on the First Index Event (QEPCTL[PCRM] = 10)

If the index event occurs during forward movement, then the position counter is reset to 0 on the next eQEP clock. If the index event occurs during the reverse movement, then the position counter is reset to the value in the QPOSMAX register on the next eQEP clock. Note that this is done only on the first occurrence and subsequently the position-counter value is not reset on an index event; rather, the position-counter value is reset based on the maximum position as described in Section 21.5.1.2 .

The first index marker fields (QEPSTS[FIDF] and QEPSTS[FIMF]) are not applicable in this mode.

21.5.1.4 Position Counter Reset on Unit Time-out Event (QEPCTL[PCRM] = 11)

In this mode, QPOSCNT is set to 0 or QPOMAX, depending on the direction mode selected by QDECCTL[QSRC] bits on a unit time event. This is useful for frequency measurement.

21.5.2 Position Counter Latch

The eQEP index and strobe input can be configured to latch the position counter (QPOSCNT) into QPOSILAT and QPOSSLAT, respectively, on occurrence of a definite event on these pins.

21.5.2.1 Index Event Latch

In some applications, it is not desirable to reset the position counter on every index event and instead it can be required to operate the position counter in full 32-bit mode (QEPCTL[PCRM] = 01 and QEPCTL[PCRM] = 10 modes).

In such cases, the eQEP position counter can be configured to latch on the following events and direction information is recorded in the QEPSTS[QDLF] bit on every index event marker.

- Latch on Rising edge (QEPCTL[IEL] = 01)
- Latch on Falling edge (QEPCTL[IEL] = 10)
- Latch on Index Event Marker (QEPCTL[IEL] = 11)

This is particularly useful as an error checking mechanism to check if the position counter accumulated the correct number of counts between index events. As an example, the 1000-line encoder must count 4000 times when moving in the same direction between the index events.

The index event latch interrupt flag (QFLG[IEL]) is set when the position counter is latched to the QPOSILAT register. The index event latch configuration bits (QEPCTL[IEL]) are ignored when QEPCTL[PCRM] = 00.

Latch on Rising Edge (QEPCTL[IEL] = 01)	The position-counter value (QPOSCNT) is latched to the QPOSILAT register on every rising edge of an index input.
Latch on Falling Edge (QEPCTL[IEL] = 10)	The position-counter value (QPOSCNT) is latched to the QPOSILAT register on every falling edge of index input.
Latch on Index Event Marker/Software Index Marker (QEPCTL[IEL] = 11	The first index marker is defined as the quadrature edge following the first index edge. The eQEP peripheral records the occurrence of the first index marker (QEPSTS[FIMF]) and the direction on the first index event marker (QEPSTS[FIDF]) in the QEPSTS registers. The eQEP peripheral also remembers the quadrature edge on the first index marker so that the same relative quadrature transition is used for latching the position counter (QEPCTL[IEL] = 11).

Figure 21-11 shows the position counter latch using an index event marker.

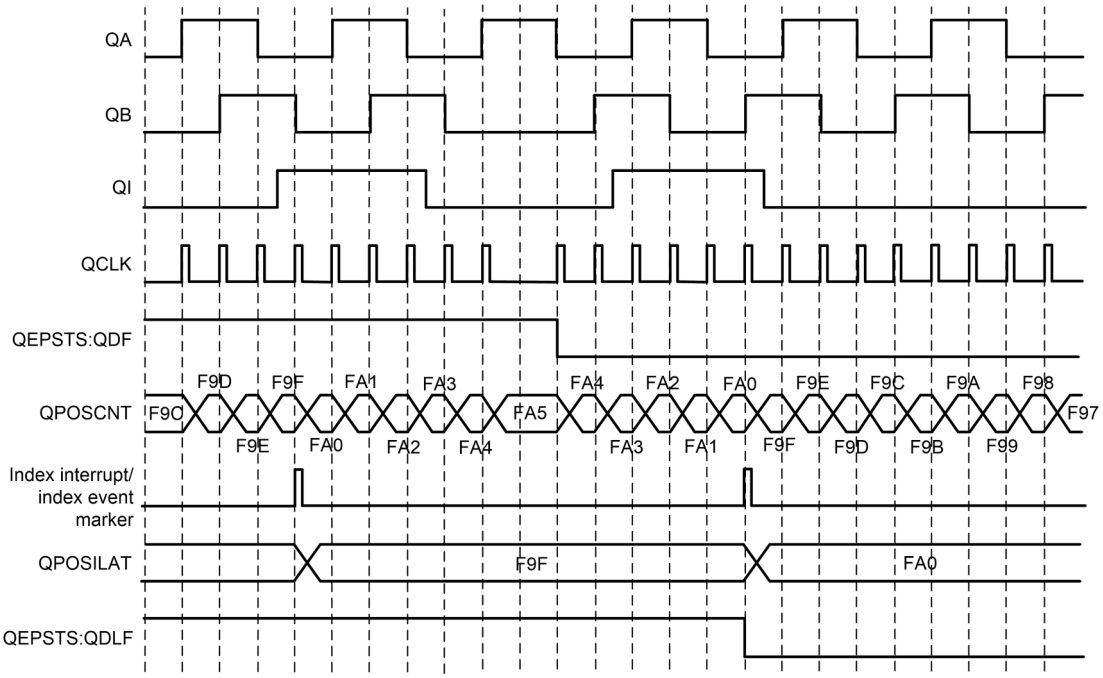


Figure 21-11. Software Index Marker for 1000-line Encoder (QEPCTL[IEL] = 1)

21.5.2.2 Strobe Event Latch

The position-counter value is latched to the QPOSSLAT register on the rising edge of the strobe input by clearing the QEPCTL[SEL] bit.

If the QEPCTL[SEL] bit is set, then the position-counter value is latched to the QPOSSLAT register on the rising edge of the strobe input for forward direction, and on the falling edge of the strobe input for reverse direction as shown in Figure 21-12.

The strobe event latch interrupt flag (QFLG[SEL) is set when the position counter is latched to the QPOSSLAT register.

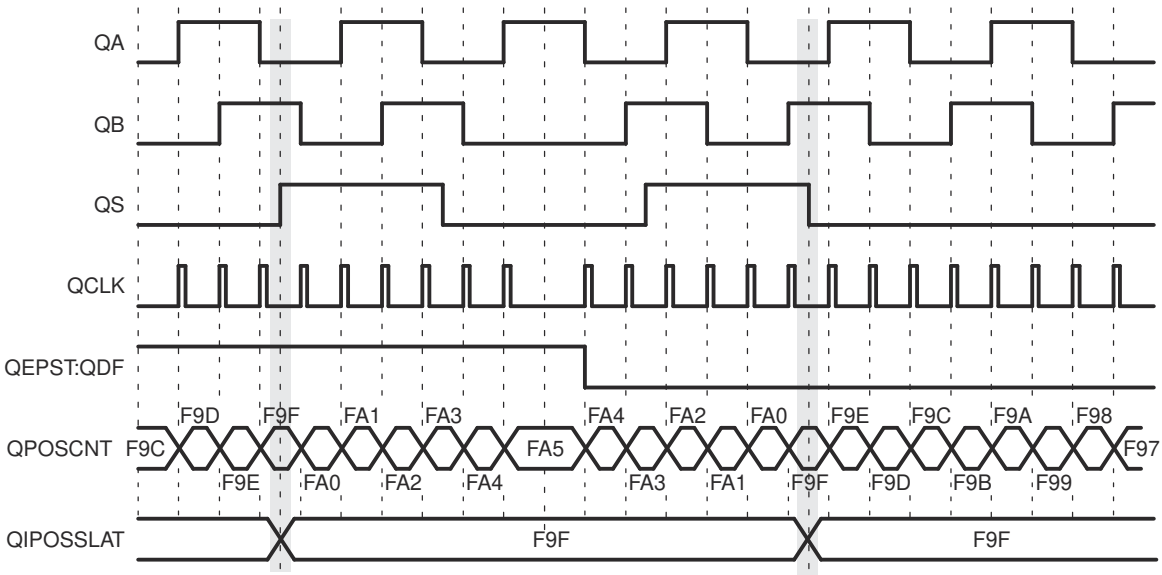


Figure 21-12. Strobe Event Latch (QEPCTL[SEL] = 1)

ADVANCE INFORMATION

There is an added feature on Type 2.0 eQEP where position-counter value can also be latched on ADCSOCA and ADCSOSCB events by configuring the register QEPSTROBESEL.STROBESEL as shown in Figure 21-13. To use the ADCSOCA/B events for the QS signal, configuration of the QEPSRCSEL.QEPSSEL to be non-zero is needed..

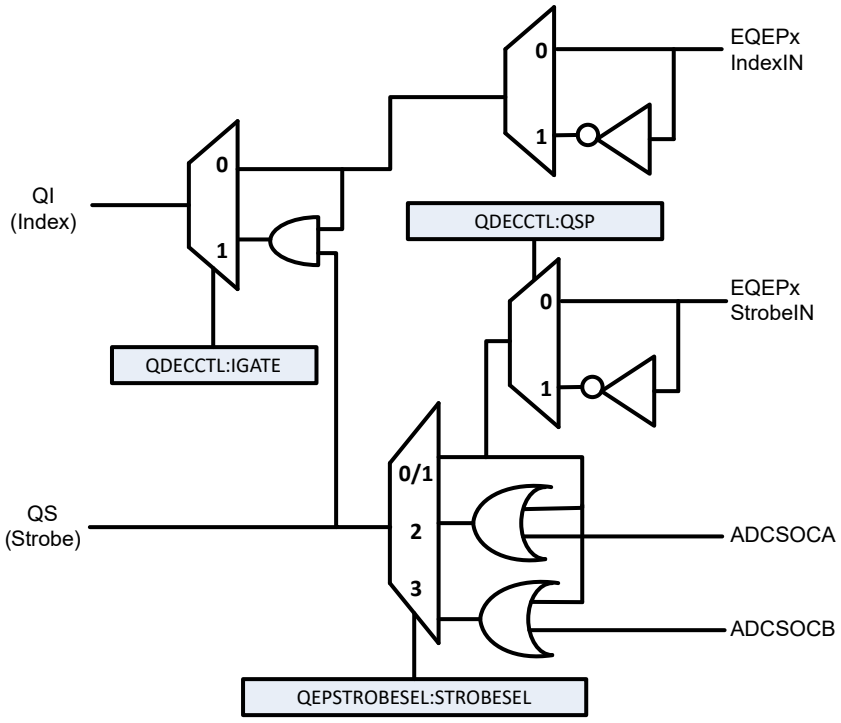


Figure 21-13. Latching Position Counter on ADCSOCA/ADCSOCB Event

21.5.3 Position Counter Initialization

The position counter can be initialized using the following events:

- Index event
- Strobe event
- Software initialization

Index Event Initialization (IEI)	The QEPI index input can be used to trigger the initialization of the position counter at the rising or falling edge of the index input. If the QEPCTL[IEI] bits are 10, then the position counter (QPOSCNT) is initialized with a value in the QPOSINIT register on the rising edge of index input. Conversely, if the QEPCTL[IEI] bits are 11, initialization is on the falling edge of the index input.
Strobe Event Initialization (SEI)	If the QEPCTL[SEI] bits are 10, then the position counter is initialized with a value in the QPOSINIT register on the rising edge of strobe input. If QEPCTL[SEL] bits are 11, then the position counter is initialized with a value in the QPOSINIT register on the rising edge of strobe input for forward direction and on the falling edge of strobe input for reverse direction.
Software Initialization (SWI)	The position counter can be initialized in software by writing a 1 to the QEPCTL[SWI] bit. This bit is not automatically cleared. While the bit is still set, if a 1 is written to the bit again, the position counter is re-initialized.

ADVANCE INFORMATION

21.5.4 eQEP Position-compare Unit

The eQEP peripheral includes a position-compare unit that is used to generate a sync output and interrupt on a position-compare match. Figure 21-14 shows a diagram. The position-compare (QPOSCMP) register is shadowed and shadow mode can be enabled or disabled using the QPOSCTL[PSSHDW] bit. If the shadow mode is not enabled, the CPU writes directly to the active position compare register.

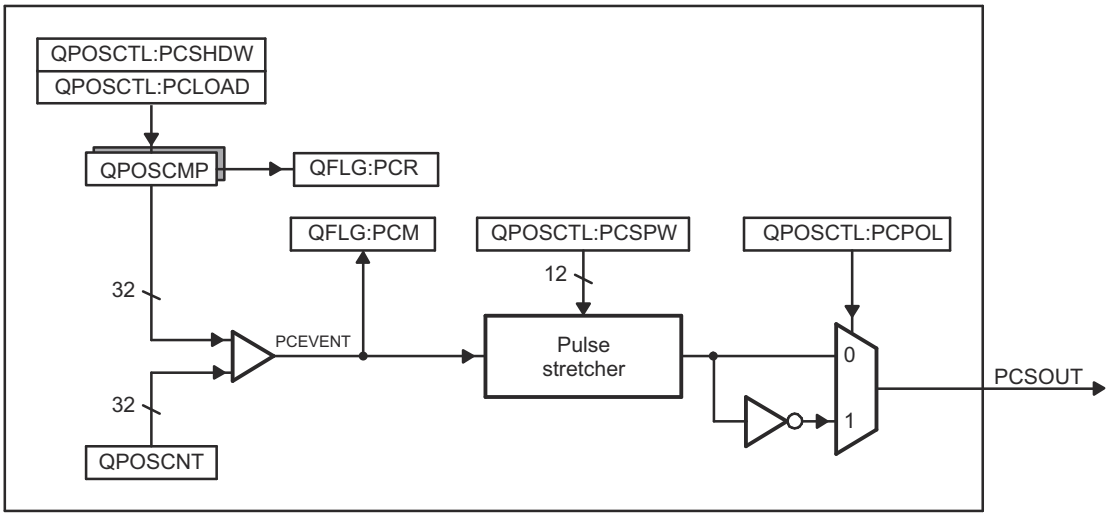


Figure 21-14. eQEP Position-compare Unit

In shadow mode, you can configure the position-compare unit (QPOSCTL[PCLOAD]) to load the shadow register value into the active register on the following events, and to generate the position-compare ready (QFLG[PCR]) interrupt after loading.

- Load on compare match
- Load on position-counter zero event

The position-compare match (QFLG[PCM]) is set when the position-counter value (QPOSCNT) matches with the active position-compare register (QPOSCMP) and the position-compare sync output of the programmable pulse width is generated on compare-match to trigger an external device.

For example, if QPOSCMP = 2, the position-compare unit generates a position-compare event on 1 to 2 transitions of the eQEP position counter for forward counting direction and on 3 to 2 transitions of the eQEP position counter for reverse counting direction (see Figure 21-15).

See the register section for the layout of the eQEP Position-Compare Control Register (QPOSCTL) and description of the QPOSCTL bit fields.

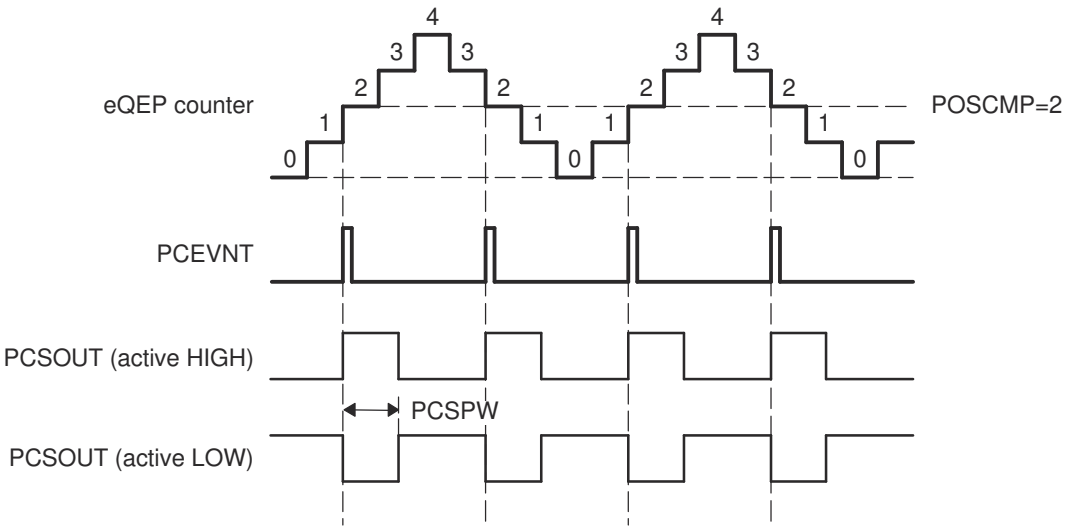


Figure 21-15. eQEP Position-compare Event Generation Points

The pulse stretcher logic in the position-compare unit generates a programmable position-compare sync pulse output on the position-compare match. In the event of a new position-compare match while a previous position-compare pulse is still active, then the pulse stretcher generates a pulse of specified duration from the new position-compare event as shown in Figure 21-16.

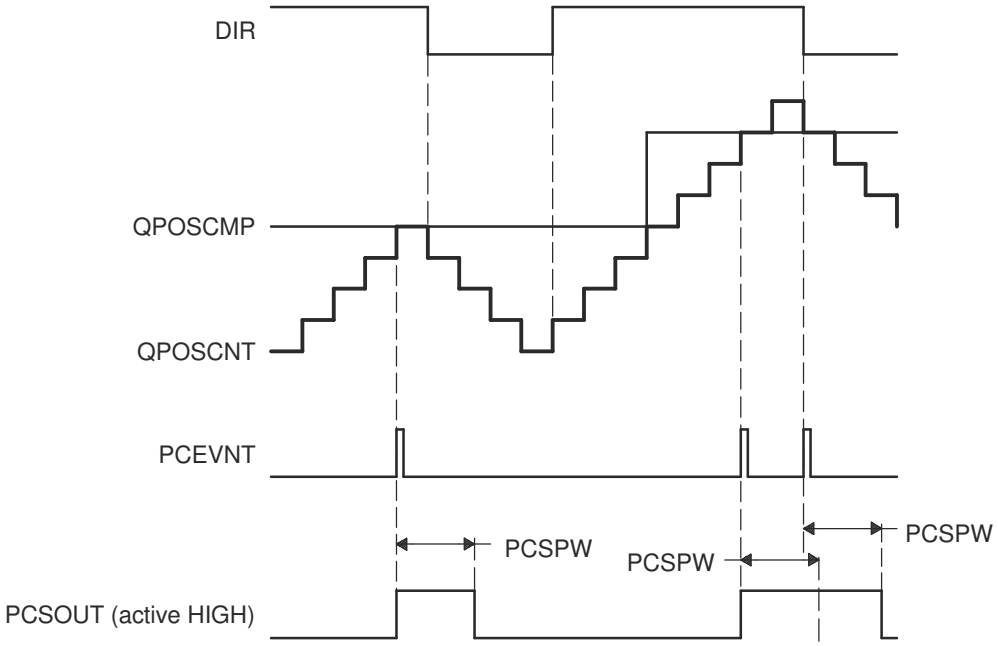


Figure 21-16. eQEP Position-compare Sync Output Pulse Stretcher

21.6 eQEP Edge Capture Unit

The eQEP peripheral includes an integrated edge capture unit to measure the elapsed time between the unit position events as shown in Figure 21-17. This feature is typically used for low-speed measurement using the following formula:

$$v(k) = \frac{X}{t(k) - t(k-1)} = \frac{X}{\Delta T} \quad (34)$$

where:

- X = Unit position is defined by integer multiple of quadrature edges (see Figure 21-18)
- ΔT = Elapsed time between unit position events
- v(k) = Velocity at time instant "k"

The eQEP capture timer (QCTMR) runs from prescaled MCLK and the prescaler is programmed by the QCAPCTL[CCPS] bits. The capture timer (QCTMR) value is latched into the capture period register (QCPRD) on every unit position event and then the capture timer is reset, a flag is set in QEPSTS:UPEVNT to indicate that new value is latched into the QCPRD register. Software can check this status flag before reading the period register for low speed measurement, and clear the flag by writing 1.

Time measurement (ΔT) between unit position events is correct if the following conditions are met:

- No more than 65,535 counts have occurred between unit position events.
- No direction change between unit position events.

If the QEP capture timer overflows between unit position events, then the timer sets the QEP capture overflow flag (QEPSTS[COEF]) in the status register.

Note

The QCPRDLAT register is not set to 0xFFFF if the QEPSTS[COEF] bit gets set.

If direction change occurs between the unit position events, then the direction error flag (QEPSTS[CDEF]) is set in the status register and the QCPRDLAT register is set to 0xFFFF.

The Capture Timer (QCTMR) and Capture Period register (QCPRD) can be configured to latch on the following events:

- CPU read of QPOSCNT register
- Unit time-out event

If the QEPCTL[QCLM] bit is cleared, then the capture timer and capture period values are latched into the QCTMRLAT and QCPRDLAT registers, respectively, when the CPU reads the position counter (QPOSCNT).

If the QEPCTL[QCLM] bit is set, then the position counter, capture timer, and capture period values are latched into the QPOSLAT, QCTMRLAT and QCPRDLAT registers, respectively, on unit time out.

Figure 21-19 shows the capture unit operation along with the position counter.

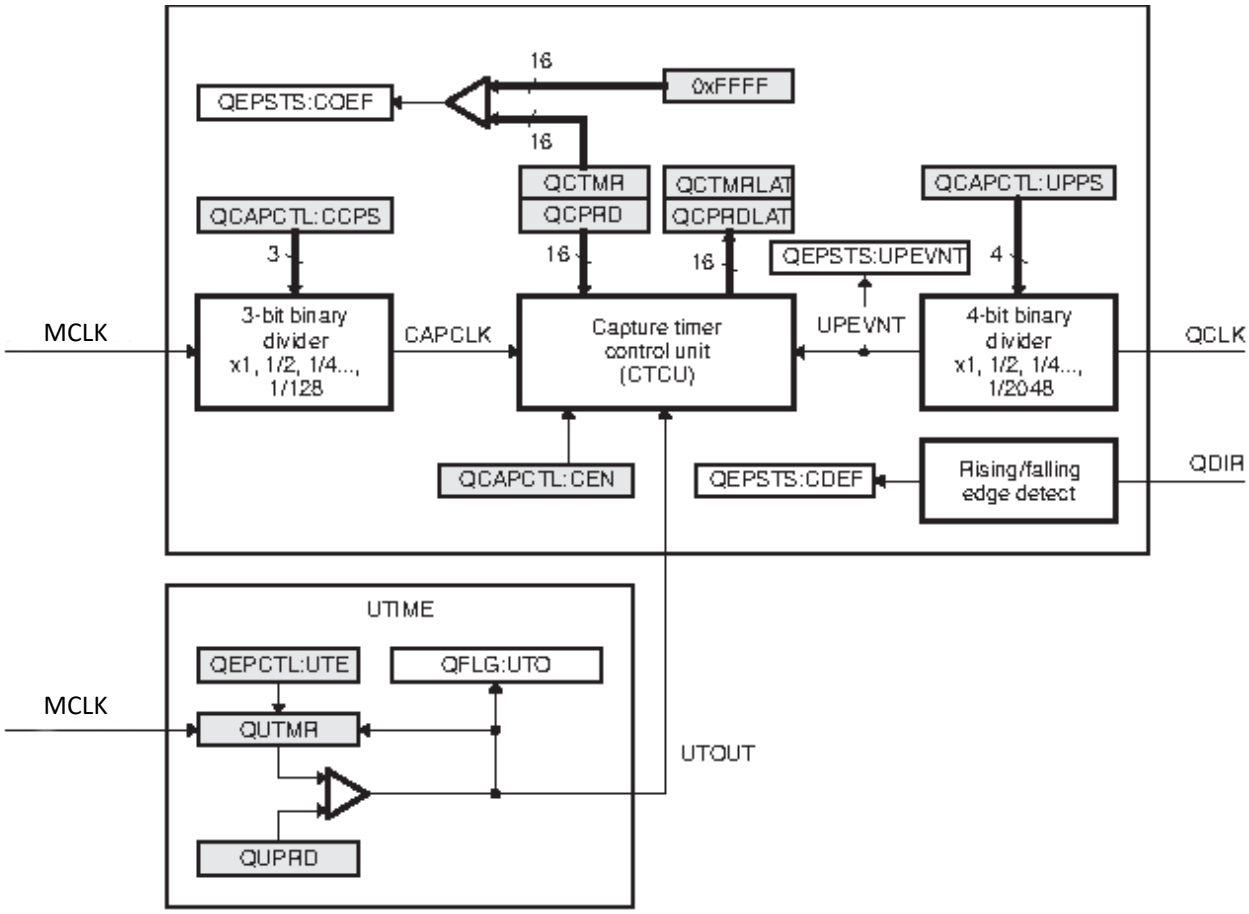
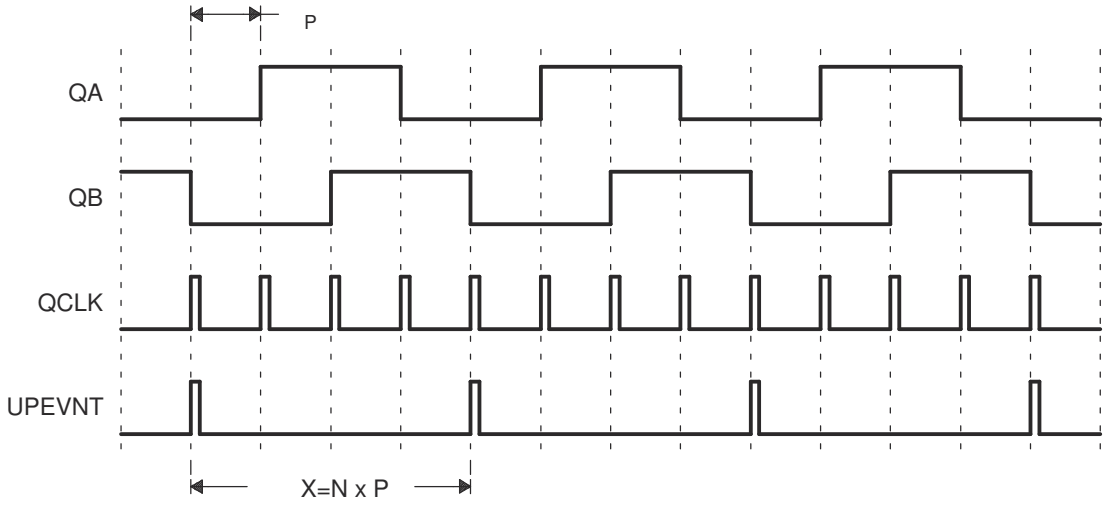


Figure 21-17. eQEP Edge Capture Unit

CAUTION

The QCAPCTL[UPPS] prescaler cannot be modified dynamically (such as switching the unit event prescaler from QCLK/4 to QCLK/8). Doing so can result in undefined behavior. The QCAPCTL[CPPS] prescaler can be modified dynamically (such as switching CAPCLK prescaling mode from MCLK/4 to MCLK/8) only after the capture unit is disabled.

ADVANCE INFORMATION



Note

N = Number of quadrature periods selected using QCAPCTL[UPPS] bits

Figure 21-18. Unit Position Event for Low Speed Measurement (QCAPCTL[UPPS] = 0010)

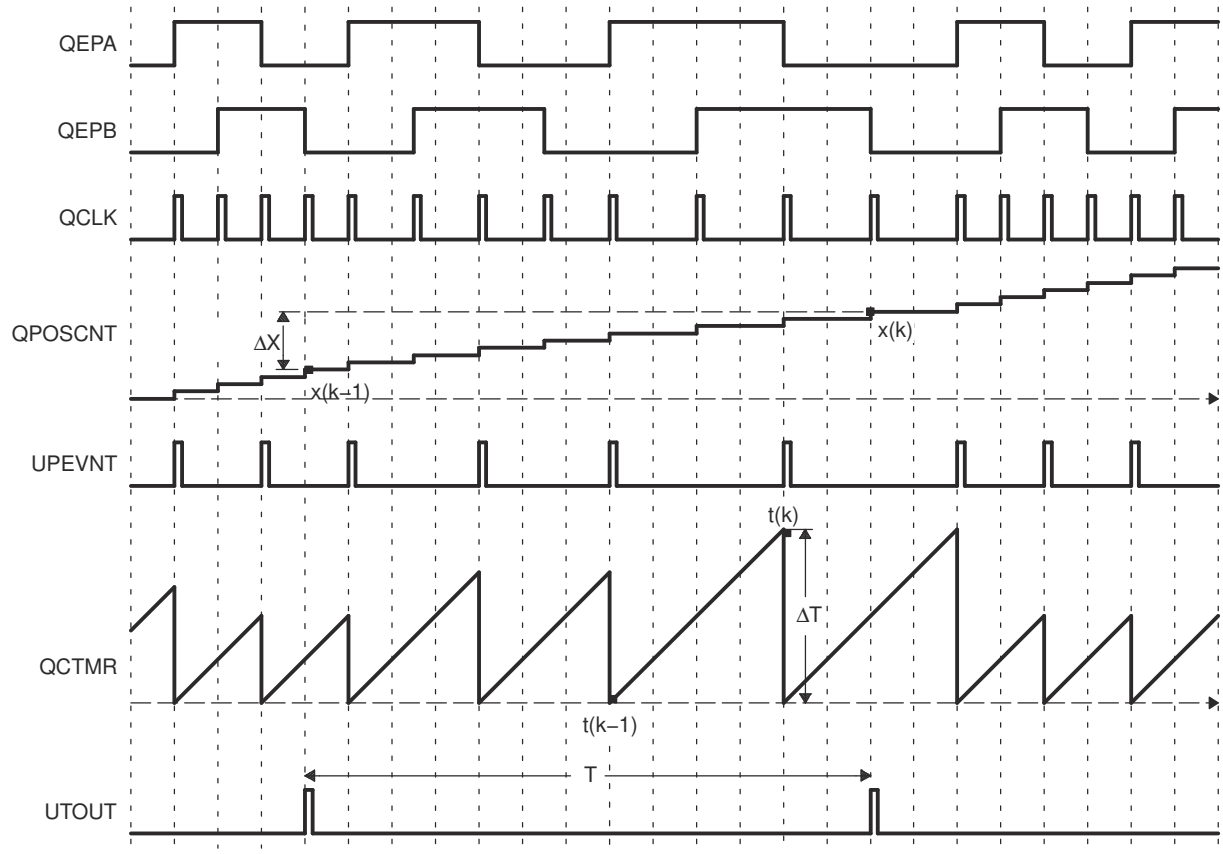


Figure 21-19. eQEP Edge Capture Unit - Timing Details

ADVANCE INFORMATION

Velocity calculation equation:

$$v(k) = \frac{x(k) - x(k-1)}{T} = \frac{\Delta X}{T} \text{ o} \quad (35)$$

where:

- v(k) = Velocity at time instant k
- x(k) = Position at time instant k
- x(k-1) = Position at time instant k-1
- T = Fixed unit time or inverse of velocity calculation rate
- ΔX = Incremental position movement in unit time
- X = Fixed unit position
- ΔT = Incremental time elapsed for unit position movement
- t(k) = Time instant "k"
- t(k-1) = Time instant "k-1"

Unit time (T) and unit period (X) are configured using the QUPRD and QCAPCTL[UPPS] registers. Incremental position output and incremental time output is available in the QPOSLAT and QCPRDLAT registers.

Parameter	Relevant Register to Configure or Read the Information
T	Unit Period Register (QUPRD)
ΔX	Incremental Position = QPOSLAT(k) - QPOSLAT(K-1)
X	Fixed-unit position defined by sensor resolution and QCAPCTL[UPPS] bits
ΔT	Capture Period Latch (QCPRDLAT)

21.7 eQEP Watchdog

The eQEP peripheral contains a 16-bit watchdog timer (Figure 21-20) that monitors the quadrature clock to indicate proper operation of the motion-control system. The eQEP watchdog timer is clocked from MCLK/64 and the quadrature clock event (pulse) resets the watchdog timer. If no quadrature clock event is detected until a period match (QWDPRD = QWDTMR), then the watchdog timer times out and the watchdog interrupt flag is set (QFLG[WTO]). The time-out value is programmable through the watchdog period register (QWDPRD).

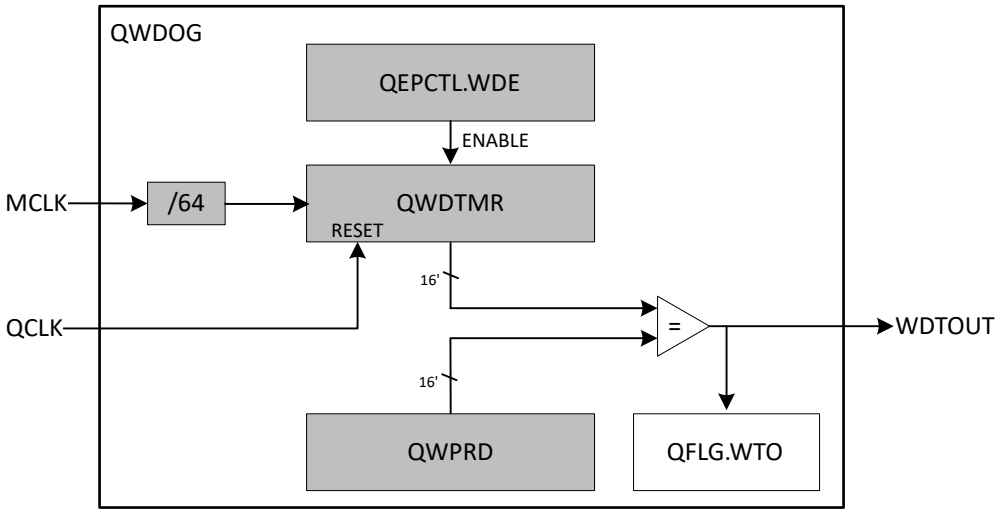


Figure 21-20. eQEP Watchdog Timer

21.8 eQEP Unit Timer Base

The eQEP peripheral includes a 32-bit timer (QUTMR) that is clocked by MCLK to generate periodic interrupts for velocity calculations, see Figure 21-21. Whenever the unit timer (QUTMR) matches the unit period register (QUPRD), the eQEP peripheral resets the unit timer (QUTMR) and also generates the unit time out interrupt flag (QFLG[UTO]). The unit timer gets reset whenever timer value equals to configured period value.

The eQEP peripheral can be configured to latch the position counter, capture timer, and capture period values on a unit time out event so that latched values are used for velocity calculation as described in Section 21.6 .

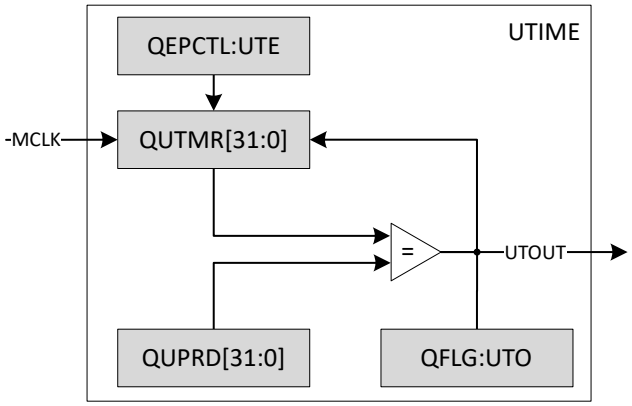


Figure 21-21. eQEP Unit Timer Base

21.9 QMA Module

The QEP Mode Adapter (QMA) is designed to extend the eQEP module capabilities to support the additional modes described. Figure 21-22 depicts how the QMA module is integrated into to the eQEP module.

At reset, by default QMA logic is bypassed and the EQEPA and EQEPB inputs from the pins go directly into the eQEP module. When QMA module is enabled by configuring the QMACTRL[MODE] register, the EQEPA and EQEPB input are processed by this module and modified version of EQEPA and EQEPB signals are sent to the eQEP module. The QMA module requires the eQEP module to be configured in the Direction-Count mode and generates a clock signal on EQEPA input and direction signal on EQEPB input as needed for the proper operation of the intended mode.

- The xCLKMOD block inside the QMA module looks at the transitions on external EQEPA and EQEPB signals to generate the clock signal on the EQEPA input to the eQEP module.
- The xDIRMOD block inside the QMA module looks at the transitions on external EQEPA and EQEPB signals to generate the direction signal on the EQEPB input to the eQEP module.

The QMA module has error detection logic to detect illegal transitions on EQEPA and EQEPB input signals. The QMA module's error and interrupt are integrated inside the eQEP module as described in Section 21.10 .

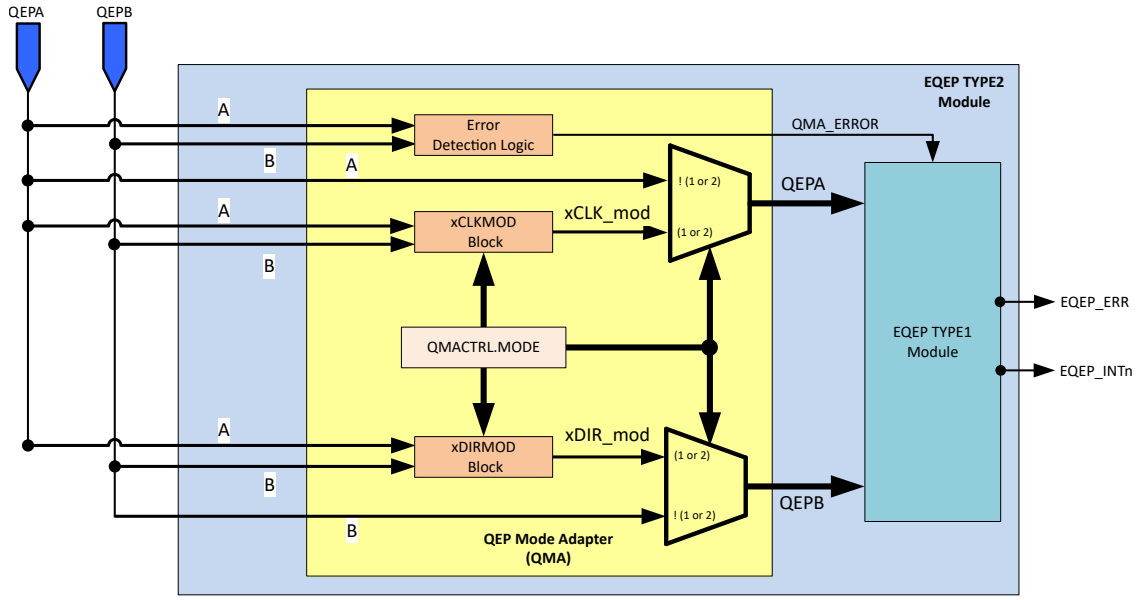


Figure 21-22. QMA Module Block Diagram

ADVANCE INFORMATION

21.9.1 Modes of Operation

The QMA module can be operated in the following modes by configuring the QMACTRL register:

- QMA Mode-1 (QMACTRL[MODE] = 1)
- QMA Mode-2 (QMACTRL[MODE] = 2)

21.9.1.1 QMA Mode-1 (QMACTRL[MODE] = 1)

This mode is used when the default state of EQEPA and EQEPB inputs is high. In this mode, outputs of QMA correspond to the following as shown in Figure 21-23:

- EQEPA Output of QMA is the AND of EQEPA and EQEPB inputs coming from the pin
- EQEPB Output of QMA is the direction signal generated by QMA based on EQEPA and EQEPB inputs

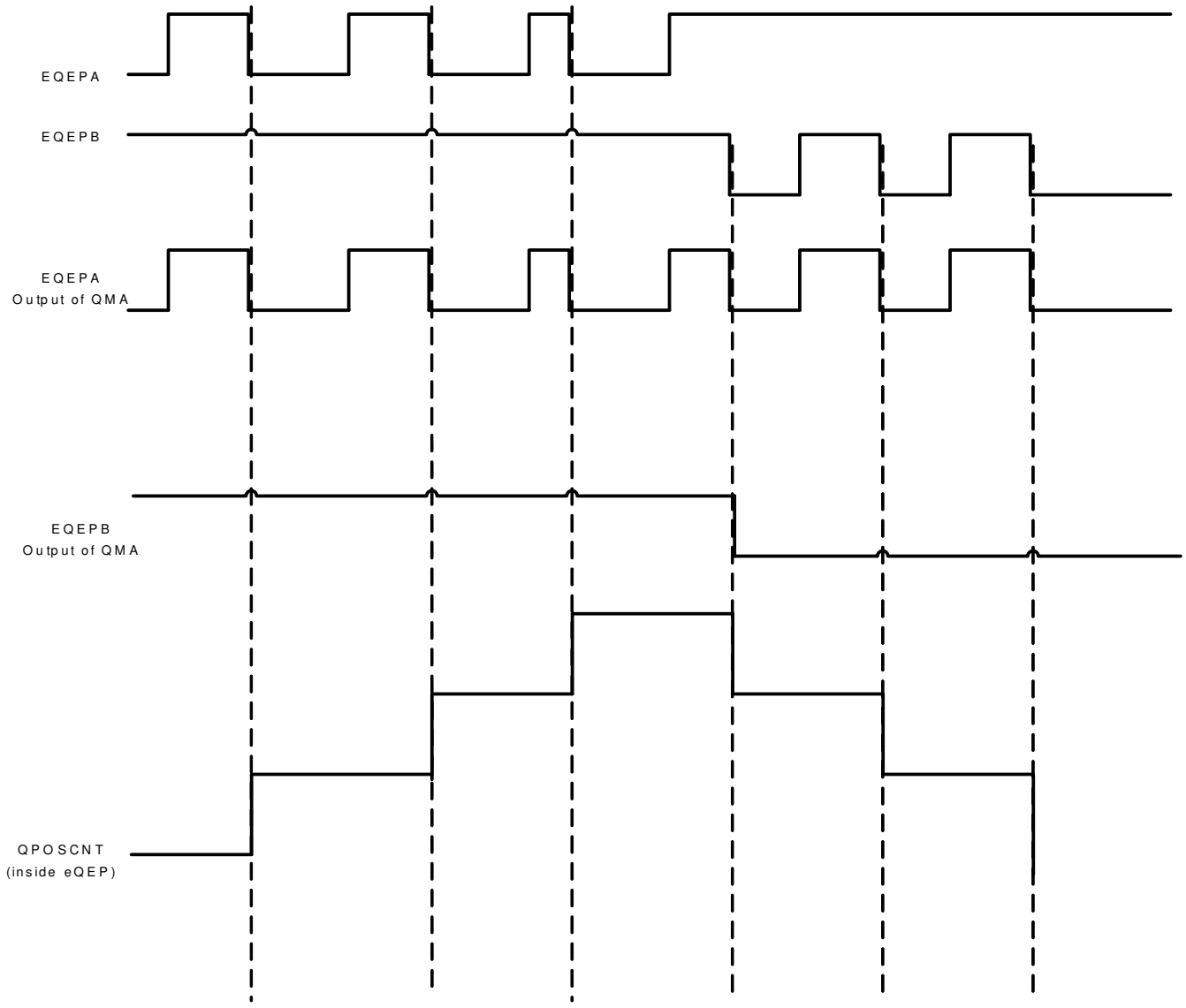


Figure 21-23. QMA Mode-1

21.9.1.2 QMA Mode-2 (QMACTRL[MODE] = 2)

This mode is used when the default state of EQEPA and EQEPB inputs is low. In this mode, outputs of QMA correspond to the following as shown in Figure 21-24:

- EQEPA Output of QMA is the OR of EQEPA and EQEPB inputs coming from the pin
- EQEPB Output of QMA is the direction signal generated by QMA based on EQEPA and EQEPB inputs

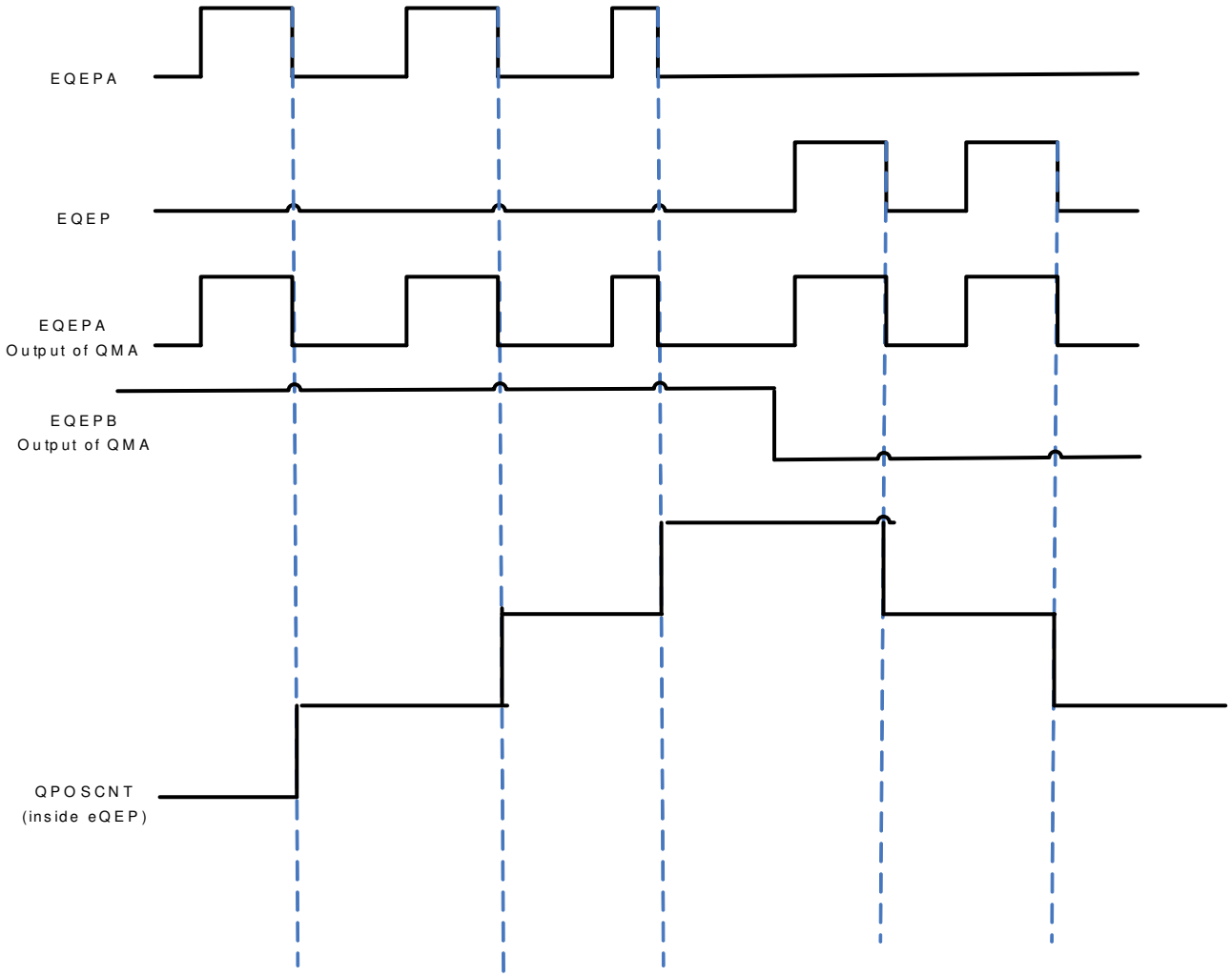


Figure 21-24. QMA Mode-2

21.9.2 Interrupt and Error Generation

The error detection logic detects illegal transitions on EQEPA and EQEPB signals and generates an error signal. This error signal can be used to generate eQEP interrupt and error output. Refer to Section 21.10 for details.

21.10 eQEP Interrupt Structure

Figure 21-25 shows how the interrupt mechanism works in the eQEP module.

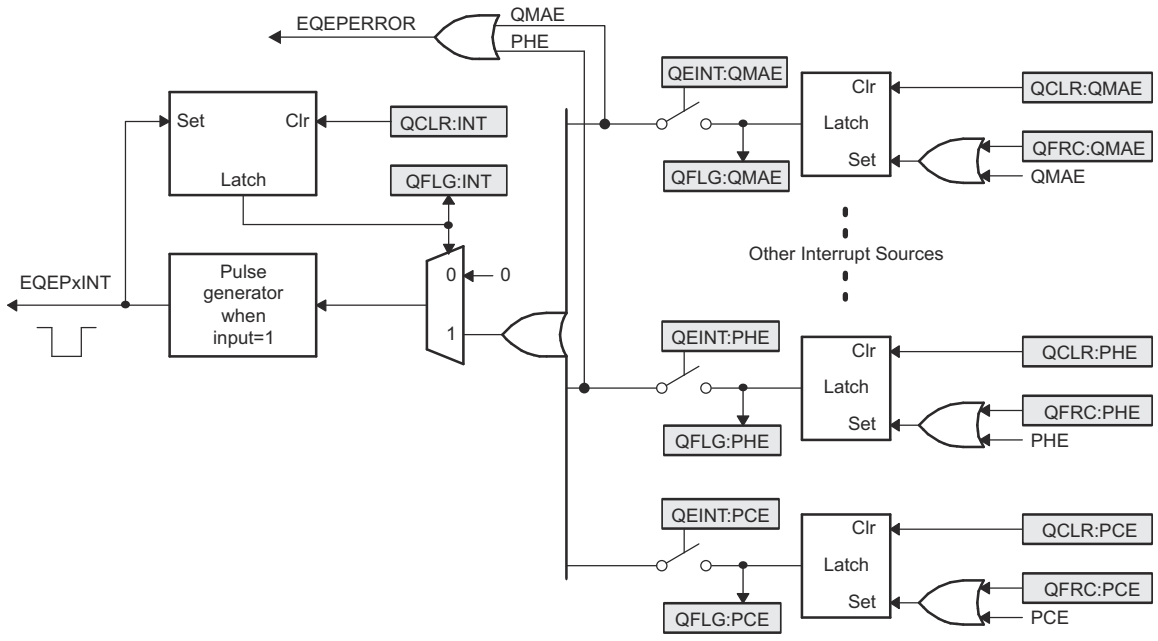


Figure 21-25. eQEP Interrupt Generation

Eleven interrupt events (PCE, PHE, QDC, WTO, PCU, PCO, PCR, PCM, SEL, IEL, and UTO) can be generated. The interrupt control register (QEINT) is used to enable/disable individual interrupt event sources. The interrupt flag register (QFLG) indicates if any interrupt event has been latched and contains the global interrupt flag bit (INT).

A pulse interrupt is generated to the NVIC when the following conditions are met:

1. Interrupt is enabled for eQEP event inside QEINT register
2. Interrupt flag for eQEP event inside QFLG register is set, and
3. Global interrupt status flag bit QFLG[INT] had been cleared for previously generated interrupt event. The interrupt service routine needs to clear the global interrupt flag bit and the serviced event, by way of the interrupt clear register (QCLR), before any other interrupt pulses are generated. If either flags inside the QFLG register are not cleared, further interrupt events do not generate an additional interrupt."
4. You can force an interrupt event by way of the interrupt force register (QFRC), which is useful for test/debug purposes.

21.11 Software

21.11.1 EQEP Examples

NOTE: These examples are located in the device SDK installation at the following location: {SDK_VERSION#}/driverlib/{DEVICE_GPN}/examples/{CORE_IF_MULTICORE}/eqep

21.11.1.1 Frequency Measurement Using eQEP

FILE: eqep_ex1_freq_cal.c

This example calculates the frequency of an input signal using the eQEP module. MCPWM1A is configured to generate this input signal with a frequency of 5kHz. There is an interrupt once every period with a call to the frequency calculation function. This example uses the IQMath library to simplify high-precision calculations.

In addition to the main example file, the following files must be included in this project:
- *eqep_ex1_calculation.c* - contains frequency calculation function
- *eqep_ex1_calculation.h* - includes initialization values for frequency structure

The configuration for this example is as follows:
- Maximum frequency is configured to 10KHz (baseFreq)
- Minimum frequency is assumed at 50Hz for capture pre-scalar selection

SPEED_FR: High Frequency Measurement is obtained by counting the external input pulses for 10ms (unit timer set to 100Hz). \[SPEED_FR = \frac{Count\ Delta}{10ms} \]

SPEED_PR: Low Frequency Measurement is obtained by measuring time period of input edges. Time measurement is averaged over 64 edges for better results and the capture unit performs the time measurement using pre-scaled MCLK.

Note that the pre-scaler for capture unit clock is selected such that the capture timer does not overflow at the required minimum frequency. This example runs indefinitely until the user stops the timer.

For more information about the frequency calculation see the comments at the beginning of eqep_ex1_calculation.c and the XLS file provided with the project, eqep_ex1_calculation.xls.

External Connections
- Connect GPIO6/eQEP1A to GPIO0/MCPWM1A

Watch Variables
- *freq.freqHzFR* - Frequency measurement using position counter/unit time out
- *freq.freqHzPR* - Frequency measurement using capture unit

21.11.1.2 Position and Speed Measurement Using eQEP

FILE: eqep_ex2_pos_speed.c

This example provides position and speed measurement using the capture unit and speed measurement using unit time out of the eQEP module. MCPWM1 and a GPIO are configured to generate simulated eQEP signals. The MCPWM module interrupts once every period and calls the position/speed calculation function. This example uses the IQMath library to simplify high-precision calculations.

In addition to the main example file, the following files must be included in this project:
- *eqep_ex2_calculation.c* - contains position/speed calculation function
- *eqep_ex2_calculation.h* - includes initialization values for position/speed structure

The configuration for this example is as follows:
- Maximum speed is configured to 6000rpm (baseRPM)
- Minimum speed is assumed at 10rpm for capture pre-scalar selection
- Pole pair is configured to 2 (polePairs)
- Encoder resolution is configured to 4000 counts/revolution (mechScaler)
- Which means: 4000 / 4 = 1000 line/revolution quadrature encoder (simulated by MCPWM1)
- MCPWM1 (simulating QEP encoder signals) is configured for a 5kHz frequency or 300 rpm (= 4 * 5000 cnts/sec * 60 sec/min) / 4000 cnts/rev)

SPEEDRPM_FR: High Speed Measurement is obtained by counting the QEP input pulses for 10ms (unit timer set to 100Hz).

SPEEDRPM_FR = (Position Delta / 10ms) * 60 rpm

SPEEDRPM_PR: Low Speed Measurement is obtained by measuring time period of QEP edges. Time measurement is averaged over 64 edges for better results and the capture unit performs the time measurement using pre-scaled MCLK .

Note that the pre-scaler for capture unit clock is selected such that the capture timer does not overflow at the required minimum frequency. This example runs indefinitely until the user ends the program.

For more information about the position/speed calculation see the comments at the beginning of eqep_ex2_calculation.c and the XLS file provided with the project, eqep_ex2_calculation.xls.

External Connections

- Connect GPIO6/eQEP1A to GPIO0/MCPWM1A (simulates eQEP Phase A signal)
- Connect GPIO7/eQEP1B to GPIO1/MCPWM1B (simulates eQEP Phase B signal)
- Connect GPIO43/eQEP1I to GPIO2 (simulates eQEP Index Signal)

Watch Variables

- *posSpeed.speedRPMFR* - Speed meas. in rpm using QEP position counter
- *posSpeed.speedRPMPR* - Speed meas. in rpm using capture unit
- *posSpeed.thetaMech* - Motor mechanical angle (Q15)
- *posSpeed.thetaElec* - Motor electrical angle (Q15)

21.11.1.3 PWM Frequency Measurement using EQEP via XBAR connection

FILE: eqep_ex3_epwm_xbar.c

This example calculates the frequency of a PWM signal using the eQEP module. PWM1A is configured to generate this input signal with a frequency of 5kHz. This PWM signal is connected to input of eQEP using Input XBAR and PWM XBAR. The MCPWM module interrupts once every period and calls the frequency calculation function. This example uses the IQMath library to simplify high-precision calculations.

In addition to the main example file, the following files must be included in this project:

- *eqep_ex1_calculation.c* - contains frequency calculation function
- *eqep_ex1_calculation.h* - includes initialization values for frequency structure

The configuration for this example is as follows

- Maximum frequency is configured to 10KHz (baseFreq)
- Minimum frequency is assumed at 50Hz for capture pre-scalar selection
- GPIO0 is connected to output of INPUT_XBAR1
- INPUT_XBAR1 is connected to output of PWMXBAR at TRIP4
- eQEPA source is configured as PWMXBAR.1 output (TRIP4)

SPEED_FR: High Frequency Measurement is obtained by counting the external input pulses for 10ms (unit timer set to 100Hz). \[SPEED_FR = \frac{Count\ Delta}{10ms} \]

SPEED_PR: Low Frequency Measurement is obtained by measuring time period of input edges. Time measurement is averaged over 64 edges for better results and the capture unit performs the time measurement using pre-scaled MCLK .

Note that the pre-scaler for capture unit clock is selected such that the capture timer does not overflow at the required minimum frequency. This example runs indefinitely until the user stops the program.

For more information about the frequency calculation see the comments at the beginning of eqep_ex1_calculation.c and the XLS file provided with the project, eqep_ex1_calculation.xls.

Watch Variables

- *freq.freqHzFR* - Frequency measurement using position counter/unit time out
- *freq.freqHzPR* - Frequency measurement using capture unit

21.11.1.4 Frequency Measurement Using eQEP via unit timeout interrupt

FILE: eqep_ex4_freq_cal_interrupt.c

This example will calculate the frequency of an input signal using the eQEP module. MCPWM1A is configured to generate this input signal with a frequency of 5kHz. EQEP unit timeout is set which generates an interrupt every *UNIT_PERIOD* microseconds and frequency calculation occurs continuously.

The configuration for this example is as follows:

- PWM frequency is specified as 5000Hz
- UNIT_PERIOD is specified as 10000µs
- Min frequency is (1 / (2*10ms)), that is, 50Hz
- Highest frequency can be (2^32) / ((2*10ms))
- Resolution of frequency measurement is 50Hz

freq : Frequency Measurement is obtained by counting the external input pulses for UNIT_PERIOD (unit timer set to 10ms).

External Connections for Control Card
- Connect GPIO6/eQEP1A to GPIO0/MCPWM1A

Watch Variables
- *freq* - Frequency measurement using position counter/unit time out
- *pass* - If measured frequency matches with PWM frequency then pass = 1 else 0

21.11.1.5 Motor speed and direction measurement using eQEP via unit timeout interrupt

FILE: eqep_ex5_speed_dir_motor.c

This example can be used to sense the speed and direction of motor using eQEP in quadrature encoder mode. MCPWM1A is configured to simulate motor encoder signals with frequency of 5kHz on both A and B pins with 90 degree phase shift (so as to run this example without motor). EQEP unit timeout is set which will generate an interrupt every *UNIT_PERIOD* microseconds and speed calculation occurs continuously based on the direction of motor

The configuration for this example is as follows
- PWM frequency is specified as 5000Hz
- UNIT_PERIOD is specified as 10000µs
- Simulated quadrature signal frequency is 20000Hz (4 * 5000)
- Encoder holes assumed as 1000
- Thus Simulated motor speed is 300rpm (5000 * (60 / 1000))

freq : Simulated quadrature signal frequency measured by counting the external input pulses for UNIT_PERIOD (unit timer set to 10ms). *speed* : Measure motor speed in rpm *dir* : Indicates clockwise (1) or anticlockwise (-1)

External Connections (if motor encoder signals are simulated by MCPWM)
With motor
- Comment in "MOTOR" in includes
- Connect GPIO6/eQEP1A to GPIO0/MCPWM1A (simulates eQEP Phase A signal)
- Connect GPIO7/eQEP1B to GPIO1/MCPWM1B (simulates eQEP Phase B signal)

Watch Variables
- *freq* : Simulated motor frequency measurement is obtained by counting the external input pulses for UNIT_PERIOD (unit timer set to 10ms).
- *speed* : Measure motor speed in rpm
- *dir* : Indicates clockwise (1) or anticlockwise (-1)
- *pass* - If measured quadrature frequency matches with input quadrature frequency (4 * PWM frequency) then pass = 1 else fail = 1 (** only when "MOTOR" is commented out)

21.12 EQEP Registers

This Section describes the EQEP Registers.

21.12.1 EQEP Base Address Table

Table 21-4. EQEP Base Address Table

Bit Field Name		DriverLib Name	Base Address
Instance	Structure		
EQep0Regs	EQEP_REGS	EQEP0	0x4044_8000

Table 21-4. EQEP Base Address Table (continued)

Bit Field Name		DriverLib Name	Base Address
Instance	Structure		
EQep1Regs	EQEP_REGS	EQEP1	0x4044_9000
EQep2Regs	EQEP_REGS	EQEP2	0x4044_A000

21.12.2 EQEP_REGS Registers

Table 21-5 lists the memory-mapped registers for the EQEP_REGS registers. All register offset addresses not listed in Table 21-5 should be considered as reserved locations and the register contents should not be modified.

Table 21-5. EQEP_REGS Registers

Offset	Acronym	Register Name	Write Protection	Section
0h	QPOSCNT	Position Counter		Go
4h	QPOSINIT	Position Counter Init		Go
8h	QPOSMAX	Maximum Position Count		Go
Ch	QPOSCMP	Position Compare		Go
10h	QPOSILAT	Index Position Latch		Go
14h	QPOSSLAT	Strobe Position Latch		Go
18h	QPOSLAT	Position Latch		Go
1Ch	QUTMR	QEP Unit Timer		Go
20h	QUPRD	QEP Unit Period		Go
24h	QWDTMR	QEP Watchdog Timer		Go
26h	QWDPRD	QEP Watchdog Period		Go
28h	QDECCTL	Quadrature Decoder Control		Go
2Ah	QEPCTL	QEP Control		Go
2Ch	QCAPCTL	Qaudrature Capture Control		Go
2Eh	QPOSCTL	Position Compare Control		Go
30h	QEINT	QEP Interrupt Control		Go
32h	QFLG	QEP Interrupt Flag		Go
34h	QCLR	QEP Interrupt Clear		Go
36h	QFRC	QEP Interrupt Force		Go
38h	QEPSTS	QEP Status		Go
3Ah	QCTMR	QEP Capture Timer		Go
3Ch	QCPRD	QEP Capture Period		Go
3Eh	QCTMRLAT	QEP Capture Latch		Go
40h	QCPRDLAT	QEP Capture Period Latch		Go
60h	REV	QEP Revision Number		Go
64h	QEPSTROBESEL	QEP Strobe select register		Go
68h	QMACTRL	QMA Control register		Go
6Ch	QEPSRCSEL	QEP Source Select Register		Go

Complex bit access types are encoded to fit into small table cells. Table 21-6 shows the codes that are used for access types in this section.

Table 21-6. EQEP_REGS Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
R-0	R -0	Read Returns 0s
Write Type		
W	W	Write
W1C	W 1C	Write 1 to clear

Table 21-6. EQEP_REGS Access Type Codes (continued)

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

1 QPOSCNT Register (Offset = 0h) [Reset = 00000000h]

QPOSCNT is shown in Figure 21-26 and described in Table 21-7.

Return to the Summary Table.

Position Counter

Figure 21-26. QPOSCNT Register

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

Table 21-8. QPOSCNT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	QPOSCNT	R/W	0h	Position Counter This 32-bit position counter register counts up/down on every eQEP pulse based on direction input. This counter acts as a position integrator whose count value is proportional to position from a give reference point. This Register acts as a Read ONLY register while counter is counting up/down. Note: It is recommended to only write to the position counter register (QPOSCNT) during initialization, i.e. when the eQEP position counter is disabled (QPEN bit of QEPCTL is zero). Once the position counter is enabled (QPEN bit is one), writing to the eQEP position counter register (QPOSCNT) may cause unexpected results. Reset type: SYSRSn

2 QPOSINIT Register (Offset = 4h) [Reset = 00000000h]

QPOSINIT is shown in Figure 21-27 and described in Table 21-8.

Return to the Summary Table.

Position Counter Init

Figure 21-27. QPOSINIT Register

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

Table 21-10. QPOSINIT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	QPOSINIT	R/W	0h	Position Counter Init This register contains the position value that is used to initialize the position counter based on external strobe or index event. The position counter can be initialized through software. Writes to this register should always be full 32-bit writes. Reset type: SYSRSn

3 QPOSMAX Register (Offset = 8h) [Reset = 00000000h]

QPOSMAX is shown in Figure 21-28 and described in Table 21-9.

Return to the Summary Table.

Maximum Position Count

Figure 21-28. QPOSMAX Register

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

Table 21-12. QPOSMAX Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	QPOSMAX	R/W	0h	Maximum Position Count This register contains the maximum position counter value. Writes to this register should always be full 32-bit writes. Reset type: SYSRSn

ADVANCE INFORMATION

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

QPOSCMP is shown in Figure 21-29 and described in Table 21-10.

Return to the Summary Table.

Position Compare

Figure 21-29. QPOSCMP Register

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

Table 21-14. QPOSCMP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	QPOSCMP	R/W	0h	Position Compare The position-compare value in this register is compared with the position counter (QPOSCNT) to generate sync output and/or interrupt on compare match. Writes to this register should always be full 32-bit writes. Reset type: SYSRSn

5 QPOSILAT Register (Offset = 10h) [Reset = 00000000h]

QPOSILAT is shown in Figure 21-30 and described in Table 21-11.

Return to the Summary Table.

Index Position Latch

Figure 21-30. QPOSILAT Register

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

Table 21-16. QPOSILAT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	QPOSILAT	R	0h	Index Position Latch The position-counter value is latched into this register on an index event as defined by the QEPCTL[IEL] bits. Reset type: SYSRSn

6 QPOSSLAT Register (Offset = 14h) [Reset = 00000000h]

QPOSSLAT is shown in Figure 21-31 and described in Table 21-12.

Return to the Summary Table.

Strobe Position Latch

Figure 21-31. QPOSSLAT Register

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

Table 21-18. QPOSSLAT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	QPOSSLAT	R	0h	Strobe Position Latch The position-counter value is latched into this register on a strobe event as defined by the QEPCTL[SEL] bits. Reset type: SYSRSn

ADVANCE INFORMATION

7 QPOSLAT Register (Offset = 18h) [Reset = 00000000h]

QPOSLAT is shown in Figure 21-32 and described in Table 21-13.

Return to the Summary Table.

Position Latch

Figure 21-32. QPOSLAT Register

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

Table 21-20. QPOSLAT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	QPOSLAT	R	0h	Position Latch The position-counter value is latched into this register on a unit time out event. Reset type: SYSRSn

ADVANCE INFORMATION

8 QUTMR Register (Offset = 1Ch) [Reset = 00000000h]

QUTMR is shown in Figure 21-33 and described in Table 21-14.

Return to the Summary Table.

QEP Unit Timer

Figure 21-33. QUTMR Register

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

Table 21-22. QUTMR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	QUTMR	R/W	0h	QEP Unit Timer This register acts as time base for unit time event generation. When this timer value matches the unit time period value a unit time event is generated. Writes to this register should always be full 32-bit writes. Reset type: SYSRSn

9 QUPRD Register (Offset = 20h) [Reset = 00000000h]

QUPRD is shown in Figure 21-34 and described in Table 21-15.

Return to the Summary Table.

QEP Unit Period

Figure 21-34. QUPRD Register

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

Table 21-24. QUPRD Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	QUPRD	R/W	0h	QEP Unit Period This register contains the period count for the unit timer to generate periodic unit time events. These events latch the eQEP position information at periodic intervals and optionally generate an interrupt. Writes to this register should always be full 32-bit writes. Reset type: SYSRSn

10 QWDTMR Register (Offset = 24h) [Reset = 0000h]

QWDTMR is shown in Figure 21-35 and described in Table 21-16.

Return to the Summary Table.

QEP Watchdog Timer

Figure 21-35. QWDTMR Register

15	14	13	12	11	10	9	8
QWDTMR							
R/W-0h							
7	6	5	4	3	2	1	0
QWDTMR							
R/W-0h							

Table 21-26. QWDTMR Register Field Descriptions

Bit	Field	Type	Reset	Description
15-0	QWDTMR	R/W	0h	QEP Watchdog Timer This register acts as time base for the watchdog to detect motor stalls. When this timer value matches with the watchdog's period value a watchdog timeout interrupt is generated. This register is reset upon edge transition in quadrature-clock indicating the motion. Reset type: SYSRSn

11 QWDPRD Register (Offset = 26h) [Reset = 0000h]

QWDPRD is shown in Figure 21-36 and described in Table 21-17.

Return to the Summary Table.

QEP Watchdog Period

Figure 21-36. QWDPRD Register

15	14	13	12	11	10	9	8
QWDPRD							
R/W-0h							
7	6	5	4	3	2	1	0
QWDPRD							
R/W-0h							

Table 21-28. QWDPRD Register Field Descriptions

Bit	Field	Type	Reset	Description
15-0	QWDPRD	R/W	0h	QEP Watchdog Period This register contains the time-out count for the eQEP peripheral watch dog timer. When the watchdog timer value matches the watchdog period value, a watchdog timeout interrupt is generated. Reset type: SYSRSn

12 QDECCTL Register (Offset = 28h) [Reset = 0000h]

QDECCTL is shown in Figure 21-37 and described in Table 21-18.

Return to the Summary Table.

Quadrature Decoder Control

Figure 21-37. QDECCTL Register

15	14	13	12	11	10	9	8
QSRC		SOEN	SPSEL	XCR	SWAP	IGATE	QAP
R/W-0h		R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
QBP	QIP	QSP	RESERVED				QIDIRE
R/W-0h	R/W-0h	R/W-0h	R-0h				R/W-0h

Table 21-30. QDECCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
15-14	QSRC	R/W	0h	Position-counter source selection Reset type: SYSRSn 0h (R/W) = Quadrature count mode (QCLK = iCLK, QDIR = iDIR) 1h (R/W) = Direction-count mode (QCLK = xCLK, QDIR = xDIR) 2h (R/W) = UP count mode for frequency measurement (QCLK = xCLK, QDIR = 1) 3h (R/W) = DOWN count mode for frequency measurement (QCLK = xCLK, QDIR = 0)
13	SOEN	R/W	0h	Sync output-enable Reset type: SYSRSn 0h (R/W) = Disable position-compare sync output 1h (R/W) = Enable position-compare sync output
12	SPSEL	R/W	0h	Sync output pin selection Reset type: SYSRSn 0h (R/W) = Index pin is used for sync output 1h (R/W) = Strobe pin is used for sync output
11	XCR	R/W	0h	External Clock Rate Reset type: SYSRSn 0h (R/W) = 2x resolution: Count the rising/falling edge 1h (R/W) = 1x resolution: Count the rising edge only
10	SWAP	R/W	0h	CLK/DIR Signal Source for Position Counter Reset type: SYSRSn 0h (R/W) = Quadrature-clock inputs are not swapped 1h (R/W) = Quadrature-clock inputs are swapped
9	IGATE	R/W	0h	Index pulse gating option Reset type: SYSRSn 0h (R/W) = Disable gating of Index pulse 1h (R/W) = Gate the index pin with strobe
8	QAP	R/W	0h	QEPA input polarity Reset type: SYSRSn 0h (R/W) = No effect 1h (R/W) = Negates QEPA input

Table 21-30. QDECCTL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
7	QBP	R/W	0h	QEPB input polarity Reset type: SYSRSn 0h (R/W) = No effect 1h (R/W) = Negates QEPB input
6	QIP	R/W	0h	QEPI input polarity Reset type: SYSRSn 0h (R/W) = No effect 1h (R/W) = Negates QEPI input
5	QSP	R/W	0h	QEPS input polarity Reset type: SYSRSn 0h (R/W) = No effect 1h (R/W) = Negates QEPS input
4-1	RESERVED	R	0h	Reserved
0	QIDIRE	R/W	0h	0 - Compatible mode, Behavior same as existing devices 1 - Enhancement for Direction change during Index will be enabled: On QEPI direction change, the incoming posedge of QA can erroneously update/reset the position counter of the eQEP. This bit only needs to be enabled if the application requires a direction change occurring at the same time as an incoming QEPI signal, or when erroneous PC resets are observed. Reset type: SYSRSn

13 QEPCTL Register (Offset = 2Ah) [Reset = 0000h]

QEPCTL is shown in Figure 21-38 and described in Table 21-19.

Return to the Summary Table.

QEP Control

Figure 21-38. QEPCTL Register

15	14	13	12	11	10	9	8
FREE_SOFT		PCRM		SEI		IEI	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	
7	6	5	4	3	2	1	0
SWI	SEL	IEL		QPEN	QCLM	UTE	WDE
R/W-0h	R/W-0h	R/W-0h		R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 21-32. QEPCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
15-14	FREE_SOFT	R/W	0h	Emulation mode Reset type: SYSRSn 0h (R/W) = QPOSCNT behavior Position counter stops immediately on emulation suspend 0h (R/W) = QWDTMR behavior Watchdog counter stops immediately 0h (R/W) = QUTMR behavior Unit timer stops immediately 0h (R/W) = QCTMR behavior Capture Timer stops immediately 1h (R/W) = QPOSCNT behavior Position counter continues to count until the rollover 1h (R/W) = QWDTMR behavior Watchdog counter counts until WD period match roll over 1h (R/W) = QUTMR behavior Unit timer counts until period rollover 1h (R/W) = QCTMR behavior Capture Timer counts until next unit period event 2h (R/W) = QPOSCNT behavior Position counter is unaffected by emulation suspend 2h (R/W) = QWDTMR behavior Watchdog counter is unaffected by emulation suspend 2h (R/W) = QUTMR behavior Unit timer is unaffected by emulation suspend 2h (R/W) = QCTMR behavior Capture Timer is unaffected by emulation suspend 3h (R/W) = Same as FREE_SOFT_2
13-12	PCRM	R/W	0h	Postion counter reset Reset type: SYSRSn 0h (R/W) = Position counter reset on an index event 1h (R/W) = Position counter reset on the maximum position 2h (R/W) = Position counter reset on the first index event 3h (R/W) = Position counter reset on a unit time event

ADVANCE INFORMATION

Table 21-32. QEPCTL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
11-10	SEI	R/W	0h	Strobe event initialization of position counter Reset type: SYSRSn 0h (R/W) = Does nothing (action disabled) 1h (R/W) = Does nothing (action disabled) 2h (R/W) = Initializes the position counter on rising edge of the QEPS signal 3h (R/W) = Clockwise Direction: Initializes the position counter on the rising edge of QEPS strobe Counter Clockwise Direction: Initializes the position counter on the falling edge of QEPS strobe
9-8	IEI	R/W	0h	Index event init of position count Reset type: SYSRSn 0h (R/W) = Do nothing (action disabled) 1h (R/W) = Do nothing (action disabled) 2h (R/W) = Initializes the position counter on the rising edge of the QEPI signal (QPOSCNT = QPOSINIT) 3h (R/W) = Initializes the position counter on the falling edge of QEPI signal (QPOSCNT = QPOSINIT)
7	SWI	R/W	0h	Software init position counter Reset type: SYSRSn 0h (R/W) = Do nothing (action disabled) 1h (R/W) = Initialize position counter (QPOSCNT=QPOSINIT). This bit is not cleared automatically
6	SEL	R/W	0h	Strobe event latch of position counter Reset type: SYSRSn 0h (R/W) = The position counter is latched on the rising edge of QEPS strobe (QPOSSLAT = POSCCNT). Latching on the falling edge can be done by inverting the strobe input using the QSP bit in the QDECCTL register 1h (R/W) = Clockwise Direction: Position counter is latched on rising edge of QEPS strobe Counter Clockwise Direction: Position counter is latched on falling edge of QEPS strobe
5-4	IEL	R/W	0h	Index event latch of position counter (software index marker) Reset type: SYSRSn 0h (R/W) = Reserved 1h (R/W) = Latches position counter on rising edge of the index signal 2h (R/W) = Latches position counter on falling edge of the index signal 3h (R/W) = Software index marker. Latches the position counter and quadrature direction flag on index event marker. The position counter is latched to the QPOSILAT register and the direction flag is latched in the QEPSTS[QDLF] bit. This mode is useful for software index marking.
3	QPEN	R/W	0h	Quadrature position counter enable/software reset Reset type: SYSRSn 0h (R/W) = Reset the eQEP peripheral internal operating flags/read-only registers. Control/configuration registers are not disturbed by a software reset. When QPEN is disabled, some flags in the QFLG register do not get reset or cleared and show the actual state of that flag. 1h (R/W) = eQEP position counter is enabled

Table 21-32. QEPCTL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
2	QCLM	R/W	0h	QEP capture latch mode Reset type: SYSRSn 0h (R/W) = Latch on position counter read by CPU. Capture timer and capture period values are latched into QCTMRLAT and QCPRDLAT registers when CPU reads the QPOSCNT register. 1h (R/W) = Latch on unit time out. Position counter, capture timer and capture period values are latched into QPOSLAT, QCTMRLAT and QCPRDLAT registers on unit time out.
1	UTE	R/W	0h	QEP unit timer enable Reset type: SYSRSn 0h (R/W) = Disable eQEP unit timer 1h (R/W) = Enable unit timer
0	WDE	R/W	0h	QEP watchdog enable Reset type: SYSRSn 0h (R/W) = Disable the eQEP watchdog timer 1h (R/W) = Enable the eQEP watchdog timer

14 QCAPCTL Register (Offset = 2Ch) [Reset = 0000h]

QCAPCTL is shown in Figure 21-39 and described in Table 21-20.

Return to the Summary Table.

Qaudrature Capture Control

Figure 21-39. QCAPCTL Register

15	14	13	12	11	10	9	8
CEN	RESERVED						
R/W-0h	R-0h						
7	6	5	4	3	2	1	0
RESERVED	CCPS			UPPS			
R-0h	R/W-0h			R/W-0h			

Table 21-34. QCAPCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
15	CEN	R/W	0h	Enable eQEP capture Reset type: SYSRSn 0h (R/W) = eQEP capture unit is disabled 1h (R/W) = eQEP capture unit is enabled
14-7	RESERVED	R	0h	Reserved
6-4	CCPS	R/W	0h	eQEP capture timer clock prescaler Reset type: SYSRSn 0h (R/W) = CAPCLK = SYSCLKOUT/1 1h (R/W) = CAPCLK = SYSCLKOUT/2 2h (R/W) = CAPCLK = SYSCLKOUT/4 3h (R/W) = CAPCLK = SYSCLKOUT/8 4h (R/W) = CAPCLK = SYSCLKOUT/16 5h (R/W) = CAPCLK = SYSCLKOUT/32 6h (R/W) = CAPCLK = SYSCLKOUT/64 7h (R/W) = CAPCLK = SYSCLKOUT/128
3-0	UPPS	R/W	0h	Unit position event prescaler Reset type: SYSRSn 0h (R/W) = UPEVNT = QCLK/1 1h (R/W) = UPEVNT = QCLK/2 2h (R/W) = UPEVNT = QCLK/4 3h (R/W) = UPEVNT = QCLK/8 4h (R/W) = UPEVNT = QCLK/16 5h (R/W) = UPEVNT = QCLK/32 6h (R/W) = UPEVNT = QCLK/64 7h (R/W) = UPEVNT = QCLK/128 8h (R/W) = UPEVNT = QCLK/256 9h (R/W) = UPEVNT = QCLK/512 Ah (R/W) = UPEVNT = QCLK/1024 Bh (R/W) = UPEVNT = QCLK/2048 Ch (R/W) = Reserved Dh (R/W) = Reserved Eh (R/W) = Reserved Fh (R/W) = Reserved

15 QPOSCTL Register (Offset = 2Eh) [Reset = 0000h]

QPOSCTL is shown in Figure 21-40 and described in Table 21-21.

Return to the Summary Table.

Position Compare Control

Figure 21-40. QPOSCTL Register

15	14	13	12	11	10	9	8
PCSHDW	PCLOAD	PCPOL	PCE	PCSPW			
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h			
7	6	5	4	3	2	1	0
PCSPW							
R/W-0h							

Table 21-36. QPOSCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
15	PCSHDW	R/W	0h	Position compare of shadow enable Reset type: SYSRSn 0h (R/W) = Shadow disabled, load Immediate 1h (R/W) = Shadow enabled
14	PCLOAD	R/W	0h	Position compare of shadow load Reset type: SYSRSn 0h (R/W) = Load on QPOSCNT = 0 1h (R/W) = Load when QPOSCNT = QPOSCMP
13	PCPOL	R/W	0h	Polarity of sync output Reset type: SYSRSn 0h (R/W) = Active HIGH pulse output 1h (R/W) = Active LOW pulse output
12	PCE	R/W	0h	Position compare enable/disable Reset type: SYSRSn 0h (R/W) = Disable position compare unit 1h (R/W) = Enable position compare unit
11-0	PCSPW	R/W	0h	Select-position-compare sync output pulse width Reset type: SYSRSn 0h (R/W) = 1 * 4 * SYSCLKOUT cycles 1h (R/W) = 2 * 4 * SYSCLKOUT cycles FFFh (R/W) = 4096 * 4 * SYSCLKOUT cycles

16 QEINT Register (Offset = 30h) [Reset = 0000h]

QEINT is shown in Figure 21-41 and described in Table 21-22.

Return to the Summary Table.

QEP Interrupt Control

Figure 21-41. QEINT Register

15	14	13	12	11	10	9	8
RESERVED			QMAE	UTO	IEL	SEL	PCM
R-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
PCR	PCO	PCU	WTO	QDC	QPE	PCE	RESERVED
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R-0h

Table 21-38. QEINT Register Field Descriptions

Bit	Field	Type	Reset	Description
15-13	RESERVED	R	0h	Reserved
12	QMAE	R/W	0h	QMA Error Interrupt enable Reset type: SYSRSn 0h (R/W) = Interrupt is disabled 1h (R/W) = Interrupt is enabled
11	UTO	R/W	0h	Unit time out interrupt enable Reset type: SYSRSn 0h (R/W) = Interrupt is disabled 1h (R/W) = Interrupt is enabled
10	IEL	R/W	0h	Index event latch interrupt enable Reset type: SYSRSn 0h (R/W) = Interrupt is disabled 1h (R/W) = Interrupt is enabled
9	SEL	R/W	0h	Strobe event latch interrupt enable Reset type: SYSRSn 0h (R/W) = Interrupt is disabled 1h (R/W) = Interrupt is enabled
8	PCM	R/W	0h	Position-compare match interrupt enable Reset type: SYSRSn 0h (R/W) = Interrupt is disabled 1h (R/W) = Interrupt is enabled
7	PCR	R/W	0h	Position-compare ready interrupt enable Reset type: SYSRSn 0h (R/W) = Interrupt is disabled 1h (R/W) = Interrupt is enabled
6	PCO	R/W	0h	Position counter overflow interrupt enable Reset type: SYSRSn 0h (R/W) = Interrupt is disabled 1h (R/W) = Interrupt is enabled

Table 21-38. QEINT Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
5	PCU	R/W	0h	Position counter underflow interrupt enable Reset type: SYSRSn 0h (R/W) = Interrupt is disabled 1h (R/W) = Interrupt is enabled
4	WTO	R/W	0h	Watchdog time out interrupt enable Reset type: SYSRSn 0h (R/W) = Interrupt is disabled 1h (R/W) = Interrupt is enabled
3	QDC	R/W	0h	Quadrature direction change interrupt enable Reset type: SYSRSn 0h (R/W) = Interrupt is disabled 1h (R/W) = Interrupt is enabled
2	QPE	R/W	0h	Quadrature phase error interrupt enable Reset type: SYSRSn 0h (R/W) = Interrupt is disabled 1h (R/W) = Interrupt is enabled
1	PCE	R/W	0h	Position counter error interrupt enable Reset type: SYSRSn 0h (R/W) = Interrupt is disabled 1h (R/W) = Interrupt is enabled
0	RESERVED	R	0h	Reserved

17 QFLG Register (Offset = 32h) [Reset = 0000h]

QFLG is shown in Figure 21-42 and described in Table 21-23.

Return to the Summary Table.

QEP Interrupt Flag

Figure 21-42. QFLG Register

15	14	13	12	11	10	9	8
RESERVED			QMAE	UTO	IEL	SEL	PCM
R-0h			R-0h	R-0h	R-0h	R-0h	R-0h
7	6	5	4	3	2	1	0
PCR	PCO	PCU	WTO	QDC	PHE	PCE	INT
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

Table 21-40. QFLG Register Field Descriptions

Bit	Field	Type	Reset	Description
15-13	RESERVED	R	0h	Reserved
12	QMAE	R	0h	QMA Error interrupt flag Reset type: SYSRSn 0h (R/W) = No interrupt generated 1h (R/W) = Interrupt was generated
11	UTO	R	0h	Unit time out interrupt flag Reset type: SYSRSn 0h (R/W) = No interrupt generated 1h (R/W) = Set by eQEP unit timer period match
10	IEL	R	0h	Index event latch interrupt flag Reset type: SYSRSn 0h (R/W) = No interrupt generated 1h (R/W) = This bit is set after latching the QPOSCNT to QPOSILAT
9	SEL	R	0h	Strobe event latch interrupt flag Reset type: SYSRSn 0h (R/W) = No interrupt generated 1h (R/W) = This bit is set after latching the QPOSCNT to QPOSSLAT
8	PCM	R	0h	eQEP compare match event interrupt flag Reset type: SYSRSn 0h (R/W) = No interrupt generated 1h (R/W) = This bit is set on position-compare match
7	PCR	R	0h	Position-compare ready interrupt flag Reset type: SYSRSn 0h (R/W) = No interrupt generated 1h (R/W) = This bit is set after transferring the shadow register value to the active position compare register
6	PCO	R	0h	Position counter overflow interrupt flag Reset type: SYSRSn 0h (R/W) = No interrupt generated 1h (R/W) = This bit is set on position counter overflow.

Table 21-40. QFLG Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
5	PCU	R	0h	Position counter underflow interrupt flag Reset type: SYSRSn 0h (R/W) = No interrupt generated 1h (R/W) = This bit is set on position counter underflow.
4	WTO	R	0h	Watchdog timeout interrupt flag Reset type: SYSRSn 0h (R/W) = No interrupt generated 1h (R/W) = Set by watchdog timeout
3	QDC	R	0h	Quadrature direction change interrupt flag Reset type: SYSRSn 0h (R/W) = No interrupt generated 1h (R/W) = Interrupt was generated
2	PHE	R	0h	Quadrature phase error interrupt flag Reset type: SYSRSn 0h (R/W) = No interrupt generated 1h (R/W) = Set on simultaneous transition of QEPA and QEPB
1	PCE	R	0h	Position counter error interrupt flag Reset type: SYSRSn 0h (R/W) = No interrupt generated 1h (R/W) = Position counter error
0	INT	R	0h	Global interrupt status flag Reset type: SYSRSn 0h (R/W) = No interrupt generated 1h (R/W) = Interrupt was generated

18 QCLR Register (Offset = 34h) [Reset = 0000h]

QCLR is shown in Figure 21-43 and described in Table 21-24.

Return to the Summary Table.

QEP Interrupt Clear

Figure 21-43. QCLR Register

15	14	13	12	11	10	9	8
RESERVED			QMAE	UTO	IEL	SEL	PCM
R-0h			R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h
7	6	5	4	3	2	1	0
PCR	PCO	PCU	WTO	QDC	PHE	PCE	INT
R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h	R-0/W1S-0h

Table 21-42. QCLR Register Field Descriptions

Bit	Field	Type	Reset	Description
15-13	RESERVED	R	0h	Reserved
12	QMAE	R-0/W1S	0h	Clear QMA Error interrupt flag Reset type: SYSRSn 0h (R/W) = No effect 1h (R/W) = Clears the interrupt flag
11	UTO	R-0/W1S	0h	Clear unit time out interrupt flag Reset type: SYSRSn 0h (R/W) = No effect 1h (R/W) = Clears the interrupt flag
10	IEL	R-0/W1S	0h	Clear index event latch interrupt flag Reset type: SYSRSn 0h (R/W) = No effect 1h (R/W) = Clears the interrupt flag
9	SEL	R-0/W1S	0h	Clear strobe event latch interrupt flag Reset type: SYSRSn 0h (R/W) = No effect 1h (R/W) = Clears the interrupt flag
8	PCM	R-0/W1S	0h	Clear eQEP compare match event interrupt flag Reset type: SYSRSn 0h (R/W) = No effect 1h (R/W) = Clears the interrupt flag
7	PCR	R-0/W1S	0h	Clear position-compare ready interrupt flag Reset type: SYSRSn 0h (R/W) = No effect 1h (R/W) = Clears the interrupt flag
6	PCO	R-0/W1S	0h	Clear position counter overflow interrupt flag Reset type: SYSRSn 0h (R/W) = No effect 1h (R/W) = Clears the interrupt flag

Table 21-42. QCLR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
5	PCU	R-0/W1S	0h	Clear position counter underflow interrupt flag Reset type: SYSRSn 0h (R/W) = No effect 1h (R/W) = Clears the interrupt flag
4	WTO	R-0/W1S	0h	Clear watchdog timeout interrupt flag Reset type: SYSRSn 0h (R/W) = No effect 1h (R/W) = Clears the interrupt flag
3	QDC	R-0/W1S	0h	Clear quadrature direction change interrupt flag Reset type: SYSRSn 0h (R/W) = No effect 1h (R/W) = Clears the interrupt flag
2	PHE	R-0/W1S	0h	Clear quadrature phase error interrupt flag Reset type: SYSRSn 0h (R/W) = No effect 1h (R/W) = Clears the interrupt flag
1	PCE	R-0/W1S	0h	Clear position counter error interrupt flag Reset type: SYSRSn 0h (R/W) = No effect 1h (R/W) = Clears the interrupt flag
0	INT	R-0/W1S	0h	Global interrupt clear flag Reset type: SYSRSn 0h (R/W) = No effect 1h (R/W) = Clears the interrupt flag

19 QFRC Register (Offset = 36h) [Reset = 0000h]

QFRC is shown in Figure 21-44 and described in Table 21-25.

Return to the Summary Table.

QEP Interrupt Force

Figure 21-44. QFRC Register

15	14	13	12	11	10	9	8
RESERVED			QMAE	UTO	IEL	SEL	PCM
R-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
PCR	PCO	PCU	WTO	QDC	PHE	PCE	RESERVED
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R-0h

Table 21-44. QFRC Register Field Descriptions

Bit	Field	Type	Reset	Description
15-13	RESERVED	R	0h	Reserved
12	QMAE	R/W	0h	Force QMA error interrupt Reset type: SYSRSn 0h (R/W) = No effect 1h (R/W) = Force the interrupt
11	UTO	R/W	0h	Force unit time out interrupt Reset type: SYSRSn 0h (R/W) = No effect 1h (R/W) = Force the interrupt
10	IEL	R/W	0h	Force index event latch interrupt Reset type: SYSRSn 0h (R/W) = No effect 1h (R/W) = Force the interrupt
9	SEL	R/W	0h	Force strobe event latch interrupt Reset type: SYSRSn 0h (R/W) = No effect 1h (R/W) = Force the interrupt
8	PCM	R/W	0h	Force position-compare match interrupt Reset type: SYSRSn 0h (R/W) = No effect 1h (R/W) = Force the interrupt
7	PCR	R/W	0h	Force position-compare ready interrupt Reset type: SYSRSn 0h (R/W) = No effect 1h (R/W) = Force the interrupt
6	PCO	R/W	0h	Force position counter overflow interrupt Reset type: SYSRSn 0h (R/W) = No effect 1h (R/W) = Force the interrupt

Table 21-44. QFRC Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
5	PCU	R/W	0h	Force position counter underflow interrupt Reset type: SYSRSn 0h (R/W) = No effect 1h (R/W) = Force the interrupt
4	WTO	R/W	0h	Force watchdog time out interrupt Reset type: SYSRSn 0h (R/W) = No effect 1h (R/W) = Force the interrupt
3	QDC	R/W	0h	Force quadrature direction change interrupt Reset type: SYSRSn 0h (R/W) = No effect 1h (R/W) = Force the interrupt
2	PHE	R/W	0h	Force quadrature phase error interrupt Reset type: SYSRSn 0h (R/W) = No effect 1h (R/W) = Force the interrupt
1	PCE	R/W	0h	Force position counter error interrupt Reset type: SYSRSn 0h (R/W) = No effect 1h (R/W) = Force the interrupt
0	RESERVED	R	0h	Reserved

20 QEPSTS Register (Offset = 38h) [Reset = 0000h]

QEPSTS is shown in Figure 21-45 and described in Table 21-26.

Return to the Summary Table.

QEP Status

Figure 21-45. QEPSTS Register

15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
UPEVNT	FIDF	QDF	QDLF	COEF	CDEF	FIMF	PCEF
R/W1C-0h	R-0h	R-0h	R-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R-0h

Table 21-46. QEPSTS Register Field Descriptions

Bit	Field	Type	Reset	Description
15-8	RESERVED	R	0h	Reserved
7	UPEVNT	R/W1C	0h	Unit position event flag Reset type: SYSRSn 0h (R/W) = No unit position event detected 1h (R/W) = Unit position event detected. Write 1 to clear
6	FIDF	R	0h	Direction on the first index marker Status of the direction is latched on the first index event marker. Reset type: SYSRSn 0h (R/W) = Counter-clockwise rotation (or reverse movement) on the first index event 1h (R/W) = Clockwise rotation (or forward movement) on the first index event
5	QDF	R	0h	Quadrature direction flag Reset type: SYSRSn 0h (R/W) = Counter-clockwise rotation (or reverse movement) 1h (R/W) = Clockwise rotation (or forward movement)
4	QDLF	R	0h	eQEP direction latch flag Reset type: SYSRSn 0h (R/W) = Counter-clockwise rotation (or reverse movement) on index event marker 1h (R/W) = Clockwise rotation (or forward movement) on index event marker
3	COEF	R/W1C	0h	Capture overflow error flag Reset type: SYSRSn 0h (R/W) = Overflow has not occurred. 1h (R/W) = Overflow occurred in eQEP Capture timer (QEPCTMR). This bit is cleared by writing a '1'.
2	CDEF	R/W1C	0h	Capture direction error flag Reset type: SYSRSn 0h (R/W) = Capture direction error has not occurred. 1h (R/W) = Direction change occurred between the capture position event. This bit is cleared by writing a '1'.

Table 21-46. QEPSTS Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
1	FIMF	R/W1C	0h	First index marker flag Reset type: SYSRSn 0h (R/W) = First index pulse has not occurred. 1h (R/W) = Set by first occurrence of index pulse. This bit is cleared by writing a '1'.
0	PCEF	R	0h	Position counter error flag. This bit is not sticky and it is updated for every index event. Reset type: SYSRSn 0h (R/W) = No error occurred during the last index transition 1h (R/W) = Position counter error

21 QCTMR Register (Offset = 3Ah) [Reset = 0000h]

QCTMR is shown in Figure 21-46 and described in Table 21-27.

Return to the Summary Table.

QEP Capture Timer

Figure 21-46. QCTMR Register

15	14	13	12	11	10	9	8
QCTMR							
R/W-0h							
7	6	5	4	3	2	1	0
QCTMR							
R/W-0h							

Table 21-48. QCTMR Register Field Descriptions

Bit	Field	Type	Reset	Description
15-0	QCTMR	R/W	0h	This register provides time base for edge capture unit. Reset type: SYSRSn

22 QCPRD Register (Offset = 3Ch) [Reset = 0000h]

QCPRD is shown in Figure 21-47 and described in Table 21-28.

Return to the Summary Table.

QEP Capture Period

Figure 21-47. QCPRD Register

15	14	13	12	11	10	9	8
QCPRD							
R/W-0h							
7	6	5	4	3	2	1	0
QCPRD							
R/W-0h							

Table 21-50. QCPRD Register Field Descriptions

Bit	Field	Type	Reset	Description
15-0	QCPRD	R/W	0h	This register holds the period count value between the last successive eQEP position events Reset type: SYSRSn

23 QCTMRLAT Register (Offset = 3Eh) [Reset = 0000h]

QCTMRLAT is shown in Figure 21-48 and described in Table 21-29.

Return to the Summary Table.

QEP Capture Latch

Figure 21-48. QCTMRLAT Register

15	14	13	12	11	10	9	8
QCTMRLAT							
R-0h							
7	6	5	4	3	2	1	0
QCTMRLAT							
R-0h							

Table 21-52. QCTMRLAT Register Field Descriptions

Bit	Field	Type	Reset	Description
15-0	QCTMRLAT	R	0h	The eQEP capture timer value can be latched into this register on two events viz., unit timeout event, reading the eQEP position counter. Reset type: SYSRSn

24 QCPRDLAT Register (Offset = 40h) [Reset = 0000h]

QCPRDLAT is shown in Figure 21-49 and described in Table 21-30.

Return to the Summary Table.

QEP Capture Period Latch

Figure 21-49. QCPRDLAT Register

15	14	13	12	11	10	9	8
QCPRDLAT							
R-0h							
7	6	5	4	3	2	1	0
QCPRDLAT							
R-0h							

Table 21-54. QCPRDLAT Register Field Descriptions

Bit	Field	Type	Reset	Description
15-0	QCPRDLAT	R	0h	eQEP capture period value can be latched into this register on two events viz., unit timeout event, reading the eQEP position counter. Reset type: SYSRSn

25 REV Register (Offset = 60h) [Reset = 00000009h]

REV is shown in Figure 21-50 and described in Table 21-31.

Return to the Summary Table.

QEP Revision Number

Figure 21-50. REV Register

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

Table 21-56. REV Register Field Descriptions

Bit	Field	Type	Reset	Description
31-6	RESERVED	R-0	0h	Reserved
5-3	MINOR	R	1h	This field specifies the Minor Revision number for the eQEP IP. Reset type: N/A
2-0	MAJOR	R	1h	This field specifies the Major Revision number for the eQEP IP. Reset type: N/A

26 QEPSTROBESEL Register (Offset = 64h) [Reset = 00000000h]

QEPSTROBESEL is shown in Figure 21-51 and described in Table 21-32.

Return to the Summary Table.

QEP Strobe select register

Figure 21-51. QEPSTROBESEL Register

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

Table 21-58. QEPSTROBESEL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R-0	0h	Reserved
1-0	STROBESEL	R/W	0h	Strobe source select: Reset type: SYSRSn 0h (R/W) = QEP Strobe after polarity mux 1h (R/W) = QEP Strobe after polarity mux 2h (R/W) = QEP Strobe after polarity mux ORed with ADCSOCA 3h (R/W) = QEP Strobe after polarity mux ORed with ADCSOCB

27 QMACTRL Register (Offset = 68h) [Reset = 00000000h]

QMACTRL is shown in Figure 21-52 and described in Table 21-33.

Return to the Summary Table.

QMA Control register

Figure 21-52. QMACTRL Register

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

Table 21-60. QMACTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R-0	0h	Reserved
2-0	MODE	R/W	0h	Select Mode for QMA mode: 000 : QMA Module is bypassed. 001 : QMA Mode-1 operation selected 010 : QMA Mode-2 operation selected 011 : QMA Module is bypassed (reserved) 1xx : QMA Module is bypassed (reserved) Reset type: SYSRSn

ADVANCE INFORMATION

28 QEPSRCSEL Register (Offset = 6Ch) [Reset = 00000000h]

QEPSRCSEL is shown in Figure 21-53 and described in Table 21-34.

Return to the Summary Table.

QEP Source Select Register

Figure 21-53. QEPSRCSEL Register

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

Table 21-62. QEPSRCSEL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R-0	0h	Reserved
15-12	QEPSSEL	R/W	0h	QEP Strobe source select: 0x0: From device Pins (Default). Others: Tied to zero. Reset type: SYSRSn
11-8	QEPISEL	R/W	0h	QEP Index source select: 0x0: Device Pin (Default) 0x1 to 0x1F : Refer to EQEP chaper of TRM Note: eQEP needs to be disabled before configuring these bits as it can lead to unexpected behavior if eQEP is running. Reset type: SYSRSn
7-4	QEPBSEL	R/W	0h	QEPB source select: 0x0: Device Pin (Default) 0x1 to 0x1F : Refer to EQEP chaper of TRM Note: eQEP needs to be disabled before configuring these bits as it can lead to unexpected behavior if eQEP is running. Reset type: SYSRSn
3-0	QEPASEL	R/W	0h	QEPA source select: 0x0: Device Pin (Default) 0x1 to 0x1F : Refer to EQEP chaper of TRM Note: eQEP needs to be disabled before configuring these bits as it can lead to unexpected behavior if eQEP is running. Reset type: SYSRSn

Chapter 22

Crossbar (X-BAR)

The crossbars (referred to as X-BAR throughout this chapter) provide flexibility to connect device inputs, outputs, and internal resources in a variety of configurations.

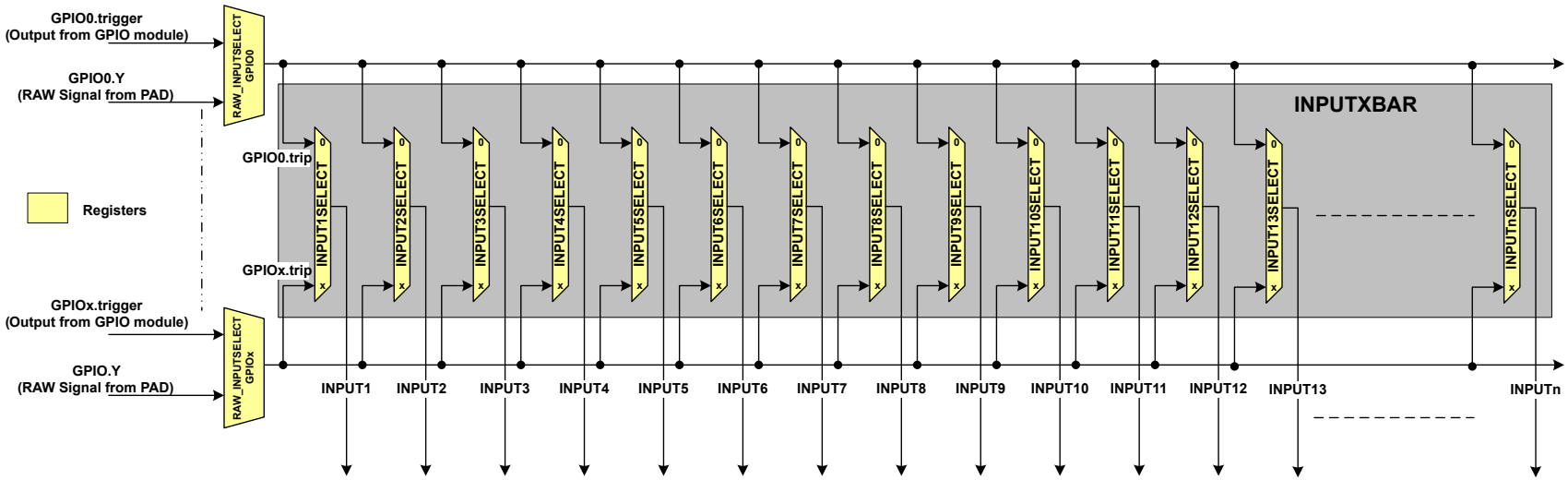
ADVANCE INFORMATION

22.1 INPUTXBAR

On this device, the Input X-BAR is used to route signals from a GPIO to many different IP blocks such as the ADC, eCAP, MCPWM, and external interrupts. The input of each Input X-BAR instance (INPUTx) can be any GPIO, while the output of each instance connects to various IP blocks in the device. This flexibility relieves some of the constraints on peripheral muxing by allowing the user to connect any GPIO to the specified outputs of each Input X-BAR instance. Note that the GPIO selected by the Input X-BAR can be configured as either an input or an output. The Input X-BAR simply connects the signal on the input buffer to the output of the selected Input X-BAR instance. Therefore, you can do things such as route the output of an MCPWM to the eCAP module for a frequency test).

The Input X-BAR is configured by way of the INPUTxSELECT registers. The destinations for each INPUTx are shown in Figure 22-1. For additional details on how each Input X-BAR connects to other IP blocks throughout the device, look for references to Input X-BAR in the chapter associated with that IP. Note that the destinations of each INPUTx are fixed and are not user-configurable. For more information on configuring the Input X-BAR, see the INPUT_XBAR_REGS register definitions in the *XBAR Registers* section.

Figure 22-1. Input X-BAR Architecture



The outputs of the INPUTXBAR are connected to the OUTPUTXBAR, eQEP, and various other peripherals within the device. The diagram below shows this interconnect.

ADVANCE INFORMATION

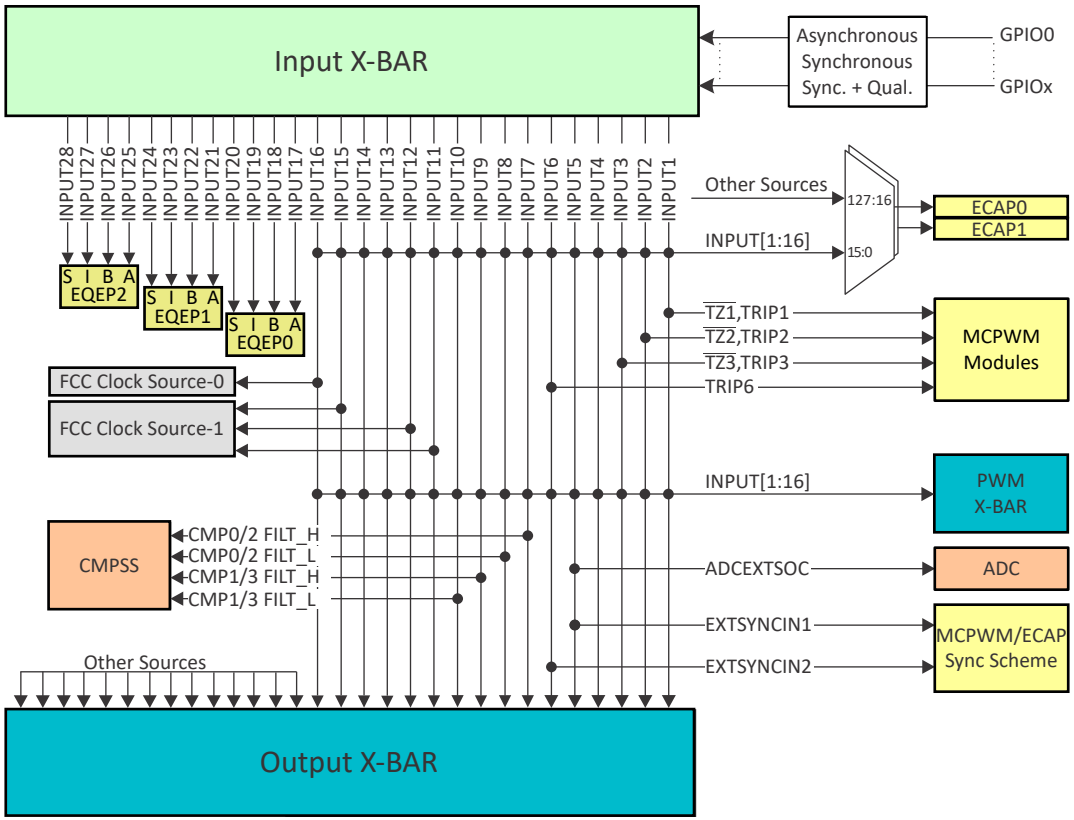


Figure 22-2. INPUTXBAR Connectivity

Note

INPUTXBARx, INPUTXBAR_INPUTx, and INPUTx (when referenced in the context of Input X-BAR) are equivalent in software and documentation.

ADVANCE INFORMATION

Table 22-1. Input X-BAR Destinations

OUTPUT	ECAP	EQEP	PWM XBAR	OUTPUT XBAR	ADC START OF CONVERSION	PWM/ECAP/ADC SYNC	FCC	CMPSS
1	Yes	-	Yes	Yes	-	-	-	-
2	Yes	-	Yes	Yes	-	-	-	-
3	Yes	-	Yes	Yes	-	-	-	-
4	Yes	-	Yes	Yes	-	-	-	-
5	Yes	-	Yes	Yes	ADCEXTSOC	EXTSYNCIN1	-	-
6	Yes	-	Yes	Yes	-	EXTSYNCIN2	-	-
7	Yes	-	Yes	Yes	-	-	-	CMPSS0_EXT_FILTIN_H[1] / CMPSS2_EXT_FILTIN_H[1]
8	Yes	-	Yes	Yes	-	-	-	CMPSS0_EXT_FILTIN_L[1] / CMPSS2_EXT_FILTIN_L[1]
9	Yes	-	Yes	Yes	-	-	-	CMPSS1_EXT_FILTIN_H[1] / CMPSS3_EXT_FILTIN_H[1]
10	Yes	-	Yes	Yes	-	-	-	CMPSS1_EXT_FILTIN_L[1] / CMPSS3_EXT_FILTIN_L[1]
11	Yes	-	Yes	Yes	-	-	CLK1	-
12	Yes	-	Yes	Yes	-	-	CLK1	-
13	Yes	-	Yes	Yes	-	-	-	-
14	Yes	-	Yes	Yes	-	-	-	-
15	Yes	-	Yes	Yes	-	-	CLK1	-
16	Yes	-	Yes	Yes	-	-	CLK0	-
17	-	EQEP0A	-	-	-	-	-	-
18	-	EQEP0B	-	-	-	-	-	-
19	-	EQEP0I	-	-	-	-	-	-
20	-	EQEP0S	-	-	-	-	-	-
21	-	EQEP1A	-	-	-	-	-	-
22	-	EQEP1B	-	-	-	-	-	-
23	-	EQEP1I	-	-	-	-	-	-

Table 22-1. Input X-BAR Destinations (continued)

OUTPUT	ECAP	EQEP	PWM XBAR	OUTPUT XBAR	ADC START OF CONVERSION	PWM/ECAP/ADC SYNC	FCC	CMPSS
24	-	EQEP1S	-	-	-	-	-	-
25	-	EQEP2A	-	-	-	-	-	-
26	-	EQEP2B	-	-	-	-	-	-
27	-	EQEP2I	-	-	-	-	-	-
28	-	EQEP2S	-	-	-	-	-	-
29	-	-	-	-	-	-	-	-
30	-	-	-	-	-	-	-	-
31	-	-	-	-	-	-	-	-
32	-	-	-	-	-	-	-	-

22.2 MCPWM and GPIO Output X-BAR

This section describes the MCPWM and GPIO Output X-BAR. .

22.2.1 MCPWM X-BAR

The MCPWM X-BAR brings signals to the MCPWM modules. Specifically, the MCPWM X-BAR outputs are connected to the TZ inputs of the MCPWM modules. Refer to the *Multi-Channel Pulse Width Modulator (MCPWM)* chapter for more information. Figure 22-3 shows the architecture of the MCPWM X-BAR. Note that the architecture of the MCPWM X-BAR is identical to the architecture of the GPIO Output X-BAR (with the exception of the output latch).

22.2.1.1 MCPWM X-BAR Architecture

The MCPWM X-BAR has eight outputs that are routed to each MCPWM module. Figure 22-3 represents the architecture of a single output, but this output is identical to the architecture of all of the other outputs.

First, determine the signals that can be passed to the MCPWM by referencing Table 22-2. Select up to one signal for each TRIPx output. Select the inputs to MCPWM X-BAR using the TRIPxMUX0TO15CFG and TRIPxMUX16TO31CFG registers. To pass any signal through to the MCPWM, enable the signal using the TRIPxMUXENABLE register. All signals that are enabled are logically ORed before being passed on to the respective TRIPx signal on the MCPWM. To optionally invert the signal, use the TRIPOUTINV register.

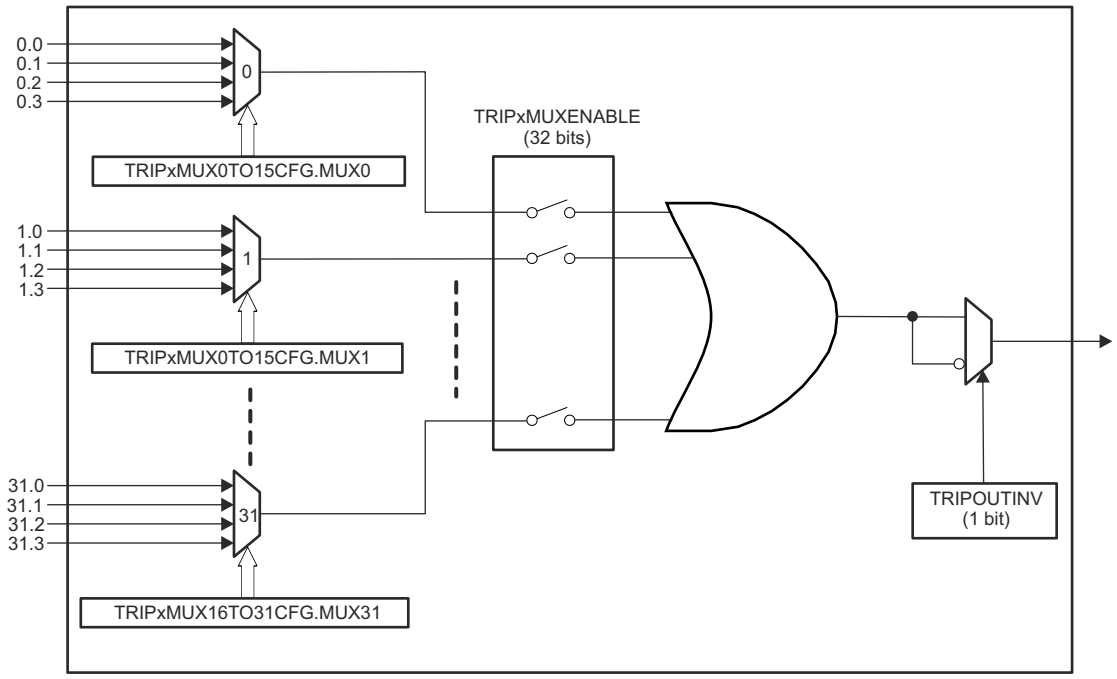


Figure 22-3. MCPWM X-BAR Architecture - Single Output

ADVANCE INFORMATION

Note

Do not use "Reserved" signals in your application.

Table 22-3. MCPWM X-BAR Mux Configuration Table

Group	Bit	Input Signal
G0	0	CMPSS0_CTRIPH
G0	1	CMPSS0_CTRIPL
G0	2	CMPSS1_CTRIPH
G0	3	CMPSS1_CTRIPL
G0	4	CMPSS2_CTRIPH
G0	5	CMPSS2_CTRIPL
G0	6	CMPSS3_CTRIPH
G0	7	CMPSS3_CTRIPL
G0	8	ADC0EVT1
G0	9	ADC0EVT2
G0	10	ADC0EVT3
G0	11	ADC0EVT4
G0	12	ADC1EVT1
G0	13	ADC1EVT2
G0	14	ADC1EVT3
G0	15	ADC1EVT4
G0	16	ADC2EVT1
G0	17	ADC2EVT2
G0	18	ADC2EVT3
G0	19	ADC2EVT4
G0	20	INPUTXBAR1
G0	21	INPUTXBAR2
G0	22	INPUTXBAR3
G0	23	INPUTXBAR4
G0	24	INPUTXBAR5
G0	25	INPUTXBAR6
G0	26	INPUTXBAR7
G0	27	INPUTXBAR8
G0	28	INPUTXBAR9
G0	29	INPUTXBAR10
G0	30	INPUTXBAR11
G0	31	INPUTXBAR12
G1	0	INPUTXBAR13
G1	1	INPUTXBAR14
G1	2	INPUTXBAR15
G1	3	INPUTXBAR16
G1	4	ECAP0_OUT
G1	5	ECAP1_OUT
G1	6	ADCSOCA
G1	7	ADCSOCB
G1	8	EXTSYNCOUT

ADVANCE INFORMATION

Table 22-3. MCPWM X-BAR Mux Configuration Table (continued)

Group	Bit	Input Signal
G1	9	MCAN_INT0
G1	10	Reserved (Tie Low)
G1	11	Reserved (Tie Low)
G1	12	EQEP0_ERR
G1	13	EQEP1_ERR
G1	14	EQEP2_ERR
G1	15	CPU_HALTED
G1	16	Reserved (Tie Low)
G1	17	Reserved (Tie Low)
G1	18	Reserved (Tie Low)
G1	19	Reserved (Tie Low)
G1	20	Reserved (Tie Low)
G1	21	Reserved (Tie Low)
G1	22	Reserved (Tie Low)
G1	23	Reserved (Tie Low)
G1	24	Reserved (Tie Low)
G1	25	Reserved (Tie Low)
G1	26	Reserved (Tie Low)
G1	27	Reserved (Tie Low)
G1	28	Reserved (Tie Low)
G1	29	Reserved (Tie Low)
G1	30	Reserved (Tie Low)
G1	31	Reserved (Tie Low)

22.2.2 GPIO Output X-BAR

The GPIO Output X-BAR takes signals from inside the device and brings them out to a GPIO . Figure 22-4 shows the architecture of the GPIO Output X-BAR. The X-BAR contains eight outputs and each contains at least one position on the GPIO mux, denoted as OUTPUTXBARx. The X-BAR allows the selection of a single input or a logical-OR of many inputs.

Note

Refer to the device data sheet for the exact number of GPIO signals available for a given package.

22.2.2.1 GPIO Output X-BAR Architecture

The GPIO Output X-BAR has eight outputs that are routed to the GPIO module. Figure 22-4 represents the architecture of a single output, but this output is identical to the architecture of all of the other outputs. Note that the architecture of the Output X-BAR (with the exception of the output latch) is similar to the architecture of the MCPWM X-BAR.

First, determine the signals that can be passed to the GPIO by referencing Table 22-4. Select up to one signal per mux for each OUTPUTXBARx output. Select the inputs to each mux using the OUTPUTxMUX0TO15CFG and OUTPUTxMUX16TO31CFG registers. To pass any signal through to the GPIO , enable the mux in the OUTPUTxMUXENABLE register. All muxes that are enabled are logically ORed before being passed on to the respective OUTPUTx signal on the GPIO module. To optionally invert the signal, use the OUTPUTINV register. The final output is only recognized on the GPIO , if the proper OUTPUTx muxing options are selected using the IOMUX registers.

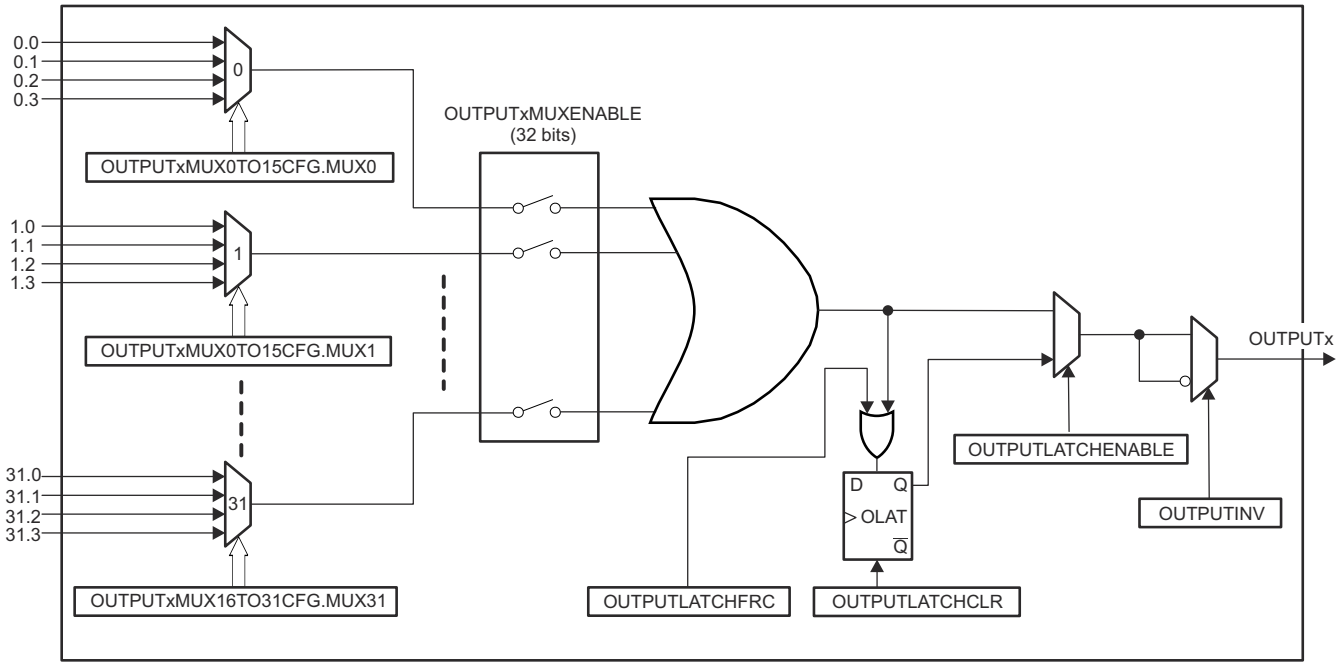


Figure 22-4. GPIO Output X-BAR Architecture

Note

Do not use "Reserved" signals in your application.

Table 22-4. Output X-BAR Mux Configuration Table

Group	Bit	Input Signal
G0	0	CMPSS0_CTRIPOUTH

ADVANCE INFORMATION

Table 22-4. Output X-BAR Mux Configuration Table (continued)

Group	Bit	Input Signal
G0	1	CMPSS0_CTRIPOUTL
G0	2	CMPSS1_CTRIPOUTH
G0	3	CMPSS1_CTRIPOUTL
G0	4	CMPSS2_CTRIPOUTH
G0	5	CMPSS2_CTRIPOUTL
G0	6	CMPSS3_CTRIPOUTH
G0	7	CMPSS3_CTRIPOUTL
G0	8	ADC0EVT1
G0	9	ADC0EVT2
G0	10	ADC0EVT3
G0	11	ADC0EVT4
G0	12	ADC1EVT1
G0	13	ADC1EVT2
G0	14	ADC1EVT3
G0	15	ADC1EVT4
G0	16	ADC2EVT1
G0	17	ADC2EVT2
G0	18	ADC2EVT3
G0	19	ADC2EVT4
G0	20	INPUTXBAR1
G0	21	INPUTXBAR2
G0	22	INPUTXBAR3
G0	23	INPUTXBAR4
G0	24	INPUTXBAR5
G0	25	INPUTXBAR6
G0	26	INPUTXBAR7
G0	27	INPUTXBAR8
G0	28	INPUTXBAR9
G0	29	INPUTXBAR10
G0	30	INPUTXBAR11
G0	31	INPUTXBAR12
G1	0	INPUTXBAR13
G1	1	INPUTXBAR14
G1	2	INPUTXBAR15
G1	3	INPUTXBAR16
G1	4	ECAP0_OUT
G1	5	ECAP1_OUT
G1	6	ADCSOCA

Table 22-4. Output X-BAR Mux Configuration Table (continued)

Group	Bit	Input Signal
G1	7	ADCSOCB
G1	8	EXTSYNCOUT
G1	9	MCAN_INT0
G1	10	Reserved (Tie Low)
G1	11	Reserved (Tie Low)
G1	12	EQEP0_ERR
G1	13	EQEP1_ERR
G1	14	EQEP2_ERR
G1	15	EQEP0_INDEX_SYNCOUT
G1	16	EQEP0_STROBE_SYNCOUT
G1	17	EQEP1_INDEX_SYNCOUT
G1	18	EQEP1_STROBE_SYNCOUT
G1	19	EQEP2_INDEX_SYNCOUT
G1	20	EQEP2_STROBE_SYNCOUT
G1	21	XCLKOUT
G1	22	Reserved (Tie Low)
G1	23	Reserved (Tie Low)
G1	24	Reserved (Tie Low)
G1	25	Reserved (Tie Low)
G1	26	Reserved (Tie Low)
G1	27	Reserved (Tie Low)
G1	28	Reserved (Tie Low)
G1	29	Reserved (Tie Low)
G1	30	Reserved (Tie Low)
G1	31	Reserved (Tie Low)

22.2.3 X-BAR Flags

With the exception of the CMPSS signals, the MCPWM X-BAR and the Output X-BAR have all of the same input signals. Due to the inputs being similar between the MCPWM X-BAR and Output X-BAR, all X-BAR modules leverage a single set of input flags to indicate which input signals have been triggered. This allows software to check the input flags when an event occurs. See Figure 22-5 for more information. There is a bit allocated for each input signal in one of the XBARFLGx registers. The flag remains set until cleared through the appropriate XBARCLRx register.

Note

Not all input sources are routed to all X-BAR modules. Refer to the X-BAR specific configuration tables for exact connections.

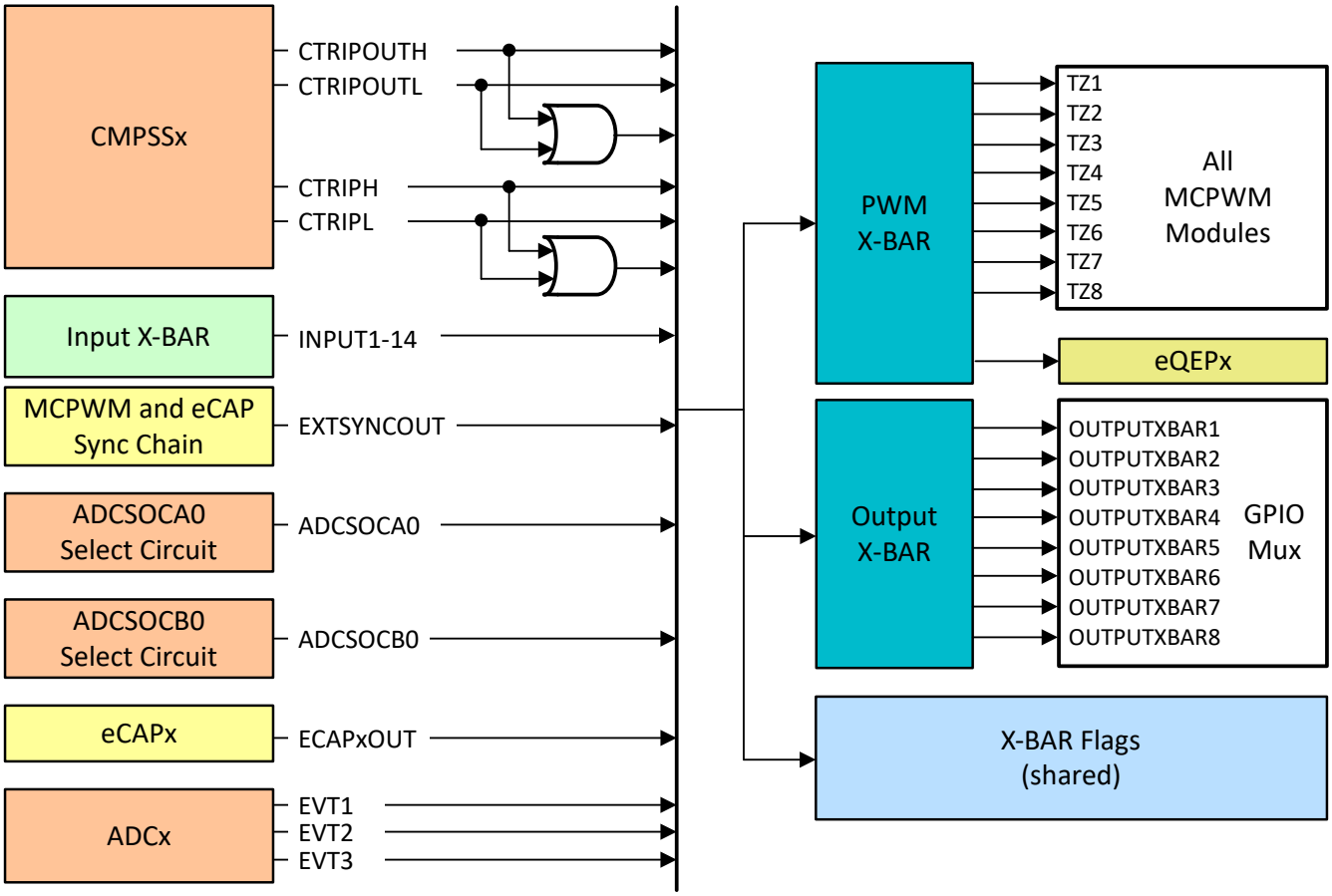


Figure 22-5. X-BAR Input Sources

22.3 XBAR Registers

This Section describes the XBAR Registers.

22.3.1 XBAR Base Address Table

Table 22-5. XBAR Base Address Table

Bit Field Name		DriverLib Name	Base Address
Instance	Structure		
InputXbarRegs	INPUT_XBAR_REGS	INPUTXBAR	0x4046_8000
PwmXbarRegs	EPWM_XBAR_REGS	PWMXBAR	0x4046_9000
OutputXbarRegs	OUTPUTXBAR_REGS	OUTPUTXBAR	0x4046_A000

Table 22-5. XBAR Base Address Table (continued)

Bit Field Name		DriverLib Name	Base Address
Instance	Structure		
SyncRegs	SYNC_SOC_REGS	SYNC	0x4046_B000
Outputxbar0flagsRegs	OUTPUTXBAR_FLAG_REGS	OUTPUTXBAR0_FLAGS	0x4047_0000
Outputxbar1flagsRegs	OUTPUTXBAR_FLAG_REGS	OUTPUTXBAR1_FLAGS	0x4047_1000
Outputxbar2flagsRegs	OUTPUTXBAR_FLAG_REGS	OUTPUTXBAR2_FLAGS	0x4047_2000
Outputxbar3flagsRegs	OUTPUTXBAR_FLAG_REGS	OUTPUTXBAR3_FLAGS	0x4047_3000
Outputxbar4flagsRegs	OUTPUTXBAR_FLAG_REGS	OUTPUTXBAR4_FLAGS	0x4047_4000
Outputxbar5flagsRegs	OUTPUTXBAR_FLAG_REGS	OUTPUTXBAR5_FLAGS	0x4047_5000
Outputxbar6flagsRegs	OUTPUTXBAR_FLAG_REGS	OUTPUTXBAR6_FLAGS	0x4047_6000
Outputxbar7flagsRegs	OUTPUTXBAR_FLAG_REGS	OUTPUTXBAR7_FLAGS	0x4047_7000
InputxbarflagsRegs	INPUT_FLAG_XBAR_REGS	INPUTXBAR_FLAGS	0x4049_0000

22.3.2 INPUT_XBAR_REGS Registers

Table 22-6 lists the memory-mapped registers for the INPUT_XBAR_REGS registers. All register offset addresses not listed in Table 22-6 should be considered as reserved locations and the register contents should not be modified.

Table 22-6. INPUT_XBAR_REGS Registers

Offset	Acronym	Register Name	Write Protection	Section
0h	INPUTSELECT	Input Select Register (GPIO0 to x)		Go
400h	INPUTSELECTLOCK1	Input Select Lock Register 1		Go

Complex bit access types are encoded to fit into small table cells. Table 22-7 shows the codes that are used for access types in this section.

Table 22-7. INPUT_XBAR_REGS Access Type Codes

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

1 INPUTSELECT Register (Offset = 0h) [Reset = 000000FEh]

INPUTSELECT is shown in Figure 22-6 and described in Table 22-8.

Return to the Summary Table.

Input Select Register (GPIO0 to x)

Figure 22-6. INPUTSELECT Register

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

Table 22-9. INPUTSELECT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R-0	0h	Reserved
7-0	SELECT	R/W	FEh	Select GPIO for INPUT{#} signal: 0x0 : Select GPIO0 0x1 : Select GPIO1 0x2 : Select GPIO2 ... 0xFD: '1' will be driven to the destination 0xFE: '1' will be driven to the destination 0xFF: '0' will be driven to the destination NOTE: SELECT value greater than the available number of GPIO pins on a device (except 0xFF) will cause the destination to be driven '1'. Reset type: XRSn

2 INPUTSELECTLOCK1 Register (Offset = 400h) [Reset = 00000000h]

INPUTSELECTLOCK1 is shown in Figure 22-7 and described in Table 22-9.

Return to the Summary Table.

Input Select Lock Register.
Any bit in this register, once set can only be cleared through SYSRSn. Write of 0 to any bit of this register has no effect. Reads to the registers which have LOCK protection are always allowed.

Figure 22-7. INPUTSELECTLOCK1 Register

31	30	29	28	27	26	25	24
INPUT32SELECT	INPUT31SELECT	INPUT30SELECT	INPUT29SELECT	INPUT28SELECT	INPUT27SELECT	INPUT26SELECT	INPUT25SELECT
R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h
23	22	21	20	19	18	17	16
INPUT24SELECT	INPUT23SELECT	INPUT22SELECT	INPUT21SELECT	INPUT20SELECT	INPUT19SELECT	INPUT18SELECT	INPUT17SELECT
R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h
15	14	13	12	11	10	9	8
INPUT16SELECT	INPUT15SELECT	INPUT14SELECT	INPUT13SELECT	INPUT12SELECT	INPUT11SELECT	INPUT10SELECT	INPUT9SELECT
R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h
7	6	5	4	3	2	1	0
INPUT8SELECT	INPUT7SELECT	INPUT6SELECT	INPUT5SELECT	INPUT4SELECT	INPUT3SELECT	INPUT2SELECT	INPUT1SELECT
R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h	R/WSonce-0h

Table 22-11. INPUTSELECTLOCK1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	INPUT32SELECT	R/WSonce	0h	Lock bit for INPUT32SELECT Register 0: Register is not locked 1: Register is locked Reset type: XRSn
30	INPUT31SELECT	R/WSonce	0h	Lock bit for INPUT31SELECT Register 0: Register is not locked 1: Register is locked Reset type: XRSn
29	INPUT30SELECT	R/WSonce	0h	Lock bit for INPUT30SELECT Register 0: Register is not locked 1: Register is locked Reset type: XRSn
28	INPUT29SELECT	R/WSonce	0h	Lock bit for INPUT29SELECT Register 0: Register is not locked 1: Register is locked Reset type: XRSn
27	INPUT28SELECT	R/WSonce	0h	Lock bit for INPUT28SELECT Register 0: Register is not locked 1: Register is locked Reset type: XRSn

Table 22-11. INPUTSELECTLOCK1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
26	INPUT27SELECT	R/WSonce	0h	Lock bit for INPUT27SELECT Register 0: Register is not locked 1: Register is locked Reset type: XRSn
25	INPUT26SELECT	R/WSonce	0h	Lock bit for INPUT26SELECT Register 0: Register is not locked 1: Register is locked Reset type: XRSn
24	INPUT25SELECT	R/WSonce	0h	Lock bit for INPUT25SELECT Register 0: Register is not locked 1: Register is locked Reset type: XRSn
23	INPUT24SELECT	R/WSonce	0h	Lock bit for INPUT24SELECT Register 0: Register is not locked 1: Register is locked Reset type: XRSn
22	INPUT23SELECT	R/WSonce	0h	Lock bit for INPUT23SELECT Register 0: Register is not locked 1: Register is locked Reset type: XRSn
21	INPUT22SELECT	R/WSonce	0h	Lock bit for INPUT22SELECT Register 0: Register is not locked 1: Register is locked Reset type: XRSn
20	INPUT21SELECT	R/WSonce	0h	Lock bit for INPUT21SELECT Register 0: Register is not locked 1: Register is locked Reset type: XRSn
19	INPUT20SELECT	R/WSonce	0h	Lock bit for INPUT20SELECT Register 0: Register is not locked 1: Register is locked Reset type: XRSn
18	INPUT19SELECT	R/WSonce	0h	Lock bit for INPUT19SELECT Register 0: Register is not locked 1: Register is locked Reset type: XRSn
17	INPUT18SELECT	R/WSonce	0h	Lock bit for INPUT18SELECT Register 0: Register is not locked 1: Register is locked Reset type: XRSn
16	INPUT17SELECT	R/WSonce	0h	Lock bit for INPUT17SELECT Register 0: Register is not locked 1: Register is locked Reset type: XRSn

Table 22-11. INPUTSELECTLOCK1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
15	INPUT16SELECT	R/WSonce	0h	Lock bit for INPUT16SELECT Register 0: Register is not locked 1: Register is locked Reset type: XRSn
14	INPUT15SELECT	R/WSonce	0h	Lock bit for INPUT15SELECT Register 0: Register is not locked 1: Register is locked Reset type: XRSn
13	INPUT14SELECT	R/WSonce	0h	Lock bit for INPUT14SELECT Register 0: Register is not locked 1: Register is locked Reset type: XRSn
12	INPUT13SELECT	R/WSonce	0h	Lock bit for INPUT13SELECT Register 0: Register is not locked 1: Register is locked Reset type: XRSn
11	INPUT12SELECT	R/WSonce	0h	Lock bit for INPUT12SELECT Register 0: Register is not locked 1: Register is locked Reset type: XRSn
10	INPUT11SELECT	R/WSonce	0h	Lock bit for INPUT11SELECT Register 0: Register is not locked 1: Register is locked Reset type: XRSn
9	INPUT10SELECT	R/WSonce	0h	Lock bit for INPUT10SELECT Register 0: Register is not locked 1: Register is locked Reset type: XRSn
8	INPUT9SELECT	R/WSonce	0h	Lock bit for INPUT9SELECT Register 0: Register is not locked 1: Register is locked Reset type: XRSn
7	INPUT8SELECT	R/WSonce	0h	Lock bit for INPUT8SELECT Register 0: Register is not locked 1: Register is locked Reset type: XRSn
6	INPUT7SELECT	R/WSonce	0h	Lock bit for INPUT7SELECT Register 0: Register is not locked 1: Register is locked Reset type: XRSn
5	INPUT6SELECT	R/WSonce	0h	Lock bit for INPUT6SELECT Register 0: Register is not locked 1: Register is locked Reset type: XRSn

Table 22-11. INPUTSELECTLOCK1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
4	INPUT5SELECT	R/WSonce	0h	Lock bit for INPUT5SELECT Register 0: Register is not locked 1: Register is locked Reset type: XRSn
3	INPUT4SELECT	R/WSonce	0h	Lock bit for INPUT4SELECT Register 0: Register is not locked 1: Register is locked Reset type: XRSn
2	INPUT3SELECT	R/WSonce	0h	Lock bit for INPUT3SELECT Register 0: Register is not locked 1: Register is locked Reset type: XRSn
1	INPUT2SELECT	R/WSonce	0h	Lock bit for INPUT2SELECT Register 0: Register is not locked 1: Register is locked Reset type: XRSn
0	INPUT1SELECT	R/WSonce	0h	Lock bit for INPUT1SELECT Register 0: Register is not locked 1: Register is locked Reset type: XRSn

22.3.3 EPWM_XBAR_REGS Registers

Table 22-10 lists the memory-mapped registers for the EPWM_XBAR_REGS registers. All register offset addresses not listed in Table 22-10 should be considered as reserved locations and the register contents should not be modified.

Table 22-12. EPWM_XBAR_REGS Registers

Offset	Acronym	Register Name	Write Protection	Section
30h	PWMXBAROutInvert	Output Signal Invert Select		Go
80h	PWMXBARLock	Configuration Lock register		Go
100h + formula	PWMXBARG0SEL_j	PWMXBAR G0 Input Select		Go
104h + formula	PWMXBARG1SEL_j	PWMXBAR G1 Input Select		Go

Complex bit access types are encoded to fit into small table cells. Table 22-11 shows the codes that are used for access types in this section.

Table 22-13. EPWM_XBAR_REGS Access Type Codes

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

1 PWMXBAROutInvert Register (Offset = 30h) [Reset = 00000000h]

PWMXBAROutInvert is shown in Figure 22-8 and described in Table 22-12.

Return to the Summary Table.

Output Signal Invert Select

Figure 22-8. PWMXBAROutInvert Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
XBAR8	XBAR7	XBAR6	XBAR5	XBAR4	XBAR3	XBAR2	XBAR1
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 22-15. PWMXBAROutInvert Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R-0	0h	Reserved
15	RESERVED	R/W	0h	Reserved
14	RESERVED	R/W	0h	Reserved
13	RESERVED	R/W	0h	Reserved
12	RESERVED	R/W	0h	Reserved
11	RESERVED	R/W	0h	Reserved
10	RESERVED	R/W	0h	Reserved
9	RESERVED	R/W	0h	Reserved
8	RESERVED	R/W	0h	Reserved
7	XBAR8	R/W	0h	Select Output: 0 No inversion 1 Output inverted Reset type: XRSn
6	XBAR7	R/W	0h	Select Output: 0 No inversion 1 Output inverted Reset type: XRSn
5	XBAR6	R/W	0h	Select Output: 0 No inversion 1 Output inverted Reset type: XRSn
4	XBAR5	R/W	0h	Select Output: 0 No inversion 1 Output inverted Reset type: XRSn

Table 22-15. PWMXBAROutInvert Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3	XBAR4	R/W	0h	Select Output: 0 No inversion 1 Output inverted Reset type: XRSn
2	XBAR3	R/W	0h	Select Output: 0 No inversion 1 Output inverted Reset type: XRSn
1	XBAR2	R/W	0h	Select Output: 0 No inversion 1 Output inverted Reset type: XRSn
0	XBAR1	R/W	0h	Select Output: 0 No inversion 1 Output inverted Reset type: XRSn

2 PWMXBARLock Register (Offset = 80h) [Reset = 00000000h]

PWMXBARLock is shown in Figure 22-9 and described in Table 22-13.

Return to the Summary Table.

Configuration Lock register

Figure 22-9. PWMXBARLock Register

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

Table 22-17. PWMXBARLock Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	KEY	R-0/W	0h	Bit-0 of this register can be set only if KEY= 0x5a5a Reset type: XRSn
15-1	RESERVED	R-0	0h	Reserved
0	LOCK	R/WSonce	0h	Locks the configuration for EPWM-XBAR. Once the configuration is locked, writes to the below registers for EPWM-XBAR is blocked. Registers Affected by the LOCK mechanism: PWMXBAR Gx SEL PWMXBAR.Invert 0: Writes to the above registers are allowed 1: Writes to the above registers are blocked Note: [1] LOCK mechanism only apples to writes. Reads are never blocked. Reset type: XRSn

ADVANCE INFORMATION

3 PWMXBARG0SEL_j Register (Offset = 100h + formula) [Reset = 00000000h]

PWMXBARG0SEL_j is shown in Figure 22-10 and described in Table 22-14.

Return to the Summary Table.

PWMXBAR G0 Input Select

Offset = 100h + (j * 40h); where j = 0h to 7h

Figure 22-10. PWMXBARG0SEL_j Register

31	30	29	28	27	26	25	24
INPUT31	INPUT30	INPUT29	INPUT28	INPUT27	INPUT26	INPUT25	INPUT24
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
INPUT23	INPUT22	INPUT21	INPUT20	INPUT19	INPUT18	INPUT17	INPUT16
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
INPUT15	INPUT14	INPUT13	INPUT12	INPUT11	INPUT10	INPUT9	INPUT8
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
INPUT7	INPUT6	INPUT5	INPUT4	INPUT3	INPUT2	INPUT1	INPUT0
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 22-19. PWMXBARG0SEL_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31	INPUT31	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
30	INPUT30	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
29	INPUT29	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
28	INPUT28	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
27	INPUT27	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
26	INPUT26	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn

Table 22-19. PWMXBARG0SEL_j Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
25	INPUT25	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
24	INPUT24	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
23	INPUT23	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
22	INPUT22	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
21	INPUT21	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
20	INPUT20	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
19	INPUT19	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
18	INPUT18	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
17	INPUT17	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
16	INPUT16	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
15	INPUT15	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn

Table 22-19. PWMXBARG0SEL_j Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
14	INPUT14	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
13	INPUT13	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
12	INPUT12	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
11	INPUT11	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
10	INPUT10	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
9	INPUT9	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
8	INPUT8	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
7	INPUT7	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
6	INPUT6	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
5	INPUT5	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
4	INPUT4	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn

Table 22-19. PWMXBARG0SEL_j Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3	INPUT3	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
2	INPUT2	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
1	INPUT1	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
0	INPUT0	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn

4 PWMXBARG1SEL_j Register (Offset = 104h + formula) [Reset = 00000000h]

PWMXBARG1SEL_j is shown in Figure 22-11 and described in Table 22-15.

Return to the Summary Table.

PWMXBAR G1 Input Select

Offset = 104h + (j * 40h); where j = 0h to 7h

Figure 22-11. PWMXBARG1SEL_j Register

31	30	29	28	27	26	25	24
INPUT31	INPUT30	INPUT29	INPUT28	INPUT27	INPUT26	INPUT25	INPUT24
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
INPUT23	INPUT22	INPUT21	INPUT20	INPUT19	INPUT18	INPUT17	INPUT16
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
INPUT15	INPUT14	INPUT13	INPUT12	INPUT11	INPUT10	INPUT9	INPUT8
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
INPUT7	INPUT6	INPUT5	INPUT4	INPUT3	INPUT2	INPUT1	INPUT0
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 22-21. PWMXBARG1SEL_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31	INPUT31	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
30	INPUT30	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
29	INPUT29	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
28	INPUT28	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
27	INPUT27	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
26	INPUT26	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn

Table 22-21. PWMXBARG1SEL_j Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
25	INPUT25	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
24	INPUT24	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
23	INPUT23	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
22	INPUT22	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
21	INPUT21	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
20	INPUT20	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
19	INPUT19	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
18	INPUT18	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
17	INPUT17	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
16	INPUT16	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
15	INPUT15	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn

Table 22-21. PWMXBARG1SEL_j Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
14	INPUT14	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
13	INPUT13	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
12	INPUT12	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
11	INPUT11	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
10	INPUT10	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
9	INPUT9	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
8	INPUT8	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
7	INPUT7	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
6	INPUT6	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
5	INPUT5	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
4	INPUT4	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn

Table 22-21. PWMXBARG1SEL_j Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3	INPUT3	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
2	INPUT2	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
1	INPUT1	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
0	INPUT0	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn

22.3.4 OUTPUTXBAR_REGS Registers

Table 22-16 lists the memory-mapped registers for the OUTPUTXBAR_REGS registers. All register offset addresses not listed in Table 22-16 should be considered as reserved locations and the register contents should not be modified.

Table 22-22. OUTPUTXBAR_REGS Registers

Offset	Acronym	Register Name	Write Protection	Section
14h	OUTPUTXBARFlagInvert	Output Signal inversion register		Go
24h	OUTPUTXBAROutLatch	Output Signal Select Latch		Go
30h	OUTPUTXBAROutInvert	Output Signal Invert Select		Go
80h	OUTPUTXBARLock	Configuration Lock register		Go
100h + formula	OUTPUTXBARG0SEL_j	OUTPUTXBAR G0 Input Select		Go
104h + formula	OUTPUTXBARG1SEL_j	OUTPUTXBAR G1 Input Select		Go

Complex bit access types are encoded to fit into small table cells. Table 22-17 shows the codes that are used for access types in this section.

Table 22-23. OUTPUTXBAR_REGS Access Type Codes

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

1 OUTPUTXBARFlagInvert Register (Offset = 14h) [Reset = 00000000h]

OUTPUTXBARFlagInvert is shown in Figure 22-12 and described in Table 22-18.

Return to the Summary Table.

Output Signal inversion register

Figure 22-12. OUTPUTXBARFlagInvert Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED	RESERVED	RESERVED	RESERVED	XBAR12	XBAR11	XBAR10	XBAR9
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
XBAR8	XBAR7	XBAR6	XBAR5	XBAR4	XBAR3	XBAR2	XBAR1
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 22-25. OUTPUTXBARFlagInvert Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R-0	0h	Reserved
15	RESERVED	R/W	0h	Reserved
14	RESERVED	R/W	0h	Reserved
13	RESERVED	R/W	0h	Reserved
12	RESERVED	R/W	0h	Reserved
11	XBAR12	R/W	0h	Output Signal Invert Before Latch: 0 Signal Is Not Inverted 1 Signal Is Inverted Reset type: XRSn
10	XBAR11	R/W	0h	Output Signal Invert Before Latch: 0 Signal Is Not Inverted 1 Signal Is Inverted Reset type: XRSn
9	XBAR10	R/W	0h	Output Signal Invert Before Latch: 0 Signal Is Not Inverted 1 Signal Is Inverted Reset type: XRSn
8	XBAR9	R/W	0h	Output Signal Invert Before Latch: 0 Signal Is Not Inverted 1 Signal Is Inverted Reset type: XRSn
7	XBAR8	R/W	0h	Output Signal Invert Before Latch: 0 Signal Is Not Inverted 1 Signal Is Inverted Reset type: XRSn

Table 22-25. OUTPUTXBARFlagInvert Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
6	XBAR7	R/W	0h	Output Signal Invert Before Latch: 0 Signal Is Not Inverted 1 Signal Is Inverted Reset type: XRSn
5	XBAR6	R/W	0h	Output Signal Invert Before Latch: 0 Signal Is Not Inverted 1 Signal Is Inverted Reset type: XRSn
4	XBAR5	R/W	0h	Output Signal Invert Before Latch: 0 Signal Is Not Inverted 1 Signal Is Inverted Reset type: XRSn
3	XBAR4	R/W	0h	Output Signal Invert Before Latch: 0 Signal Is Not Inverted 1 Signal Is Inverted Reset type: XRSn
2	XBAR3	R/W	0h	Output Signal Invert Before Latch: 0 Signal Is Not Inverted 1 Signal Is Inverted Reset type: XRSn
1	XBAR2	R/W	0h	Output Signal Invert Before Latch: 0 Signal Is Not Inverted 1 Signal Is Inverted Reset type: XRSn
0	XBAR1	R/W	0h	Output Signal Invert Before Latch: 0 Signal Is Not Inverted 1 Signal Is Inverted Reset type: XRSn

2 OUTPUTXBAROutLatch Register (Offset = 24h) [Reset = 00000000h]

OUTPUTXBAROutLatch is shown in Figure 22-13 and described in Table 22-19.

Return to the Summary Table.

Output Signal Select Latch

Figure 22-13. OUTPUTXBAROutLatch Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED	RESERVED	RESERVED	RESERVED	XBAR12	XBAR11	XBAR10	XBAR9
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
XBAR8	XBAR7	XBAR6	XBAR5	XBAR4	XBAR3	XBAR2	XBAR1
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 22-27. OUTPUTXBAROutLatch Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R-0	0h	Reserved
15	RESERVED	R/W	0h	Reserved
14	RESERVED	R/W	0h	Reserved
13	RESERVED	R/W	0h	Reserved
12	RESERVED	R/W	0h	Reserved
11	XBAR12	R/W	0h	Output Signal Select Latch: 0 Latch Event Not Selected As Output 1 Latched Event Selected As Output Reset type: XRSn
10	XBAR11	R/W	0h	Output Signal Select Latch: 0 Latch Event Not Selected As Output 1 Latched Event Selected As Output Reset type: XRSn
9	XBAR10	R/W	0h	Output Signal Select Latch: 0 Latch Event Not Selected As Output 1 Latched Event Selected As Output Reset type: XRSn
8	XBAR9	R/W	0h	Output Signal Select Latch: 0 Latch Event Not Selected As Output 1 Latched Event Selected As Output Reset type: XRSn
7	XBAR8	R/W	0h	Output Signal Select Latch: 0 Latch Event Not Selected As Output 1 Latched Event Selected As Output Reset type: XRSn

Table 22-27. OUTPUTXBAROutLatch Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
6	XBAR7	R/W	0h	Output Signal Select Latch: 0 Latch Event Not Selected As Output 1 Latched Event Selected As Output Reset type: XRSn
5	XBAR6	R/W	0h	Output Signal Select Latch: 0 Latch Event Not Selected As Output 1 Latched Event Selected As Output Reset type: XRSn
4	XBAR5	R/W	0h	Output Signal Select Latch: 0 Latch Event Not Selected As Output 1 Latched Event Selected As Output Reset type: XRSn
3	XBAR4	R/W	0h	Output Signal Select Latch: 0 Latch Event Not Selected As Output 1 Latched Event Selected As Output Reset type: XRSn
2	XBAR3	R/W	0h	Output Signal Select Latch: 0 Latch Event Not Selected As Output 1 Latched Event Selected As Output Reset type: XRSn
1	XBAR2	R/W	0h	Output Signal Select Latch: 0 Latch Event Not Selected As Output 1 Latched Event Selected As Output Reset type: XRSn
0	XBAR1	R/W	0h	Output Signal Select Latch: 0 Latch Event Not Selected As Output 1 Latched Event Selected As Output Reset type: XRSn

3 OUTPUTXBAROutInvert Register (Offset = 30h) [Reset = 00000000h]

OUTPUTXBAROutInvert is shown in Figure 22-14 and described in Table 22-20.

Return to the Summary Table.

Output Signal Invert Select

Figure 22-14. OUTPUTXBAROutInvert Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED	RESERVED	RESERVED	RESERVED	XBAR12	XBAR11	XBAR10	XBAR9
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
XBAR8	XBAR7	XBAR6	XBAR5	XBAR4	XBAR3	XBAR2	XBAR1
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 22-29. OUTPUTXBAROutInvert Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R-0	0h	Reserved
15	RESERVED	R/W	0h	Reserved
14	RESERVED	R/W	0h	Reserved
13	RESERVED	R/W	0h	Reserved
12	RESERVED	R/W	0h	Reserved
11	XBAR12	R/W	0h	Select Output: 0 No inversion 1 Output inverted Reset type: XRSn
10	XBAR11	R/W	0h	Select Output: 0 No inversion 1 Output inverted Reset type: XRSn
9	XBAR10	R/W	0h	Select Output: 0 No inversion 1 Output inverted Reset type: XRSn
8	XBAR9	R/W	0h	Select Output: 0 No inversion 1 Output inverted Reset type: XRSn
7	XBAR8	R/W	0h	Select Output: 0 No inversion 1 Output inverted Reset type: XRSn

Table 22-29. OUTPUTXBAROutInvert Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
6	XBAR7	R/W	0h	Select Output: 0 No inversion 1 Output inverted Reset type: XRSn
5	XBAR6	R/W	0h	Select Output: 0 No inversion 1 Output inverted Reset type: XRSn
4	XBAR5	R/W	0h	Select Output: 0 No inversion 1 Output inverted Reset type: XRSn
3	XBAR4	R/W	0h	Select Output: 0 No inversion 1 Output inverted Reset type: XRSn
2	XBAR3	R/W	0h	Select Output: 0 No inversion 1 Output inverted Reset type: XRSn
1	XBAR2	R/W	0h	Select Output: 0 No inversion 1 Output inverted Reset type: XRSn
0	XBAR1	R/W	0h	Select Output: 0 No inversion 1 Output inverted Reset type: XRSn

4 OUTPUTXBARLock Register (Offset = 80h) [Reset = 00000000h]

OUTPUTXBARLock is shown in Figure 22-15 and described in Table 22-21.

Return to the Summary Table.

Configuration Lock register

Figure 22-15. OUTPUTXBARLock Register

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

Table 22-31. OUTPUTXBARLock Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	KEY	R-0/W	0h	Bit-0 of this register can be set only if KEY= 0x5a5a Reset type: XRSn
15-1	RESERVED	R-0	0h	Reserved
0	LOCK	R/WSonce	0h	Locks the configuration for EOUTPUT-XBAR. Once the configuration is locked, writes to the below registers for EOUTPUT-XBAR is blocked. Registers Affected by the LOCK mechanism: OUTPUTXBAR Gx SEL OUTPUTXBAR.Invert 0: Writes to the above registers are allowed 1: Writes to the above registers are blocked Note: [1] LOCK mechanism only apples to writes. Reads are never blocked. Reset type: XRSn

5 OUTPUTXBARG0SEL_j Register (Offset = 100h + formula) [Reset = 00000000h]

OUTPUTXBARG0SEL_j is shown in Figure 22-16 and described in Table 22-22.

Return to the Summary Table.

OUTPUTXBAR G0 Input Select

Offset = 100h + (j * 40h); where j = 0h to Bh

Figure 22-16. OUTPUTXBARG0SEL_j Register

31	30	29	28	27	26	25	24
INPUT31	INPUT30	INPUT29	INPUT28	INPUT27	INPUT26	INPUT25	INPUT24
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
INPUT23	INPUT22	INPUT21	INPUT20	INPUT19	INPUT18	INPUT17	INPUT16
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
INPUT15	INPUT14	INPUT13	INPUT12	INPUT11	INPUT10	INPUT9	INPUT8
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
INPUT7	INPUT6	INPUT5	INPUT4	INPUT3	INPUT2	INPUT1	INPUT0
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 22-33. OUTPUTXBARG0SEL_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31	INPUT31	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
30	INPUT30	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
29	INPUT29	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
28	INPUT28	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
27	INPUT27	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
26	INPUT26	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn

Table 22-33. OUTPUTXBARG0SEL_j Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
25	INPUT25	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
24	INPUT24	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
23	INPUT23	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
22	INPUT22	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
21	INPUT21	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
20	INPUT20	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
19	INPUT19	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
18	INPUT18	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
17	INPUT17	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
16	INPUT16	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
15	INPUT15	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn

Table 22-33. OUTPUTXBARG0SEL_j Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
14	INPUT14	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
13	INPUT13	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
12	INPUT12	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
11	INPUT11	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
10	INPUT10	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
9	INPUT9	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
8	INPUT8	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
7	INPUT7	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
6	INPUT6	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
5	INPUT5	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
4	INPUT4	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn

Table 22-33. OUTPUTXBARG0SEL_j Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3	INPUT3	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
2	INPUT2	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
1	INPUT1	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
0	INPUT0	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn

6 OUTPUTXBARG1SEL_j Register (Offset = 104h + formula) [Reset = 00000000h]

OUTPUTXBARG1SEL_j is shown in Figure 22-17 and described in Table 22-23.

Return to the Summary Table.

OUTPUTXBAR G1 Input Select

Offset = 104h + (j * 40h); where j = 0h to Bh

Figure 22-17. OUTPUTXBARG1SEL_j Register

31	30	29	28	27	26	25	24
INPUT31	INPUT30	INPUT29	INPUT28	INPUT27	INPUT26	INPUT25	INPUT24
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
INPUT23	INPUT22	INPUT21	INPUT20	INPUT19	INPUT18	INPUT17	INPUT16
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
INPUT15	INPUT14	INPUT13	INPUT12	INPUT11	INPUT10	INPUT9	INPUT8
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
INPUT7	INPUT6	INPUT5	INPUT4	INPUT3	INPUT2	INPUT1	INPUT0
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 22-35. OUTPUTXBARG1SEL_j Register Field Descriptions

Bit	Field	Type	Reset	Description
31	INPUT31	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
30	INPUT30	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
29	INPUT29	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
28	INPUT28	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
27	INPUT27	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
26	INPUT26	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn

Table 22-35. OUTPUTXBARG1SEL_j Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
25	INPUT25	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
24	INPUT24	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
23	INPUT23	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
22	INPUT22	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
21	INPUT21	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
20	INPUT20	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
19	INPUT19	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
18	INPUT18	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
17	INPUT17	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
16	INPUT16	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
15	INPUT15	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn

Table 22-35. OUTPUTXBARG1SEL_j Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
14	INPUT14	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
13	INPUT13	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
12	INPUT12	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
11	INPUT11	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
10	INPUT10	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
9	INPUT9	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
8	INPUT8	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
7	INPUT7	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
6	INPUT6	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
5	INPUT5	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
4	INPUT4	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn

Table 22-35. OUTPUTXBARG1SEL_j Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3	INPUT3	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
2	INPUT2	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
1	INPUT1	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn
0	INPUT0	R/W	0h	Select input: 0 Not Selected 1 Input Selected Reset type: XRSn

22.3.5 SYNC_SOC_REGS Registers

Table 22-24 lists the memory-mapped registers for the SYNC_SOC_REGS registers. All register offset addresses not listed in Table 22-24 should be considered as reserved locations and the register contents should not be modified.

Table 22-36. SYNC_SOC_REGS Registers

Offset	Acronym	Register Name	Write Protection	Section
0h	SYNCSELECT	Sync Input and Output Select Register		Go
4h	ADCSOCOUTSELECT	External ADCSOC Select Register		Go
8h	SYNCSOCLOCK	SYNCSEL and EXTADCSOC Select Lock register		Go

Complex bit access types are encoded to fit into small table cells. Table 22-25 shows the codes that are used for access types in this section.

Table 22-37. SYNC_SOC_REGS Access Type Codes

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

1 SYNCSELECT Register (Offset = 0h) [Reset = 00000000h]

SYNCSELECT is shown in Figure 22-18 and described in Table 22-26.

Return to the Summary Table.

Sync Input and Output Select Register

Figure 22-18. SYNCSELECT Register

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

Table 22-39. SYNCSELECT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R-0	0h	Reserved
28-24	SYNCOUT	R/W	0h	Select Syncout Source: 00000: PWM0SYNCOUT selected to drive the EXTSYNCOUT pin. 00001: PWM1SYNCOUT selected to drive the EXTSYNCOUT pin. 00010: PWM2SYNCOUT selected to drive the EXTSYNCOUT pin. 00011: PWM3SYNCOUT selected to drive the EXTSYNCOUT pin. 00100: PWM4SYNCOUT selected to drive the EXTSYNCOUT pin. 00101: Reserved 00110: Reserved 00111: Reserved 01000: Reserved 01001: Reserved 01010: Reserved 01011: Reserved 01100: Reserved 01101: Reserved 01110: Reserved 01111: Reserved 10000: Reserved 10001: Reserved 10010: Reserved 10011: Reserved 10100: Reserved 10101: Reserved 10110: Reserved 10111: Reserved 11000: ECAP0SYNCOUT selected to drive the EXTSYNCOUT pin. 11001: ECAP1SYNCOUT selected to drive the EXTSYNCOUT pin. 11010: Reserved 11011: Reserved 11100: Reserved 11101: Reserved 11110: Reserved 11111: Reserved Notes: [1] Reserved position defaults to 00 selection Reset type: SYSRSn
23-0	RESERVED	R-0	0h	Reserved

ADVANCE INFORMATION

2 ADCSOCOUTSELECT Register (Offset = 4h) [Reset = 00000000h]

ADCSOCOUTSELECT is shown in Figure 22-19 and described in Table 22-27.

Return to the Summary Table.

External ADCSOC Select Register

Figure 22-19. ADCSOCOUTSELECT Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED			PWM4SOCBEN	PWM3SOCBEN	PWM2SOCBEN	PWM1SOCBEN	PWM0SOCBEN
R-0-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED			PWM4SOCAEN	PWM3SOCAEN	PWM2SOCAEN	PWM1SOCAEN	PWM0SOCAEN
R-0-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 22-41. ADCSOCOUTSELECT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-21	RESERVED	R-0	0h	Reserved
20	PWM4SOCBEN	R/W	0h	ADCSOCBOn source select: 0: Respective PWM SOCB output is not selected 1: Respective PWM SOCB output is selected Reset type: SYSRSn
19	PWM3SOCBEN	R/W	0h	ADCSOCBOn source select: 0: Respective PWM SOCB output is not selected 1: Respective PWM SOCB output is selected Reset type: SYSRSn
18	PWM2SOCBEN	R/W	0h	ADCSOCBOn source select: 0: Respective PWM SOCB output is not selected 1: Respective PWM SOCB output is selected Reset type: SYSRSn
17	PWM1SOCBEN	R/W	0h	ADCSOCBOn source select: 0: Respective PWM SOCB output is not selected 1: Respective PWM SOCB output is selected Reset type: SYSRSn
16	PWM0SOCBEN	R/W	0h	ADCSOCBOn source select: 0: Respective PWM SOCB output is not selected 1: Respective PWM SOCB output is selected Reset type: SYSRSn
15-5	RESERVED	R-0	0h	Reserved
4	PWM4SOCAEN	R/W	0h	ADCSOCAOn source select: 0: Respective PWM SOCA output is not selected 1: Respective PWM SOCA output is selected Reset type: SYSRSn

ADVANCE INFORMATION

Table 22-41. ADCSOCOUTSELECT Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3	PWM3SOCAEN	R/W	0h	ADCSOCAOn source select: 0: Respective PWM SOCA output is not selected 1: Respective PWM SOCA output is selected Reset type: SYSRSn
2	PWM2SOCAEN	R/W	0h	ADCSOCAOn source select: 0: Respective PWM SOCA output is not selected 1: Respective PWM SOCA output is selected Reset type: SYSRSn
1	PWM1SOCAEN	R/W	0h	ADCSOCAOn source select: 0: Respective PWM SOCA output is not selected 1: Respective PWM SOCA output is selected Reset type: SYSRSn
0	PWM0SOCAEN	R/W	0h	ADCSOCAOn source select: 0: Respective PWM SOCA output is not selected 1: Respective PWM SOCA output is selected Reset type: SYSRSn

3 SYNCSOCLOCK Register (Offset = 8h) [Reset = 00000000h]

SYNCSOCLOCK is shown in Figure 22-20 and described in Table 22-28.

Return to the Summary Table.

SYNCSEL and EXTADCSOC Select Lock register

Figure 22-20. SYNCSOCLOCK Register

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

Table 22-43. SYNCSOCLOCK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R-0	0h	Reserved
1	ADCSOCOUTSELECT	R/WSonce	0h	ADCSOCOUTSELECT Register Lock bit: 0: Respective register is not locked 1: Respective register is locked. Notes: [1] Any bit in this register, once set can only be creaed through a SYSRSn. Write of 0 to any bit of this regtister has no effect [2] The locking mechanism applies to only writes. Reads to the registers which have LOCK protection are always allowed Reset type: SYSRSn
0	SYNCSELECT	R/WSonce	0h	SYNCSELECT Register Lock bit: 0: Respective register is not locked 1: Respective register is locked. Notes: [1] Any bit in this register, once set can only be creaed through a SYSRSn. Write of 0 to any bit of this regtister has no effect [2] The locking mechanism applies to only writes. Reads to the registers which have LOCK protection are always allowed Reset type: SYSRSn

22.3.6 OUTPUTXBAR_FLAG_REGS Registers

Table 22-29 lists the memory-mapped registers for the OUTPUTXBAR_FLAG_REGS registers. All register offset addresses not listed in Table 22-29 should be considered as reserved locations and the register contents should not be modified.

Table 22-44. OUTPUTXBAR_FLAG_REGS Registers

Offset	Acronym	Register Name	Write Protection	Section
10h	OUTPUTXBARStatus	Output Signal Status register		Go
18h	OUTPUTXBARFlag	Output latched flag register		Go
1Ch	OUTPUTXBARFlagClear	Output latched flag clear register		Go
20h	OUTPUTXBARFlagForce	Output latched flag Force register		Go

Complex bit access types are encoded to fit into small table cells. Table 22-30 shows the codes that are used for access types in this section.

Table 22-45. OUTPUTXBAR_FLAG_REGS Access Type Codes

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

1 OUTPUTXBARStatus Register (Offset = 10h) [Reset = 00000000h]

OUTPUTXBARStatus is shown in Figure 22-21 and described in Table 22-31.

Return to the Summary Table.

Output Signal Status register

Figure 22-21. OUTPUTXBARStatus Register

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

Table 22-47. OUTPUTXBARStatus Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R-0	0h	Reserved
0	STS	R	0h	Output Signal Status: 0 Low State 1 High State Reset type: XRSn

2 OUTPUTXBARFlag Register (Offset = 18h) [Reset = 00000000h]

OUTPUTXBARFlag is shown in Figure 22-22 and described in Table 22-32.

Return to the Summary Table.

Output latched flag register

Figure 22-22. OUTPUTXBARFlag Register

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

Table 22-49. OUTPUTXBARFlag Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R-0	0h	Reserved
0	FLG	R	0h	Output Signal Latched Flag: 0 No Input Latched Event 1 Latched Event Reset type: XRSn

3 OUTPUTXBARFlagClear Register (Offset = 1Ch) [Reset = 00000000h]

OUTPUTXBARFlagClear is shown in Figure 22-23 and described in Table 22-33.

Return to the Summary Table.

Output latched flag clear register

Figure 22-23. OUTPUTXBARFlagClear Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
R-0-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED															FLG
R-0-0h															R-0/W1C-0h

Table 22-51. OUTPUTXBARFlagClear Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R-0	0h	Reserved
0	FLG	R-0/W1C	0h	Output Signal Latched Flag Clear: Write of 1 will clear flag. If Flag is being cleared by S/W and a new flag is being latched, hardware has priority. Reset type: XRSn

ADVANCE INFORMATION

4 OUTPUTXBARFlagForce Register (Offset = 20h) [Reset = 00000000h]

OUTPUTXBARFlagForce is shown in Figure 22-24 and described in Table 22-34.

Return to the Summary Table.

Output latched flag Force register

Figure 22-24. OUTPUTXBARFlagForce Register

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

Table 22-53. OUTPUTXBARFlagForce Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R-0	0h	Reserved
0	FLG	R-0/W1S	0h	Output Signal Latched Flag Force: Write of 1 will set flag. Reset type: XRSn

22.3.7 INPUT_FLAG_XBAR_REGS Registers

Table 22-35 lists the memory-mapped registers for the INPUT_FLAG_XBAR_REGS registers. All register offset addresses not listed in Table 22-35 should be considered as reserved locations and the register contents should not be modified.

Table 22-54. INPUT_FLAG_XBAR_REGS Registers

Offset	Acronym	Register Name	Write Protection	Section
0h	XBARFLG1	X-Bar Input Flag Register 1		Go
4h	XBARFLG2	X-Bar Input Flag Register 2		Go
10h	XBARCLR1	X-Bar Input Flag Clear Register 1		Go
14h	XBARCLR2	X-Bar Input Flag Clear Register 2		Go

Complex bit access types are encoded to fit into small table cells. Table 22-36 shows the codes that are used for access types in this section.

Table 22-55. INPUT_FLAG_XBAR_REGS Access Type Codes

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

1 XBARFLG1 Register (Offset = 0h) [Reset = 00000000h]

XBARFLG1 is shown in Figure 22-25 and described in Table 22-37.

Return to the Summary Table.

X-Bar Input Flag Register 1

Figure 22-25. XBARFLG1 Register

31	30	29	28	27	26	25	24
INPUTXBAR12	INPUTXBAR11	INPUTXBAR10	INPUTXBAR9	INPUTXBAR8	INPUTXBAR7	INPUTXBAR6	INPUTXBAR5
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
23	22	21	20	19	18	17	16
INPUTXBAR4	INPUTXBAR3	INPUTXBAR2	INPUTXBAR1	ADC2EVT4	ADC2EVT3	ADC2EVT2	ADC2EVT1
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
15	14	13	12	11	10	9	8
ADC1EVT4	ADC1EVT3	ADC1EVT2	ADC1EVT1	ADC0EVT4	ADC0EVT3	ADC0EVT2	ADC0EVT1
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
7	6	5	4	3	2	1	0
CMPSS3_CTRIPL	CMPSS3_CTRIPH	CMPSS2_CTRIPL	CMPSS2_CTRIPH	CMPSS1_CTRIPL	CMPSS1_CTRIPH	CMPSS0_CTRIPL	CMPSS0_CTRIPH
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

Table 22-57. XBARFLG1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	INPUTXBAR12	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: Corresponding Input was triggered 0: Corresponding Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
30	INPUTXBAR11	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: Corresponding Input was triggered 0: Corresponding Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
29	INPUTXBAR10	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: Corresponding Input was triggered 0: Corresponding Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
28	INPUTXBAR9	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: Corresponding Input was triggered 0: Corresponding Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn

Table 22-57. XBARFLG1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
27	INPUTXBAR8	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: Corresponding Input was triggered 0: Corresponding Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
26	INPUTXBAR7	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: Corresponding Input was triggered 0: Corresponding Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
25	INPUTXBAR6	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: Corresponding Input was triggered 0: Corresponding Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
24	INPUTXBAR5	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: Corresponding Input was triggered 0: Corresponding Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
23	INPUTXBAR4	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: Corresponding Input was triggered 0: Corresponding Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
22	INPUTXBAR3	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: Corresponding Input was triggered 0: Corresponding Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
21	INPUTXBAR2	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: Corresponding Input was triggered 0: Corresponding Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn

Table 22-57. XBARFLG1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
20	INPUTXBAR1	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: Corresponding Input was triggered 0: Corresponding Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
19	ADC2EVT4	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: Corresponding Input was triggered 0: Corresponding Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
18	ADC2EVT3	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: Corresponding Input was triggered 0: Corresponding Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
17	ADC2EVT2	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: Corresponding Input was triggered 0: Corresponding Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
16	ADC2EVT1	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: Corresponding Input was triggered 0: Corresponding Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
15	ADC1EVT4	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: Corresponding Input was triggered 0: Corresponding Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
14	ADC1EVT3	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: Corresponding Input was triggered 0: Corresponding Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn

Table 22-57. XBARFLG1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
13	ADC1EVT2	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: Corresponding Input was triggered 0: Corresponding Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
12	ADC1EVT1	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: Corresponding Input was triggered 0: Corresponding Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
11	ADC0EVT4	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: Corresponding Input was triggered 0: Corresponding Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
10	ADC0EVT3	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: Corresponding Input was triggered 0: Corresponding Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
9	ADC0EVT2	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: Corresponding Input was triggered 0: Corresponding Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
8	ADC0EVT1	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: Corresponding Input was triggered 0: Corresponding Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
7	CMPSS3_CTRIPL	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: Corresponding Input was triggered 0: Corresponding Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn

Table 22-57. XBARFLG1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
6	CMPSS3_CTRIPH	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: Corresponding Input was triggered 0: Corresponding Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
5	CMPSS2_CTRIPL	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: Corresponding Input was triggered 0: Corresponding Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
4	CMPSS2_CTRIPH	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: Corresponding Input was triggered 0: Corresponding Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
3	CMPSS1_CTRIPL	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: Corresponding Input was triggered 0: Corresponding Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
2	CMPSS1_CTRIPH	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: Corresponding Input was triggered 0: Corresponding Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
1	CMPSS0_CTRIPL	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: Corresponding Input was triggered 0: Corresponding Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
0	CMPSS0_CTRIPH	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: Corresponding Input was triggered 0: Corresponding Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn

2 XBARFLG2 Register (Offset = 4h) [Reset = 00000000h]

XBARFLG2 is shown in Figure 22-26 and described in Table 22-38.

Return to the Summary Table.

X-Bar Input Flag Register 2

Figure 22-26. XBARFLG2 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED			EQEP2_STROBE_SYNCOUT	EQEP2_INDEX_SYNCOUT	EQEP1_STROBE_SYNCOUT	EQEP1_INDEX_SYNCOUT	EQEP0_STROBE_SYNCOUT
R-0-0h			R-0h	R-0h	R-0h	R-0h	R-0h
15	14	13	12	11	10	9	8
EQEP0_INDEX_SYNCOUT	EQEP2_ERR	EQEP1_ERR	EQEP0_ERR	MCAN0_FEVT2	MCAN0_FEVT1	MCAN0_FEVT0	EXTSYNCOUT
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
7	6	5	4	3	2	1	0
ADCSOCB	ADCSOCA	ECAP1_OUT	ECAP0_OUT	INPUTXBAR16	INPUTXBAR15	INPUTXBAR14	INPUTXBAR13
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

Table 22-59. XBARFLG2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-21	RESERVED	R-0	0h	Reserved
20	EQEP2_STROBE_SYNCOUT	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: Corresponding Input was triggered 0: Corresponding Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
19	EQEP2_INDEX_SYNCOUT	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: Corresponding Input was triggered 0: Corresponding Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
18	EQEP1_STROBE_SYNCOUT	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: Corresponding Input was triggered 0: Corresponding Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn

Table 22-59. XBARFLG2 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
17	EQEP1_INDEX_SYNCOUT	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: Corresponding Input was triggered 0: Corresponding Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
16	EQEP0_STROBE_SYNCOUT	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: Corresponding Input was triggered 0: Corresponding Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
15	EQEP0_INDEX_SYNCOUT	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: Corresponding Input was triggered 0: Corresponding Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
14	EQEP2_ERR	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: Corresponding Input was triggered 0: Corresponding Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
13	EQEP1_ERR	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: Corresponding Input was triggered 0: Corresponding Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
12	EQEP0_ERR	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: Corresponding Input was triggered 0: Corresponding Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
11	MCAN0_FEVT2	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: Corresponding Input was triggered 0: Corresponding Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn

Table 22-59. XBARFLG2 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
10	MCAN0_FEVT1	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: Corresponding Input was triggered 0: Corresponding Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
9	MCAN0_FEVT0	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: Corresponding Input was triggered 0: Corresponding Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
8	EXTSYNCOUT	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: Corresponding Input was triggered 0: Corresponding Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
7	ADCSOCB	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: Corresponding Input was triggered 0: Corresponding Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
6	ADCSOCA	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: Corresponding Input was triggered 0: Corresponding Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
5	ECAP1_OUT	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: Corresponding Input was triggered 0: Corresponding Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
4	ECAP0_OUT	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: Corresponding Input was triggered 0: Corresponding Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn

Table 22-59. XBARFLG2 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3	INPUTXBAR16	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: Corresponding Input was triggered 0: Corresponding Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
2	INPUTXBAR15	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: Corresponding Input was triggered 0: Corresponding Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
1	INPUTXBAR14	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: Corresponding Input was triggered 0: Corresponding Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn
0	INPUTXBAR13	R	0h	This register is used to Flag the inputs of the X-Bars to provide software knowledge of the input sources which got triggered. 1: Corresponding Input was triggered 0: Corresponding Input was not triggered Note: [1] setting of this bit has priority over clear by software Reset type: CPU1.SYSRSn

3 XBARCLR1 Register (Offset = 10h) [Reset = 00000000h]

XBARCLR1 is shown in Figure 22-27 and described in Table 22-39.

Return to the Summary Table.

X-Bar Input Flag Clear Register 1

Figure 22-27. XBARCLR1 Register

31	30	29	28	27	26	25	24
INPUTXBAR12	INPUTXBAR11	INPUTXBAR10	INPUTXBAR9	INPUTXBAR8	INPUTXBAR7	INPUTXBAR6	INPUTXBAR5
R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h
23	22	21	20	19	18	17	16
INPUTXBAR4	INPUTXBAR3	INPUTXBAR2	INPUTXBAR1	ADC2EVT4	ADC2EVT3	ADC2EVT2	ADC2EVT1
R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h
15	14	13	12	11	10	9	8
ADC1EVT4	ADC1EVT3	ADC1EVT2	ADC1EVT1	ADC0EVT4	ADC0EVT3	ADC0EVT2	ADC0EVT1
R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h
7	6	5	4	3	2	1	0
CMPSS3_CTRIPL	CMPSS3_CTRIPH	CMPSS2_CTRIPL	CMPSS2_CTRIPH	CMPSS1_CTRIPL	CMPSS1_CTRIPH	CMPSS0_CTRIPL	CMPSS0_CTRIPH
R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h

Table 22-61. XBARCLR1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	INPUTXBAR12	R-0/W1C	0h	Writing 1 to a bit in this register clears the corresponding bit in the XBARFLG1 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
30	INPUTXBAR11	R-0/W1C	0h	Writing 1 to a bit in this register clears the corresponding bit in the XBARFLG1 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
29	INPUTXBAR10	R-0/W1C	0h	Writing 1 to a bit in this register clears the corresponding bit in the XBARFLG1 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
28	INPUTXBAR9	R-0/W1C	0h	Writing 1 to a bit in this register clears the corresponding bit in the XBARFLG1 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
27	INPUTXBAR8	R-0/W1C	0h	Writing 1 to a bit in this register clears the corresponding bit in the XBARFLG1 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
26	INPUTXBAR7	R-0/W1C	0h	Writing 1 to a bit in this register clears the corresponding bit in the XBARFLG1 register. Writing 0 has no effect Reset type: CPU1.SYSRSn

Table 22-61. XBARCLR1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
25	INPUTXBAR6	R-0/W1C	0h	Writing 1 to a bit in this register clears the corresponding bit in the XBARFLG1 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
24	INPUTXBAR5	R-0/W1C	0h	Writing 1 to a bit in this register clears the corresponding bit in the XBARFLG1 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
23	INPUTXBAR4	R-0/W1C	0h	Writing 1 to a bit in this register clears the corresponding bit in the XBARFLG1 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
22	INPUTXBAR3	R-0/W1C	0h	Writing 1 to a bit in this register clears the corresponding bit in the XBARFLG1 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
21	INPUTXBAR2	R-0/W1C	0h	Writing 1 to a bit in this register clears the corresponding bit in the XBARFLG1 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
20	INPUTXBAR1	R-0/W1C	0h	Writing 1 to a bit in this register clears the corresponding bit in the XBARFLG1 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
19	ADC2EVT4	R-0/W1C	0h	Writing 1 to a bit in this register clears the corresponding bit in the XBARFLG1 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
18	ADC2EVT3	R-0/W1C	0h	Writing 1 to a bit in this register clears the corresponding bit in the XBARFLG1 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
17	ADC2EVT2	R-0/W1C	0h	Writing 1 to a bit in this register clears the corresponding bit in the XBARFLG1 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
16	ADC2EVT1	R-0/W1C	0h	Writing 1 to a bit in this register clears the corresponding bit in the XBARFLG1 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
15	ADC1EVT4	R-0/W1C	0h	Writing 1 to a bit in this register clears the corresponding bit in the XBARFLG1 register. Writing 0 has no effect Reset type: CPU1.SYSRSn

Table 22-61. XBARCLR1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
14	ADC1EVT3	R-0/W1C	0h	Writing 1 to a bit in this register clears the corresponding bit in the XBARFLG1 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
13	ADC1EVT2	R-0/W1C	0h	Writing 1 to a bit in this register clears the corresponding bit in the XBARFLG1 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
12	ADC1EVT1	R-0/W1C	0h	Writing 1 to a bit in this register clears the corresponding bit in the XBARFLG1 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
11	ADC0EVT4	R-0/W1C	0h	Writing 1 to a bit in this register clears the corresponding bit in the XBARFLG1 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
10	ADC0EVT3	R-0/W1C	0h	Writing 1 to a bit in this register clears the corresponding bit in the XBARFLG1 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
9	ADC0EVT2	R-0/W1C	0h	Writing 1 to a bit in this register clears the corresponding bit in the XBARFLG1 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
8	ADC0EVT1	R-0/W1C	0h	Writing 1 to a bit in this register clears the corresponding bit in the XBARFLG1 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
7	CMPSS3_CTRIPL	R-0/W1C	0h	Writing 1 to a bit in this register clears the corresponding bit in the XBARFLG1 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
6	CMPSS3_CTRIPH	R-0/W1C	0h	Writing 1 to a bit in this register clears the corresponding bit in the XBARFLG1 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
5	CMPSS2_CTRIPL	R-0/W1C	0h	Writing 1 to a bit in this register clears the corresponding bit in the XBARFLG1 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
4	CMPSS2_CTRIPH	R-0/W1C	0h	Writing 1 to a bit in this register clears the corresponding bit in the XBARFLG1 register. Writing 0 has no effect Reset type: CPU1.SYSRSn

Table 22-61. XBARCLR1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3	CMPSS1_CTRIPL	R-0/W1C	0h	Writing 1 to a bit in this register clears the corresponding bit in the XBARFLG1 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
2	CMPSS1_CTRIPH	R-0/W1C	0h	Writing 1 to a bit in this register clears the corresponding bit in the XBARFLG1 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
1	CMPSS0_CTRIPL	R-0/W1C	0h	Writing 1 to a bit in this register clears the corresponding bit in the XBARFLG1 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
0	CMPSS0_CTRIPH	R-0/W1C	0h	Writing 1 to a bit in this register clears the corresponding bit in the XBARFLG1 register. Writing 0 has no effect Reset type: CPU1.SYSRSn

4 XBARCLR2 Register (Offset = 14h) [Reset = 00000000h]

XBARCLR2 is shown in Figure 22-28 and described in Table 22-40.

Return to the Summary Table.

X-Bar Input Flag Clear Register 2

Figure 22-28. XBARCLR2 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED			EQEP2_STROBE_SYNCOUT	EQEP2_INDEX_SYNCOUT	EQEP1_STROBE_SYNCOUT	EQEP1_INDEX_SYNCOUT	EQEP0_STROBE_SYNCOUT
R-0-0h			R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h
15	14	13	12	11	10	9	8
EQEP0_INDEX_SYNCOUT	EQEP2_ERR	EQEP1_ERR	EQEP0_ERR	MCAN0_FEVT2	MCAN0_FEVT1	MCAN0_FEVT0	EXTSYNCOUT
R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h
7	6	5	4	3	2	1	0
ADCSOCB	ADCSOCA	ECAP1_OUT	ECAP0_OUT	INPUTXBAR16	INPUTXBAR15	INPUTXBAR14	INPUTXBAR13
R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h

Table 22-63. XBARCLR2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-21	RESERVED	R-0	0h	Reserved
20	EQEP2_STROBE_SYNCOUT	R-0/W1C	0h	Writing 1 to a bit in this register clears the corresponding bit in the XBARFLG2 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
19	EQEP2_INDEX_SYNCOUT	R-0/W1C	0h	Writing 1 to a bit in this register clears the corresponding bit in the XBARFLG2 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
18	EQEP1_STROBE_SYNCOUT	R-0/W1C	0h	Writing 1 to a bit in this register clears the corresponding bit in the XBARFLG2 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
17	EQEP1_INDEX_SYNCOUT	R-0/W1C	0h	Writing 1 to a bit in this register clears the corresponding bit in the XBARFLG2 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
16	EQEP0_STROBE_SYNCOUT	R-0/W1C	0h	Writing 1 to a bit in this register clears the corresponding bit in the XBARFLG2 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
15	EQEP0_INDEX_SYNCOUT	R-0/W1C	0h	Writing 1 to a bit in this register clears the corresponding bit in the XBARFLG2 register. Writing 0 has no effect Reset type: CPU1.SYSRSn

Table 22-63. XBARCLR2 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
14	EQEP2_ERR	R-0/W1C	0h	Writing 1 to a bit in this register clears the corresponding bit in the XBARFLG2 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
13	EQEP1_ERR	R-0/W1C	0h	Writing 1 to a bit in this register clears the corresponding bit in the XBARFLG2 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
12	EQEP0_ERR	R-0/W1C	0h	Writing 1 to a bit in this register clears the corresponding bit in the XBARFLG2 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
11	MCAN0_FEVT2	R-0/W1C	0h	Writing 1 to a bit in this register clears the corresponding bit in the XBARFLG2 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
10	MCAN0_FEVT1	R-0/W1C	0h	Writing 1 to a bit in this register clears the corresponding bit in the XBARFLG2 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
9	MCAN0_FEVT0	R-0/W1C	0h	Writing 1 to a bit in this register clears the corresponding bit in the XBARFLG2 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
8	EXTSYNCOUT	R-0/W1C	0h	Writing 1 to a bit in this register clears the corresponding bit in the XBARFLG2 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
7	ADCSOCB	R-0/W1C	0h	Writing 1 to a bit in this register clears the corresponding bit in the XBARFLG2 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
6	ADCSOCA	R-0/W1C	0h	Writing 1 to a bit in this register clears the corresponding bit in the XBARFLG2 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
5	ECAP1_OUT	R-0/W1C	0h	Writing 1 to a bit in this register clears the corresponding bit in the XBARFLG2 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
4	ECAP0_OUT	R-0/W1C	0h	Writing 1 to a bit in this register clears the corresponding bit in the XBARFLG2 register. Writing 0 has no effect Reset type: CPU1.SYSRSn

Table 22-63. XBARCLR2 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3	INPUTXBAR16	R-0/W1C	0h	Writing 1 to a bit in this register clears the corresponding bit in the XBARFLG2 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
2	INPUTXBAR15	R-0/W1C	0h	Writing 1 to a bit in this register clears the corresponding bit in the XBARFLG2 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
1	INPUTXBAR14	R-0/W1C	0h	Writing 1 to a bit in this register clears the corresponding bit in the XBARFLG2 register. Writing 0 has no effect Reset type: CPU1.SYSRSn
0	INPUTXBAR13	R-0/W1C	0h	Writing 1 to a bit in this register clears the corresponding bit in the XBARFLG2 register. Writing 0 has no effect Reset type: CPU1.SYSRSn

Chapter 23

Unified Communication Peripheral (UNICOMM)

UNICOMM is a run-time configurable unified communications module capable of supporting either a UART, I^2C or SPI protocol. Certain UNICOMM variants provide extended support for LIN and SMBus protocols.

ADVANCE INFORMATION

23.1 Overview

UNICOMM is a unified serial communication peripheral capable of operating in multiple different standard serial protocols including UART, SPI, and I2C. Register settings configure which protocol is selected for a given UNICOMM peripheral instance via the IPMODE field. A UNICOMM instance can only be configured for one protocol mode at a time.

Each UNICOMM instance can be configured to operate in one of the following protocol modes:

- UART - Universal Asychronous Receiver/Transmitter
- SPI - Serial Peripheral Interface
- I2CC - Inter Integrated Circuit Controller
- I2CT - Inter Integrated Circuit Target

This document commonly uses the abbreviation UCx to refer to a specific UNICOMM instance, with x being the instance number (for example, UC3 refers to the 3rd UNICOMM instance.) Individual UNICOMM instances are grouped together into designated Scalable Peripheral Groupings (SPGs) for high level configurations such as loopback modes and I2C pairings. Each grouping uses the naming scheme SPGx (or Sx in the UNICOMM register blocks) where x represents the group number the UNICOMM instance belongs to.

There are six UNICOMM instances grouped in two SPG instances present on this device. Below are the configuration options and peripheral types available for each UNICOMM instance. UNICOMM instances UC0, UC1 and UC2 are grouped into SPG0/S0 and instances UC3, UC4, and UC5 are grouped together in SPG1/S1.

Table 23-1. Available UNICOMM Configurations Per Instance

SPG Instance	UNICOMM Instance	Supported Serial Protocols	Available Peripheral Types
SPG0	UC0	UART, SPI, I2C	Basic UART, Basic SPI, Basic I2C Controller/Target
	UC1	UART, SPI, I2C	Basic UART, Basic SPI, Basic I2C Controller/Target
	UC2	UART, LIN, I2C, SMBUS	Basic+ UART, Advanced I2C Controller/Target
SPG1	UC3	UART, SPI, I2C	Basic UART, Basic SPI, Basic I2C Controller/Target
	UC4	UART, SPI, I2C	Basic UART, Basic SPI, Basic I2C Controller/Target
	UC5	UART, LIN, I2C, SMBUS	Basic+ UART, Advanced I2C Controller/Target

This chapter describes all the features and configurations that are common across all UNICOMM instances regardless of the IPMODE configured. This includes FIFO operation, enabling and resetting a UNICOMM instance, the generic transmit and receive sequence, and generic initialization sequence for UNICOMM.

There are also two test mode features, loopback and I2C pairing, that involve interaction between two or more UNICOMM instances. This chapter begins by describing these top-level test features, and their dependance on the UNICOMM's SPG (Scalable Peripheral Group).

The detailed and protocol-specific operation of each UNICOMM protocol mode is described in each of the below dedicated chapters:

- UART - Chapter 24
- SPI - Chapter 26
- I2CC / I2CT - Chapter 25

23.1.1 Block Diagram

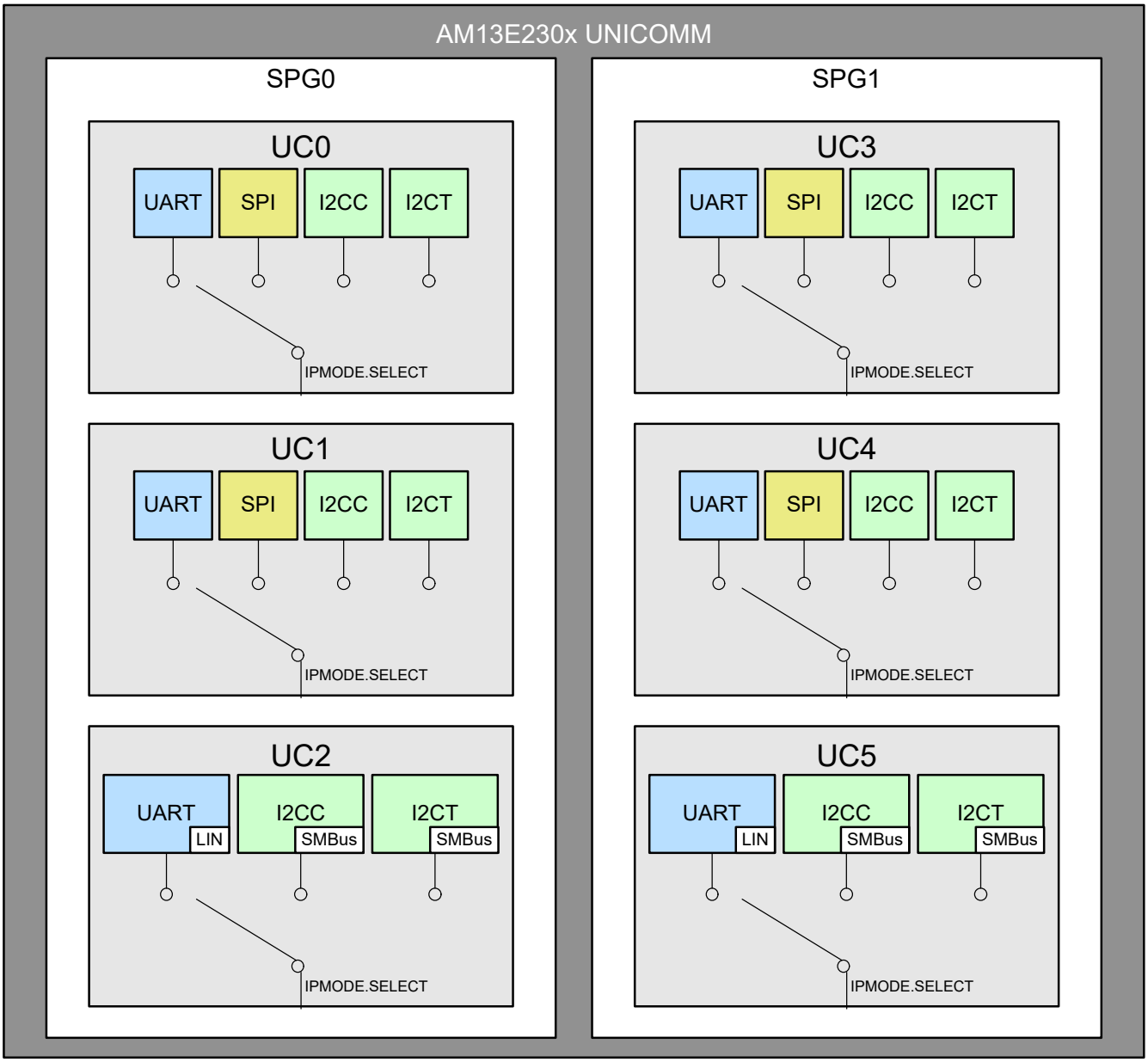


Figure 23-1. Unicomm Block Diagram

23.2 Unicomm Architecture

23.2.1 Scalable Peripheral Group (SPG) Configurations

The Scalable Peripheral Group (SPG) level configurations consist internal loopback operation and I2C pairing operation in compliance with the SMBUS protocol.

Note

Users not employing either of the loopback or I2C pairing test fetures can skip to Section 23.2.2 .

ADVANCE INFORMATION

23.2.1.1 Loopback Operation

UNICOMM supports a multiple self-test modes which internally connect the output UNICOMM signals to the input UNICOMM signals. This mode is helpful for initial code development and debugging. There are two types of internal loopback modes in UNICOMM, internal loopback and inter-module internal loopback. Figure 23-2 shows an example of these two loopback setups with UNICOMM UART, as well as an external loopback hardware setup example.

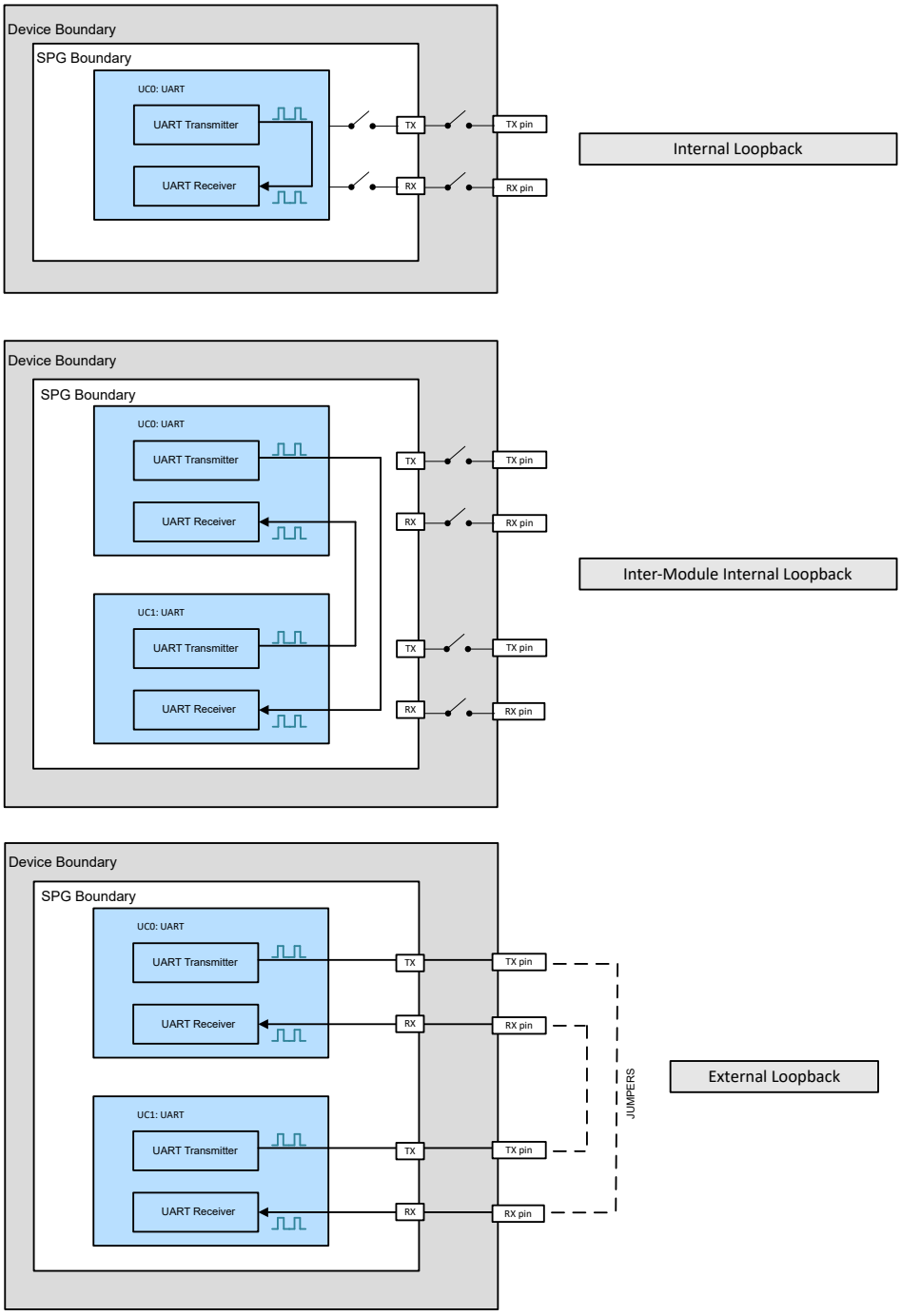


Figure 23-2. UNICOMM Loopback Types

Internal Loopback

For UNICOMM instances configured as UART or SPI, internal loopback within a *single* UNICOMM instance is available. The configuration for this mode is done in the UNICOMMUART_REGS and the UNICOMMSPI_REGS; see UART Internal Loopback and SPI Internal Loopback sections in the corresponding peripheral chapters.

- UART
  - UCi TX signal / UCi RX signal
- SPI
  - UCi SCLK signal / UCi SCLK signal
  - UCi SDO signal / UCi SDI signal
  - UCi SDI signal / UCi SDO signal

Inter-Module Internal Loopback

Inter-module internal loopback is available between *two different* UNICOMM instances configured for the same communication type that are in the same SPG group. The Figure 23-1 shows which UC instances belong in each SPG grouping. In this loopback method, the following signals are connected between two different UC instances, represented as "i" and "j" below. Inter-module internal loopback is configured in the LPBKx.CONTROLLER and LPBKx.PERIPHERAL fields of the LPBKx registers. Then, enable the loopback pair by setting the EN bit in the LPBKx register. There is one configurable loopback pair available for each UNICOMM group on this device.

- UART
  - UCi TX signal / UCj RX signal
- SPI
  - UCi CS0 signal / UCj CS0 signal
  - UCi SCLK signal / UCj SCLK signal
  - UCi SDO signal / UCj SDI signal
  - UCi SDI signal / UCj SDO signal
- I2C
  - UCi SCL signal / UCj SCL signal
  - UCi SDA signal / UCj SDA signal

External Loopback

Additionally, users can configure any two UNICOMM instances in the UART IPMODE and connect the signals externally to validate the hardware setup. This requires no specific loopback configurations in any of the UNICOMM registers since the loopback is done external to the device.

UART

- UCi TX signal / UCj RX signal

Note

When two UC instances are configured as an internal loopback pair (internal or inter-module internal), the associated signals are not brought out to device pins.

23.2.1.2 I2C Pairings

I2C paring supports specialized I2C modes where multiple controllers or a I2CC/I2CT pair are connected to a single external bus. Multi-controller configurations and protocols like SMBus/PMBus require this setup. I2C pairing is available between two UNICOMM instances when both of the below are true:

1. Both UNICOMM modules are configured for I2C communication in the IPMODE register. Controller-Controller and Controller-Target pairing are supported.
2. Both UNICOMM modules belong to the same SPG instance. Check the Figure 23-1 to see which SPG instance the UNICOMM instance belongs to.

I2C pairing connects the SDA line to the SDA line and the SCL line to the SCL line between two different UNICOMM instances inside the same scalable peripheral group (SPG). The I2C pairing control is set through the PAIRx registers.

To create an I2C pair within an SPG:

1. Select the UNICOMM to be used as the first I2C (I2CC) in the pair and find the index number of the SPG the UNICOMM instance belongs to.
2. Select the UNICOMM to be used as the second I2C (I2CC or I2CT) in the pair and verify that the index number of the SPG for the UNICOMM matches that of the SPG index from number 1 (for example: both from SPG0).
3. Write the local UNICOMM index (relative to each SPG) of the *first* instance to PAIRx.CTL.
    a. The local index of UC0 and UC3 is 0x0
    b. The local index of UC1 and UC4 is 0x1
    c. The local index of UC2 and UC5 is 0x2
4. Write the local UNICOMM index (relative to each SPG) of the *second* instance to PAIRx.TARGET.
    a. The local index of UC0 and UC3 is 0x0
    b. The local index of UC1 and UC4 is 0x1
    c. The local index of UC2 and UC5 is 0x2
5. Enable the loopback by writing a '1' to PAIRx.EN.

I2C pairing functions similar to a loopback, except for that the SDA/SCL signals propogate to the device pins. When an I2C pair is enabled, the SDA/SCL signals associated with the UNICOMM module defined in PAIR.TARGET propogates and the UNICOMM module defined in PAIR.CTL is driven IDLE.

23.2.2 FIFO Operation

Each UNICOMM peripheral has two dedicated 16-level FIFOs, one for receive operations and one for transmit operations. The FIFOs reduce service overhead by effeciently storing data to be sent while the transmitter is busy or storing received data to be processed while the CPU is busy. The TXDATA and RXDATA registers are used to interface with the transmit and receive FIFOs respectively.

The FIFO level for each UNICOMM instance is configured in the IFLS register within the peripheral-specific interface registers, even though the configuration options are the same across all IP modes.

23.2.2.1 Receive FIFO Levels

The configurable FIFO condition options for the Receive (RX) FIFO are:

- FIFO is full
- FIFO is at least 1/16 full - "not empty"
- FIFO is ¼ full
- FIFO is ½ full
- FIFO is ¾ full
- FIFO is 15/16 full - "almost full"
- FIFO is 1/16 full - "almost empty"

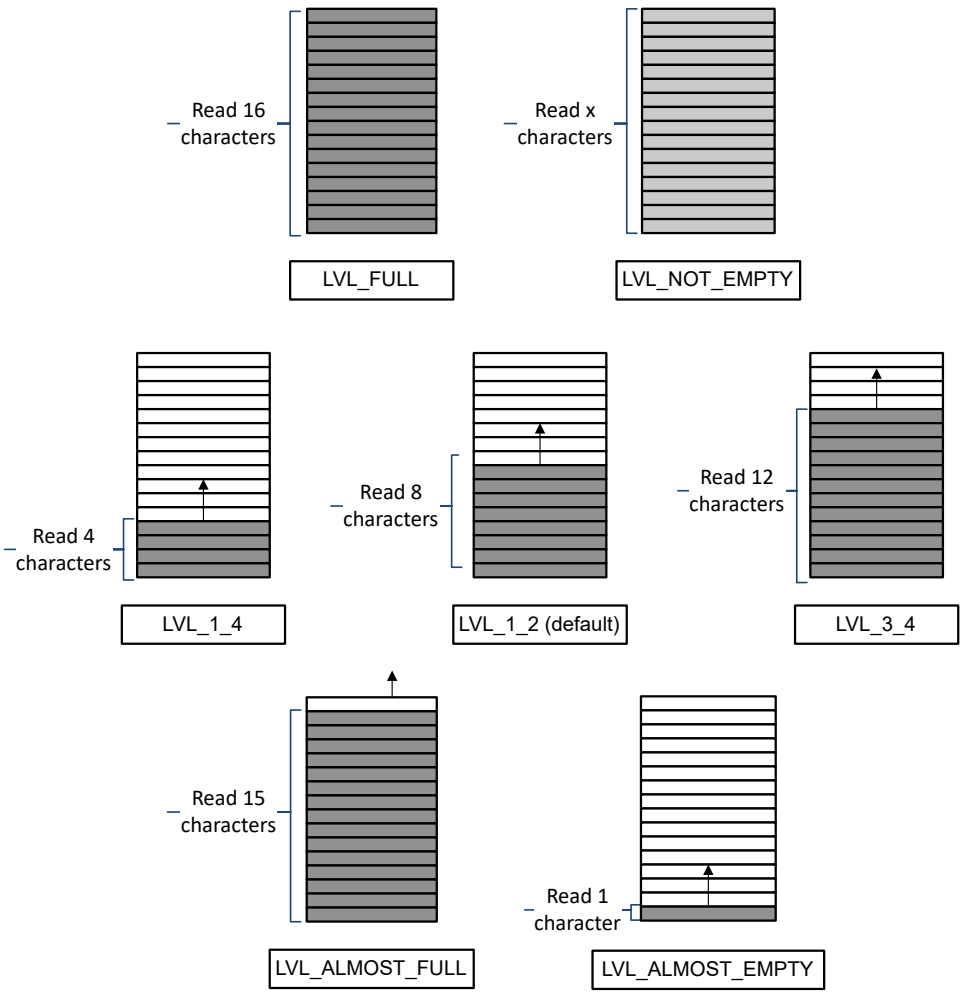


Figure 23-3. UNICOMM Receive FIFO Levels

These various conditions are typically used to generate a CPU_INT or DMA_TRIG_RX event where the CPU/RX subsequently reads a specific number of characters into memory from RXDATA. The number of characters read in each Interrupt Service Routine (ISR) equals how many are guaranteed present in the FIFO as indicated by the flag. Typically, the FIFO level is selected so that the CPU reads the largest number of characters possible in each ISR which is the highest possible even factor of the number of characters sent by the transmitting device in each frame. For the "not empty" condition, the application can read one element at a time while the condition is satisfied.

Note

Debugger reads of RXDATA have no effect on the contents in the RX FIFO.

23.2.2.2 Transmitter FIFO Levels

The configurable FIFO condition options for the Transmit (TX) FIFO are:

- FIFO is empty
- FIFO is at least 1/16 empty - "not full"
- FIFO is ¼ full
- FIFO is ½ full
- FIFO is ¾ full
- FIFO is 15/16 full - "almost full"
- FIFO is 1/16 full - "almost empty"

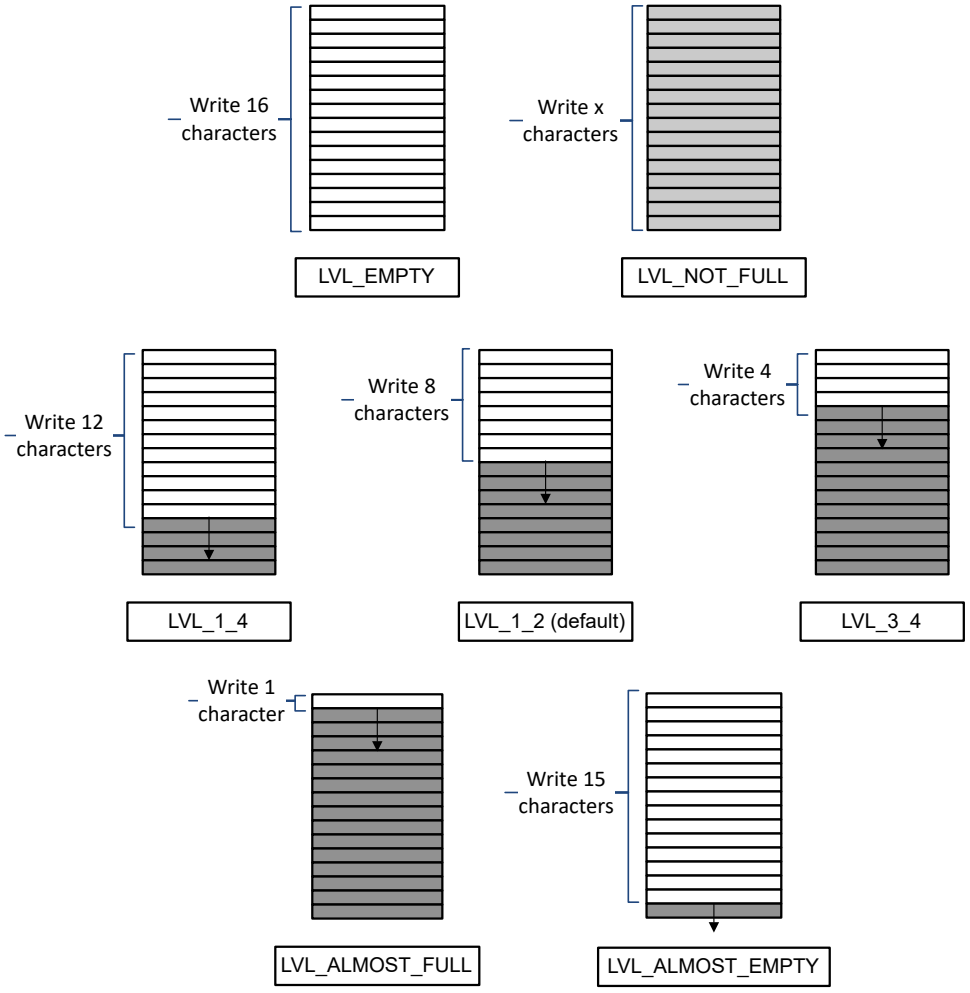


Figure 23-4. UNICOMM Transmit FIFO Levels

These various conditions are typically used to generate a CPU_INT or DMA_TRIG_TX event where the CPU/DMA subsequently writes a specific number of characters into the TXDATA register. The number of characters written in each Interrupt Service Routine (ISR) equals how many spots are guaranteed empty in the FIFO as indicated by the flag. Typically, the FIFO level is selected so that the CPU writes the largest number of characters possible in each ISR. Often this value is the highest possible even factor of the number of characters that are sent in each frame. For the "not full" condition, the application can write one element at a time while the condition is satisfied.

Note

Debugger writes of TXDATA will add elements to the TX FIFO the same as a CPU write.

23.2.2.3 Clearing FIFO Contents

Setting the IFLS.RXCLR and IFLS.TXCLR register bits clears the contents of each of the respective FIFOs. Follow the below sequence:

1. Write a 1 to the RXCLR or TXCLR bit in the IFLS peripheral-specific register.
2. Wait until the RXCLR or TXCLR status bit in the STAT register is set to 1.
3. Write a 0 to the RXCLR or TXCLR bit in the IFLS peripheral-specific register.

Note

Performing the above clear sequence before changing the FIFO level configuration in the RXIFSEL or TXIFSEL fields is recommended.

Note

Avoid performing step 1 repeatedly before full sequence is complete.

23.2.2.4 FIFO Status Flags

Each of the common available FIFO status bits are explained in the table below. All status bits are included in the below registers for each UNICOMM peripheral and can be used by application software to poll the FIFO status as needed.

- UART - STAT register
- SPI - STAT register
- I2CC - SR register
- I2CT - SR register

Table 23-2. UNICOMM FIFO Status Bits

Status Bit	Description
RXFE	This bit is set when the RX FIFO is empty.
RXFF	This bit is set when the RX FIFO is full.
RXCLR	This bit is set when a clear of the RX FIFO, started by setting the IFLS.RXCLR bit, is complete.
TXFE	This bit is set when TX FIFO is empty.
TXFF	This bit is set when TX FIFO is full.
TXCLR	This bit is set when a clear of the TX FIFO, started by setting the IFLS.TXCLR bit, is complete.

23.2.3 Interrupts

Each UNICOMM peripheral contains the six standard interrupts Section 13.1.1.1 : IIDX, IMASK, RIS, MIS, ISET and ICLR in the CPU_INT register group. These configuration, control, and status registers dictate how the UNICOMM module interfaces with the device interrupt controller module. Unique conditional flags are available in these registers to generate CPU interrupts for each specific peripheral mode (UART, SPI, I2CC, or I2CT). Each UNICOMM instance has one interrupt line connected to the interrupt controller.

23.2.3.1 Receive Interrupt Sequence

The receive interrupt flags for each peripheral specific UNICOMM instance are described below. These flags are raised when there is data present in the RX FIFO that is ready to be read by the CPU or DMA.

- UART - RXINT flag
- SPI - RX flag
- I2CC - RXTRG flag
- I2CT - RXTRG flag

Below is the high-level sequence for receiving a fixed amount of data with interrupts.

1. The RX FIFO contains greater than or equal to the number of bytes/words designated by the programmed trigger level in IFLS.RXIFSEL.
2. The interrupt flag for the configured peripheral mode is set and an interrupt is triggered on the CPU.
3. In the Interrupt Service Routine (ISR), the CPU reads the fixed amount of data from the RXDATA register (repeated reads). This data is then read from the RX FIFO automatically by hardware.

Depending on the peripheral mode configured, additional steps before and/or after this sequence may be required to receive the data using the corresponding protocol. See the peripheral-specific UNICOMM chapters for more details about the software steps needed for each IP.

23.2.3.2 Transmit Interrupt Sequence

The transmit interrupt flags for each peripheral specific UNICOMM instance are described below. These flags are raised when there is space present in the TX FIFO for the CPU or DMA to write data.

- UART - TXINT flag
- SPI - TX flag
- I2CC - TXTRG flag
- I2CT - TXTRG flag

Below is the high-level sequence for transmitting a fixed amount of data with interrupts.

1. The TX FIFO contains less than or equal to the number of bytes/words designated by the programmed trigger level in IFLS.TXIFSEL.
2. The interrupt flag for the configured peripheral mode is set and an interrupt is triggered on the CPU.
3. In the Interrupt Service Routine (ISR), the CPU writes the fixed amount of data to the TXDATA register. This data is then written to the TX FIFO automatically by hardware.

Depending on the peripheral mode configured, additional steps before and/or after this sequence may be required to transmit the data using the corresponding protocol. See the peripheral-specific UNICOMM chapters for more details about the software steps needed for each IP.

23.2.4 DMA Operation

Each UNICOMM peripheral has access to the DMA module on the device and can be configured to trigger DMA channels. From the UNICOMM point-of-view, DMA triggers are setup through the registers in the DMA_TRIG_RX and DMA_TRIG_TX register groups. Different conditionals are available to trigger the DMA_TRIG_RX line and the DMA_TRIG_TX line depending on the IPMODE configured. See the below sections from the peripheral-specific chapters for the options available.

- UART mode DMA conditionals: UART DMA Event Support
- SPI mode DMA conditionals: SPI DMA Event Support
- I2CC/I2CT mode DMA conditionals: I2C DMA Event Support

Note

Some modules only have one available conditional per DMA line.

See Section 9.2.3 for how to configure one of the UNICOMM signals as the trigger source for a DMA channel in the DMA registers.

23.3 High-Level Initialization

Below are the generic intialization steps to configure a UNICOMM instance and the recommended order for software to execute these steps. Peripheral specific initializations are listed in the respective sections.

1. Put the UNICOMM instance into reset by setting the RSCTL.RESETASSERT register bit from Section 23.6 .
2. Enable power to the UNICOMM instance by writing "0h26" to the KEY field and setting the EN bit for the PWREN register in the UNICOMM_REGS registers. Wait four or more cycles after this is set before writing to the individual peripheral registers.
3. Configure the peripheral mode for the UCx instance in the IPMODE.SELECT field from the UNICOMM_REGS registers to operate as a UART, SPI, I2C Controller, or I2C Target.
4. Enable the pin functionality for each GPIO being used in the peripheral configuration in the IOMUX_REGS registers. See the Chapter 14 section for more information.

Note

IOMUX configurations must be done after the peripheral mode (IPMODE.SELECT) is configured (previous step).

5. (Optionally) Configure the inter-module internal loopback/pairing settings in the SPG_REGS registers between two UCx modules in the same SPG. See Figure 23-1 to determine which UCx pairings are available on this device.
   a. Configure the loopback pairs (maximum of one pair per SPG module) in the LPBKx register field.
   b. Configure the I2C pairs (maximum of one pair per SPG module) in the PAIRx register field.
6. Configure the below settings in the UNICOMMUART_REGS / UNICOMMSPI_REGS / UNICOMMI2CC_REGS / UNICOMMI2CT_REGS registers .
   a. To enable clocking of the UNICOMM module, configure the clock source in the CLKSEL register to be MCLKDIV2, the functional clock of UNICOMM.
   b. Configure the clock divider for the UNICOMM instance's peripheral-specific functional clock in the CLKDIV.RATIO register field. This decides how much to divide down the overall UNICOMM clock source for each specific UNICOMM instance clock.
   c. Perform peripheral-specific initializations (example: SPI peripheral vs. controller mode, UART baud rate etc.) Follow the detailed steps in in the UART Initialization, SPI Initialization, I2C Controller Initialization, and I2C Target Initialization chapters.

23.4 Enables & Resets

Each UNICOMM instance (UCx) has a dedicated reset and enable bitfield in the UNICOMM_REGS register group.

The UNICOMM instance can be reset by writing a 1 to the RSTCTL.RESETASSERT bit. Writing this field resets the internal state of the UCx module, all configuration registers for the UC instance and the UC peripheral, and clears all of the FIFOs. The STAT.RESETKY bit is set once the UCx module has undergone a reset. To clear the RESETKY bit, write to RSTCTL.RESETSTKYCLR bit and the RESETKY bit won't be set until the next reset is performed.

A UNICOMM instance can be enabled at the UCx level by setting the PWREN.ENABLE bit. Clocking of the UC peripheral-specific registers is not enabled until this PWREN.ENABLE bit is set. If no IPMODE has been selected, a write to these registers has no effect.

Software Reset Considerations

A software reset can be executed with setting the RESETASSERT together with the KEY in the RSTCTL register. An ongoing transfer is terminated immediately and can leave the software in an undefined state. Therefore, before requesting a reset, an ongoing transfer must be terminated.

Hardware Reset Considerations

A hardware reset also initializes the IO configuration. This sets the IOs to a high impedance state. If pull-up resistors are connected externally (for example in the case of UNICOMM-I2C), the IOs are pulled high on reset. If there are no pull-up resistors connected externally, the IOs float after a UNICOMM reset.

23.5 Suspending Communication

Each UNICOMM peripheral contains a SUSPEND bitfield, see the below registers for each module. When this bit is set, the module returns to an idle state and stops further communication on the external pins. The transmit line/lines are driven to the idle state and any further toggles on the receives line/lines are not processed. To suspend the module, follow the below steps.

1. Check that the UNICOMM module is in an IDLE state
   a. UNICOMM-UART
      i. STAT.BUSY needs to read '0'.
      ii. In LIN mode, if transmitting, the checksum needs to be loaded into the TX FIFO.

iii. In LIN mode, if receiving, the checksum needs to be received from the RX FIFO.
iv. In idleline multiprocessor mode, if transmitting, all address and data bytes for the current frame need to be loaded into the TX FIFO
v. In idleline multiprocessor mode, if receiving, SR.IDLE needs to read '1'.
vi. If using RTS/CTS flow control, if transmitting, pull nRTS line high after TX FIFO is empty.
vii. If using RTS/CTS flow control, if receiving, pull nCTS line high after receiving the current byte.

b. UNICOMM-SPI
   - In Controller mode, CS needs to be idle.
   - In Peripheral mode, CS needs to read back as idle.

c. UNICOMM-I2CC
   - SR.BUSY needs to read '0'.
   - If transmitting, the receiving target must send a NACK
   - If receiving, NACK must be sent on the line
   - In multi-controller mode, STOP condition needs to be sent on the line

d. UNICOMM-I2CT
   - SR.BUSY needs to read '0'.
   - If transmitting, receiving controller must send a NACK
   - If receiving and clock stretching is enabled, NACK must be sent on the line
   - If receiving and clock stetching is disabled, transmitting controller must send a STOP or address a different target

2. Set the SUSPEND bit
   a. UNICOMM-UART: LCRH.SUSPEND=1
   b. UNICOMM-SPI: CTL1.SUSPEND=1
   c. UNICOMM-I2CC: CTR.SUSPEND=1
   d. UNICOMM-I2CT: CTR.SUSPEND=1
3. Clear the RX FIFO
4. Disable the module
   a. UNICOMM-UART: CTL0.ENABLE=0
   b. UNICOMM-SPI: CTL1.ENABLE=0
   c. UNICOMM-I2CC: CTR.ENABLE=0
   d. UNICOMM-I2CT: CR.ENABLE=0

Note

Before re-enabling the UNICOMM module, the SUSPEND bit needs to be cleared.

23.6 UNICOMM Registers

This Section describes the UNICOMM Registers.

23.6.1 UNICOMM Base Address Table

Table 23-3. UNICOMM Base Address Table

Bit Field Name		DriverLib Name	Base Address
Instance	Structure		
Unicomm0Regs	UNICOMM_REGS	UNICOMM0	0x4063_0000
Unicomm1Regs	UNICOMM_REGS	UNICOMM1	0x4063_2000
Unicomm2Regs	UNICOMM_REGS	UNICOMM2	0x4063_4000
Unicomm3Regs	UNICOMM_REGS	UNICOMM3	0x4067_0000
Unicomm4Regs	UNICOMM_REGS	UNICOMM4	0x4067_2000
Unicomm5Regs	UNICOMM_REGS	UNICOMM5	0x4067_4000

23.6.2 UNICOMM_REGS Registers

Table 23-4 lists the memory-mapped registers for the UNICOMM_REGS registers. All register offset addresses not listed in Table 23-4 should be considered as reserved locations and the register contents should not be modified.

Table 23-4. UNICOMM_REGS Registers

Offset	Acronym	Register Name	Section
800h	PWREN	Power enable	Go
804h	RSTCTL	Reset Control	Go
814h	STAT	Status Register	Go
1100h	IPMODE	Mode Selection Register	Go

Complex bit access types are encoded to fit into small table cells. Table 23-5 shows the codes that are used for access types in this section.

Table 23-5. UNICOMM_REGS Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
Write Type		
W	W	Write
WK	W K	Write Write protected by a key
Reset or Default Value		
-n		Value after reset or the default value

ADVANCE INFORMATION

1 PWREN Register (Offset = 800h) [Reset = 00000000h]

PWREN is shown in Figure 23-5 and described in Table 23-6.

Return to the Summary Table.

Register to control the power state

Figure 23-5. PWREN Register

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

Table 23-7. PWREN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	KEY	W	0h	KEY to allow Power State Change 26h = KEY to allow write access to this register
23-1	RESERVED	R/W	0h	
0	ENABLE	R/WK	0h	Enable the power [EXT_GPRCM.GPRCM.PWREN.KEY] must be set to 26h to write to this bit. 0h = Disable Power 1h = Enable Power

ADVANCE INFORMATION

2 RSTCTL Register (Offset = 804h) [Reset = 00000000h]

RSTCTL is shown in Figure 23-6 and described in Table 23-7.

Return to the Summary Table.

Register to control reset assertion and de-assertion

Figure 23-6. RSTCTL Register

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

Table 23-9. RSTCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	KEY	W	0h	Unlock key B1h = KEY to allow write access to this register
23-2	RESERVED	W	0h	
1	RESETSTKYCLR	WK	0h	Clear the RESETSTKY bit in the STAT register [EXT_GPRCM.GPRCM.RSTCTL.KEY] must be set to B1h to write to this bit. 0h = Writing 0 has no effect 1h = Clear reset sticky bit
0	RESETASSERT	WK	0h	Assert reset to the peripheral [EXT_GPRCM.GPRCM.RSTCTL.KEY] must be set to B1h to write to this bit. 0h = Writing 0 has no effect 1h = Assert reset

ADVANCE INFORMATION

3 STAT Register (Offset = 814h) [Reset = 00000000h]

STAT is shown in Figure 23-7 and described in Table 23-8.

Return to the Summary Table.

peripheral enable and reset status

Figure 23-7. STAT Register

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

Table 23-11. STAT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-17	RESERVED	R	0h	
16	RESETSTKY	R	0h	This bit indicates, if the peripheral was reset, since this bit was cleared by RESETSTKYCLR in the RSTCTL register 0h = The peripheral has not been reset since this bit was last cleared by RESETSTKYCLR in the RSTCTL register 1h = The peripheral was reset since the last bit clear
15-0	RESERVED	R	0h	

4 IPMODE Register (Offset = 1100h) [Reset = 00000000h]

IPMODE is shown in Figure 23-8 and described in Table 23-9.

Return to the Summary Table.

Used for selecting mode - UART / I2C Controller / I2C Target / SPI

Figure 23-8. IPMODE Register

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

Table 23-13. IPMODE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R/W	0h	
1-0	SELECT	R/W	0h	Free run control 0h = UNICOMM is in UART Mode 1h = UNICOMM is in SPI Mode. Selection only valid for UC0, UC1, UC3, and UC4. 2h = UNICOMM is in I2C Controller Mode 3h = UNICOMM is in I2C Target Mode

23.7 SPG Registers

This Section describes the SPG Registers.

23.7.1 SPG Base Address Table

Table 23-14. SPG Base Address Table

Bit Field Name		DriverLib Name	Base Address
Instance	Structure		
Spg0Regs	SPG_REGS	SPG0	0x4063_F000
Spg1Regs	SPG_REGS	SPG1	0x4067_F000

ADVANCE INFORMATION

23.7.2 SPG_REGS Registers

Table 23-11 lists the memory-mapped registers for the SPG_REGS registers. All register offset addresses not listed in Table 23-11 should be considered as reserved locations and the register contents should not be modified.

Table 23-15. SPG_REGS Registers

Offset	Acronym	Register Name	Section
120h	LPBK0	Loopback between SPG instances User has to ensure that the SPGs looped back have the same underlying IP instantiated; i.e. both UART, both SPI, both I2C, etc.,	Go
1C0h	PAIR0	Pairing of I2C for MULTI-MASTER/SMBUS applications	Go
C00h	TEST0	Test 0 register	Go

Complex bit access types are encoded to fit into small table cells. Table 23-12 shows the codes that are used for access types in this section.

Table 23-16. SPG_REGS Access Type Codes

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

ADVANCE INFORMATION

1 LPBK0 Register (Offset = 120h) [Reset = 00000000h]

LPBK0 is shown in Figure 23-9 and described in Table 23-13.

Return to the Summary Table.

Loopback between SPG instances User has to ensure that the SPGs looped back have the same underlying IP instantiated
i.e. both UART, both SPI, both I2C, etc.,

Figure 23-9. LPBK0 Register

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

Table 23-18. LPBK0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R/W	0h	
15-12	PERIPHERAL	R/W	0h	Select SPG peripheral for loopback. Signals invovled in loopback based on interface chosen 1. SPI : CS, SCLK, SDI, SDO 2. I2C : SCL, SDA 3. UART : TX, RX 0h = SPG responder select for loopback. 1h = SPG responder select for loopback. 2h = SPG responder select for loopback. 3h = SPG responder select for loopback. 4h = SPG responder select for loopback. 5h = SPG responder select for loopback. 6h = SPG responder select for loopback. 7h = SPG responder select for loopback. 8h = SPG responder select for loopback. 9h = SPG responder select for loopback. Ah = SPG responder select for loopback. Bh = SPG responder select for loopback. Ch = SPG responder select for loopback. Dh = SPG responder select for loopback. Eh = SPG responder select for loopback. Fh = SPG responder select for loopback.
11-8	CONTROLLER	R/W	0h	Select SPG controller for loopback. Signals invovled in loopback based on interface chosen 1. SPI : CS, SCLK, SDI, SDO 2. I2C : SCL, SDA 3. UART : TX, RX 0h = SPG commander select for loopback. 1h = SPG commander select for loopback. 2h = SPG commander select for loopback. 3h = SPG commander select for loopback. 4h = SPG commander select for loopback. 5h = SPG commander select for loopback. 6h = SPG commander select for loopback. 7h = SPG commander select for loopback. 8h = SPG commander select for loopback. 9h = SPG commander select for loopback. Ah = SPG commander select for loopback. Bh = SPG commander select for loopback. Ch = SPG commander select for loopback. Dh = SPG commander select for loopback. Eh = SPG commander select for loopback. Fh = SPG commander select for loopback.
7-1	RESERVED	R/W	0h	

Table 23-18. LPBK0 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
0	EN	R/W	0h	Loopback enable 0h = Does not select this clock as a source 1h = Select this clock as a source

2 PAIR0 Register (Offset = 1C0h) [Reset = 00000000h]

PAIR0 is shown in Figure 23-10 and described in Table 23-14.

Return to the Summary Table.

Pairing of I2C for MULTI-MASTER/SMBUS applications

Figure 23-10. PAIR0 Register

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

Table 23-20. PAIR0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R/W	0h	
15-12	TARGET	R/W	0h	UNICOMM instance for loopback. This instance will be treated as the target 0h = SPG responder select for loopback. 1h = SPG responder select for loopback. 2h = SPG responder select for loopback. 3h = SPG responder select for loopback. 4h = SPG responder select for loopback. 5h = SPG responder select for loopback. 6h = SPG responder select for loopback. 7h = SPG responder select for loopback. 8h = SPG responder select for loopback. 9h = SPG responder select for loopback. Ah = SPG responder select for loopback. Bh = SPG responder select for loopback. Ch = SPG responder select for loopback. Dh = SPG responder select for loopback. Eh = SPG responder select for loopback. Fh = SPG responder select for loopback.
11-8	CTL	R/W	0h	UNICOMM instance for loopback. This instance will be treated as the controller. 0h = SPG commander select for loopback. 1h = SPG commander select for loopback. 2h = SPG commander select for loopback. 3h = SPG commander select for loopback. 4h = SPG commander select for loopback. 5h = SPG commander select for loopback. 6h = SPG commander select for loopback. 7h = SPG commander select for loopback. 8h = SPG commander select for loopback. 9h = SPG commander select for loopback. Ah = SPG commander select for loopback. Bh = SPG commander select for loopback. Ch = SPG commander select for loopback. Dh = SPG commander select for loopback. Eh = SPG commander select for loopback. Fh = SPG commander select for loopback.
7-1	RESERVED	R/W	0h	
0	EN	R/W	0h	Enable 0h = Does not select this clock as a source 1h = Select this clock as a source

3 TEST0 Register (Offset = C00h) [Reset = 00000000h]

TEST0 is shown in Figure 23-11 and described in Table 23-15.

Return to the Summary Table.

Test 0 register.

Figure 23-11. TEST0 Register

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

Table 23-22. TEST0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R/W	0h	
3-0	DTB_MUX_SEL	R/W	0h	This bit field is used to select DTB mux digital output signals.

Chapter 24

Universal Asynchronous Receiver/Transmitter (UART)

The functionality of a UNICOMM module when configured to operate as a UART is described in this section. The protocol mode of a given UNICOMM instance is decided by the programmed SELECT field of the IPMODE register. If a UNICOMM instance is configured for another peripheral mode, the UART functionality on that UNICOMM instance is disabled and unusable.

ADVANCE INFORMATION

24.1 Overview

24.1.1 Purpose of the Peripheral

The Universal Asynchronous Receiver/Transmitter (UART) interface is a single-ended, asynchronous serial communication port that transmits (TX) and receives (RX) a programmable length serial bit stream into and out of the device at a programmable baud rate. The UNICOMM UART module can be used to communicate with another device supporting the standard UART protocol.

The **Basic** UNICOMM UART module features include additional support for:

- ISO7816 Smart Card

The **Basic+** UNICOMM UART module features include additional support for:

- LIN (Local Interconnect Network)

This device has both Basic and Basic+ implementations of the UART UNICOMM module on different UNICOMM instances. The *Available UNICOMM Configurations Per Instance* table lists which UNICOMM UART mode is present for each UCx instance inside each SPG. This document outlines the functionality of each feature.

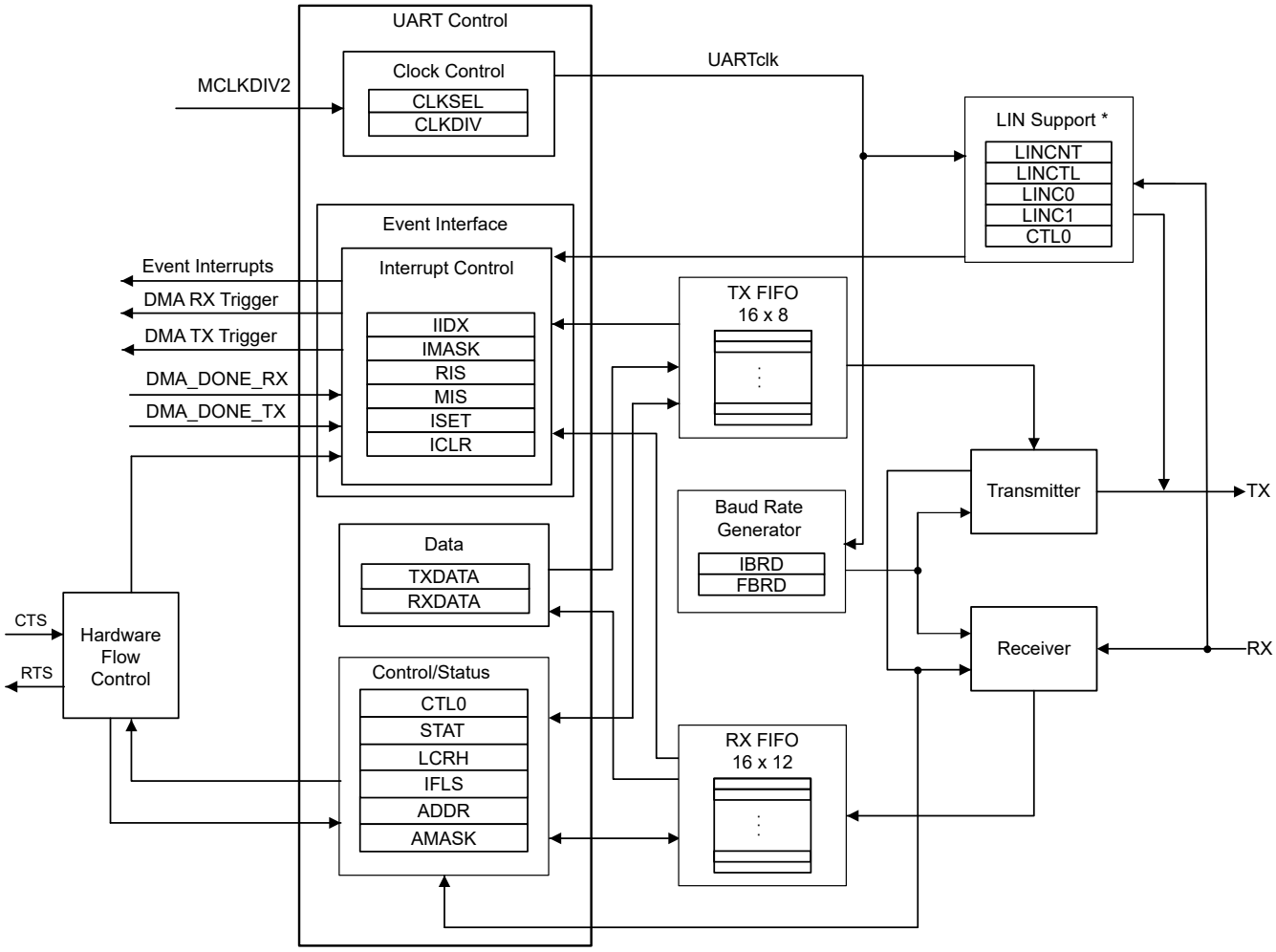
24.1.2 Features

The UNICOMM-UART peripheral mode of the UNICOMM module includes the following features:

- Fully programmable serial interface:
  - 5, 6, 7 or 8 data bits
  - Even, odd, stick, or no-parity bit generation and detection
  - 1 or 2 stop bit generation
  - LSB-first or MSB-first data transmit and receive
  - Line-break detection
  - Programmable baud-rate generation with oversampling by 16, 8 or 3
- Separate transmit (TX) and receive (RX) buffers/first-in first-out buffers (FIFOs)
- Direct Memory Access (DMA) support
- Available Interrupts:
  - Transmit (TX) and receive (RX) FIFO levels
  - End of transmission (EOT)
  - Receive timeout
  - Line timeout
  - Break, frame, parity, noise, and overrun error
  - Receive positive and negative edge conditions
  - Address matching
  - CTS condition
  - LIN counter capture, match, and overflow events
  - DMA done conditions
- Local Interconnect Network (LIN) hardware support (Basic+ instances)
- Hardware Flow Control (CTS/RTS)
- RS485 Flow Control support
- Idle-line Multiprocessor Mode
- 9-Bit UART Mode
- ISO7816 Smartcard support (Basic instances)
- Internal loopback test mode

To support UART protocol, a UNICOMM module must be configured as a UART in the UNICOMM top level IPMODE register.

24.1.3 Functional Block Diagram



*Only Basic+ UNICOMM UART instances have built-in LIN support

Figure 24-1. UART Functional Block Diagram

24.2 Peripheral Functional Description

24.2.1 Clock Control

The UART internal functional clock is selected and divided from the functional clock of the IP.

- Use the CLKSEL register to enable MCLKDIV2 as the functional clock source for the UNICOMM UART module.
- Use the CLKDIV register to select the divide ratio of the UART functional clock. The options available are divide by 1 through divide by 64.

The selected source clock is always available and the frequency depends on the power mode, for more information, see the *Clock Module (CKM)* section. After enabling the UART module by setting the CTL0.ENABLE bit, the module is ready to start receiving and transmitting data.

24.2.2 General Architecture and Protocol

The UART protocol transmits and receives characters at a programmable bit rate, called a baud rate, shared between the two communicating devices. This is done asychronously, meaning there is no connected clock pin

for this protocol. The configured baud rate is generated by each device's functional clock and is used by both the UNICOMM UART transmitter and receiver submodules.

In general, configuration registers can only be programmed when the UNICOMM UART module is disabled (the ENABLE bit in the CTL0 register is cleared). One exception is the baud-rate divisor registers (IBRD and FBRD), which can be modified without disabling the UART. If the UNICOMM UART is disabled during an ongoing transmit or receive operation, the current transaction completes before the UART halts operation.

24.2.2.1 Signal Descriptions

UART communications require two pins: Receive Data (RX) and Transmit Data (TX):

- RX (Receive Data): The serial data input. Oversampling techniques are used on the receive signal for accuracy of incoming data.
- TX (Transmit Data): The serial data output. When the transmitter is enabled and no data needs to be transmitted, the TX pin is held high (idle). In some bidirectional protocols like ISO7816 Smart card, this pin is also used to receive data.

In hardware flow control mode, the following pins are also used.

- CTS (Clear To Send): The flow control input. When driven high by an external signal, this signal blocks the data transmission at the end of the current transfer.
- RTS (Request To Send): The flow control output. When held low, this signal indicates that the UART is ready to receive data.

24.2.2.2 Transmit and Receive Logic

The transmit logic performs parallel-to-serial conversion on the data written to the TXDATA register. The UNICOMM-UART transmitter automatically buffers the data written to the TXDATA register into the TX FIFO. The control logic outputs the serial bit stream beginning with a start bit and followed by the data bits (LSB first), parity bit, and the stop bits according to the programmed configuration.

The receive logic performs serial-to-parallel conversion on the received bit stream after a valid start pulse has been detected. Overrun, parity, frame error checking, and line-break detection are also performed, and this status accompanies the data that is written to the receive FIFO. Each RX FIFO element is 12-bits wide to include the data bits and all status information.

In the UNICOMMUART Registers, the CTL0.ENABLE bit is used to enable and disable the UNICOMM-UART module, the CTL0.TXE and CTL0.RXE bits are used to enable the UART transmitter and UART receiver, the LCRH.WLEN bit is used to configure the number of data bits transmitted or received in each frame, the LCRH.PEN is used to enable parity mode, and the LCRH.STP2 is used to send two stop bits (rather than the default one bit).

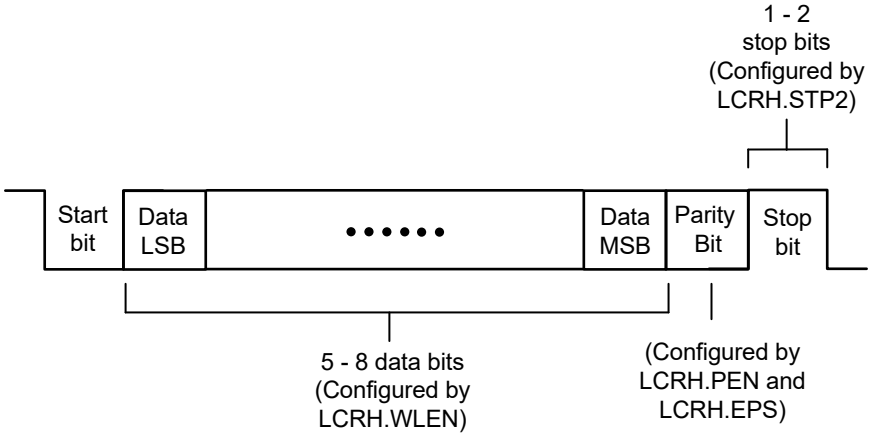


Figure 24-2. UART Character Frame

ADVANCE INFORMATION

24.2.2.3 Bit Sampling

UNICOMM UART supports oversampling at 3x, 8x, and 16x the UART bus baud-rate.

- By default, CTL0.HSE is set to 0, i.e.16X oversampling is selected. The center sample (sample 8) is used for sampling data on RX line.
- CTL0.HSE set to '1', selects 8X oversampling. The center sample (sample 4) is used for sampling data on RX line.
- CTL0.HSE set to '2' selects 3X oversampling. The center sample (sample 2) is used for sampling data on RX line.

The aforementioned scenarios assume an IBRD =1 and FBRD =0.

Depending on the application:

- Select oversampling by 3 or 8 to achieve higher speed with UARTclk/8 or UARTclk/3. In this case the receiver tolerance to clock deviation is reduced.
- Select oversampling by 16 to increase the tolerance of the receiver to clock deviations. The maximum speed is limited to UARTclk/16.

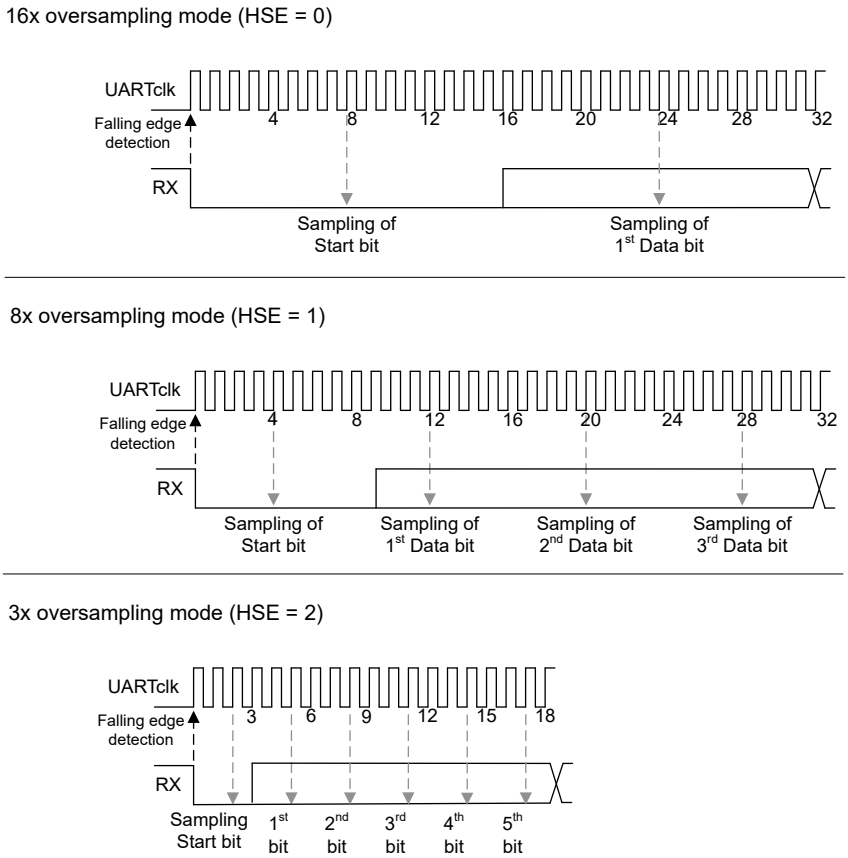


Figure 24-3. UART Oversampling mode

24.2.2.4 Baud Rate Generation

The baud-rate divisor is a 22-bit number consisting of a 16-bit integer and a 6-bit fractional part. The number formed by these two values is used by the baud-rate generator to determine the bit sample period. Having a fractional baud-rate divisor allows the UART to generate all of the standard baud-rates very accurately. An example calculation is demonstrated below, however tools like the TI System Configurator can perform these calculations automatically.

ADVANCE INFORMATION

The 16-bit integer is loaded through the UART Integer Baud-Rate Divisor IBRD register and the 6-bit fractional part is loaded with the UART Fractional Baud-Rate Divisor FBRD register.

The baud-rate divisor can be calculated by using the following formula:

BRD = UART Clock / (Oversampling x Baud rate) (36)

UART Clock (UARTclk) is the clock output of the UART clock control logic, configured by CLKSEL and CLKDIV. Oversampling is selected by the HSE bit in the CTL0 register and can be configured to 16, 8 or 3.

- IBRD = INT(BRD), integer portion of the BRD
- FBRD = BRD % 64 , fractional portion of the BRD

The integer portion of the BRD is loaded into IBRD register. The 6-bit fractional number must be loaded into the FBRD register.

Note

When IBRD = 0, FBRD is ignored and no data gets transmitted or received by the UART. Similarly, when IBRD = 65535 (that is 0xFFFF), then FBRD must not be greater than zero. The ongoing transmission or reception is aborted if these requirements are not met.

The following example shows a simple method to calculate IBRD.DIVINT and FBRD.DIVFRAC for a baud rate of 19200 bit/s:

UART Clock = MCLKDIV2 / CLKDIV.RATIO = 40 MHz
CTL0.HSE Oversampling = 16
Baud Rate = 19200 bit/s

$$BRD = \frac{UARTclk}{Oversampling \times Baud\ Rate} = \frac{40MHz}{16 \times 19200\ bit/s} = 130.2083333$$

IBRD.DIVINT= 130 (=82h)

FBRRD.DIFRAC
= INT((.2083333 x 64) + 0.5)
= INT(13.833333)
= 13d (= Dh)

Note: The adder '+0.5' ensures rounding to the closest integer value to keep the rounding error as small as possible

Figure 24-4. Baud Rate Configuration

When updating the baud-rate divisor (IBRD or IFRD), the LCRH register must also be written, so any changes to the baud rate divisor must be followed by a write to the LCRH register for the changes to take effect. The contents of the IBRD and FBRD registers are not updated until transmission or reception of the current character is complete.

24.2.2.5 Data Transmission

Data received or transmitted is stored in two 16-element FIFOs, where the receive FIFO has an extra four bits per character for status information.

Transmit data:

For transmission, data is written into the TXDATA register. If the UART is enabled, a write to TXDATA causes a data frame to start transmitting with the parameters indicated in the LCRH register. Data continues to be transmitted until there is no data left in the transmit FIFO. The BUSY bit in the STAT register is asserted as soon as data is written to the transmit FIFO (that is, if the FIFO is non-empty) and remains asserted while data is being transmitted. The BUSY bit goes low only when the transmit FIFO is empty, and the last character has been

ADVANCE INFORMATION

transmitted to the shift register, including the stop bits. STAT.BUSY also is set during the generation of a BREAK signal.

Receive data:

When the UART receiver is in an idle state and data input transitions to a low level (i.e. START condition detect), the receive counter starts running and data is sampled as per the oversampling settings in CTL0.HSE, except for the START/STOP bits. During the START bit, the receiver checks that every sampled value reads '0', and if not, the receiver state machine transitions back to the IDLE state. During the STOP bit/s, the receiver checks that every sampled value reads '1'. If not, the framing error flag (RXDATA.FRMERR) is set. When a full word is received, data is stored in the receive FIFO along with any associated error bits. The STAT.BUSY flag is set to high when a START condition is detected and set to low after a STOP bit has been received (i.e. after the entire duration of STOP bit).

24.2.2.6 Error and Status

For received data, the data byte and the 4-bit status (break, frame, parity, and overrun) is pushed onto the 12-bit wide receive buffer/FIFO. The error and status can be can be retrieved by reading RXDATA register as showing in Table 24-1. Each of the four below conditions can also set the interrupt flag RXINT if the corresponding interrupt bit is enabled in the IMASK register.

Table 24-1. UART Error and Conditions

Error Condition(1)	Bit Field	Description
Framing error	FRMERR	A framing error occurs when a low stop bit is detected. When two stop bits are used, both stop bits are checked for a framing error. When a framing error is detected, the FRMERR bit is set.
Parity error	PARERR	A parity error is a mismatch between the number of 1s in a character and the value of the parity bit. Address bits are included in the parity calculation if included in the character. When a parity error is detected, the PARERR bit is set.
Receive overrun	OVRERR	An overrun error occurs when a character is loaded into RXDATA/FIFO before the prior character has been read. When an overrun occurs, the OVRERR bit is set.
Break condition	BRKERR	A break is detected when the receive input is held low for longer than a full character transmission time, meaning the start bit, all received data, parity bit, and stop bit/bits are 0. When a break condition is detected, the BRKERR bit is set.

(1) Framing error and break condition are not set when LIN mode is enabled, this pattern is used to signal a Sync frame for LIN.

The UART module flag status can also be checked by reading the STAT register, as shown in Table 24-2.

Table 24-2. UART Flag Status

Bit Field	Description
BUSY	This bit is set as soon as the transmit FIFO becomes non-empty (regardless of whether UART is enabled). In IDLELINE mode (when the CTL0.MODE field is configured for IDLELINE), the BUSY signal also stays set during the idle time generation.
RXFE	This bit is set when the receive FIFO is empty.
RXFF	This bit is set when the receive FIFO is full.
TXFE	This bit is set when transmit FIFO is empty.
TXFF	This bit is set when transmit FIFO is full.
CTS	This bit is set when CTS signal is asserted (low) and cleared when CTS signal is not asserted (high).
IDLE	This bit is set when an IDLE line has been detected in idle-line multiprocessor mode. The IDLE bit is used as an address tag for each block of characters. In idle line multiprocessor format, this bit is set when a received character is an address.

24.2.2.7 DMA Operation

The UNICOMM-UART provides an interface to the DMA module with separate channels for transmit and receive. The DMA operation of the UART is enabled through the UART Event and DMA registers. When DMA operation

is enabled, the UART asserts a DMA request on the receive or transmit channel when the associated FIFO can transfer data. The DMA transfer requests are handled automatically by the DMA controller based on how the DMA channel is configured (burst size, transfer size, source address etc.).

For more information about the DMA module and DMA-specific configurations, see the Direct Memory Access (DMA) chapter.

- For the receive channel, a DMA transfer request is asserted when the amount of data in the receive FIFO is at or above the FIFO trigger level configured using the RXIFLSEL bit in IFLS register or if the receive timeout has triggered. In the receive timeout trigger case, the amount of data received up to that point is transferred.

Note

The DMA moves all data present in the RX FIFO once triggered, not just the amount configured by the RXIFLSEL field.

- For the transmit channel, a DMA transfer request is asserted whenever the transmit FIFO contains fewer characters than the FIFO trigger level configured using the TXIFLSEL bit in IFLS register.

24.2.2.8 Internal Loopback Operation

The UNICOMM-UART can be placed into an internal loopback mode for diagnostic or debug work by setting the LBE bit in the CTL0 register. In loopback mode, the UART operates with the following behavior:

- Data transmitted on the TX output is received on the RX input
- Data transmitted on the TX output is not propagated to the TX IO pin
- Data received on the RX IO pin is ignored

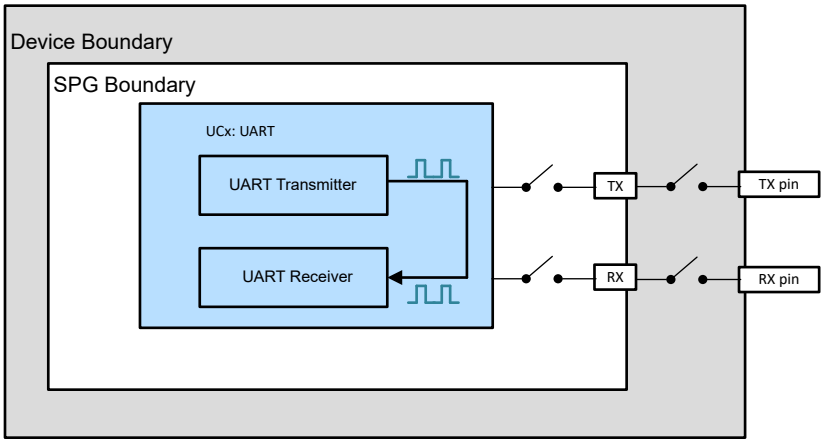


Figure 24-5. UART Internal Loopback

24.2.3 Additional Protocol and Feature Support

24.2.3.1 Local Interconnect Network (LIN) Support

LIN hardware support is only available for Basic+ UNICOMM-UART instances.

The Local Interconnect Network (LIN) protocol standard is based on the UART serial data link format. The LIN communication protocol is a single-commander/multiple-responder topology with message identification for multicast transmission between network nodes.

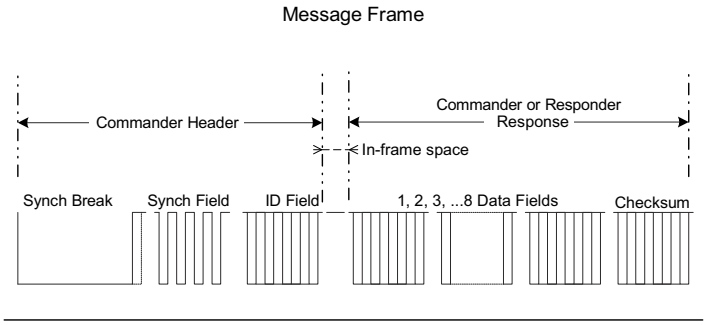


Figure 24-6. LIN Protocol Message Structure

The LIN commander issues a BREAK field, SYNC field, and PID (Protected Identifier) field at the start of every frame. The four LIN registers are used so that the LIN responder software driver can reasonably detect BREAK and SYNC fields and measure the necessary timing parameters to adjust the baud rate or detect an error.

See the *LIN Standards and Specifications* page for more information about the LIN protocol standards. The UNICOMM-UART module in LIN mode can operate as a LIN commander or a LIN responder when configured properly. To support the LIN protocol while reducing the number of interrupts needed in software, the following hardware enhancements are included in the UNICOMM-UART module in the Basic+ module variants:

- **LIN Control** - LINCTL
  - Configuration of LIN mode behavior.
- **LIN Counter** - LINCNT
  - 16-bit up-counter clocked by the UARTclk.
  - CPU_INT.IMASK.LINOVF flag - Interrupt capability on counter overflow.
- **LIN Capture 0** - LINC0
  - 16-bit Capture 0 with two configurable modes:
    - If LINCTL.LIN0CAP is set - Capture of the LINCNT value on RX pin falling edge. Interrupt capability on capture (CPU_INT.IMASK.LINC0).
    - If LINCTL.LIN0_MATCH is set - Compare of the LINCNT with interrupt capability on match (CPU_INT.IMASK.LINC0).
- **LIN Capture 1** - LINC1: 16-bit Capture 1.
  - Capture LINCNT value on RX pin rising edge. Interrupt capability on capture (CPU_INT.IMASK.LINC0).

Additionally, software can manually control the logic level output of the UNICOMM-UART TXD pin by writing the CTL0.TXD_OUT bit when the CTL0.TXD_OUT_EN is set to 1 with CTL0.TXE set to 0 (transmits disabled). This feature is also needed to achieve a LIN protocol on UNICOMM-UART.

Note

Full hardware support of the LIN protocol is not available in UNICOMM-UART. The above registers are used to aid in a software LIN implementation of the LIN-specific features added on top of the UART protocol.

24.2.3.1.1 LIN Commander Transmit

The below highlights how a UNICOMM-UART can implement each portion of the LIN Commander transmission using the registers available in the module.

Transmitting the Break Field

The break signal can be sent automatically by the UNICOMM-UART by setting the BRK bit in LCRH register for greater than 13*Tbits. This bit needs to be set before any data is written into the transmit data register TXDATA.

Transmitting the Sync Field

ADVANCE INFORMATION

The sync field can be sent by writing 0x55 to the TXDATA register to be transmitted using the standard UNICOMM-UART transmitter.

Transmitting the PID Field

Write the ID byte to the TXDATA register to be transmitted using the standard UNICOMM-UART transmitter. Make sure CTL0.TXE is set.

Transmitting the Data Fields

Write the data bytes to TXDATA register to be transmitted using the standard UNICOMM-UART transmitter. Make sure CTL0.TXE is set.

Transmitting the Checksum

Calculate and write the checksum byte to the TXDATA register to be transmitted using the standard UNICOMM-UART transmitter. Make sure CTL0.TXE is set.

24.2.3.1.2 LIN Responder Receive

Receiving the Break Field

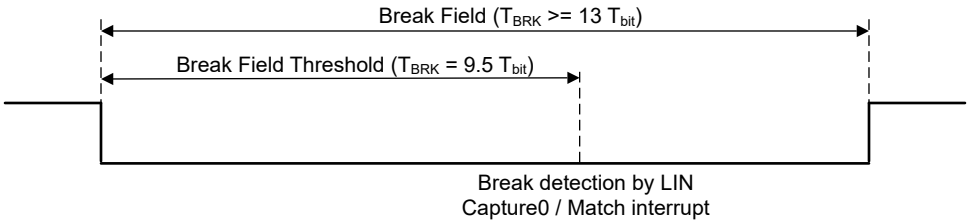


Figure 24-7. LIN Break Field Detection

Use the break field detection and compare modes to receive and validate the BREAK field of a LIN message. The following configurations are needed:

1. Initialize the LIN counter register to 0 (LINCNT = 0)
2. Enable counter compare match mode (LINCTL.LINC0_MATCH = 1)
3. Load LINC0 (Counter Capture 0 register) with the counter value corresponding to 9.5 x T_{bit} (refer to the *LIN Standards and Specifications* for details on this timing).
4. Enable the LINC0 match interrupt (CPU_INT.IMASK.LINC0 = 1)
5. Setup LIN count control (configure the LINCTL register):
   - Enable count while low signal on RXD pin (LINCTL.CNTRXLOW = 1)
   - Enable LIN counter clearing on RXD pin falling edge (LINCTL.ZERONE = 1)
   - Enable LIN counter (LINCTL.CTRENA = 1)

   Optional:
   - Enable the rising edge on UART TX signal interrupt (CPU_INT.IMASK.RXPE = 1), when the RXPE interrupt fires, the software can read the LINCNT directly to see the BREAK field timing.
   - Enable the LIN counter overflow interrupt (CPU_INT.IMASK.LINOVF = 1) to detect the BREAK field is too long and overflows 16-bit counter. The timeout can be calculated as $t_{Timeout}$= 2^{16}/ UARTclk.

Receiving the Sync Field

ADVANCE INFORMATION

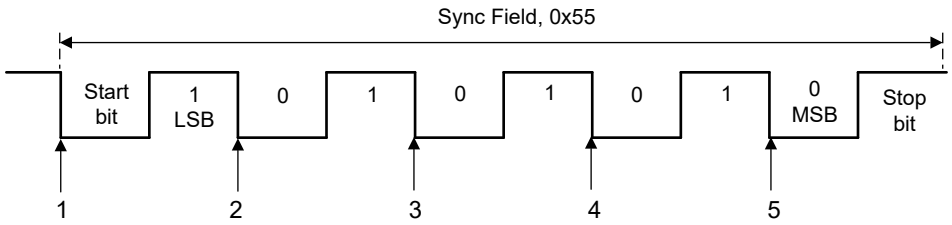


Figure 24-8. LIN Sync Field Detection

Sync field validation is required to verify accuracy of the LIN header and to calculate the commander baud rate. The sync field consists of the data 0x55 inside a byte field (see Figure 24-8). After software detects a valid BREAK field, the counter can be set to measure the SYNC field. Figure 24-8 shows the SYNC byte format. The LINCNT is set to 0 on the start bit falling edge and counts continuously. The LINC0 capture or RX falling edge interrupt fires at the falling edges of the RX line. During interrupt processing, software can measure the individual HIGH/LOW times of the bits themselves with the values in the LINC0 and LINC1 registers to verify that all of the timings are valid.

The following flow describes a LIN sync field validation procedure:

1. Initialize LIN counter to 0 (LINCNT = 0) after detecting a valid break field.
2. Enable interrupt on RX falling edge (CPU_INT.IMASK.RXNE = 1)
3. Setup LIN count control (LINCTL):
   - Enable LIN counter capture on raising RX edge (LINCTL.LINC1CAP = 1)
   - Enable LIN counter capture on falling RX edge (LINCTL.LINC0CAP = 1)
   - Enable LIN counter clearing on RX falling edge (LINCTL.ZERONE = 1)
   - Enable LIN counter (LINCTL.CTRENA = 1)

Actions at each falling edge of the RX line for the sync field as showing below:

1. LIN counter is set to 0 and start counting on the falling RX edge. (LINCTL.ZERONE = 1)
2. RX falling edge interrupt trigger (RXNE):
   - Read capture register LINC0 (falling edge) and LINC1 (rising edge) values
   - Verify bit times
3. RX falling edge interrupt trigger (RXNE):
   - Read capture register LINC0 (falling edge) and LINC1 (rising edge) values
   - Verify bit times
4. RX falling edge interrupt trigger (RXNE):
   - Read capture register LINC0 (falling edge) and LINC1 (rising edge) values
   - Verify bit times
5. RX falling edge interrupt trigger (RXNE):
   - Read capture register LINC0 (falling edge) and LINC1 (rising edge) values
   - Verify bit times
   - Calculate the proper baud rate to set. Software must set the baud rate before the start bit of the PID field after sync field.

On each interrupt occurrence, the capture registers must be read and the bit times need to be validated by the application software. In case of a bit time verification error, the sync field analysis process must be aborted and the application software must switch back to break detection.

In case of errors like a breaking commander communication during sync field detection, a timeout interrupt can be generated by enabling the LIN counter overflow (IMASK.LINOVF = 1). When the counter overflows, the interrupt handler can abort the sync field analysis and switch back to break detection. The time the counter overflow interrupt occurs can be calculated as $t_{Timeout} = 2^{16}$/ UART clock.

Note

The sync field is automatically stored in the RX FIFO and can be misread as the PID. Therefore, flush the RX FIFO after the sync field is received and before the PID is received.

Receiving the PID Field

Read the PID byte from the RXDATA register using the standard UNICOMM-UART receiver. Make sure CTL0.RXE is set.

Receiving the Data Fields

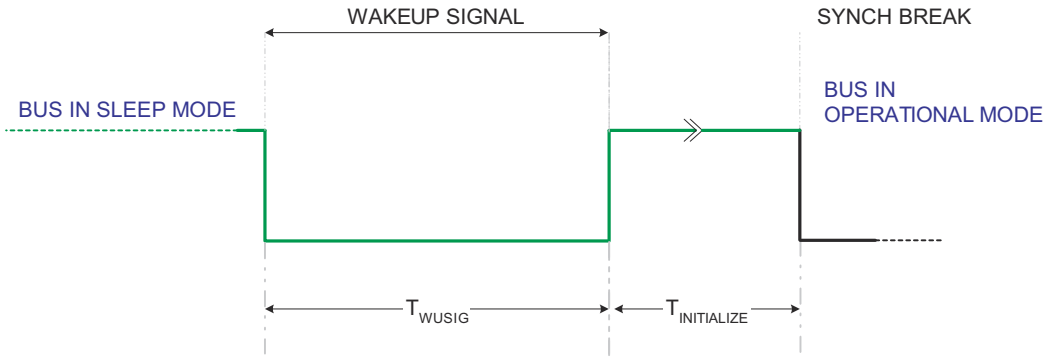
Read the data fields from the RXDATA register using the standard UNICOMM-UART receiver. Make sure CTL0.RXE is set.

Receiving the Checksum Field

Read the checksum byte from the RXDATA register using the standard UNICOMM-UART receiver and decode the value in software. Make sure CTL0.RXE is set.

24.2.3.1.3 LIN Wakeup

Per the LIN specification, any node in a sleeping LIN cluster can request a wake up by transmitting a wake up signal. This signal type is shown in Figure 24-9.



Note

$0.25ms \leq T_{WUSIG} \leq 5ms$

Figure 24-9. LIN Wakeup

To generate a wake signal on the TX pin with the UNICOMM-UART as a LIN controller, the application must use the CTL0.TXD_CTL_EN and CTL0.TXD_OUT bits. Set the TXD_CTL_EN bit to enable register control of the TXD pin, then clear the TXD_OUT bit for the duration of T_{WUSIG} .

24.2.3.2 Flow Control

In the default UART mode (CTL0.MODE set to 0), hardware flow control between two devices is accomplished by connecting the RTS output to the CTS input on the receiving device, and connecting the RTS output on the receiving device to the CTS input. The RTS output signal is active low, the CTS input expects a low signal on a send request as shown in Figure 24-10.

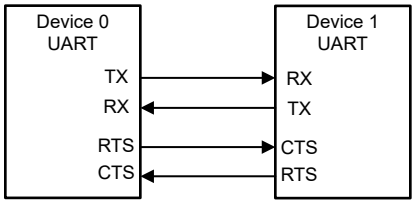


Figure 24-10. Flow Control

The CTS input controls the transmitter. The transmitter can only transmit data when the CTS input is asserted low. When RTS flow control is enabled, the RTS output signal indicates the state of the RX FIFO. CTS of the transmitting device remains asserted low until, the preprogrammed watermark level is reached, indicating that the receiver has no space to store additional characters.

The CTSEN and RTSEN bits in the CTL0 register specify the flow control mode as shown in Table 24-3.

Table 24-3. Flow Control Enable

CTSEN	RTSEN	Description
1	1	RTS and CTS flow control enabled
1	0	Only CTS flow control enabled
0	1	Only RTS flow control enabled
0	0	Both RTS and CTS flow control disabled

The RTS bit can be driven by hardware or software. When RTSEN is 1 (i.e. hardware-controlled mode), the value of CTL0.RTS bit is ignored and RTS output signal is generated by hardware trigger levels as described below.

RTS flow control:

The RTS flow control logic is linked to the programmable RX FIFO watermark levels, which can be configured using the IFLS register. When RTS flow control is enabled, the RTS is asserted (low) until the RX FIFO is filled up to the watermark level. When the RX FIFO watermark level is reached, the RTS signal is de-asserted (high), indicating that there is no more room to receive any more data. The other device's transmission of data is expected to cease after the current character has been transmitted. The RTS signal is reasserted (low) when data has been read out of the RX FIFO until the FIFO is filled to less than the watermark level. If RTS flow control is disabled and the UNICOMM-UART is still enabled, then data is received until the RX FIFO is full, or there is no more data transmitted.

The RTS signal is de-asserted when the FIFO watermark level is reached by putting the last received character into the FIFO. This means on a back to back transmit, another character transfer could already have been started by the other device. In this case, the watermark level must be set to one level lower so that all data can be received and put into the FIFO.

CTS flow control:

If CTS flow control is disabled and the UNICOMM-UART is enabled, data is continuously transmitted until the TX FIFO is empty.

If CTS flow control is enabled, then the transmitter checks that the CTS signal is asserted low before transmitting each byte. The data continues to be transmitted while CTS is asserted (low), and the TX FIFO is not empty. If the TX FIFO is empty and the CTS signal is asserted (low) no data is transmitted. If the CTS signal is de-asserted (high), then the current character transmission is completed before the transmission is halted.

24.2.3.3 RS485 Support

RS485 is a standard used in serial communications systems. This standard can be used effectively over long distances and in electrically noisy environments. Multiple receivers can be connected to a RS-485 network. These characteristics make RS-485 useful in industrial control systems and other similar applications.

The RS485 direction signal controls the external RS485 PHY. The RTS I/O is used in this mode as the direction signal. The signal is set automatically to high once a data transmit is started. This siganl stays high between back to back byte transmits. If a data receive operation is ongoing, a new transmit must be delayed until this data has been received and the direction signal has been set to transmit.

The data exchange sequence is show in Figure 24-11:

- Wait until an ongoing receive has been finished.
- Activate direction signal to transmit on RTS Pin
- Send data (one or more bytes)

- Wait till an ongoing receive has been finished.
- Deactivate direction signal to transmit on RTS Pin

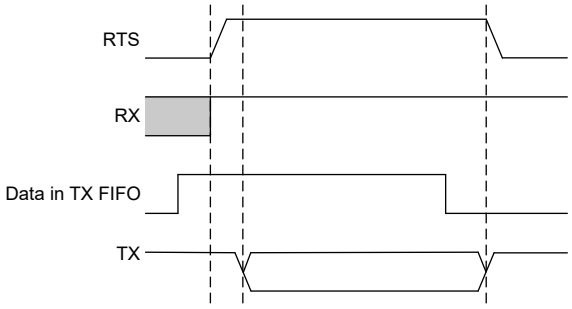


Figure 24-11. RS485 Data Exchange

Two bit fields to the LCRH register define the setup and hold time of the external driver direction control:

- EXTDIR_SETUP bits define number of functional clocks between RTS assertion and beginning of START bit
- EXTDIR_HOLD bits define number of functional clocks between STOP (entire bit duration) and RTS de-assertion

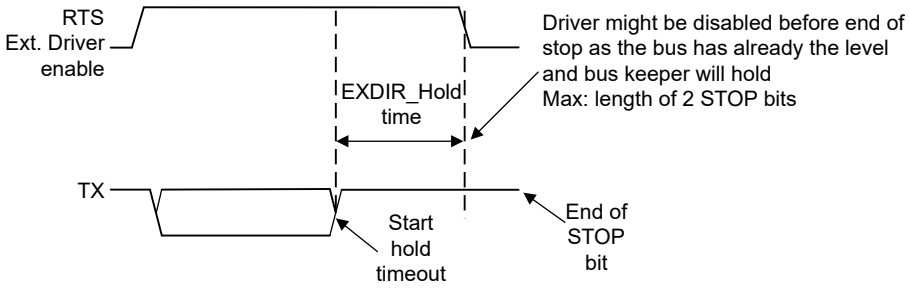


Figure 24-12. RS485 external driver control

24.2.3.4 Idle-Line Multiprocessor

This section applies to UNICOMM-UART configurations which support the UART-IDLELINE-MULTIPROC feature.

When IDLELINE is set in the CTL0.MODE register bits, the idle-line multiprocessor format is selected. Blocks of data are separated by an idle time on the transmit or receive lines (Figure 24-13). An idle receive line is detected when ten or more continuous ones (marks) are received after the one or two stop bits of a character. The baud-rate generator is switched off after reception of an idle line until the next start edge is detected. When an idle line is detected, the IDLE bit in UARTx.STAT is set. In Idle-Line mode the UART receiver operates in no parity mode and the UART word length (UARTx.LCRH.WLEN) must be set to 8bit.

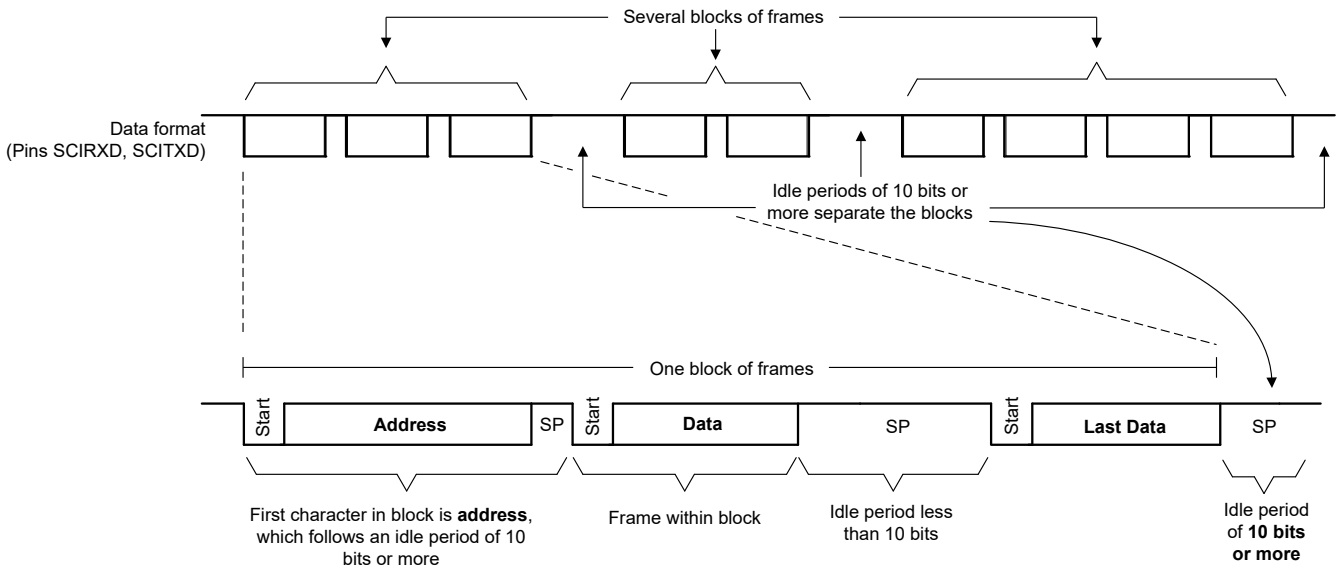


Figure 24-13. Idle-Line Multiprocessor

The first character received after an idle period is an address character. The IDLE bit in UARTx.STAT register is used as an address tag for each block of characters.

If an address character is received it is compared against the ADDR register with the AMASK applied. If the received character matches, the address character and all following received characters are transferred into the RXDATA buffer/FIFO and interrupts are generated until the next address without a match is received. The IDLE bit in STAT register is automatically cleared when the address does match; otherwise the IDLE bit is set until the address is matched.

When SENDIDLE control bit is set, the UART inserts an idle period of 11 bit times on the bus. Application has to ensure that there are no active transactions on the transmit line prior to setting SENDIDLE bit.

Once SENDIDLE bit is set, application has to wait for SENDIDLE status bit to be asserted and then clear the SENDIDLE control bit to proceed. The next transfer can then begin with an address character.

The following procedure sends out an idle frame to indicate an address character followed by associated data:

1. Set SENDIDLE then write the address character to TXDATA. TXDATA must be ready for new data (TXINT interrupt must be 1). This generates an idle period of exactly 11 bits followed by the address character. SENDIDLE is reset automatically when the address character has been transferred (all bits are sent out of shift register).
2. Write desired data characters to TXDATA. TXDATA must be ready for new data (TXINT interrupt must be 1). The data written to TXDATA is transferred to the shift register and transmitted as soon as the shift register is ready for new data.

The idle-line time (10 bit periods) must not be exceeded between address and data transmission or between data transmissions. Otherwise, the transmitted data is misinterpreted as an address.

When using IDLELINE mode, the UART Word Length in the LCRH register must be set to:

- 8-bit word length (WLEN bits 6:5 configured to 0x3)

BUSY bit in IDLELINE:

- TX: BUSY is set during the generation of an IDLELINE signal and while the address and data bytes are sent.
- RX: The BUSY signal while receiving data and till first 10 idle bits are received.

ADVANCE INFORMATION

24.2.3.5 9-Bit UART Mode

9-bit mode is enabled by setting the MODE field to ADDR9BIT in the CTL0 register. This feature is useful in a multi-drop configuration where a single controller connected to multiple peripherals can communicate with a particular peripheral through the peripheral's programmed address or a set of addresses, along with a qualifier for an address byte. In 9-bit UART mode, the parity enable/mode bits are ignored and the UART word length (LCRH.WLEN) must be set to 8 bits.

Receive Transaction:

In 9-bit mode, a UNICOMM-UART peripheral checks for the address qualifier at the location of the parity bit. If set, the received byte is compared with the preprogrammed address in ADDR register:

- If the address matches, a 9-bit mode address match interrupt (ADDR_MATCH) is generated, if enabled, and further data is received into the RX FIFO.
- If the address does not match, the address byte and the subsequent data bytes are not received into the RX FIFO.

The address is predefined in the ADDR register to match with the received byte. Matching can be extended to a set of addresses using the address mask in the AMASK register. By default, the AMASK is 0xFF, meaning that only the specified address can generate a match.

Transmit Transaction:

All the transmits in 9-bit UART mode are interpreted as:

- Address bytes, if the 9th bit is set
- Data bytes, if the 9th bit is cleared

In 9-bit mode, the 9th bit can be controlled by software. The value set in the EPS bit of the LCRH register determines the value of the 9th bit for transmit transactions. To indicate an address byte, the application must set the EPS bit before the byte transmission. To indicate a data byte, the EPS bit must be cleared before the byte transmission. For a complete transmit transaction, the address byte must be transmitted as a single byte transaction with EPS bit set, followed by a data byte burst where the EPS bit is cleared.

Table 24-4. 9th Bit Handling

PEN	EPS	9^{th} Bit (Transmitted or Verified)
0	X	Not transmitted or verified
1	0	0 (= Data)
1	1	1 (= Address)

24.2.3.6 ISO7816 Smart Card Support

ISO7816 Smart Card support is only available for Basic UNICOMM-UART instances.

The UNICOMM-UART Smart Card support feature enables basic communication with an ISO7816 smart card. When configuring the CTL0.MODE field, select the smart card option, the TXD signal is used as a bit clock, and the RXD signal is used as the half-duplex communication line connected to the smart card. Further smart card signals are not supported by UNICOMM-UART.

The clock rate of the UART clock in ISO7816 mode must be in the range of 1 MHz to 5 MHz when using ISO7816 mode, the LCRH register must be set to:

- 8-bit word length (WLEN bits configured to 0x3)
- Even parity (PEN set and EPS set)
- No stick parity (SPS cleared)

In ISO7816 mode, the UART automatically uses 2 stop bits and oversampling of 16 is used; LCRH.STP2 & CTL0.HSE bits are ignored.

If a parity error is detected during a transmission, the RX line is pulled low during the second stop bit. In this case, the UNICOMM-UART aborts the transmission and flushes the TX FIFO. Additionally,the module raises a

parity error interrupt, allowing the software to detect the problem and initiate retransmission of the affected data, as the UART does not support automatic retransmission in this case. The UART does not support automatic retransmission on parity errors. If a parity error is detected on transmission, all further transmit operations are aborted and software must handle retransmission of the affected byte or message.

In Smartcard Mode, the receiver in case of parity error drives the line low and a parity interrupt flag is asserted. The transmitter responds based on the value of this bit.

24.2.3.7 Address Detection

This section applies when a UNICOMM-UART module is configured for Idle-line multiprocessor mode or 9-Bit UART mode in the CTL0.MODE register field.

The ADDR register is used to set the specific address that is matched to the received address byte. This register is used in conjunction with AMASK register to form a match for address-byte received. Only bits where the AMASK is set to '1' are considered. So for full address the AMASK register is set to 0xFF.

Table 24-5. Address Detection

Condition	Idle Line Mode	9-Bit Mode
Address match	Address and Data is moved to RXDATA	Address and Data is moved to RXDATA
Address mismatch	Address and Data get dropped	Address and Data get dropped

24.2.4 Initialization

Follow the high-level UNICOMM configurations in Section 23.3 before executing the below UART-specific configurations.

To enable and initialize the UART, perform the following initialization steps:

1. Clear the ENABLE bit in the CTL0 register before making any of the below configuration changes.
2. Use the baud-rate equation in Figure 24-4 to calculate the IBRD and FBRD registers.
3. Write the integer portion of the BRD to the IBRD register.
4. Write the fractional portion of the BRD to the FBRD register.
5. Write the desired communication mode settings to the CTL0 register (all CTL0 fields other than ENABLE).
6. Write the desired serial parameter configurations to the LCRH register.
7. Configure the desired FIFO trigger levels in the IFLS register.
8. Enable desired interrupts and/or DMA event by using CPU_INT, DMA_TRIG_RX, DMA_TRIG_TX group IMASK registers.
9. (Optionally) Configure the emulation mode for the peripheral in the PDBGCTL register.
10. Enable the UART by setting the CTL0.ENABLE bit.

Note

Before UNICOMM-UART is setup or configuration changes, the CTL0.ENABLE bit must be cleared to avoid unpredictable behavior during the updates or for the first data receive or transmitted afterward.

24.2.5 Interrupt and Events Support

The UART module contains three event publishers and no event subscribers. One event publisher (CPU_INT) manages UART interrupt requests (IRQs) to the CPU subsystem through a static event route. The second and third event publishers (DMA_TRIG_RX, DMA_TRIG_TX) are used to setup the trigger signaling for the DMA through DMA event route.

The UART events are summarized in Table 24-6.

Table 24-6. UART Events

Event	Type	Source	Destination	Configuration	Functionality
Section 13.1.4.1	Publisher	UART	CPU Subsystem	CPU_INT registers	Fixed interrupt route from UART to CPU

Table 24-6. UART Events (continued)

Event	Type	Source	Destination	Configuration	Functionality
Section 13.1.4.2	Publisher	UART	DMA	DMA_TRIG_RX registers	Fixed interrupt route from UART to DMA
Section 13.1.4.2	Publisher	UART	DMA	DMA_TRIG_TX registers	Fixed interrupt route from UART to DMA

24.2.5.1 CPU Interrupt Event Publisher (CPU_INT)

The UART module provides 19 interrupt sources which can be configured to source a CPU interrupt event. In order of decreasing interrupt priority, the CPU interrupt events from the UART are:

Table 24-7. UART CPU Interrupt Event Conditions (CPU_INT)

IIDX STAT	Name	Description
0x01	RTOUT	UART receive timeout interrupt. This interrupt is asserted when the RX FIFO is not empty, and no further data is received specified time in the IFLS.RXIFSEL field.
0x02	FRMERR	UART framing error interrupt.
0x03	PARERR	UART parity error interrupt.
0x04	BRKERR	UART break error interrupt.
0x05	OVRERR	UART receive overrun error interrupt.
0x06	RXNE	Falling edge on RX interrupt, this interrupt triggers when there is a falling edge on RX line.
0x07	RXPE	Rising edge on RX interrupt, this interrupt triggers when there is a rising edge on RX line.
0x08	LINC0	LIN capture 0 match interrupt, this interrupt triggers when the defined capture 0 value is reached in LIN counter.
0x09	LINC1	LIN capture 1 match interrupt, this interrupt triggers when the defined capture 1 value is reached in LIN counter.
0x0A	LINOVF	LIN counter overflow interrupt, this interrupt triggers when the 16bit LIN counter overflows.
0x0B	RXINT	UART receive interrupt.
0x0C	TXINT	UART transmit interrupt.
0x0D	EOT	UART end of transmission interrupt indicates that the last bit of all transmitted data and status has left the serializer and without any further data in the TX FIFO.
0x0E	ADDR_MATCH	Address match interrupt, used in protocols with address to indicate address match happened.
0x0F	CTS	UART clear to send interrupt, indicate the CTS signal status.
0x10	DMA_DONE_RX	This interrupt is set if the RX DMA channel sends the DONE signal.
0x11	DMA_DONE_TX	This interrupt is set if the TX DMA channel sends the DONE signal.
0x12	NERR	The noise error interrupt is set when the 3 sampled values used for majority voting are not the same
0x15	LTOUT	UART line timeout interrupt, This interrupt is asserted when no further data is received specified time in the IFLS.RXTOSEL bits.

The CPU interrupt event configuration is managed with the CPU_INT event management registers. See the *Event Registers* chapter.

The receive timeout interrupt is asserted when the RX FIFO is not empty, and no further data is received specified time in the IFLS.RXIFSEL bits. The receive timeout interrupt is cleared either when the FIFO becomes empty through reading all the data (or by reading the holding register), by reading the interrupt index from IIDX or when a 1 is written to the corresponding bit in the ICLR register.

The receive interrupt (RXINT) is set when the RX FIFO progresses through the programmed trigger level. The receive interrupt is cleared by reading data from the RX FIFO until it becomes less than the trigger level, by reading the interrupt index from IIDX, or by writing a '1' to the RXINT bit in ICLR.

The transmit interrupt (TXINT) is set when the TX FIFO progresses through the programmed trigger level. The transmit interrupt is based on a transition through level, therefore the FIFO must be written past the programmed trigger level otherwise no further transmit interrupts will be generated. The transmit interrupt is cleared by writing data to the TX FIFO until it becomes greater than the trigger level, by reading the interrupt index from IIDX, or by writing a '1' to the TXINT bit in ICLR.

24.2.5.2 DMA Trigger Publisher (DMA_TRIG_RX, DMA_TRIG_TX)

The DMA_TRIG_TX and DMA_TRIG_RX group registers are used to setup trigger signaling for the DMA. These registers (IMASK, RIS, MIS, ISET, ICLR) are present for any UCx configured as UART and can be found in the corresponding UNICOMMUART_REGS register structures. See Section 9.2.3 for how the DMA trigger Event works and can be configured. Each DMA channel can be triggered by any of the conditions listed in the Section 9.2.3.1 and can send a DMA_DONE signal to the corresponding UCx module.

Table 24-8. UART DMA Trigger Condition (DMA_TRIG_RX)

Name	Description
RTOUT	UART receive timeout trigger, This interrupt is asserted when the RX FIFO is not empty, and no further data is received specified time in the IFLS.RXTOSEL bits.
RXINT	UART RX FIFO level trigger.

The receive timeout trigger is asserted when the RX FIFO is not empty, and no further data is received for the amount of time programmed by the IFSEL.RXTOSEL field. The receive timeout interrupt is cleared when the RX FIFO becomes empty through reading all the data, by reading the interrupt index from IIDX, or when a 1 is written to the RTOUT bit in the ICLR register.

The receive interrupt (RXINT) is set when the RX FIFO reaches the programmed trigger level. The receive interrupt is cleared by reading data from the RX FIFO until the level becomes less than the trigger level, by reading the interrupt index from IIDX, or by writing a '1' to the RXINT bit in ICLR.

Table 24-9. UART DMA Trigger Condition (DMA_TRIG_TX)

Name	Description
TXINT	UART TX FIFO level trigger..

The transmit interrupt (TXINT) is set when the TX FIFO progresses through the programmed trigger level. The transmit interrupt is based on a transition through level, therefore the FIFO must be written past the programmed trigger level otherwise no further transmit interrupts are generated. The transmit interrupt is cleared by writing data to the TX FIFO until the level becomes greater than the trigger level, by reading the interrupt index from IIDX, or by writing a '1' to the TXINT bit in ICLR.

24.2.6 Emulation Modes

The module behavior while the device is in debug mode is controlled by the FREE and SOFT bits in PDBGCTL register.

When the device is in debug mode and set into halt mode below behavior can be configured.

Table 24-10. Debug Mode Peripheral Behavior

PDBGCTL.FREE	PDBGCTL.SOFT	Function
1	**x**	Modules continues operation
0	**0**	Module stops immediately
0	**1**	Module stops after the next transfer has been finished

24.3 UNICOMM-UART Registers

This Section describes the UNICOMM-UART Registers.

24.3.1 UNICOMM-UART Base Address Table

Table 24-11. UNICOMM-UART Base Address Table

Bit Field Name		DriverLib Name	Base Address
Instance	Structure		
Uc0uartRegs	UNICOMMUART_REGS	UC0_UART_BASE	0x4060_0000
Uc1uartRegs	UNICOMMUART_REGS	UC1_UART_BASE	0x4060_1000
Uc2uartRegs	UNICOMMUART_REGS	UC2_UART_BASE	0x4060_2000
Uc3uartRegs	UNICOMMUART_REGS	UC3_UART_BASE	0x4064_0000
Uc4uartRegs	UNICOMMUART_REGS	UC4_UART_BASE	0x4064_1000
Uc5uartRegs	UNICOMMUART_REGS	UC5_UART_BASE	0x4064_2000

24.3.2 UNICOMMUART_REGS Registers

Table 24-12 lists the memory-mapped registers for the UNICOMMUART_REGS registers. All register offset addresses not listed in Table 24-12 should be considered as reserved locations and the register contents should not be modified.

Table 24-12. UNICOMMUART_REGS Registers

Offset	Acronym	Register Name	Section
0h	CLKDIV	Clock Divider	Go
8h	CLKSEL	Clock Select for Ultra Low Power peripherals	Go
18h	PDBGCTL	Peripheral Debug Control	Go
20h	IIDX	Interrupt index	Go
28h	IMASK	Interrupt mask	Go
30h	RIS	Raw interrupt status	Go
38h	MIS	Masked interrupt status	Go
40h	ISET	Interrupt set	Go
48h	ICLR	Interrupt clear	Go
58h	IMASK	Interrupt mask	Go
60h	RIS	Raw interrupt status	Go
68h	MIS	Masked interrupt status	Go
70h	ISET	Interrupt set	Go
88h	IMASK	Interrupt mask	Go
90h	RIS	Raw interrupt status	Go
98h	MIS	Masked interrupt status	Go
A0h	ISET	Interrupt set	Go
E4h	INTCTL	Interrupt control register	Go
100h	CTL0	UART Control Register 0	Go
104h	LCRH	UART Line Control Register	Go
108h	STAT	UART Status Register	Go
10Ch	IFLS	UART Interrupt FIFO Level Select Register	Go
110h	IBRD	UART Integer Baud-Rate Divisor Register	Go
114h	FBRD	UART Fractional Baud-Rate Divisor Register	Go
118h	GFCTL	Glitch Filter Control	Go
120h	TXDATA	UART Transmit Data Register	Go
124h	RXDATA	UART Receive Data Register	Go
130h	LINCNT	UART LIN Mode Counter Register	Go
134h	LINCTL	UART LIN Mode Control Register	Go
138h	LINC0	UART LIN Mode Capture 0 Register	Go
13Ch	LINC1	UART LIN Mode Capture 1 Register	Go
148h	AMASK	Self Address Mask Register	Go
14Ch	ADDR	Self Address Register	Go

Complex bit access types are encoded to fit into small table cells. Table 24-13 shows the codes that are used for access types in this section.

Table 24-13. UNICOMMUART_REGS Access Type Codes

Access Type	Code	Description
Read Type		

ADVANCE INFORMATION

Table 24-13. UNICOMMUART_REGS Access Type Codes (continued)

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

1 CLKDIV Register (Offset = 0h) [Reset = 00000000h]

CLKDIV is shown in Figure 24-14 and described in Table 24-14.

Return to the Summary Table.

This register is used to specify module-specific divide ratio of the functional clock

Figure 24-14. CLKDIV Register

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

Table 24-15. CLKDIV Register Field Descriptions

Bit	Field	Type	Reset	Description
31-6	RESERVED	R/W	0h	
5-0	RATIO	R/W	0h	Selects divide ratio of module clock Division factor 0 : DIV_BY_1 1 : DIV_BY_2 .... 63: DIV_BY_64 0h = Do not divide clock source 1h = Divide clock source by 2 2h = Divide clock source by 3 3h = Divide clock source by 4 4h = Divide clock source by 5 5h = Divide clock source by 6 6h = Divide clock source by 7 7h = Divide clock source by 8 3Ch = Divide clock source by 61 3Dh = Divide clock source by 62 3Eh = Divide clock source by 63 3Fh = Divide clock source by 64

2 CLKSEL Register (Offset = 8h) [Reset = 00000000h]

CLKSEL is shown in Figure 24-15 and described in Table 24-15.

Return to the Summary Table.

Clock source selection for peripherals

Figure 24-15. CLKSEL Register

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

Table 24-17. CLKSEL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R/W	0h	
3	BUSCLK_SEL	R/W	0h	Selects BUS CLK as clock source if enabled 0h = Does not select this clock as a source 1h = Select this clock as a source
2-0	RESERVED	R/W	0h	

ADVANCE INFORMATION

3 PDBGCTL Register (Offset = 18h) [Reset = 00000000h]

PDBGCTL is shown in Figure 24-16 and described in Table 24-16.

Return to the Summary Table.

This register can be used by the software developer to control the behavior of the peripheral relative to the 'Core Halted' input

Figure 24-16. PDBGCTL Register

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

Table 24-19. PDBGCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R/W	0h	
1	SOFT	R/W	1h	Soft halt boundary control. This function is only available, if [FREE] is set to 'STOP' 0h = Not supported 1h = The peripheral blocks the debug freeze until it has reached a boundary where it can resume without corruption
0	FREE	R/W	1h	Free run control 0h = The peripheral freezes functionality while the Core Halted input is asserted and resumes when it is deasserted. 1h = The peripheral ignores the state of the Core Halted input

ADVANCE INFORMATION

4 IIDX Register (Offset = 20h) [Reset = 00000000h]

IIDX is shown in Figure 24-17 and described in Table 24-17.

Return to the Summary Table.

This register provides the highest priority enabled interrupt index. Value 0x00 means no event pending. Interrupt 1 is the highest priority, IIDX next highest, 4, 8, ... IIDX^31 is the least priority. That is, the least bit position that is set to 1 denotes the highest priority pending interrupt. The priority order is fixed. However, users can implement their own prioritization schemes using other registers that expose the full set of interrupts that have occurred. On each read, only one interrupt is indicated. On a read, the current interrupt (highest priority) is automatically cleared by the hardware and the corresponding interrupt flag in [RIS] and [MIS] are cleared as well. After a read from the CPU (not from the debug interface), the register is updated with the next highest priority interrupt, if none are pending, then it should display 0x0.

Figure 24-17. IIDX Register

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

Table 24-21. IIDX Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	

Table 24-21. IIDX Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
7-0	STAT	R	0h	UART Module Interrupt Vector Value. This register provides the highes priority interrupt index. A read clears the corresponding interrupt flag in RIS and MIS registers. 15h-1Fh = Reserved 00h = No interrupt pending 01h = UART receive time-out interrupt Interrupt Flag: RT Interrupt Priority: Highest 02h = UART framing error interrupt Interrupt Flag: FE 03h = UART parity error interrupt Interrupt Flag: PE 04h = UART break error interrupt Interrupt Flag: BE 05h = UART receive overrun error interrupt Interrupt Flag: OE 06h = Negative edge on UARTxRXD interrupt Interrupt Flag: RXNE 07h = Positive edge on UARTxRXD interrupt Interrupt Flag: RXPE 08h = LIN capture 0 / match interrupt Interrupt Flag: LINC0 09h = LIN capture 1 interrupt Interrupt Flag: LINC1 0Ah = LIN hardware counter overflow interrupt Interrupt Flag: LINOVF 0Bh = UART receive interrupt Interrupt Flag: RX 0Ch = UART transmit interrupt Interrupt Flag: TX 0Dh = UART end of transmission interrupt (transmit serializer empty) Interrupt Flag: EOT 0Eh = 9-bit mode address match interrupt Interrupt Flag: ADDR_MATCH Fh = UART Clear to Send Modem interrupt Interrupt Flag: CTS 10h = DMA DONE on RX 11h = DMA DONE on TX 12h = Noise Error Event 13h = DMA PRE IRQ INTERRUPT 14h = DMA PRE IRQ INTERRUPT 15h = UART line time-out interrupt Interrupt Flag

5 IMASK Register (Offset = 28h) [Reset = 00000000h]

IMASK is shown in Figure 24-18 and described in Table 24-18.

Return to the Summary Table.

Interrupt Mask. If a bit is set, then corresponding interrupt is un-masked. Un-masking the interrupt causes the raw interrupt to be visible in IIDX, as well as MIS.

Figure 24-18. IMASK Register

31	30	29	28	27	26	25	24
RESERVED							
R/W-0h							
23	22	21	20	19	18	17	16
RESERVED			LTOUT	RESERVED		NERR	DMA_DONE_TX
R/W-0h			R/W-0h	R/W-0h		R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
DMA_DONE_RX	CTS	ADDR_MATCH	EOT	TXINT	RXINT	LINOVF	LINC1
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
LINC0	RXPE	RXNE	OVRERR	BRKERR	PARERR	FRMERR	RTOUT
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 24-23. IMASK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-21	RESERVED	R/W	0h	
20	LTOUT	R/W	0h	Enable UARTOUT Line Time-Out Interrupt. 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
19-18	RESERVED	R/W	0h	
17	NERR	R/W	0h	Noise Error on triple voting. Asserted when the 3 samples of majority voting are not equal 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
16	DMA_DONE_TX	R/W	0h	Enable DMA Done on TX Event Channel Interrupt 0h = Interrupt disabled 1h = Set Interrupt Mask
15	DMA_DONE_RX	R/W	0h	Enable DMA Done on RX Event Channel Interrupt 0h = Interrupt disabled 1h = Set Interrupt Mask
14	CTS	R/W	0h	Enable UART Clear to Send Modem Interrupt. 0h = Interrupt disabled 1h = Set Interrupt Mask
13	ADDR_MATCH	R/W	0h	Enable Address Match Interrupt. 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
12	EOT	R/W	0h	Enable UART End of Transmission Interrupt Indicates that the last bit of all transmitted data and flags has left the serializer and without any further Data in the TX Fifo or Buffer. 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
11	TXINT	R/W	0h	Enable UART Transmit Interrupt. 0h = Clear Interrupt Mask 1h = Set Interrupt Mask

Table 24-23. IMASK Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
10	RXINT	R/W	0h	Enable UART Receive Interrupt. 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
9	LINOVF	R/W	0h	Enable LIN Hardware Counter Overflow Interrupt. 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
8	LINC1	R/W	0h	Enable LIN Capture 1 Interrupt. 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
7	LINC0	R/W	0h	Enable LIN Capture 0 / Match Interrupt . 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
6	RXPE	R/W	0h	Enable Positive Edge on UARTxRXD Interrupt. 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
5	RXNE	R/W	0h	Enable Negative Edge on UARTxRXD Interrupt. 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
4	OVRERR	R/W	0h	Enable UART Receive Overrun Error Interrupt. 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
3	BRKERR	R/W	0h	Enable UART Break Error Interrupt. 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
2	PARERR	R/W	0h	Enable UART Parity Error Interrupt. 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
1	FRMERR	R/W	0h	Enable UART Framing Error Interrupt. 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
0	RTOUT	R/W	0h	Enable UARTOUT Receive Time-Out Interrupt. 0h = Clear Interrupt Mask 1h = Set Interrupt Mask

6 RIS Register (Offset = 30h) [Reset = 00000000h]

RIS is shown in Figure 24-19 and described in Table 24-19.

Return to the Summary Table.

Raw interrupt status. Reflects all pending interrupts, regardless of masking. The RIS register allows the user to implement a poll scheme. A flag set in this register can be cleared by writing 1 to the ICLR register bit even if the corresponding IMASK bit is not enabled.

Figure 24-19. RIS Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED			LTOUT	RESERVED		NERR	DMA_DONE_TX
R-0h			R-0h	R-0h		R-0h	R-0h
15	14	13	12	11	10	9	8
DMA_DONE_RX	CTS	ADDR_MATCH	EOT	TXINT	RXINT	LINOVF	LINC1
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
7	6	5	4	3	2	1	0
LINC0	RXPE	RXNE	OVRERR	BRKERR	PARERR	FRMERR	RTOUT
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

Table 24-25. RIS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-21	RESERVED	R	0h	
20	LTOUT	R	0h	UARTOUT Line Time-Out Interrupt. Set: no start edge has been detected for an additional character period after reception of last character. This is irrespective of FIFO contents Clear: timeout has not occured 0h = Interrupt did not occur 1h = Interrupt occurred
19-18	RESERVED	R	0h	
17	NERR	R	0h	Noise Error on triple voting. Asserted when the 3 samples of majority voting are not equal 0h = Interrupt did not occur 1h = Interrupt occurred
16	DMA_DONE_TX	R	0h	DMA Done on TX Event Channel Interrupt 0h = Interrupt disabled 1h = Interrupt occurred
15	DMA_DONE_RX	R	0h	DMA Done on RX Event Channel Interrupt 0h = Interrupt disabled 1h = Interrupt occurred
14	CTS	R	0h	UART Clear to Send Modem Interrupt. This interrupt is raised when CTS toggles, either from '0' to '1' or from '1' to '0'. 0h = Interrupt disabled 1h = Interrupt occurred
13	ADDR_MATCH	R	0h	Address Match Interrupt. 0h = Interrupt did not occur 1h = Interrupt occurred

Table 24-25. RIS Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
12	EOT	R	0h	UART End of Transmission Interrupt Indicates that the last bit of all transmitted data and flags has left the serializer and without any further Data in the TX Fifo or Buffer. 0h = Interrupt did not occur 1h = Interrupt occurred
11	TXINT	R	0h	UART Transmit Interrupt. 0h = Interrupt did not occur 1h = Interrupt occurred
10	RXINT	R	0h	UART Receive Interrupt. 0h = Interrupt did not occur 1h = Interrupt occurred
9	LINOVF	R	0h	LIN Hardware Counter Overflow Interrupt. 0h = Interrupt did not occur 1h = Interrupt occurred
8	LINC1	R	0h	LIN Capture 1 Interrupt. 0h = Interrupt did not occur 1h = Interrupt occurred
7	LINC0	R	0h	LIN Capture 0 / Match Interrupt . 0h = Interrupt did not occur 1h = Interrupt occurred
6	RXPE	R	0h	Positive Edge on UARTxRXD Interrupt. 0h = Interrupt did not occur 1h = Interrupt occurred
5	RXNE	R	0h	Negative Edge on UARTxRXD Interrupt. 0h = Interrupt did not occur 1h = Interrupt occurred
4	OVRERR	R	0h	UART Receive Overrun Error Interrupt. UART Receive Overrun Error Interrupt is raised when the receive FIFO is flow and a new frame is received 0h = Interrupt did not occur 1h = Interrupt occurred
3	BRKERR	R	0h	UART Break Error Interrupt. Break Error Interrupt is raised when all data bits, parity bit and stop bits are '0' 0h = Interrupt did not occur 1h = Interrupt occurred
2	PARERR	R	0h	UART Parity Error Interrupt. 0h = Interrupt did not occur 1h = Interrupt occurred
1	FRMERR	R	0h	UART Framing Error Interrupt. Framing error interrupt is raised when a "low" stop bit is detected. 0h = Interrupt did not occur 1h = Interrupt occurred
0	RTOUT	R	0h	UARTOUT Receive Time-Out Interrupt. Interupt is raised when a time-out has lapsed and there are still items in the FIFO 0h = Interrupt did not occur 1h = Interrupt occurred

7 MIS Register (Offset = 38h) [Reset = 00000000h]

MIS is shown in Figure 24-20 and described in Table 24-20.

Return to the Summary Table.

Masked interrupt status. This is an AND of the IMASK and RIS registers.

Figure 24-20. MIS Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED			LTOUT	RESERVED		NERR	DMA_DONE_TX
R-0h			R-0h	R-0h		R-0h	R-0h
15	14	13	12	11	10	9	8
DMA_DONE_RX	CTS	ADDR_MATCH	EOT	TXINT	RXINT	LINOVF	LINC1
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
7	6	5	4	3	2	1	0
LINC0	RXPE	RXNE	OVRERR	BRKERR	PARERR	FRMERR	RTOUT
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

Table 24-27. MIS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-21	RESERVED	R	0h	
20	LTOUT	R	0h	Masked UARTOUT Line Time-Out Interrupt. 0h = Interrupt did not occur 1h = Interrupt occurred
19-18	RESERVED	R	0h	
17	NERR	R	0h	Noise Error on triple voting. Asserted when the 3 samples of majority voting are not equal 0h = Interrupt did not occur 1h = Interrupt occurred
16	DMA_DONE_TX	R	0h	Masked DMA Done on TX Event Channel Interrupt 0h = Interrupt did not occur 1h = Interrupt occurred
15	DMA_DONE_RX	R	0h	Masked DMA Done on RX Event Channel Interrupt 0h = Interrupt did not occur 1h = Interrupt occurred
14	CTS	R	0h	Masked UART Clear to Send Modem Interrupt. 0h = Interrupt did not occur 1h = Interrupt occurred
13	ADDR_MATCH	R	0h	Masked Address Match Interrupt. 0h = Interrupt did not occur 1h = Interrupt occurred
12	EOT	R	0h	UART End of Transmission Interrupt Indicates that the last bit of all transmitted data and flags has left the serializer and without any further Data in the TX Fifo or Buffer. 0h = Interrupt did not occur 1h = Interrupt occurred
11	TXINT	R	0h	Masked UART Transmit Interrupt. 0h = Interrupt did not occur 1h = Interrupt occurred

Table 24-27. MIS Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
10	RXINT	R	0h	Masked UART Receive Interrupt. 0h = Interrupt did not occur 1h = Interrupt occurred
9	LINOVF	R	0h	Masked LIN Hardware Counter Overflow Interrupt. 0h = Interrupt did not occur 1h = Interrupt occurred
8	LINC1	R	0h	Masked LIN Capture 1 Interrupt. 0h = Interrupt did not occur 1h = Interrupt occurred
7	LINC0	R	0h	Masked LIN Capture 0 / Match Interrupt . 0h = Interrupt did not occur 1h = Interrupt occurred
6	RXPE	R	0h	Masked Positive Edge on UARTxRXD Interrupt. 0h = Interrupt did not occur 1h = Interrupt occurred
5	RXNE	R	0h	Masked Negative Edge on UARTxRXD Interrupt. 0h = Interrupt did not occur 1h = Interrupt occurred
4	OVRERR	R	0h	Masked UART Receive Overrun Error Interrupt. 0h = Interrupt did not occur 1h = Interrupt occurred
3	BRKERR	R	0h	Masked UART Break Error Interrupt. 0h = Interrupt did not occur 1h = Interrupt occurred
2	PARERR	R	0h	Masked UART Parity Error Interrupt. 0h = Interrupt did not occur 1h = Interrupt occurred
1	FRMERR	R	0h	Masked UART Framing Error Interrupt. 0h = Interrupt did not occur 1h = Interrupt occurred
0	RTOUT	R	0h	Masked UARTOUT Receive Time-Out Interrupt. 0h = Interrupt did not occur 1h = Interrupt occurred

8 ISET Register (Offset = 40h) [Reset = 00000000h]

ISET is shown in Figure 24-21 and described in Table 24-21.

Return to the Summary Table.

Interrupt set. Allows interrupts to be set by software (useful in diagnostics and safety checks). Writing a 1 to a bit in ISET will set the event and therefore the related RIS bit also gets set. If the interrupt is enabled through the mask, then the corresponding MIS bit is also set.

Figure 24-21. ISET Register

31	30	29	28	27	26	25	24
RESERVED							
W-0h							
23	22	21	20	19	18	17	16
RESERVED			LTOUT	RESERVED		NERR	DMA_DONE_TX
W-0h			W-0h	W-0h		W-0h	W-0h
15	14	13	12	11	10	9	8
DMA_DONE_RX	CTS	ADDR_MATCH	EOT	TXINT	RXINT	LINOVF	LINC1
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h
7	6	5	4	3	2	1	0
LINC0	RXPE	RXNE	OVRERR	BRKERR	PARERR	FRMERR	RTOUT
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h

Table 24-29. ISET Register Field Descriptions

Bit	Field	Type	Reset	Description
31-21	RESERVED	W	0h	
20	LTOUT	W	0h	Set UARTOUT Line Time-Out Interrupt. 0h = Writing 0 has no effect 1h = Set Interrupt
19-18	RESERVED	W	0h	
17	NERR	W	0h	Noise Error on triple voting. Asserted when the 3 samples of majority voting are not equal 0h = Writing this has no effect 1h = Set the interrupt
16	DMA_DONE_TX	W	0h	Set DMA Done on TX Event Channel Interrupt 0h = Interrupt disabled 1h = Set Interrupt
15	DMA_DONE_RX	W	0h	Set DMA Done on RX Event Channel Interrupt 0h = Interrupt disabled 1h = Set Interrupt
14	CTS	W	0h	Set UART Clear to Send Modem Interrupt. 0h = Writing 0 has no effect 1h = Set Interrupt
13	ADDR_MATCH	W	0h	Set Address Match Interrupt. 0h = Writing 0 has no effect 1h = Set Interrupt
12	EOT	W	0h	Set UART End of Transmission Interrupt Indicates that the last bit of all transmitted data and flags has left the serializer and without any further Data in the TX Fifo or Buffer. 0h = Writing 0 has no effect 1h = Set Interrupt

Table 24-29. ISET Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
11	TXINT	W	0h	Set UART Transmit Interrupt. 0h = Writing 0 has no effect 1h = Set Interrupt
10	RXINT	W	0h	Set UART Receive Interrupt. 0h = Writing 0 has no effect 1h = Set Interrupt
9	LINOVF	W	0h	Set LIN Hardware Counter Overflow Interrupt. 0h = Writing 0 has no effect 1h = Set Interrupt
8	LINC1	W	0h	Set LIN Capture 1 Interrupt. 0h = Writing 0 has no effect 1h = Set Interrupt
7	LINC0	W	0h	Set LIN Capture 0 / Match Interrupt . 0h = Writing 0 has no effect 1h = Set Interrupt
6	RXPE	W	0h	Set Positive Edge on UARTxRXD Interrupt. 0h = Writing 0 has no effect 1h = Set Interrupt
5	RXNE	W	0h	Set Negative Edge on UARTxRXD Interrupt. 0h = Writing 0 has no effect 1h = Set Interrupt
4	OVRERR	W	0h	Set UART Receive Overrun Error Interrupt. 0h = Writing 0 has no effect 1h = Set Interrupt
3	BRKERR	W	0h	Set UART Break Error Interrupt. 0h = Writing 0 has no effect 1h = Set Interrupt
2	PARERR	W	0h	Set UART Parity Error Interrupt. 0h = Writing 0 has no effect 1h = Set Interrupt
1	FRMERR	W	0h	Set UART Framing Error Interrupt. 0h = Writing 0 has no effect 1h = Set Interrupt
0	RTOUT	W	0h	Set UARTOUT Receive Time-Out Interrupt. 0h = Writing 0 has no effect 1h = Set Interrupt

9 ICLR Register (Offset = 48h) [Reset = 00000000h]

ICLR is shown in Figure 24-22 and described in Table 24-22.

Return to the Summary Table.

Interrupt clear. Write a 1 to clear corresponding Interrupt.

Figure 24-22. ICLR Register

31	30	29	28	27	26	25	24
RESERVED							
W-0h							
23	22	21	20	19	18	17	16
RESERVED			LTOUT	RESERVED		NERR	DMA_DONE_TX
W-0h			W-0h	W-0h		W-0h	W-0h
15	14	13	12	11	10	9	8
DMA_DONE_RX	CTS	ADDR_MATCH	EOT	TXINT	RXINT	LINOVF	LINC1
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h
7	6	5	4	3	2	1	0
LINC0	RXPE	RXNE	OVRERR	BRKERR	PARERR	FRMERR	RTOUT
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h

Table 24-31. ICLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-21	RESERVED	W	0h	
20	LTOUT	W	0h	Clear UARTOUT Line Time-Out Interrupt. 0h = Writing 0 has no effect 1h = Clear Interrupt
19-18	RESERVED	W	0h	
17	NERR	W	0h	Noise Error on triple voting. Asserted when the 3 samples of majority voting are not equal 0h = Writing 0 has no effect 1h = Clear Interrupt
16	DMA_DONE_TX	W	0h	Clear DMA Done on TX Event Channel Interrupt 0h = Interrupt disabled 1h = Clear Interrupt
15	DMA_DONE_RX	W	0h	Clear DMA Done on RX Event Channel Interrupt 0h = Interrupt disabled 1h = Clear Interrupt
14	CTS	W	0h	Clear UART Clear to Send Modem Interrupt. 0h = Writing 0 has no effect 1h = Clear Interrupt
13	ADDR_MATCH	W	0h	Clear Address Match Interrupt. 0h = Writing 0 has no effect 1h = Clear Interrupt
12	EOT	W	0h	Clear UART End of Transmission Interrupt Indicates that the last bit of all transmitted data and flags has left the serializer and without any further Data in the TX Fifo or Buffer. 0h = Writing 0 has no effect 1h = Clear Interrupt
11	TXINT	W	0h	Clear UART Transmit Interrupt. 0h = Writing 0 has no effect 1h = Clear Interrupt

Table 24-31. ICLR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
10	RXINT	W	0h	Clear UART Receive Interrupt. 0h = Writing 0 has no effect 1h = Clear Interrupt
9	LINOVF	W	0h	Clear LIN Hardware Counter Overflow Interrupt. 0h = Writing 0 has no effect 1h = Clear Interrupt
8	LINC1	W	0h	Clear LIN Capture 1 Interrupt. 0h = Writing 0 has no effect 1h = Clear Interrupt
7	LINC0	W	0h	Clear LIN Capture 0 / Match Interrupt . 0h = Writing 0 has no effect 1h = Clear Interrupt
6	RXPE	W	0h	Clear Positive Edge on UARTxRXD Interrupt. 0h = Writing 0 has no effect 1h = Clear Interrupt
5	RXNE	W	0h	Clear Negative Edge on UARTxRXD Interrupt. 0h = Writing 0 has no effect 1h = Clear Interrupt
4	OVRERR	W	0h	Clear UART Receive Overrun Error Interrupt. 0h = Writing 0 has no effect 1h = Clear Interrupt
3	BRKERR	W	0h	Clear UART Break Error Interrupt. 0h = Writing 0 has no effect 1h = Clear Interrupt
2	PARERR	W	0h	Clear UART Parity Error Interrupt. 0h = Writing 0 has no effect 1h = Clear Interrupt
1	FRMERR	W	0h	Clear UART Framing Error Interrupt. 0h = Writing 0 has no effect 1h = Clear Interrupt
0	RTOUT	W	0h	Clear UARTOUT Receive Time-Out Interrupt. 0h = Writing 0 has no effect 1h = Clear Interrupt

10 IMASK Register (Offset = 58h) [Reset = 00000000h]

IMASK is shown in Figure 24-23 and described in Table 24-23.

Return to the Summary Table.

Interrupt Mask. If a bit is set, then corresponding interrupt is un-masked. Un-masking the interrupt causes the raw interrupt to be visible in IIDX, as well as MIS.

Figure 24-23. IMASK Register

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

Table 24-33. IMASK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-11	RESERVED	R/W	0h	
10	RXINT	R/W	0h	Enable UART Receive Interrupt. 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
9-1	RESERVED	R/W	0h	
0	RTOUT	R/W	0h	Enable UARTOUT Receive Time-Out Interrupt. 0h = Clear Interrupt Mask 1h = Set Interrupt Mask

ADVANCE INFORMATION

11 RIS Register (Offset = 60h) [Reset = 00000000h]

RIS is shown in Figure 24-24 and described in Table 24-24.

Return to the Summary Table.

Raw interrupt status. Reflects all pending interrupts, regardless of masking. The RIS register allows the user to implement a poll scheme.

Figure 24-24. RIS Register

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

Table 24-35. RIS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-11	RESERVED	R	0h	
10	RXINT	R	0h	UART Receive Interrupt. 0h = Interrupt did not occur 1h = Interrupt occurred
9-1	RESERVED	R	0h	
0	RTOUT	R	0h	UARTOUT Receive Time-Out Interrupt. 0h = Interrupt did not occur 1h = Interrupt occurred

12 MIS Register (Offset = 68h) [Reset = 00000000h]

MIS is shown in Figure 24-25 and described in Table 24-25.

Return to the Summary Table.

Masked interrupt status. This is an AND of the IMASK and RIS registers.

Figure 24-25. MIS Register

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

Table 24-37. MIS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-11	RESERVED	R	0h	
10	RXINT	R	0h	Masked UART Receive Interrupt. 0h = Interrupt did not occur 1h = Interrupt occurred
9-1	RESERVED	R	0h	
0	RTOUT	R	0h	Masked UARTOUT Receive Time-Out Interrupt. 0h = Interrupt did not occur 1h = Interrupt occurred

ADVANCE INFORMATION

13 ISET Register (Offset = 70h) [Reset = 00000000h]

ISET is shown in Figure 24-26 and described in Table 24-26.

Return to the Summary Table.

Interrupt set. Allows interrupts to be set by software (useful in diagnostics and safety checks). Writing a 1 to a bit in ISET will set the event and therefore the related RIS bit also gets set. If the interrupt is enabled through the mask, then the corresponding MIS bit is also set.

Figure 24-26. ISET Register

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

Table 24-39. ISET Register Field Descriptions

Bit	Field	Type	Reset	Description
31-11	RESERVED	W	0h	
10	RXINT	W	0h	Set UART Receive Interrupt. 0h = Writing 0 has no effect 1h = Set Interrupt
9-1	RESERVED	W	0h	
0	RTOUT	W	0h	Set UARTOUT Receive Time-Out Interrupt. 0h = Writing 0 has no effect 1h = Set Interrupt

14 IMASK Register (Offset = 88h) [Reset = 00000000h]

IMASK is shown in Figure 24-27 and described in Table 24-27.

Return to the Summary Table.

Interrupt Mask. If a bit is set, then corresponding interrupt is un-masked. Un-masking the interrupt causes the raw interrupt to be visible in IIDX, as well as MIS.

Figure 24-27. IMASK Register

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

Table 24-41. IMASK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-12	RESERVED	R/W	0h	
11	TXINT	R/W	0h	Enable UART Transmit Interrupt. 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
10-0	RESERVED	R/W	0h	

15 RIS Register (Offset = 90h) [Reset = 00000000h]

RIS is shown in Figure 24-28 and described in Table 24-28.

Return to the Summary Table.

Raw interrupt status. Reflects all pending interrupts, regardless of masking.

Figure 24-28. RIS Register

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

Table 24-43. RIS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-12	RESERVED	R	0h	
11	TXINT	R	0h	UART Transmit Interrupt. 0h = Interrupt did not occur 1h = Interrupt occurred
10-0	RESERVED	R	0h	

16 MIS Register (Offset = 98h) [Reset = 00000000h]

MIS is shown in Figure 24-29 and described in Table 24-29.

Return to the Summary Table.

Masked interrupt status. This is an AND of the IMASK and RIS registers.

Figure 24-29. MIS Register

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

Table 24-45. MIS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-12	RESERVED	R	0h	
11	TXINT	R	0h	Masked UART Transmit Interrupt. 0h = Interrupt did not occur 1h = Interrupt occurred
10-0	RESERVED	R	0h	

17 ISET Register (Offset = A0h) [Reset = 00000000h]

ISET is shown in Figure 24-30 and described in Table 24-30.

Return to the Summary Table.

Interrupt set. Allows interrupts to be set by software (useful in diagnostics and safety checks). Writing a 1 to a bit in ISET will set the event and therefore the related RIS bit also gets set. If the interrupt is enabled through the mask, then the corresponding MIS bit is also set.

Figure 24-30. ISET Register

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

Table 24-47. ISET Register Field Descriptions

Bit	Field	Type	Reset	Description
31-12	RESERVED	W	0h	
11	TXINT	W	0h	Set UART Transmit Interrupt. 0h = Writing 0 has no effect 1h = Set Interrupt
10-0	RESERVED	W	0h	

ADVANCE INFORMATION

18 INTCTL Register (Offset = E4h) [Reset = 00000000h]

INTCTL is shown in Figure 24-31 and described in Table 24-31.

Return to the Summary Table.

Interrupt control register

Figure 24-31. INTCTL Register

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

Table 24-49. INTCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	W	0h	
0	INTEVAL	W	0h	Writing a 1 to this field re-evaluates the interrupt sources. 0h = The interrupt or event line is disabled. 1h = The interrupt or event line is in software mode. Software must clear the RIS.

ADVANCE INFORMATION

19 CTL0 Register (Offset = 100h) [Reset = 00000038h]

CTL0 is shown in Figure 24-32 and described in Table 24-32.

Return to the Summary Table.

UART Control Register The CTL0 register is the control register. All the bits are cleared on reset except for the Transmit Enable (TXE) and Receive Enable (RXE) bits, which are set. To enable the UART module, the UARTEN bit must be set. If software requires a configuration change in the module, the UARTEN bit must be cleared before the configuration changes are written. If the UART is disabled during a transmit or receive operation, the current transaction is completed prior to the UART stopping. NOTE: The CTL0 register should not be changed while the UART is enabled or else the results are unpredictable. The following sequence is recommended for making changes to the CTL0 register. 1. Disable the UART. 2. Wait for the end of transmission or reception of the current character. 3. Clear transmit FIFO 4. Reprogram the control register. 5. Enable the UART.

Figure 24-32. CTL0 Register

31	30	29	28	27	26	25	24
RESERVED							
R/W-0h							
23	22	21	20	19	18	17	16
RESERVED				MSBFIRST	MAJVOTE	RESERVED	HSE
R/W-0h				R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
HSE	CTSEN	RTSEN	RTS	RESERVED	MODE		
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h		
7	6	5	4	3	2	1	0
RESERVED	TXD_OUT	TXD_OUT_EN	TXE	RXE	LBE	RESERVED	ENABLE
R/W-0h	R/W-0h	R/W-1h	R/W-1h	R/W-1h	R/W-0h	R/W-0h	R/W-0h

Table 24-51. CTL0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-20	RESERVED	R/W	0h	
19	MSBFIRST	R/W	0h	Most Significant Bit First This bit has effect both on the way protocol byte is transmitted and received. Notes: User needs to match the protocol to the correct value of this bit to send MSb or LSb first. The hardware engine will send the byte entirely based on this bit. 0h = Least significant bit is sent first in the protocol packet 1h = Most significant bit is sent first in the protocol packet
18	MAJVOTE	R/W	0h	Majority Vote Enable When Majority Voting is enabled, the three center bits are used to determine received sample value. In case of error (i.e. all 3 bits are not the same), noise error is detected and bits RIS.NERR and register RXDATA.NERR are set. Oversampling of 16 : bits 7, 8, 9 are used Oversampling of 8 : bits 3, 4, 5 are used Disabled : Single sample value (center value) used 0h = Majority voting is disabled 1h = Majority voting is enabled
17	RESERVED	R/W	0h	
16-15	HSE	R/W	0h	High-Speed Bit Oversampling Enable **NOTE:** The bit oversampling influences the UART baud-rate configuration. The state of this bit has no effect on clock generation in ISO7816 smart card mode (the SMART bit is set). 0h = 16x oversampling. 1h = 8x oversampling. 2h = 3x oversampling. IrDA, Manchester and DALI not supported when 3x oversampling is enabled.

ADVANCE INFORMATION

Table 24-51. CTL0 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
14	CTSEN	R/W	0h	Enable Clear To Send 0h = CTS hardware flow control is disabled. 1h = CTS hardware flow control is enabled. Data is only transmitted when the UARTxCTS signal is asserted.
13	RTSEN	R/W	0h	Enable hardware controlled Request to Send 0h = RTS hardware flow control is disabled. 1h = RTS hardware flow control is enabled. Data is only requested (by asserting UARTxRTS) when the receive FIFO/buffer has available entries.
12	RTS	R/W	0h	Request to Send If RTSEN is set the RTS output signals is controlled by the hardware logic using the FIFO fill level or buffer status. If RTSEN is cleared the RTS output is controlled by the RTS bit. The bit is the complement of the UART request to send, RTS modem status output. 0h = Signal not RTS 1h = Signal RTS
11	RESERVED	R/W	0h	
10-8	MODE	R/W	0h	Set the communication mode and protocol used. (Not defined settings uses the default setting: 0) 0h = Normal operation 1h = RS485 mode: UART needs to be IDLE with receiving data for the in EXTDIR_HOLD set time. EXTDIR_SETUP defines the time the RTS line is set to high before sending. When the buffer is empty the RTS line is set low again. A transmit will be delayed as long the UART is receiving data. 2h = The UART operates in IDLE Line Mode 3h = The UART operates in 9 Bit Address mode 4h = ISO7816 Smart Card Support The application must ensure that it sets 8-bit word length (WLEN set to 3h) and even parity (PEN set to 1, EPS set to 1, SPS set to 0) in UARTLCRH when using ISO7816 mode. The value of the STP2 bit in UARTLCRH is ignored and the number of stop bits is forced to 2. 5h = DALI Mode:
7	RESERVED	R/W	0h	
6	TXD_OUT	R/W	0h	TXD Pin Control Controls the TXD pin when TXD_OUT_EN = 1 and TXE = 0. 0h = TXD pin is low 1h = TXD pin is high
5	TXD_OUT_EN	R/W	1h	TXD Pin Control Enable. When the transmit section of the UART is disabled (TXE = 0), the TXD pin can be controlled by the TXD_OUT bit. 0h = TXD pin can not be controlled by TXD_OUT 1h = TXD pin can be controlled by TXD_OUT
4	TXE	R/W	1h	UART Transmit Enable If the UART is disabled in the middle of a transmission, it completes the current character before stopping. **NOTE:** To enable transmission, the UARTEN bit must be set. 0h = The transmit section of the UART is disabled. The UARTxTXD pin of the UART can be controlled by the TXD_CTL bit when enabled. 1h = The transmit section of the UART is enabled.
3	RXE	R/W	1h	UART Receive Enable If the UART is disabled in the middle of a receive, it completes the current character before stopping. **NOTE:** To enable reception, the UARTEN bit must be set. 0h = The receive section of the UART is disabled. 1h = The receive section of the UART is enabled.
2	LBE	R/W	0h	UART Loop Back Enable 0h = Normal operation. 1h = The UARTxTX path is fed through the UARTxRX path internally.
1	RESERVED	R/W	0h	

ADVANCE INFORMATION

Table 24-51. CTL0 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
0	ENABLE	R/W	0h	UART Module Enable. If the UART is disabled in the middle of transmission or reception, it completes the current character before stopping. If the ENABLE bit is not set, all registers can still be accessed and updated. It is recommended to setup and change the UART operation mode with having the ENABLE bit cleared to avoid unpredictable behavior during the setup or update. If disabled the UART module will not send or receive any data and the logic is held in reset state. 0h = Disable Module 1h = Enable module

20 LCRH Register (Offset = 104h) [Reset = 00000000h]

LCRH is shown in Figure 24-33 and described in Table 24-33.

Return to the Summary Table.

UART Line Control Register The LCRH register is the line control register. Serial parameters such as data length, parity, and stop bit selection are implemented in this register. When updating the baud-rate divisor (UARTIBRD or UARTIFRD), the LCRH register must also be written. The write strobe for the baud-rate divisor registers is tied to the LCRH register.

Figure 24-33. LCRH Register

31	30	29	28	27	26	25	24
RESERVED					SUSPEND	EXTDIR_HOLD	
R/W-0h					R/W-0h	R/W-0h	
23	22	21	20	19	18	17	16
EXTDIR_HOLD			EXTDIR_SETUP				
R/W-0h			R/W-0h				
15	14	13	12	11	10	9	8
RESERVED							
R/W-0h							
7	6	5	4	3	2	1	0
SENDIDLE	SPS	WLEN		STP2	EPS	PEN	BRK
R/W-0h	R/W-0h	R/W-0h		R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 24-53. LCRH Register Field Descriptions

Bit	Field	Type	Reset	Description
31-27	RESERVED	R/W	0h	
26	SUSPEND	R/W	0h	Suspend external communication 0h = Functional mode resumed 1h = External communication suspended
25-21	EXTDIR_HOLD	R/W	0h	Defines the number of UARTclk ticks the signal to control the external driver for the RS485 will be reset after the beginning of the stop bit. (If 2 STOP bits are enabled the beginning of the 2nd STOP bit.) 0h = Smallest value 1Fh = Highest possible value
20-16	EXTDIR_SETUP	R/W	0h	Defines the number of UARTclk ticks the signal to control the external driver for the RS485 will be set before the START bit is send 0h = Smallest value 1Fh = Highest possible value
15-8	RESERVED	R/W	0h	
7	SENDIDLE	R/W	0h	UART send IDLE pattern. When this bit is set, SENDIDLE period of 11 bit times will be sent on the TX line. Read STAT.SENDIDLE bit to readback current status of SENDIDLE 0h = Normal operation 1h = Send Idle Pattern

Table 24-53. LCRH Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
6	SPS	R/W	0h	UART Stick Parity Select The Stick Parity Select (SPS) bit is used to set either a permanent '1' or a permanent '0' as parity when transmitting or receiving data. Its purpose is to typically indicate the first byte of a package or to mark an address byte, for example in a multi-drop RS-485 network. When bits PEN, EPS, and SPS of UARTLCRH are set, the parity bit is transmitted and checked as a 0. When bits PEN and SPS are set and EPS is cleared, the parity bit is transmitted and checked as a 1. 0h = Disable Stick Parity 1h = Enable Stick Parity
5-4	WLEN	R/W	0h	UART Word Length The bits indicate the number of data bits transmitted or received in a frame as follows: 0h = 5 bits (default) 1h = 6 bits 2h = 7 bits 3h = 8 bits
3	STP2	R/W	0h	UART Two Stop Bits Select When in 7816 smart card mode (the SMART bit is set in the UARTCTL register), the number of stop bits is forced to 2. 0h = One stop bit is transmitted at the end of a frame. 1h = Two stop bits are transmitted at the end of a frame. The receive logic checks for two stop bits being received and provide Frame Error if either is invalid.
2	EPS	R/W	0h	UART Even Parity Select This bit has no effect when parity is disabled by the PEN bit. For 9-Bit UART Mode transmissions, this bit controls the address byte and data byte indication (9th bit). 0 = The transferred byte is a data byte 1 = The transferred byte is an address byte 0h = Odd parity is performed, which checks for an odd number of 1s. 1h = Even parity generation and checking is performed during transmission and reception, which checks for an even number of 1s in data and parity bits.
1	PEN	R/W	0h	UART Parity Enable 0h = Parity is disabled and no parity bit is added to the data frame. 1h = Parity checking and generation is enabled.
0	BRK	R/W	0h	UART Send Break (for LIN Protocol) 1. Break condition is sent on the line for as long as this bit is set 0h = Normal operation 1h = A low level is continually output on the UARTxTXD signal, after completing transmission of the current character. For the proper execution of the break command, software must set this bit for at least two frames (character periods).

21 STAT Register (Offset = 108h) [Reset = 00000000h]

STAT is shown in Figure 24-34 and described in Table 24-34.

Return to the Summary Table.

UART Status Register

Figure 24-34. STAT Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED					SENDIDLE	IDLE	CTS
R-0h					R-0h	R-0h	R-0h
7	6	5	4	3	2	1	0
TXCLR	TXFF	TXFE	RXCLR	RXFF	RXFE	RESERVED	BUSY
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

Table 24-55. STAT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-11	RESERVED	R	0h	
10	SENDIDLE	R	0h	TX FIFO Clear Status 0h = IDLE condition has not yet been sent on the line 1h = IDLE condition has been sent on the line
9	IDLE	R	0h	IDLE mode has been detected in Idleline-Multiprocessor-Mode. The IDLE bit is used as an address tag for each block of characters. In idle-line multiprocessor format, this bit is set when a received character is an address. 0h = IDLE has not been detected before last received character. (In idle-line multiprocessor mode). 1h = IDLE has been detected before last received character. (In idle-line multiprocessor mode).
8	CTS	R	1h	Clear To Send 0h = The CTS signal is not asserted (high). 1h = The CTS signal is asserted (low).
7	TXCLR	R	0h	TX FIFO Clear Status 0h = FIFO is not cleared 1h = FIFO clear is complete
6	TXFF	R	0h	UART Transmit FIFO Full The meaning of this bit depends on the state of the FEN bit in the CTL0 register. 0h = The transmitter is not full. 1h = transmit FIFO is full.
5	TXFE	R	1h	UART Transmit FIFO Empty 0h = The transmitter has data to transmit. 1h = transmit FIFO is empty.
4	RXCLR	R	0h	RX FIFO Clear Status 0h = FIFO is not cleared 1h = FIFO clear is complete
3	RXFF	R	0h	UART Receive FIFO Full 0h = The receiver can receive data. 1h = receive FIFO is full.

ADVANCE INFORMATION

Table 24-55. STAT Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
2	RXFE	R	1h	UART Receive FIFO Empty 0h = The receiver is not empty. 1h = eceive FIFO is empty.
1	RESERVED	R	0h	
0	BUSY	R	0h	UART Busy This bit is set as soon as the transmit FIFO/buffer becomes non-empty (regardless of whether UART is enabled) or if a receive data is currently ongoing (after the start edge have been detected until a complete byte, including all stop bits, has been received by the shift register). In IDLE_Line mode the Busy signal also stays set during the idle time generation. 0h = The UART is not busy. 1h = The UART is busy transmitting data. This bit remains set until the complete byte, including all stop bits, has been sent/received from/into the shift register.

22 IFLS Register (Offset = 10Ch) [Reset = 00000022h]

IFLS is shown in Figure 24-35 and described in Table 24-35.

Return to the Summary Table.

The IFLS register is the interrupt FIFO level select register. You can use this register to define the levels at which the TX, RX and timeout interrupt flags are triggered. The interrupts are generated based on a transition through a level rather than being based on the level. That is, the interrupts are generated when the fill level progresses through the trigger level. For example, if the receive trigger level is set to the half-way mark, the interrupt is triggered when the receive FIFO is filled with two or more characters. Out of reset, the TXIFLSEL and RXIFLSEL bits are configured so that the FIFOs trigger an interrupt at the half-way mark.

Figure 24-35. IFLS Register

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

Table 24-57. IFLS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-12	RESERVED	R/W	0h	
11-8	RXTOSEL	R/W	0h	UART Receive Interrupt Timeout Select. When receiving no start edge for an additional character within the set bittimes a RX interrupt is set even if the FIFO level is not reached. A value of 0 disables this function. 0h = Smallest value Fh = Highest possible value
7	RXCLR	R/W	0h	RX FIFO CLEAR. Setting this bit will clear the RX FIFO contents. 0h = Disable FIFO clear 1h = Enable FIFO Clear
6-4	RXIFLSEL	R/W	2h	FIFO Level Select for generating events (interrupt/dma). Note: for undefined settings the default configuration is used. 1h = RX FIFO >= 1/4 full 2h = RX FIFO >= 1/2 full (default) 3h = RX FIFO >= 3/4 full 4h = Opposite of empty 5h = RX FIFO is full 6h = RX_FIFO >= (MAX_FIFO_LEN -1) 7h = RX_FIFO <= 1
3	TXCLR	R/W	0h	TX FIFO CLEAR. Setting this bit will clear the TX FIFO contents. 0h = Disable FIFO clear 1h = Enable FIFO Clear

ADVANCE INFORMATION

Table 24-57. IFLS Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
2-0	TXIFLSEL	R/W	2h	FIFO Level Select for generating events (interrupt/dma). Note: for undefined settings the default configuration is used. 1h = TX FIFO <= 3/4 empty 2h = TX FIFO <= 1/2 empty (default) 3h = TX FIFO <= 1/4 empty 4h = Opposite of full 5h = TX FIFO is empty 6h = TX FIFO <= 1 7h = TX_FIFO >= (MAX_FIFO_LEN -1)

23 IBRD Register (Offset = 110h) [Reset = 00000000h]

IBRD is shown in Figure 24-36 and described in Table 24-36.

Return to the Summary Table.

When changing the IBRD register, the new value does not take effect until transmission/reception of the current character is complete. Any changes to the baud-rate divisor must be followed by a write to the UARTLCRH register. See Baud-Rate Generation chapter for configuration details.

Figure 24-36. IBRD Register

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

Table 24-59. IBRD Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R/W	0h	
15-0	DIVINT	R/W	0h	Integer Baud-Rate Divisor 0h = Smallest value FFFFh = Highest possible value

24 FBRD Register (Offset = 114h) [Reset = 00000000h]

FBRD is shown in Figure 24-37 and described in Table 24-37.

Return to the Summary Table.

UART Fractional Baud-Rate Divisor Register The FBRD register is the fractional part of the baud-rate divisor value. All the bits are cleared on reset. When changing the FBRD register, the new value does not take effect until transmission/reception of the current character is complete. Any changes to the baud-rate divisor must be followed by a write to the UARTLCRH register. See Baud-Rate Generation chapter for configuration details.

Figure 24-37. FBRD Register

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

Table 24-61. FBRD Register Field Descriptions

Bit	Field	Type	Reset	Description
31-6	RESERVED	R/W	0h	
5-0	DIVFRAC	R/W	0h	Fractional Baud-Rate Divisor 0h = Smallest value 3Fh = Highest possible value

25 GFCTL Register (Offset = 118h) [Reset = 0000000Xh]

GFCTL is shown in Figure 24-38 and described in Table 24-38.

Return to the Summary Table.

This register control the glitch filter on the RX input.

Figure 24-38. GFCTL Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																										DGFSEL					
R/W-0h																										R/W-00Bh					

Table 24-63. GFCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-6	RESERVED	R/W	0h	
5-0	DGFSEL	R/W	00Bh	Glitch Suppression Pulse Width This field controls the pulse width select for glitch suppression on the RX line. The value programmed in this field gives the number of cycles of functional clock up to which the glitch has to be suppressed on the RX line. In IRDA mode: The minimum pulse length for receive is given by: t(MIN) = (DGFSEL) / f(IRTXCLK) 0h = Bypass GF 3Fh = Highest Possible Value

26 TXDATA Register (Offset = 120h) [Reset = 00000000h]

TXDATA is shown in Figure 24-39 and described in Table 24-39.

Return to the Summary Table.

UART Transmit Data Register. This register is the transmit data register (the interface to the FIFOs). For transmitted data, if the FIFO is enabled, data written to this location is pushed onto the transmit FIFO. If the FIFO is disabled, data is stored in the transmitter holding register (the bottom word of the transmit FIFO). A write to this register initiates a transmission from the UART.

Figure 24-39. TXDATA Register

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

Table 24-65. TXDATA Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	W	0h	
7-0	DATA	W	0h	Data Transmitted or Received Data that is to be transmitted via the UART is written to this field. When read, this field contains the data that was received by the UART. 0h = Smallest value FFh = Highest possible value

27 RXDATA Register (Offset = 124h) [Reset = 00000000h]

RXDATA is shown in Figure 24-40 and described in Table 24-40.

Return to the Summary Table.

UART Receive Data Register. This register is the data receive register (the interface to the FIFOs). For received data, if the FIFO is enabled, the data byte and the 4-bit status (break, frame, parity, and overrun) is pushed onto the 12-bit wide receive FIFO. If the FIFO is disabled, the data byte and status are stored in the receiving holding register (the bottom word of the receive FIFO). The received data can be retrieved by reading this register.

Figure 24-40. RXDATA Register

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

Table 24-67. RXDATA Register Field Descriptions

Bit	Field	Type	Reset	Description
31-12	RESERVED	R	0h	
11	NERR	R	0h	Noise Error. Writing to this bit has no effect. The flag is cleared by writing 1 to the NERR bit in the UART EVENT ICLR register. 0h = No noise error occurred 1h = Noise error occurred during majority voting
10	BRKERR	R	0h	UART Break Error Writing to this bit has no effect. The flag is cleared by writing 1 to the BRKERR bit in the UART EVENT ICLR register. This error is associated with the character at the top of the FIFO. When a break occurs, only one 0 character is loaded into the FIFO. The next character is only enabled after the receive data input goes to a 1 (marking state) and the next valid start bit is received. 0h = No break condition has occurred 1h = A break condition has been detected, indicating that the receive data input was held low for longer than a full-word transmission time (defined as start, data, parity, and stop bits).
9	PARERR	R	0h	UART Parity Error Writing to this bit has no effect. The flag is cleared by writing 1 to the PARERR bit in the UART EVENT ICLR register. 0h = No parity error has occurred 1h = The parity of the received data character does not match the parity defined by bits 2 and 7 of the UARTLCRH register.
8	FRMERR	R	0h	UART Framing Error Writing to this bit has no effect. The flag is cleared by writing 1 to the FRMERR bit in the UART EVENT ICLR register. This error is associated with the character at the top of the FIFO. 0h = No framing error has occurred 1h = The received character does not have a valid stop bit sequence, which is one or two stop bits depending on the UARTLCRH.STP2 setting (a valid stop bit is 1).

Table 24-67. RXDATA Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
7-0	DATA	R	0h	Received Data. When read, this field contains the data that was received by the UART. 0h = Smallest value FFh = Highest possible value

28 LINCNT Register (Offset = 130h) [Reset = 00000000h]

LINCNT is shown in Figure 24-41 and described in Table 24-41.

Return to the Summary Table.

UART LIN Mode Counter Register

Figure 24-41. LINCNT Register

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

Table 24-69. LINCNT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R/W	0h	
15-0	VALUE	R/W	0h	16 bit up counter clocked by the functional clock of the UART. 0h = Smallest value FFFFh = Highest possible value

ADVANCE INFORMATION

29 LINCTL Register (Offset = 134h) [Reset = 00000000h]

LINCTL is shown in Figure 24-42 and described in Table 24-42.

Return to the Summary Table.

UART LIN Mode Control Register

Figure 24-42. LINCTL Register

31	30	29	28	27	26	25	24
RESERVED							
R/W-0h							
23	22	21	20	19	18	17	16
RESERVED							
R/W-0h							
15	14	13	12	11	10	9	8
RESERVED							
R/W-0h							
7	6	5	4	3	2	1	0
EN_FRM_ERR	LINC0_MATCH	LINC1CAP	LINC0CAP	RESERVED	CNTRXLOW	ZERONE	CTRENA
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 24-71. LINCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R/W	0h	
7	EN_FRM_ERR	R/W	0h	Enable FRAME ERROR 0h = Disable Frame Error Generation If STOP bit of '0' is received, then, FRAME ERROR is not generated 1h = Enable Frame Error Generation If STOP bit of '0' is received, then, FRAME ERROR is generated
6	LINC0_MATCH	R/W	0h	Counter Compare Match Mode When this bit is set to 1 a counter compare match with LINC0 register triggers an LINC0 interrupt when enabled. 0h = Counter compare match mode disabled (capture mode enabled) 1h = Counter compare match enabled (capture mode disabled)
5	LINC1CAP	R/W	0h	Capture Counter on positive RXD Edge. When enabled the counter value is captured to LINC1 register on each positive RXD edge. A LINC1 interrupt is triggered when enabled. 0h = Capture counter on positive UARTxRXD edge disabled 1h = Capture counter on positive UARTxRXD edge enabled
4	LINC0CAP	R/W	0h	Capture Counter on negative RXD Edge. When enabled the counter value is captured to LINC0 register on each negative RXD edge. A LINC0 interrupt is triggered when enabled. 0h = Capture counter on negative UARTxRXD edge disabled 1h = Capture counter on negative UARTxRXD edge enabled
3	RESERVED	R/W	0h	
2	CNTRXLOW	R/W	0h	Count while low Signal on RXD When counter is enabled and the signal on RXD is low, the counter increments. 0h = Count while low Signal on UARTxRXD disabled 1h = Count while low Signal on UARTxRXD enabled
1	ZERONE	R/W	0h	Zero on negative Edge of RXD. When enabled the counter is set to 0 and starts counting on a negative edge of RXD 0h = Zero on negative edge disabled 1h = Zero on negative edge enabled

Table 24-71. LINCTL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
0	CTRENA	R/W	0h	LIN Counter Enable. LIN counter will only count when enabled. 0h = Counter disabled 1h = Counter enabled

30 LINC0 Register (Offset = 138h) [Reset = 00000000h]

LINC0 is shown in Figure 24-43 and described in Table 24-43.

Return to the Summary Table.

UART LIN Mode Capture 0 Register

Figure 24-43. LINC0 Register

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

Table 24-73. LINC0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R/W	0h	
15-0	DATA	R/W	0h	16 Bit Capture / Compare Register Captures current LINCTR value on RXD falling edge and can generate a LINC0 interrupt when capture is enabled (LINC0CAP = 1). If compare mode is enabled (LINC0_MATCH = 1), a counter match can generate a LINC0 interrupt. 0h = Smallest value FFFFh = Highest possible value

31 LINC1 Register (Offset = 13Ch) [Reset = 00000000h]

LINC1 is shown in Figure 24-44 and described in Table 24-44.

Return to the Summary Table.

UART LIN Mode Capture 1 Register

Figure 24-44. LINC1 Register

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

Table 24-75. LINC1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R/W	0h	
15-0	DATA	R/W	0h	16 Bit Capture / Compare Register Captures current LINCTR value on RXD rising edge and can generate a LINC1 interrupt when capture is enabled (LINC1CAP = 1) 0h = Smallest value FFFFh = Highest possible value

32 AMASK Register (Offset = 148h) [Reset = 00000000h]

AMASK is shown in Figure 24-45 and described in Table 24-45.

Return to the Summary Table.

Self Address Mask Register The AMASK register is used to enable the address mask for 9-bit or Idle-Line mode. The address bits are masked to create a set of addresses to be matched with the received address byte. Used in DALI, UART 9-Bit or Idle-Line mode.

Figure 24-45. AMASK Register

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

Table 24-77. AMASK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R/W	0h	
7-0	VALUE	R/W	0h	Self Address Mask for 9-Bit Mode This field contains the address mask that creates a set of addresses that should be matched. A 0 bit in the MSK bitfield configures, that the corresponding bit in the ADDR bitfield of the UARTxADDR register is don't care. A 1 bit in the MSK bitfield configures, that the corresponding bit in the ADDR bitfield of the UARTxADDR register must match. 0h = Smallest value FFh = Highest possible value

33 ADDR Register (Offset = 14Ch) [Reset = 00000000h]

ADDR is shown in Figure 24-46 and described in Table 24-46.

Return to the Summary Table.

Self Address Register The ADDR register is used to write the specific address that should be matched with the receiving byte when the Address Mask (AMASK) is set to FFh. This register is used in conjunction with AMASK to form a match for address-byte received. Used in DALI, UART 9-Bit or Idle-Line mode.

Figure 24-46. ADDR Register

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

Table 24-79. ADDR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R/W	0h	
7-0	VALUE	R/W	0h	Self Address for 9-Bit Mode This field contains the address that should be matched when UARTxAMASK is FFh. 0h = Smallest value FFh = Highest possible value

Chapter 25

Inter-Integrated Circuit (I2C)

The functionality of a UNICOMM module when configured to operate as an I2C (either I2C Controller or I2C Target) is described in this section. The protocol mode of a given UNICOMM instance is decided by the programmed SELECT field of the IPMODE register. If a UNICOMM instance is configured for another peripheral mode, the I2CT/I2CC functionality on that UNICOMM instance is disabled and unusable.

25.1 Overview

The UNICOMM-I2CC (Controller) and UNICOMM-I2CT (Target) modules provides a standardized serial interface to transfer data between AM13E230x devices and other external I^2C devices (such as a sensors, memory, DACs, other MCUs).

25.1.1 Purpose of the Peripheral

The UNICOMM-I2CC and UNICOMM-I2CT peripherals provides bidirectional data transfer through a two-wire serial bus consisting of a data (SDA) and clock (SCL) line. The I^2C bus is widely used to interface with devices such as battery management ICs, sensors, other MCUs and so on. This I^2C peripheral provides the ability to both transmit to and receive from other I^2C devices on the bus.

The **Basic UNICOMM-I2CC** module features include additional support for:

- Digital Glitch Supression

The **Advanced UNICOMM-I2CC** module features include additional support for:

- Analog Glitch Supression
- Burst Mode
- SMBus Support (PEC, Timeout Detection)

The **Basic UNICOMM-I2CT** module features include additional support for:

- Digital Glitch Supression

The **Advanced UNICOMM-I2CT** module features include additional support for:

- Analog glitch Supression
- SMBus Support (PEC, Timeout Detection, Enhanced Acknowledgement features, Default Device/Host/Alert Response Addresses, Target Arbitration)
- Secondary Target Address & Mask

25.1.2 Features

The UNICOMM-I2CC (Controller) and UNICOMM-I2CT (Target) peripheral modes of the UNICOMM module includes the following features:

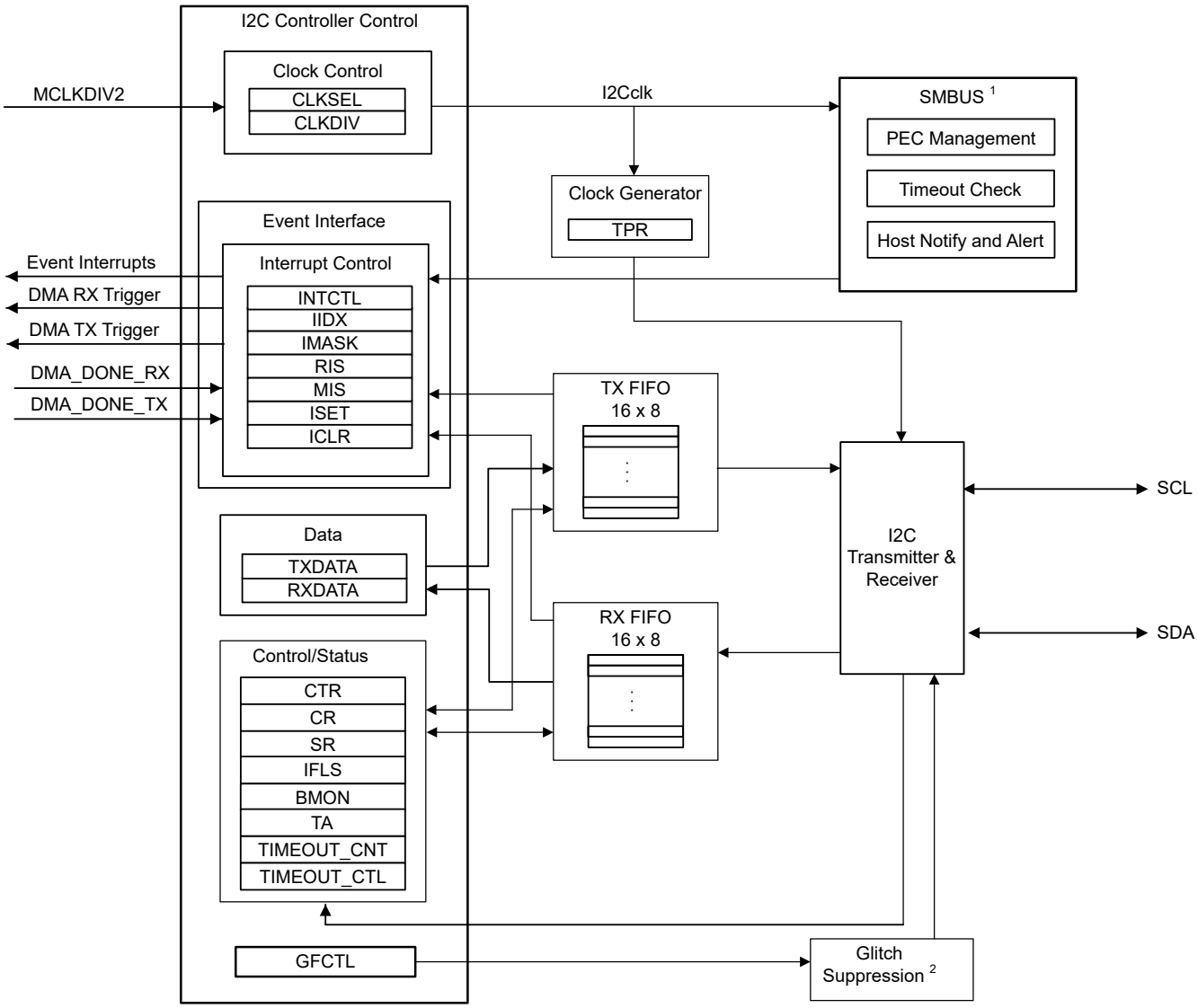
- 7-bit and 10-bit addressing
- Separate transmit (TX) and receive (RX) buffers/first-in first-out buffers (FIFOs)
- Supported transmission speeds:
  - Standard-mode (Sm) with a bit rate up to 100 kbps
  - Fast-mode (Fm) with a bit rate up to 400 kbps
  - Fast-mode Plus (Fm+) with a bit rate up to 1 Mbps
- Direct Memory Access (DMA) support
- Clock stretching (optional)
- Controller arbitration
- Multiple controller mode
- Glitch suppresion: analog (Advanced instances) and digital (Basic instances)
- Burst mode (Advanced instances)
- Dual addressing (Advanced instances)
- Available Interrupts:
  - Transmit (TX) and receive (RX) FIFO levels and empty/full conditions
  - START and STOP conditions
  - Transmit and receive done events
  - Target timeout events
  - Overrun, underrun, arbitration lost, PEC receive, and NACK error
  - DMA done conditions
- SMBus 3.0 hardware support (Advanced instances)
  - Enhanced ACK control

ADVANCE INFORMATION

- Quick command
- General call interrupt
- Clock low and high timeout
- Packet error checking (PEC)
- Host notify protocol
- Alert Response Protocol
- Address Resolution Protocol (ARP)

To support I2C protocol, a UNICOMM module must be configured as either UNICOMM-I2CC (Controller) or UNICOMM-I2CT (Target) in the UNICOMM top level IPMODE register.

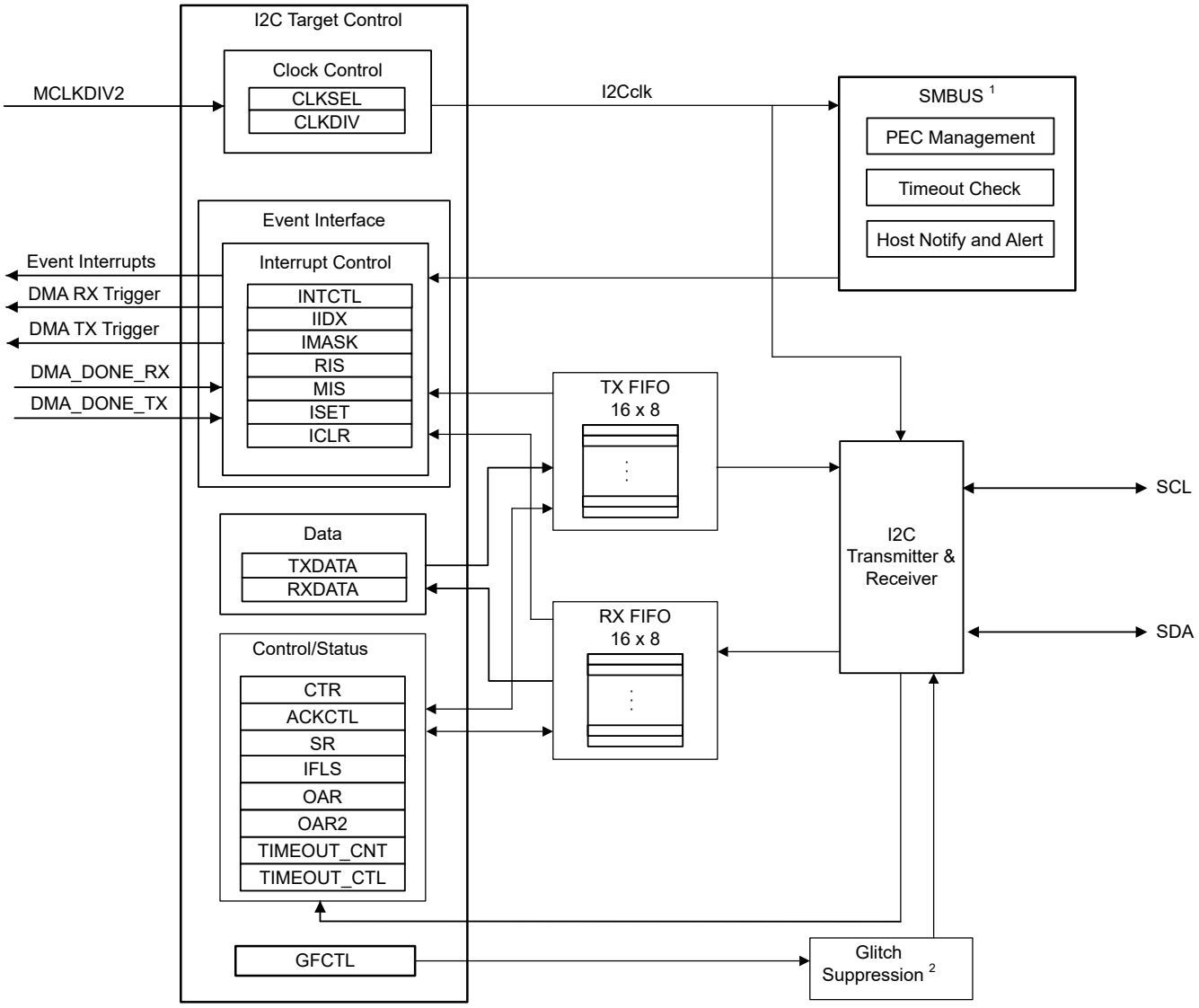
25.1.3 Functional Block Diagram



[1] Only Advanced I2CC instances have built-in SMBUS support
[2] Basic I2CC instances have only digital glitch support and Advanced I2CC instances have only analog glitch support

Figure 25-1. I2C Controller (I2CC) Functional Block Diagram

ADVANCE INFORMATION



[1] Only Advanced I2CT instances have built-in SMBUS support
[2] Basic I2CT instances have only digital glitch support and Advanced I2CT instances have only analog glitch support

Figure 25-2. I2C Target (I2CT) Functional Block Diagram

25.2 Peripheral Functional Description

25.2.1 Clock Control

25.2.1.1 Clock Select and I²C Speed

Standard, Fast, and Fast Plus modes are selected using a value in the I²C Controller Timer Period (TPR) register that results in a maximum SCL frequency of:

- 100 kbps for Standard mode
- 400 kbps for Fast mode
- 1 Mbps for Fast mode Plus

The I²C frequency I2C_FREQ is determined by the I2Cclk functional clock frequency and the TPR field, as well as the fixed SCL_LP , and SCL_HP values where:

- I2Cclk is the functional clock frequency to the I²C module. Note that the I²C internal functional clock is first divided from the source clock:
  - Use the CLKSEL register to enable MCLKDIV2 as the functional clock source for the UNICOMM I²C module.
  - Use the CLKDIV.RATIO register field to select divide ratio of I²C functional clock, options are from divide by 1 to 64.
- SCL_LP is the low phase of SCL: This is fixed at a value of 6
- SCL_HP is the high phase of SCL: This is fixed at a value of 4
- TPR is the programmed value of the TPR field of the TPR register. This value is determined by replacing the known variables in the equation below and solving for TPR.

The I²C freqency is calculated as follows:

$$I2C\_FREQ = I2Cclk / ((1+TPR) * (SCL\_LP + SCL\_HP)) \quad (37)$$

For example, if the I²C functional clock frequency is 32 MHz and target SCL frequency is 400 kHz:

I2Cclk = 32 MHz

I2C_FREQ = 400 kHz

SCL_LP = 6 , SCL_HP = 4

TPR = (I2Cclk / (I2C_FREQ * (4 + 6))) - 1

TPR = 7 (0x07)

Table 25-1. Examples of Controller Clock Setting for Typical Clock Configurations

I2Cclk	TPR Bits Standard Mode 100-kHz SCL	TPR Bits Fast Mode 400-kHz SCL	TPR Bits Fast Mode Plus 1000-kHz SCL
4 MHz	0x03	-	-
8 MHz	0x07	0x01	-
20 MHz	0x13	0x04	0x01
32 MHz	0x1F	0x07	0x02
40 MHz	0x27	0x09	0x03

The I²C functional clock must be greater than or equal to 20 times the SCL frequency, I2Cclk ≥ 20 × I2C_FREQ.

The following minimum functional clock frequencies are required when running certain I2C clock speeds:

- I2Cclk ≥ 2 MHz when working with I²C speed 0 to 100 kHz
- I2Cclk ≥ 8 MHz when working with I²C speed 100 to 400 kHz
- I2Cclk ≥ 20 MHz when working with I²C speed 400 kHz to 1 MHz

25.2.1.2 Clock Startup

The module clock source is always available once setup in the CLKSEL register. After enabling the I²C module by setting the PWREN.ENABLE bit from the UNICOMM top level, the module is ready to start receiving and transmitting data.

25.2.2 Signal Descriptions

The I²C bus consists of a clock signal and a data signal. The clock signal can be generated either internally (during controller operation) or externally (during target operation).

ADVANCE INFORMATION

Table 25-2. I²C signal descriptions

Device Pin	Function
I2Cx_SCL	I²C clock signal
I2Cx_SDA	I²C data signal

25.2.3 General Architecture

25.2.3.1 I²C Bus Functional Overview

The I²C bus uses only two signals: SDA and SCL. SDA is the bidirectional serial data line and SCL is the bidirectional serial clock line. The bus is considered idle when both lines are in high state and no transfer is ongoing.

Every transaction on the I²C bus is 9-bits long, consisting of 8 data bits and 1 acknowledge bit. A transaction is defined as the time between a valid START and STOP condition—as described in Figure 25-3. The number of bytes per transaction is unrestricted; however, each data byte must be followed by an acknowledge bit and data must be transferred MSB first. When a receiving device cannot receive another complete byte, it can hold the clock line SCL Low and force the transmitting device into a wait state.This process is commonly known as clock stretching. The data transaction continues when the receiver releases the clock SCL.

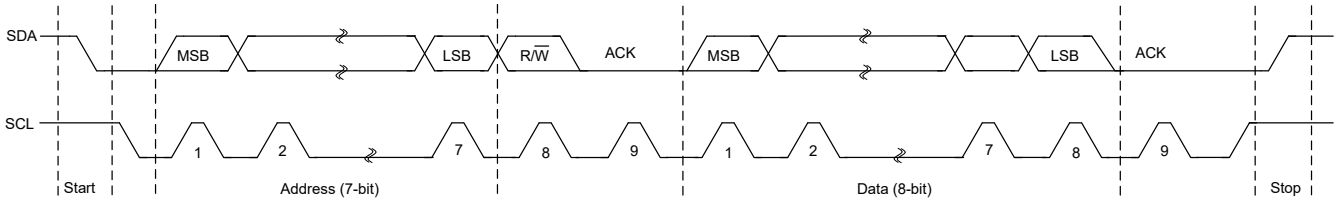


Figure 25-3. Module Data Transfer

The data on the SDA line must be stable during the high period of the clock, and the data line can only change when SCL is low (see Figure 25-4), otherwise START or STOP conditions are generated.

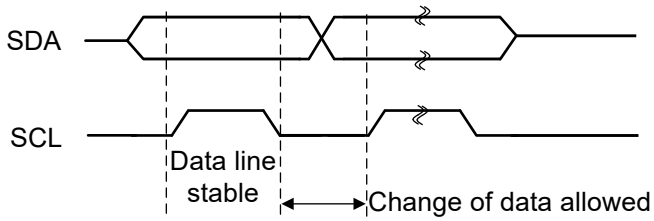


Figure 25-4. Data Change Period

25.2.3.2 START and STOP Conditions

The protocol of the I²C bus defines two states to begin and end a transaction: START and STOP. A high-to-low transition on the SDA line while the SCL is high is defined as a START condition, and a low-to-high transition on the SDA line while SCL is high is defined as a STOP condition. START and STOP conditions are always generated by the controller. The bus is considered busy after a START condition and free after a STOP condition.

ADVANCE INFORMATION

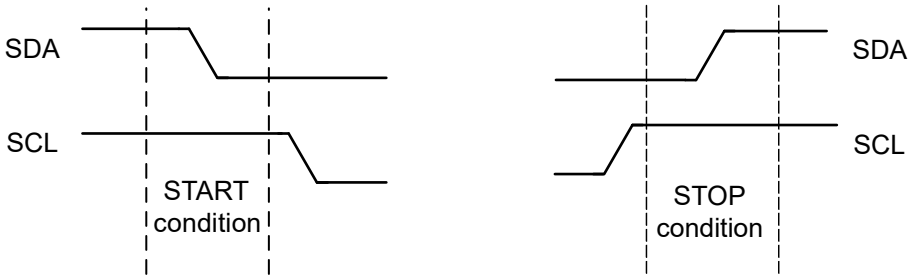


Figure 25-5. START and STOP Conditions

The STOP bit determines if the transaction stops at the end of the data cycle or continues on to a repeated START condition.

To generate a single transaction, the I^2C controller target address TA.ADDR register is written with the desired address, the TA.DIR bit should be set to 1 to start a receive operation and 0 to start a transmit operation , and the control register (CTR) is written with ACK = X (0 or 1), STOP = 1, START = 1, and FRM_START = 1 to perform the operation and generate a STOP at the end. When the operation is completed (or aborted due an error) the interrupt flags are set. When the I^2C module operates in Controller receiver mode, the ACK bit is normally (in case of no error) set causing the I^2C bus controller to transmit an acknowledge automatically after each byte. This bit must be cleared when the I^2C bus controller requires no further data to be transmitted from the target transmitter. For more information on I^2C controller mode configuration please refer to Section 25.2.4.1 .

When operating in target mode, the START and STOP bits in the CPU_INT.RIS register indicate detection of start and stop conditions on the bus and the CPU_INT.IMASK can be configured to allow START and STOP to be promoted to controller interrupts (when interrupts are enabled). For more information on I^2C target mode configuration please refer to Section 25.2.4.2 .

Bus Status Flags:

- BUSBSY is set when a START condition is detected on the bus and cleared when a STOP condition is detected on the bus.
- BUSY is also set after a START/Repeated START, and cleared after CTR.BLEN bytes of data is transferred. It is also cleared after data is NACK ‘d or a STOP.
- IDLE is set when the Controller I^2C state machine is in the IDLE state indicating there is no ongoing transfer.

25.2.3.3 7-Bit Address Format

Standard 7-bit addressing I^2C data transfers follow the format shown in Figure 25-6. To transmit/receive using a 7-bit addressing mode, set TA.MODE=0 for UNICOMM-I2CC devices and OAR.MODE=0 for UNICOMM-I2CT devices.

The I^2C controller initiates the START condition, then a target address is transmitted. This address is 7-bits long, followed by an eighth bit, which is a data direction bit, set by TA.DIR for UNICOMM-I2CC modules. A TA.DIR bit set to '0' indicates a transmit operation (send) a TA.DIR bit set to '1' indicates a request to receive data (receive). A data transfer is always terminated by a STOP condition generated by the controller. However, a controller can initiate communications with another device on the bus by generating a Repeated START condition and addressing another target without first generating a STOP condition, see section Figure 25-8. Various combinations of receive/transmit formats are then possible within a single transfer. The ninth bit is the Acknowledge bit, which is described in Section 25.2.3.7 .

The target address sent in the I^2C frame matches the unique target address of a target device on the bus. For UNICOMM-I2CT devices, this unique address is programmed in the OAR.OAR register field, where the top 3 bits are "don't care" in for 7-bit addressing mode. If the UNICOMM-I2CT device receives an address byte on the bus matching the address programmed in OAR.OAR with a read bit, it shifts the data bytes that follow into the RX FIFO. If the UNICOMM-I2CT device receives an address byte on the bus matching the address programmed in OAR.OAR with a write bit, it shifts out the data bytes in the TX FIFO and onto the SDA line.

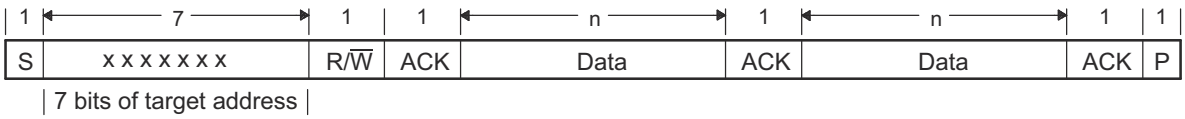


Figure 25-6. Data Format with 7-Bit Address

25.2.3.4 10-Bit Address Format

To transmit/receive using a 10-bit addressing mode, set TA.MODE=1 for UNICOMM-I2CC devices and OAR.MODE=1 for UNICOMM-I2CT devices.

Standard 10-bit addressing I^2C data transfers follow the format shown in Data Format with 10-Bit Address. The 10-bit addressing format is similar to the 7-bit addressing format, except the controller sends the target address in two separate byte transfers. After the START condition, the first byte consists of 11110b, the two MSBs of the 10-bit target address, and the R/W bit. The second byte contains the remaining 8 bits of the 10-bit target address. A data transfer is always terminated by a STOP condition or repeated START condition generated by the controller.

The target side matching is also similar to 7-bit addressing mode, except that the UNICOMM-I2CT address matches the full 10-bits of the programmed OAR.OAR field.

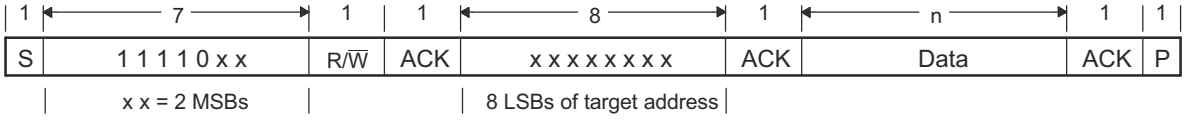


Figure 25-7. Data Format with 10-Bit Address

25.2.3.5 General Call

The general call feature is available UNICOMM-I2CT's and is enabled by setting the CTR.GENCALL bit. The General Call is identified by an address of 0x00 and R/W bit set to '0'. With this feature, the I2CT module responds to a General Call on the I^2C bus, as well as any frames with addresses matching the programmed OAR (and OAR2 if applicable and enabled) addresses. The General Call interrupt is flagged upon receiving a general call frame from the I2C controller. This interrupt can be enabled by setting the CPU_INT.IMASK.GENCALL bit.

25.2.3.6 Dual Address

The dual addresses feature is only available for Advanced UNICOMM-I2CT instances.

An additional programmable UNICOMM-I2CT Own Address Register called OAR2 is provided and can be used for matching in a 7-bit addressing format, if enabled. When the secondary address is disabled (OAR2.OAR2EN=0), the I2CT provides an ACK on the bus if the address matches the OAR field in the OAR register.

In dual address mode (OAR2.OAR2EN=1), the UNICOMM-I2CT provides an ACK on the bus if either the OAR field in the OAR register or the OAR2 field in the OAR2 register is matched. The OAR2SEL bit in the SR register indicates whether the address that was ACK'ed is the alternate address or the primary address. A cleared OAR2SEL bit indicates that either the primary address matched, or there was no OAR2 address match.

25.2.3.7 Acknowledge

All bus transactions have a required acknowledge clock cycle that is generated by the controller. During the acknowledge cycle, the transmitter (which can be the controller or target) releases the SDA line. To acknowledge the transaction, the receiver must pull down SDA during the acknowledge clock cycle. The acknowledge cycle must comply with the data validity requirements.

When a target receiver does not acknowledge the target address, SDA must be left high by the target so that the controller can generate a STOP condition and abort the current transfer or generate a repeated START condition to start a new transfer. If the controller device is acting as a receiver during a transaction, it is responsible for

acknowledging each transfer made by the target. Because the controller decides the number of bytes in the transaction, it signals the end of data to the target transmitter by not generating an acknowledge on the last data byte. The target transmitter must then release SDA to allow the controller to generate the STOP or a repeated START condition.

A target can generate an ACK/NACK manually or automatically.

- When ACKCTL.ACKOEN=0, the target will send automatic acknowledgements. The I2CT will receive and ACK all bytes automatically until the RX FIFO is full.
- When ACKCTL.ACKOEN =1, manual acknowledgments are enabled. Manual ACK override can be used to evaluate each received byte or to slow down the communication when automatic FIFO reception is not desired. When manual ACK override operation is enabled, the I^2C target module's clock is pulled low after the last data bit until this ACKCTL.ACKOVAL is written with the indicated response. The reception of new data is indicated by the RXDONE interrupt flag.

If the I^2C controller receives a NACK while transmitting data, the NACK and TXDONE bit will be set in the RIS registers. If there is still data in the TX FIFO, the TXEMPTY bit will remain clear to inform software that a TX FIFO flush may be required.

25.2.3.8 Repeated Start

The direction of data flow on SDA can be changed by the controller, without first stopping a transfer, by issuing a repeated START condition. This is called a RESTART, see data format in Figure 25-8. After a RESTART is issued, the target address is again sent out with the new data direction specified by the R/W bit.

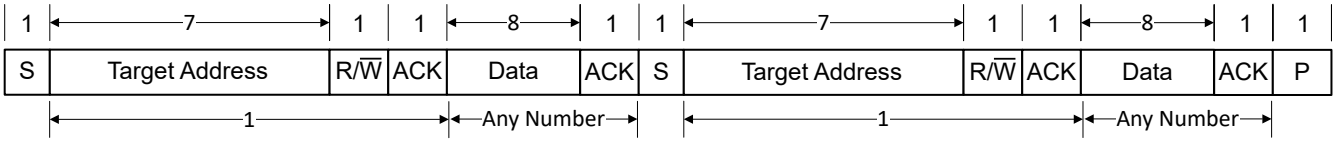


Figure 25-8. Data Format with Repeated Start

A repeated start sequence for a controller transmit or receive is as follows:

1. When the device is in the idle state, the Controller writes the target address to the TA register and configures the DIR bit for the desired transfer type.
2. Data is written to the TXDATA register (for transmit) or RXDATA register (for receive).
3. When SR.BUSY status reads back as '0', FRM_START and START bits in the CTR register are set to initiate a transfer.
4. Software waits till BUSY bit in the SR register reads back as '0'.
5. The controller can now change direction of transfer or address another target, by generating a restart condition. Software updates TA register and then, sets FRM_START. This initiates a repeated start on the line.

   Note: if there is a NACK for address, a STOP condition will be automatically generated. RESTART cannot be sent in that case.

25.2.3.9 Clock Low Timeout

Timeout Counter A

An UNICOMM-I2CT can extend the transaction by pulling the SCL line low to slow down communication. Both the UNICOMM-I2CC and UNICOMM-I2CT modules have a counter called TCNTA, which is a 12-bit programmable counter that tracks how long the clock has been held **low**. The upper 8-bits of the count value are software programmable through the TIMEOUT_CTL register. The value programmed in the TIMEOUT_CTL.TCNTLA register field must be greater than 0x01 to enable the timeout A feature.

The application can program the eight most significant bits of the counter to reflect the acceptable cumulative low period in a transaction. Each count is equal to a timeout period of ((1 + TPR) × 12) I2Cclk's where the

ADVANCE INFORMATION

TPR is the programmable timer period. TIMEOUT_CNT.TCNTA counts continuously for the entire time SCL is held in a low state. When SCL goes back high, TIMEOUT_CNT.TCNTA is reloaded with the value in the TIMEOUT_CTL.TCNTLA register, and begins counting down from this value at the falling edge of SCL.

The internal I2Cclk keeps running even if SCL is held low on the bus.

The I^2C clock low timeout period is calculated as follows:

- Cumulative SCL low period = Timeout counter * One timeout period
- One timeout period = (I2Cclk period) * (1 + TPR) * (12)
- Timeout counter = TIMEOUT_CTL.TCNTLA register (this register contains the upper 8-bits of a 12-bit counter value, the lower 4-bits are set to 0)

As an example, if the I2Cclk is configured to 20 MHz and the I^2C module is operating at a 100-kHz speed, the TPR would be equal to 19. See Section 25.2.1.1 on how TPR is calculated. One timeout period is equal to (1 / 20 MHz) × (1 + 19) × 12 or 12 µs. Programming the TIMEOUT_CTL.TCNTLA register to 0xDA would translate to the value 0xDA0, because the lower 4-bits are set to 0x0. This translates to a decimal value of 3488 clocks or a cumulative clock low period of 3488 × 12 µs, or 41.856 ms at 100 kHz.

The TIMEOUTA bit in the UNICOMM-I2CC RIS register is set when the clock timeout period is reached, allowing the controller to start corrective action to resolve the remote target state. This bit is cleared after I^2C goes to idle or during the I^2C controller reset. The status of the raw SDA and SCL signals are readable by software through the SDA and SCL bits in the BMON register to help determine the state of the remote target. In the event of a timeout condition, application software must choose how it intends to attempt bus recovery. If a timeout is detected before the end of a transfer (receive or transmit), software needs to flush the FIFOs before initiating the next transfer. This timeout feature is needed for SMBus and PMBus implementation.

A UNICOMM-I2CT module can also trigger an interrupt on a configured clock timeout low condition using the same registers and fields.

Timeout Counter B

Timeout Counter B is only available for Advanced UNICOMM-I2CC and UNICOMM-I2CT instances.

Both the UNICOMM-I2CC and UNICOMM-I2CT modules have a secondary counter called TCNTB, which is a 12-bit programmable counter that tracks how long the clock has been held **high**. This counter is configured similarly to the TCNTA, where the upper 8-bits of the count value are software programmable through the TIMEOUT_CTL.TCNTLB register. The value programmed in the TIMEOUT_CTL.TCNTLB register field must be greater than 0x01 to enable the timeout B feature. There is an interrupt flag available for the TCNTB counter for UNICOMM I2CC and UNICOMM-I2CT in the event that the clock timeout period is reached, the TIMEOUTB bit in the RIS register is set.

Note

All timeout configurations must only be modified while the UNICOMM-I2C module is disabled - CR.ENABLE=0 for I2CC and CTR.ENABLE=0 for I2CT.

25.2.3.10 Clock Stretching

For a UNICOMM-I2CC, clock stretching can be disabled if none of the targets on the bus support clock stretching, allowing the controller to reach the maximum speed on the bus. Otherwise, the clock can be slowed by a target keeping the clock low or due to the clock status detection delay within the I^2C module.

For a UNICOMM-I2CT, clock stretching is activated automatically when either the RX FIFO full or TX FIFO empty is set. Clock stretching support can be enabled or disabled by configuring the CTR.CLKSTRETCH bit, where the feature is enabled by default. Clock stretching status is indicated by the TREQ and RREQ bits from the SR register described below.

- TREQ

  - When this bit is set, the I^2C target has been addressed as a target transmitter and is using clock stretching to delay the controller until data has been written to TXDATA.
- RREQ
  - When this bit is set, the I^2C controller has outstanding receive data from the I2C controller and is using clock stretching to delay the controller until the data has been read from RXDATA.

25.2.3.11 Arbitration

A controller can start a transfer only if the bus is idle. It's possible for two or more controllers to generate a START condition within minimum hold time of the START condition. In these situations, an arbitration scheme takes place on the SDA line, while SCL is high (see Figure 25-9) . The first controller transmitter that generates a logic high is overruled by the opposing controller generating a logic low and the losing controller releases the bus until the bus is idle again.

Arbitration can take place over several bits. Its first stage is a comparison of address bits, and if both controllers are trying to address the same device, arbitration continues on to the comparison of data bits.

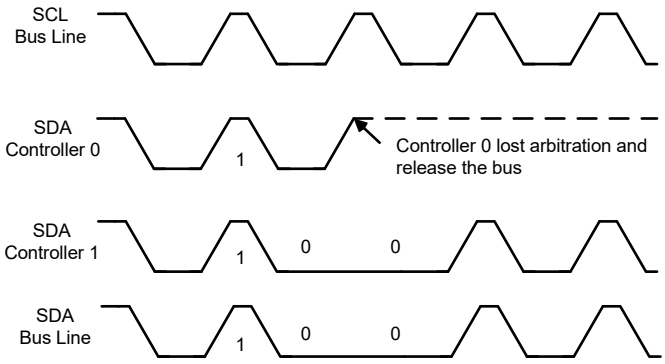


Figure 25-9. Arbitration Procedure Between Two Controller Transmitters

When an arbitration lost error occurs, the SR.ARBLST flag is set. It will be reset by the hardware with the next STOP condition is detected on the bus. Additionally the ARBLOST flags in RIS registers are set.

If arbitration is lost when the I2CC has initiated a transfer, the application should execute the following steps to correctly handle the arbitration loss:

- Clear the TX FIFO
- Once the bus is IDLE, the TXFIFO can be filled and enabled, the TXEMPTY bit can be unmasked and a new transaction can be initiated.

25.2.3.12 Multiple Controller Mode

To enable multiple I2C controllers on the bus, the CR.CTL bit must be set to enable multi-controller mode. During the arbitration procedure, the clocks from all the different controllers must be synchronized. A device that first generates a low period on SCL overrules the other devices, forcing them to start their own low periods. SCL is then held low by the device with the longest low period. The other devices must wait for SCL to be released before starting their high periods.

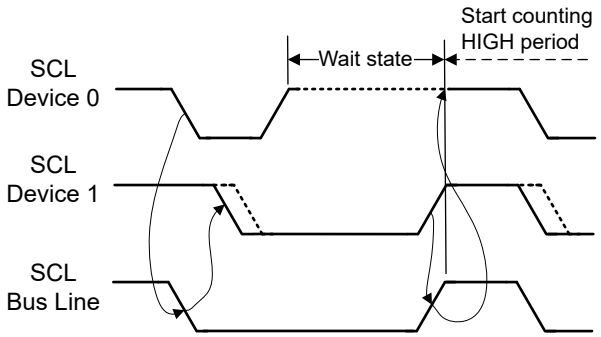


Figure 25-10. Clock Synchronization

25.2.3.13 Glitch Suppression

The UNICOMM I2CC and I2CT modules supports glitch suppression on the SCL and SDA lines up to 50ns, as specified in the I^2C specification.

Analog Glitch Filter

The analog glitch filter feature is only available for Advanced UNICOMM-I2CC and UNICOMM-I2CT instances.

In Advanced UNICOMM-I2C instances, by default, an analog glitch filter is enabled and configured to suppress spikes with a pulse width up to 50ns, configurable via the GFCTL.AGFEN field. The I^2C specification advises to suppress noise spikes of less than 50ns. This filter is disabled by clearing the GFCTL.AGFEN bit.

Digital Glitch Filter

The digital glitch filter feature is only available for Basic UNICOMM-I2CC and UNICOMM-I2CT instances.

In Basic UNICOMM-I2C instances, a digital glitch filter is available. The DGFSEL bits in the GFCTL register can be programmed to provide glitch suppression on the SCL and SDA lines and provide proper signal values. The glitch suppression value is in terms of I2Cclk cycles. To disable the digital glitch filter, set the DGFSEL field to zero. All signals are delayed internally when glitch suppression is nonzero. For example, if the DGFSEL field is set to 0x7, 31 clocks are added onto the calculation for the expected transaction time.

Table 25-3. Glitch Suppression Filter Types

	Analog Glitch Filter	Digital Glitch Filter
Default	Default enabled with 50ns	Default bypassed
UNICOMM Instances	UC2, UC5	UC0, UC1, UC3, UC4
Pulse width of suppressed spikes	Configurable periods: 5ns, 10ns, 25ns, 50ns	Configurable I2Cclk cycles: 1, 2, 3, 4, 8, 16, 31
Benefits	Filter does not depend on the I2Cclk	Programmable length provides extra filtering options and a stable filtering length
Limitation	Variation with temperature and voltage	Only enabled 3 clock cycles after the start of I^2C packet.

Note

Modifications to the GFCTL register must only be done while the CR.ENABLE bit is cleared.

25.2.3.14 Burst Mode

Burst mode is only available for Advanced UNICOMM-I2CC instances.

The UNICOMM-I2CC burst mode allows a sequence of data transfers using DMA or software to transmit or receive the data in the FIFOs. Burst mode is enabled by setting the CTR.BLEN bits in the I2CC CTR register to a value 'N', where 'N' must be greater then '1'. This field sets the number of bytes transferred by a burst. A copy of this value is automatically written to the SR.BCNT bits in the I2CC status register to be used as a down-counter

during burst transfer. The RXDONE and TXDONE interrupt flags are set after a burst of 'N' data bytes has been sent.

In the event that one of the data bytes is NACK'd during an I2CC burst sequence:

- If the CTR.STOP bit *is* set, the frame transfer is terminated. Upon receiving a not acknowledge condition, the I2CC NACK interrupt bit is set in the RIS register and the I2CC automatically sends out a STOP condition on the bus and terminates transfer.
- If the CTR.STOP bit *is not* set, hardware waits for software to proceed further. Upon receiving a not acknowledge condition, the I2CC NACK interrupt bit is set in the RIS register. Software can read the SR.BCNT field to determine the amount of data that was transferred prior to the termination of the burst if desired. Then, software can send out a STOP condition by setting CTR.STOP or a RESTART condition by setting CTR.FRM_START.

25.2.3.15 DMA Operation

The UNICOMM I^2C modules, I2CC and I2CT, each provide an interface to the DMA module with seperate channels for transmit and receive. The DMA operation of the I^2C is enabled through the I^2C DMA Event and DMA module registers. When the DMA functionality is enabled, the I^2C asserts a DMA request on the selected channel when the associated FIFO can transfer or receive data. The DMA transfer requests are handled automatically by the DMA controller based on how the DMA channel is configured (burst size, transfer size, source address etc.).

For more information about the DMA module and DMA-specific configurations, see Direct Memory Access (DMA).

- For the receive channel, a DMA transfer request is asserted when the amount of data in the receive FIFO is at or above the FIFO trigger level configured using the RXIFLSEL bit in IFLS register.
- For the transmit channel, a DMA transfer request is asserted whenever the transmit FIFO contains fewer characters than the FIFO trigger level configured using the TXIFLSEL bit in IFLS register.

25.2.3.16 SMBus 3.0 Support

The System Management Bus (SMBus) protocol is a standard two-wire interface used for industrial applications like battery management, memory SSD and several other use cases. See the *System Management Bus (SMBus) Specification* for more information about the SMBus protocol standards. Since the SMBus protocol reuses a number of features from the standard I^2C protocol, UNICOMM-I2CC peripherals can be used to support the SMBus protocol as well. The additional registers and features supporting the SMBus protocol are only present on UNICOMM instances with Advanced I2CC and I2CT capabilities.

25.2.3.16.1 Quick Command

Quick Command is a simple, compact SMBus transaction that sends an address and 1-bit of data in the DIR bit of the I2C header byte to communicate a command to the target, typically a "turn off" or "turn on". The controller has the ability to send a Quick Command by writing the target address and direction value into the TA register, followed by setting the START, STOP, and FRM_START bits in the CTR register and clearing the BLEN field to 0. The target logic also has quick command support built in.

In UNICOMM-I2C Target mode, the TX FIFO must be empty upon receiving a Quick Command and can only be loaded after decoding the received message. Follow the steps below to receive a Quick Command from an I2CT:

1. TX FIFO must be empty
2. Receive and decode a Quick Command message using the SR.QCMDST and SR.QCMDRW bits.
3. If clock stretching is enabled, software must load the TX FIFO with 0xFF to stop clock stretching and allow the I2C Controller to generate a STOP condition.

Figure 25-11. Quick Command Message

25.2.3.16.2 Acknowledge Control

I2C Target Enhanced ACK Control

The below additonal bits in the ACKCTL register provide enhanced acknowledgement functionality. The status bit SR.ACKOEN is updated automatically when the ACKOEN_ON_PECDONE, ACKOEN_ON_PECNEXT, or ACKOEN_ON_START bits in ACKCTL are set. Software has to write ACKCTL.ACKOEN to '0' to clear the acknowledge override condition generated by hardware.

- **ACKOEN_ON_PECDONE**
  - When set, this bit automatically turns on the target override mechanism following the ACK/NACK of the received PEC byte. This allows software to manually handle data bytes received after the PEC byte. The sequence is as follows:
    1. Set ACKCTL.ACKOEN_ON_PECDONE
    2. SR.ACKOEN bit is automatically set after the PEC byte has been ACK/NACK'd
    3. Receive "n" bytes using manual software ACK
    4. Clear SR.ACKOEN after "n" bytes have been received
    5. Receive next "m" bytes using the automatic hardware ACK
- **ACKOEN_ON_PECNEXT**
  - When set, this bit automatically turns on the target override mechanism following the ACK/NACK of the byte received prior to the PEC byte. Software must clear the SR.ACKOEN after ACKing/NACKing the PEC byte. Using this option, software can effectively get the synchronization point by getting an interrupt here (TREQ/RREQ condition with TXEMPTY_ONTREQ/RXFLL_ON_RREQ set), and then manually send the ACK/NACK of the PEC. In other words, the software doesn't need to rely on byte granularity RXMODE and TXMODE interrupts and get an interrupt after the entire frame has been TX/RX. The sequence is as follows:
    1. Set ACKCTL.ACKOEN_ON_PECNEXT
    2. SR.ACKOEN bit is automatically set before the PEC bytes
    3. Receive the PEC byte and manually ACK/NACK using software
    4. Clear SR.ACKOEN
    5. Receive the entire next frame using automatic hardware ACK until the next PEC byte is received
- **ACKOEN_ON_START**
  - When set, this bit automatically turns on the target override mechanism following a START condition. The sequence is as follows:
    1. Set ACKCTL.ACKOEN_ON_START
    2. SR.ACKOEN bit is automatically set when the START condition is detected
    3. Continue receiving bytes with software until remaining bytes are known
    4. Clear SR.ACKOEN
    5. Receive the rest of the frame using automatic hardware ACK until the next START condition is received

Note

The I2CT software must wait a bit duration between generating a NACK and disabling the SR.ACKOEN. These operations cannot happen in the same cycle.

I2C Controller Enhanced ACK Control

The I2CC, when in receive mode, can ACK/NACK after "n" number of bytes are received. When CTR.ACKOEN is set and the number of bytes configured by CTR.BLEN are received, an RXDONE interrupt is generated and

the transaction is either ACK'd or NACK'd depending on the value of CTR.ACK. To stop the transaction, the CTR.STOP can be set. Otherwise, software can update the CTR.BLEN field to a new value to continue with the transaction.

25.2.3.16.3 Clock Low Timeout Detection

Clock **low** timeout detection is done by setting the TIMEOUT_CTL.TCNTAEN and configuring the timeout period in the TIMEOUT_CNT.TCNTLA (the upper 8-bits of a 12-bit period) register field. The TIMEOUT_CNT.TCNTA register contains the current count value.

Example counter calculations for low SCL timeout detection is shown below.

- For SCL low detect, each count is equal to 520 cycles of the I2Cclk period.
- A pre-multiplier of 520 is used to make the total timeout period greater than 35ms (tTIMEOUT,MAX specified by SMBus 3.0), even at the highest clock frequency of 40MHz.
- At 40MHz, 1 count = 520 * 25ns = 13us
- Max count value = 12'b1111_1111_0000 = 12'd4080
- Total time configurable = 4080 * 13us = 53.04ms

25.2.3.16.4 Clock High Timeout Detection

Clock **high** timeout detection is done by setting the TIMEOUT_CTL.TCNTBEN and configuring the timeout period in the TIMEOUT_CNT.TCNTLB (the upper 8-bits of a 12-bit period) register field. The TIMEOUT_CNT.TCNTB register contains the current count value.

Example counter calculations for high SCL timeout detection is shown below.

- For SCL high detection, each count is equal to 1 * I2Cclk period.
- A pre-multiplier of 1 is used to make the accuracy of the counter remain high for 50 µs ($t_{HIGH,MAX}$) even for low frequency I2Cclk's.
- At 10MHz, 1 count = 100ns
- Total I2Cclk counts required to count 50 µs = 500
- Value of TIMEOUT_CNT.CNTLB for 50µs bus idle timeout = 500/16 = 32

25.2.3.16.5 Cumulative clock low extended timeout for controller and target

The below two cumulative timeout event specifications come from SMBus 3.0 for targets and controllers.

- $t_{LOW:SEXT}$
  - This value is defined as the cummulative time a given target device is allowed to extend the clock cycles in one message from the initial START to the STOP. Another target device, or the controller, can also extend the clock causing the combined clock low extend time to be greater than $t_{LOW:SEXT}$. Therefore, this parameter is measured with the target device as the sole target of a full-speed controller.
- $t_{LOW:MEXT}$
  - This value is defined as the cumulative time a controller device is allowed to extend the clock cycles within each byte of a message as defined from START-to-ACK, ACK-to-ACK, or ACK-to-STOP. A target device or another controller can also extend the clock causing the combined clock low time to be greater than $t_{LOW:MEXT}$ on a given byte. Therefore, this parameter is measured with a full speed target device as the sole target of the controller.

Cumulative timeout events need to be monitored in software for I2CC and I2CT. In most cases, a single clock stretch SCL timeout can act as a sufficient safeguard against cumulative events.

In controller mode, the UNICOMM-I2CC only clock stretches on byte boundaries. This aligns with the definition of $t_{LOW:MEXT}$ and the Counter A low timeout method outlined in Section 25.2.3.16.3 can be used.

In target mode, since clock stretch events can happen at different byte boundaries within a transaction for UNICOMM-I2CT, software must use DMA and FIFO capabilities to avoid exceeding $t_{LOW:SEXT}$. Typically, software takes a longer time to process the first command byte of an SMBUS frame, so the Counter A method can safeguard against software latencies. The rest of the transaction contains byte transfers that must not have software delays.

25.2.3.16.6 Packet Error Checking (PEC)

PEC is a CRC-8 error checking byte which is calculated using all the message bytes of a transaction, including addresses and read/write bits. The ACK/NACK, START, and STOP bits do not contribute to the PEC calculation. The PEC is appended to the message by the device that supplied the last data byte.

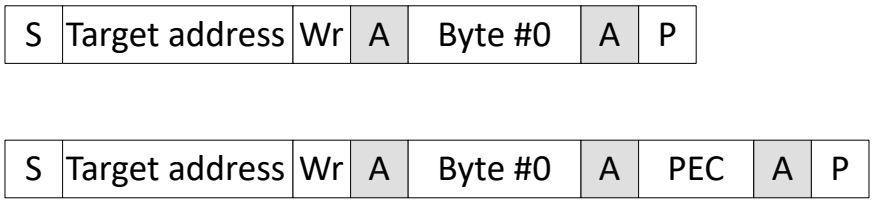


Figure 25-12. Send Byte Message with and without PEC

The PEC Polynomial calculation is the below.

$$x^8 + x^2 + x^1 + 1$$

The PECCTL register on Advanced UNICOMM-I2CC and UNICOMM-I2CT instancessupports PEC transmission and checking for both controller and target modes. Each of the fields from the PECCTL register are described below.

- PECEN
  - This bit enables the SMB Packet Error Checking (PEC) mechanism. When enabled, the PEC is calculated on all bits except the START, STOP, and ACK/NACK.
- PECCNT
  - When this field is set to a non-zero value, the number of I2C data bytes are counted and compared against this value to keep track of each PEC byte in a transmission/reception.

Note

For a I2C Target, dynamic changes made to PECCNT must be done while the SCL is being stretched and only if the previous PECCNT= 0.

Note

For a I2C Controller, dynamic changes made to PECCNT must be done only if the previous PECCNT=0. A PECCNT value of 1 is an invalid configuration as this indicates one PEC byte and zero data bytes in the transaction, which is not possible per the SMBus standard.

For transmitting, an extra byte needs to be inserted into the transmit sequence for each transaction by writing a dummy byte to the TXDATA register. If PECEN is set, I2CC/I2CT calculates the PEC and replaces this dummy byte with the PEC value during transmission when the data byte count reaches the value in PECCNT. For reception, a PEC byte is received the same as a data byte through the RX FIFO. If PECEN is set and the data byte count is equal to the value programmed in PECCNT, the I2CC/I2CT checks the next received byte since that is the PEC. If the PEC byte is non-zero, the PEC_RX_ERR flag in RIS is set and a NACK is transmitted on the line automatically.

The PECSR register is also added to provide status flags relating to PEC operation and functionality for both UNICOMM-I2CC and UNICOMM-I2CT. Each of the fields from the PECSR register are described below.

- PECBYTECNT
  - This is the current PEC Byte Count of the I2C controller or target.
- PECSTS_CHECK
  - This status bit indicates if the PEC was checked in the transaction that occurred before the last STOP. The flag is updated on a STOP condition.
- PECSTS_ERROR

– This status bit indicates if a PEC check error has occurred in the transaction before the last STOP condition. The flag is updated on a STOP condition.

In a typical I2C Target use case, software can set PECEN=1 and PECCNT=0 and enabled the manual ACK override feature, by setting the ACKOEN bit, until the remaining packet length is known. Software then can set the PECCNT to the remaining packet length (including the PEC byte). Once transmission/reception of the packet is complete, manual ACK override mode can be disabled by clearing ACKOEN.

25.2.3.16.7 Host Notify Protocol

To support the SMBus host notify protocol specifications, UNICOMM-I2CT Advanced instances contain a default host address. When the CTR.EN_DEFHOSTADR bit is set, the default host address of **7'b000_1000** is always used by the Target address match logic for comparison. When this bit is cleared, the default host address is not used for matching.

25.2.3.16.8 Alert Response Protocol

To support the SMBus host notify protocol specifications, UNICOMM-I2CT Advanced instances contain an alert response address. When the CTR.EN_ALRESPADR bit is set, the alert response address of **7'b000_1100** is always matched by the Target address match logic. When this bit is cleared, the alert response address is not matched. GPIO signaling for alarm situations must be handled in software.

S	Alert response addr	Rd	A	Target address	NA	P

Figure 25-13. Alert Response Message

25.2.3.16.9 Address Resolution Protocol

SMBus target address conflicts can be resolved by dynamically assigning a new unique address to each target device. The Address Resolution Protocol (ARP) outlined below is used by the UNICOMM-I2CC in this event.

1. Prepare to ARP
   a. Controller sends a **Prepare to ARP** command to all the targets on the bus by using the default device address. All targets must ACK this command and clear own addresses.
2. Get UDID
   a. Controller sends the **Get UDID** command to all the targets on the bus. All targets return the unique UDID back to the controller.
   b. Target arbitration happens since all targets are trying to send a UDID to the controller.
3. Assign address
   a. Controller sends the **Assign Address** command to all the target UDID's to assign each device with a new unique address. The controller needs to keep track of the used address pool and corresponding UDID's in software during this process. All targets need to monitor the UDID bytes and change the own address to the assigned address in the event of a UDID match. For UNICOMM-I2CT instances, this can be done in software. The commander can repeat Step 2 and 3, until there are no ACK's from a target after sending a **Get UDID** command, meaning all ARP-capable targets now have valid target addresses assigned.

Additionally to support the ARP process, UNICOMM-I2CT Advanced instances contain a default device address. When the CTR.EN_DEFDEVADR bit is set, a default device address of **7'b110_0001** is always used by the target address match logic. When this bit is cleared, the default device address is not matched. If arbitration is lost, the RIS.ARBLOST interrupt is set, similar to the arbitration logic of the UNICOMM-I2CC module.

25.2.4 Protocol Descriptions

25.2.4.1 I²C Controller Mode

25.2.4.1.1 I2C Controller Initialization

Follow the high-level UNICOMM configurations in Section 23.3 before executing the below I2CC-specific configurations.

1. Clear the ENABLE bit in the CR register before making any of the below configuration changes.
2. Set the desired SCL clock speed by writing the TPR bit in TPR register with the correct value. For example, with a 20MHz I2Cclk and 100kbps desired SCL clock, a TPR value of 19 (0x13) needs to be written to TPR. For more information about how to calculate the TPR value, refer to *Clock Control*.
3. Configure the desired FIFO trigger levels in the IFLS register.
4. Enable/Disable clock stretching in the CR register.
5. Specify the target address, initial direction of transfer, and addressing mode (7-bit or 10-bit) by writing the TA register.
6. Enable desired interrupts and/or DMA event by using CPU_INT, DMA_TRIG_RX, DMA_TRIG_TX group IMASK registers.
7. (Optionally) Configure the emulation mode for the peripheral in the PDBGCTL register.
8. Enable the I2CC by setting the CR.ENABLE bit.
9. Specify the initial length of transfer, acknowledgement settings, and START/STOP conditions through the CTR register. If operating in controller-transmitter mode, TXDATA can be written with data to be transmitted. Start a frame by writing to the FRM_START bit.

25.2.4.1.2 I2C Controller Status

The application can read the SR register to check the current state of the UNICOMM-I2CC. Status bits that are common across all UNICOMM peripherals and relate to the TX and RX FIFOs are explained in Section 23.2.2.4 from the top-level SPGSS chapter.

25.2.4.1.3 I2C Controller Receive Mode

For an UNICOMM-I2CC to start receiving data out of the idle mode, the user needs to set the START bit in CTR register to generate the START condition. Then the controller automatically sends the START condition followed by the target address as soon as the bus is free.

First, set TA.DIR to 1 to enable receive mode and set CTR.START to 1 to enable the START condition. CTR.BLEN can then be programmed to indicate the number of bytes (n) for the receive operation on Advanced I2CC instances (for Basic UNICOMM-I2CC instances, this length is hardcoded to 1). The CTR.ACK and CTR.STOP bit can then be set or cleared based on user configuration. CTR.FRM_START is set to begin the transaction on the line. The packet format is: START + ADDR + R + (DATA*n) +(ACK/NACK) + STOP. The last data ACK/NACK depends on the ACK bit and an additional STOP condition depends on STOP bit.

After last byte is received, the RXDONE interrupt in the CPU_INT.IIDX register is set to indicate that Controller-Receive transaction is complete. User can use the RXTRG interrupt in CPU_INT.IIDX register to read the data from the RX FIFO. This interrupt triggers when the controller RX FIFO contains >= defined bytes, the trigger level can be defined by using the RXIFSEL bit in IFLS register. The flow chart of controller receiver mode is shown in Figure 25-14.

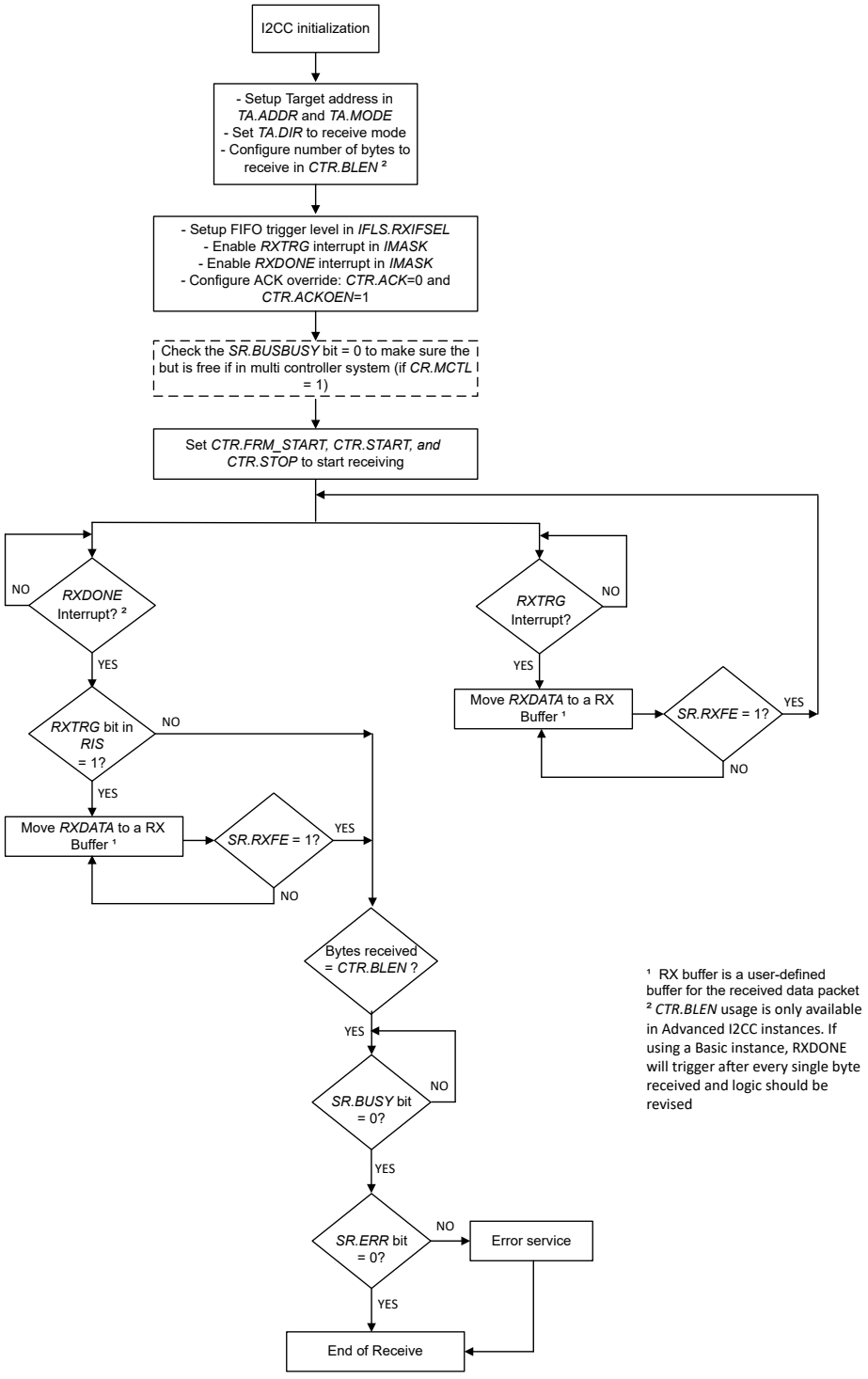


Figure 25-14. Controller Receive Mode

25.2.4.1.4 I2C Controller Transmitter Mode

For a controller to start transmitting data out of idle mode, the user needs to set the START bit in the CTR register to generate the START condition. Then the controller automatically sends the START condition followed by the target address as soon as the bus is free. The data written into TXDATA is transmitted if arbitration is not lost during transmission of the target address.

First, clear TA.DIR to 0 to enable transmit mode and set CTR.START to 1 to enable the START condition. CTR.BLEN can then be programmed to indicate the number of bytes (n) for the transmit operation on Advanced I2CC instances (for Basic instances this length is hardcoded to 1). The CTR.STOP bit can then be set or cleared based on user configuration. CTR.FRM_START is then set to begin the transaction on the line. The packet format is START + ADDR + W + (DATA*n) + STOP, where transmission of the STOP condition depends on the STOP bit.

After the last byte is transmitted, the TXDONE interrupt in the CPU_INT.IIDX register is set to indicate that the controller transmit transaction is completed. The user can also use the TXEMPTY interrupt from the CPU_INT.IIDX register to see if the TX FIFO is empty and ready to load more data. This interrupt triggers if all data in the transmit FIFO has been shifted out. The flow chart with the nexessary software steps for controller-transmitter mode is shown in Figure 25-15.

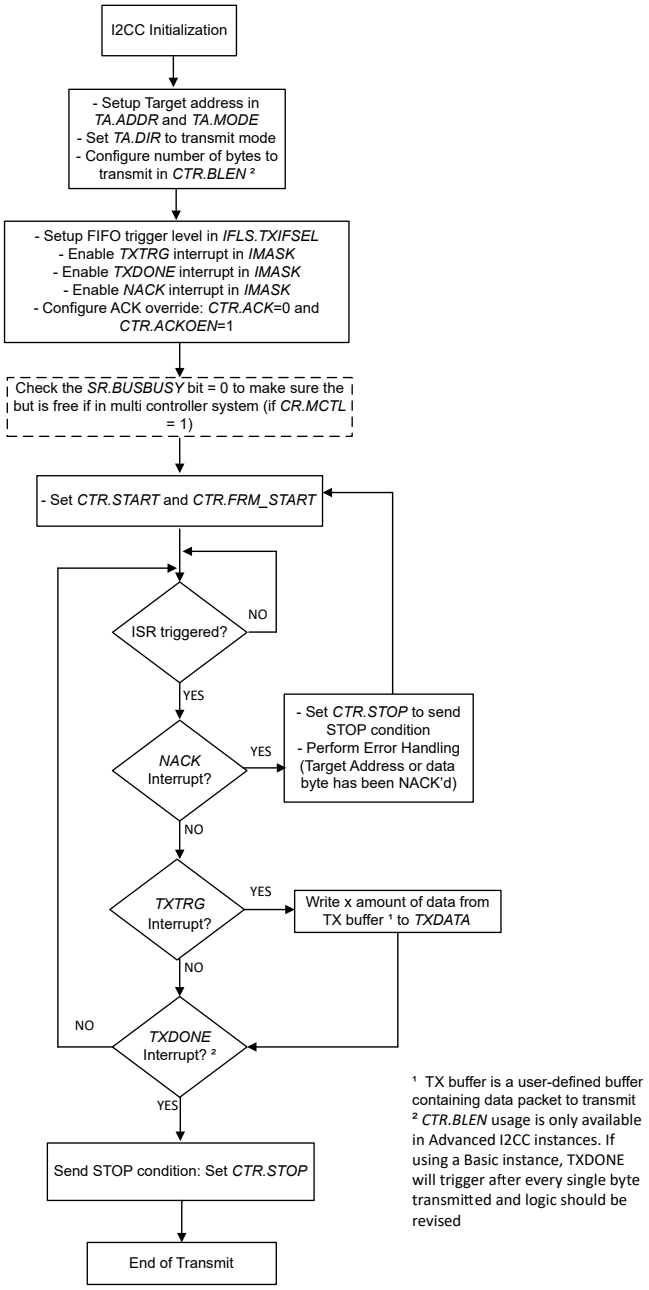


Figure 25-15. Controller Transmitter Mode

25.2.4.1.5 Controller Configuration

The UNICOMM-I2CC control register (CTR) and UNICOMM-I2CC target address register (TA) are used during the application run to setup controller transmit and receive transactions. The following settings can be used to control each transaction before software reads/writes TXDATA and RXDATA.

- Length indicates the number of bytes for the transaction and is configured by CTR.BLEN bits (Only available in Advanced UNICOMM I2CC instances. In Basic I2CC instances, transaction length is fixed at one.)
- Direction (transmit or receive) is configured by TA.DIR bit
- ACK generation is configured by CTR.ACK and CTR.ACKOEN bit.
- STOP condition generation is configured by CTR.STOP bit.
- START or Repeated START condition generation is configured by the CTR.START bit.
- Start of transmission is enabled by the CTR.FRM_START bit.
- Transmit/Receive status and Error status can be checked with the SR.BUSY and SR.ERR bits respectively.

Controller Transmit Data Transactions

Table 25-4. Start Transmit From Idle Mode

Length	Direction	ACK	STOP	START	FRM_START	Frame Format	Notes
n (n>0)	0	X	0 or 1	1	1	START+ADDR+R/W+ DATA*n+STOP	Sending of STOP conndition depends on the CTR.STOP bit.

Table 25-5. Continue Transmit when Last Transmission Finished without STOP

Length	Direction	ACK	STOP	START	FRM_START	Frame Format	Notes
n (n>0)	0	X	0 or 1	0	1	DATA*n+ ACK/ NACK+STOP	Sending of STOP conndition depends on the CTR.STOP bit.

If there is a NACK response from the target, the controller automatically sends out a STOP condition to finish the transmit. The Controller is unable to send a RESTART after a ADDR or DATA NACK.

Controller Receive Data Transactions

Table 25-6. Start Receive from Idle Mode

Length	Direction	ACK	STOP	START	FRM_START	Frame Format	Notes
n (n>0)	1	0 or 1	0 or 1	1	1	START+ADDR +R/ W+DATA *n +ACK/NACK +STOP	The last data ACK or NACK depends on the CTR.ACK bit; additional sending of STOP condition depends on CTR.STOP bit.

Table 25-7. Continue Receive when Last Receive Finished without STOP or NACK

Length	Direction	ACK	STOP	START	FRM_START	Frame Format	Notes
n (n>0)	1	0 or 1	0 or 1	0	1	DATA*n+ ACK/NACK +STOP	The last data ACK or NACK depends on the CTR.ACK bit; additional sending of STOP condition depends on CTR.STOP bit.

This configuration is not allowed if last transaction ended with NACK, since NACK can only be followed by a STOP or RESTART condition. The ACK and STOP bits must not be set to 1 at the same time, as the target needs to be informed to release bus lines before sending out STOP.

Controller Repeated START Transactions

If the last transmit or receive finished without a STOP, a Repeated START can be generated to initiate a new transaction

Table 25-8. Repeated Start Transmit

Length	Direction	ACK	STOP	START	FRM_START	Frame Format	Notes
n (n>0)	0	0 or 1	0 or 1	1	1	Restart+ADDR +R/ W+DATA*n +STOP	Additional sending of STOP depends on CTR.STOP bit.

If there is a NACK response from the target, the controller automatically sends out a STOP condition to finish the transmit. The Controller is unable to send a RESTART after an ADDR or DATA NACK.

Table 25-9. Repeated Start Receive

Length	Direction	ACK	STOP	START	FRM_START	Frame Format	Notes
n (n>0)	1	0 or 1	0 or 1	1	1	Restart+ADDR +R/ W+DATA*n +ACK/ NACK+STOP	The last data followed by ACK or NACK depend on CTR.ACK bit; additional sending of STOP depends on CTR.STOP bit.

The ACK and STOP bits must not be set to 1 at the same time, as the target needs to be informed to release bus lines before sending out STOP.

Controller Transmit STOP-only Transaction

Table 25-10. Send STOP only

Length	Direction	ACK	STOP	START	FRM_START	Frame Format	Notes
n (n>0)	X	X	1	0	1	STOP	STOP command

Only send this transaction after the previous transaction successfully finishes, The STOP condition can't be sent without a NACK to the target if the controller is currently in receive mode.

Controller Quick Command Transaction

The Quick command can only be sent at the beginning of a transaction, not following other transactions (without a STOP) or repeatted start.

Table 25-11. Controller Quick Command

Length	Direction	ACK	STOP	START	FRM_START	Frame Format	Notes
0	0/1	X	1	1	1	START+ADDR +R/ W+STOP	Quick command

25.2.4.2 I^2C Target Mode

25.2.4.2.1 I2C Target Initialization

Follow the high-level UNICOMM configurations in Section 23.3 before executing the below I2CT-specific configurations.

1. Clear the ENABLE bit in the CTR register before making any of the below configuration changes.
2. Configure at least one target address by writing the OAR register. An additional target address can be enabled and configured for Advanced I2CT UNICOMM instances using the OAR2 register.

ADVANCE INFORMATION

3. Configure the desired FIFO trigger levels in the IFLS register.
4. Configure acknowledgement settings in the ACKCTL register.
5. Configure clock stretching, TXTRG/RXTRG interrupt modes, default device addresses, and SMBUS feature settings (only available on Advanced I2CT instances) in the CTR register.
6. Enable desired interrupts and/or DMA event by using CPU_INT, DMA_TRIG_RX, DMA_TRIG_TX group IMASK registers.
7. (Optionally) Configure the emulation mode for the peripheral in the PDBGCTL register.
8. Enable the I2CT by setting the CTR.ENABLE bit.

25.2.4.2.2 I2C Target Status

The application can read the SR register to check the current state of the UNICOMM-I2CT. Status bits that are common across all UNICOMM peripherals and relate to the TX and RX FIFOs are explained in Section 23.2.2.4 from the top-level SPGSS chapter.

25.2.4.2.3 I2C Target Receiver Mode

Target receiver mode is entered when the target address transmitted by the controller matches the I2CT's own address and a **low** R/W bit is received. In target receiver mode, serial data bits received on SDA are shifted in with the clock pulses that are generated by the controller device. The target device does not generate the clock, but can hold SCL low if intervention of the CPU is required after a byte has been received.

After the a data byte is received, the RXDONE interrupt in CPU_INT.IIDX register is set. The I^2C module automatically acknowledges the received data or the user can manually send an acknowledge after each byte received by configuring the ACKCTL register.

When the controller generates a START condition, the START interrupt flag from the CPU_INT.IIDX register is set. When the controller generates a STOP condition, the STOP interrupt flag from the CPU_INT.IIDX register is set.

The user can also use the RXTRG interrupt in CPU_INT.IIDX register to read the data from the receive FIFO. This interrupt triggers when the receive FIFO contains >= defined bytes. The FIFO trigger level can be defined in the RXIFSEL bit in the IFLS register.

The RXDONE approach can be used if the target wants to slow down communication to evaluate reception of every byte, while the RXTRG approach is used to maximize throughput and avoid clock stretching.

The flow chart of using RXDONE and RXFIFOTRG interrupt to read the receive data are shown in Figure 25-16and Figure 25-17 respectively.

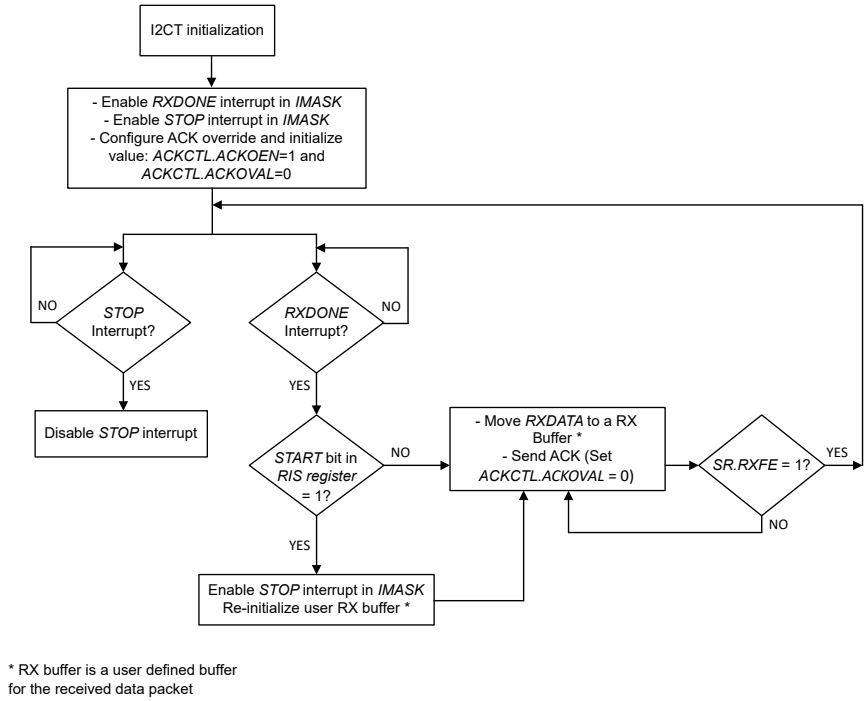


Figure 25-16. Target Receiver Mode using RXDONE and ACK Override

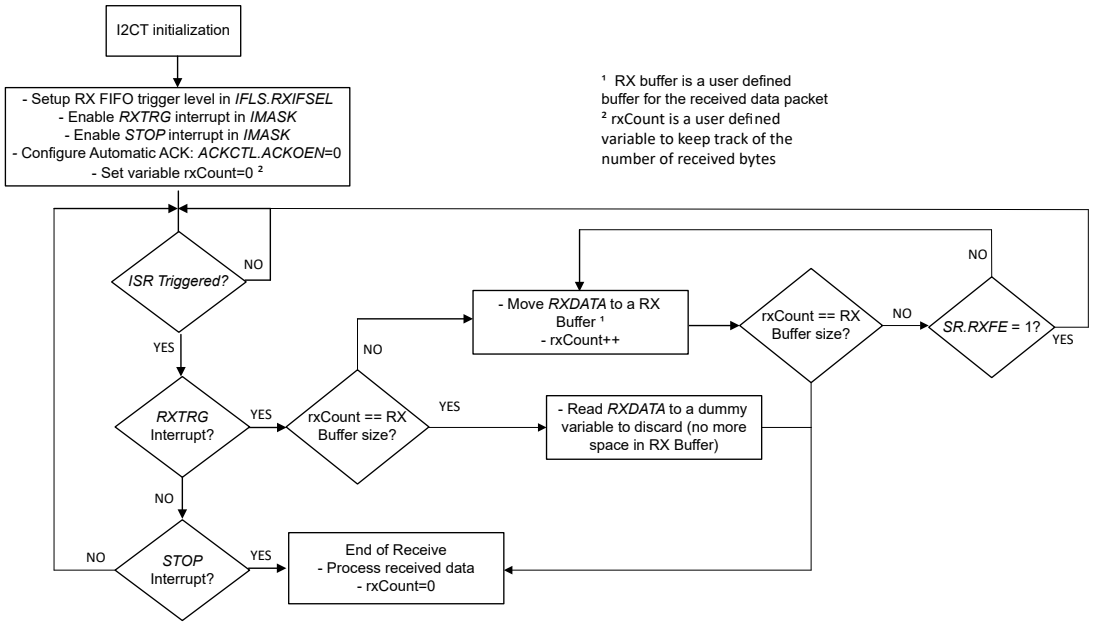


Figure 25-17. Target Receiver Mode using RXTRG and Automatic ACK

25.2.4.2.4 I2C Target Transmitter Mode

Target transmitter mode is entered when the target address transmitted by the controller matches the I2CT's own address with a **high** R/W bit. The target transmitter shifts the serial data out on SDA with the clock pulses that are generated by the controller device. The target device does not generate the clock, but can hold SCL low if intervention of the CPU is required after a byte has been transmitted.

After a data byte is transmitted, the TXDONE interrupt in CPU_INT.IIDX register is set to indicate that a byte has been transmitted.

When the controller generates a START condition, the START interrupt in CPU_INT.IIDX register is set. When the controller generates a STOP condition, the STOP interrupt in CPU_INT.IIDX register is set.

The user can also use the TXTRG interrupt from the CPU_INT.IIDX register to load data to the transmit FIFO. This interrupt triggers when the transmit FIFO contains <= defined bytes. The FIFO trigger level can be defined by using TXIFSEL field in the IFLS register.

The TXDONE approach is used if the target wants to slow down communication to evaluate the transmission of every byte, while the TXTRG approach is used to maximize throughput and avoid clock stretching.

The flow chat of using TXDONE and TXFIFOTRG interrupt to transmit data are shown in Figure 25-18and Figure 25-19.

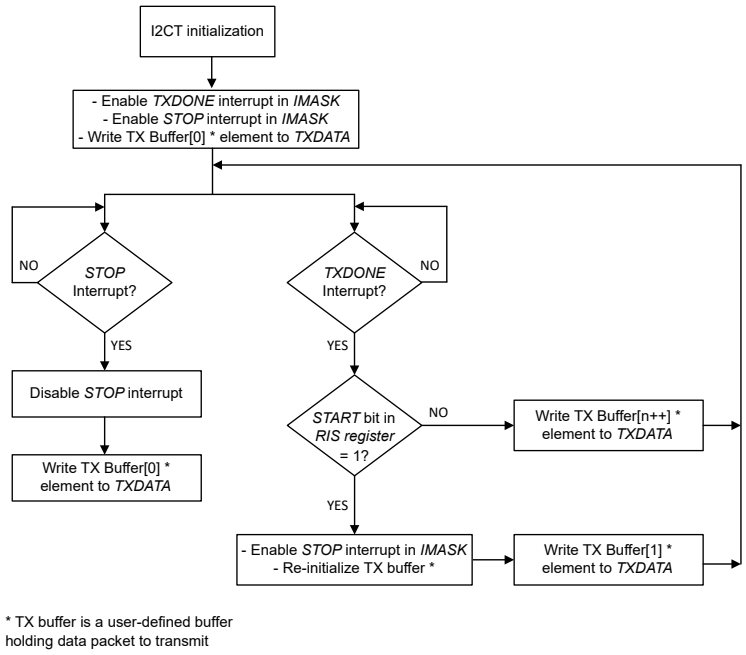


Figure 25-18. Target Transmitter Mode using TXDONE

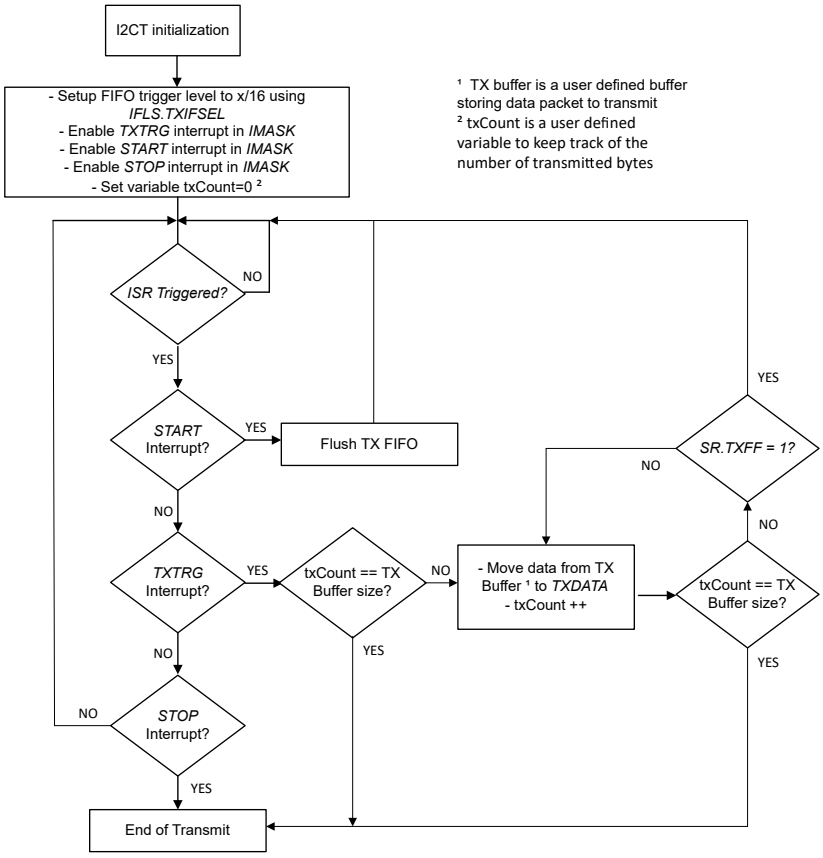


Figure 25-19. Target Transmitter Mode using TXTRG

25.2.5 Reset Considerations

Software Reset Considerations

A Software reset can be executed by setting the RESETASSERT bit together with the KEY bit in the RSTCTL register. Resets must only be issued after terminating a transaction.

Hardware Reset Considerations

A hardware reset also re-initializes the IO configuration. This sets the IOs to a high-impedance state and with the external pullup resistors for I^2C, the lines are pulled high.

Table 25-12 and Table 25-13 shows the behavior of status bits when Controller or Target gets disabled.

Table 25-12. Status Bits when Controller is Disabled

Register	Bit	Behavior when Controller is Disabled	Behavior when Target Disabled	Behavior after Controller/Target is Enabled
SR	BUSY	Reset State	Don't care	Updates START condition sending
	ERR	Reset State	Don't care	Updates on next event detected
	ADRACK	Reset State	Don't care	Updates on next event detected
	DATACK	Reset State	Don't care	Updates on next event detected
	ARBLST	Reset State	Don't care	Updates on next event detected
	IDLE	Reset State	Don't care	Updates on next event detected
	BUSBSY	Reset State	Don't care	Updates on next START detected on bus (or SDA or SCL is low)
	BCNT	Reset State	Don't care	Updates on next event detected
BMON	SCL	Reset State	Don't care	Updates with Controller enable

Table 25-12. Status Bits when Controller is Disabled (continued)

Register	Bit	Behavior when Controller is Disabled	Behavior when Target Disabled	Behavior after Controller/Target is Enabled
	SDA	Reset State	Don't care	Updates with Controller enable

Table 25-13. Status Bits when Target is Disabled

Register	Bit	Behavior When Controller Disabled	Behavior When Target Disabled	Behavior After Controller/Target Enabled
SR	RREQ	Don't care	Reset State	Updates on next event detected
	TREQ	Don't care	Reset State	Updates on next event detected
	OAR2SEL	Don't care	Reset State	Updates on next event detected
	QCMDST	Don't care	Reset State	Updates on next event detected
	QCMDRW	Don't care	Reset State	Updates on next event detected

25.2.6 Interrupt and Events Support

The I²C module contains three event publishers and no event subscribers. One event publisher (CPU_INT) manages I²C interrupt requests (IRQs) to the CPU subsystem through a static event route. The second and third event publishers (DMA_TRIG1, DMA_TRIG0) are used to setup the trigger signaling for the DMA through DMA event route.

The I²C events are summarized in Table 25-14.

Table 25-14. I2C Events

Event	Type	Source	Destination	Configuration	Functionality
Section 13.1.4.1	Publisher	I²C	CPU Subsystem	CPU_INT registers	Fixed interrupt route from I²C to CPU
Section 13.1.4.2	Publisher	I²C	DMA	DMA_TRIG1 registers	Fixed interrupt route from I²C to DMA
Section 13.1.4.2	Publisher	I²C	DMA	DMA_TRIG0 registers	Fixed interrupt route from I²C to DMA

25.2.6.1 CPU Interrupt Event Publisher (CPU_INT)

The I²C module provides 16 interrupt sources for I2C controllers and 17 interrupt sources for I2C targets which can be configured to source a CPU interrupt event. In order of decreasing interrupt priority, the CPU interrupt events from the I²C are:

Table 25-15. I²C CPU Interrupt Event Conditions for Controller (CPU_INT)

IIDX STAT	Name	Description
0x01	RXDONE	Controller receive transaction completed
0x02	TXDONE	Controller transmit transaction completed
0x03	RXTRG	Controller receive trigger, occurs when receive buffer has data
0x04	TXTRG	Controller transmit trigger, occurs when transmit buffer is empty
0x05	RXFULL	Controller RXFIFO full event. This interrupt is set if an RX FIFO is full.
0x07	TXEMPTY	Controller transmit FIFO empty interrupt. This interrupt is set if all data in the Transmit FIFO have been shifted out and the transmit goes into idle mode.

Table 25-15. I²C CPU Interrupt Event Conditions for Controller (CPU_INT) (continued)

IIDX STAT	Name	Description
0x08	NACK	Controller Address/Data NACK interrupt
0x09	START	Controller START detection interrupt
0x0A	STOP	Controller STOP detection interrupt
0x0B	ARBLOST	Controller arbitration lost interrupt
0x0C	PEC_RX_ERR	Controller PEC error occurred
0x0D	TIMEOUTA	Controller Timeout A occurred (clock low timeout)
0x0F	TIMEOUTB	Controller Timeout B occurred (clock high timeout)
0x010	DMA_DONE_RX	Controller DMA TX done
0x011	DMA_DONE_TX	Controller DMA RX done

Table 25-16. I²C CPU Interrupt Event Conditions for Target (CPU_INT)

IIDX STAT	Name	Description
0x01	RXDONE	Target Receive Data Interrupt
0x02	TXDONE	Target Transmit Transaction completed Interrupt
0x03	RXTRG	Target Receive Trigger
0x04	TXTRG	Target Transmit Trigger
0x05	RXFULL	Target RXFIFO full event. This interrupt is set if an Target RX FIFO is full.
0x06	TXEMPTY	Target Transmit FIFO Empty interrupt mask. This interrupt is set if all data in the Transmit FIFO have been shifted out and the transmit goes into idle mode.
0x07	TX_UNFL	Target TX FIFO underflow
0x08	RX_OVFL	Target RX FIFO overflow
0x09	GENCALL	Target General Call Interrupt
0x0A	START	Target Start Condition Interrupt
0x0B	STOP	Target Stop Condition Interrupt
0x0C	PEC_RX_ERR	Target RX Pec Error Interrupt
0x0D	TIMEOUTA	Target Timeout A Interrupt
0x0E	TIMEOUTB	Target Timeout B Interrupt
0x10	DMA_DONE_RX	Target DMA Done on Event Channel RX
0x11	DMA_DONE_TX	Target DMA Done on Event Channel TX
0x12	ARBLOST	Target Arbitration Lost Interrupt

ADVANCE INFORMATION

The CPU interrupt event configuration is managed with the CPU_INT event management registers. See *Event Registers* for guidance on configuring the Event registers for CPU interrupts.

25.2.6.2 DMA Trigger Publisher (DMA_TRIG_RX, DMA_TRIG_TX)

The DMA_TRIG_TX and DMA_TRIG_RX group registers are used to setup trigger signaling for the DMA. These registers (IMASK, RIS, MIS, ISET, ICLR) are present for any UCx configured as an I2C Commander or an I2C Target and can be found in the corresponding UNICOMMI2CC_REGS and UNICOMMI2CT_REGS register structures. See Section 9.2.3 for how the DMA trigger Event works and can be configured. Each DMA channel can be triggered by any of the conditions listed in the Section 9.2.3.1 and can send a DMA_DONE signal to the corresponding UCx module.

Table 25-17. I²C DMA Trigger Condition (DMA_TRIG_RX and DMA_TRIG_TX)

Trigger Name	Register Structure	Register Group	Description
RXTRG	UNICOMMI2CC_REGS	DMA_TRIG_RX	Controller receive FIFO event. Trigger when I2CC-configured UCx module's Receive FIFO contains >= IFLS.RXIFSEL defined bytes

Table 25-17. I²C DMA Trigger Condition (DMA_TRIG_RX and DMA_TRIG_TX) (continued)

Trigger Name	Register Structure	Register Group	Description
TXTRG	UNICOMMI2CC_REGS	DMA_TRIG_TX	Controller transmit FIFO event. Trigger when I2CC-configured UCx module's Transmit FIFO contains <= IFLS.TXIFSEL defined bytes
RXTRG	UNICOMMI2CT_REGS	DMA_TRIG_RX	Target receive FIFO event. Trigger when I2CT-configured UCx module's Receive FIFO contains >= IFLS.RXIFSEL defined bytes
TXTRG	UNICOMMI2CT_REGS	DMA_TRIG_TX	Target transmit FIFO event. Trigger when I2CT-configured UCx module's Transmit FIFO contains <= IFLS.TXIFSEL defined bytes

Take an example where UC0 is configured with the I2CC IPMODE and UC1 is configured with the I2CT IPMODE. To use different DMA channels for each of the operations (controller-transmitter, controller-receiver, target-transmitter, target-receiver) four channels are set up as shown in Figure 25-20. For both UC modules, RXTRG is set as the trigger source for DMA_TRIG_RX and TXTRG is set as the trigger source for DMA_TRIG_TX. The DMA_DONE signal.

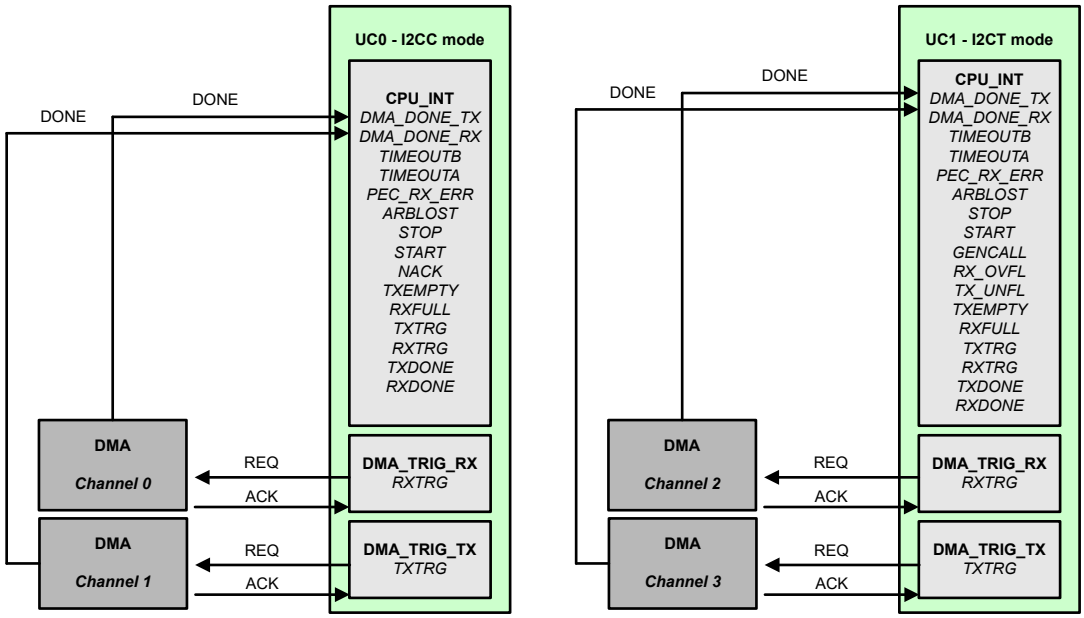


Figure 25-20. I²C DMA Trigger and Status Example

25.2.7 Emulation Modes

The module behavior while the device is in debug mode is controlled by the FREE and SOFT bits in PDBGCTL register.

When the device is in debug mode and set into halt mode below behavior can be configured.

Table 25-18. Debug Mode Peripheral Behavior

PDBGCTL.FREE	PDBGCTL.SOFT	Function
1	**x**	Modules continues operation
0	**0**	Module stops immediately
0	**1**	Module stops after the next transfer has been finished

25.3 UNICOMM-I2C Registers

This Section describes the UNICOMM-I2C Registers.

25.3.1 UNICOMM-I2C Base Address Table

Table 25-19. UNICOMM-I2C Base Address Table

Bit Field Name		DriverLib Name	Base Address
Instance	Structure		
Uc0i2ccRegs	UNICOMMI2CC_REGS	UC0_I2CC_BASE	0x4060_8000
Uc1i2ccRegs	UNICOMMI2CC_REGS	UC1_I2CC_BASE	0x4060_9000
Uc2i2ccRegs	UNICOMMI2CC_REGS	UC2_I2CC_BASE	0x4060_A000
Uc0i2ctRegs	UNICOMMI2CT_REGS	UC0_I2CT_BASE	0x4061_0000
Uc1i2ctRegs	UNICOMMI2CT_REGS	UC1_I2CT_BASE	0x4061_1000
Uc2i2ctRegs	UNICOMMI2CT_REGS	UC2_I2CT_BASE	0x4061_2000
Uc3i2ccRegs	UNICOMMI2CC_REGS	UC3_I2CC_BASE	0x4064_8000
Uc4i2ccRegs	UNICOMMI2CC_REGS	UC4_I2CC_BASE	0x4064_9000
Uc5i2ccRegs	UNICOMMI2CC_REGS	UC5_I2CC_BASE	0x4064_A000
Uc3i2ctRegs	UNICOMMI2CT_REGS	UC3_I2CT_BASE	0x4065_0000
Uc4i2ctRegs	UNICOMMI2CT_REGS	UC4_I2CT_BASE	0x4065_1000
Uc5i2ctRegs	UNICOMMI2CT_REGS	UC5_I2CT_BASE	0x4065_2000

25.3.2 UNICOMMI2CT_REGS Registers

Table 25-20 lists the memory-mapped registers for the UNICOMMI2CT_REGS registers. All register offset addresses not listed in Table 25-20 should be considered as reserved locations and the register contents should not be modified.

Table 25-20. UNICOMMI2CT_REGS Registers

Offset	Acronym	Register Name	Section
0h	CLKDIV	Clock Divider	Go
8h	CLKSEL	Clock Select for Ultra Low Power peripherals	Go
18h	PDBGCTL	Peripheral Debug Control	Go
20h	IIDX	Interrupt index	Go
28h	IMASK	Interrupt mask	Go
30h	RIS	Raw interrupt status	Go
38h	MIS	Masked interrupt status	Go
40h	ISET	Interrupt set	Go
48h	ICLR	Interrupt clear	Go
58h	IMASK	Interrupt mask	Go
60h	RIS	Raw interrupt status	Go
68h	MIS	Masked interrupt status	Go
70h	ISET	Interrupt set	Go
88h	IMASK	Interrupt mask	Go
90h	RIS	Raw interrupt status	Go
98h	MIS	Masked interrupt status	Go
A0h	ISET	Interrupt set	Go
E4h	INTCTL	Interrupt control register	Go
100h	CTR	I2C Target Control Register	Go
104h	ACKCTL	I2C Target ACK Control	Go
108h	SR	Status Register	Go
10Ch	IFLS	Interrupt FIFO Level Select Register	Go
118h	GFCTL	I2C Glitch Filter Control	Go
120h	TTXDATA	I2C TXData	Go
124h	RXDATA	I2C RXData	Go
128h	PECSR	PEC status register	Go
148h	OAR2	Own Address 2	Go
14Ch	OAR	I2C Own Address	Go
150h	TIMEOUT_CNT	I2C Timeout Count Register	Go
154h	TIMEOUT_CTL	I2C Timeout Count Control Register	Go
158h	PECCTL	I2C PEC control register	Go

Complex bit access types are encoded to fit into small table cells. Table 25-21 shows the codes that are used for access types in this section.

Table 25-21. UNICOMMI2CT_REGS Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
Write Type		

Table 25-21. UNICOMMI2CT_REGS Access Type Codes (continued)

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

1 CLKDIV Register (Offset = 0h) [Reset = 00000000h]

CLKDIV is shown in Figure 25-21 and described in Table 25-22.

Return to the Summary Table.

This register is used to specify module-specific divide ratio of the functional clock

Figure 25-21. CLKDIV Register

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

Table 25-23. CLKDIV Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R/W	0h	
2-0	RATIO	R/W	0h	Selects divide ratio of module clock Division factor 0 : DIV_BY_1 1 : DIV_BY_2 .... 63: DIV_BY_64 0h = Do not divide clock source 1h = Divide clock source by 2 2h = Divide clock source by 3 3h = Divide clock source by 4 4h = Divide clock source by 5 5h = Divide clock source by 6 6h = Divide clock source by 7 7h = Divide clock source by 8 3Ch = Divide clock source by 8 3Dh = Divide clock source by 8 3Eh = Divide clock source by 8 3Fh = Divide clock source by 8

2 CLKSEL Register (Offset = 8h) [Reset = 00000000h]

CLKSEL is shown in Figure 25-22 and described in Table 25-23.

Return to the Summary Table.

Clock source selection for peripherals

Figure 25-22. CLKSEL Register

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

Table 25-25. CLKSEL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R/W	0h	
3	BUSCLK_SEL	R/W	0h	Selects BUS CLK as clock source if enabled 0h = Does not select this clock as a source 1h = Select this clock as a source
2-0	RESERVED	R/W	0h	

ADVANCE INFORMATION

3 PDBGCTL Register (Offset = 18h) [Reset = 00000000h]

PDBGCTL is shown in Figure 25-23 and described in Table 25-24.

Return to the Summary Table.

This register can be used by the software developer to control the behavior of the peripheral relative to the 'Core Halted' input

Figure 25-23. PDBGCTL Register

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

Table 25-27. PDBGCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R/W	0h	
1	SOFT	R/W	1h	Soft halt boundary control. This function is only available, if [FREE] is set to 'STOP' 0h = Not supported 1h = The peripheral blocks the debug freeze until it has reached a boundary where it can resume without corruption
0	FREE	R/W	1h	Free run control 0h = The peripheral freezes functionality while the Core Halted input is asserted and resumes when it is deasserted. 1h = The peripheral ignores the state of the Core Halted input

ADVANCE INFORMATION

4 IIDX Register (Offset = 20h) [Reset = 00000000h]

IIDX is shown in Figure 25-24 and described in Table 25-25.

Return to the Summary Table.

This register provides the highest priority enabled interrupt index.

Figure 25-24. IIDX Register

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

Table 25-29. IIDX Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	
7-0	STAT	R	0h	I2C Module Interrupt Vector Value. This register provides the highes priority interrupt index. A read clears the corresponding interrupt flag in RIS and MISC. 15h-1Fh = Reserved 00h = No interrupt pending 1h = Receive Done Flag 2h = Transmit Done Flag 3h = receive FIFO Trigger Level 4h = transmit FIFO Trigger level 5h = RX FIFO FULL Event/interrupt pending 6h = Transmit FIFO/Buffer Empty Event/interrupt pending 7h = Target TX FIFO underflow 8h = Target RX FIFO overflow event 9h = General Call Event Ah = Start Event Bh = Stop Event Ch = PEC receive error event Dh = Timeout A Event Eh = Timeout B Event 10h = DMA DONE on Channel RX 11h = DMA DONE on Channel TX 12h = Arbitration Lost 13h = DMA PRE IRQ INTERRUPT 14h = DMA PRE IRQ INTERRUPT

5 IMASK Register (Offset = 28h) [Reset = 00000000h]

IMASK is shown in Figure 25-25 and described in Table 25-26.

Return to the Summary Table.

Interrupt Mask. If a bit is set, then corresponding interrupt is un-masked. Un-masking the interrupt causes the raw interrupt to be visible in IIDX, as well as MIS.

Figure 25-25. IMASK Register

31	30	29	28	27	26	25	24
RESERVED							
R/W-0h							
23	22	21	20	19	18	17	16
RESERVED						ARBLOST	DMA_DONE_TX
R/W-0h						R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
DMA_DONE_RX	RESERVED	TIMEOUTB	TIMEOUTA	PEC_RX_ERR	STOP	START	GENCALL
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
RX_OVFL	TX_UNFL	TXEMPTY	RXFULL	TXTRG	RXTRG	TXDONE	RXDONE
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 25-31. IMASK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	RESERVED	R/W	0h	
17	ARBLOST	R/W	0h	Arbitration Lost Interrupt 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
16	DMA_DONE_TX	R/W	0h	DMA Done on Event Channel TX 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
15	DMA_DONE_RX	R/W	0h	DMA Done on Event Channel RX 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
14	RESERVED	R/W	0h	
13	TIMEOUTB	R/W	0h	Timeout B Interrupt 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
12	TIMEOUTA	R/W	0h	Timeout A Interrupt 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
11	PEC_RX_ERR	R/W	0h	RX Pec Error Interrupt 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
10	STOP	R/W	0h	Stop Condition Interrupt 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
9	START	R/W	0h	Start Condition Interrupt 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
8	GENCALL	R/W	0h	General Call Interrupt 0h = Clear Interrupt Mask 1h = Set Interrupt Mask

Table 25-31. IMASK Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
7	RX_OVFL	R/W	0h	Target RX FIFO overflow 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
6	TX_UNFL	R/W	0h	Target TX FIFO underflow 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
5	TXEMPTY	R/W	0h	Target Transmit FIFO Empty interrupt mask. This interrupt is set if all data in the Transmit FIFO have been shifted out and the transmit goes into idle mode. 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
4	RXFULL	R/W	0h	RXFIFO full event. This interrupt is set if an Target RX FIFO is full. 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
3	TXTRG	R/W	0h	Transmit Trigger 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
2	RXTRG	R/W	0h	Receive Trigger 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
1	TXDONE	R/W	0h	Transmit Transaction completed Interrupt 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
0	RXDONE	R/W	0h	Receive Data Interrupt Signals that a byte has been received 0h = Clear Interrupt Mask 1h = Set Interrupt Mask

6 RIS Register (Offset = 30h) [Reset = 00000000h]

RIS is shown in Figure 25-26 and described in Table 25-27.

Return to the Summary Table.

Raw interrupt status. Reflects all pending interrupts, regardless of masking. The RIS register allows the user to implement a poll scheme. A flag set in this register can be cleared by writing 1 to the ICLR register bit even if the corresponding IMASK bit is not enabled.

Figure 25-26. RIS Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED						ARBLOST	DMA_DONE_TX
R-0h						R-0h	R-0h
15	14	13	12	11	10	9	8
DMA_DONE_RX	RESERVED	TIMEOUTB	TIMEOUTA	PEC_RX_ERR	STOP	START	GENCALL
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
7	6	5	4	3	2	1	0
RX_OVFL	TX_UNFL	TXEMPTY	RXFULL	TXTRG	RXTRG	TXDONE	RXDONE
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

Table 25-33. RIS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	RESERVED	R	0h	
17	ARBLOST	R	0h	Arbitration Lost Interrupt 0h = Interrupt did not occur 1h = Set Interrupt Mask
16	DMA_DONE_TX	R	0h	DMA Done on Event Channel TX 0h = Clear interrupt 1h = Set interrupt
15	DMA_DONE_RX	R	0h	DMA Done on Event Channel RX 0h = Clear interrupt 1h = Set interrupt
14	RESERVED	R	0h	
13	TIMEOUTB	R	0h	Timeout B Interrupt 0h = Interrupt did not occur 1h = Interrupt occured
12	TIMEOUTA	R	0h	Timeout A Interrupt 0h = Interrupt did not occur 1h = Set Interrupt Mask
11	PEC_RX_ERR	R	0h	RX Pec Error Interrupt 0h = Interrupt did not occur 1h = Interrupt ocuured
10	STOP	R	0h	Stop Condition Interrupt: set on a STOP condition if this target was being addressed 0h = Clear Interrupt 1h = Interrupt is set when target is addressed and STOP condition is received

ADVANCE INFORMATION

Table 25-33. RIS Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
9	START	R	0h	Start Condition Interrupt: is set after a START condition is received and this target is addressed 0h = Clear interrupt 1h = Set when a START condition is receivied and address matches target's address
8	GENCALL	R	0h	General Call Interrupt: set when a general call is received 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
7	RX_OVFL	R	0h	Receive FIFO overflow 0h = Interrupt did not occur 1h = Interrupt Occured
6	TX_UNFL	R	0h	Transmit FIFO underflow 0h = Interrupt did not occur 1h = Interrupt occured
5	TXEMPTY	R	0h	Transmit FIFO Empty Interrupt. This interrupt is set if all data in the Transmit FIFO have been shifted out and FSM goes into idle mode. 0h = Interrupt did not occur 1h = Interrupt occured
4	RXFULL	R	0h	RXFIFO full event. This interrupt is set if an RX FIFO is full. 0h = Clear Interrupt Mask 1h = Interrupt occured
3	TXTRG	R	0h	Transmit Trigger When FIFO is present, triggered as per IFLS settings When FIFO is not present, trigger when transmit buffer is empty 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
2	RXTRG	R	0h	Receive Trigger When FIFO is present, triggered as per IFLS settings when FIFO is not present, this indicates when buffer is full 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
1	TXDONE	R	0h	Transmission Done Interrupt: Set after a byte is transmitted 0h = Interrupt did not occur 1h = Set Interrupt Mask
0	RXDONE	R	0h	Receive Done Interrupt: Set after a byte is received 0h = Interrupt did not occur 1h = Set Interrupt Mask

7 MIS Register (Offset = 38h) [Reset = 00000000h]

MIS is shown in Figure 25-27 and described in Table 25-28.

Return to the Summary Table.

Masked interrupt status. This is an AND of the IMASK and RIS registers.

Figure 25-27. MIS Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED						ARBLOST	DMA_DONE_TX
R-0h						R-0h	R-0h
15	14	13	12	11	10	9	8
DMA_DONE_RX	RESERVED	TIMEOUTB	TIMEOUTA	PEC_RX_ERR	STOP	START	GENCALL
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
7	6	5	4	3	2	1	0
RX_OVFL	TX_UNFL	TXEMPTY	RXFULL	TXTRG	RXTRG	TXDONE	RXDONE
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

Table 25-35. MIS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	RESERVED	R	0h	
17	ARBLOST	R	0h	Arbitration Lost Interrupt 0h = Interrupt did not occur 1h = Set Interrupt Mask
16	DMA_DONE_TX	R	0h	DMA Done on Event Channel TX 0h = Clear MIS 1h = Set MIS
15	DMA_DONE_RX	R	0h	DMA Done on Event Channel RX 0h = Clear MIS 1h = Set MIS
14	RESERVED	R	0h	
13	TIMEOUTB	R	0h	Timeout B Interrupt 0h = Clear interrupt mask 1h = Set interrupt mask
12	TIMEOUTA	R	0h	Timeout A Interrupt 0h = Interrupt did not occur 1h = Set Interrupt Mask
11	PEC_RX_ERR	R	0h	Target RX Pec Error Interrupt 0h = Clear interrupt mask 1h = Set interrupt mask
10	STOP	R	0h	STOP Detection Interrupt 0h = Clear MIS 1h = Set MIS
9	START	R	0h	START Detection Interrupt 0h = Clear MIS 1h = Set MIS
8	GENCALL	R	0h	General Call Interrupt 0h = Interrupt did not occur 1h = Set Interrupt Mask

Table 25-35. MIS Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
7	RX_OVFL	R	0h	Target RX FIFO overflow 0h = Clear interrupt mask 1h = Set interrupt mask
6	TX_UNFL	R	0h	Target TX FIFO underflow 0h = Clear interrupt mask 1h = Set interrupt mask
5	TXEMPTY	R	0h	Transmit FIFO Empty interrupt mask. This interrupt is set if all data in the Transmit FIFO have been shifted out and the transmit goes into idle mode. 0h = Interrupt did not occur 1h = Interrupt occured
4	RXFULL	R	0h	RXFIFO full event. This interrupt is set if an RX FIFO is full. 0h = Clear Interrupt Mask 1h = Interrupt occured
3	TXTRG	R	0h	Transmit Trigger 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
2	RXTRG	R	0h	Receive Trigger 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
1	TXDONE	R	0h	Transmit Transaction completed Interrupt 0h = Interrupt did not occur 1h = Set Interrupt Mask
0	RXDONE	R	0h	Receive Data Interrupt Signals that a byte has been received 0h = Interrupt did not occur 1h = Set Interrupt Mask

8 ISET Register (Offset = 40h) [Reset = 00000000h]

ISET is shown in Figure 25-28 and described in Table 25-29.

Return to the Summary Table.

Interrupt set. Allows interrupts to be set by software (useful in diagnostics and safety checks). Writing a 1 to a bit in ISET will set the event and therefore the related RIS bit also gets set. If the interrupt is enabled through the mask, then the corresponding MIS bit is also set.

Figure 25-28. ISET Register

31	30	29	28	27	26	25	24
RESERVED							
W-0h							
23	22	21	20	19	18	17	16
RESERVED						ARBLOST	DMA_DONE_TX
W-0h						W-0h	W-0h
15	14	13	12	11	10	9	8
DMA_DONE_RX	RESERVED	TIMEOUTB	TIMEOUTA	PEC_RX_ERR	STOP	START	GENCALL
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h
7	6	5	4	3	2	1	0
RX_OVFL	TX_UNFL	TXEMPTY	RXFULL	TXTRG	RXTRG	TXDONE	RXDONE
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h

Table 25-37. ISET Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	RESERVED	W	0h	
17	ARBLOST	W	0h	Arbitration Lost Interrupt 0h = Writing 0 has no effect 1h = Set Interrupt Mask
16	DMA_DONE_TX	W	0h	DMA Done on Event Channel TX 0h = Writing 0 has no effect 1h = Set interrupt
15	DMA_DONE_RX	W	0h	DMA Done on Event Channel RX 0h = Writing 0 has no effect 1h = Set interrupt
14	RESERVED	W	0h	
13	TIMEOUTB	W	0h	Timeout B Interrupt 0h = Writing 0 has no effect 1h = Set interrupt
12	TIMEOUTA	W	0h	Timeout A interrupt 0h = Writing 0 has no effect 1h = Set Interrupt Mask
11	PEC_RX_ERR	W	0h	Target RX Pec Error Interrupt 0h = Writing 0 has no effect 1h = Set interrupt
10	STOP	W	0h	Stop Condition Interrupt 0h = Writing 0 has no effect 1h = Set interrupt
9	START	W	0h	Start Condition Interrupt 0h = Writing 0 has no effect 1h = Set interrupt

Table 25-37. ISET Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
8	GENCALL	W	0h	General Call Interrupt 0h = Writing 0 has no effect 1h = Set Interrupt Mask
7	RX_OVFL	W	0h	Target RX FIFO overflow 0h = Writing 0 has no effect 1h = Set interrupt
6	TX_UNFL	W	0h	Target TX FIFO underflow 0h = Writing 0 has no effect 1h = Set interrupt
5	TXEMPTY	W	0h	Transmit FIFO Empty interrupt mask. This interrupt is set if all data in the Transmit FIFO have been shifted out and the transmit goes into idle mode. 0h = Writing 0 has no effect 1h = Set Interrupt
4	RXFULL	W	0h	RXFIFO full event. This interrupt is set if an RX FIFO is full. 0h = Clear Interrupt Mask 1h = Set Interrupt
3	TXTRG	W	0h	Transmit Trigger 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
2	RXTRG	W	0h	Receive Trigger 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
1	TXDONE	W	0h	Target Transmit Transaction completed Interrupt 0h = Writing 0 has no effect 1h = Set Interrupt Mask
0	RXDONE	W	0h	Target Receive Data Interrupt Signals that a byte has been received 0h = Writing 0 has no effect 1h = Set Interrupt Mask

9 ICLR Register (Offset = 48h) [Reset = 00000000h]

ICLR is shown in Figure 25-29 and described in Table 25-30.

Return to the Summary Table.

Interrupt clear. Write a 1 to clear corresponding Interrupt.

Figure 25-29. ICLR Register

31	30	29	28	27	26	25	24
RESERVED							
W-0h							
23	22	21	20	19	18	17	16
RESERVED						ARBLOST	DMA_DONE_TX
W-0h						W-0h	W-0h
15	14	13	12	11	10	9	8
DMA_DONE_RX	RESERVED	TIMEOUTB	TIMEOUTA	PEC_RX_ERR	STOP	START	GENCALL
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h
7	6	5	4	3	2	1	0
RX_OVFL	TX_UNFL	TXEMPTY	RXFULL	TXTRG	RXTRG	TXDONE	RXDONE
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h

Table 25-39. ICLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	RESERVED	W	0h	
17	ARBLOST	W	0h	Arbitration Lost Interrupt 0h = Writing 0 has no effect 1h = Set Interrupt Mask
16	DMA_DONE_TX	W	0h	DMA Done on Event Channel TX 0h = Writing 0 has no effect 1h = Clear interrupt
15	DMA_DONE_RX	W	0h	DMA Done on Event Channel RX 0h = Writing 0 has no effect 1h = Clear interrupt
14	RESERVED	W	0h	
13	TIMEOUTB	W	0h	Timeout B Interrupt 0h = Writing 0 has no effect 1h = Clear Interrupt
12	TIMEOUTA	W	0h	Timeout A interrupt 0h = Writing 0 has no effect 1h = Set Interrupt Mask
11	PEC_RX_ERR	W	0h	Target RX Pec Error Interrupt 0h = Writing 0 has no effect 1h = Clear Interrupt
10	STOP	W	0h	Target STOP Detection Interrupt 0h = Writing 0 has no effect 1h = Clear interrupt
9	START	W	0h	Target START Detection Interrupt 0h = Writing 0 has no effect 1h = Clear interrupt
8	GENCALL	W	0h	General Call Interrupt 0h = Writing 0 has no effect 1h = Set Interrupt Mask

Table 25-39. ICLR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
7	RX_OVFL	W	0h	Target RX FIFO overflow 0h = Writing 0 has no effect 1h = Clear Interrupt
6	TX_UNFL	W	0h	Target TX FIFO underflow 0h = Writing 0 has no effect 1h = Clear Interrupt
5	TXEMPTY	W	0h	Transmit FIFO Empty interrupt mask. This interrupt is set if all data in the Transmit FIFO have been shifted out and the transmit goes into idle mode. 0h = Writing 0 has no effect 1h = Clear Interrupt
4	RXFULL	W	0h	RXFIFO full event. This interrupt is set if an RX FIFO is full. 0h = Clear Interrupt Mask 1h = Clear Interrupt
3	TXTRG	W	0h	Transmit Trigger 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
2	RXTRG	W	0h	Receive Trigger 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
1	TXDONE	W	0h	Target Transmit Transaction completed Interrupt 0h = Writing 0 has no effect 1h = Set Interrupt Mask
0	RXDONE	W	0h	Target Receive Data Interrupt Signals that a byte has been received 0h = Writing 0 has no effect 1h = Set Interrupt Mask

10 IMASK Register (Offset = 58h) [Reset = 00000000h]

IMASK is shown in Figure 25-30 and described in Table 25-31.

Return to the Summary Table.

Interrupt Mask. If a bit is set, then corresponding interrupt is un-masked. Un-masking the interrupt causes the raw interrupt to be visible in IIDX, as well as MIS.

Figure 25-30. IMASK Register

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

Table 25-41. IMASK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R/W	0h	
2	RXTRG	R/W	0h	Receive Trigger 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
1-0	RESERVED	R/W	0h	

ADVANCE INFORMATION

11 RIS Register (Offset = 60h) [Reset = 00000000h]

RIS is shown in Figure 25-31 and described in Table 25-32.

Return to the Summary Table.

Raw interrupt status. Reflects all pending interrupts, regardless of masking. The RIS register allows the user to implement a poll scheme. A flag set in this register can be cleared by writing 1 to the ICLR register bit even if the corresponding IMASK bit is not enabled.

Figure 25-31. RIS Register

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

Table 25-43. RIS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R	0h	
2	RXTRG	R	0h	Receive Trigger 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
1-0	RESERVED	R	0h	

ADVANCE INFORMATION

12 MIS Register (Offset = 68h) [Reset = 00000000h]

MIS is shown in Figure 25-32 and described in Table 25-33.

Return to the Summary Table.

Masked interrupt status. This is an AND of the IMASK and RIS registers.

Figure 25-32. MIS Register

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

Table 25-45. MIS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R	0h	
2	RXTRG	R	0h	Receive Trigger 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
1-0	RESERVED	R	0h	

ADVANCE INFORMATION

13 ISET Register (Offset = 70h) [Reset = 00000000h]

ISET is shown in Figure 25-33 and described in Table 25-34.

Return to the Summary Table.

Interrupt set. Allows interrupts to be set by software (useful in diagnostics and safety checks). Writing a 1 to a bit in ISET will set the event and therefore the related RIS bit also gets set. If the interrupt is enabled through the mask, then the corresponding MIS bit is also set.

Figure 25-33. ISET Register

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

Table 25-47. ISET Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	W	0h	
2	RXTRG	W	0h	Receive Trigger 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
1-0	RESERVED	W	0h	

ADVANCE INFORMATION

14 IMASK Register (Offset = 88h) [Reset = 00000000h]

IMASK is shown in Figure 25-34 and described in Table 25-35.

Return to the Summary Table.

Interrupt Mask. If a bit is set, then corresponding interrupt is un-masked. Un-masking the interrupt causes the raw interrupt to be visible in IIDX, as well as MIS.

Figure 25-34. IMASK Register

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

Table 25-49. IMASK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R/W	0h	
3	TXTRG	R/W	0h	Transmit Trigger 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
2-0	RESERVED	R/W	0h	

ADVANCE INFORMATION

15 RIS Register (Offset = 90h) [Reset = 00000000h]

RIS is shown in Figure 25-35 and described in Table 25-36.

Return to the Summary Table.

Raw interrupt status. Reflects all pending interrupts, regardless of masking. The RIS register allows the user to implement a poll scheme. A flag set in this register can be cleared by writing 1 to the ICLR register bit even if the corresponding IMASK bit is not enabled.

Figure 25-35. RIS Register

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

Table 25-51. RIS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R	0h	
3	TXTRG	R	0h	Transmit Trigger 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
2-0	RESERVED	R	0h	

16 MIS Register (Offset = 98h) [Reset = 00000000h]

MIS is shown in Figure 25-36 and described in Table 25-37.

Return to the Summary Table.

Masked interrupt status. This is an AND of the IMASK and RIS registers.

Figure 25-36. MIS Register

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

Table 25-53. MIS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R	0h	
3	TXTRG	R	0h	Transmit Trigger 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
2-0	RESERVED	R	0h	

ADVANCE INFORMATION

17 ISET Register (Offset = A0h) [Reset = 00000000h]

ISET is shown in Figure 25-37 and described in Table 25-38.

Return to the Summary Table.

Interrupt set. Allows interrupts to be set by software (useful in diagnostics and safety checks). Writing a 1 to a bit in ISET will set the event and therefore the related RIS bit also gets set. If the interrupt is enabled through the mask, then the corresponding MIS bit is also set.

Figure 25-37. ISET Register

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

Table 25-55. ISET Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	W	0h	
3	TXTRG	W	0h	Transmit Trigger 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
2-0	RESERVED	W	0h	

18 INTCTL Register (Offset = E4h) [Reset = 00000000h]

INTCTL is shown in Figure 25-38 and described in Table 25-39.

Return to the Summary Table.

Interrupt control register

Figure 25-38. INTCTL Register

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

Table 25-57. INTCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	W	0h	
0	INTEVAL	W	0h	Writing a 1 to this field re-evaluates the interrupt sources. 0h = The interrupt or event line is disabled. 1h = The interrupt or event line is in software mode. Software must clear the RIS.

ADVANCE INFORMATION

19 CTR Register (Offset = 100h) [Reset = 00300404h]

CTR is shown in Figure 25-39 and described in Table 25-40.

Return to the Summary Table.

Control Register

Figure 25-39. CTR Register

31	30	29	28	27	26	25	24
RESERVED							
R/W-0h							
23	22	21	20	19	18	17	16
RESERVED		WUEN	CLKSTRETCH	RESERVED			
R/W-0h		R/W-1h	R/W-1h	R/W-0h			
15	14	13	12	11	10	9	8
RESERVED				SUSPEND	RESERVED	EN_DEFDEVADDR	EN_ALRESPADR
R/W-0h				R/W-0h	R/W-1h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
EN_DEFHOSTADR	RXFULL_ON_REQ	TXWAIT_STALE_TXFIFO	TXTRIG_TXMODE	TXEMPTY_ON_TREQ	RESERVED	GENCALL	ENABLE
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-1h	R/W-0h	R/W-0h

Table 25-59. CTR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-22	RESERVED	R/W	0h	
21	WUEN	R/W	1h	Target Wakeup Enable 0h = When 0, the Target is not allowed to clock stretch on START detection 1h = When 1, the Target is allowed to clock stretch on START detection and wait for faster clock to be abvailable. This allows clean wake up support for I2C in low power mode use cases
20	CLKSTRETCH	R/W	1h	Clock Stretch Enable 0h = clock stretching is disabled 1h = clock stretching is enabled
19-12	RESERVED	R/W	0h	
11	SUSPEND	R/W	0h	Suspend external communication 0h = Functional mode resumed 1h = External communication suspended
10	RESERVED	R/W	1h	
9	EN_DEFDEVADR	R/W	0h	Enable Default device address 0h = When this bit is 0, the default device address is not matched. NOTE: it may still be matched if programmed inside SOAR/SOAR2. 1h = When this bit is 1, default device address of 7'h110_0001 is always matched by the Target address match logic.
8	EN_ALRESPADR	R/W	0h	Enable Alert Response Address 0h = When this bit is 0, the alert response address is not matched. NOTE: it may still be matched if programmed inside SOAR/SOAR2 1h = When this bit is 1, alert response address of 7'h000_1100 is always matched by the Target address match logic.
7	EN_DEFHOSTADR	R/W	0h	Enable Default Host Address 0h = When this bit is 0, the default host address is not matched NOTE: it may still be matched if programmed inside SOAR/SOAR2 1h = When this bit is 1, default host address of 7'h000_1000 is always matched by the Target address match logic.

Table 25-59. CTR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
6	RXFULL_ON_RREQ	R/W	0h	Rx full interrupt generated on RREQ condition as indicated in SSR 0h = When 0, RIS:SRXFULL will be set when only the Target RX FIFO is full. This allows the SRXFULL interrupt to be used to indicate that the I2C bus is being clock stretched and that the FW must either read the RX FIFO or ACK/NACK the current Rx byte. 1h = When 1, RIS:SRXFULL will be set when the Target State Machine is in the RX_WAIT or RX_ACK_WAIT states which occurs when the I2C transaction is clock stretched because the RX FIFO is full or the ACKOEN has been set and the state machine is waiting for FW to ACK/NACK the current byte.
5	TXWAIT_STALE_TXFIFO	R/W	0h	Tx transfer waits when stale data in Tx FIFO. This prevents stale bytes left in the TX FIFO from automatically being sent on the next I2C packet. Note: this should be used with TXEMPTY_ON_TREQ set to prevent the Target State Machine from waiting for TX FIFO data without an interrupt notification when the FIFO data is stale. 0h = When 0, the TX FIFO empty signal to the Target State Machine indicates that the TX FIFO is empty. 1h = When 1, the TX FIFO empty signal to the Target State Machine will indicate that the TX FIFO is empty or that the TX FIFO data is stale. The TX FIFO data is determined to be stale when there is data in the TX FIFO when the Target State Machine leaves the TXMODE as defined in the SSR register. This can occur is a Stop or timeout occur when there are bytes left in the TX FIFO.
4	TXTRIG_TXMODE	R/W	0h	Tx Trigger when Target FSM is in Tx Mode 0h = No special behavior 1h = When 1, RIS:TXTRG will be set when the Target TX FIFO has reached the trigger level AND the Target State Machine is in the TXMODE as defined in the SSR register. When cleared RIS:TXTRG will be set when the Target TX FIFO is at or above the trigger level. This setting can be used to hold off the TX DMA until a transaction starts. This allows the DMA to be configured when the I2C is idle but have it wait till the transaction starts to load the Target TX FIFO, so it can load from a memory buffer that might be changing over time.
3	TXEMPTY_ON_TREQ	R/W	0h	Tx Empty Interrupt on TREQ 0h = When 0, RIS:TXEMPTY will be set when only the Target TX FIFO is empty. This allows the TXEMPTY interrupt to be used to indicate that the I2C bus is being clock stretched and that Target TX data is required. 1h = When 1, RIS:TXEMPTY will be set when the Target State Machine is in the TX_WAIT state which occurs when the TX FIFO is empty AND the I2C transaction is clock stretched waiting for the FIFO to receive data.
2	RESERVED	R/W	1h	
1	GENCALL	R/W	0h	General call response enable Modify only when UCSWRST = 1. 0b = Do not respond to a general call 1b = Respond to a general call 0h = Do not respond to a general call 1h = Respond to a general call
0	ENABLE	R/W	0h	Setting this bit enables the module. 0h = Disables module operation. 1h = Enables module operation.

20 ACKCTL Register (Offset = 104h) [Reset = 00000000h]

ACKCTL is shown in Figure 25-40 and described in Table 25-41.

Return to the Summary Table.

This register enables the I2C Target to Not Acknowledge (NACK) for invalid data or command or Acknowledge (ACK) for valid data or command. The I2C clock is pulled low after the last data bit until this register is written.

Figure 25-40. ACKCTL Register

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

Table 25-61. ACKCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-5	RESERVED	R/W	0h	
4	ACKOEN_ON_PECDONE	R/W	0h	When set, this bit will automatically turn on target acknowledge enable field following the ACK/NACK of the received PEC byte. Status bit ACKOEN will reflect a value of '1' when automatic override enable is used. 0h = No special behavior 1h = When set this bit will automatically turn on the Target ACKOEN field following the ACK/NACK of the received PEC byte.
3	ACKOEN_ON_PECNEXT	R/W	0h	When set this bit will automatically turn on the Target acknowledge override following a ACK/NACK of the byte received just prior to the PEC byte. However, setting ACKCTL.ACKOEN bit will not automatically be ACKed/NACKed by the State Machine and firmware must perform this function by writing Target.ACKCTL register. Status bit ACKOEN will reflect a value of '1' when automatic override enable is used. 0h = No special behavior 1h = When set this bit will automatically turn on the Target ACKOEN field following the ACK/NACK of the byte received just prior to the PEC byte. Note that when ACKOEN is set the PEC byte will not automatically be ACKed/NACKed by the State Machine and FW must perform this function by writing Target_SACKCTL.
2	ACKOEN_ON_START	R/W	0h	When set this bit will automatically enable target override acknowlede following a Start Condition. Status bit ACKOEN will reflect a value of '1' when automatic override enable is used. 0h = No special behavior 1h = When set this bit will automatically turn on the Target ACKOEN field following a Start Condition.
1	ACKOVAL	R/W	0h	ACK Override Value Note: Override control is not applicable to Address frame. Bytes following address frame can be acknowledged using ACKOVAL. 0h = An ACK is sent indicating valid data or command. 1h = A NACK is sent indicating invalid data or command.

Table 25-61. ACKCTL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
0	ACKOEN	R/W	0h	I2C Target ACK Override Enable Read SR.ACKOEN to check current status of this bit 0h = A response in not provided. 1h = An ACK or NACK is sent according to the value written to the ACKOVAL bit.

21 SR Register (Offset = 108h) [Reset = 00000000h]

SR is shown in Figure 25-41 and described in Table 25-42.

Return to the Summary Table.

Status register

Figure 25-41. SR Register

31	30	29	28	27	26	25	24
RESERVED						ADDRMATCH	
R-0h						R-0h	
23	22	21	20	19	18	17	16
ADDRMATCH							
R-0h							
15	14	13	12	11	10	9	8
ACKOEN	TXFF	TXFE	RXFF	RXFE	TXCLR	RXCLR	STALE_TXFIFO
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
7	6	5	4	3	2	1	0
TXMODE	BUSBSY	QCMDRW	QCMDST	OAR2SEL	RXMODE	TREQ	RREQ
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

Table 25-63. SR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	RESERVED	R	0h	
25-16	ADDRMATCH	R	0h	Indicates the address for which Target address match happened 0h = Minimum Value 3FFh = Maximum Value
15	ACKOEN	R	0h	Status of ACK Override Enable 0h = ACK override is disabled 1h = ACK override was enabled in design
14	TXFF	R	0h	Transmit FIFO Full The meaning of this bit depends on the state of the FEN bit in the CTL0 register. 0h = The transmitter is not full. 1h = If the FIFO is disabled (FEN is 0), the transmit holding register is full. If the FIFO is enabled (FEN is 1), the transmit FIFO is full.
13	TXFE	R	1h	Transmit FIFO Empty The meaning of this bit depends on the state of the FEN bit in the CTL0 register. 0h = The transmitter has data to transmit. 1h = If the FIFO is disabled (FEN is 0), the transmit holding register is empty. If the FIFO is enabled (FEN is 1), the transmit FIFO is empty.
12	RXFF	R	0h	Receive FIFO Full The meaning of this bit depends on the state of the FEN bit in the CTL0 register. 0h = The receiver can receive data. 1h = If the FIFO is disabled (FEN is 0), the receive holding register is full. If the FIFO is enabled (FEN is 1), the receive FIFO is full.
11	RXFE	R	1h	Receive FIFO Empty The meaning of this bit depends on the state of the FEN bit in the CTL0 register. 0h = The receiver is not empty. 1h = If the FIFO is disabled (FEN is 0), the receive holding register is empty. If the FIFO is enabled (FEN is 1), the receive FIFO is empty.
10	TXCLR	R	0h	TX FIFO Clear Status 0h = FIFO is not cleared 1h = FIFO clear is complete

Table 25-63. SR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
9	RXCLR	R	0h	RX FIFO Clear Status 0h = FIFO is not cleared 1h = FIFO clear is complete
8	STALE_TXFIFO	R	0h	Stale Tx FIFO 0h = Tx FIFO is not stale 1h = The TX FIFO is stale. This occurs when the TX FIFO was not emptied during the previous I2C transaction.
7	TXMODE	R	0h	FSM is in TX MODE 0h = State Machine is not in TX_DATA, TX_WAIT, TX_ACK or ADDR_ACK state with the bus direction set to read. 1h = State Machine is in TX_DATA, TX_WAIT, TX_ACK or ADDR_ACK state with the bus direction set to read.
6	BUSBSY	R	0h	I2C bus is busy 0h = The I2C Bus is not busy 1h = The I2C Bus is busy. This is cleared on a timeout.
5	QCMDRW	R	0h	Quick Command Read / Write This bit only has meaning when the QCMDST bit is set. Value Description: 0: Quick command was a write 1: Quick command was a read 0h = Quick command was a write 1h = Quick command was a read
4	QCMDST	R	0h	Quick Command Status Value Description: 0: The last transaction was a normal transaction or a transaction has not occurred. 1: The last transaction was a Quick Command transaction 0h = The last transaction was a normal transaction or a transaction has not occurred. 1h = The last transaction was a Quick Command transaction.
3	OAR2SEL	R	0h	OAR2 Address Matched This bit gets reevaluated after every address comparison. 0h = Either the OAR2 address is not matched or the match is in legacy mode. 1h = OAR2 address matched and ACKed by the Target.
2	RXMODE	R	0h	Target FSM is in Rx MODE 0h = The Target State Machine is not in the RX_DATA, RX_ACK, RX_WAIT, RX_ACK_WAIT or ADDR_ACK state with the bus direction set to write. 1h = The Target State Machine is in the RX_DATA, RX_ACK, RX_WAIT, RX_ACK_WAIT or ADDR_ACK state with the bus direction set to write.
1	TREQ	R	0h	Transmit Request 0h = No outstanding transmit request. 1h = I2C Target is addressed as a transmitter and is using clock stretching to delay the Controller until data has been written to the TXDATA FIFO (TX FIFO is empty).
0	RREQ	R	0h	Receive Request 0h = No outstanding receive data. 1h = Module has outstanding receive data and is using clock stretching to delay the Controller until the data has been read from the RXDATA FIFO (RX FIFO is full).

22 IFLS Register (Offset = 10Ch) [Reset = 00000022h]

IFLS is shown in Figure 25-42 and described in Table 25-43.

Return to the Summary Table.

The IFLS register is the interrupt FIFO level select register. You can use this register to define the levels at which the TX, RX and timeout interrupt flags are triggered. The interrupts are generated based on a transition through a level rather than being based on the level. That is, the interrupts are generated when the fill level progresses through the trigger level. For example, if the receive trigger level is set to the half-way mark, the interrupt is triggered when the receive FIFO is filled with two or more characters. Out of reset, the TXIFLSEL and RXIFLSEL bits are configured so that the FIFOs trigger an interrupt at the half-way mark.

Figure 25-42. IFLS Register

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

Table 25-65. IFLS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R/W	0h	
7	RXCLR	R/W	0h	RX FIFO CLEAR. Setting this bit will clear the RX FIFO contents. 0h = Disable FIFO clear 1h = Enable FIFO Clear
6-4	RXIFLSEL	R/W	2h	FIFO Level Select for generating events (interrupt/dma). Note: for undefined settings the default configuration is used. 1h = RX FIFO >= 1/4 full 2h = RX FIFO >= 1/2 full (default) 3h = RX FIFO >= 3/4 full 4h = Opposite of empty 5h = RX FIFO is full 6h = RX_FIFO >= (MAX_FIFO_LEN -1) 7h = RX_FIFO <= 1
3	TXCLR	R/W	0h	TX FIFO CLEAR. Setting this bit will clear the TX FIFO contents. 0h = Disable FIFO clear 1h = Enable FIFO Clear
2-0	TXIFLSEL	R/W	2h	FIFO Level Select for generating events (interrupt/dma). Note: for undefined settings the default configuration is used. 1h = TX FIFO <= 3/4 empty 2h = TX FIFO <= 1/2 empty (default) 3h = TX FIFO <= 1/4 empty 4h = Opposite of full 5h = TX FIFO is empty 6h = TX FIFO <= 1 7h = TX_FIFO >= (MAX_FIFO_LEN -1)

23 GFCTL Register (Offset = 118h) [Reset = 00000000h]

GFCTL is shown in Figure 25-43 and described in Table 25-44.

Return to the Summary Table.

This register controls the glitch filter on the SCL and SDA lines

Figure 25-43. GFCTL Register

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

Table 25-67. GFCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	RESERVED	R/W	0h	
8	AGFEN	R/W	0h	Analog Glitch Suppression Enable 0h = Analog Glitch Filter disable 1h = Analog Glitch Filter enable
7-3	RESERVED	R/W	0h	
2-0	DGFSEL	R/W	0h	Glitch Suppression Pulse Width This field controls the pulse width select for glitch suppression on the SCL and SDA lines. The following values are the glitch suppression values in terms of functional clocks. (Core Domain only) 0h = Bypass 1h = 1 clock 2h = 2 clocks 3h = 3 clocks 4h = 4 clocks 5h = 8 clocks 6h = 16 clocks 7h = 31 clocks

24 TTXDATA Register (Offset = 120h) [Reset = 00000000h]

TTXDATA is shown in Figure 25-44 and described in Table 25-45.

Return to the Summary Table.

Transmit Data Register. This register is the transmit data register (the interface to the FIFOs). For transmitted data, if the FIFO is enabled, data written to this location is pushed onto the transmit FIFO. If the FIFO is disabled, data is stored in the transmitter holding register (the bottom word of the transmit FIFO).

Figure 25-44. TTXDATA Register

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

Table 25-69. TTXDATA Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	W	0h	
7-0	DATA	W	0h	Transmit Data This byte contains the data to be transferred during the next transaction. 0h = Smallest value FFh = Highest possible value

25 RXDATA Register (Offset = 124h) [Reset = 00000000h]

RXDATA is shown in Figure 25-45 and described in Table 25-46.

Return to the Summary Table.

RX FIFO Read Data Byte This field contains the current byte being read in the RX FIFO stack. If the FIFO is disabled, the data byte and status are stored in the receiving holding register (the bottom word of the receive FIFO). The received data can be retrieved by reading this register.

Figure 25-45. RXDATA Register

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

Table 25-71. RXDATA Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	
7-0	DATA	R	0h	Received Data. This field contains the last received data. 0h = Smallest value FFh = Highest possible value

26 PECSR Register (Offset = 128h) [Reset = 00000000h]

PECSR is shown in Figure 25-46 and described in Table 25-47.

Return to the Summary Table.

PEC Status Register

Figure 25-46. PECSR Register

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

Table 25-73. PECSR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	RESERVED	R	0h	
17	PECSTS_ERROR	R	0h	This status bit indicates if a PEC check error occurred in the transaction that occurred before the last Stop. Latched on Stop. 0h = Indicates PEC check error did not occurr in the transaction that occurred before the last Stop 1h = Indicates PEC check error occurred in the transaction that occurred before the last Stop
16	PECSTS_CHECK	R	0h	This status bit indicates if the PEC was checked in the transaction that occurred before the last Stop. Latched on Stop. 0h = Indicates PEC was not checked in the transaction that occurred before the last Stop 1h = Indicates PEC was checked in the transaction that occurred before the last Stop
15-9	RESERVED	R	0h	
8-0	PECBYTECNT	R	0h	This is the current PEC Byte Count of the State Machine. 0h = Minimum Value 1FFh = Maximum Value

27 OAR2 Register (Offset = 148h) [Reset = 00000000h]

OAR2 is shown in Figure 25-47 and described in Table 25-48.

Return to the Summary Table.

This register consists of seven address bits that identify the alternate address for the I2C device on the I2C bus.

Figure 25-47. OAR2 Register

31	30	29	28	27	26	25	24
RESERVED							
R/W-0h							
23	22	21	20	19	18	17	16
RESERVED	OAR2_MASK						
R/W-0h	R/W-0h						
15	14	13	12	11	10	9	8
RESERVED							
R/W-0h							
7	6	5	4	3	2	1	0
OAR2EN	OAR2						
R/W-0h	R/W-0h						

Table 25-75. OAR2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-23	RESERVED	R/W	0h	
22-16	OAR2_MASK	R/W	0h	Own Address 2 Mask: This field specifies bits A6 through A0 of the Target address. The bits with value '1' in SOAR2.OAR2_MASK field will make the corresponding incoming address bits to match by default regardless of the value inside SOAR2.OAR2 i.e. corresponding SOAR2.OAR2 bit is a don't care. 0h = Minimum Value 7Fh = Maximum Value
15-8	RESERVED	R/W	0h	
7	OAR2EN	R/W	0h	Own Address 2 Enable 0h = The alternate address is disabled. 1h = Enables the use of the alternate address in the OAR2 field.
6-0	OAR2	R/W	0h	Own Address 2This field specifies the alternate OAR2 address. 0h = Smallest value 7Fh = Highest possible value

28 OAR Register (Offset = 14Ch) [Reset = 00004000h]

OAR is shown in Figure 25-48 and described in Table 25-49.

Return to the Summary Table.

This register consists of seven address bits that identify the I2C device on the I2C bus.

Figure 25-48. OAR Register

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

Table 25-77. OAR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R/W	0h	
15	MODE	R/W	0h	This bit selects the adressing mode to be used. When 0, 7-bit addressing is used. When 1, 10-bit addressing is used. 0h = Enable 7-bit addressing 1h = Enable 10-bit addressing
14	OAREN	R/W	1h	Own Address Enable 0h = Disable OAR address 1h = Enable OAR address
13-10	RESERVED	R/W	0h	
9-0	OAR	R/W	0h	Own Address: This field specifies bits A9 through A0 of the Target address. In 7-bit addressing mode as selected by I2CSOAR.MODE bit, the top 3 bits are don't care 0h = Smallest value 3FFh = Highest possible value

29 TIMEOUT_CNT Register (Offset = 150h) [Reset = 00020002h]

TIMEOUT_CNT is shown in Figure 25-49 and described in Table 25-50.

Return to the Summary Table.

This register contains the upper 8 bits of a 12-bit current counter values for counter A and B. The lower four bits of the counter are not user visible and are always 0h.

Figure 25-49. TIMEOUT_CNT Register

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

Table 25-79. TIMEOUT_CNT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	RESERVED	R	0h	
23-16	TCNTB	R	2h	Timeout Count B Current Count: This field contains the upper 8 bits of a 12-bit current counter for timeout counter B 0h = Smallest Value FFh = Highest possible value
15-8	RESERVED	R	0h	
7-0	TCNTA	R	2h	Timeout Count A Current Count: This field contains the upper 8 bits of a 12-bit current counter for timeout counter A 0h = Smallest Value FFh = Highest possible value

30 TIMEOUT_CTL Register (Offset = 154h) [Reset = 00020002h]

TIMEOUT_CTL is shown in Figure 25-50 and described in Table 25-51.

Return to the Summary Table.

This register contains controls for Timeout Counters A and B

Figure 25-50. TIMEOUT_CTL Register

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

Table 25-81. TIMEOUT_CTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31	TCNTBEN	R/W	0h	Timeout Counter B Enable 0h = Disable Timeout Counter B 1h = Enable Timeout Counter B
30-24	RESERVED	R/W	0h	
23-16	TCNTLB	R/W	2h	Timeout Count B Load: Counter B is used for SCL High Detection. This field contains the upper 8 bits of a 12-bit pre-load value for the Timeout B count. NOTE: The value of CNTLB must be greater than 1h. Each count is equal to 1* clock period. For example, with 10MHz functional clock one timeout period will be equal to1*100ns. 0h = Smallest possible value FFh = Highest possible value
15	TCNTAEN	R/W	0h	Timeout Counter A Enable 0h = Disable Timeout Counter B 1h = Enable Timeout Counter B
14-8	RESERVED	R/W	0h	
7-0	TCNTLA	R/W	2h	Timeout counter A load value Counter A is used for SCL low detection. This field contains the upper 8 bits of a 12-bit pre-load value for the Timeout A count. NOTE: The value of CNTLA must be greater than 1h. Each count is equal to 520 times the timeout period of functional clock. For example, with 8MHz functional clock and a 100KHz operating I2C clock, one timeout period will be equal to (1 / 8MHz) * 520 or 65 us. 0h = Smallest Value FFh = Highest possible value

31 PECCTL Register (Offset = 158h) [Reset = 00000000h]

PECCTL is shown in Figure 25-51 and described in Table 25-52.

Return to the Summary Table.

PEC Control Register

Figure 25-51. PECCTL Register

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

Table 25-83. PECCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-13	RESERVED	R/W	0h	
12	PECEN	R/W	0h	PEC Enable This bit enables the SMB Packet Error Checking (PEC). When enabled the PEC is calculated on all bits except the Start, Stop, Ack and Nack. The PEC LSFR and the Byte Counter is set to 0 when the State Machine is in the IDLE state, which occur following a Stop or when a timeout occurs. The Counter is also set to 0 after the PEC byte is sent or received. Note that the NACK is automatically send following a PEC byte that results in a PEC error. The PEC Polynomial is x^8 + x^2 + x^1 + 1. 0h = PEC transmission and check is disabled 1h = PEC transmission and check is enabled
11-9	RESERVED	R/W	0h	
8-0	PECCNT	R/W	0h	When this field is non zero, the number of I2C data bytes are counted. When the byte count = PECCNT and the state machine is transmitting, the contents of the LSFR is loaded into the shift register instead of the byte received from the Tx FIFO. When the state machine is receiving, after the last bit of this byte is received the LSFR is checked and if it is non-zero, a PEC RX Error interrupt is generated. The I2C packet must be padded to include the PEC byte for both transmit and receive. In transmit mode the FIFO must be loaded with a dummy PEC byte. In receive mode the PEC byte will be passed to the Rx FIFO. In the normal use case, FW would set PECEN=1 and PECCNT=0 and use the ACKOEN until the remaining SMB packet length is known. FW would then set the PECCNT to the remaining packet length (Including PEC bye). FW would then configure DMA to allow the packet to complete unassisted and exit NoAck mode. Note that when the byte count = PEC CNT, the byte count is reset to 0 and multiple PEC calculation can automatically occur within a single I2C transaction 0h = Minimum Value 1FFh = Maximum Value

25.3.3 UNICOMMI2CC_REGS Registers

Table 25-53 lists the memory-mapped registers for the UNICOMMI2CC_REGS registers. All register offset addresses not listed in Table 25-53 should be considered as reserved locations and the register contents should not be modified.

Table 25-84. UNICOMMI2CC_REGS Registers

Offset	Acronym	Register Name	Section
0h	CLKDIV	Clock Divider	Go
8h	CLKSEL	Clock Select for Ultra Low Power peripherals	Go
18h	PDBGCTL	Peripheral Debug Control	Go
20h	IIDX	Interrupt index	Go
28h	IMASK	Interrupt mask	Go
30h	RIS	Raw interrupt status	Go
38h	MIS	Masked interrupt status	Go
40h	ISET	Interrupt set	Go
48h	ICLR	Interrupt clear	Go
58h	IMASK	Interrupt mask	Go
60h	RIS	Raw interrupt status	Go
68h	MIS	Masked interrupt status	Go
70h	ISET	Interrupt set	Go
88h	IMASK	Interrupt mask	Go
90h	RIS	Raw interrupt status	Go
98h	MIS	Masked interrupt status	Go
A0h	ISET	Interrupt set	Go
E4h	INTCTL	Interrupt control register	Go
100h	CTR	Control Register	Go
104h	CR	Configuration	Go
108h	SR	Status Register	Go
10Ch	IFLS	Interrupt FIFO Level Select Register	Go
110h	TPR	Timer Period	Go
118h	GFCTL	I2C Glitch Filter Control	Go
11Ch	BMON	Bus Monitor	Go
120h	TXDATA	TXData	Go
124h	RXDATA	RXData	Go
128h	PECSR	PEC status register	Go
14Ch	TA	Target Address Register	Go
150h	TIMEOUT_CNT	I2C Timeout Count Register	Go
154h	TIMEOUT_CTL	I2C Timeout Count Control Register	Go
158h	PECCTL	I2C PEC control register	Go

Complex bit access types are encoded to fit into small table cells. Table 25-54 shows the codes that are used for access types in this section.

Table 25-85. UNICOMMI2CC_REGS Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read

Table 25-85. UNICOMMI2CC_REGS Access Type Codes (continued)

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

1 CLKDIV Register (Offset = 0h) [Reset = 00000000h]

CLKDIV is shown in Figure 25-52 and described in Table 25-55.

Return to the Summary Table.

This register is used to specify module-specific divide ratio of the functional clock

Figure 25-52. CLKDIV Register

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

Table 25-87. CLKDIV Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R/W	0h	
2-0	RATIO	R/W	0h	Selects divide ratio of module clock Division factor 0 : DIV_BY_1 1 : DIV_BY_2 .... 63: DIV_BY_64 0h = Do not divide clock source 1h = Divide clock source by 2 2h = Divide clock source by 3 3h = Divide clock source by 4 4h = Divide clock source by 5 5h = Divide clock source by 6 6h = Divide clock source by 7 7h = Divide clock source by 8 3Ch = Divide clock source by 8 3Dh = Divide clock source by 8 3Eh = Divide clock source by 8 3Fh = Divide clock source by 8

2 CLKSEL Register (Offset = 8h) [Reset = 00000000h]

CLKSEL is shown in Figure 25-53 and described in Table 25-56.

Return to the Summary Table.

Clock source selection.

Figure 25-53. CLKSEL Register

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

Table 25-89. CLKSEL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R/W	0h	
3	BUSCLK_SEL	R/W	0h	Selects BUS CLK as clock source if enabled 0h = Does not select this clock as a source 1h = Select this clock as a source
2-0	RESERVED	R/W	0h	

ADVANCE INFORMATION

3 PDBGCTL Register (Offset = 18h) [Reset = 00000000h]

PDBGCTL is shown in Figure 25-54 and described in Table 25-57.

Return to the Summary Table.

This register can be used by the software developer to control the behavior of the peripheral relative to the 'Core Halted' input

Figure 25-54. PDBGCTL Register

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

Table 25-91. PDBGCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R/W	0h	
1	SOFT	R/W	1h	Soft halt boundary control. This function is only available, if [FREE] is set to 'STOP' 0h = Not supported 1h = The peripheral blocks the debug freeze until it has reached a boundary where it can resume without corruption
0	FREE	R/W	1h	Free run control 0h = The peripheral freezes functionality while the Core Halted input is asserted and resumes when it is deasserted. 1h = The peripheral ignores the state of the Core Halted input

ADVANCE INFORMATION

4 IIDX Register (Offset = 20h) [Reset = 00000000h]

IIDX is shown in Figure 25-55 and described in Table 25-58.

Return to the Summary Table.

This register provides the highest priority enabled interrupt index.

Figure 25-55. IIDX Register

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

Table 25-93. IIDX Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	
7-0	STAT	R	0h	I2C Module Interrupt Vector Value. This register provides the highes priority interrupt index. A read clears the corresponding interrupt flag in RIS and MISC. 15h-1Fh = Reserved 00h = No interrupt pending 01h = Data received 02h = data transmitted 03h = Receive Trigger 04h = Transmit Trigger 5h = RX FIFO FULL Event/interrupt pending 6h = Transmit FIFO/Buffer Empty Event/interrupt pending 08h = Address/Data NACK 09h = Start Event 0Ah = Stop Event 0Bh = Arbitration Lost Ch = PEC Receive Error Event Dh = Timeout A Event Eh = Timeout B Event 10h = DMA DONE on Channel RX 11h = DMA DONE on Channel TX 13h = DMA PRE IRQ INTERRUPT 14h = DMA PRE IRQ INTERRUPT

5 IMASK Register (Offset = 28h) [Reset = 00000000h]

IMASK is shown in Figure 25-56 and described in Table 25-59.

Return to the Summary Table.

Interrupt Mask. If a bit is set, then corresponding interrupt is un-masked. Un-masking the interrupt causes the raw interrupt to be visible in IIDX, as well as MIS.

Figure 25-56. IMASK Register

31	30	29	28	27	26	25	24
RESERVED							
R/W-0h							
23	22	21	20	19	18	17	16
RESERVED							DMA_DONE_TX
R/W-0h							R/W-0h
15	14	13	12	11	10	9	8
DMA_DONE_RX	RESERVED	TIMEOUTB	TIMEOUTA	PEC_RX_ERR	ARBLOST	STOP	START
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
NACK	RESERVED	TXEMPTY	RXFULL	TXTRG	RXTRG	TXDONE	RXDONE
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 25-95. IMASK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-17	RESERVED	R/W	0h	
16	DMA_DONE_TX	R/W	0h	DMA Done on Event Channel TX 0h = Interrupt disabled 1h = Set Interrupt Mask
15	DMA_DONE_RX	R/W	0h	DMA Done on Event Channel RX 0h = Interrupt disabled 1h = Set Interrupt Mask
14	RESERVED	R/W	0h	
13	TIMEOUTB	R/W	0h	Timeout B Interrupt 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
12	TIMEOUTA	R/W	0h	TIMEOUTA interrupt 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
11	PEC_RX_ERR	R/W	0h	RX Pec Error Interrupt 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
10	ARBLOST	R/W	0h	Arbitration Lost interrupt 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
9	STOP	R/W	0h	STOP interrupt 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
8	START	R/W	0h	START interrupt 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
7	NACK	R/W	0h	NACK interrupt 0h = Clear Interrupt Mask 1h = Set Interrupt Mask

Table 25-95. IMASK Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
6	RESERVED	R/W	0h	
5	TXEMPTY	R/W	0h	Transmit FIFO/Buffer Empty interrupt. 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
4	RXFULL	R/W	0h	RXFIFO full event 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
3	TXTRG	R/W	0h	Transmit Trigger 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
2	RXTRG	R/W	0h	Receive Trigger 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
1	TXDONE	R/W	0h	Transmit Transaction completed Interrupt 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
0	RXDONE	R/W	0h	RXDONE interrupt 0h = Clear Interrupt Mask 1h = Set Interrupt Mask

6 RIS Register (Offset = 30h) [Reset = 00000000h]

RIS is shown in Figure 25-57 and described in Table 25-60.

Return to the Summary Table.

Raw interrupt status. Reflects all pending interrupts, regardless of masking. The RIS register allows the user to implement a poll scheme. A flag set in this register can be cleared by writing 1 to the ICLR register bit even if the corresponding IMASK bit is not enabled.

Figure 25-57. RIS Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							DMA_DONE_TX
R-0h							R-0h
15	14	13	12	11	10	9	8
DMA_DONE_RX	RESERVED	TIMEOUTB	TIMEOUTA	PEC_RX_ERR	ARBLOST	STOP	START
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
7	6	5	4	3	2	1	0
NACK	RESERVED	TXEMPTY	RXFULL	TXTRG	RXTRG	TXDONE	RXDONE
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

Table 25-97. RIS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-17	RESERVED	R	0h	
16	DMA_DONE_TX	R	0h	DMA Done on Event Channel TX 0h = Interrupt disabled 1h = Set Interrupt Mask
15	DMA_DONE_RX	R	0h	DMA Done on Event Channel RX 0h = Interrupt disabled 1h = Set Interrupt Mask
14	RESERVED	R	0h	
13	TIMEOUTB	R	0h	Timeout B Interrupt 0h = Interrupt did not occur 1h = Interrupt occured
12	TIMEOUTA	R	0h	Timeout A Interrupt 0h = Interrupt did not occur 1h = Set Interrupt Mask
11	PEC_RX_ERR	R	0h	Receive PEC Error Interrupt: set when the calculated PEC does not match received PEC 0h = Interrupt did not occur 1h = Interrupt Occured
10	ARBLOST	R	0h	Arbitration Lost Interrupt: in multi-controller systems, when this controller loses out during arbitration 0h = Interrupt did not occur 1h = Set Interrupt Mask
9	STOP	R	0h	STOP Detection Interrupt 0h = Interrupt did not occur 1h = Set Interrupt Mask
8	START	R	0h	START Detection Interrupt 0h = Interrupt did not occur 1h = Set Interrupt Mask

Table 25-97. RIS Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
7	NACK	R	0h	Address/Data NACK Interrupt: set when address or data nack is received 0h = Interrupt did not occur 1h = Set Interrupt Mask
6	RESERVED	R	0h	
5	TXEMPTY	R	0h	Transmit FIFO Empty: set when Transmit FIFO is empty 0h = Interrupt did not occur 1h = Interrupt occured
4	RXFULL	R	0h	RXFIFO full event. This interrupt is set when receive FIFO is full. 0h = Interrupt did not occur 1h = Interrupt occured
3	TXTRG	R	0h	Transmit Trigger When FIFO is present, as per IFLS settings When FIFO is not present, trigger when transmit buffer is empty 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
2	RXTRG	R	0h	Receive Trigger When FIFO is present, as per IFLS settings when FIFO is not present, this indicates when buffer is full 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
1	TXDONE	R	0h	Transmit Done Interrupt: TXDONE interrupt is raised when a burst length completes Or, in case of quick command, when a quick command with R/Wn bit set to '0' 0h = Interrupt did not occur 1h = Set Interrupt Mask
0	RXDONE	R	0h	Receive Done Interrupt: RXDONE interrupt is raised when a burst length completes Or, in case of quick command, when a quick command with R/Wn bit set to '1' 0h = Interrupt did not occur 1h = Set Interrupt Mask

7 MIS Register (Offset = 38h) [Reset = 00000000h]

MIS is shown in Figure 25-58 and described in Table 25-61.

Return to the Summary Table.

Masked interrupt status. This is an AND of the IMASK and RIS registers.

Figure 25-58. MIS Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							DMA_DONE_TX
R-0h							R-0h
15	14	13	12	11	10	9	8
DMA_DONE_RX	RESERVED	TIMEOUTB	TIMEOUTA	PEC_RX_ERR	ARBLOST	STOP	START
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
7	6	5	4	3	2	1	0
NACK	RESERVED	TXEMPTY	RXFULL	TXTRG	RXTRG	TXDONE	RXDONE
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

Table 25-99. MIS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-17	RESERVED	R	0h	
16	DMA_DONE_TX	R	0h	DMA Done on Event Channel TX 0h = Interrupt disabled 1h = Set Interrupt Mask
15	DMA_DONE_RX	R	0h	DMA Done on Event Channel RX 0h = Interrupt disabled 1h = Set Interrupt Mask
14	RESERVED	R	0h	
13	TIMEOUTB	R	0h	Timeout B Interrupt 0h = Clear interrupt mask 1h = Set interrupt mask
12	TIMEOUTA	R	0h	Timeout A Interrupt 0h = Interrupt did not occur 1h = Set Interrupt Mask
11	PEC_RX_ERR	R	0h	RX Pec Error Interrupt 0h = Clear interrupt mask 1h = Set interrupt mask
10	ARBLOST	R	0h	Arbitration Lost Interrupt 0h = Interrupt did not occur 1h = Set Interrupt Mask
9	STOP	R	0h	STOP Detection Interrupt 0h = Interrupt did not occur 1h = Set Interrupt Mask
8	START	R	0h	START Detection Interrupt 0h = Interrupt did not occur 1h = Set Interrupt Mask
7	NACK	R	0h	Address/Data NACK Interrupt 0h = Interrupt did not occur 1h = Set Interrupt Mask

Table 25-99. MIS Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
6	RESERVED	R	0h	
5	TXEMPTY	R	0h	Transmit FIFO Empty masked interrupt. This interrupt is set if all data in the Transmit FIFO have been shifted out and the transmit goes into idle mode. 0h = Interrupt did not occur 1h = Interrupt occured
4	RXFULL	R	0h	Masked RXFIFO full event 0h = Interrupt did not occur 1h = Interrupt occured
3	TXTRG	R	0h	Transmit Trigger 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
2	RXTRG	R	0h	Receive Trigger 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
1	TXDONE	R	0h	Transmit Transaction completed Interrupt 0h = Interrupt did not occur 1h = Set Interrupt Mask
0	RXDONE	R	0h	Receive Data Interrupt 0h = Interrupt did not occur 1h = Set Interrupt Mask

8 ISET Register (Offset = 40h) [Reset = 00000000h]

ISET is shown in Figure 25-59 and described in Table 25-62.

Return to the Summary Table.

Interrupt set. Allows interrupts to be set by software (useful in diagnostics and safety checks). Writing a 1 to a bit in ISET will set the event and therefore the related RIS bit also gets set. If the interrupt is enabled through the mask, then the corresponding MIS bit is also set.

Figure 25-59. ISET Register

31	30	29	28	27	26	25	24
RESERVED							
W-0h							
23	22	21	20	19	18	17	16
RESERVED							DMA_DONE_TX
W-0h							W-0h
15	14	13	12	11	10	9	8
DMA_DONE_RX	RESERVED	TIMEOUTB	TIMEOUTA	PEC_RX_ERR	ARBLOST	STOP	START
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h
7	6	5	4	3	2	1	0
NACK	RESERVED	TXEMPTY	RXFULL	TXTRG	RXTRG	TXDONE	RXDONE
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h

Table 25-101. ISET Register Field Descriptions

Bit	Field	Type	Reset	Description
31-17	RESERVED	W	0h	
16	DMA_DONE_TX	W	0h	DMA Done on Event Channel TX 0h = Interrupt disabled 1h = Set Interrupt Mask
15	DMA_DONE_RX	W	0h	DMA Done on Event Channel RX 0h = Interrupt disabled 1h = Set Interrupt Mask
14	RESERVED	W	0h	
13	TIMEOUTB	W	0h	Timeout B Interrupt 0h = Writing 0 has no effect 1h = Set interrupt
12	TIMEOUTA	W	0h	Timeout A interrupt 0h = Writing 0 has no effect 1h = Set Interrupt Mask
11	PEC_RX_ERR	W	0h	RX Pec Error Interrupt 0h = Writing 0 has no effect 1h = Set interrupt
10	ARBLOST	W	0h	Arbitration Lost Interrupt 0h = Writing 0 has no effect 1h = Set Interrupt Mask
9	STOP	W	0h	STOP Detection Interrupt 0h = Writing 0 has no effect 1h = Set Interrupt Mask
8	START	W	0h	START Detection Interrupt 0h = Writing 0 has no effect 1h = Set Interrupt Mask

ADVANCE INFORMATION

Table 25-101. ISET Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
7	NACK	W	0h	Address/Data NACK Interrupt 0h = Writing 0 has no effect 1h = Set Interrupt Mask
6	RESERVED	W	0h	
5	TXEMPTY	W	0h	Transmit FIFO Empty interrupt mask. This interrupt is set if all data in the Transmit FIFO have been shifted out and the transmit goes into idle mode. 0h = Writing 0 has no effect 1h = Set Interrupt
4	RXFULL	W	0h	RXFIFO full event. 0h = Writing 0 has no effect 1h = Set Interrupt
3	TXTRG	W	0h	Transmit Trigger 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
2	RXTRG	W	0h	Receive Trigger 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
1	TXDONE	W	0h	Transmit Transaction completed Interrupt 0h = Writing 0 has no effect 1h = Set Interrupt Mask
0	RXDONE	W	0h	Receive Data Interrupt Signals that a byte has been received 0h = Writing 0 has no effect 1h = Set Interrupt Mask

9 ICLR Register (Offset = 48h) [Reset = 00000000h]

ICLR is shown in Figure 25-60 and described in Table 25-63.

Return to the Summary Table.

Interrupt clear. Write a 1 to clear corresponding Interrupt.

Figure 25-60. ICLR Register

31	30	29	28	27	26	25	24
RESERVED							
W-0h							
23	22	21	20	19	18	17	16
RESERVED							DMA_DONE_TX
W-0h							W-0h
15	14	13	12	11	10	9	8
DMA_DONE_RX	RESERVED	TIMEOUTB	TIMEOUTA	PEC_RX_ERR	ARBLOST	STOP	START
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h
7	6	5	4	3	2	1	0
NACK	RESERVED	TXEMPTY	RXFULL	TXTRG	RXTRG	TXDONE	RXDONE
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h

Table 25-103. ICLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-17	RESERVED	W	0h	
16	DMA_DONE_TX	W	0h	DMA Done on Event Channel TX 0h = Interrupt disabled 1h = Set Interrupt Mask
15	DMA_DONE_RX	W	0h	DMA Done on Event Channel RX 0h = Interrupt disabled 1h = Set Interrupt Mask
14	RESERVED	W	0h	
13	TIMEOUTB	W	0h	Timeout B Interrupt 0h = Writing 0 has no effect 1h = Clear Interrupt
12	TIMEOUTA	W	0h	Timeout A interrupt 0h = Writing 0 has no effect 1h = Set Interrupt Mask
11	PEC_RX_ERR	W	0h	RX Pec Error Interrupt 0h = Writing 0 has no effect 1h = Clear Interrupt
10	ARBLOST	W	0h	Arbitration Lost Interrupt 0h = Writing 0 has no effect 1h = Set Interrupt Mask
9	STOP	W	0h	STOP Detection Interrupt 0h = Writing 0 has no effect 1h = Set Interrupt Mask
8	START	W	0h	START Detection Interrupt 0h = Writing 0 has no effect 1h = Set Interrupt Mask
7	NACK	W	0h	Address/Data NACK Interrupt 0h = Writing 0 has no effect 1h = Set Interrupt Mask

Table 25-103. ICLR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
6	RESERVED	W	0h	
5	TXEMPTY	W	0h	Transmit FIFO Empty interrupt mask. This interrupt is set if all data in the Transmit FIFO have been shifted out and the transmit goes into idle mode. 0h = Writing 0 has no effect 1h = Clear Interrupt
4	RXFULL	W	0h	RXFIFO full event. 0h = Writing 0 has no effect 1h = Clear Interrupt
3	TXTRG	W	0h	Transmit FIFO Trigger 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
2	RXTRG	W	0h	Receive Trigger 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
1	TXDONE	W	0h	Transmit Transaction completed Interrupt 0h = Writing 0 has no effect 1h = Set Interrupt Mask
0	RXDONE	W	0h	Receive Data Interrupt Signals that a byte has been received 0h = Writing 0 has no effect 1h = Set Interrupt Mask

10 IMASK Register (Offset = 58h) [Reset = 00000000h]

IMASK is shown in Figure 25-61 and described in Table 25-64.

Return to the Summary Table.

Interrupt Mask. If a bit is set, then corresponding interrupt is un-masked. Un-masking the interrupt causes the raw interrupt to be visible in IIDX, as well as MIS.

Figure 25-61. IMASK Register

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

Table 25-105. IMASK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R/W	0h	
2	RXTRG	R/W	0h	Receive Trigger 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
1-0	RESERVED	R/W	0h	

ADVANCE INFORMATION

11 RIS Register (Offset = 60h) [Reset = 00000000h]

RIS is shown in Figure 25-62 and described in Table 25-65.

Return to the Summary Table.

Raw interrupt status. Reflects all pending interrupts, regardless of masking. The RIS register allows the user to implement a poll scheme. A flag set in this register can be cleared by writing 1 to the ICLR register bit even if the corresponding IMASK bit is not enabled.

Figure 25-62. RIS Register

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

Table 25-107. RIS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R	0h	
2	RXTRG	R	0h	Receive Trigger When FIFO is present, trigger when RX FIFO contains >= defined bytes when FIFO is not present, this indicates when buffer is full 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
1-0	RESERVED	R	0h	

12 MIS Register (Offset = 68h) [Reset = 00000000h]

MIS is shown in Figure 25-63 and described in Table 25-66.

Return to the Summary Table.

Masked interrupt status. This is an AND of the IMASK and RIS registers.

Figure 25-63. MIS Register

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

Table 25-109. MIS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R	0h	
2	RXTRG	R	0h	Receive Trigger 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
1-0	RESERVED	R	0h	

ADVANCE INFORMATION

13 ISET Register (Offset = 70h) [Reset = 00000000h]

ISET is shown in Figure 25-64 and described in Table 25-67.

Return to the Summary Table.

Interrupt set. Allows interrupts to be set by software (useful in diagnostics and safety checks). Writing a 1 to a bit in ISET will set the event and therefore the related RIS bit also gets set. If the interrupt is enabled through the mask, then the corresponding MIS bit is also set.

Figure 25-64. ISET Register

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

Table 25-111. ISET Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	W	0h	
2	RXTRG	W	0h	Receive Trigger 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
1-0	RESERVED	W	0h	

ADVANCE INFORMATION

14 IMASK Register (Offset = 88h) [Reset = 00000000h]

IMASK is shown in Figure 25-65 and described in Table 25-68.

Return to the Summary Table.

Interrupt Mask. If a bit is set, then corresponding interrupt is un-masked. Un-masking the interrupt causes the raw interrupt to be visible in IIDX, as well as MIS.

Figure 25-65. IMASK Register

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

Table 25-113. IMASK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R/W	0h	
3	TXTRG	R/W	0h	Transmit Trigger 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
2-0	RESERVED	R/W	0h	

ADVANCE INFORMATION

15 RIS Register (Offset = 90h) [Reset = 00000000h]

RIS is shown in Figure 25-66 and described in Table 25-69.

Return to the Summary Table.

Raw interrupt status. Reflects all pending interrupts, regardless of masking. The RIS register allows the user to implement a poll scheme. A flag set in this register can be cleared by writing 1 to the ICLR register bit even if the corresponding IMASK bit is not enabled.

Figure 25-66. RIS Register

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

Table 25-115. RIS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R	0h	
3	TXTRG	R	0h	Transmit Trigger 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
2-0	RESERVED	R	0h	

ADVANCE INFORMATION

16 MIS Register (Offset = 98h) [Reset = 00000000h]

MIS is shown in Figure 25-67 and described in Table 25-70.

Return to the Summary Table.

Masked interrupt status. This is an AND of the IMASK and RIS registers.

Figure 25-67. MIS Register

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

Table 25-117. MIS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R	0h	
3	TXTRG	R	0h	Transmit Trigger 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
2-0	RESERVED	R	0h	

ADVANCE INFORMATION

17 ISET Register (Offset = A0h) [Reset = 00000000h]

ISET is shown in Figure 25-68 and described in Table 25-71.

Return to the Summary Table.

Interrupt set. Allows interrupts to be set by software (useful in diagnostics and safety checks). Writing a 1 to a bit in ISET will set the event and therefore the related RIS bit also gets set. If the interrupt is enabled through the mask, then the corresponding MIS bit is also set.

Figure 25-68. ISET Register

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

Table 25-119. ISET Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	W	0h	
3	TXTRG	W	0h	Transmit Trigger 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
2-0	RESERVED	W	0h	

ADVANCE INFORMATION

18 INTCTL Register (Offset = E4h) [Reset = 00000000h]

INTCTL is shown in Figure 25-69 and described in Table 25-72.

Return to the Summary Table.

Interrupt control register

Figure 25-69. INTCTL Register

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

Table 25-121. INTCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	W	0h	
0	INTEVAL	W	0h	Writing a 1 to this field re-evaluates the interrupt sources. 0h = The interrupt or event line is disabled. 1h = The interrupt or event line is in software mode. Software must clear the RIS.

ADVANCE INFORMATION

19 CTR Register (Offset = 100h) [Reset = 00000000h]

CTR is shown in Figure 25-70 and described in Table 25-73.

Return to the Summary Table.

This control register configures the I2C controller operation. The START bit generates the START or REPEATED START condition. The STOP bit determines if the cycle stops at the end of the data cycle or continues to the next transfer cycle, which could be a repeated START. To generate a single transmit cycle, the Target Address (TA) register is written with the desired address, the RS bit is cleared, and this register is written with ACK = X (0 or 1), STOP = 1, START = 1, and RUN = 1 to perform the operation and stop. When the operation is completed (or aborted due an error), an byte transaction completed interrupt becomes active and the data may be read from the RXDATA register. When the I2C module operates in Controller receiver mode, a set ACK bit causes the I2C bus controller to transmit an acknowledge automatically after each byte. This bit must be cleared when the I2C bus controller requires no further data to be transmitted from the Target transmitter.

Figure 25-70. CTR Register

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

Table 25-123. CTR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	RESERVED	R/W	0h	
27-16	BLEN	R/W	0h	I2C transaction length This field contains the programmed length of bytes of the Transaction. 0h = Smallest value FFFh = Highest possible value
15-7	RESERVED	R/W	0h	
6	SUSPEND	R/W	0h	Suspend external communication 0h = Functional mode resumed 1h = External communication suspended
5	RD_ON_TXEMPTY	R/W	0h	Read on TX Empty 0h = No special behavior 1h = When 1 the Controller will transmit all bytes from the TX FIFO before continuing with the programmed Burst Run Read. If the DIR is not set to Read in the MSA then this bit is ignored. The Start must be set in the MCTR for proper I2C protocol. The Controller will first send the Start Condition, I2C Address with R/W bit set to write, before sending the bytes in the TX FIFO. When the TX FIFO is empty, the I2C transaction will continue as programmed in MTCR and MSA without sending a Stop Condition. This is intended to be used to perform simple I2C command based reads transition that will complete after initiating them without having to get an interrupt to turn the bus around.

Table 25-123. CTR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
4	ACKOEN	R/W	0h	ACK overrride Enable 0h = No special behavior 1h = When 1 and the Controller is receiving data and the number of bytes indicated in MBLEN have been received, the state machine will generate an rxdone interrupt and wait at the start of the ACK for FW to indicate if an ACK or NACK should be sent. The ACK or NACK is selected by writing the MCTR register and setting ACK accordingly. The other fields in this register can also be written at this time to continue on with the transaction. If a NACK is sent the state machine will automatically send a Stop.
3	ACK	R/W	0h	Data Acknowledge Enable. Software needs to configure this bit to send the ACK or NACK. See field decoding in Table: MCTR Field decoding. 0h = The last received data byte of a transaction is not acknowledged automatically . 1h = The last received data byte of a transaction is acknowledged automatically .
2	STOP	R/W	0h	Generate STOP 0h = The controller does not generate the STOP condition. 1h = The controller generates the STOP condition. See field decoding in Table: MCTR Field decoding.
1	START	R/W	0h	Generate START 0h = The controller does not generate the START condition. 1h = The controller generates the START or repeated START condition. See field decoding in Table: MCTR Field decoding.
0	FRM_START	R/W	0h	Start Transfer 0h = In standard mode, this encoding means the Controller is unable to transmit or receive data. 1h = The Controller is able to transmit or receive data. See field decoding in Table: MCTR Field decoding.

20 CR Register (Offset = 104h) [Reset = 00000000h]

CR is shown in Figure 25-71 and described in Table 25-74.

Return to the Summary Table.

Configuration register

Figure 25-71. CR Register

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

Table 25-125. CR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R/W	0h	
2	CLKSTRETCH	R/W	0h	Clock Stretching. This bit controls the support for clock stretching of the I2C bus. 0h = Disables the clock stretching detection. This can be disabled if no Target on the bus does support clock stretching, so that the maximum speed on the bus can be reached. 1h = Enables the clock stretching detection. Enabling the clock stretching ensures compliance to the I2C standard but could limit the speed due the clock stretching.
1	MCTL	R/W	0h	MultiController mode. In MultiController mode the SCL high time counts once the SCL line has been detected high. If this is not enabled the high time counts as soon as the SCL line has been set high by the I2C controller. 0h = Disable MultiController mode. 1h = Enable MultiController mode.
0	ENABLE	R/W	0h	Enable module. After this bit has been set, it should not be set again unless it has been cleared by writing a 0 or by a reset, otherwise transfer failures may occur. 0h = Disables operation. 1h = Enables operation.

21 SR Register (Offset = 108h) [Reset = 00000000h]

SR is shown in Figure 25-72 and described in Table 25-75.

Return to the Summary Table.

The status register indicates the state of the I2C bus controller.

Figure 25-72. SR Register

31	30	29	28	27	26	25	24
RESERVED				BCNT			
R-0h				R-0h			
23	22	21	20	19	18	17	16
BCNT							
R-0h							
15	14	13	12	11	10	9	8
RESERVED	TXFF	TXFE	RXFF	RXFE	TXCLR	RXCLR	RESERVED
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
7	6	5	4	3	2	1	0
RESERVED	BUSBSY	IDLE	ARBLST	DATACK	ADRACK	ERR	BUSY
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

Table 25-127. SR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	RESERVED	R	0h	
27-16	BCNT	R	0h	Transaction Count This field contains the current count-down value of the transaction. 0h = Smallest value FFFh = Highest possible value
15	RESERVED	R	0h	
14	TXFF	R	0h	Transmit FIFO Full The meaning of this bit depends on the state of the FEN bit in the CTL0 register. 0h = The transmitter is not full. 1h = If the FIFO is disabled (FEN is 0), the transmit holding register is full. If the FIFO is enabled (FEN is 1), the transmit FIFO is full.
13	TXFE	R	1h	Transmit FIFO Empty The meaning of this bit depends on the state of the FEN bit in the CTL0 register. 0h = The transmitter has data to transmit. 1h = If the FIFO is disabled (FEN is 0), the transmit holding register is empty. If the FIFO is enabled (FEN is 1), the transmit FIFO is empty.
12	RXFF	R	0h	Receive FIFO Full The meaning of this bit depends on the state of the FEN bit in the CTL0 register. 0h = The receiver can receive data. 1h = If the FIFO is disabled (FEN is 0), the receive holding register is full. If the FIFO is enabled (FEN is 1), the receive FIFO is full.
11	RXFE	R	1h	Receive FIFO Empty The meaning of this bit depends on the state of the FEN bit in the CTL0 register. 0h = The receiver is not empty. 1h = If the FIFO is disabled (FEN is 0), the receive holding register is empty. If the FIFO is enabled (FEN is 1), the receive FIFO is empty.
10	TXCLR	R	0h	TX FIFO Clear Status 0h = FIFO is not cleared 1h = FIFO clear is complete
9	RXCLR	R	0h	RX FIFO Clear Status 0h = FIFO is not cleared 1h = FIFO clear is complete

Table 25-127. SR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
8-7	RESERVED	R	0h	
6	BUSBSY	R	0h	I2C Bus is Busy Controller State Machine will wait until this bit to be cleared before starting a transaction. When first enabling the Controller in multi Controller environments, FW should wait for one I2C clock period after setting ACTIVE high before writing to the CTR register to start the transaction so that if SCL goes low it will trigger the BUSBSY. 0h = The I2C bus is idle. 1h = 'This Status bit is set on a START or when SCL goes low. It is cleared on a STOP, or when a SCL high bus busy timeout occurs and SCL and SDA are both high. This status is cleared when the ACTIVE bit is low. Note that the Controller State Machine will wait until this bit is cleared before starting an I2C transaction. When first enabling the Controller in multi Controller environments, FW should wait for one I2C clock period after setting ACTIVE high before writing to the MTCR register to start the transaction so that if SCL goes low it will trigger the BUSBSY.
5	IDLE	R	1h	I2C Idle 0h = The I2C controller is not idle. 1h = The I2C controller is idle.
4	ARBLST	R	0h	Arbitration Lost 0h = The I2C controller won arbitration. 1h = The I2C controller lost arbitration.
3	DATACK	R	0h	Acknowledge Data 0h = The transmitted data was acknowledged 1h = The transmitted data was not acknowledged.
2	ADRACK	R	0h	Acknowledge Address 0h = The transmitted address was acknowledged 1h = The transmitted address was not acknowledged.
1	ERR	R	0h	Error The error can be from the Target address not being acknowledged or the transmit data not being acknowledged. 0h = No error was detected on the last operation. 1h = An error occurred on the last operation.
0	BUSY	R	0h	FSM Busy The BUSY bit is set during an ongoing transaction, so is set during the transmit/receive of the amount of data set in MBLEN including START, RESTART, Address and STOP signal generation when required for the current transaction. 0h = The controller is idle. 1h = The controller is busy.

22 IFLS Register (Offset = 10Ch) [Reset = 00000022h]

IFLS is shown in Figure 25-73 and described in Table 25-76.

Return to the Summary Table.

The IFLS register is the interrupt FIFO level select register. You can use this register to define the levels at which the TX, RX and timeout interrupt flags are triggered. The interrupts are generated based on a transition through a level rather than being based on the level. That is, the interrupts are generated when the fill level progresses through the trigger level. For example, if the receive trigger level is set to the half-way mark, the interrupt is triggered when the receive FIFO is filled with two or more characters. Out of reset, the TXIFLSEL and RXIFLSEL bits are configured so that the FIFOs trigger an interrupt at the half-way mark.

Figure 25-73. IFLS Register

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

Table 25-129. IFLS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R/W	0h	
7	RXCLR	R/W	0h	RX FIFO CLEAR. Setting this bit will clear the RX FIFO contents. 0h = Disable FIFO clear 1h = Enable FIFO Clear
6-4	RXIFLSEL	R/W	2h	FIFO Level Select for generating events (interrupt/dma). Note: for undefined settings the default configuration is used. 1h = RX FIFO >= 1/4 full 2h = RX FIFO >= 1/2 full (default) 3h = RX FIFO >= 3/4 full 4h = Opposite of empty 5h = RX FIFO is full 6h = RX_FIFO >= (MAX_FIFO_LEN -1) 7h = RX_FIFO <= 1
3	TXCLR	R/W	0h	TX FIFO CLEAR. Setting this bit will clear the TX FIFO contents. 0h = Disable FIFO clear 1h = Enable FIFO Clear
2-0	TXIFLSEL	R/W	2h	FIFO Level Select for generating events (interrupt/dma). Note: for undefined settings the default configuration is used. 1h = TX FIFO <= 3/4 empty 2h = TX FIFO <= 1/2 empty (default) 3h = TX FIFO <= 1/4 empty 4h = Opposite of full 5h = TX FIFO is empty 6h = TX FIFO <= 1 7h = TX_FIFO >= (MAX_FIFO_LEN -1)

23 TPR Register (Offset = 110h) [Reset = 00000001h]

TPR is shown in Figure 25-74 and described in Table 25-77.

Return to the Summary Table.

This register is programmed to set the timer period for the SCL clock and assign the SCL clock to standard mode.

Figure 25-74. TPR Register

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

Table 25-131. TPR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-7	RESERVED	R/W	0h	
6-0	TPR	R/W	1h	Timer Period This field is used in the equation to configure SCL_PERIOD : SCL_PERIOD = (1 + TPR) x (SCL_LP + SCL_HP) x INT_CLK_PRD where: SCL_PRD is the SCL line period (I2C clock). TPR is the Timer Period register value (range of 1 to 127). SCL_LP is the SCL Low period (fixed at 6). SCL_HP is the SCL High period (fixed at 4). CLK_PRD is the functional clock period in ns. 0h = Smallest value 7Fh = Highest possible value

24 GFCTL Register (Offset = 118h) [Reset = 00000000h]

GFCTL is shown in Figure 25-75 and described in Table 25-78.

Return to the Summary Table.

This register controls the glitch filter on the SCL and SDA lines

Figure 25-75. GFCTL Register

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

Table 25-133. GFCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	RESERVED	R/W	0h	
8	AGFEN	R/W	0h	Analog Glitch Suppression Enable 0h = Analog Glitch Filter disable 1h = Analog Glitch Filter enable
7-3	RESERVED	R/W	0h	
2-0	DGFSEL	R/W	0h	Glitch Suppression Pulse Width This field controls the pulse width select for glitch suppression on the SCL and SDA lines. The following values are the glitch suppression values in terms of functional clocks. (Core Domain only) 0h = Bypass 1h = 1 clock 2h = 2 clocks 3h = 3 clocks 4h = 4 clocks 5h = 8 clocks 6h = 16 clocks 7h = 31 clocks

25 BMON Register (Offset = 11Ch) [Reset = 00000003h]

BMON is shown in Figure 25-76 and described in Table 25-79.

Return to the Summary Table.

This register is used to determine the SCL and SDA signal status.

Figure 25-76. BMON Register

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

Table 25-135. BMON Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R	0h	
1	SDA	R	1h	I2C SDA Status 0h = The I2CSDA signal is low. 1h = The I2CSDA signal is high. Note: During and right after reset, the SDA pin is in GPIO input mode without the internal pull enabled. For proper I2C operation, the user should have the external pull-up resistor in place before starting any I2C operations.
0	SCL	R	1h	I2C SCL Status 0h = The I2CSCL signal is low. 1h = The I2CSCL signal is high. Note: During and right after reset, the SCL pin is in GPIO input mode without the internal pull enabled. For proper I2C operation, the user should have the external pull-up resistor in place before starting any I2C operations.

26 TXDATA Register (Offset = 120h) [Reset = 00000000h]

TXDATA is shown in Figure 25-77 and described in Table 25-80.

Return to the Summary Table.

Transmit Data Register. This register is the transmit data register (the interface to the FIFOs). For transmitted data, if the FIFO is enabled, data written to this location is pushed onto the transmit FIFO. If the FIFO is disabled, data is stored in the transmitter holding register (the bottom word of the transmit FIFO).

Figure 25-77. TXDATA Register

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

Table 25-137. TXDATA Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	W	0h	
7-0	DATA	W	0h	Transmit Data This byte contains the data to be transferred during the next transaction. 0h = Smallest value FFh = Highest possible value

27 RXDATA Register (Offset = 124h) [Reset = 00000000h]

RXDATA is shown in Figure 25-78 and described in Table 25-81.

Return to the Summary Table.

RX FIFO Read Data Byte This field contains the current byte being read in the RX FIFO stack. If the FIFO is disabled, the data byte and status are stored in the receiving holding register (the bottom word of the receive FIFO). The received data can be retrieved by reading this register.

Figure 25-78. RXDATA Register

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

Table 25-139. RXDATA Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	
7-0	DATA	R	0h	Received Data. This field contains the last received data. 0h = Smallest value FFh = Highest possible value

28 PECSR Register (Offset = 128h) [Reset = 00000000h]

PECSR is shown in Figure 25-79 and described in Table 25-82.

Return to the Summary Table.

PEC Status Register

Figure 25-79. PECSR Register

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

Table 25-141. PECSR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	RESERVED	R	0h	
17	PECSTS_ERROR	R	0h	This status bit indicates if a PEC check error occurred in the transaction that occurred before the last Stop. Latched on Stop. 0h = Indicates PEC check error did not occurr in the transaction that occurred before the last Stop 1h = Indicates if a PEC check error occurred in the transaction that occurred before the last Stop
16	PECSTS_CHECK	R	0h	This status bit indicates if the PEC was checked in the transaction that occurred before the last Stop. Latched on Stop. 0h = Indicates PEC was not checked in the transaction that occurred before the last Stop 1h = Indicates if the PEC was checked in the transaction that occurred before the last Stop
15-9	RESERVED	R	0h	
8-0	PECBYTECNT	R	0h	PEC Byte Count This is the current PEC Byte Count of the Controller State Machine. 0h = Minimum Value 1FFh = Maximum Value

29 TA Register (Offset = 14Ch) [Reset = 00000000h]

TA is shown in Figure 25-80 and described in Table 25-83.

Return to the Summary Table.

Target Address Register

Figure 25-80. TA Register

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

Table 25-143. TA Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R/W	0h	
15	MODE	R/W	0h	This bit selects the adressing mode to be used When 0, 7-bit addressing is used. When 1, 10-bit addressing is used. 0h = 7-bit addressing mode 1h = 10-bit addressing mode
14-11	RESERVED	R/W	0h	
10-1	ADDR	R/W	0h	I2C Target Address This field specifies bits A9 through A0 of the Target address. In 7-bit addressing mode as selected by MSA.MODE bit, the top 3 bits are don't care 0h = Smallest value 3FFh = Highest possible value
0	DIR	R/W	0h	Receive/Send The DIR bit specifies if the next operation is a Receive (High) or Transmit (Low). 0h = Transmit 1h = Receive 0h = in transmit mode. 1h = is in receive mode.

30 TIMEOUT_CNT Register (Offset = 150h) [Reset = 00020002h]

TIMEOUT_CNT is shown in Figure 25-81 and described in Table 25-84.

Return to the Summary Table.

This register contains the upper 8 bits of a 12-bit current counter values for counter A and B. The lower four bits of the counter are not user visible and are always 0h.

Figure 25-81. TIMEOUT_CNT Register

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

Table 25-145. TIMEOUT_CNT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	RESERVED	R	0h	
23-16	TCNTB	R	2h	Timeout Count B Current Count: This field contains the upper 8 bits of a 12-bit current counter for timeout counter B 0h = Smallest Value FFh = Highest possible value
15-8	RESERVED	R	0h	
7-0	TCNTA	R	2h	Timeout Count A Current Count: This field contains the upper 8 bits of a 12-bit current counter for timeout counter A 0h = Smallest Value FFh = Highest possible value

31 TIMEOUT_CTL Register (Offset = 154h) [Reset = 00020002h]

TIMEOUT_CTL is shown in Figure 25-82 and described in Table 25-85.

Return to the Summary Table.

This register contains controls for Timeout Counters A and B

Figure 25-82. TIMEOUT_CTL Register

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

Table 25-147. TIMEOUT_CTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31	TCNTBEN	R/W	0h	Timeout Counter B Enable 0h = Disable Timeout Counter B 1h = Enable Timeout Counter B
30-24	RESERVED	R/W	0h	
23-16	TCNTLB	R/W	2h	Timeout Count B Load: Counter B is used for SCL High Detection. This field contains the upper 8 bits of a 12-bit pre-load value for the Timeout B count. NOTE: The value of CNTLB must be greater than 1h. Each count is equal to 1* clock period. For example, with 10MHz functional clock one timeout period will be equal to1*100ns. 0h = Smallest possible value FFh = Highest possible value
15	TCNTAEN	R/W	0h	Timeout Counter A Enable 0h = Disable Timeout Counter B 1h = Enable Timeout Counter B
14-8	RESERVED	R/W	0h	
7-0	TCNTLA	R/W	2h	Timeout counter A load value Counter A is used for SCL low detection. This field contains the upper 8 bits of a 12-bit pre-load value for the Timeout A count. NOTE: The value of CNTLA must be greater than 1h. Each count is equal to 520 times the timeout period of functional clock. For example, with 8MHz functional clock and a 100KHz operating I2C clock, one timeout period will be equal to (1 / 8MHz) * 520 or 65 us. 0h = Smallest Value FFh = Highest possible value

32 PECCTL Register (Offset = 158h) [Reset = 00000000h]

PECCTL is shown in Figure 25-83 and described in Table 25-86.

Return to the Summary Table.

PEC Control Register

Figure 25-83. PECCTL Register

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

Table 25-149. PECCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-13	RESERVED	R/W	0h	
12	PECEN	R/W	0h	PEC Enable This bit enables the SMB Packet Error Checking (PEC). When enabled the PEC is calculated on all bits except the Start, Stop, Ack and Nack. The PEC LSFR and the Byte Counter is set to 0 when the State Machine is in the IDLE state, which occur following a Stop or when a timeout occurs. The Counter is also set to 0 after the PEC byte is sent or received. Note that the NACK is automatically send following a PEC byte that results in a PEC error. The PEC Polynomial is x^8 + x^2 + x^1 + 1. 0h = PEC is disabled 1h = PEC is enabled
11-9	RESERVED	R/W	0h	
8-0	PECCNT	R/W	0h	PEC Count When this field is non zero, the number of I2C bytes are counted (Note that although the PEC is calculated on the I2C address it is not counted at a byte). When the byte count = PECCNT and the state machine is transmitting, the contents of the LSFR is loaded into the shift register instead of the byte received from the Tx FIFO. When the state machine is receiving, after the last bit of this byte is received the LSFR is checked and if it is non-zero, a PEC RX Error interrupt is generated. The I2C packet must be padded to include the PEC byte for both transmit and receive. In transmit mode the FIFO must be loaded with a dummy PEC byte. In receive mode the PEC byte will be passed to the Rx FIFO. In the normal Controller use case, FW would set PECEN=1 and PECCNT=SMB packet length (Not including Target Address byte, but including the PEC byte). FW would then configure DMA to allow the packet to complete unassisted and write MCTR to initiate the transaction. Note that when the byte count = PEC CNT, the byte count is reset to 0 and multiple PEC calculation can automatically occur within a single I2C transaction. Note that any write to the I2PECCTL Register will clear the current PEC Byte Count in the State Machine. 0h = Minimum Value 1FFh = Maximum Value

Chapter 26

Serial Peripheral Interface (SPI)

The functionality of a UNICOMM module when configured to operate as a SPI is described in this section. The protocol mode of a given UNICOMM instance is decided by the programmed SELECT field of the IPMODE register. If a UNICOMM instance is configured for another peripheral mode, the SPI functionality on that UNICOMM instance is disabled and unusable.

ADVANCE INFORMATION

26.1 Overview

The UNICOMM-SPI module provides a standardized serial interface for transferring data between AM13E230x devices and external devices (such as Sensors, Memory, ADCs, DACs, or other MCUs) using the SPI protocol.

26.1.1 Purpose of the Peripheral

The SPI module acts as a controller or peripheral interface for synchronous serial communication with peripheral devices and other controllers. The transmit and receive paths are buffered with internal, independent FIFO memories allowing up to 16 entries with 16-bit width. A DMA interface is also provided to allow the data exchange with the transmit and receive FIFOs.

26.1.2 Features

The UNICOMM-SPI peripheral mode of the UNICOMM module includes the following features:

- Configurable as a controller or a peripheral
- Programmable clock bit rate and prescaler
- Separate transmit (TX) and receive (RX) buffers/first-in first-out buffers (FIFOs)
- Programmable 4 -16 bit data frame size (Controller Mode)
- Programmable 7-16 bit data frame size (Peripheral Mode)
- One chip select (CS) pin
- Direct Memory Access (DMA) support
- Available Interrupts:
  - Transmit (TX) and receive (RX) FIFO levels and empty/full conditions
  - Parity and overrun error
  - Idle condition
  - Receive positive and negative edge conditions
  - DMA done conditions
- Programmable SPI mode support for:
  - Motorola SPI frame format: four phase + polarity combinations
    - Falling edge without phase delay: SPICLK active-high. SPI transmits data on the falling edge of the SPICLK signal and receives data on the rising edge of the SPICLK signal.
    - Falling edge with phase delay: SPICLK active-high. SPI transmits data one half-cycle ahead of the falling edge of the SPICLK signal and receives data on the falling edge of the SPICLK signal.
    - Rising edge without phase delay: SPICLK inactive-low. SPI transmits data on the rising edge of the SPICLK signal and receives data on the falling edge of the SPICLK signal.
    - Rising edge with phase delay: SPICLK inactive-low. SPI transmits data one half-cycle ahead of the rising edge of the SPICLK signal and receives data on the rising edge of the SPICLK signal
  - Texas Instruments frame format
- Single-bit parity is supported in both transmit and receive paths
- Internal loopback test mode

To support SPI protocol, a UNICOMM module must be configured as a SPI in the UNICOMM top level IPMODE register.

26.1.3 Functional Block Diagram

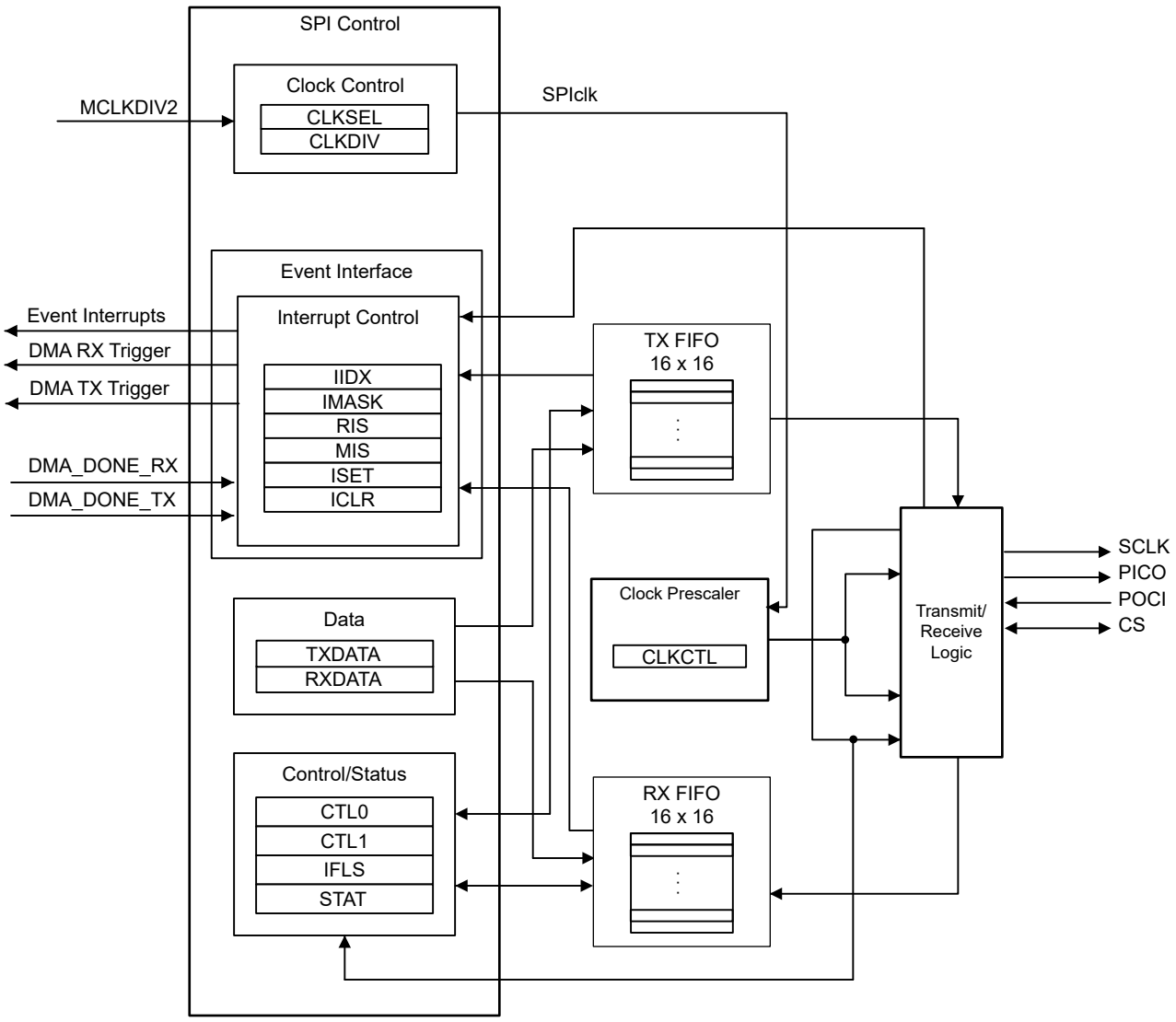


Figure 26-1. SPI Functional Block Diagram

26.2 Peripheral Functional Description

26.2.1 Clock Control

The SPI functional clock SPIclk is selected and divided from the clock sourced to this module.

- Use the CLKSEL register to enable MCLKDIV2 as the functional clock source for the UNICOMM SPI module.
- Use CLKDIV register to select the divide ratio of the SPI functional clock. Options are from divide by 1 to 8.

The SPI module must be enabled before being configured for use by using the ENABLE bit in the UNICOMM_REGS PWREN register. When the SPI is being setup or the configuration is being changed, the ENABLE bit needs to be cleared to avoid unpredictable behavior during the updates or the first data received or transmitted following an update.

The maximum SPI frequency supported with controller and peripheral mode depends on the device clock option and IO option. Please refer to specific device data sheet spec for more information.

26.2.2 General Architecture

In SPI communication, the controller transfers data by toggling the SPICLK signal. For both the peripheral and the controller, data is shifted out of the shift registers on one edge of the SPICLK and latched into the shift register on the opposite SPICLK clock edge. As a result, both devices send and receive data simultaneously. The application determines whether the data is meaningful or dummy data. There are three possible methods for data transmission:

- Controller sends data; peripheral sends dummy data.
- Controller sends data; peripheral sends data.
- Controller sends dummy data; peripheral sends data.

The controller can initiate data transfer at any time because the controller controls the SPICLK signal. The application, however, determines how the controller detects when the peripheral is ready to broadcast data. The UNICOMM-SPI operates in either controller or peripheral mode. The CTL1.CP bit selects the operating mode and the source of the SPICLK signal.

26.2.2.1 Chip Select Control

UNICOMM-SPI can be configured to be controller mode by setting the CTL1.CP bit to 1, or in peripheral mode by clearing the CTL1.CP bit. The chip select signal needs to be provided by the controller in four-wire mode and the chip select polarity can be inverted by configuring the PINCM.CSx.INV register. In UNICOMM-SPI peripheral mode, the clock is provided by the controller and used by the peripheral to capture the data. The peripheral has the option to operate in 3-wire or 4-wire mode by configuring the CTL0.FRF field. 4-wire mode only accepts data transfers if the CS is activated.

The CTL0.CSCLR register field determines the behavior of the internal bit counter of the UNICOMM-SPI in peripheral mode when the controller de-asserts the CS signal. This bit is only relevant when the UNICOMM-SPI is configured as a SPI peripheral (CTL1.CP = 0).

- CTL0.CSCLR = 0: The transmit/receive bit counter state is retained when the CS signal disables the peripheral.
- CTL0.CSCLR = 1: The transmit/receive bit counter is cleared when the CS signal disables the peripheral.

When using the Motorola 4-wire mode as a UNICOMM-SPI peripheral with the CTL0.CSCLR = 1 setting, the SPI controller device must follow the below constraints for proper communication. Following these constraints helps the UNICOMM-SPI peripheral to synchronize to the controller during initialization or in the case of a disturbance or glitch on the clock line.

- The CS disable period must be longer than 2 SPIclk cycles before the CS pin is re-asserted
- The CS lead time (CS active to the first bit clock edge) must be at least 2 SPIclk clock cycles

26.2.2.2 Data Format

The control bit CTL1.MSB defines the direction of the data input and output with the most-significant-bit (MSB) or least-significant-bit (LSB) first. If the parity is enabled, the parity bit is always received last.

The CTL0.DSS control register bits determine the bit length per transfer. This can be configured to be between 4-16 bits for UNICOMM-SPI Controller mode and 7-16 bits for UNICOMM-SPI Peripheral mode.

A transfer is triggered by writing to the TXDATA register. The data write must have at least the number of bits of the transfer. For example, if only a byte is written to TXDATA, but the length of the transfer is > 8, the missing bits are filled with 0s. On the receive path, the received data is moved to the RX FIFO after the number of bits defined by the CTL0.DSS register have been received.

The RXDATA and TXDATA registers must be accessed with at least the bits covering one transfer.

- 4-8 bits : byte access (Peripheral mode: 7-8 Bits)
- 9-16 bits : 16 bit access

The clock polarity register field (CTL0.SPO) controls the clock polarity of SPICLK when data is not being transferred and is only valid in the Motorola SPI frame mode.

- 0h = Peripheral produces a steady state LOW value on the SPICLK pin when data is not being transferred.
- 1h = Peripheral produces a steady state HIGH value on the SPICLK pin when data is not being transferred.

The clock phase register field (CTL0.SPH) selects the clock edge that captures data and allows the module to change state. This setting has the most impact on the first bit transmitted by either permitting or not permitting a clock transition before the first data capture edge. Please refer to Motorola SPI frame mode section to check the diagrams.

- 0h = Data is captured on the first SPICLK edge transition.
- 1h = Data is captured on the second SPICLK edge transition.

The UNICOMM-SPI can be configured to work in peripheral mode with CTL1.CP bit = 0. In peripheral mode, the clock is provided by the controller and available for the peripheral on the CLK pins which needs to be configured for input. The Clock Select and divider control bits are not used. The CS input signal is used to select/enable the data receive path of the peripheral in 4 wire mode.

The SPI can be configured to work as Controller with CTL1.CP bit = 1. In controller mode, the SPICLK signal needs to be generated by selecting the available clock sources with the clock select bits. The SPI controller also needs to control the CS signal depending on the selected protocol.

Parity checking is a feature in UNICOMM-SPI used to improve the robustness of the communication. The parity transmit enable (CTL1.PTEN) and parity receive enable (CTL1.PREN) register bits add a parity bit to the UNICOMM-SPI transmission or reception respectively.When the CTL1.PTEN bit is set, each data transfer includes an additional parity bit calculated by the previously transmitted data bits. If CTL1.PTEN is cleared, there is no additional bit added to the transmission. When the CTL1.PREN bit is set, the last received bit is used as parity to evaluate the integrity of the previous bits that were received. The CTL1.PES bit selects whether the parity mode is even or odd for both transmit and receive parity operations. When a parity error is detected on a received character, the interrupt flag RIS.PER is set to mark the data as invalid.

26.2.2.3 Delayed Data Sampling

In some circumstances, the data inputted to a UNICOMM-SPI controller on the POCI pin has some delay due to runtime conditions. At high SPICLK speeds, the setup and hold requirements of the UNICOMM-SPI become very strict. As a result, when a data frame is delayed on the POCI pin, the UNICOMM-SPI controller can incorrectly sample the previous data value at the SPICLK sampling edge.

To compensate for this, the delayed sampling feature can be enabled with the CLKCTL.DSAMPLE bits. Delayed sampling is only available when the UNICOMM-SPI is configured for controller mode. The delay configuration of CLKCTL.DSAMPLE is adjusted in steps of SPIclk clock cycles. The maximum allowed delay must not exceed the length of one data frame. If CLKCTL.DSAMPLE is set to zero, delayed sampling is not applied.

See the device data sheet for the minimum and maximum SPICLK speeds with and without the delay sampling feature enabled.

26.2.2.4 Clock Generation

The SPI includes a programmable bit rate clock divider and prescaler to generate the serial output clock (SCLK).

The module supports bit rates up to the clock source (MCLKDIV2) divided by 2.

SPIclk is the output after clock division is performed according to ratio selected by the CLKDIV register.

SPIclk = MCLKDIV2 Frequency / (1 + CLKDIV)

SPI Sampling Clock (SCLK) is the output after dividing the SPIclk by the prescalar value. SCLK = SPIclk / ((1 + SCR)*2)

Note

The input clock frequency must be at least 2 times the desired SPIclk frequency when using the minimum CLKDIV value.

26.2.2.5 SPI FIFO Operation

See Section 23.2.2 first for an overview of the FIFO operation and status flags available to every UNICOMM instance. The below details FIFO considerations specific to the SPI module.

Transmit FIFO

The CPU or DMA writes data to the TX FIFO by writing the TXDATA register. Data is stored in the FIFO until read out by the SPI transmitter logic. When configured as a controller or a peripheral, parallel data is written into the TX FIFO before serial conversion and transmission to the attached peripheral or controller, respectively, through the PICO or POCI pin.

In peripheral mode, the SPI transmits data each time the controller initiates a transaction. If the TX FIFO is empty and the controller initiates a transfer, the peripheral transmits the most-recent value written to the transmit FIFO. User software is responsible to make valid data available to the FIFO as needed. The SPI can be configured to generate an interrupt or a DMA request when the FIFO reaches a threshold level empty. There is also an interrupt flag called TXFIFO_UNF that indicates if a TX FIFO underflow condition has occurred in when the UNICOMM-SPI is in peripheral mode.

Receive FIFO

Received data from the serial interface is stored in the RX FIFO until read out by the CPU or DMA, which accesses the RX FIFO by reading the RXDATA register.

When configured as a controller or peripheral, serial data is received through the POCI or PICO pin. If the RX FIFO is full and new data is written into the FIFO without reading data the RXFIFO overflow event is set. The receive FIFO has a RXFULL interrupt to indicate a FIFO full condition.

26.2.2.6 DMA Operation

The UNICOMM-SPI provides an interface to the DMA module, with separate transmit and receive lines. The DMA operation of the SPI is enabled through the SPI DMA Event and DMA module registers. When DMA operation is enabled, the SPI asserts a DMA request on the receive or transmit channel when the associated FIFO can transfer data. The DMA transfer requests are handled automatically by the DMA controller based on the DMA channel configurations (burst size, transfer size, source address, etc).

For more information about the DMA module and DMA-specific configurations, see Direct Memory Access (DMA).

- For the receive channel, a DMA transfer request is asserted when the RX FIFO contains at least the FIFO trigger level configured by the RXIFLSEL bit in the IFLS register.
- For the transmit channel, a DMA transfer request is asserted whenever the TX FIFO contains fewer characters than the FIFO trigger level configured using the TXIFLSEL bit in the IFLS register.

26.2.3 Internal Loopback Operation

The SPI module can be placed into an internal loopback mode for diagnostic or debug work by setting the LBM bit in the CTL1 register. In loopback mode, the data from the TX FIFO can serially transmitted into the RX FIFO. Thedata from the RX FIFO can be read to check whether correct transmission has occurred or not. The external toggling of the IOs has no effect when the module is set in the internal loopback mode.

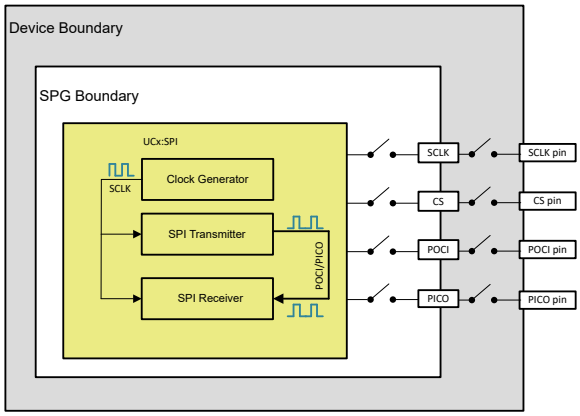


Figure 26-2. SPI Internal Loopback

26.2.4 Protocol Descriptions

The protocol format mode can be selected by using CTL0.FRF register. The supported options include Motorola 3-wire, Motorola 4-wire, and Texas Instruments Synchronous.

26.2.4.1 Motorola SPI Frame Format

The Motorola SPI interface is a 4-wire interface where the CS signal behaves as a peripheral select. In the 3-wire mode, the CS signal is not required and the module behaves as if always selected. The main feature of the Motorola SPI format is that the inactive state and phase of the SCLK signal can be programmed through the SPO and SPH bits in the CTL0 control register.

SPO Clock Polarity Bit

If the CTL0.SPO clock polarity control bit is clear, the bit produces a steady-state low value on the SCLK pin when data is not being transferred. If the CTL0.SPO bit is set, the bit places a steady-state high value on the SCLK pin when data is not being transferred.

SPH Phase-Control Bit

The CTL0.SPH phase-control bit selects the clock edge that captures data, and allows it to change state. The state of this bit has the most impact on the first bit transmitted, by either allowing or not allowing a clock transition before the first data capture edge. If the CTL0.SPH phase-control bit is clear, data is captured on the first clock edge transition. If the SPH bit is set, data is captured on the second clock edge transition.

The serial clock (SCLK) is held inactive while the SPI is IDLE. SCLK transitions at the programmed frequency only during active transmission or reception of data. The IDLE state of SCLK provides a receive timeout indication that occurs when the RX FIFO still contains data after a timeout period.

Motorola SPI Frame Format with SPO = 0 and SPH = 0

Figure 26-3 shows signal sequences for Motorola SPI format with SPO = 0 and SPH = 0.

ADVANCE INFORMATION

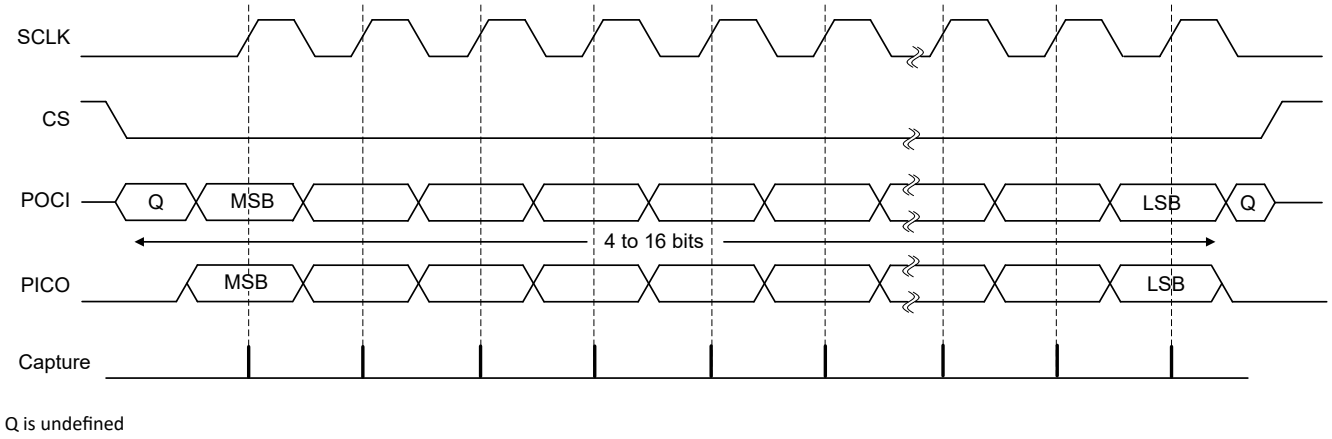


Figure 26-3. Motorola SPI Format With SPO = 0 and SPH = 0

In this configuration, the following occurs during idle periods:

- SCLK is forced low
- CS is forced high
- The transmit data line PICO is forced low
- The UNICOMM-SPI *enables* the SCLK pin when configured as a controller
- The UNICOMM-SPI *disables* the SCLK pin when configured as a peripheral

If the SPI is enabled and the TX FIFO contains valid data, the CS controller signal is driven low at the start of transmission, enabling peripheral data onto the POCI input line of the controller. The controller PICO output pin is enabled.

One-half SCLK period later, valid controller data is transferred to the PICO pin. Once both the controller and peripheral data are set, the SCLK controller clock pin goes high after an additional one-half SCLK period. The data is now captured on the rising edges and propagated on the falling edges of the SCLK signal.

For a single-word transmission, after all bits of the data word are transferred, the CS line is returned to the IDLE high state one SCLK period after the last bit is captured. For continuous back-to-back transmissions, the CS signal must pulse high between each data-word transfer because the peripheral-select pin freezes the data in the serial peripheral register and prevents altering the data if the SPH bit is clear. The controller device must raise the CS pin of the peripheral device between each data transfer to enable the serial peripheral data write. When the continuous transfer completes, the CS pin is returned to the IDLE state one SCLK period after the last bit is captured.

Motorola SPI Frame Format with SPO = 0 and SPH = 1

Figure 26-4 shows the signal sequence for Motorola SPI format with SPO = 0 and SPH = 1.

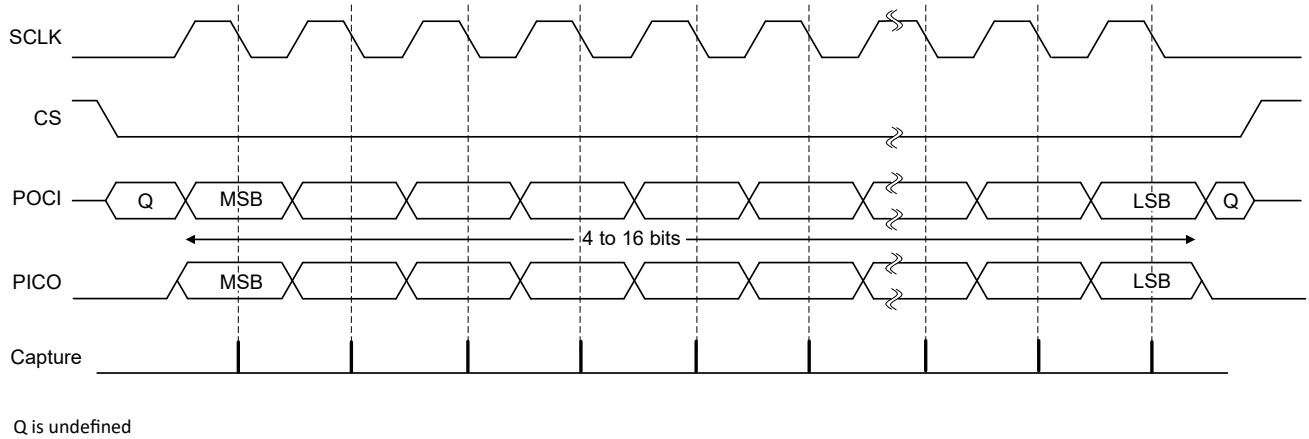


Figure 26-4. Motorola SPI Frame Format With SPO = 0 and SPH = 1

In this configuration, the following occurs during idle periods:

- SCLK is forced low
- CS is forced high
- The transmit data line PICO is forced low
- The UNICOMM-SPI *enables* the SCLK pin when configured as a controller
- The UNICOMM-SPI *disables* the SCLK pin when configured as a peripheral

If the SPI is enabled and valid data is present in the TX FIFO, CS controller signal goes low at the start of transmission.The controller PICO output is enabled. After an additional one-half SCLK period, both controller and peripheral valid data are enabled onto the respective transmission lines. At the same time, SCLK is enabled with a rising-edge transition. Data is then captured on the falling edges and propagated on the rising edges of the SCLK signal.

Fora single-word transfer, after all bits are transferred, the CS line is returned to the IDLE high state one SCLK period after the last bit is captured. For continuous back-to-back transfers, the CS pin is held low between successive data words and terminates like a single-word transfer.

Motorola SPI Frame Format with SPO = 1 and SPH = 0

Figure 26-5 shows signal sequences for Motorola SPI format with SPO = 1 and SPH = 0.

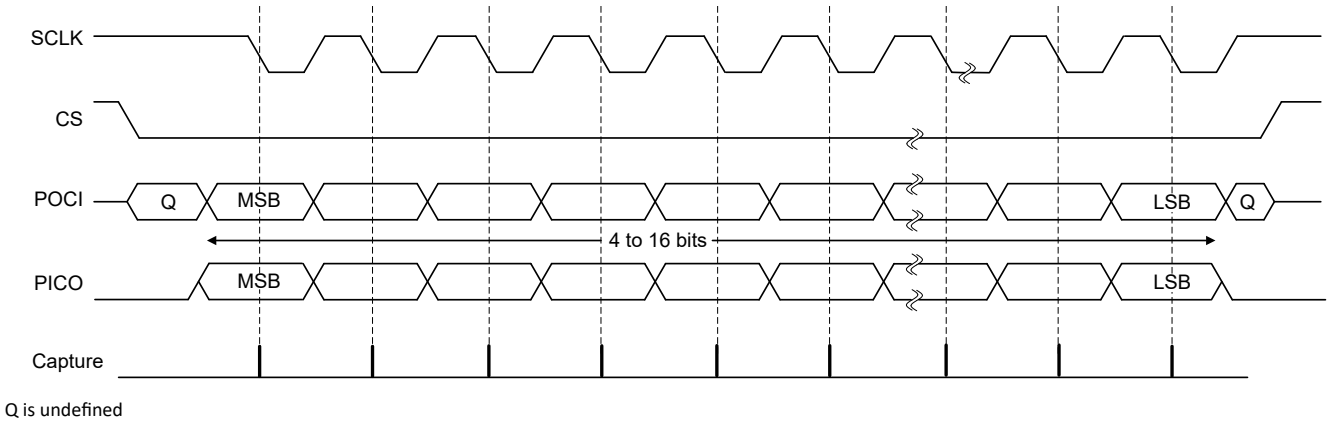


Figure 26-5. Motorola SPI Frame Format With SPO = 1 and SPH = 0

In this configuration, the following occurs during idle periods:

- SCLK is forced high
- CS is forced high
- The transmit data line PICO is arbitrarily forced low

- The UNICOMM-SPI *enables* the SCLK pin when configured as a controller
- The UNICOMM-SPI *disables* the SCLK pin when configured as a peripheral

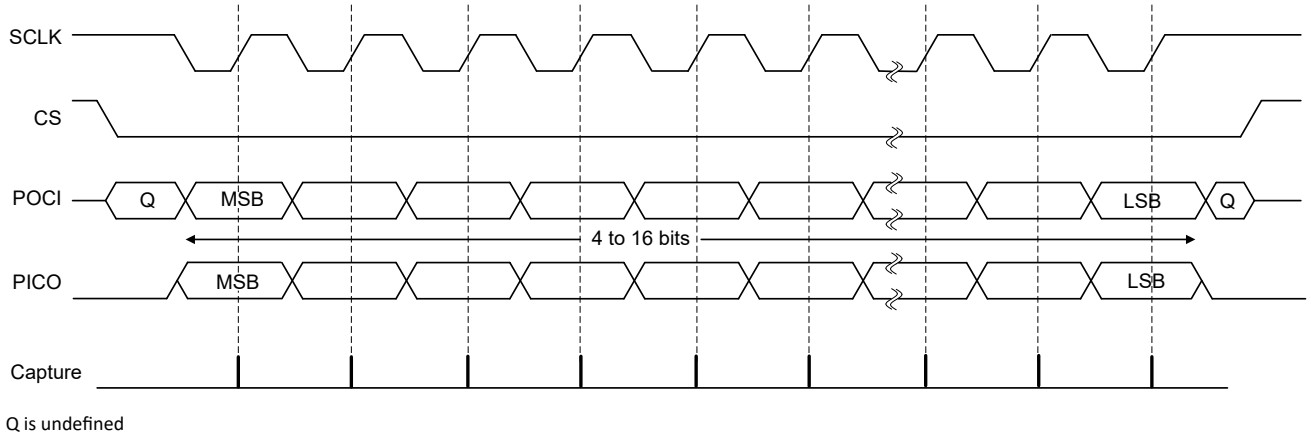
If the SPI is enabled and valid data is in the TX FIFO, the SPI CS controller signal goes low at the start of transmission and transfers the peripheral data to the controller's POCI line immediately. The controller PICO output pin is enabled.

One-half SCLK period later, valid controller data is transferred to the PICO line. When both the controller and peripheral data have been set, the SCLK controller clock pin becomes low after one additional half SCLK period. Data is captured on the falling edges and propagated on the rising edges of the SCLK signal.

For a single-word transmission after all bits of the data word are transferred, the CS line is returned to the IDLE high state one SCLK period after the last bit is captured. For continuous back-to-back transmissions, the CS signal must pulse high between each data-word transfer, as the peripheral-select pin freezes the data in the serial peripheral register and prevents the data from being altered when the SPH bit is clear. The controller device must raise the CS pin of the peripheral device between each data transfer to enable the serial peripheral data write. When the continuous transfer completes, the CS pin returns to the IDLE state one SCLK period after the last bit is captured.

Motorola SPI Frame Format with SPO = 1 and SPH = 1

Figure 26-6 shows the signal sequence for Motorola SPI format with SPO = 1 and SPH = 1.



In this configuration, the following occurs during idle periods:

- SCLK is forced high
- CS is forced high
- The transmit data line PICO is arbitrarily forced low
- The UNICOMM-SPI *enables* the SCLK pin when configured as a controller
- The UNICOMM-SPI *disables* the SCLK pin when configured as a peripheral

If the SPI is enabled and valid data is in the TX FIFO, the start of transmission is signified by the CS controller signal going low. The controller PICO output pin is enabled. After an additional one-half SCLK period, both controller and peripheral data is transmitted onto their respective transmission lines. At the same time, SCLK is enabled with a falling-edge transition. Data is then captured on the rising edges and propagated on the falling edges of the SCLK signal.

For a single word transmission, after all bits are transferred, the CS line returns to its IDLE high state one SCLK period after the last bit is captured. For continuous back-to-back transmissions, the CS pin remains in its active low state until the final bit of the last word is captured and then returns to its IDLE state. For continuous back-to-back transfers, the CS pin is held low between successive data words and terminates like a single-word transfer.

Figure 26-6. Motorola SPI Frame Format With SPO = 1 and SPH = 1

26.2.4.2 Texas Instruments Synchronous Serial Frame Format

The SPI peripheral is compatible with Texas Instruments Synchronous Serial frame format.

Figure 26-7 shows the TI synchronous serial frame format for a single and continuous transmitted frame.

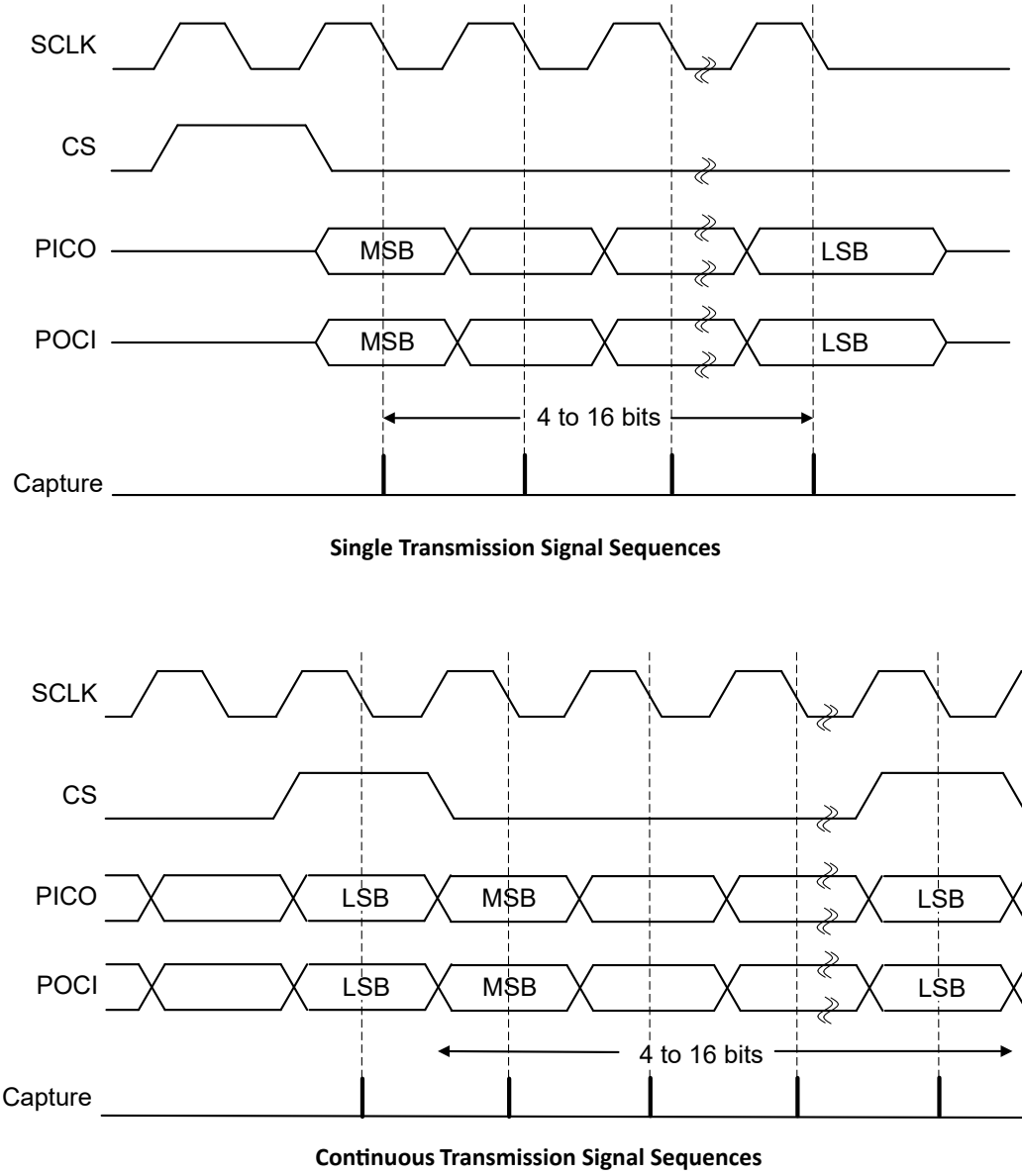


Figure 26-7. TI Synchronous Serial Frame Format

SCLK and CS are forced low and the transmit data line PICO is put in tristate whenever the SPI is idle. When the bottom entry of the TX FIFO contains data, CS is pulsed high for one SCLK period. The transmitted value is also transferred from the TX FIFO to the serial shift register of the transmit logic. On the next rising edge of SCLK, the MSB of the 4- to 16-bit data frame is shifted out on the PICO pin. Likewise, the MSB of the received data is shifted onto the POCI pin by the off-chip serial peripheral device. Both the SPI and the off-chip serial peripheral device then clock each data bit into the respective serial shifters on each falling edge of SCLK. The received data is transferred from the serial shifter to the RX FIFO on the first rising edge of SCLK after the least significant bit (LSB) is latched.

The serial clock (SCLK) is held inactive while the SPI is idle and SCLK transitions at the programmed frequency only during active transmission or reception of data. The IDLE state of SCLK provides a receive time-out indication that occurs when the RX FIFO still contains data after a time-out period.

26.2.5 Status Flags

The table below lists the behavior of the status bits in the STAT register. Status bits that are common across all UNICOMM peripherals and relate to the TX and RX FIFOs are explained in Section 23.2.2.4 from the top-level SPGSS chapter.

Table 26-1. UNICOMM-SPI Status Bits

Status Bit	Description
BUSY	1. BUSY bit is set (“ACTIVE”) when there is an ongoing transmission and reception. If in peripheral mode and CTL1.POD is set, i.e. the peripheral cannot transmit data on the POCI line, the BUSY bit is set during reception. 2. BUSY bit is cleared (“IDLE”) when there is no ongoing transaction (receive or transmit)

26.2.6 Initialization

Follow the high-level UNICOMM configurations in Section 23.3 before executing the below SPI-specific configurations.

Note

Pull-ups can be used to avoid unnecessary toggles on the SPI pins, which can take the peripheral to a wrong state. In addition, if the SCLK signal is programmed to steady state High through the SPO bit in the CTL0 register, then software must also configure the GPIO port pin corresponding to the SCLK signal as a pull-up.

To enable and initialize the SPI, perform the following initialization steps:

1. Clear the ENABLE bit in the CTL1 register before making any of the below configuration changes.
2. Select whether the SPI is a controller or peripheral:
   - For controller mode, set the CP bit in the CTL1 register.
   - For peripheral mode, clear the CP bit in the CTL1 register.
3. Configure the CTL0 and CTL1 registers based on the desired protocol, data width and other SPI-specific configurations.
4. Configure the desired FIFO trigger levels in the IFLS register.
5. Enable desired interrupts and/or DMA event by using CPU_INT, DMA_TRIG_RX, DMA_TRIG_TX group IMASK registers.
6. (Optionally) Configure the emulation mode for the peripheral in the PDBGCTL register.
7. Enable the SPI by setting the ENABLE bit in the CTL1 register.

26.2.7 Interrupt and Events Support

The SPI module contains three event publishers and no event subscribers. One event publisher (CPU_INT) manages SPI interrupt requests (IRQs) to the CPU subsystem through a static event route. The second and third event publishers (DMA_TRIG_RX, DMA_TRIG_TX) are used to setup the trigger signaling for the DMA through DMA event route.

The SPI events are summarized in Table 26-2.

Table 26-2. SPI Events

Event	Type	Source	Destination	Configuration	Functionality
Section 13.1.4.1	Publisher	SPI	CPU Subsystem	CPU_INT registers	Fixed interrupt route from SPI to CPU
Section 13.1.4.2	Publisher	SPI	DMA	DMA_TRIG_RX registers	Fixed interrupt route from SPI RX to DMA
Section 13.1.4.2	Publisher	SPI	DMA	DMA_TRIG_TX registers	Fixed interrupt route from SPI TX to DMA

26.2.7.1 CPU Interrupt Event Publisher (CPU_INT)

The SPI module provides 10 interrupt sources that can source a CPU interrupt event. Table 26-3lists the CPU interrupt events from the SPI in order of decreasing priority.

Table 26-3. SPI CPU_INT Trigger Conditions

IIDX STAT	Name	Description
0x01	PER	Parity error event. This bit if a Parity error has been detected.
0x02	RXFIFO_OVF	RXFIFO overflow event. This interrupt is set when an RX FIFO overflow is detected.
0x03	RX	Receive FIFO event. This interrupt is set when the selected receive FIFO level is reached.
0x04	RXFULL	RX FIFO full event. This interrupt is set when the RX FIFO is full.
0x05	TXFIFO_UNF	TX FIFO underflow event.
0x06	TX	Transmit FIFO event. This interrupt is set when the selected transmit FIFO level is reached.
0x07	TXEMPTY	Transmit FIFO empty interrupt. This is set if all data in the transmit FIFO has been shifted out.
0x08	IDLE	SPI Idle. SPI has finished transfers and entered IDLE mode. This bit is set when STAT.BUSY bit goes low.
0x09	DMA_DONE_RX	This interrupt is set if the RX DMA channel sends the DONE signal.
0x10	DMA_DONE_TX	This interrupt is set if the TX DMA channel sends the DONE signal.

The CPU interrupt event configuration is managed with the CPU_INT event management registers. See *Using Event Registers* for guidance on configuring the Event registers.

26.2.7.2 DMA Trigger Publisher (DMA_TRIG_RX, DMA_TRIG_TX)

The DMA_TRIG_TX and DMA_TRIG_RX group registers are used to setup trigger signaling for the DMA. These registers (IMASK, RIS, MIS, ISET, ICLR) are present for any UCx configured as SPI and can be found in the corresponding UNICOMMSPI_REGS register structures. See Section 9.2.3 for how the DMA trigger Event works and can be configured. Each DMA channel can be triggered by any of the conditions listed in the Section 9.2.3.1 and can send a DMA_DONE signal to the corresponding UCx module.

Table 26-4. SPI DMA_TRIG_TX DMA Trigger Condition

IIDX STAT	Name	Description
0x05	TX	Transmit FIFO event. This interrupt is set if the selected transmit FIFO level has been reached.

Table 26-5. SPI DMA_TRIG_RX and DMA_TRIG_TX Trigger

Trigger Name	Register Structure	Register Group	Description
RX	UNICOMMSPI_REGS	DMA_TRIG_RX	Receive FIFO event. Trigger when SPI-configured UCx module's Receive FIFO contains >= IFLS.RXIFSEL defined bytes

Table 26-5. SPI DMA_TRIG_RX and DMA_TRIG_TX Trigger (continued)

Trigger Name	Register Structure	Register Group	Description
TX	UNICOMMSPI_REGS	DMA_TRIG_TX	Transmit FIFO event. Trigger when SPI-configured UCx module's Transmit FIFO contains <= IFLS.TXIFSEL defined bytes

26.2.8 Emulation Modes

The module behavior while the device is in debug mode is controlled by the FREE and SOFT bits in PDBGCTL register.

When the device is in debug mode and set into halt mode below behavior can be configured.

Table 26-6. Debug Mode Peripheral Behavior

PDBGCTL.FREE	PDBGCTL.SOFT	Function
1	**x**	Modules continues operation
0	**0**	Module stops immediately
0	**1**	Module stops after the next transfer has been finished

26.3 UNICOMM-SPI Registers

This Section describes the UNICOMM-SPI Registers.

26.3.1 UNICOMM-SPI Base Address Table

Table 26-7. UNICOMM-SPI Base Address Table

Bit Field Name		DriverLib Name	Base Address
Instance	Structure		
Uc0spiRegs	UNICOMMSPI_REGS	UC0_SPI_BASE	0x4061_8000
Uc1spiRegs	UNICOMMSPI_REGS	UC1_SPI_BASE	0x4061_9000
Uc3spiRegs	UNICOMMSPI_REGS	UC3_SPI_BASE	0x4065_8000
Uc4spiRegs	UNICOMMSPI_REGS	UC4_SPI_BASE	0x4065_9000

ADVANCE INFORMATION

26.3.2 UNICOMMSPI_REGS Registers

Table 26-8 lists the memory-mapped registers for the UNICOMMSPI_REGS registers. All register offset addresses not listed in Table 26-8 should be considered as reserved locations and the register contents should not be modified.

Table 26-8. UNICOMMSPI_REGS Registers

Offset	Acronym	Register Name	Section
0h	CLKDIV	Clock Divider	Go
8h	CLKSEL	Clock Select for Ultra Low Power peripherals	Go
18h	PDBGCTL	Peripheral Debug Control	Go
20h	IIDX	Interrupt Index Register	Go
28h	IMASK	Interrupt mask	Go
30h	RIS	Raw interrupt status	Go
38h	MIS	Masked interrupt status	Go
40h	ISET	Interrupt set	Go
48h	ICLR	Interrupt clear	Go
58h	IMASK	Interrupt mask	Go
60h	RIS	Raw interrupt status	Go
68h	MIS	Masked interrupt status	Go
70h	ISET	Interrupt set	Go
88h	IMASK	Interrupt mask	Go
90h	RIS	Raw interrupt status	Go
98h	MIS	Masked interrupt status	Go
A0h	ISET	Interrupt set	Go
E4h	INTCTL	Interrupt control register	Go
100h	CTL0	SPI control register 0	Go
108h	STAT	Status Register	Go
10Ch	IFLS	Interrupt FIFO Level Select Register	Go
110h	CLKCTL	Clock prescaler and divider register.	Go
120h	TXDATA	TXDATA Register	Go
124h	RXDATA	RXDATA Register	Go
14Ch	CTL1	SPI control register 1	Go

Complex bit access types are encoded to fit into small table cells. Table 26-9 shows the codes that are used for access types in this section.

Table 26-9. UNICOMMSPI_REGS Access Type Codes

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

1 CLKDIV Register (Offset = 0h) [Reset = 00000000h]

CLKDIV is shown in Figure 26-8 and described in Table 26-10.

Return to the Summary Table.

This register is used to specify module-specific divide ratio of the functional clock

Figure 26-8. CLKDIV Register

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

Table 26-11. CLKDIV Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R/W	0h	
2-0	RATIO	R/W	0h	Selects divide ratio of module clock 0h = Do not divide clock source 1h = Divide clock source by 2 2h = Divide clock source by 3 3h = Divide clock source by 4 4h = Divide clock source by 5 5h = Divide clock source by 6 6h = Divide clock source by 7 7h = Divide clock source by 8

2 CLKSEL Register (Offset = 8h) [Reset = 00000000h]

CLKSEL is shown in Figure 26-9 and described in Table 26-11.

Return to the Summary Table.

Clock source selection for peripherals

Figure 26-9. CLKSEL Register

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

Table 26-13. CLKSEL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R/W	0h	
3	BUSCLK_SEL	R/W	0h	Selects buscclk as clock source if enabled 0h = Does not select this clock as a source 1h = Select this clock as a source
2-0	RESERVED	R/W	0h	

3 PDBGCTL Register (Offset = 18h) [Reset = 00000000h]

PDBGCTL is shown in Figure 26-10 and described in Table 26-12.

Return to the Summary Table.

This register can be used by the software developer to control the behavior of the peripheral relative to the 'Core Halted' input

Figure 26-10. PDBGCTL Register

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

Table 26-15. PDBGCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R/W	0h	
1	SOFT	R/W	1h	Soft halt boundary control. This function is only available, if [FREE] is set to 'STOP' 0h = Not supported 1h = The peripheral blocks the debug freeze until it has reached a boundary where it can resume without corruption
0	FREE	R/W	1h	Free run control 0h = The peripheral freezes functionality while the Core Halted input is asserted and resumes when it is deasserted. 1h = The peripheral ignores the state of the Core Halted input

ADVANCE INFORMATION

4 IIDX Register (Offset = 20h) [Reset = 00000000h]

IIDX is shown in Figure 26-11 and described in Table 26-13.

Return to the Summary Table.

This register provides the highest priority enabled interrupt index. Value 0x00 means no event pending. Interrupt 1 is the highest priority, IIDX next highest, 4, 8, ... IIDX^31 is the least priority. That is, the least bit position that is set to 1 denotes the highest priority pending interrupt. The priority order is fixed. However, users can implement their own prioritization schemes using other registers that expose the full set of interrupts that have occurred. On each read, only one interrupt is indicated. On a read, the current interrupt (highest priority) is automatically cleared by the hardware and the corresponding interrupt flag in [RIS] and [MIS] are cleared as well. After a read from the CPU (not from the debug interface), the register is updated with the next highest priority interrupt, if none are pending, then it should display 0x0.

Figure 26-11. IIDX Register

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

Table 26-17. IIDX Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	
7-0	STAT	R	0h	Interrupt index status 00h = No interrupt pending 1h = Transmit Parity Event/interrupt pending 2h = RX FIFO Overflow Event/interrupt pending 3h = Receive interrupt 4h = RX FIFO Full Interrupt 5h = TX FIFO underflow interrupt 6h = Transmit Event 7h = Transmit Buffer Empty Event/interrupt pending 9h = End of Transmit Event/interrupt pending Ah = SPI receive time-out interrupt 10h = DMA DONE on RX 11h = DMA DONE on TX 13h = DMA PRE IRQ RX INTERRUPT 14h = DMA PRE IRQ TX INTERRUPT 15h = SPI line time-out interrupt

5 IMASK Register (Offset = 28h) [Reset = 00000000h]

IMASK is shown in Figure 26-12 and described in Table 26-14.

Return to the Summary Table.

Interrupt Mask. If a bit is set, then corresponding interrupt is un-masked. Un-masking the interrupt causes the raw interrupt to be visible in IIDX, as well as MIS.

Figure 26-12. IMASK Register

31	30	29	28	27	26	25	24
RESERVED							
R/W-0h							
23	22	21	20	19	18	17	16
RESERVED							DMA_DONE_TX
R/W-0h							R/W-0h
15	14	13	12	11	10	9	8
DMA_DONE_RX	RESERVED						IDLE
R/W-0h	R/W-0h						R/W-0h
7	6	5	4	3	2	1	0
RESERVED	TXEMPTY	TX	TXFIFO_UNF	RXFULL	RX	RXFIFO_OVF	PER
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 26-19. IMASK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-17	RESERVED	R/W	0h	
16	DMA_DONE_TX	R/W	0h	Enable DMA Done on TX Event Channel Interrupt 0h = Interrupt disabled 1h = Set Interrupt Mask
15	DMA_DONE_RX	R/W	0h	Enable DMA Done on RX Event Channel Interrupt 0h = Interrupt disabled 1h = Set Interrupt Mask
14-9	RESERVED	R/W	0h	
8	IDLE	R/W	0h	SPI Idle event mask. 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
7	RESERVED	R/W	0h	
6	TXEMPTY	R/W	0h	Transmit FIFO Empty event mask. 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
5	TX	R/W	0h	Mask Transmit Event 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
4	TXFIFO_UNF	R/W	0h	TX FIFO underflow interrupt mask 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
3	RXFULL	R/W	0h	RX FIFO Full Interrupt Mask 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
2	RX	R/W	0h	Mask Receive Event 0h = Clear Interrupt Mask 1h = Set Interrupt Mask

Table 26-19. IMASK Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
1	RXFIFO_OVF	R/W	0h	RXFIFO overflow event mask. 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
0	PER	R/W	0h	Parity error event mask. 0h = Clear Interrupt Mask 1h = Set Interrupt Mask

6 RIS Register (Offset = 30h) [Reset = 00000000h]

RIS is shown in Figure 26-13 and described in Table 26-15.

Return to the Summary Table.

Raw interrupt status. Reflects all pending interrupts, regardless of masking. The RIS register allows the user to implement a poll scheme. A flag set in this register can be cleared by writing 1 to the ICLR register bit even if the corresponding IMASK bit is not enabled.

Figure 26-13. RIS Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							DMA_DONE_TX
R-0h							R-0h
15	14	13	12	11	10	9	8
DMA_DONE_RX	RESERVED						IDLE
R-0h	R-0h						R-0h
7	6	5	4	3	2	1	0
RESERVED	TXEMPTY	TX	TXFIFO_UNF	RXFULL	RX	RXFIFO_OVF	PER
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

Table 26-21. RIS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-17	RESERVED	R	0h	
16	DMA_DONE_TX	R	0h	DMA Done on TX Event Channel Interrupt 0h = Interrupt disabled 1h = Interrupt occured
15	DMA_DONE_RX	R	0h	DMA Done on RX Event Channel Interrupt 0h = Interrupt disabled 1h = Interrupt occured
14-9	RESERVED	R	0h	
8	IDLE	R	0h	SPI has done finished transfers and changed into IDLE mode. This bit is set when BUSY goes low. 0h = Interrupt did not occur 1h = Interrupt occurred
7	RESERVED	R	0h	
6	TXEMPTY	R	0h	Transmit FIFO Empty interrupt mask. This interrupt is set if all data in the Transmit FIFO have been move to the shift register. 0h = Interrupt did not occur 1h = Interrupt occurred
5	TX	R	0h	Transmit event When FIFO is present, this bit is set when Tx FIFO threshold is reached When FIFO is not present, this bit is set when tx buffer is empty 0h = Interrupt did not occur 1h = Interrupt occurred
4	TXFIFO_UNF	R	0h	TX FIFO Underflow Interrupt 0h = Interrupt did not occur 1h = Interrupt occurred
3	RXFULL	R	0h	RX FIFO Full Interrupt 0h = Interrupt did not occur 1h = Interrupt occurred

Table 26-21. RIS Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
2	RX	R	0h	Receive event When FIFO is present, this bit is set when FIFO threshold is reached When FIFO is not present, this bit is set when a frame is received 0h = Interrupt did not occur 1h = Interrupt occurred
1	RXFIFO_OVF	R	0h	RXFIFO overflow event. This interrupt is set if an RX FIFO overflow has been detected. 0h = Interrupt did not occur 1h = Interrupt occurred
0	PER	R	0h	Parity error event: this bit is set if a Parity error has been detected 0h = Interrupt did not occur 1h = Interrupt occurred

7 MIS Register (Offset = 38h) [Reset = 00000000h]

MIS is shown in Figure 26-14 and described in Table 26-16.

Return to the Summary Table.

Masked interrupt status. This is an AND of the IMASK and RIS registers.

Figure 26-14. MIS Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							DMA_DONE_TX
R-0h							R-0h
15	14	13	12	11	10	9	8
DMA_DONE_RX	RESERVED						IDLE
R-0h	R-0h						R-0h
7	6	5	4	3	2	1	0
RESERVED	TXEMPTY	TX	TXFIFO_UNF	RXFULL	RX	RXFIFO_OVF	PER
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

Table 26-23. MIS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-17	RESERVED	R	0h	
16	DMA_DONE_TX	R	0h	Masked DMA Done on TX Event Channel Interrupt 0h = Interrupt did not occur 1h = Interrupt occured
15	DMA_DONE_RX	R	0h	Masked DMA Done on RX Event Channel Interrupt 0h = Interrupt did not occur 1h = Interrupt occured
14-9	RESERVED	R	0h	
8	IDLE	R	0h	Masked SPI IDLE mode event. 0h = Interrupt did not occur 1h = Interrupt occurred
7	RESERVED	R	0h	
6	TXEMPTY	R	0h	Masked Transmit FIFO Empty event. 0h = Interrupt did not occur 1h = Interrupt occurred
5	TX	R	0h	Masked Transmit FIFO event. This interrupt is set if the selected Transmit FIFO level has been reached. 0h = Interrupt did not occur 1h = Interrupt occurred
4	TXFIFO_UNF	R	0h	TX FIFO underflow interrupt 0h = Interrupt did not occur 1h = Interrupt occurred
3	RXFULL	R	0h	RX FIFO Full Interrupt 0h = Interrupt did not occur 1h = Interrupt occurred
2	RX	R	0h	Masked Receive Event 0h = Interrupt did not occur 1h = Interrupt occurred

Table 26-23. MIS Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
1	RXFIFO_OVF	R	0h	Masked RXFIFO overflow event. This interrupt is set if an RX FIFO overflow has been detected. 0h = Interrupt did not occur 1h = Interrupt occurred
0	PER	R	0h	Masked Parity error event: this bit if a Parity error has been detected 0h = Interrupt did not occur 1h = Interrupt occurred

8 ISET Register (Offset = 40h) [Reset = 00000000h]

ISET is shown in Figure 26-15 and described in Table 26-17.

Return to the Summary Table.

Interrupt set. Allows interrupts to be set by software (useful in diagnostics and safety checks). Writing a 1 to a bit in ISET will set the event and therefore the related RIS bit also gets set. If the interrupt is enabled through the mask, then the corresponding MIS bit is also set.

Figure 26-15. ISET Register

31	30	29	28	27	26	25	24
RESERVED							
W-0h							
23	22	21	20	19	18	17	16
RESERVED							DMA_DONE_TX
W-0h							W-0h
15	14	13	12	11	10	9	8
DMA_DONE_RX	RESERVED						IDLE
W-0h	W-0h						W-0h
7	6	5	4	3	2	1	0
RESERVED	TXEMPTY	TX	TXFIFO_UNF	RXFULL	RX	RXFIFO_OVF	PER
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h

Table 26-25. ISET Register Field Descriptions

Bit	Field	Type	Reset	Description
31-17	RESERVED	W	0h	
16	DMA_DONE_TX	W	0h	Set DMA Done on TX Event Channel Interrupt 0h = Interrupt disabled 1h = Set Interrupt
15	DMA_DONE_RX	W	0h	Set DMA Done on RX Event Channel Interrupt 0h = Interrupt disabled 1h = Set Interrupt
14-9	RESERVED	W	0h	
8	IDLE	W	0h	Set SPI IDLE mode event. 0h = Writing 0 has no effect 1h = Set Interrupt
7	RESERVED	W	0h	
6	TXEMPTY	W	0h	Set Transmit FIFO Empty event. 0h = Writing 0 has no effect 1h = Set Interrupt
5	TX	W	0h	Set Transmit event. 0h = Writing 0 has no effect 1h = Set Interrupt
4	TXFIFO_UNF	W	0h	Set TX FIFO Underflow Event 0h = Writing has no effect 1h = Set interrupt
3	RXFULL	W	0h	Set RX FIFO Full Event 0h = Writing has no effect 1h = Set Interrupt
2	RX	W	0h	Set Receive event. 0h = Writing 0 has no effect 1h = Set Interrupt

Table 26-25. ISET Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
1	RXFIFO_OVF	W	0h	Set RXFIFO overflow event. 0h = Writing 0 has no effect 1h = Set Interrupt
0	PER	W	0h	Set Parity error event. 0h = Writing 0 has no effect 1h = Set Interrupt

9 ICLR Register (Offset = 48h) [Reset = 00000000h]

ICLR is shown in Figure 26-16 and described in Table 26-18.

Return to the Summary Table.

Interrupt clear. Write a 1 to clear corresponding Interrupt.

Figure 26-16. ICLR Register

31	30	29	28	27	26	25	24
RESERVED							
W-0h							
23	22	21	20	19	18	17	16
RESERVED							DMA_DONE_TX
W-0h							W-0h
15	14	13	12	11	10	9	8
DMA_DONE_RX	RESERVED						IDLE
W-0h	W-0h						W-0h
7	6	5	4	3	2	1	0
RESERVED	TXEMPTY	TX	TXFIFO_UNF	RXFULL	RX	RXFIFO_OVF	PER
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h

Table 26-27. ICLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-17	RESERVED	W	0h	
16	DMA_DONE_TX	W	0h	Clear DMA Done on TX Event Channel Interrupt 0h = Interrupt disabled 1h = Clear Interrupt
15	DMA_DONE_RX	W	0h	Clear DMA Done on RX Event Channel Interrupt 0h = Interrupt disabled 1h = Clear Interrupt
14-9	RESERVED	W	0h	
8	IDLE	W	0h	Clear SPI IDLE mode event. 0h = Writing 0 has no effect 1h = Clear Interrupt
7	RESERVED	W	0h	
6	TXEMPTY	W	0h	Clear Transmit FIFO Empty event. 0h = Writing 0 has no effect 1h = Clear Interrupt
5	TX	W	0h	Clear Transmit event. 0h = Writing 0 has no effect 1h = Clear Interrupt
4	TXFIFO_UNF	W	0h	Clear TXFIFO underflow event 0h = Writing has no effect 1h = Clear interrupt
3	RXFULL	W	0h	Clear RX FIFO underflow event 0h = Writing has no effect 1h = Clear interrupt
2	RX	W	0h	Clear Receive event. 0h = Writing 0 has no effect 1h = Clear Interrupt

Table 26-27. ICLR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
1	RXFIFO_OVF	W	0h	Clear RXFIFO overflow event. 0h = Writing 0 has no effect 1h = Clear Interrupt
0	PER	W	0h	Clear Parity error event. 0h = Writing 0 has no effect 1h = Clear Interrupt

10 IMASK Register (Offset = 58h) [Reset = 00000000h]

IMASK is shown in Figure 26-17 and described in Table 26-19.

Return to the Summary Table.

Interrupt Mask. If a bit is set, then corresponding interrupt is un-masked. Un-masking the interrupt causes the raw interrupt to be visible in IIDX, as well as MIS.

Figure 26-17. IMASK Register

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

Table 26-29. IMASK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R/W	0h	
2	RX	R/W	0h	Receive FIFO event mask. 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
1-0	RESERVED	R/W	0h	

11 RIS Register (Offset = 60h) [Reset = 00000000h]

RIS is shown in Figure 26-18 and described in Table 26-20.

Return to the Summary Table.

Raw interrupt status. Reflects all pending interrupts, regardless of masking. The RIS register allows the user to implement a poll scheme. A flag set in this register can be cleared by writing 1 to the ICLR register bit even if the corresponding IMASK bit is not enabled.

Figure 26-18. RIS Register

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

Table 26-31. RIS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R	0h	
2	RX	R	0h	Receive FIFO event.This interrupt is set if the selected Receive FIFO level has been reached 0h = Interrupt did not occur 1h = Interrupt occurred
1-0	RESERVED	R	0h	

12 MIS Register (Offset = 68h) [Reset = 00000000h]

MIS is shown in Figure 26-19 and described in Table 26-21.

Return to the Summary Table.

Masked interrupt status. This is an AND of the IMASK and RIS registers.

Figure 26-19. MIS Register

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

Table 26-33. MIS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R	0h	
2	RX	R	0h	Receive FIFO event mask. 0h = Interrupt did not occur 1h = Interrupt occurred
1-0	RESERVED	R	0h	

13 ISET Register (Offset = 70h) [Reset = 00000000h]

ISET is shown in Figure 26-20 and described in Table 26-22.

Return to the Summary Table.

Interrupt set. Allows interrupts to be set by software (useful in diagnostics and safety checks). Writing a 1 to a bit in ISET will set the event and therefore the related RIS bit also gets set. If the interrupt is enabled through the mask, then the corresponding MIS bit is also set.

Figure 26-20. ISET Register

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

Table 26-35. ISET Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	W	0h	
2	RX	W	0h	Set Receive FIFO event. 0h = Writing 0 has no effect 1h = Set Interrupt
1-0	RESERVED	W	0h	

ADVANCE INFORMATION

14 IMASK Register (Offset = 88h) [Reset = 00000000h]

IMASK is shown in Figure 26-21 and described in Table 26-23.

Return to the Summary Table.

Interrupt Mask. If a bit is set, then corresponding interrupt is un-masked. Un-masking the interrupt causes the raw interrupt to be visible in IIDX, as well as MIS.

Figure 26-21. IMASK Register

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

Table 26-37. IMASK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-6	RESERVED	R/W	0h	
5	TX	R/W	0h	Transmit FIFO event mask. 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
4-0	RESERVED	R/W	0h	

15 RIS Register (Offset = 90h) [Reset = 00000000h]

RIS is shown in Figure 26-22 and described in Table 26-24.

Return to the Summary Table.

Raw interrupt status. Reflects all pending interrupts, regardless of masking. The RIS register allows the user to implement a poll scheme. A flag set in this register can be cleared by writing 1 to the ICLR register bit even if the corresponding IMASK bit is not enabled.

Figure 26-22. RIS Register

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

Table 26-39. RIS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-6	RESERVED	R	0h	
5	TX	R	0h	Transmit FIFO event: A read returns the current mask for transmit FIFO interrupt. On a write of 1, the mask for transmit FIFO interrupt is set which means the interrupt state will be reflected in MIS.TXMIS. A write of 0 clears the mask which means MIS.TXMIS will not reflect the interrupt. 0h = Interrupt did not occur 1h = Interrupt occurred
4-0	RESERVED	R	0h	

16 MIS Register (Offset = 98h) [Reset = 00000000h]

MIS is shown in Figure 26-23 and described in Table 26-25.

Return to the Summary Table.

Masked interrupt status. This is an AND of the IMASK and RIS registers.

Figure 26-23. MIS Register

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

Table 26-41. MIS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-6	RESERVED	R	0h	
5	TX	R	0h	Masked Transmit FIFO event 0h = Interrupt did not occur 1h = Interrupt occurred
4-0	RESERVED	R	0h	

ADVANCE INFORMATION

17 ISET Register (Offset = A0h) [Reset = 00000000h]

ISET is shown in Figure 26-24 and described in Table 26-26.

Return to the Summary Table.

Interrupt set. Allows interrupts to be set by software (useful in diagnostics and safety checks). Writing a 1 to a bit in ISET will set the event and therefore the related RIS bit also gets set. If the interrupt is enabled through the mask, then the corresponding MIS bit is also set.

Figure 26-24. ISET Register

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

Table 26-43. ISET Register Field Descriptions

Bit	Field	Type	Reset	Description
31-6	RESERVED	W	0h	
5	TX	W	0h	Set Transmit FIFO event. 0h = Writing 0 has no effect 1h = Set Interrupt
4-0	RESERVED	W	0h	

ADVANCE INFORMATION

18 INTCTL Register (Offset = E4h) [Reset = 00000000h]

INTCTL is shown in Figure 26-25 and described in Table 26-27.

Return to the Summary Table.

Interrupt control register

Figure 26-25. INTCTL Register

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

Table 26-45. INTCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	W	0h	
0	INTEVAL	W	0h	Writing a 1 to this field re-evaluates the interrupt sources. 0h = The interrupt or event line is disabled. 1h = The interrupt or event line is in software mode. Software must clear the RIS.

ADVANCE INFORMATION

19 CTL0 Register (Offset = 100h) [Reset = 00000000h]

CTL0 is shown in Figure 26-26 and described in Table 26-28.

Return to the Summary Table.

SPI control register 0

Figure 26-26. CTL0 Register

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

Table 26-47. CTL0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-15	RESERVED	R/W	0h	
14	CSCLR	R/W	0h	Clear shift register counter on CS inactive This bit is relevant only in the peripheral, CTL1.CP=0. 0h = Disable automatic clear of shift register when CS goes to disable. 1h = Enable automatic clear of shift register when CS goes to disable.
13-10	RESERVED	R/W	0h	
9	SPH	R/W	0h	CLKOUT phase (Motorola SPI frame format only) This bit selects the clock edge that captures data and enables it to change state. It has the most impact on the first bit transmitted by either permitting or not permitting a clock transition before the first data capture edge. 0h = Data is captured on the first clock edge transition. 1h = Data is captured on the second clock edge transition.
8	SPO	R/W	0h	CLKOUT polarity (Motorola SPI frame format only) 0h = SPI produces a steady state LOW value on the CLKOUT 1h = SPI produces a steady state HIGH value on the CLKOUT
7	RESERVED	R/W	0h	
6-5	FRF	R/W	0h	Frame format Select 0h = Motorola SPI frame format (3 wire mode) 1h = Motorola SPI frame format (4 wire mode) 2h = TI synchronous serial frame format 3h = Reserved - not to be selected

Table 26-47. CTL0 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
4-0	DSS	R/W	0h	Data Size Select. Values 0 - 2 are reserved and shall not be used. 3h = 4_BIT : 4-bit data SPI allows only values up to 16 Bit 3h (R/W) = Data Size Select bits: 4 4h (R/W) = Data Size Select bits: 5 5h (R/W) = Data Size Select bits: 6 6h (R/W) = Data Size Select bits: 7 7h (R/W) = Data Size Select bits: 8 8h (R/W) = Data Size Select bits: 9 9h (R/W) = Data Size Select bits: 10 Ah (R/W) = Data Size Select bits: 11 Bh (R/W) = Data Size Select bits: 12 Ch (R/W) = Data Size Select bits: 13 Dh (R/W) = Data Size Select bits: 14 Eh (R/W) = Data Size Select bits: 15 Fh (R/W) = Data Size Select bits: 16

20 STAT Register (Offset = 108h) [Reset = 00000000h]

STAT is shown in Figure 26-27 and described in Table 26-29.

Return to the Summary Table.

Status Register

Figure 26-27. STAT Register

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

Table 26-49. STAT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	RESERVED	R	0h	
8	BUSY	R	0h	Busy 0h = SPI is in idle mode. 1h = SPI is currently transmitting and/or receiving data
7	TXCLR	R	0h	TX FIFO Clear Status 0h = FIFO is not cleared 1h = FIFO clear is complete
6	TXFF	R	0h	Transmit FIFO Full The meaning of this bit depends on the state of the FEN bit in the CTL0 register. 0h = The transmitter is not full. 1h = If the FIFO is disabled (FEN is 0), the transmit holding register is full. If the FIFO is enabled (FEN is 1), the transmit FIFO is full.
5	TXFE	R	1h	Transmit FIFO empty. 0h = Transmit FIFO is not empty. 1h = Transmit FIFO is empty.
4	RXCLR	R	0h	RX FIFO Clear Status 0h = FIFO is not cleared 1h = FIFO clear is complete
3	RXFF	R	0h	Receive FIFO Full The meaning of this bit depends on the state of the FEN bit in the CTL0 register. 0h = The receiver can receive data. 1h = If the FIFO is disabled (FEN is 0), the receive holding register is full. If the FIFO is enabled (FEN is 1), the receive FIFO is full.
2	RXFE	R	1h	Receive FIFO empty. 0h = Receive FIFO is not empty. 1h = Receive FIFO is empty.
1-0	RESERVED	R	0h	

21 IFLS Register (Offset = 10Ch) [Reset = 00000022h]

IFLS is shown in Figure 26-28 and described in Table 26-30.

Return to the Summary Table.

The IFLS register is the interrupt FIFO level select register. You can use this register to define the levels at which the TX, RX and timeout interrupt flags are triggered. The interrupts are generated based on a transition through a level rather than being based on the level. That is, the interrupts are generated when the fill level progresses through the trigger level. For example, if the receive trigger level is set to the half-way mark, the interrupt is triggered when the receive FIFO is filled with two or more characters. Out of reset, the TXIFLSEL and RXIFLSEL bits are configured so that the FIFOs trigger an interrupt at the half-way mark.

Figure 26-28. IFLS Register

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

Table 26-51. IFLS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R/W	0h	
7	RXCLR	R/W	0h	RX FIFO CLEAR. Setting this bit will clear the RX FIFO contents. 0h = Disable FIFO clear 1h = Enable FIFO Clear
6-4	RXIFLSEL	R/W	2h	FIFO Level Select for generating events (interrupt/dma). Note: for undefined settings the default configuration is used. 1h = RX FIFO >= 1/4 full 2h = RX FIFO >= 1/2 full (default) 3h = RX FIFO >= 3/4 full 4h = Opposite of empty 5h = RX FIFO is full 6h = RX_FIFO >= (MAX_FIFO_LEN -1) 7h = RX_FIFO <= 1
3	TXCLR	R/W	0h	TX FIFO CLEAR. Setting this bit will clear the TX FIFO contents. 0h = Disable FIFO clear 1h = Enable FIFO Clear
2-0	TXIFLSEL	R/W	2h	FIFO Level Select for generating events (interrupt/dma). Note: for undefined settings the default configuration is used. 1h = TX FIFO <= 3/4 empty 2h = TX FIFO <= 1/2 empty (default) 3h = TX FIFO <= 1/4 empty 4h = Opposite of full 5h = TX FIFO is empty 6h = TX FIFO <= 1 7h = TX_FIFO >= (MAX_FIFO_LEN -1)

22 CLKCTL Register (Offset = 110h) [Reset = 00000000h]

CLKCTL is shown in Figure 26-29 and described in Table 26-31.

Return to the Summary Table.

Clock prescaler and divider register. This register contains the settings for the Clock prescaler and divider settings.

Figure 26-29. CLKCTL Register

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

Table 26-53. CLKCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	DSAMPLE	R/W	0h	Delayed sampling value. In controller mode the data on the input pin will be delayed sampled by the defined clock cycles of internal functional clock hence relaxing the setup time of input data. This setting is useful in systems where the board delays and external peripheral delays are more than the input setup time of the controller. Please refer to the datasheet for values of controller input setup time and assess what DSAMPLE value meets the requirement of the system. Note: High values of DSAMPLE can cause HOLD time violations and must be factored in the calculations. 0h = Smallest value Fh = Highest possible value
27-10	RESERVED	R/W	0h	
9-0	SCR	R/W	0h	Serial clock divider: This is used to generate the transmit and receive bit rate of the SPI. The SPI bit rate is (SPI's functional clock frequency)/((SCR+1)*2). SCR is a value from 0-1023. 0h = Smallest value 3FFh = Highest possible value

23 TXDATA Register (Offset = 120h) [Reset = 00000000h]

TXDATA is shown in Figure 26-30 and described in Table 26-32.

Return to the Summary Table.

TXDATA Register Writing puts the data into the TX FIFO. Reading this register returns the last written value. When PACKEN=0, only the lower 16-bits of data written into the register is transferred to one 16-bits wide TX FIFO entry When PACKEN=1, upper and lower 16-bits of 32-bit write data are transferred to two16-bits wide TX FIFO entry

Figure 26-30. TXDATA Register

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

Table 26-55. TXDATA Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	W	0h	
15-0	DATA	W	0h	Transmit Data When read, last written value will be returned. If the last write to this field was a 32-bit write (with PACKEN=1), 32-bits will be returned and if the last write was a 16-bit write (PACKEN=0), those 16-bits will be returned. When written, one or two FIFO entries will be written depending on PACKEN value. Data values are removed from the transmit FIFO one value at a time by the transmit logic. It is loaded into the transmit serial shifter, then serially shifted out onto the TXD output pin at the programmed bit rate. When a data size of less than 16 bits is selected, the user must right-justify data written to the transmit FIFO. The transmit logic ignores the unused bits. 0h = Smallest value FFFFh = Highest possible value

24 RXDATA Register (Offset = 124h) [Reset = 00000000h]

RXDATA is shown in Figure 26-31 and described in Table 26-33.

Return to the Summary Table.

RXDATA Register Reading this register returns value(s) of FIFO. If the FIFO is empty the last read value is returned. Writing has not effect and is ignored. When PACKEN=1,two entries of the FIFO are returned as a 32-bit value. When PACKEN=0, 1 entry of FIFO is returned as 16-bit value.

Figure 26-31. RXDATA Register

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

Table 26-57. RXDATA Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	
15-0	DATA	R	0h	Received Data When PACKEN=1,two entries of the FIFO are returned as a 32-bit value. When PACKEN=0, 1 entry of FIFO is returned as 16-bit value. As data values are removed by the receive logic from the incoming data frame, they are placed into the entry in the receive FIFO, pointed to by the current FIFO write pointer. Received data less than 16 bits is automatically right justified in the receive buffer. 0h = Smallest value FFFFh = Highest possible value

25 CTL1 Register (Offset = 14Ch) [Reset = 00000004h]

CTL1 is shown in Figure 26-32 and described in Table 26-34.

Return to the Summary Table.

SPI control register 1

Figure 26-32. CTL1 Register

31	30	29	28	27	26	25	24
RESERVED							
R/W-0h							
23	22	21	20	19	18	17	16
RESERVED							
R/W-0h							
15	14	13	12	11	10	9	8
RESERVED						SUSPEND	PTEN
R/W-0h						R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
RESERVED	PES	PREN	MSB	POD	CP	LBM	ENABLE
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-1h	R/W-0h	R/W-0h

Table 26-59. CTL1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-10	RESERVED	R/W	0h	
9	SUSPEND	R/W	0h	Suspend external communication 0h = Functional mode resumed 1h = External communication suspended
8	PTEN	R/W	0h	Parity transmit enable If enabled, parity transmission will be done for both controller and peripheral modes. 0h = Parity transmission is disabled 1h = Parity transmission is enabled
7	RESERVED	R/W	0h	
6	PES	R/W	0h	Even Parity Select 0h = Odd Parity mode 1h = Even Parity mode
5	PREN	R/W	0h	Parity receive enable If enabled, parity reception check will be done for both controller and peripheral modes In case of a parity miss-match the parity error flag RIS.PER will be set. 0h = Disable Parity receive function 1h = Enable Parity receive function
4	MSB	R/W	0h	MSB first select. Controls the direction of the receive and transmit shift register. 0h = LSB first 1h = MSB first
3	POD	R/W	0h	Peripheral-mode: Data output disabled This bit is relevant only in Peripheral mode. In multiple-peripheral system topologies, SPI controller can broadcast a message to all peripherals, while only one peripheral drives the line. POD can be used by the SPI peripheral to disable driving data on the line. 0h = SPI can drive the POCI output in peripheral mode. 1h = SPI cannot drive the POCI output in peripheral mode.
2	CP	R/W	1h	Controller or peripheral mode select. This bit can be modified only when SPI is disabled, CTL1.ENABLE=0. 0h = Select Peripheral mode 1h = Select Controller Mode

Table 26-59. CTL1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
1	LBM	R/W	0h	Loop back mode 0h = Disable loopback mode 1h = Enable loopback mode
0	ENABLE	R/W	0h	SPI enable 0h = Disable module function 1h = Enable module function

27.1 CAN-FD

The Modular Controller Area Network (MCAN) peripheral supports both communication through classic CAN and CAN-FD protocols.

27.1.1 MCAN Overview

The Controller Area Network (CAN) is a serial communications protocol that efficiently supports distributed real-time control with a high level of reliability. CAN has high immunity to electrical interference and the ability to detect various type of errors. In CAN, many short messages are broadcast to the entire network, which provides data consistency in every node of the system.

The MCAN module supports both classic CAN and CAN FD (CAN with flexible data-rate) protocols. The CAN FD feature allows higher throughput and increased payload per data frame. Classic CAN and CAN FD devices can coexist on the same network without any conflict provided that partial network transceivers, which can detect and ignore CAN FD without generating bus errors, are used by the classic CAN devices. The MCAN module is compliant to ISO 11898-1:2015.

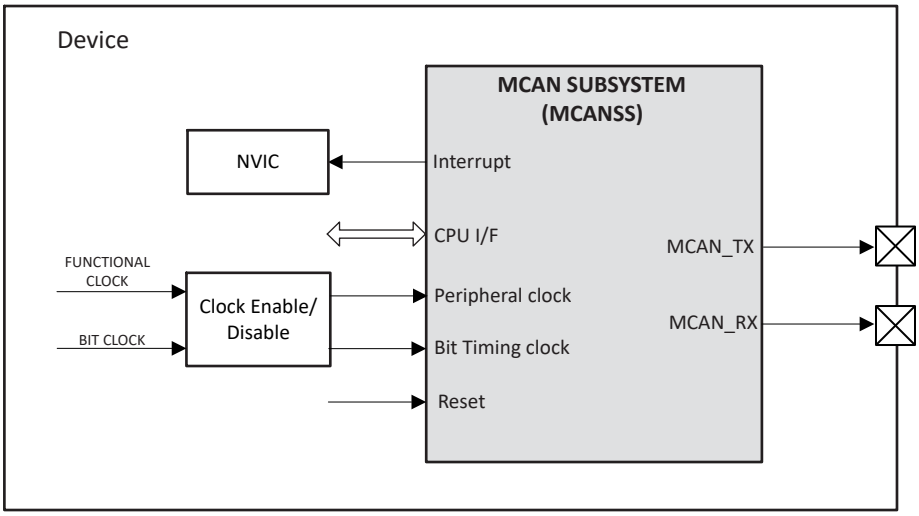


Figure 27-1. MCAN Block Diagram

27.1.1.1 MCAN Features

The MCAN module implements the following features:

- Conforms with CAN Protocol 2.0 A, B and ISO 11898-1:2015
- Full CAN FD support (up to 64 data bytes)
- AUTOSAR and SAE J1939 support
- Up to 32 dedicated transmit buffers
- Configurable transmit FIFO, up to 32 elements
- Configurable transmit queue, up to 32 elements
- Configurable transmit Event FIFO, up to 32 elements
- Up to 14 dedicated receive buffers
- Two configurable receive FIFOs, up to 14 elements each with 1kB message RAM.
- Up to 128 filter elements
- Loop-back mode for self-test
- Maskable interrupt
  - 2 configurable interrupt lines
  - Correctable ECC
  - Counter overflow
  - Clock stop or wakeup
- Non-maskable interrupt (uncorrectable ECC)
- Two clock domains (CAN clock and host clock)
- ECC check for Message RAM
- Clock stop and wakeup support
- Timestamp counter
- Up to 1-Mbps nominal bit rate, up to 8-Mbps data bit rate

Non-supported features:

- Host bus firewall
- Clock calibration
- Debug over CAN

27.1.2 MCAN Environment

The CAN network physical layer consists of a two-wire differential bus, usually twisted pair, and provides a high level of interference immunity. An external CAN transceiver IC is needed to access the bus.

Figure 27-2 shows typical MCAN wiring. Table 27-1 describes the external signals of the MCAN module.

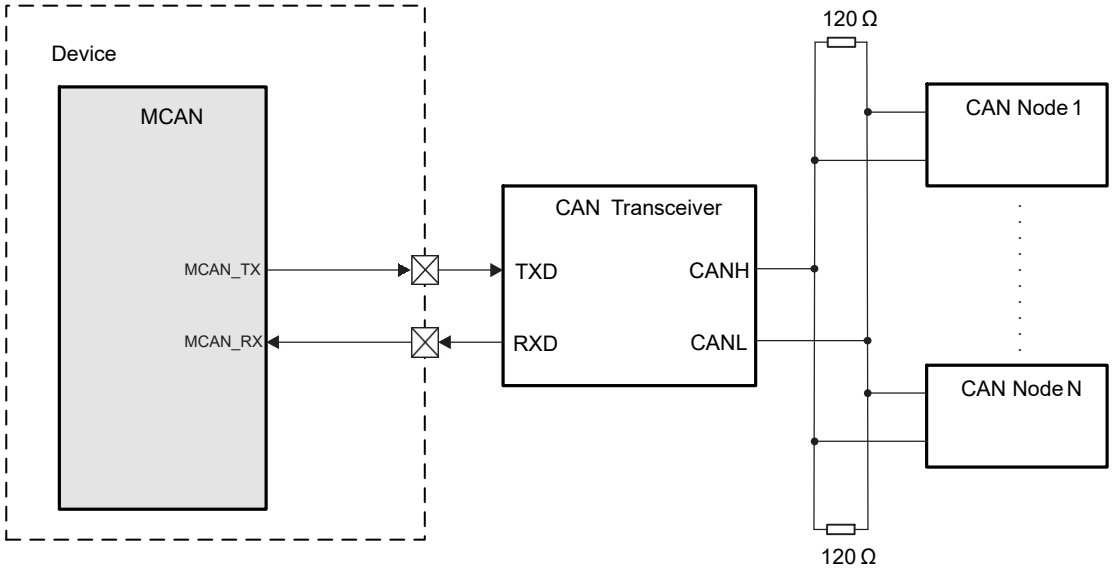


Figure 27-2. MCAN Typical Bus Wiring

Table 27-1. MCAN I/O Description

Module Signal	I/O	Description	Value at Reset
MCAN_RX	Input	Serial data input from external CAN transceiver.	HiZ
MCAN_TX	Output	Serial data output to external CAN transceiver.	HiZ

27.1.3 CAN Network Basics

The network basics are:

- The CAN bus is a 2-wire differential bus using non-return-to-zero (NRZ) encoding and has two states:
  - Recessive state (logical 1)
  - Dominant state (logical 0)
- When the bus is idle, any node can initiate a transmission to any other node. When two or more nodes (ECUs) attempt to transmit at the same time, a nondestructive arbitration technique sends messages in order of priority so that no messages are lost.
- The message transmission is multicast. Data messages transmitted are identifier-based, not address-based.
- The content of the message is labeled by the identifier that is unique throughout the network (for example: RPM, temperature, position, pressure, and so forth).
- All nodes on the network receive the message and each performs an acceptance test on the identifier. If the message is relevant, it is processed; otherwise, it is ignored.
- The unique identifier also determines the priority of the message (the lower the numerical value of the identifier, the higher the priority).
- Data is transmitted and received using message frames, consisting of the following basic fields:
  - Arbitration field
  - Control field
  - Data field (up to 8 bytes for classical CAN and up to 64 bytes for CAN FD)
  - CRC field
  - ACK field

For more information, see *ISO 11898-1:2015: CAN data link layer and physical signaling*.

MCAN power up and clock sequence :

1. Select CANCLK from the available clock sources: the HFCLK (XTAL or HFCLK_IN) or the SYSPLL (SYSPLLCLK0 or SYSPLLCLK2X).
2. Enable the corresponding clock source.
3. Wait until the clock source is stable (SYSCTL_CLKSTATUS[*GOOD] bit = 1).
4. Enable MCAN power by setting PWREN for MCAN.
5. Wait until MCAN is ready (SYSCTL_SYSSTATUS[MCAN0READY] = 1).

27.1.4 MCAN Functional Description

The MCAN module performs CAN protocol communication according to ISO 11898-1:2015. The bit rate can be programmed to values up to 8Mbit/s. Additional transceiver hardware is required for the connection to the physical layer (CAN bus).

For communication on a CAN network, individual message frames can be configured. The message frames and identifier masks are stored in the Message RAM.

All functions concerning the handling of messages are implemented in the Message Handler.

The register set of the MCAN module can be accessed directly through the module interface. These registers are used to control and configure the CAN core and the Message Handler, and to access the Message RAM.

Figure 27-3 shows the MCAN module block diagram, followed by the description of the MCAN module blocks.

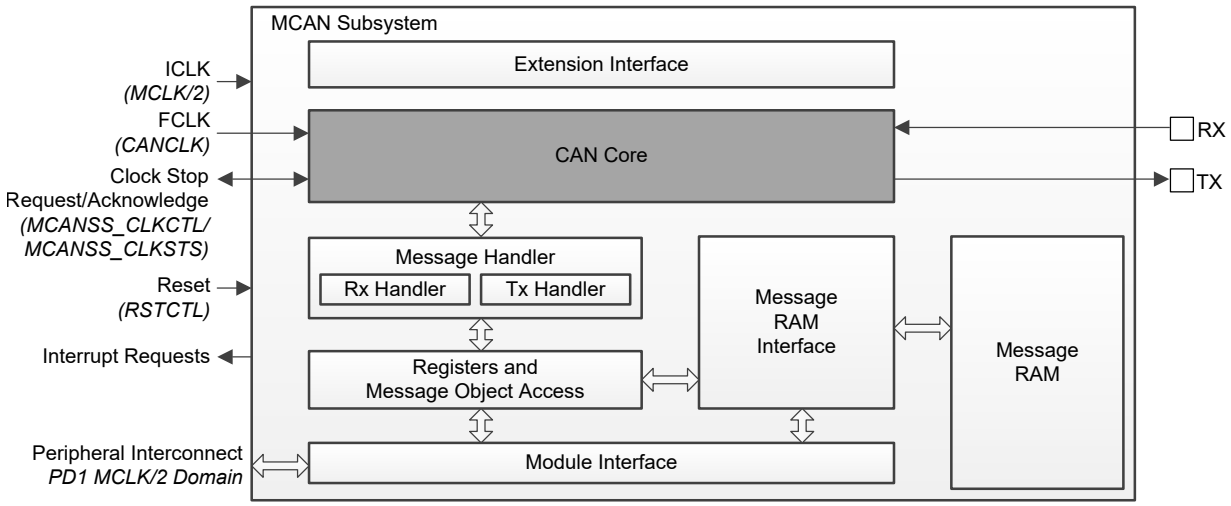


Figure 27-3. MCAN Block Diagram

- **CAN Core**: The CAN core consists of the CAN protocol controller and the Rx/Tx shift register. It handles all ISO 11898-1:2015 protocol functions and supports 11-bit and 29-bit identifiers.
- **Message Handler**: the Message Handler (Rx Handler and Tx Handler) is a state machine that controls the data transfer between the single-ported Message RAM and the CAN core's Rx/Tx shift register. It also handles the acceptance filtering and Interrupt generation as programmed in the control registers.
- **Message RAM**: the main purpose of the Message RAM is to store Rx/Tx messages, Tx Event elements, and Message ID Filter elements (for more information, see Section 27.1.4.19).
- **Message RAM Interface**: enables a connection between the Message RAM and the other blocks in the MCAN module.
- **Registers and Message Object Access**: Data consistency is ensured by indirect accesses to the message objects. During normal operation, all software and DMA accesses to the Message RAM are done through interface registers. The interface registers have the same word-length as the Message RAM.
- **Module Interface**: The MCAN module registers are accessed by the user's software through a 32-bit peripheral bus interface.
- **Clocking**: Two clocks are provided to the MCAN module: the peripheral synchronous clock (interface clock - MCAN_ICLK sourced from MCLK/2) and the peripheral asynchronous clock (functional clock - MCAN_FCLKsourced from CANCLK).

- **Extension Interface**: All selected internal status and control signals are routed to this interface (except for the indication signals of configuration change enable bit (MCAN_CCCR.CCE) and Interrupt Register bits (MCAN_IR).

27.1.4.1 Clock Setup

Peripheral asynchronous clock (MCAN_FCLK) for MCAN can be clocked either through HFXT or SYSPLL. This clock configuration has to be done through SYSCTL REGISTERS. *Refer to CANCLK (CAN-FD Functional Clock) under the Clocks section.*

27.1.4.2 Module Clocking Requirements

Two clocks are provided to the MCAN module:

- Host Clock : peripheral synchronous clock (MCAN_ICLK) as the general module clock source, and
- CAN Clock: peripheral asynchronous clock (MCAN_FCLK) provided to the CAN core for generating the CAN bit timing.

Within the MCAN module, there is a synchronization mechanism implemented to make sure there is safe data transfer between the two clock domains. There is synchronization between the signals from the Host clock domain to the CAN clock domain and conversely, and between the reset signal (MCAN_RST) to the Host clock domain and to the CAN clock domain.

Note

MCAN_ICLK must always be higher or equal to MCAN_FCLK to achieve a stable functionality of the MCAN module: $f_{ICLK} \geq f_{FCLK}$

The CAN-FD supports higher speeds of operation and as such has more stringent timing requirements than the classic CAN. For performance, TI recommends using the lowest N-divider value that maintains a working PLL REF_CLK for the system. Lower N-divider values increase the loop bandwidth of the PLL, which in turn improves timing margins for CAN-FD.

The peripheral asynchronous clock (MCAN_FCLK) can be clocked either through HFXT or SYSPLL. The configuration of this has to be done through the SYSCTL registers. Refer to the *Clocks* module for more information

27.1.4.3 Interrupt Requests

The MCAN module generates interrupt requests that are configured through the Host CPU. Placing the MCAN module in suspend mode prevents the interrupt requests from propagating to the Host CPU. The MCAN core has two interrupt lines and 30 internal interrupt sources. Each source can be configured to drive one of the two interrupt lines. The interrupts are 'level high' interrupts. The MCAN core provides two interrupt requests (MCANSS_INT0 and MCANSS_INT1).

For more information, see the following registers:

- Interrupt Register (MCAN_IR)
- Interrupt Enable (MCAN_IE)
- Interrupt Line Select (MCAN_ILS)
- Interrupt Line Enable (MCAN_ILE)

The MCAN module supports an External Timestamp Counter. The External Timestamp Counter produces an interrupt when it rolls over (see Section 27.1.4.12.1).

For more information, see the following registers:

- Interrupt Clear Shadow Register (MCANSS_ICS)
- Interrupt Raw Status Register (MCANSS_IRS)
- Interrupt Enable Clear Shadow Register (MCANSS_IECS)
- Interrupt Enable Register (MCANSS_IE)
- Interrupt Enable Status Register (MCANSS_IES)
- End Of Interrupt Register (MCANSS_EOI)
- External Timestamp Prescaler Register (MCANSS_EXT_TS_PRESCALER)
- External Timestamp Unserviced Interrupts Counter Register (MCANSS_EXT_TS_UNSERVICED_INTR_CNTR)

To clear IRQ_INT0, IRQ_INT1 and TS_WAKE interrupts, write to the EOI bitfield for the corresponding interrupt number that is described in the MCANSS_EOI register.

After servicing an interrupt (external timestamp, interrupt 0/1), write ‘1’ in the corresponding bit in MCANSS_EOI register to clear the interrupt. In case of an ECC interrupt, after clearing the ECC interrupt

source, application software must also write a '1' to the EOI registers (MCANERR_SEC_EOI.EOI_WR/ CANERR_DED_EOI.EOI_WR). For more information see Section 27.1.4.14.2

The IRQ sequence is as follows:

1. Enable one of the interrupt sources by setting corresponding bit in IMASK.
2. Set MCANSS_IE to enable the IRQ, set MCANSS_ILS for line0 or line1.
3. Wait for the interrupt source to be triggered.
4. The interrupt is triggered and the application jumps into IRQ service routine (ISR).
5. In the ISR, check if IRQ is triggered by the expected source by reading IIDX. Reading IIDX clears the IRQ source, so there is no need to write ICLR to clear the IRQ.

   Due to the design of MCAN, step (6) is required in the ISR to clear IRQ. Check the interrupt source to determine if step 7 is needed:
6. Write 1 to MCAN_IR to clear the interrupt.
7. If the source is IRQ_INT0, IRQ_INT1, or TS_WAKE, also clear MCANSS_EOI, otherwise the next IRQ will not be recognized.

27.1.4.4 Operating Modes

The operating modes are discussed in the following sections.

27.1.4.4.1 Normal Operation

Once the MCAN module is initialized and the MCAN_CCCR.INIT bit is reset to zero, the MCAN module synchronizes itself to the CAN bus and is ready for communication. After passing the acceptance filtering, received messages including Message Identifier (ID) and Data Length Code (DLC) are stored into a dedicated Rx Buffer or into Rx FIFO 0/Rx FIFO 1.

For messages to be transmitted, dedicated Tx buffers, and a Tx FIFO or a Tx queue can be initialized or updated.

Note

Automated transmission upon reception of remote frames is not supported.

27.1.4.4.2 CAN Classic

CAN supports transmission of data up to 8 bytes with standard (11 bit) or (29 bit) identifier.

27.1.4.4.3 CAN FD Operation

The CAN FD standard allows extended frames to be sent, up to 64 data bytes in a single frame at a higher bit rate for the data phase of a frame, up to 8 Mbps. The CAN FD standard introduces the ability to switch from one bit rate to another. Extended Data Length (EDL), as shown in Figure 27-4 and described in Table 27-2, sets a data length of up to 8 or 64 data bytes. Bit Rate Switching (BRS) indicates whether two bit rates (the data phase is transmitted at a different bit rate compared to the arbitration phase) are enabled.

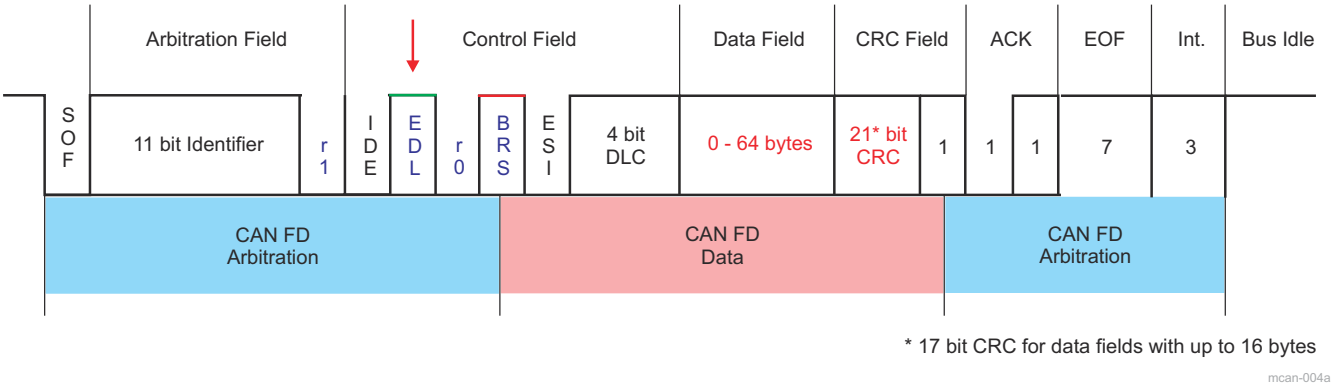


Figure 27-4. CAN FD Frame

Table 27-2. CAN FD Frame Description

Bit/Field	Description
SOF	Start of Frame
Arbitration	Contains the identifier (address) of the message. Used for determining which node gets to transmit.
Control	Contains information about the frame format, including the Data Length Code (DLC, specifies # of data bytes in the frame)
Data	Message data: Up to 8 bytes in classic CAN, up to 64 bytes in CAN-FD
IDE	Identifier extension (for 29 bit extended ID)
FDF	Flexible Data Format
BRS	Bit Rate Switching
ESI	Error Status Indicator

Table 27-2. CAN FD Frame Description (continued)

Bit/Field	Description
DLC	Data Length Code
CRC	Cyclic Redundancy check
ACK	Acknowledgement bits to indicate successful reception of the frame

There are two variants of CAN FD frame transmission:

- CAN FD frame transmission without bit rate switching
- CAN FD frame transmission where control field, data field, and CRC field are transmitted with a higher bit rate than the beginning and the end of the frame

In the CAN frames, FDF = recessive (logical 1) signifies a CAN FD frame, FDF = dominant (logical 0) signifies a Classic CAN frame. In a CAN FD frame, the two bits following FDF - res and BRS, decide whether the bit rate inside of this CAN FD frame is switched. A CAN FD bit rate switch is signified by res = dominant and BRS = recessive. Note that the coding of res = recessive is reserved for future expansion of the protocol.
If the MCAN module receives a frame with FDF = recessive and res = recessive, the MCAN signals a Protocol Exception Event by setting the MCAN_PSR.PXE bit. When Protocol Exception Handling is enabled (MCAN_CCCR.PXHD = 0), this causes the operation state to change from Receiver (MCAN_PSR.ACT = 10) to Integrating (MCAN_PSR.ACT = 00) at the next sample point. In case Protocol Exception Handling is disabled (MCAN_CCCR.PXHD = 1), the MCAN treats a recessive bit as an error and responds with an error frame.

CAN FD operation is enabled by programming the MCAN_CCCR.FDOE bit. If MCAN_CCCR.FDOE = 1, transmission and reception of CAN FD frames is enabled. Transmission and reception of Classic CAN frames is always possible. Whether a CAN FD frame or a Classic CAN frame is transmitted can be configured using the FDF bit in the respective Tx Buffer element.

With MCAN_CCCR.FDOE = 0, received frames are interpreted as Classic CAN frames, which leads to the transmission of an error frame when receiving a CAN FD frame. When CAN FD operation is disabled, no CAN FD frames are transmitted even if the FDF bit of a Tx Buffer element is set. The MCAN_CCCR.FDOE and MCAN_CCCR.BRSE bits can only be changed while the MCAN_CCCR.INIT and MCAN_CCCR.CCE bits are both set. With MCAN_CCCR.FDOE = 0, the setting of bits FDF and BRS is ignored and frames are transmitted in Classic CAN format.

With MCAN_CCCR.FDOE = 1 and MCAN_CCCR.BRSE = 0, only FDF bit of a Tx Buffer element is evaluated. With MCAN_CCCR.FDOE = 1 and MCAN_CCCR.BRSE = 1, transmission of CAN FD frames with bit rate switching is enabled. All Tx Buffer elements with bits FDF and BRS set are transmitted in CAN FD format with bit rate switching.

A mode change during CAN operation is only recommended under the following conditions:

- The failure rate in the CAN FD data phase is significantly higher than in the CAN FD arbitration phase. In this case, disable the CAN FD bit rate switching option for transmissions.
- During system startup, all nodes are transmitting Classic CAN messages until verified that the nodes are able to communicate in CAN FD format. If this is true, all nodes switch to CAN FD operation.
- Wakeup messages in CAN Partial Networking have to be transmitted in Classic CAN format.
- End-of-line programming in case not all nodes are CAN FD capable. Non CAN FD nodes are held in Silent mode until programming has completed. Then all nodes switch back to Classic CAN communication.

The coding of the DLC in the CAN FD format differs from the standard CAN format. The DLC codes 0 to 8 have the same coding as in standard CAN (0 to 8 data bytes), the codes 9 to 15, which in standard CAN all code a data field of 8 bytes, are coded according to Table 27-3.

Table 27-3. DLC Coding in CAN FD

DLC	9	10	11	12	13	14	15
Number of Data Bytes	12	16	20	24	32	48	64

CAN-FD Bit Timing

For CAN FD frames, the bit timing is switched inside the frame after the BRS (Bit Rate Switch) bit in case this bit is recessive. In the CAN FD arbitration phase, before the BRS bit, the nominal CAN bit timing (see Figure 27-5) is used as configured by the Nominal Bit Timing and Prescaler Register (MCAN_NBTP). In the following CAN FD data phase, the data phase bit timing is used as configured by the Data Bit Timing and Prescaler Register (MCAN_DBTP). The bit timing is switched back from the data phase timing at the CRC delimiter or when an error is detected, whichever occurs first.

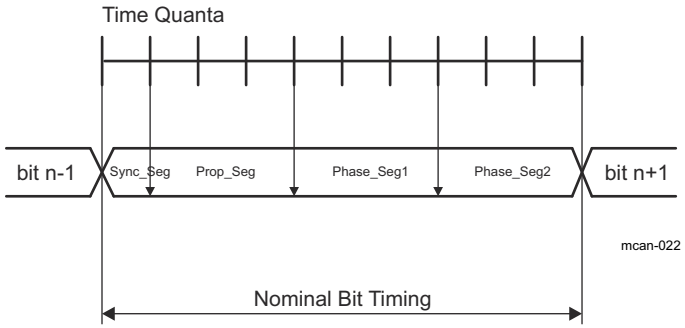


Figure 27-5. CAN Bit Timing

The maximum configurable data phase bit timing depends on the CAN clock frequency (MCAN_FCLK). Example: with MCAN_FCLK = 20MHz and the shortest configurable bit time of 4 t_q (time quanta), the bit rate in the data phase is 5 Mbit/s.

In both data frame formats, CAN FD and CAN FD with bit rate switching, the value of the Error Status Indicator (ESI) bit depends on the transmitter error state (see MCAN_PSR.RESI bit) monitored at the start of the transmission. If the transmitter has an error passive flag, the ESI bit is transmitted recessive; else, the ESI bit is transmitted dominant.

27.1.4.5 Software Initialization

A software initialization begins when the MCAN_CCCR.INIT bit is set to 1. This is done either by software or by a hardware reset, when an uncorrected bit error is detected in the Message RAM, or by going to a Bus_Off state. While the MCAN_CCCR.INIT bit is set, the message transfer is stopped and the status of the output TX pin is recessive (high). The counters of the Error Management Logic (EML) are unchanged. Setting the MCAN_CCCR.INIT bit does not change any configuration register. Resetting the MCAN_CCCR.INIT bit finishes the software initialization. After waiting for the occurrence of a sequence of 11 consecutive recessive bits (indication for Bus_Idle state) the message transfer starts.

Access to the MCAN configuration registers is only enabled when both MCAN_CCCR.INIT and MCAN_CCCR.CCE bits are set (write protection).

The MCAN_CCCR.CCE bit can only be set/reset while the MCAN_CCCR.INIT = 1. The MCAN_CCCR.CCE bit is automatically reset when the MCAN_CCCR.INIT bit is reset.

The following registers are reset when the MCAN_CCCR.CCE bit is set:

- MCAN_HPMS - High Priority Message Status
- MCAN_RXF0S - Rx FIFO 0 Status
- MCAN_RFX1S - Rx FIFO 1 Status
- MCAN_TXFQS - Tx FIFO/Queue Status
- MCAN_TXBRP - Tx Buffer Request Pending
- MCAN_TXBTO - Tx Buffer Transmission Occurred
- MCAN_TXBCF - Tx Buffer Cancellation Finished
- MCAN_TXEFS - Tx Event FIFO Status

The Timeout Counter value MCAN_TOCV.TOC field is preset to the value configured by the MCAN_TOCC.TOP field when the MCAN_CCCR.CCE bit is set.

In addition, the Tx Handler and Rx Handler are held in idle state while MCAN_CCCR.CCE = 1.

The following registers are only writable while MCAN_CCCR.CCE = 0

- MCAN_TXBAR - Tx Buffer Add Request
- MCAN_TXBCR - Tx Buffer Cancellation Request

MCAN_CCCR.TEST and MCAN_CCCR.MON bits can only be set by the Host CPU while MCAN_CCCR.INIT = 1 and MCAN_CCCR.CCE = 1. Both bits are reset at any time. The MCAN_CCCR.DAR bit can only be set/reset while MCAN_CCCR.INIT = 1 and MCAN_CCCR.CCE = 1.

Table 27-4 shows the steps to configure the MCAN module.

Table 27-4. Steps to Configure MCAN Module

Step	Operation	Description	Pseudo Code
1	Initialize MCAN_CCCR	Set MCAN_CCCR.INIT bit and check that the bit has been set	INIT = 1; If INIT ≠ 1, wait until set
2	Unlock protected registers	Set MCAN_CCCR.CCE bit	CCE = 1;
3	Configure CAN mode	Set MCAN_CCCR.FDOE bit to CAN FD	FDOE = 1 for CAN FD FDOE = 0 for Classic CAN
4	Configure Bit Rate Switching	Set MCAN_CCCR.BRSE bit	BRSE = 1 for bit rate switching BRSE = 0 for no bit rate switching
5	Set nominal bit timing(1)	Set MCAN_NBTP register	
6	Lock protected registers	Clear MCAN_CCCR.CCE bit	CCE = 0;
7	Return MCAN module to normal operation	Clear MCAN_CCCR.INIT bit and check that the bit has been cleared	INIT = 0; If INIT ≠ 0, wait until cleared

(1) See the MCAN_NBTP register on how to program CAN bit timing in the *MCAN_REGS Registers* section.

ADVANCE INFORMATION

27.1.4.6 Transmitter Delay Compensation

27.1.4.6.1 Description

When only one CAN FD node is transmitting and all other nodes are receivers, the length of the bus line has no impact. When transmitting using the TX pin, the MCAN module receives the transmitted data from its CAN transceiver using the RX pin. The received data is delayed. If the transmitter delay is greater than TSEG1 (time segment before sample point), a bit error is detected.

The MCAN module provides a delay compensation mechanism to compensate for the transmitter delay. The compensation mechanism enables transmission with higher bit rates during the CAN FD data phase independent of the delay of a specific CAN transceiver. Without transmitter delay compensation the bit rate in the data phase is limited by the transmitter delay.

The mechanism enables configurations where the data bit time is shorter than the transmitter delay (it is described in detail in ISO 11898-1:2015). The transmitter delay compensation is enabled by setting the MCAN_DBTP.TDC bit to 1.

The delayed transmit data is compared against the received data at the Secondary Sample Point (SSP) to check for bit errors during the data phase of transmitting nodes. If a bit error is detected, the transmitter reacts on this bit error at the next following regular sample point. During the arbitration phase, the delay compensation is always disabled.

The received bit is compared against the transmitted bit at the SSP. The SSP position is defined as the sum of the measured delay from the MCAN transmit output TX pin through the transceiver to the receive input RX pin plus the transmitter delay compensation offset configured by the MCAN_TDCR.TDCO field (see Figure 27-6).

The transmitter delay compensation offset is used to adjust the position of the SSP inside the received bit (example: half of the bit time in the data phase). The position of the SSP is rounded down to the next integer number of mtq.

The actual transmitter delay compensation value can be checked by reading the MCAN_PSR.TDCV field. This field is cleared when the MCAN_CCCR_INIT bit is set and is updated at each transmission of CAN FD frame while the MCAN_DBTP.TDC bit is set.

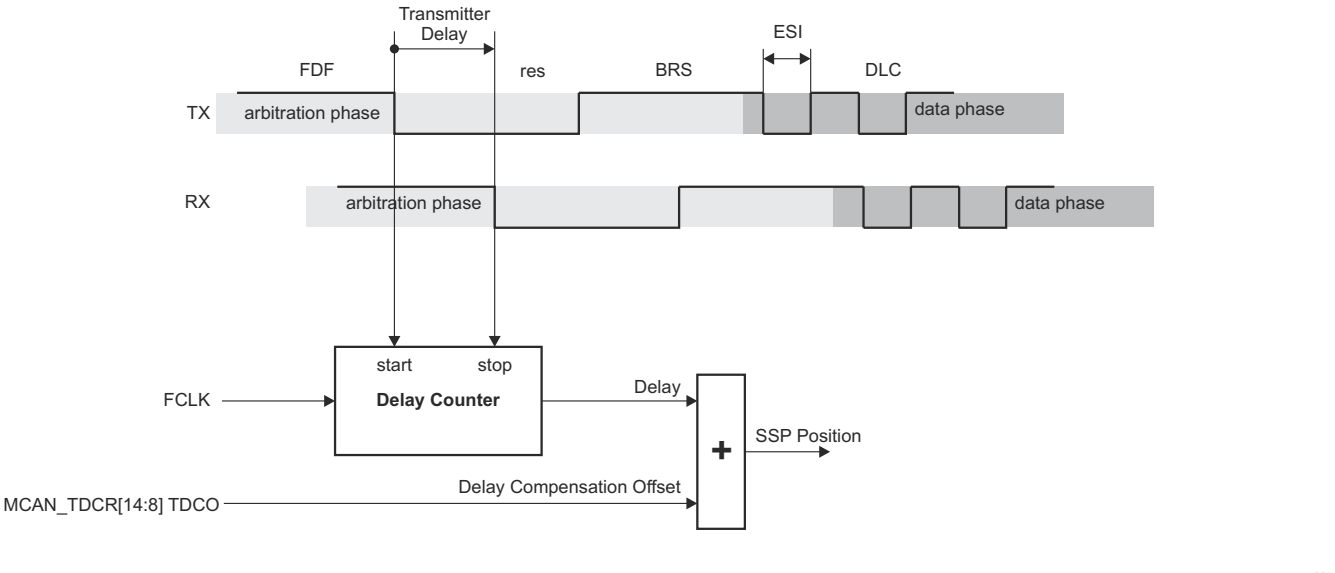


Figure 27-6. Transmitter Delay Measurement

ADVANCE INFORMATION

27.1.4.6.2 Transmitter Delay Compensation Measurement

When transmitter delay compensation is enabled (by programming MCAN_DBTP.TDC = 1), the measurement is started within each transmitted CAN FD frame at the falling edge of FDF bit to bit r0. The measurement is stopped when this edge is seen at the receive input RX pin of the transmitter. The resolution of this measurement is one mtq (see Figure 27-6). The mtq (minimum time quantum) dimension is equal to the CAN clock period (MCAN_FCLK).

The use of a transmitter delay compensation filter window can be enabled by programming the MCAN_TDCR.TDCF field. This filter feature defines a minimum value for the SSP position to avoid the case in which a dominant glitch inside the received FDF bit ends the delay compensation measurement before the falling edge of the received res bit, resulting in an early taken SSP position. Dominant edges on the RX pin that result in an earlier SSP position are ignored for transmitter delay measurement. The measurement is stopped when the SSP position is at least MCAN_TDCR.TDCF field and the RX pin is low.

The following boundary conditions have to be considered:

- The sum of the measured delay from the TX pin to the RX pin and the configured transmitter delay compensation offset (MCAN_TDCR.TDCO field) has to be less than six bit times in the data phase.
- The sum of the measured delay from the TX pin to the RX pin and the configured transmitter delay compensation offset (MCAN_TDCR.TDCO) field has to be less or equal 127 mtq. In case this sum exceeds 127 mtq, the maximum value of 127 mtq is used for transmitter delay compensation.
- The data phase ends at the sample point of the CRC delimiter, that stops checking of receive bits at the SSPs.

27.1.4.7 Restricted Operation Mode

In restricted operation mode, the CAN node is able to receive data and remote frames and give acknowledgment to valid frames, but the node does not send data frames, remote frames, active error frames, or overload frames. In the case of an error condition or overload condition, the node does not send dominant bits; instead the node waits for the occurrence of bus idle condition to resynchronize itself to the CAN communication. The receive and transmit error counters (MCAN_ECR.REC and MCAN_ECR.TEC) are frozen while CAN error logging (MCAN_ECR.CEL) is active. The Host CPU can set the MCAN module into Restricted Operation Mode by setting the MCAN_CCCR.ASM bit. The bit can only be set by the Host CPU at any time when both MCAN_CCCR.CCE and MCAN_CCCR.INIT bits are set to 1.

The restricted operation mode is automatically entered when the Tx Handler is not able to read data from the Message RAM in time. To leave restricted operation mode, the Host CPU has to reset the MCAN_CCCR.ASM bit. This mode can be used in applications that adapt themselves to different CAN bit rates. In this case, the application tests different bit rates and leaves the restricted operation mode after the node has received a valid frame.

Note

The Restricted Operation Mode must not be combined with the Loop Back Mode.

27.1.4.8 Bus Monitoring Mode

Entering bus monitoring mode is done by setting the MCAN_CCCR.MON bit to 1. In this mode (see ISO 11898-1:2015, *Bus Monitoring* section), the MCAN module is able to receive valid data and remote frames, but cannot start a transmission. The MCAN module sends only recessive bits on the CAN bus. If the MCAN module is required to send a dominant bit (ACK bit, overload flag, active error flag), the bit is rerouted internally so that the MCAN module monitors this dominant bit, although the CAN bus may remain in recessive state. In bus monitoring mode, the MCAN_TXBRP register is held in reset state. The bus monitoring mode can be used to analyze the traffic on a CAN bus without affecting the bus by the transmission of dominant bits. Figure 27-7 shows the connection of the TX and RX signals to the MCAN module in bus monitoring mode.

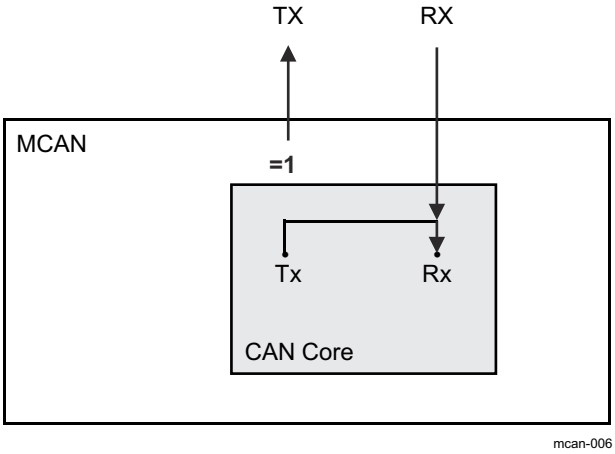


Figure 27-7. Connection of Signals in Bus Monitoring Mode

27.1.4.9 Disabled Automatic Retransmission (DAR) Mode

According to the CAN Specification (see ISO11898-1:2015, *Recovery Management* section), the MCAN module provides means for automatic retransmission of frames that have lost arbitration or that have been disturbed by errors during transmission. By default automatic retransmission is enabled (see the MCAN_CCCR.DAR bit).

27.1.4.9.1 Frame Transmission in DAR Mode

In DAR mode all transmissions are automatically canceled after they have started on the CAN bus. A Tx buffer's Tx Request Pending (MCAN_TXBRP[xx]) TRPx bit is reset after successful transmission, when a transmission has not yet been started at the point of cancellation, has been aborted due to lost arbitration, or when an error occurred during frame transmission.

Successful transmission:

- Corresponding Tx Buffer Transmission Occurred (MCAN_TXBTO.TOx) TOx bit is set
- Corresponding Tx Buffer Cancellation Finished (MCAN_TXBCF.CFx) CFx bit is not set

Successful transmission in spite of cancellation:

- Corresponding Tx Buffer Transmission Occurred (MCAN_TXBTO.TOx) TOx bit is set
- Corresponding Tx Buffer Cancellation Finished (MCAN_TXBCF.CFx) CFx bit is set

Arbitration lost or frame transmission disturbed:

- Corresponding Tx Buffer Transmission Occurred (MCAN_TXBTO.TOx) TOx bit is not set
- Corresponding Tx Buffer Cancellation Finished (MCAN_TXBCF.CFx) CFx bit is set

In the case of a successful frame transmission, and if storage of Tx events is enabled, a Tx Event FIFO element is written with Event Type ET = 10 (transmission in spite of cancellation).

27.1.4.10 Clock Stop Mode

Entering the clock stop mode can be achieved by programming one of two bits:

- Clock stop request bit in MCAN_IP (MCAN_CCCR.CSR)
- Stop request bit in MCAN_WRAPPER (MCANSS_CLKCTL.STOPREQ)

The register bit within the MCAN_IP (MCAN_CCCR.CSR) reads as 1 as long as the MCAN_WRAPPER bit (MCANSS_CLKCTL.STOPREQ) is asserted.

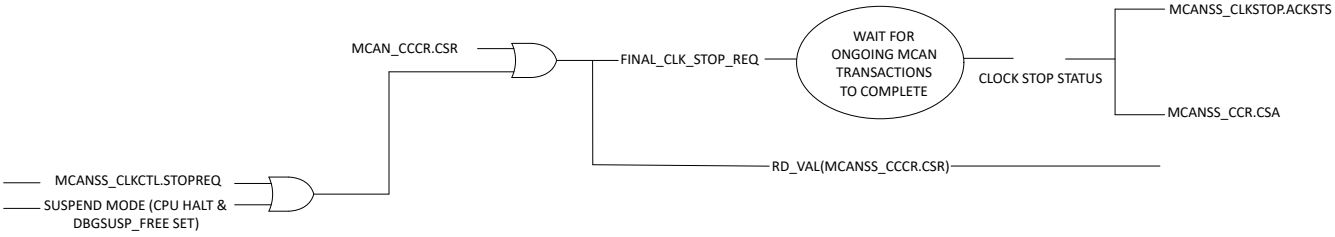


Figure 27-8. Clock Stop Request

After all pending transmission requests have completed, the MCAN module waits until the bus idle state is detected and then sets MCAN_CCCR.INIT to 1 to prevent further CAN transfers. The MCAN module then acknowledges the MCAN is ready for power down by setting the clock stop acknowledge bit MCAN_CCCR.CSA to 1 (MCANSS_CLKSTS.CLKSTOP_ACKSTS reflects the same value as well). In this state, before clocks are switched off, further register accesses can be made. However, a write access to the MCAN_CCCR.INIT bit has no effect. Module clock inputs MCAN_ICLK and MCAN_FCLK can now be switched off using MCANSS_CLKEN.CLK_REQEN.

To exit the power-down mode, the application has to turn on module clocks before clearing the clock stop request bit (make sure both MCAN_CCCR.CSR and MCANSS_CLKCTL.STOPREQ are cleared). MCAN acknowledges removal of the clock request by clearing the MCANSS_CLKSTS.CLKSTOP_ACKSTS

ADVANCE INFORMATION

and MCAN_CCCR.CSA bits. The application can now restart CAN communication by clearing the MCAN_CCCR.INIT bit.

Automatic wakeup from the power-down mode (due to activity on MCAN Rx) is supported through the MCANSS_CTRL.AUTOWAKEUP and MCANSS_CTRL.WAKEUPREQEN bits (for more information, see Section 27.1.4.10.2).

Clock stop and automatic wakeup is illustrated in Figure 27-9.

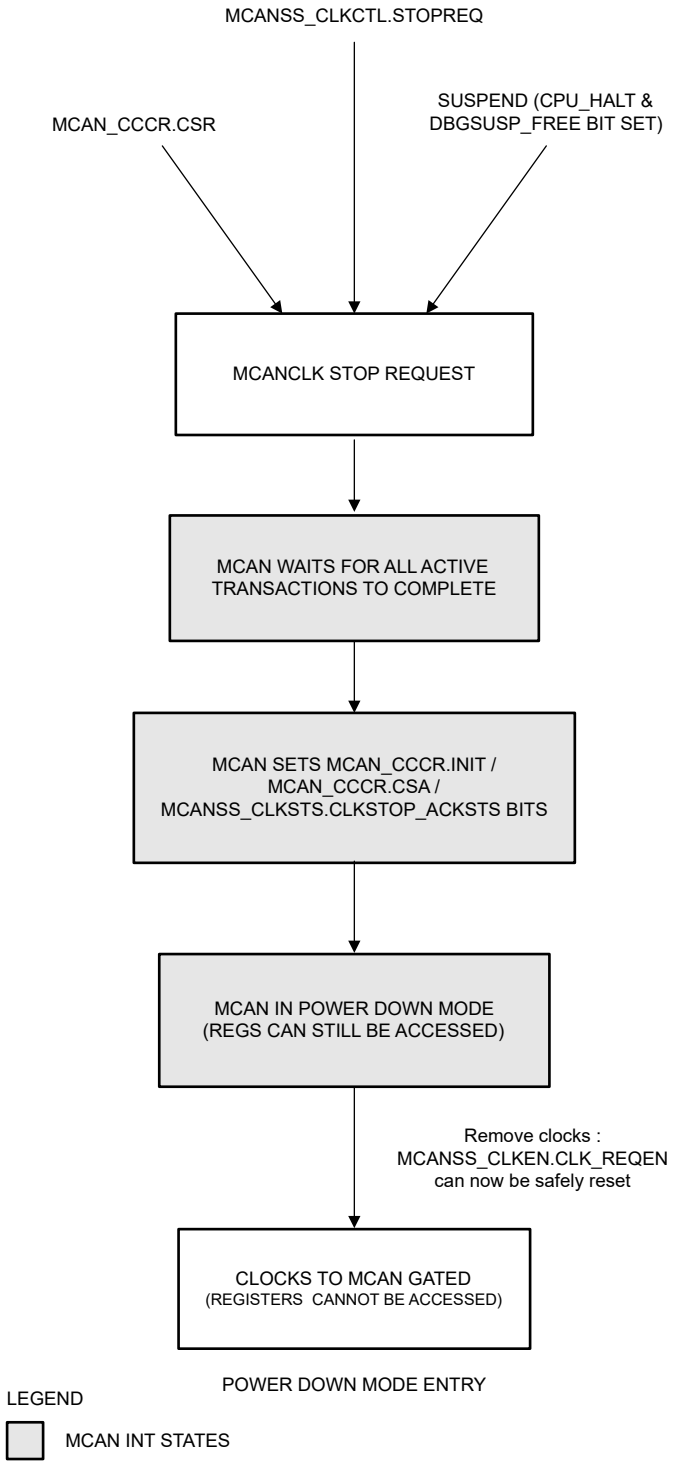


Figure 27-9. Power Down Entry

27.1.4.10.1 Suspend Mode

The MCAN module supports two suspend modes:

- Immediate
- Graceful

In a graceful suspend mode (see the MCANSS_CTRL.DBGSUSP_FREE bit) when the suspend request is asserted, a clock stop request to the MCAN core is performed. The MCAN core responds with a clock stop acknowledge when all pending Tx messages have been processed and an Idle line had been detected. At that point, the MCAN_CCCR.INIT bit is set and the MCAN core stays Idle. The suspend state can be verified by reading the MCAN_CCCR.INIT bit.

The automatic wakeup feature is enabled by setting the MCANSS_CTRL.AUTOWAKEUP and MCANSS_CTRL.WAKEUPREQEN bits to 1 (for more information, see Section 27.1.4.10.2). When suspend request is removed, if no external clock stop request is active, a read-modify-write to the MCAN_CCCR.INIT bit is performed to clear the bit.

During suspend mode the auto-clear feature is disabled. The following register fields have an auto-clear feature:

- MCAN_ECR.CEL
- MCAN_PSR.LEC
- MCAN_PSR.DLEC
- MCAN_PSR.RESI
- MCAN_PSR.RBRS
- MCAN_PSR.RFDF
- MCAN_PSR.PXE

27.1.4.10.2 Wakeup Request

Issuing a clock stop request puts the MCAN module into power-down mode (Sleep Mode). During transition from IDLE to ACTIVE, if the MCANSS_CTRL.AUTOWAKEUP and MCANSS_CTRL.WAKEUPREQEN bits are enabled, after the MCAN Core respond to the removal of the clock stop request with removing the clock stop acknowledge, a read-modify-write is issued to clear the MCAN_CCCR.INIT bit and the MCAN core resumes operation. Please note that after clock stop request has been removed by the hardware the first frame (wakeup frame) is not received. This is because after the clock stop is issued; there are no active clocks running into the IP. Therefore after removing the clock stop request; the wakeup frame that enables clock has to be re-transmitted.

If the MCANSS_CTRL.WAKEUPREQEN bit is set, the MCAN module provides a wakeup request on the following wakeup event:

- The receive RX pin is dominant (logical 0)

The wakeup request is deasserted when any of the following conditions occur:

- Clock stop request is removed and clock stop acknowledge is deasserted
- A reset is applied to the MCAN module

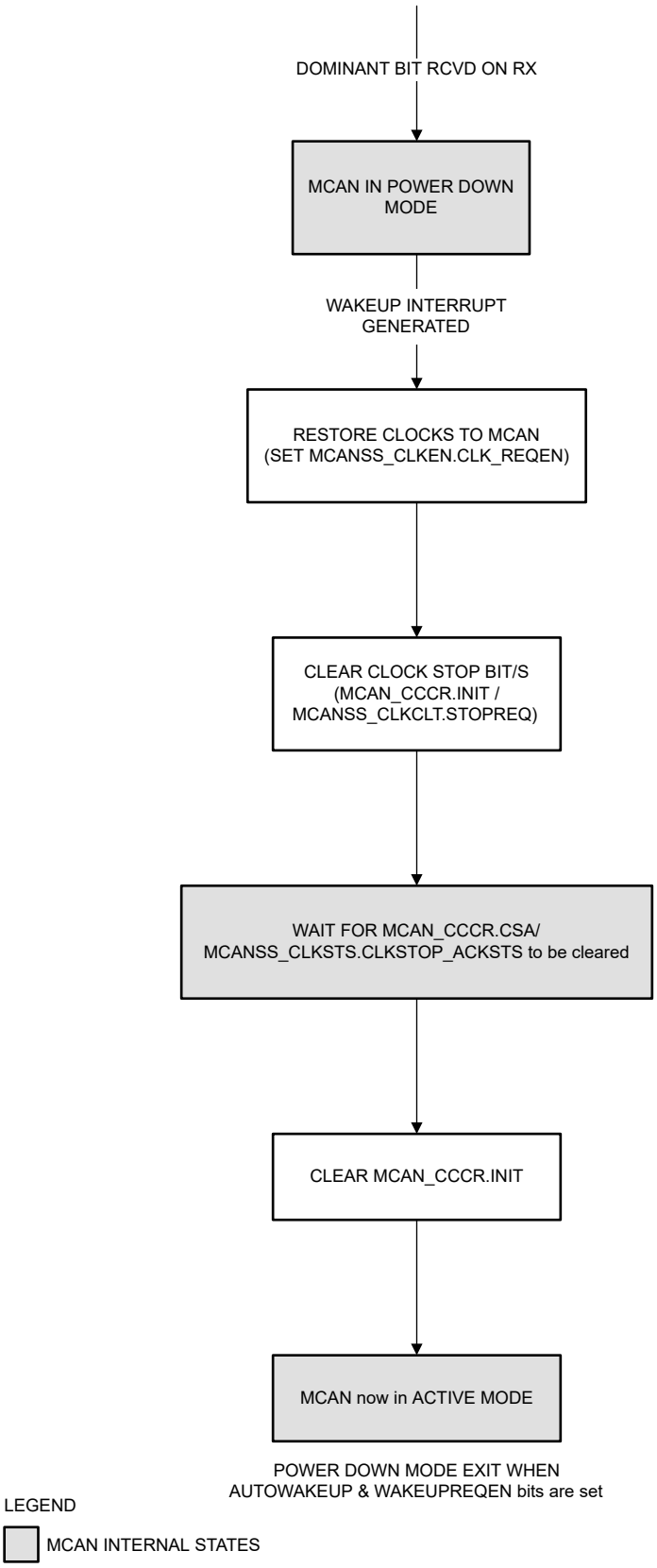


Figure 27-10. Auto Wakeup Enabled Exit from Power Down

Please note the suggested user notes while issuing a wakeup request.

- If there is a message that is being received on the CAN Bus when a Clock Stop Request (CSR) is issued; it will create a wakeup interrupt at the end of the reception resulting in the MCAN module not entering the clock stop mode. To avoid unnecessary wake ups; the user should check the node activity bit (ACT) in the protocol status register(PSR) ensuring an idle state before setting the request.
- The user should also avoid creating a SW polling loop to only exit with a stop clock acknowledge (CSA) since the timing of the software check and wake up interrupt clearing the CSA could cause an infinite software loop
- If there is excessive noise on the Rx pin, it may cause frequent wake up. To avoid this it is recommended that the user waits for a specified time to see if a wake up was received and if not, assert the clock stop state.
- The wakeup logic can be unexpectedly disabled if MCAN is reset while clock stop request is asserted. Therefore, the user should avoid asserting a soft reset to MCAN while CSR is asserted.

27.1.4.11 Test Modes

The MCAN_TEST register write access is enabled by setting the test mode enable MCAN_CCCR.TEST bit to 1. The MCAN_TEST register allows the configuration of the test modes and test functions.

The transmit (TX) pin has four different output functions which can be selected by programming the MCAN_TEST.TX field. The default function is the serial data output. The pin can also be driven with a constant dominant or recessive value. It is also possible to drive the sample-point signal to monitor the bit-timing.

The actual value of the receive (RX) pin can be monitored from MCAN_TEST.RX bit. Both functions can be used to check the physical layer. Due to the synchronization mechanism between the CAN clock (MCANx_FCLK) and Host clock (MCANx_ICLK) domain, there can be a delay of several Host clock periods between writing to the MCAN_TEST.TX field until the new configuration is visible at the output TX pin. This applies also when reading input RX pin by way of the MCAN_TEST.RX bit.

Note

Test modes can be used for self-test only. The software control for TX pin interferes with all CAN protocol functions. It is not recommended to use test modes for an application.

27.1.4.11.1 External Loop Back Mode

The MCAN module can be set into external loop back mode by programming MCAN_TEST.LBCK to 1. In loop backmode, the MCAN treats its own transmitted messages as received messages and stores them (if they pass acceptance filtering) into an Rx Buffer or an Rx FIFO as shown in below figure. Figure shows the connection of the TX and RX pins to the MCAN module in external loop back mode.

This mode is provided for hardware self-test. To be independent from external stimulation, the MCAN module ignores acknowledge errors (recessive bit sampled in the acknowledge slot of a data/remote frame) in loop back mode. In this mode, the MCAN module performs an internal feedback from the Tx output to the Rx input. The actual value of the RX input pin is disregarded by the MCAN module. The transmitted messages are monitored at the TX pin.

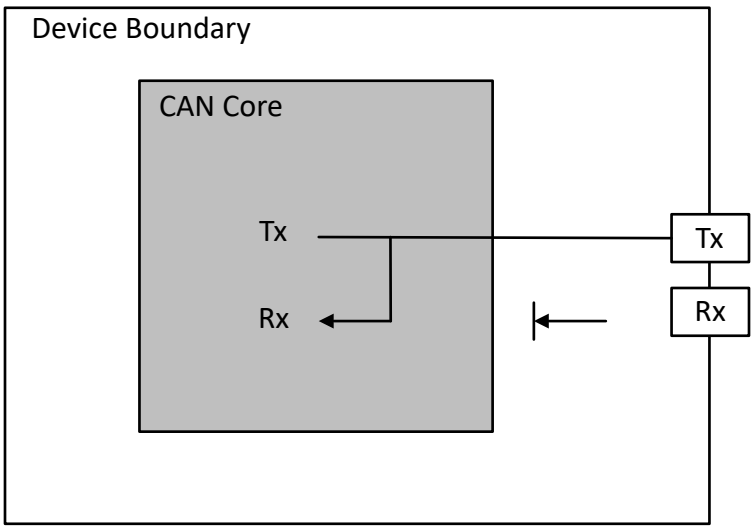


Figure 27-11. External Loop Back Mode

27.1.4.11.2 Internal Loop Back Mode

The MCAN module can be set into internal loop back mode by programming MCAN_TEST.LBCK and MCAN_CCCR.MON bits to 1. The internal loop back mode is used for a Hot Self-test. The Hot Self-test allows

the MCAN module to be tested without affecting a running CAN system connected to the TX and RX pins. In this mode, the RX pin is disconnected from the MCAN module and the TX pin is held recessive. Figure 27-12 shows the connection of the TX and RX pins to the MCAN module in internal loop back mode.

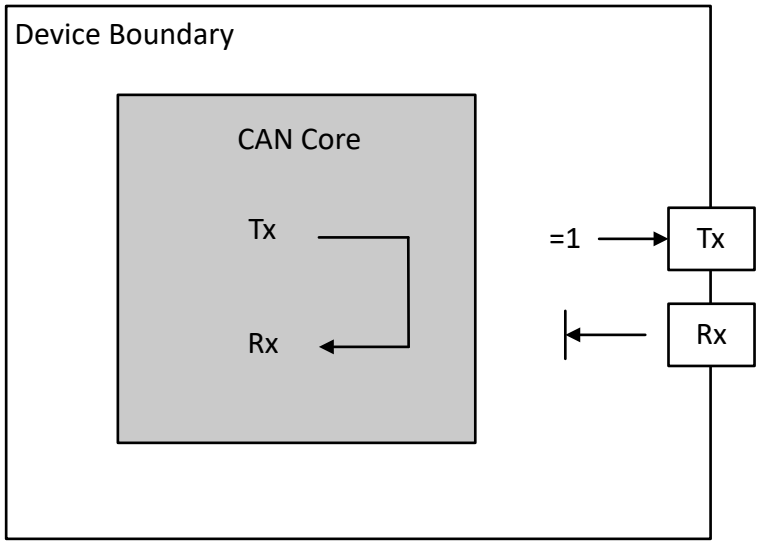


Figure 27-12. Internal Loop Back Mode

27.1.4.12 Timestamp Generation

The MCAN module has an integrated 16-bit wrap-around counter for timestamp generation. The timestamp counter prescaler MCAN_TSCC.TCP field can be configured to clock the counter in multiples of CAN bit times (1-16). The counter is readable by way of the MCAN_TSCV.TSC field. A write access to the MCAN_TSCV register resets the counter to zero. When the timestamp counter wraps around the interrupt MCAN_IR.TSW flag is set. On start of a frame reception/transmission the counter value is captured and stored into the timestamp section of an Rx Buffer/Rx FIFO (RXTS[15:0]) or Tx Event FIFO (TXTS[15:0]) element. For more information regarding FIFO timestamp configuration, see Section 27.1.4.19 .

27.1.4.12.1 External Timestamp Counter

For CAN FD operation mode, the MCAN core requires an external timestamp counter. An externally generated 16-bit vector can substitute the integrated 16-bit CAN bit time counter (internal timestamp counter) for receive and transmit timestamp generation. An external 16-bit timestamp counter can be used by programming the MCAN_TSCC.TSS field.

The external timestamp counter uses the interface clock (MCANx_ICLK) as a reference clock. The MCAN core accepts a 16-bit timestamp. A 24-bit prescaler provides a programmable resolution for the timestamp (see MCANSS_EXT_TS_PRESCALER.PRESCALER bit field). The external timestamp counter can be enabled or disabled through the MCANSS_CTRL.EXT_TS_CNTR_EN bit. When disabled, the counter is reset back to zero. While enabled, the counter keeps incrementing. When the timestamp rolls over, the MCAN_IRQ_TS interrupt is generated.

When the timestamp rolls over, the MCANSS_IRS register is set. The MCANSS_IE register can be affected by writing to the MCANSS_IESS register to set or to the MCANSS_IECS register to clear. The MCANSS_IESS register is a shadow register mapped to the same address as the MCANSS_IE register. The level interrupt is a reflection of both MCANSS_IRS and MCANSS_IE being set. The MCANSS_IES register reflects the level interrupt. When a rollover event occurs, the interrupt counter is incremented. Writing to the MCANSS_ICS register to clear the MCANSS_IRS register also decrements the interrupt counter. Writing to the MCANSS_EOI register issues another pulse, if the interrupt counter is not zero.

The rollover event can be artificially simulated by software through writing to the Interrupt Set Shadow register (MCANSS_ISS). The MCANSS_ISS register is a shadow register mapped to the same address as the MCANSS_IRS register.

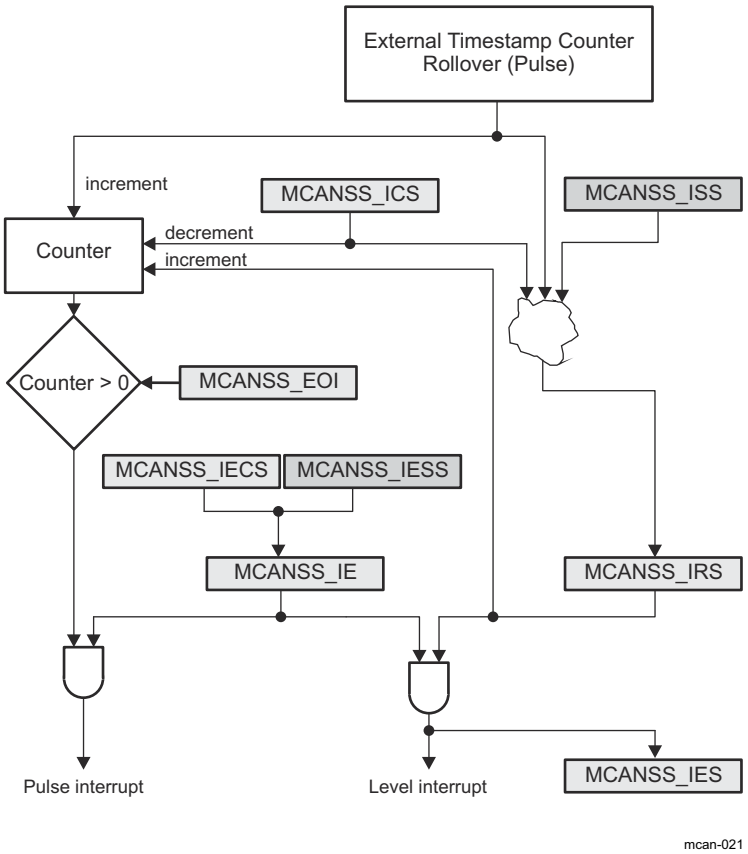


Figure 27-13. External Timestamp Counter Interrupt

27.1.4.13 Timeout Counter

The MCAN module has an integrated 16-bit timeout counter. The timeout counter is used to signal timeout conditions for the Rx FIFO 0, Rx FIFO 1, and Tx Event FIFO Message RAM elements. The timeout counter is configured using the MCAN_TOCC register and is enabled using the MCAN_TOCC.ETOC bit. The timeout counter operates as down-counter and uses the same prescaler programmed by the MCAN_TSCC.TCP field as the timestamp counter. The actual counter value can be monitored from the MCAN_TOCV.TOC field. The timeout counter can be started only when MCAN_CCCR.INIT = 0 and stopped when MCAN_CCCR.INIT = 1 (example: when the MCAN enters Bus_Off state). The operation mode is selected by the MCAN_TOCC.TOS field. When continuous mode is selected, the counter starts when MCAN_CCCR.INIT = 0, a write to the MCAN_TOCV register presets the counter to the value configured by the MCAN_TOCC.TOP field and continues down-counting.

In case the timeout counter is controlled by one of the FIFOs, an empty FIFO presets the counter to the value configured by the MCAN_TOCC.TOP field. Down-counting is started when the first FIFO element is stored. Writing to the MCAN_TOCV register has no effect. When the counter reaches zero, the interrupt MCAN_IR.TOO flag is set.

In continuous mode, the counter is immediately restarted at the value configured by the MCAN_TOCC.TOP field.

Note

The clock signal for the timeout counter is derived from the CAN core sample point signal. Therefore, the point in time where the timeout counter is decremented can vary due to the synchronization/re-synchronization mechanism of the CAN core. If the bit rate switch (BRS) feature in CAN-FD is used, the timeout counter is clocked differently in arbitration and data field.

27.1.4.14 Safety

The Message RAM (MSG_RAM) is wrapped in an ECC wrapper providing SECDED parity functionality. The MCAN ECC wrapper is controlled by an ECC aggregator.

27.1.4.14.1 MCAN ECC Wrapper

The ECC wrapper provides single error correction (SEC) and double error detection (DED) parity to the message memory content. The ECC wrapper has side band signals for error notification. The ECC Wrapper implements an error injection test mode.

The error correction is done using a lazy write back. When an error is detected, the error is noted in a FIFO queue that waits for an access gap to write the data back and refresh the memory. If a transaction writes new data to the compromised entry before the lazy write back completes, the write back is discarded.

27.1.4.14.2 MCAN ECC Aggregator

This section describes the functional details of the MCAN ECC aggregator module.

27.1.4.14.2.1 MCAN ECC Aggregator Overview

The MCAN ECC aggregator module supports the following general features:

- Provides a mechanism to control and monitor the ECC RAM in the MCAN module.
- Provides software access to all the ECC related registers.
- Supports software readable status of ECC single/double-bit errors and associated info such as RAM address and data bits that are in error.
- Aggregates level pending status from the ECC RAM into a single interrupt to the Host CPU.

The following feature is not supported:

- Statistics such as tracking the number of single and double-bit errors. If needed, these operations can be handled by software.

27.1.4.14.2.2 MCAN ECC Aggregator Registers

There are three groups of registers in the MCAN ECC aggregator module:

- **Global registers:** Aggregator Revision Register (MCANERR_REV), ECC Vector Register (MCANERR_VECTOR), Misc Status Register (MCANERR_STAT), ECC Control Register (MCANERR_CTRL), and ECC Wrapper Revision Register (MCANERR_WRAP_REV).
- **Control and status registers:** ECC Error Control Registers (MCANERR_ERR_CTRL1 and MCANERR_ERR_CTRL2) and ECC Error Status Registers (MCANERR_ERR_STAT1, MCANERR_ERR_STAT2, and MCANERR_ERR_STAT3).
- **Interrupt registers:** interrupt status, interrupt enable set, interrupt enable clear, and EOI (End Of Interrupt) registers that are part of a standard interrupt module. For more information, see the following registers:
  - MCANERR_SEC_EOI
  - MCANERR_SEC_STATUS
  - MCANERR_SEC_ENABLE_SET
  - MCANERR_SEC_ENABLE_CLR
  - MCANERR_DED_EOI
  - MCANERR_DED_STATUS
  - MCANERR_DED_ENABLE_SET
  - MCANERR_DED_ENABLE_CLR

27.1.4.14.3 Reads to ECC Control and Status Registers

The reads to the ECC control and status registers are triggered by writing a 'read message' to the ECC Vector Register as:

1. Software writes value (the ECC RAM ID) to the MCANERR_VECTOR.ECC_VECTOR field to select the ECC RAM for control or status.
2. Software writes 1 to the MCANERR_VECTOR.RD_SVBUS bit to trigger a read.
3. Software writes read address to the MCANERR_VECTOR.RD_SVBUS_ADDRESS field.
4. Software then polls the MCANERR_VECTOR.RD_SVBUS_DONE bit to check if the bit is 1. This bit indicates that the read operation has completed.
5. Software reads the data from the ECC control or status register. The following clock cycle (MCAN_ICLK) returns the read data.

27.1.4.14.4 ECC Interrupts

The ECC aggregator module aggregates the level pending status from the ECC RAM into a single EOI-handshake based interrupt to the Host CPU. Software is expected to follow the sequence described:

1. Software enables the interrupts for the ECC RAM by writing to the MCANERR_SEC_ENABLE_SET/ MCANERR_DED_ENABLE_SET register.
2. Software writes the ECC RAM ID in the MCANERR_VECTOR.ECC_VECTOR.
3. Software writes the MCANERR_VECTOR.RD_SVBUS bit to trigger the read.
4. Software writes the MCANERR_ERR_STAT1 register address to the MCANERR_VECTOR.RD_SVBUS_ADDRESS field. Software needs to load the 'read message' in the rMCANERR_VECTOR register again, if the software needs to read the MCANERR_ERR_STAT2 register.
5. Software polls the MCANERR_VECTOR.RD_SVBUS_DONE bit. When this bit is set, a read of the MCANERR_ERR_STAT1/MCANERR_ERR_STAT2 register is performed.
6. After the interrupt has been serviced, software clears the interrupt status by writing to the MCANERR_ERR_STAT1.CLR_ECC_SEC or MCANERR_ERR_STAT1.CLR_ECC_DED bit depending on the type of the ECC error.
7. Software polls the MCANERR_ERR_STAT1 register to verify that the status bit has been cleared.
8. Software writes to the MCANERR_SEC_EOI/MCANERR_DED_EOI register to clear the interrupt.
9. After clearing the ECC interrupt source, the application software must also write 1 to the MCANERR_SEC_EOI.EOI_WR /MCANERR_DED_EOI.EOI_WR bits.

27.1.4.15 Tx Handling

The Tx handler is used to handle the Tx requests. It controls the transfer of transmit messages from the dedicated Tx buffers, the Tx FIFO, and the Tx Queue to the CAN Core, the Tx Event FIFO, and the Put and GetIndex operations. The MCAN module supports up to 32 Tx buffers. These Tx buffers can be configured as dedicated Tx buffers, Tx FIFO, or Tx Queue and as combination of dedicated Tx buffers/Tx FIFO or dedicated Tx buffers/Tx Queue. For each Tx Buffer element Classical CAN or CAN FD transmission mode can be configured. Table below shows the possible configurations for message transmission.

Table 27-5. Message Transmission configurations

MCAN_CCCR Register		Buffer Element		Frame Transmission
BRSE	FDOE	FDF	BRS	
ignored	0	ignored	ignored	Classic CAN
0	1	0	ignored	Classic CAN
0	1	1	ignored	CAN FD without bit rate switching
1	1	0	ignored	Classic CAN
1	1	1	0	CAN FD without bit rate switching
1	1	1	1	CAN FD with bit rate switching

When the Tx Buffer RequestPending (MCAN_TXBRP) register is updated, or when a transmission has been started the Tx Handler starts scanning to check for the highest priority pending Tx request. The Tx Buffer with the lowest Message ID has highest priority.

Note

AUTOSAR requires at least three Tx Queue buffers and support of transmit cancellation

ADVANCE INFORMATION

27.1.4.15.1 Transmit Pause

The transmit pause feature is intended for use in CAN networks where the CAN Message IDs are specific and cannot easily be changed. These Message IDs may have a higher priority than other defined Message IDs, while in a specific application their relative priority should be inverse. This allows for a case where one ECU sends a burst of CAN messages that cause CAN messages from another ECU to be delayed (paused).

The transmit pause feature is enabled by the MCAN_CCCR.TXP bit. By default this bit is disabled (MCAN_CCCR.TXP= 0). Each time after successfully transmitted message, a pause for two CAN bit times occurs before the start of the next transmission. This allows the other CAN nodes in the network to transmit messages even if their Message IDs have lower priority.

27.1.4.15.2 Dedicated Tx Buffers

Dedicated Tx buffers are intended for message transmission under complete control of the Host CPU. There are two options.

- Each dedicated Tx Buffer is configured with a specific Message ID
- Two or more dedicated Tx buffers are configured with the same Message ID. In this case the Tx Buffer with the lowest buffer number is transmitted first

After the data section has been updated, a transmission is requested by an Add Request. This is done using the MCAN_TXBAR[x]ARn bit (where x = 0 to 31). The requested messages arbitrate internally with messages from an optional Tx FIFO or Tx Queue and externally with messages on the CAN bus, and are sent out according to their Message ID.

The following table shows the Tx Buffer, Tx FIFO, and Tx Queue Element Size. A Dedicated Tx Buffer allocates element size 32-bit words in the Message RAM. The start address of a dedicated Tx Buffer in the Message RAM is calculated by adding transmit buffer index from 0 to 31 (MCAN_TXFQS.TFQP) × Element size to the Tx Buffer start address MCAN_TXBC.TBSA field

Table 27-6. Tx Buffer, Tx FIFO, Tx Queue Element Size

MCAN_TXESC.TBDS	DataField [Bytes]	Element Size [RAM Words]
000	8	4
001	12	5
010	16	6
011	20	7
100	24	8
101	32	10
110	48	14
111	64	18

ADVANCE INFORMATION

27.1.4.15.3 Tx FIFO

Tx FIFO mode is configured by setting bit MCAN_TXBC.TFQM = 0. The data stored in the Tx FIFO messages are transmitted starting with the message referenced by the Get Index MCAN_TXFQS.TFGI field. After each transmission the Get Index is incremented until the Tx FIFO is empty. The Tx FIFO Free Level MCAN_TXFQS.TFFL field indicates the number of the available free Tx FIFO elements. The Tx FIFO allows transmission of messages with the same Message ID from different Tx buffers in the order these messages have been written to the Tx FIFO.

New transmit messages have to be written to the Tx FIFO starting with the Tx Buffer referenced by the Put Index MCAN_TXFQS.TFQP field. After each Add Request (MCAN_TXBAR[x] ARn = 1) the Put Index is incremented to the next free Tx FIFO element. When the Put Index reaches the Get Index (MCAN_TXFQS.TFQP = MCAN_TXFQS.TFGI), Tx FIFO Full condition is signaled by bit MCAN_TXFQS.TFQF = 1. In this case no further messages should be written to the Tx FIFO until the next message has been transmitted and the Get Index has been incremented.

The number of requested Tx buffers should not exceed the number of free Tx buffers as indicated by the Tx FIFO Free Level MCAN_TXFQS.TFFL field.

In case a transmission request for the Tx Buffer referenced by the Get Index is canceled, the Get Index is incremented to the next Tx Buffer with pending transmission request and the Tx FIFO Free Level MCAN_TXFQS.TFFL field is recalculated. In case transmission cancellation is applied to any other Tx Buffer - the Get Index and the FIFO Free Level remain unchanged.

A Tx FIFO element allocates element size 32-bit words in the Message RAM. The start address of the next available (free) Tx FIFO Buffer is calculated by adding Tx FIFO/Queue Put Index MCAN_TXFQS.TFQP (from 0 to 31) × Element Size to the Tx Buffer Start Address MCAN_TXBC.TBSA field.

27.1.4.15.4 Tx Queue

Tx Queue mode is configured by setting bit MCAN_TXBC.TFQM = 1. The data stored in the Tx Queue messages are transmitted starting with the highest priority message (lowest Message ID). In case two or more Queue buffers are configured with the same Message ID, the Queue Buffer with the lowest buffer number is transmitted first.

New transmit messages have to be written to the Tx FIFO starting with the Tx Buffer referenced by the Put Index MCAN_TXFQS.TFQP field. Each Add Request cyclically increments the Put Index to the next free Tx Buffer. In case of Tx Queue Full condition (MCAN_TXFQS.TFQF = 1), the Put Index is not valid and no further message should be written to the Tx Queue until at least one of the requested messages has been sent out or a pending transmission request has been canceled.

The application may use the MCAN_TXBRP register instead of the Put Index and may place messages to any Tx Buffer without pending transmission request.

A Tx Queue Buffer allocates element size 32-bit words in the Message RAM. The start address of the next available (free) Tx Queue Buffer is calculated by adding Tx FIFO/Queue Put Index MCAN_TXFQS.TFQP (from 0 to 31) × Element Size to the Tx Buffer Start Address MCAN_TXBC.TBSA field.

27.1.4.15.5 Mixed Dedicated Tx Buffers/Tx FIFO

For this combination the Tx buffers section in the Message RAM is separated in two parts:

- Dedicated Tx buffers: the number of dedicated Tx buffers is configured by MCAN_TXBC.NDTB field.
- Tx FIFO: the number of Tx buffers assigned to the Tx FIFO is configured by the MCAN_TXBC.TFQS field. If the MCAN_TXBC.TFQS field is empty(zero)- only dedicated Tx buffers are used.

Tx prioritization:

- Scan Dedicated Tx buffers and oldest pending Tx FIFO Buffer (referenced by the MCAN_TXFQS.TFGI field)
- Buffer with lowest Message ID gets highest priority and is transmitted next
- The following figure shows Mixed Dedicated Tx buffers/Tx FIFO example.

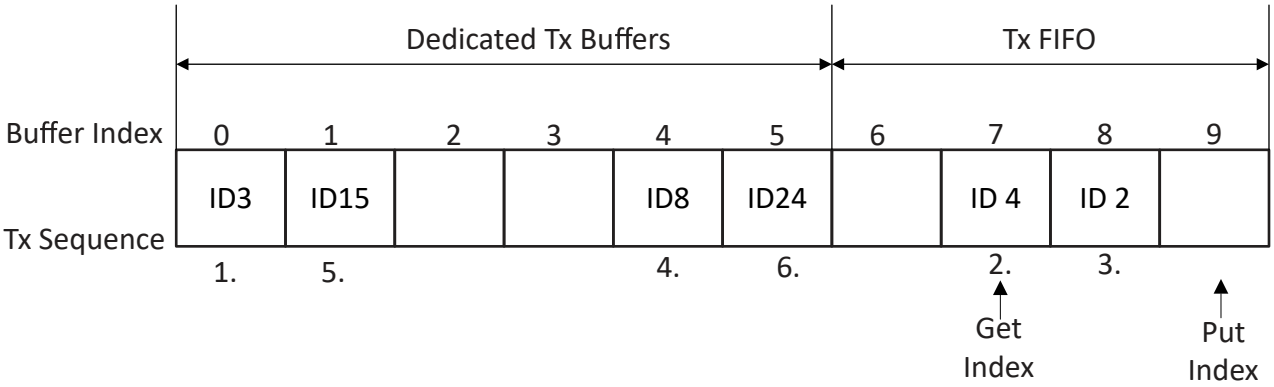


Figure 27-14. Mixed Dedicated Tx Buffers/ Tx FIFO (example)

Note

The SysConfig tool allows for dynamic configuration of MSG_RAM, TX Buffers/FIFO/Queue, and RX Buffers/FIFO/Queue with automatic memory layout/partitioning and auto-generating configuration code. It is strongly recommended to use the SysConfig tool for these configurations.

ADVANCE INFORMATION

27.1.4.15.6 Mixed Dedicated Tx Buffers/Tx Queue

For this combination the Tx buffers section in the Message RAM is separated in two parts:

- Dedicated Tx buffers: the number of Dedicated Tx buffers is configured by the MCAN_TXBC.NDTB field.
- Tx Queue: the number of Tx buffers assigned to the Tx Queue is configured by the MCAN_TXBC.TFQS field If MCAN_TXBC.TFQS field is empty (zero) - only Dedicated Tx buffers are used.

Tx prioritization:

- Scan all Tx buffers with activated transmission request
- Tx Buffer with lowest Message ID gets highest priority and is transmitted next
- The following figure shows Mixed Dedicated Tx buffers/Tx Queue example.

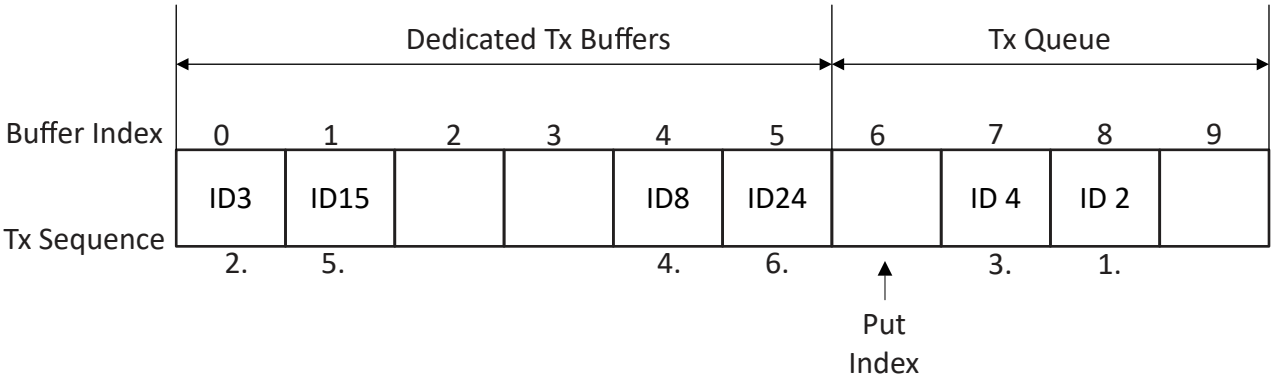


Figure 27-15. Mixed Dedicated Tx Buffers/ Tx FIFO (Example)

27.1.4.15.7 Transmit Cancellation

This feature is especially intended for gateway and AUTOSAR based applications. The Host CPU can cancel a requested transmission from a dedicated Tx Buffer or a Tx Queue Buffer by setting bit MCAN_TXBCR[n] CRn = 1 (where n = 0 - 31). The corresponding bit position n is equivalent to the number of the Tx buffer.

Transmit cancellation is not intended for Tx FIFO operation.

Successful cancellation is signaled by setting the corresponding bit of the MCAN_TXBCF register (MCAN_TXBCF[n] CFn = 1).

If transmission from a Tx Buffer is already ongoing and a transmit cancellation is requested, the corresponding MCAN_TXBRP[n] TRPn bit remains set as long as the transmission is in progress. If the transmission was successful, the corresponding MCAN_TXBTO[n] TOn and MCAN_TXBCF[n] CFn bits are set. If the transmission was not successful, only the corresponding bit MCAN_TXBCF[n] CFn = 1.

Note

If pending transmission is canceled immediately before this transmission could have been started, a short time window occurs where no transmission is started even if another message is also pending in this node. This may enable another node to transmit a message that may have a lower priority than the second message in this node.

27.1.4.15.8 Tx Event Handling

To support Tx Event Handling, the Message RAM has implemented a Tx Event FIFO section. Up to 32 Tx Event FIFO elements can be configured. Please see the Tx Event FIFO element chapter. After message transmission on the CAN bus, Message ID and Timestamp are stored in a Tx Event FIFO element. To link a Tx Event to a Tx Event FIFO element, the Message Marker from the transmitted Tx Buffer is copied into the Tx Event FIFO element.

A Tx Event FIFO full condition is signaled by the MCAN_IR.TEFF bit. In this case no further elements are written to the Tx Event FIFO until at least one element has been read out and the Tx Event FIFO Get Index has been incremented (MCAN_TXEFS.EFGI). In case a Tx Event occurs while the Tx Event FIFO is full, this event is rejected and interrupt flag MCAN_IR.TEFL bit is set.

The Tx Event FIFO watermark can be configured to avoid a Tx Event FIFO overflow. When the Tx Event FIFO fill level reaches the Tx Event FIFO watermark configured by the MCAN_TXEFC.EFWM field, interrupt flag MCAN_IR.TEFW is set. When reading from the Tx Event FIFO, two times the Tx Event FIFO Get Index MCAN_TXEFS.EFGI field has to be added to the Tx Event FIFO start address MCAN_TXEFC.EFSA field.

27.1.4.15.9 FIFO Acknowledge Handling

The Get Indices of the two Rx FIFOs (Rx FIFO 0 or Rx FIFO 1) and the Tx Event FIFO are controlled by writing to the corresponding FIFO Acknowledge Index (see MCAN_RXF0A, MCAN_RXF1A, and MCAN_TXEFA). Writing to the FIFO Acknowledge Index will set the FIFO Get Index to the FIFO Acknowledge Index plus one and thereby updates the FIFO Fill Level.

There are two use cases:

- A single element has been read from the FIFO: the Get Index value is written to the FIFO Acknowledge Index.
- A sequence of elements has been read from the FIFO: the Get Index value (Index of the last element read) is written to the FIFO Acknowledge Index at the end of that read sequence.

The Host CPU has free access to the Message RAM. Special care has to be taken when reading FIFO elements in an arbitrary order (Get Index not considered). This can be useful when reading a High Priority Message from one of the two Rx FIFOs. In this case the FIFO's Acknowledge Index should not be written because this would set the Get Index to a wrong position and also changes the FIFO's Fill Level. In this case some of the older FIFO elements would be lost.

Note

The application has to ensure that a valid value is written to the FIFO Acknowledge Index. The MCAN module does not check for erroneous values.

27.1.4.16 Rx Handling

The Rx Handler controls the following operations:

- Acceptance filtering
- The transfer of received messages to the Rx buffers or to one of the two Rx FIFOs (Rx FIFO 0 or Rx FIFO 1)
- Rx FIFO Put and Get Index operations

27.1.4.16.1 Acceptance Filtering

The MCAN module employs two sets of acceptance filters - one set for standard and one set for extended identifiers. These filters can be assigned to an Rx Buffer or to one of the two Rx FIFOs.

The main features of the filter elements are:

- Each filter element can be configured as:
  - Range filter (from - to)
  - Filter for specific IDs (for one or two dedicated IDs)
  - Classic bit mask filter
- Each filter element can be enabled/disabled individually
- Each filter element can be configured for acceptance or rejection filtering
- Filters are checked sequentially and execution (acceptance filtering procedure) stops at the first matching filter element or when the end of the filter list is reached

Related configuration registers are:

- Global Filter Configuration (MCAN_GFC) register
- Standard ID Filter Configuration (MCAN_SIDFC) register
- Extended ID Filter Configuration (MCAN_XIDFC) register
- Extended ID AND Mask (MCAN_XIDAM) register

Depending on the configuration of the filter element (see SFEC/EFEC in Section 27.1.4.19) if filter matches, one of the following actions is performed:

- Received frame is stored in FIFO 0 or FIFO 1
- Received frame is stored in Rx Buffer
- Received frame is stored in Rx Buffer and generation of pulse at filter event pin is performed. This is high level single MCAN_ICLK pulse.
- Received frame is rejected
- Set High Priority Message interrupt flag MCAN_IR.HPM
- Set High Priority Message interrupt flag MCAN_IR.HPM and store received frame in FIFO 0 or FIFO 1

Acceptance filtering starts when complete Message ID is received. Acceptance filtering stops at the first matching enabled filter element or when the end of the filter list is reached. If a filter element matches - the Rx Handler starts writing the received message data in portions of 32-bit to the matching Rx Buffer or Rx FIFO. If an error condition occurs (for example: CRC error), this message is rejected with the following impact on the affected Rx Buffer or Rx FIFO:

- Rx Buffer: New Data flag (MCAN_NDAT1/MCAN_NDAT2) of matching Rx Buffer is not set, but Rx Buffer (partly) overwritten with received data (for error type, see MCAN_PSR.LEC and MCAN_PSR.DLEC fields,respectively).
- Rx FIFO: Put index of matching Rx FIFO is not updated, but related Rx FIFO element (partly) overwritten with received data (for error type, see MCAN_PSR.LEC and MCAN_PSR.DLEC fields, respectively). If matching Rx FIFO is configured to operate in overwrite mode, the boundary conditions described in Section 27.1.4.17.2 have to be considered.

Note

When an accepted message is written to one of the two Rx FIFOs, or into an Rx Buffer, the unmodified received identifier is stored independent of the filters used. The result of the acceptance filter process is strongly depending on the sequence of configured filter elements.

27.1.4.16.1.1 Range Filter

Each filter element can be configured to operate as Range Filter (Standard Filter Type SFT = 00/Extended Filter Type EFT = 00). The filter matches for all received message frames with IDs in the range from SFID1 to SFID2 (SFID2 ≥ SFID1) respectively in the range from EFID1 to EFID2 (EFID2 ≥ EFID1). For more information see Section 27.1.4.19.5 and Section 27.1.4.19.6 .

There are two options for range filtering of extended frames:

- Extended Filter Type EFT = 00: The Extended ID AND Mask (MCAN_XIDAM) is used for Range Filtering. The Message ID of received frames is ANDed with the Extended ID AND Mask (MCAN_XIDAM) before the range filter is applied.
- Extended Filter Type EFT = 11: The Extended ID AND Mask (MCAN_XIDAM) is not used for Range Filtering.

27.1.4.16.1.2 Filter for Specific IDs

Each filter element can be configured to filter one or two dedicated Message IDs (Standard Filter Type SFT =01/ Extended Filter Type EFT =01). To filter only one specific Message ID, the filter element has to be configured with SFID1 = SFID2 respectively EFID1 = EFID2. For more information, see Section 27.1.4.19.5 and Section 27.1.4.19.6 .

27.1.4.16.1.3 Classic Bit Mask Filter

Classic bit mask filtering can filter groups of Message IDs (Standard Filter Type SFT =10/Extended Filter Type EFT =10). This is done by masking single bits of a received Message ID. In this case SFID1/EFID1 element is used as Message ID filter, while the SFID2/EFID2 element is used as filter mask.

A 0 bit at the filter mask (SFID2/EFID2) masks out the corresponding bit position of the configured Message ID filter (SFID1/EFID1) and the value of the received Message ID at that bit position is not relevant for acceptance filtering. Only those bits of the received Message ID where the corresponding mask bits are 1 are relevant for acceptance filtering.

There are two interesting cases:

- All mask bits are 1: a match occurs only when the received Message ID and the configured Message ID filter are identical.
- All mask bits are 0: all Message IDs match.

27.1.4.16.1.4 Standard Message ID Filtering

Figure 27-16 shows the standard Message ID (11-bit ID) filtering flow. Section 27.1.4.19.5 describes the standard Message ID filter element.

The Remote Transmission Request (RTR) and Extended Identifier (XTD) bits of the received frames are compared against the list of configured filter elements. This is controlled by the following registers:

- Global Filter Configuration (MCAN_GFC) register
- Standard ID Filter Configuration (MCAN_SIDFC) register

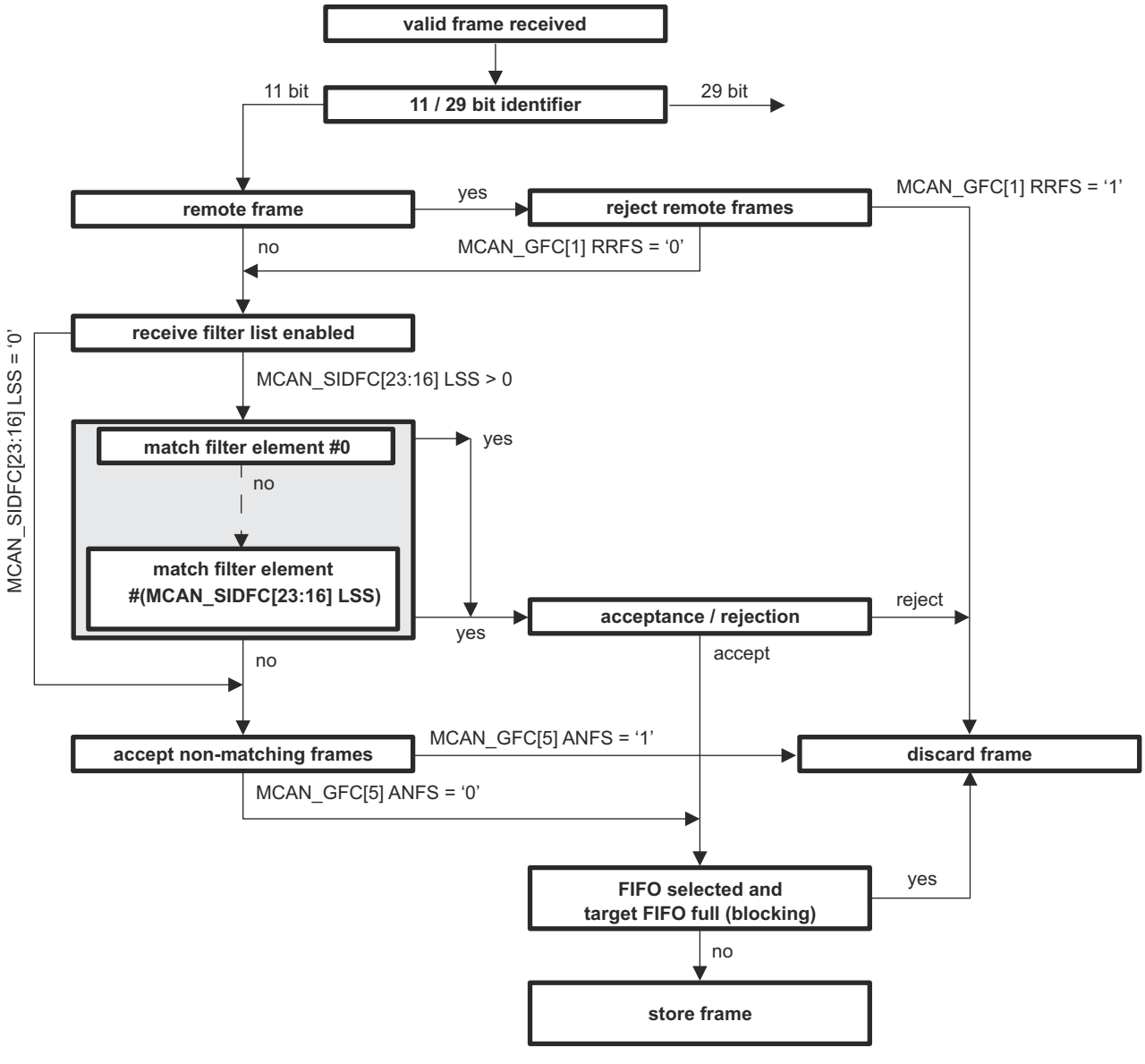


Figure 27-16. Standard Message ID Filter Path

Note

In MCAN_GFC[1]RRFS and MCAN_GFC[5]ANFS; the [1] and [5] denotes the bit position for the RRFS and ANFS bits respectively in the MCAN_GFC register

27.1.4.16.1.5 Extended Message ID Filtering

Figure 27-17 shows the extended Message ID (29-bit ID) filtering flow. Section 27.1.4.19.6 describes the extended Message ID filter element.

The Remote Transmission Request (RTR) and Extended Identifier (XTD) bits of the received frames are compared against the list of configured filter elements. This is controlled by the following registers:

- Global Filter Configuration (MCAN_GFC) register
- Extended ID Filter Configuration (MCAN_XIDFC) register

Note that before the filter list is executed, the received identifier is ANDed with the Extended ID AND Mask (MCAN_XIDAM).

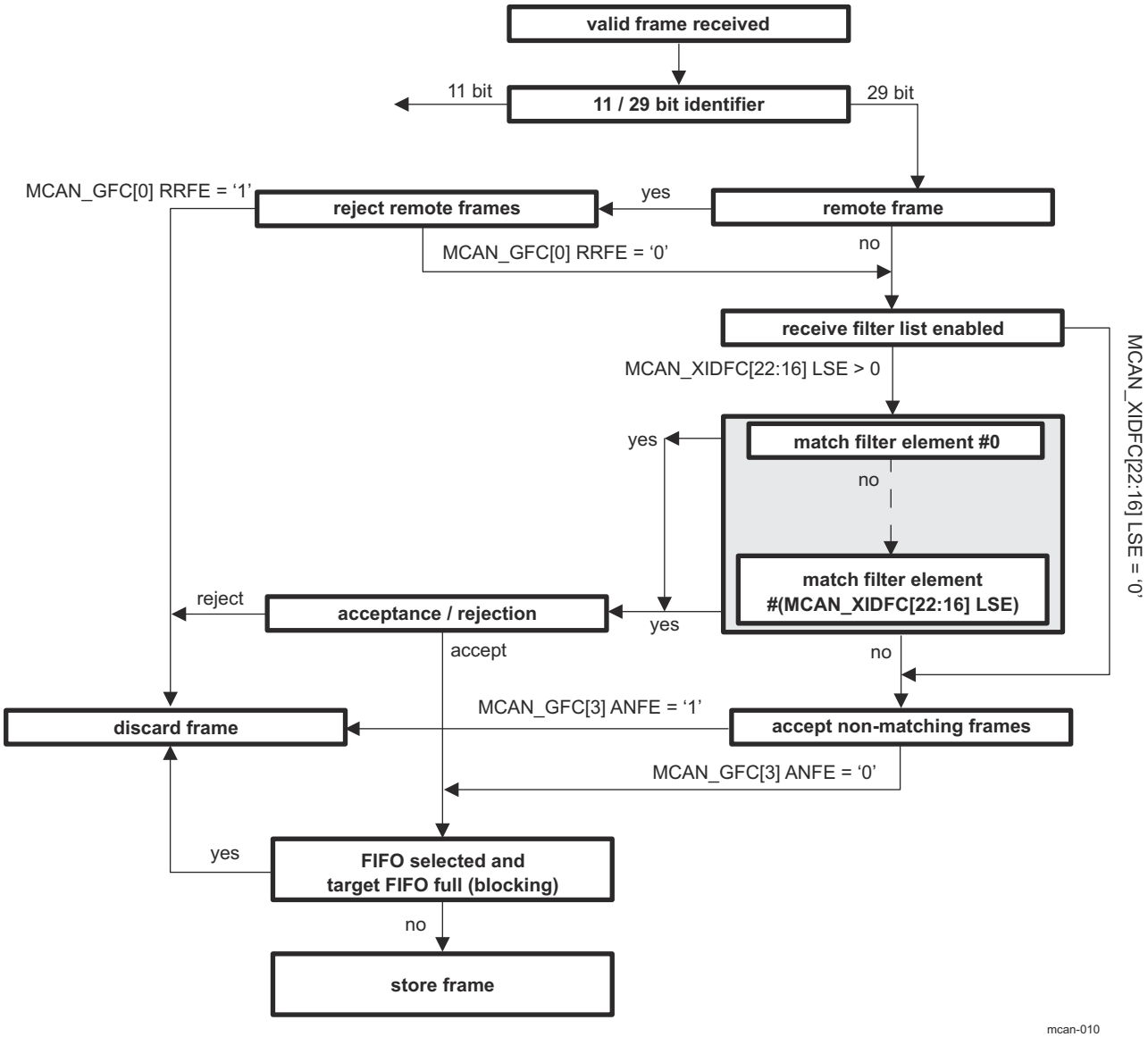


Figure 27-17. Extended Message ID Filter Path

In MCAN_GFC[0]RRFS and MCAN_GFC[3]ANFS; the [0] and [3] denotes the bit position for the RRFS and ANFS bits respectively in the MCAN_GFC register

27.1.4.17 Rx FIFOs

The configuration of the Rx FIFOs (Rx FIFO 0 and Rx FIFO 1) can be done by way of the MCAN_RXF0C and MCAN_RXF1C registers. Each Rx FIFO can be configured to store up to 64 received messages.

After acceptance filtering the received messages that passed are transferred to the Rx FIFO. The filter mechanisms available for the Rx FIFO 0 and Rx FIFO 1 are described in Section 27.1.4.16.1 . The Rx FIFO element is described in Section 27.1.4.19.2 .

The Rx FIFO watermark can be used to prevent an Rx FIFO overflow. If the Rx FIFO fill level reaches the Rx FIFO watermark configured by the MCAN_RXFnC[30:24] FnWM filed (where: n = 0 or 1), an interrupt flag MCAN_IR.RF0W/MCAN_IR.RF1W is set.

When the Rx FIFO Put Index reaches the Rx FIFO Get Index (MCAN_RXFnS[21:16] FnPI = MCAN_RXFnS[13:8] FnGI), an Rx FIFO Full condition is signaled by the MCAN_RXFnS[24] FnF status bit and interrupt flag MCAN_IR.RF0F/MCAN_IR.RF1F is set. Figure 27-18 shows Rx FIFO Status. The FIFOs fill level is presented in the MCAN_RXFnS[6:0] FnFL field (the number of elements stored in Rx FIFO).

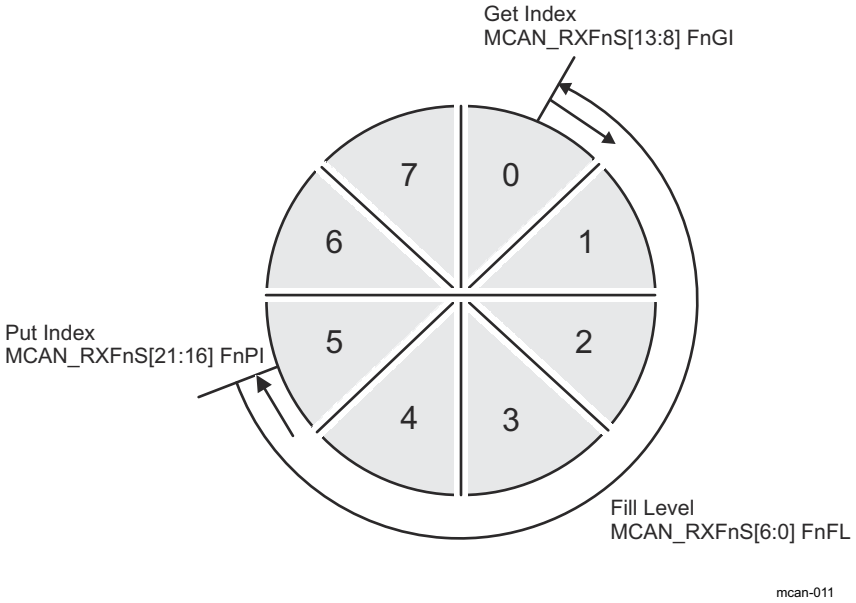


Figure 27-18. Rx FIFO Status

Rx FIFOs start address in the Message RAM (MCAN_RXFnC[15:2]FnSA field) has to be configured when reading from an Rx FIFO (Rx FIFO Get Index - MCAN_RXFnS[13:8] FnGI). Table 27-7 presents Rx Buffer/Rx FIFO Element Size for different Rx Buffer / Rx FIFO Data Field Size which is configured by way of the MCAN_RXESC register.

Table 27-7. Rx Buffer/Rx FIFO Element Size

MCAN_RXESC Register RBDS/F0DS/F1DS Bits	Data Field [bytes]	FIFO Element Size [RAM words]
000	8	4
001	12	5
010	16	6
011	20	7
100	24	8
101	32	10
110	48	14
111	64	18

ADVANCE INFORMATION

27.1.4.17.1 Rx FIFO Blocking Mode

The Rx FIFO blocking mode is the default operation mode for the Rx FIFOs and is configured by MCAN_RXFnC[31] FnOM = 0.

If an Rx FIFO full condition is reached (MCAN_RXFnS[21:16] FnPI = MCAN_RXFnS[13:8] FnGI), no further messages are written to the corresponding Rx FIFO until at least one message has been read out and the Rx FIFO Get Index has been incremented. An Rx FIFO full condition is signaled by the MCAN_RXFnS[24] FnF = 1 and interrupt flag MCAN_IR.RF0F/MCAN_IR.RF1F is set.

In case a message is received while the corresponding Rx FIFO is full, this message is rejected and the message lost condition is signaled by MCAN_RXFnS[25] RFnL = 1 and interrupt flag MCAN_IR.RF0L/ MCAN_IR.RF1L is set.

27.1.4.17.2 Rx FIFO Overwrite Mode

The Rx FIFO overwrite mode is configured by MCAN_RXFnC[31] FnOM = 1. When an Rx FIFO full condition is reached (MCAN_RXFnS[21:16] FnPI = MCAN_RXFnS[13:8] FnGI) signaled by MCAN_RXFnS[24] FnF = 1, the next accepted message for the FIFO overwrites the oldest FIFO message. Put index/Get index are both incremented by one.

In overwrite mode if an Rx FIFO full condition is signaled, reading of the Rx FIFO elements starts at least at get index + 1. The reason for this is a received message is written to the Message RAM (Put index) while the Host CPU is reading from the Message RAM (Get index). In this case, inconsistent data can be read from the respective Rx FIFO element. The problem is solved by adding an offset to the Get index when reading from the Rx FIFO. The offset depends on how fast the Host CPU accesses the Rx FIFO. Figure 27-19 shows an offset of two with respect to the Get index when reading the Rx FIFO. In this case, the two messages stored in element 1 and 2 are lost.

After reading from the Rx FIFO, the number of the last element read has to be written to the Rx FIFO Acknowledge Index MCAN_RXFnA[5:0] FnAI. This increments the get index to that element number. In case the Put index has not been incremented to this Rx FIFO element, the Rx FIFO full condition is reset (MCAN_RXFnS[24] FnF = 0).

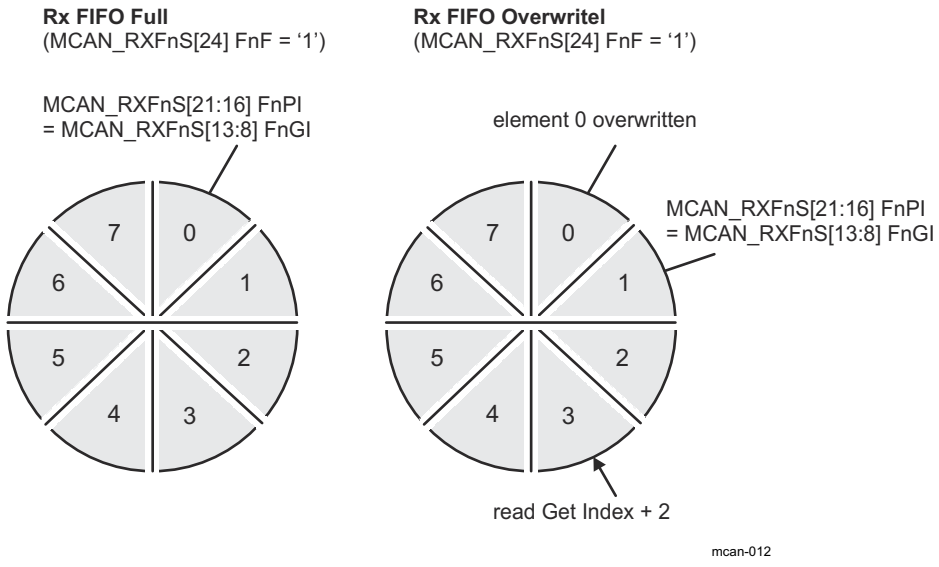


Figure 27-19. Rx FIFO Overflow Handling

27.1.4.18 Dedicated Rx Buffers

The MCAN supports up to 64 dedicated Rx buffers. The start address of the Rx buffers section in the Message RAM is configured by way of the MCAN_RXBC.RBSA field. To store in an Rx Buffer a Standard or Extended Message ID Filter Element with SFEC/EFEC = 111 and SFID2/EFID2[10:9] = 00 has to be configured (see Section 27.1.4.19.5 and Section 27.1.4.19.6).

After a received message has been accepted by a filter element, the message is stored into the Rx Buffer in the Message RAM referenced by the filter element (the format is the same as for an Rx FIFO element). In addition, the flag MCAN_IR.DRX (message stored in Dedicated Rx Buffer) is set.

Table 27-8 shows Example Filter Configuration for Rx buffers.

Table 27-8. Example Filter Configuration for Rx Buffers

Filter Element	SFID1[10:0] EFID1[28:0]	SFID2[10:9] EFID2[10:9]	SFID2[5:0] EFID2[5:0]
0	ID message 1	00	00 0000
1	ID message 2	00	00 0001
2	ID message 3	00	00 0010

After the last word of a matching received message has been written to the Message RAM, the respective New Data flag in register MCAN_NDAT1/MCAN_NDAT2 is set. As long as the New Data flag is set, the respective Rx Buffer is locked against updates from received matching frames. The New Data flags have to be reset by the Host CPU by writing a 1 to the respective bit position.

While an Rx buffer's New Data flag is set, a Message ID Filter Element referencing this specific Rx Buffer will not match, causing the acceptance filtering to continue. Following Message ID Filter Elements can cause the received message to be stored into another Rx Buffer, or into an Rx FIFO, or the message can be rejected, depending on filter configuration.

27.1.4.18.1 Rx Buffer Handling

Rx Buffer Handling include the following steps:

- Reset interrupt flag MCAN_IR.DRX
- Read New Data registers
- Read messages from Message RAM
- Reset New Data flags of processed messages

27.1.4.19 Message RAM

The MCAN module has a Message RAM. The main purpose of the Message RAM is to store:

- Received Messages
- Transmit Messages
- Tx Event Elements
- Message ID Filter Elements

27.1.4.19.1 Message RAM Configuration

The MCAN module can have different Message RAM sizes. The maximum size of the message RAM is 1kB and is offset 0x3FF from the base address. An example of the MCAN module being configured for 1KB size with a width of 32 bits is desribed here.

The Message RAM can include each of the sections listed in Message RAM Configuration. It is not necessary to configure each of the sections (a section in the Message RAM can be 0) and there is no restriction with respect to the sequence of the sections. For parity checking or ECC, a respective number of bits has to be added to each word. When the MCAN module addresses the Message RAM, it addresses 32-bit words. The start addresses are configurable and they are 32-bit word addresses.

The element size can be configured for:

- Rx FIFO 0, by way of the MCAN_RXESC.F0DS field
- Rx FIFO, 1 by way of the MCAN_RXESC.F1DS field
- Rx buffers, by way of the MCAN_RXESC.RBDS field
- Tx buffers, by way of the MCAN_TXESC.TBDS field

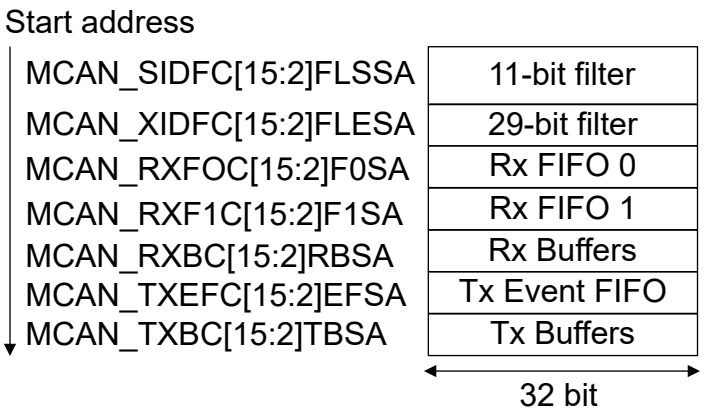


Figure 27-20. Message RAM Configuration

The host CPU configures the following information in the message RAM:

- Start addresses of the memory sections
- Number of elements in each section
- The size of the elements in some sections

Note

The MCAN module does not check for errors in the Message RAM configuration. The configuration of the start addresses of the different sections and the number of elements of each section has to be done carefully. This prevents falsification or loss of data.

An example of the max elements that can be accommodated for each section based on a RAM size of 1kB for the Rx FIFO is provided in Message Transmission configurations table

Table 27-9. Message Transmission configurations

Frame Size, DLC Code (Bytes)	Element Size (Words)	Maximum Elements
8	4	64
12	5	51
16	6	43
20	7	37
24	8	32
32	10	26
48	14	18

Table 27-9. Message Transmission configurations (continued)

Frame Size, DLC Code (Bytes)	Element Size (Words)	Maximum Elements
64	18	14

27.1.4.19.2 Rx Buffer and FIFO Element

Up to 64 Rx buffers and two Rx FIFOs can be configured in the Message RAM. Each Rx FIFO section can be configured to store up to 64 received messages. The element size can be configured for storage of CAN FD messages with up to 64 bytes data field by way of the MCAN_RXESC register.

Figure 27-21 shows the Rx Buffer/Rx FIFO element structure. Table 27-10 shows the Rx Buffer/Rx FIFO element field descriptions.

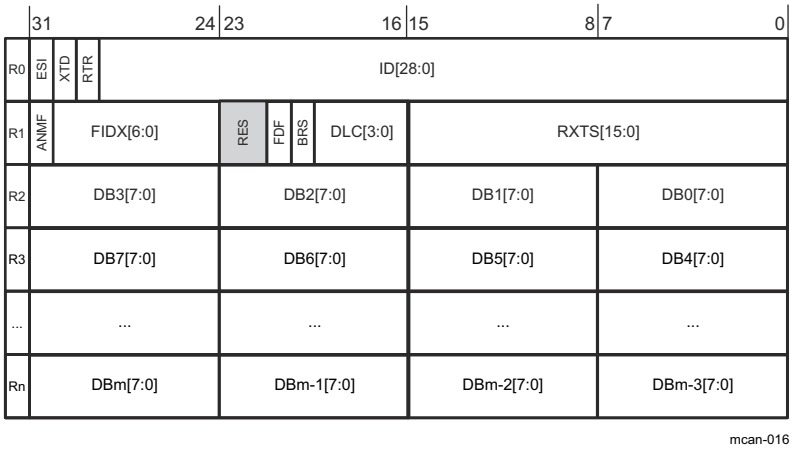


Figure 27-21. Rx Buffer/Rx FIFO Element Structure

Table 27-10. Rx Buffer/Rx FIFO Element Field Descriptions

Word	Bits	Field Name	Description
R0	31	ESI	Error State Indicator • 0x0: Transmitting node is error active • 0x1: Transmitting node is error passive
	30	XTD	Extended Identifier Signals to the Host CPU whether the received frame has a standard or extended identifier. • 0x0: 11-bit standard identifier • 0x1: 29-bit extended identifier
	29	RTR	Remote Transmission Request Signals to the Host CPU whether the received frame is a data frame or a remote frame. • 0x0: Received frame is a data frame • 0x1: Received frame is a remote frame **Note:** There are no remote frames in CAN FD format. In case a CAN FD frame was received (FDF = 1), RTR bit reflects the state of the reserved r1 bit (RES[23]). In CAN FD frames (FDF=1), the dominant RRS (Remote Request Substitution) bit replaces the RTR (Remote Transmission Request) bit.
	28:0	ID[28:0]	Identifier Standard or extended identifier depending on XTD bit. A standard identifier is stored into ID[28:18].

ADVANCE INFORMATION

Table 27-10. Rx Buffer/Rx FIFO Element Field Descriptions (continued)

Word	Bits	Field Name	Description
R1	31	ANMF	Accepted Non-matching Frame Acceptance of non-matching frames can be enabled using the MCAN_GFC.ANFS and MCAN_GFC.ANFE fields. • 0x0: Received frame matching filter index FIDX field • 0x1: Received frame did not match any Rx filter element
	30:24	FIDX[6:0]	Filter Index 0x0-0x7F (0-127): Index of matching Rx acceptance filter element (invalid if ANMF = 1). Range is 0 to MCAN_SIDFC.LSS - 1 respectively MCAN_XIDFC.LSE - 1.
	23:22	RES	Reserved
	21	FDF	FD Format • 0x0: Standard frame format • 0x1: CAN FD frame format (new DLC-coding and CRC)
	20	BRS	Bit Rate Switch • 0x0: Frame received without bit rate switching • 0x1: Frame received with bit rate switching
	19:16	DLC[3:0]	Data Length Code • 0x0-0x8 (0-8): CAN + CAN FD: received frame has 0-8 data bytes • 0x9-0xF (9-15): CAN: received frame has 8 data bytes • 0x9-0xF (9-15): CAN FD: received frame has 12/16/20/24/32/48/64 data bytes
	15:0	RXTS[15:0]	Rx Timestamp Timestamp Counter value captured on start of frame reception. Resolution depending on configuration of the Timestamp Counter Prescaler MCAN_TSCC.TCP.
R2	31:24	DB3[7:0]	Data Byte 3
	23:16	DB2[7:0]	Data Byte 2
	15:8	DB1[7:0]	Data Byte 1
	7:0	DB0[7:0]	Data Byte 0
R3	31:24	DB7[7:0]	Data Byte 7
	23:16	DB6[7:0]	Data Byte 6
	15:8	DB5[7:0]	Data Byte 5
	7:0	DB4[7:0]	Data Byte 4
...	...	...	...
Rn	31:24	DBm[7:0]	Data Byte m
	23:16	DBm-1[7:0]	Data Byte m-1
	15:8	DBm-2[7:0]	Data Byte m-2
	7:0	DBm-3[7:0]	Data Byte m-3

Note: Depending on the configuration of the element size (MCAN_RXESC), between two and sixteen 32-bit words (Rn = 3-17) are used for storage of a CAN message's data field.

ADVANCE INFORMATION

27.1.4.19.3 Tx Buffer Element

The Tx buffers section can be configured to hold dedicated Tx buffers as well as a Tx FIFO/Tx Queue. In case that the Tx buffers section is shared by dedicated Tx buffers and a Tx FIFO/Tx Queue, the dedicated Tx buffers start at the beginning of the Tx buffers section followed by the buffers assigned to the Tx FIFO or Tx Queue. The Tx Handler makes difference between dedicated Tx buffers and Tx FIFO/Tx Queue by way of the MCAN_TXBC.TFQS and MCAN_TXBC.NDTB fields. The element size can be configured for storage of CAN FD messages with up to 64 bytes data field by way of the MCAN_TXESC register.

Figure 27-22 shows the Tx Buffer element structure. Table 27-11 shows the Tx Buffer element field descriptions.

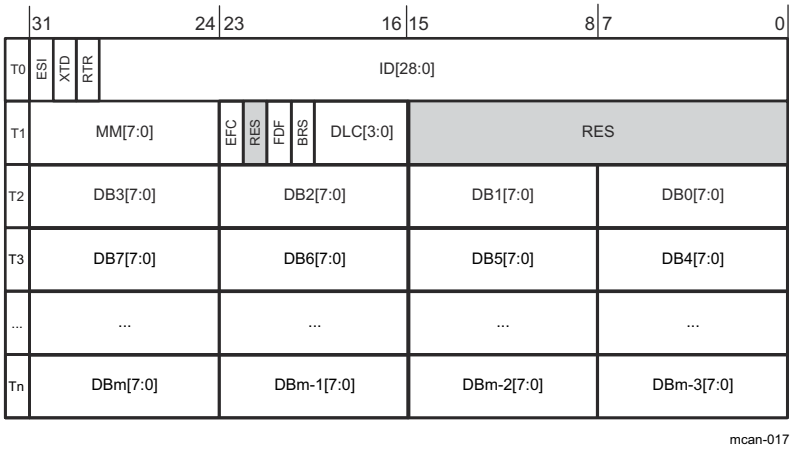


Figure 27-22. Tx Buffer Element Structure

Table 27-11. Tx Buffer Element Field Descriptions

Word	Bits	Field Name	Description
T0	31	ESI	Error State Indicator • 0x0: ESI bit in CAN FD format depends only on error passive flag • 0x1: ESI bit in CAN FD format transmitted recessive **Note:** The ESI bit of the transmit buffer is ORed with the error passive flag to decide the value of the ESI bit in the transmitted CAN FD frame. As required by the CAN FD protocol specification, an error active node can optionally transmit the ESI bit recessive, but an error passive node always transmits the ESI bit recessive.
	30	XTD	Extended Identifier • 0x0: 11-bit standard identifier • 0x1: 29-bit extended identifier
	29	RTR	Remote Transmission Request • 0x0: Transmit data frame • 0x1: Transmit remote frame **Note:** When RTR = 1, the MCAN module transmits a remote frame according to ISO11898-1:2015, even if the MCAN_CCCR.FDOE bit enables the transmission in CAN FD format.
	28:0	ID[28:0]	Identifier Standard or extended identifier depending on XTD bit. A standard identifier has to be written to ID[28:18].

ADVANCE INFORMATION

Table 27-11. Tx Buffer Element Field Descriptions (continued)

Word	Bits	Field Name	Description
T1	31:24	MM[7:0]	Message Marker Written by Host CPU during Tx Buffer configuration. Copied into Tx Event FIFO element for identification of Tx message status (see also MM[7:0] field in Table 27-12).
	23	EFC	Event FIFO Control • 0x0: Don't store Tx events • 0x1: Store Tx events
	22	RES	Reserved
	21	FDF	FD Format • 0x0: Frame transmitted in Classic CAN format • 0x1: Frame transmitted in CAN FD format
	20	BRS	Bit Rate Switch • 0x0: CAN FD frames transmitted without bit rate switching • 0x1: CAN FD frames transmitted with bit rate switching **Note:** ESI, FDF, and BRS bits are only evaluated when CAN FD operation is enabled using the MCAN_CCCR.FDOE bit. BRS bit is only evaluated when MCAN_CCCR.BRSE = 1.
	19:16	DLC[3:0]	Data Length Code • 0x0-0x8 (0-8): CAN + CAN FD: transmit frame has 0-8 data bytes • 0x9-0xF (9-15): CAN: transmit frame has 8 data bytes • 0x9-0xF (9-15): CAN FD: transmit frame has 12/16/20/24/32/48/64 data bytes
	15:0	RES	Reserved
T2	31:24	DB3[7:0]	Data Byte 3
	23:16	DB2[7:0]	Data Byte 2
	15:8	DB1[7:0]	Data Byte 1
	7:0	DB0[7:0]	Data Byte 0
T3	31:24	DB7[7:0]	Data Byte 7
	23:16	DB6[7:0]	Data Byte 6
	15:8	DB5[7:0]	Data Byte 5
	7:0	DB4[7:0]	Data Byte 4
...	...	...	...
Tn	31:24	DBm[7:0]	Data Byte m
	23:16	DBm-1[7:0]	Data Byte m-1
	15:8	DBm-2[7:0]	Data Byte m-2
	7:0	DBm-3[7:0]	Data Byte m-3

Note

Depending on the configuration of the element size (MCAN_TXESC), between two and sixteen 32-bit words (Tn = 3-17) are used for storage of a CAN message's data field.

ADVANCE INFORMATION

27.1.4.19.4 Tx Event FIFO Element

Each element stores information about transmitted messages. By reading the Tx Event FIFO the Host CPU gets this information in the order the messages were transmitted. Status information about the Tx Event FIFO can be obtained from the MCAN_TXEFS register.

Figure 27-23 shows the Tx Event FIFO element structure. Table 27-12 shows the Tx Event FIFO element field descriptions.

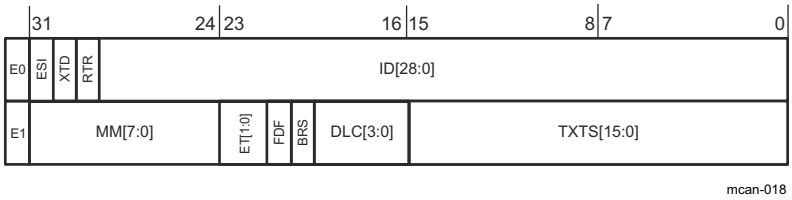


Figure 27-23. Tx Event FIFO Element Structure

Table 27-12. Tx Event FIFO Element Field Descriptions

Word	Bits	Field Name	Description
E0	31	ESI	Error State Indicator • 0x0: Transmitting node is error active • 0x1: Transmitting node is error passive
	30	XTD	Extended Identifier • 0x0: 11-bit standard identifier • 0x1: 29-bit extended identifier
	29	RTR	Remote Transmission Request • 0x0: Data frame transmitted • 0x1: Remote frame transmitted
	28:0	ID[28:0]	Identifier Standard or extended identifier depending on XTD bit. A standard identifier has to be written to ID[28:18].
E1	31:24	MM[7:0]	Message Marker Copied from Tx Buffer into Tx Event FIFO element for identification of Tx message status (see also MM[7:0] field in Table 27-11).
	23:22	ET[1:0]	Event Type • 0x0: Reserved • 0x1: Tx event • 0x2: Transmission in spite of cancellation (always set for transmissions in DAR mode) • 0x3: Reserved
	21	FDF	FD Format • 0x0: Standard frame format • 0x1: CAN FD frame format (new DLC-coding and CRC)

Table 27-12. Tx Event FIFO Element Field Descriptions (continued)

Word	Bits	Field Name	Description
	20	BRS	Bit Rate Switch • 0x0: Frame transmitted without bit rate switching • 0x1: Frame transmitted with bit rate switching
	19:16	DLC[3:0]	Data Length Code • 0x0-0x8 (0-8): CAN + CAN FD: frame with 0-8 data bytes transmitted • 0x9-0xF (9-15): CAN: frame with 8 data bytes transmitted • 0x9-0xF (9-15): CAN FD: frame with 12/16/20/24/32/48/64 data bytes transmitted
	15:0	TXTS[15:0]	Tx Timestamp Timestamp Counter value captured on start of frame transmission. Resolution depending on configuration of the Timestamp Counter Prescaler MCAN_TSCC.TCP filed.

27.1.4.19.5 Standard Message ID Filter Element

Up to 128 filter elements can be configured for 11-bit standard IDs. When accessing a Standard Message ID Filter element, the element address is the Filter List Standard Start Address MCAN_SIDFC.FLSSA field plus the index of the filter element (0-127).

Figure 27-24 shows the Standard Message ID Filter element structure. Table 27-13 shows the Standard Message ID Filter element field descriptions.

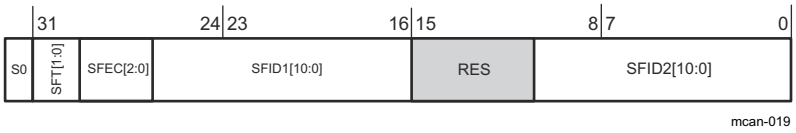


Figure 27-24. Standard Message ID Filter Element Structure

Table 27-13. Standard Message ID Filter Element Field Descriptions

Word	Bits	Field Name	Description
S0	31:30	SFT[1:0]	Standard Filter Type • 0x0: Range filter from SFID1 to SFID2 (SFID2 ≥ SFID1) • 0x1: Dual ID filter for SFID1 or SFID2 • 0x2: Classic filter: SFID1 = filter; SFID2 = mask • 0x3: Filter element disabled **Note:** With SFT = 11 the filter element is disabled and the acceptance filtering continues (same behavior as with SFEC = 000)

ADVANCE INFORMATION

Table 27-13. Standard Message ID Filter Element Field Descriptions (continued)

Word	Bits	Field Name	Description
	29:27	SFEC[2:0]	Standard Filter Element Configuration All enabled filter elements are used for acceptance filtering of standard frames. Acceptance filtering stops at the first matching enabled filter element or when the end of the filter list is reached. If SFEC = 100, 101, or 110 match sets interrupt flag MCAN_IR.HPM and, if enabled, an interrupt is generated. In this case, the MCAN_HPMS register is updated with the status of the priority match. • 0x0: Disable filter element • 0x1: Store in Rx FIFO 0 if filter matches • 0x2: Store in Rx FIFO 1 if filter matches • 0x3: Reject ID if filter matches • 0x4: Set priority if filter matches • 0x5: Set priority and store in FIFO 0 if filter matches • 0x6: Set priority and store in FIFO 1 if filter matches • 0x7: Store into Rx Buffer , configuration of SFT[1:0] ignored
	26:16	SFID1[10:0]	Standard Filter ID 1 When filtering for Rx buffers this field defines the ID of a standard message to be stored. The received identifiers must match exactly, no masking mechanism is used.
	15:11	RES	Reserved
	10:0	SFID2[10:0]	Standard Filter ID 2 This bit field has a different meaning depending on the configuration of SFEC: • SFEC = 001 - 110: Second ID of standard ID filter element • SFEC = 111: Filter for Rx buffers
		SFID2[10:9]	This filed is decides whether the received message is stored into an Rx Buffer or treated as message A, B, or C of the debug message sequence. • 0x0: Store message into an Rx Buffer • 0x1: Debug Message A • 0x2: Debug Message B • 0x3: Debug Message C **Note:** Debug feature is not supported.
		SFID2[8:6]	This field is used to control the filter event pins at the Extension Interface. A one at the respective bit position enables generation of a pulse at the related filter event pin with the duration of one MCAN_ICKL period in case the filter matches. **Note:** Only two filter event pins are supported.
		SFID2[5:0]	This field defines the offset to the Rx Buffer Start Address MCAN_RXBC.RBSA field for storage of a matching message.

27.1.4.19.6 Extended Message ID Filter Element

Up to 64 filter elements can be configured for 29-bit extended IDs. When accessing an Extended Message ID Filter element, the element address is the Filter List Extended Start Address MCAN_XIDFC.FLESA field plus two times the index of the filter element (0-63).

Figure 27-25 shows the Extended Message ID Filter element structure. Table 27-14 shows the Extended Message ID Filter element field descriptions.

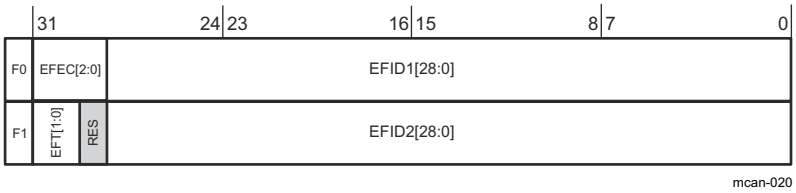


Figure 27-25. Extended Message ID Filter Element Structure

Table 27-14. Extended Message ID Filter Element Field Descriptions

Word	Bits	Field Name	Description
F0	31:29	EFEC[2:0]	Extended Filter Element Configuration All enabled filter elements are used for acceptance filtering of extended frames. Acceptance filtering stops at the first matching enabled filter element or when the end of the filter list is reached. If EFEC = 100, 101, or 110 match sets interrupt flag MCAN_IR.HPM and, if enabled, an interrupt is generated. In this case, the MCAN_HPMS register is updated with the status of the priority match. • 0x0: Disable filter element • 0x1: Store in Rx FIFO 0 if filter matches • 0x2: Store in Rx FIFO 1 if filter matches • 0x3: Reject ID if filter matches • 0x4: Set priority if filter matches • 0x5: Set priority and store in FIFO 0 if filter matches • 0x6: Set priority and store in FIFO 1 if filter matches • 0x7: Store into Rx Buffer or as debug message, configuration of EFT[1:0] ignored
	28:0	EFID1[28:0]	Extended Filter ID 1 First ID of extended ID filter element. When filtering for Rx buffers this field defines the ID of an extended message to be stored. The received identifiers must match exactly, only XIDAM masking mechanism (see Section 27.1.4.16.1.5) is used.
F1	31:30	EFT[1:0]	Extended Filter Type • 0x0: Range filter from EFID1 to EFID2 (EFID2 ≥ EFID1) • 0x1: Dual ID filter for EFID1 or EFID2 • 0x2: Classic filter: EFID1 = filter, EFID2 = mask • 0x3: Range filter from EFID1 to EFID2 (EFID2 ≥ EFID1), XIDAM mask not applied
	29	RES	Reserved

Table 27-14. Extended Message ID Filter Element Field Descriptions (continued)

Word	Bits	Field Name	Description
	28:0	EFID2[28:0]	Extended Filter ID 2 This bit field has a different meaning depending on the configuration of EFEC: • EFEC = 001 - 110: Second ID of extended ID filter element • EFEC = 111: Filter for Rx buffers
		EFID2[10:9]	This field decides whether the received message is stored into an Rx Buffer or treated as message A, B, or C of the debug message sequence. • 0x0: Store message into an Rx Buffer • 0x1: Debug Message A • 0x2: Debug Message B • 0x3: Debug Message C **Note:** Debug feature is not supported.
		EFID2[8:6]	This field is used to control the filter event pins at the Extension Interface. A one at the respective bit position enables generation of a pulse at the related filter event pin with the duration of one MCAN_ICKL period in case the filter matches. **Note:** Only two filter event pins are supported.
		EFID2[5:0]	This field defines the offset to the Rx Buffer Start Address MCAN_RXBC.RBSA field for storage of a matching message.

27.1.5 MCAN Integration

Figure 27-26 shows the integration of the MCAN module in the device.

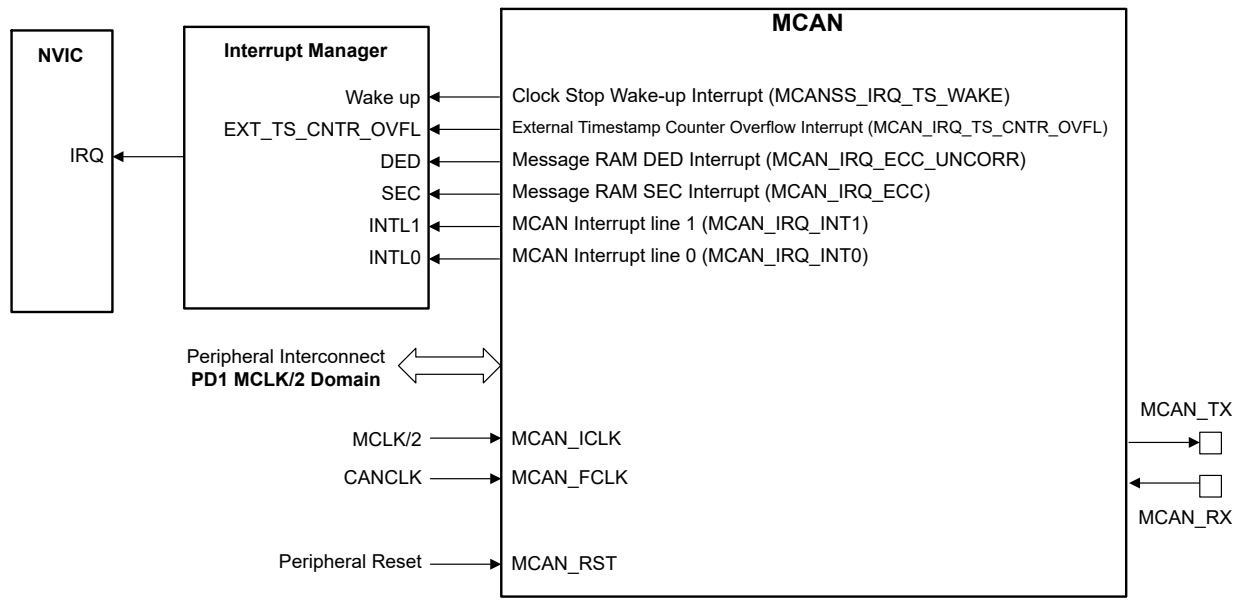


Figure 27-26. MCAN Integration

Table 27-15 summarizes the integration of the MCAN module in the device.

ADVANCE INFORMATION

Table 27-15. MCAN Clocks and Resets

Destination Signal Name	Source Signal Name	Description
Clocks		
MCAN_ICLK	MCLK/2	Interface clock for the MCAN module
MCAN_FCLK	CANCLK	Bit timing clock for MCAN
Resets		
MCAN_RST	Peripheral Reset	Asynchronous reset signal to the MCAN module

Note

For AM13E23x devices, MCLK = 200MHz and the maximum CANCLK speed is 100MHz.

27.1.6 Interrupt and Event Support

The MCAN module contains one event publisher (CPU_INT) that manages MCAN interrupt requests (IRQs) to the CPU subsystem via a static event route.

The MCAN events are summarized in Table 27-16.

Table 27-16. MCAN Events

Event	Type	Source	Destination	Route	Configuration	Functionality
CPU interrupt	Publisher	MCAN	CPU Subsystem	Static route	CPU_INT registers	Fixed interrupt route from MCAN to CPU
MCAN_INT0	Publisher	MCAN	eCAP Mux Index 16	Static route	ECAPSYNCINSEL register	Fixed interrupt route from MCAN to eCAP Input Mux

27.1.6.1 CPU Interrupt Event Publisher (CPU_INT)

The MCAN module provides different interrupt sources which can be configured to source a CPU interrupt event. In order of decreasing interrupt priority, the CPU interrupt events from the MCAN are shown in Table 27-17.

Table 27-17. MCAN Interrupt Event Conditions (CPU_INT)

Index (IIDX)	Name	Description
0x1	MCAN_IRQ_INT0	MCAN interrupt 0
0x2	MCAN_IRQ_INT1	MCAN interrupt 1
0x3	MCAN_IRQ_ECC	MCAN ECC SEC (single error correct) interrupt
0x4	MCAN_IRQ_ECC_UNCORR	MCAN ECC uncorrectable / DED (double error detect) interrupt
0x5	MCANSS_IRQ_TS_CNTR_OVFL	MCAN timestamp counter overflow interrupt
0x6	MCANSS_IRQ_TS_WAKE	MCAN clock stop wakeup interrupt

The CPU interrupt event configuration is managed with the CPU_INT event management registers. Interrupt (RIS) flags are cleared upon software reading the IIDX register or writing to the respective ICLR register bits.

See *Using Event Registers* for guidance on configuring the event registers for CPU interrupts.

27.2 MCAN Registers

This Section describes the MCAN Registers.

27.2.1 MCAN Base Address Table

Table 27-18. MCAN Base Address Table

Bit Field Name		DriverLib Name	Base Address
Instance	Structure		
Mcan0Regs	MCAN_REGS	MCAN0	0x4011_0000

27.2.2 MCAN_REGS Registers

Table 27-19 lists the memory-mapped registers for the MCAN_REGS registers. All register offset addresses not listed in Table 27-19 should be considered as reserved locations and the register contents should not be modified.

Table 27-19. MCAN_REGS Registers

Offset	Acronym	Register Name	Section
6004h	CANRX	CAN RX IO	Go
6008h	CANTX	CAN TX IO	Go
6204h	CANRX	FUPDATE version of CANRX	Go
6208h	CANTX	FUPDATE version of CANTX	Go
6480h	CPU_CONNECT_0	CPU Connect	Go
6800h	PWREN	Power enable	Go
6804h	RSTCTL	Reset Control	Go
6814h	STAT	Status Register	Go
7000h	MCAN_CREL	MCAN Core Release Register	Go
7004h	MCAN_ENDN	MCAN Endian Register	Go
700Ch	MCAN_DBTP	MCAN Data Bit Timing and Prescaler Register	Go
7010h	MCAN_TEST	MCAN Test Register	Go
7014h	MCAN_RWD	MCAN RAM Watchdog	Go
7018h	MCAN_CCCR	MCAN CC Control Register	Go
701Ch	MCAN_NBTP	MCAN Nominal Bit Timing and Prescaler Register	Go
7020h	MCAN_TSCC	MCAN Timestamp Counter Configuration	Go
7024h	MCAN_TSCV	MCAN Timestamp Counter Value	Go
7028h	MCAN_TOCC	MCAN Timeout Counter Configuration	Go
702Ch	MCAN_TOCV	MCAN Timeout Counter Value	Go
7040h	MCAN_ECR	MCAN Error Counter Register	Go
7044h	MCAN_PSR	MCAN Protocol Status Register	Go
7048h	MCAN_TDCR	MCAN Transmitter Delay Compensation Register	Go
7050h	MCAN_IR	MCAN Interrupt Register	Go
7054h	MCAN_IE	MCAN Interrupt Enable	Go
7058h	MCAN_ILS	MCAN Interrupt Line Select	Go
705Ch	MCAN_ILE	MCAN Interrupt Line Enable	Go
7080h	MCAN_GFC	MCAN Global Filter Configuration	Go
7084h	MCAN_SIDFC	MCAN Standard ID Filter Configuration	Go
7088h	MCAN_XIDFC	MCAN Extended ID Filter Configuration	Go
7090h	MCAN_XIDAM	MCAN Extended ID and Mask	Go
7094h	MCAN_HPMS	MCAN High Priority Message Status	Go
7098h	MCAN_NDAT1	MCAN New Data 1	Go
709Ch	MCAN_NDAT2	MCAN New Data 2	Go
70A0h	MCAN_RXF0C	MCAN Rx FIFO 0 Configuration	Go
70A4h	MCAN_RXF0S	MCAN Rx FIFO 0 Status	Go
70A8h	MCAN_RXF0A	MCAN Rx FIFO 0 Acknowledge	Go
70ACh	MCAN_RXBC	MCAN Rx Buffer Configuration	Go
70B0h	MCAN_RXF1C	MCAN Rx FIFO 1 Configuration	Go
70B4h	MCAN_RXF1S	MCAN Rx FIFO 1 Status	Go
70B8h	MCAN_RXF1A	MCAN Rx FIFO 1 Acknowledge	Go

Table 27-19. MCAN_REGS Registers (continued)

Offset	Acronym	Register Name	Section
70BCh	MCAN_RXESC	MCAN Rx Buffer / FIFO Element Size Configuration	Go
70C0h	MCAN_TXBC	MCAN Tx Buffer Configuration	Go
70C4h	MCAN_TXFQS	MCAN Tx FIFO / Queue Status	Go
70C8h	MCAN_TXESC	MCAN Tx Buffer Element Size Configuration	Go
70CCh	MCAN_TXBRP	MCAN Tx Buffer Request Pending	Go
70D0h	MCAN_TXBAR	MCAN Tx Buffer Add Request	Go
70D4h	MCAN_TXBCR	MCAN Tx Buffer Cancellation Request	Go
70D8h	MCAN_TXBTO	MCAN Tx Buffer Transmission Occurred	Go
70DCh	MCAN_TXBCF	MCAN Tx Buffer Cancellation Finished	Go
70E0h	MCAN_TXBTIE	MCAN Tx Buffer Transmission Interrupt Enable	Go
70E4h	MCAN_TXBCIE	MCAN Tx Buffer Cancellation Finished Interrupt Enable	Go
70F0h	MCAN_TXEFC	MCAN Tx Event FIFO Configuration	Go
70F4h	MCAN_TXEFS	MCAN Tx Event FIFO Status	Go
70F8h	MCAN_TXEFA	MCAN Tx Event FIFO Acknowledge	Go
7200h	MCANSS_PID	MCAN Subsystem Revision Register	Go
7204h	MCANSS_CTRL	MCAN Subsystem Control Register	Go
7208h	MCANSS_STAT	MCAN Subsystem Status Register	Go
720Ch	MCANSS_ICS	MCAN Subsystem Interrupt Clear Shadow Register	Go
7210h	MCANSS_IRS	MCAN Subsystem Interrupt Raw Satus Register	Go
7214h	MCANSS_IECS	MCAN Subsystem Interrupt Enable Clear Shadow Register	Go
7218h	MCANSS_IE	MCAN Subsystem Interrupt Enable Register	Go
721Ch	MCANSS_IES	MCAN Subsystem Interrupt Enable Status	Go
7220h	MCANSS_EOI	MCAN Subsystem End of Interrupt	Go
7224h	MCANSS_EXT_TS_PRESCALER	MCAN Subsystem External Timestamp Prescaler 0	Go
7228h	MCANSS_EXT_TS_UNSERVICED_INTR_CNTR	MCAN Subsystem External Timestamp Unserviced Interrupts Counter	Go
7400h	MCANERR_REV	MCAN Error Aggregator Revision Register	Go
7408h	MCANERR_VECTOR	MCAN ECC Vector Register	Go
740Ch	MCANERR_STAT	MCAN Error Misc Status	Go
7410h	MCANERR_WRAP_REV	MCAN ECC Wrapper Revision Register	Go
7414h	MCANERR_CTRL	MCAN ECC Control	Go
7418h	MCANERR_ERR_CTRL1	MCAN ECC Error Control 1 Register	Go
741Ch	MCANERR_ERR_CTRL2	MCAN ECC Error Control 2 Register	Go
7420h	MCANERR_ERR_STAT1	MCAN ECC Error Status 1 Register	Go
7424h	MCANERR_ERR_STAT2	MCAN ECC Error Status 2 Register	Go
7428h	MCANERR_ERR_STAT3	MCAN ECC Error Status 3 Register	Go
743Ch	MCANERR_SEC_EOI	MCAN Single Error Corrected End of Interrupt Register	Go
7440h	MCANERR_SEC_STATUS	MCAN Single Error Corrected Interrupt Status Register	Go
7480h	MCANERR_SEC_ENABLE_SET	MCAN Single Error Corrected Interrupt Enable Set Register	Go
74C0h	MCANERR_SEC_ENABLE_CLR	MCAN Single Error Corrected Interrupt Enable Clear Register	Go
753Ch	MCANERR_DED_EOI	MCAN Double Error Detected End of Interrupt Register	Go
7540h	MCANERR_DED_STATUS	MCAN Double Error Detected Interrupt Status Register	Go
7580h	MCANERR_DED_ENABLE_SET	MCAN Double Error Detected Interrupt Enable Set Register	Go

Table 27-19. MCAN_REGS Registers (continued)

Offset	Acronym	Register Name	Section
75C0h	MCANERR_DED_ENABLE_CLR	MCAN Double Error Detected Interrupt Enable Clear Register	Go
7600h	MCANERR_AGGR_ENABLE_SET	MCAN Error Aggregator Enable Set Register	Go
7604h	MCANERR_AGGR_ENABLE_CLR	MCAN Error Aggregator Enable Clear Register	Go
7608h	MCANERR_AGGR_STATUS_SET	MCAN Error Aggregator Status Set Register	Go
760Ch	MCANERR_AGGR_STATUS_CLR	MCAN Error Aggregator Status Clear Register	Go
7820h	IIDX	Interrupt Index Register	Go
7828h	IMASK	Interrupt mask	Go
7830h	RIS	Raw interrupt status	Go
7838h	MIS	Masked interrupt status	Go
7840h	ISET	Interrupt set	Go
7848h	ICLR	Interrupt clear	Go
78E0h	EVT_MODE	Event Mode	Go
78FCh	DESC	Module Description	Go
7900h	MCANSS_CLKEN	MCAN module clock enable	Go
7904h	MCANSS_CLKDIV	Clock divider	Go
7908h	MCANSS_CLKCTL	MCAN-SS clock stop control register	Go
790Ch	MCANSS_CLKSTS	MCANSS clock stop status register	Go

Complex bit access types are encoded to fit into small table cells. Table 27-20 shows the codes that are used for access types in this section.

Table 27-20. MCAN_REGS Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
R-0	R -0	Read Returns 0s
RC	R C	Read to Clear
RS	R S	Read to Set
Write Type		
W	W	Write
W1C	W 1C	Write 1 to clear
W1S	W 1S	Write 1 to set
W1SQ	W 1S Q	Write 1 to set Qualified. A condition must be met for this operation to occur.
WD	W D	Write Decrement. Decrements the specified bit field by the amount written.
WI	W I	Write Increment. Increments the specified bit field by the amount written.

ADVANCE INFORMATION

Table 27-20. MCAN_REGS Access Type Codes (continued)

Access Type	Code	Description
WK	W K	Write Write protected by a key
WQ	W Q	Write Qualified. A condition must be met for this operation to occur.
Reset or Default Value		
-n		Value after reset or the default value

1 CANRX Register (Offset = 6004h) [Reset = 00000000h]

CANRX is shown in Figure 27-27 and described in Table 27-21.

Return to the Summary Table.

CAN RX IO

Figure 27-27. CANRX Register

31	30	29	28	27	26	25	24
RESERVED	GFLT	SLEW	WCOMP	WUEN	INV	HIGHZ1	HIGHZ0
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
RESERVED	DRV			HYSTEN	INENA	PIPU	PIPD
R/W-0h	R/W-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
GSTATE		RESERVED					
R/W-0h		R/W-0h					
7	6	5	4	3	2	1	0
PSTATE		RESERVED					
R/W-0h		R/W-0h					

Table 27-22. CANRX Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	
30	GFLT	R/W	0h	Glitch Filter Enable 0h = No internal glitch filter 1h = Use internal glitch filter
29	SLEW	R/W	0h	Reserved Slew Rate Control 0h = No Slew Rate Control 1h = Use Slew Rate Control
28	WCOMP	R/W	0h	Wake up compare value 0h = Match 0 will wake 1h = Match 1 will wake
27	WUEN	R/W	0h	Wake up enable 0h = Wake up not enabled 1h = Wake up enabled
26	INV	R/W	0h	Invert digital input/output relative to peripheral/GPIO 0h = Input and output are non-inverted 1h = Input and output are inverted
25	HIGHZ1	R/W	0h	High-Z instead of high output 0h = Pin can be driven high 1h = Pin is tri-stated instead of driven high
24	HIGHZ0	R/W	0h	High-Z instead of low output 0h = Pin can be driven low 1h = Pin is tri-stated instead of driven low
23	RESERVED	R/W	0h	
22-20	DRV	R/W	0h	Drive strength options 0h = Lowest drive strength 1h = Drive strength 2/8 2h = Drive strength 3/8 3h = Drive strength 4/8 4h = Drive strength 5/8 5h = Drive strength 6/8 6h = Drive strength 7/8 7h = Highest drive strength

Table 27-22. CANRX Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
19	HYSTEN	R/W	0h	Hysteresis enable 0h = No hysteresis 1h = Hysteresis on
18	INENA	R/W	0h	Input enable 0h = Inputs 0 to connected core 1h = Inputs IO pad value to connected core
17	PIPU	R/W	0h	Pull up enable 0h = No pull up 1h = Pull up
16	PIPD	R/W	0h	Pull down enable 0h = No pull down 1h = Pull down
15-14	GSTATE	R/W	0h	GPIO Channel State 0h = G-Channel is in Unassigned State 1h = G-Channel is in Handover State 2h = G-Channel is in Connected State and not Locked (That is F field is allowed to change without going back through Unassigned state) 3h = G-Channel is in Connected State and Locked (That is F field is not allowed to change to a different non-Zero value until both G and P channels go to Unassigned)
13-8	RESERVED	R/W	0h	
7-6	PSTATE	R/W	0h	Peripheral-Analog Channel State 0h = P-Channel is in Unassigned State 1h = P-Channel is in Handover State 2h = P-Channel is in Connected State and not Locked (That is F field is allowed to change without going back through Unassigned state) 3h = P-Channel is in Connected State and Locked (That is F field is not allowed to change to a different non-Zero value until both G and P channels go to Unassigned)
5-0	RESERVED	R/W	0h	

2 CANTX Register (Offset = 6008h) [Reset = 00000000h]

CANTX is shown in Figure 27-28 and described in Table 27-22.

Return to the Summary Table.

CAN TX IO

Figure 27-28. CANTX Register

31	30	29	28	27	26	25	24
RESERVED	GFLT	SLEW	WCOMP	WUEN	INV	HIGHZ1	HIGHZ0
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
RESERVED	DRV			HYSTEN	INENA	PIPU	PIPD
R/W-0h	R/W-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
GSTATE		RESERVED					
R/W-0h		R/W-0h					
7	6	5	4	3	2	1	0
PSTATE		RESERVED					
R/W-0h		R/W-0h					

Table 27-24. CANTX Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	
30	GFLT	R/W	0h	Glitch Filter Enable 0h = No internal glitch filter 1h = Use internal glitch filter
29	SLEW	R/W	0h	Reserved Slew Rate Control 0h = No Slew Rate Control 1h = Use Slew Rate Control
28	WCOMP	R/W	0h	Wake up compare value 0h = Match 0 will wake 1h = Match 1 will wake
27	WUEN	R/W	0h	Wake up enable 0h = Wake up not enabled 1h = Wake up enabled
26	INV	R/W	0h	Invert digital input/output relative to peripheral/GPIO 0h = Input and output are non-inverted 1h = Input and output are inverted
25	HIGHZ1	R/W	0h	High-Z instead of high output 0h = Pin can be driven high 1h = Pin is tri-stated instead of driven high
24	HIGHZ0	R/W	0h	High-Z instead of low output 0h = Pin can be driven low 1h = Pin is tri-stated instead of driven low
23	RESERVED	R/W	0h	
22-20	DRV	R/W	0h	Drive strength options 0h = Lowest drive strength 1h = Drive strength 2/8 2h = Drive strength 3/8 3h = Drive strength 4/8 4h = Drive strength 5/8 5h = Drive strength 6/8 6h = Drive strength 7/8 7h = Highest drive strength

Table 27-24. CANTX Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
19	HYSTEN	R/W	0h	Hysteresis enable 0h = No hysteresis 1h = Hysteresis on
18	INENA	R/W	0h	Input enable 0h = Inputs 0 to connected core 1h = Inputs IO pad value to connected core
17	PIPU	R/W	0h	Pull up enable 0h = No pull up 1h = Pull up
16	PIPD	R/W	0h	Pull down enable 0h = No pull down 1h = Pull down
15-14	GSTATE	R/W	0h	GPIO Channel State 0h = G-Channel is in Unassigned State 1h = G-Channel is in Handover State 2h = G-Channel is in Connected State and not Locked (That is F field is allowed to change without going back through Unassigned state) 3h = G-Channel is in Connected State and Locked (That is F field is not allowed to change to a different non-Zero value until both G and P channels go to Unassigned)
13-8	RESERVED	R/W	0h	
7-6	PSTATE	R/W	0h	Peripheral-Analog Channel State 0h = P-Channel is in Unassigned State 1h = P-Channel is in Handover State 2h = P-Channel is in Connected State and not Locked (That is F field is allowed to change without going back through Unassigned state) 3h = P-Channel is in Connected State and Locked (That is F field is not allowed to change to a different non-Zero value until both G and P channels go to Unassigned)
5-0	RESERVED	R/W	0h	

3 CANRX Register (Offset = 6204h) [Reset = 00000000h]

CANRX is shown in Figure 27-29 and described in Table 27-23.

Return to the Summary Table.

FUPDATE version of CANRX

Figure 27-29. CANRX Register

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

Table 27-26. CANRX Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	RESERVED	R/W	0h	
27-2	IOADDR	W	0h	IO Address. This is the address that corresponds to the SOC address[27:2] of the module IP instance specific IO signal in the "Full Write" subregion of the pinmux subregion.
1	LOCK	W	0h	Sets lock bit 0h = Writing this value has no effect 1h = Set channel lock bit
0	GSEL	W	0h	GPIO channel Select 0: Select the P-Channel for the F update 1: Select the G-Channel for the F update 0h = Select the P-Channel for the F update 1h = Select the G-Channel for the F update

4 CANTX Register (Offset = 6208h) [Reset = 00000000h]

CANTX is shown in Figure 27-30 and described in Table 27-24.

Return to the Summary Table.

FUPDATE version of CANTX

Figure 27-30. CANTX Register

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

Table 27-28. CANTX Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	RESERVED	R/W	0h	
27-2	IOADDR	W	0h	IO Address. This is the address that corresponds to the SOC address[27:2] of the module IP instance specific IO signal in the "Full Write" subregion of the pinmux subregion.
1	LOCK	W	0h	Sets lock bit 0h = Writing this value has no effect 1h = Set channel lock bit
0	GSEL	W	0h	GPIO channel Select 0: Select the P-Channel for the F update 1: Select the G-Channel for the F update 0h = Select the P-Channel for the F update 1h = Select the G-Channel for the F update

5 CPU_CONNECT_0 Register (Offset = 6480h) [Reset = 00000000h]

CPU_CONNECT_0 is shown in Figure 27-31 and described in Table 27-25.

Return to the Summary Table.

Directly connect peripheral publisher port to application processor

Figure 27-31. CPU_CONNECT_0 Register

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

Table 27-30. CPU_CONNECT_0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R/W	0h	
1	RESERVED	R/W	0h	Reserved
0	RESERVED	R/W	0h	

6 PWREN Register (Offset = 6800h) [Reset = 00000000h]

PWREN is shown in Figure 27-32 and described in Table 27-26.

Return to the Summary Table.

Register to control the power state

Figure 27-32. PWREN Register

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

Table 27-32. PWREN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	KEY	W	0h	KEY to allow Power State Change 26h = KEY to allow write access to this register
23-1	RESERVED	R/W	0h	
0	ENABLE	R/WK	0h	Enable the power [EXT_GPRCM.PWREN.KEY] must be set to 26h to write to this bit. 0h = Disable Power 1h = Enable Power

ADVANCE INFORMATION

7 RSTCTL Register (Offset = 6804h) [Reset = 00000000h]

RSTCTL is shown in Figure 27-33 and described in Table 27-27.

Return to the Summary Table.

Register to control reset assertion and de-assertion

Figure 27-33. RSTCTL Register

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

Table 27-34. RSTCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	KEY	W	0h	Unlock key B1h = KEY to allow write access to this register
23-2	RESERVED	W	0h	
1	RESETSTKYCLR	WK	0h	Clear the RESETSTKY bit in the STAT register [EXT_GPRCM.RSTCTL.KEY] must be set to B1h to write to this bit. 0h = Writing 0 has no effect 1h = Clear reset sticky bit
0	RESETASSERT	WK	0h	Assert reset to the peripheral [EXT_GPRCM.RSTCTL.KEY] must be set to B1h to write to this bit. 0h = Writing 0 has no effect 1h = Assert reset

ADVANCE INFORMATION

8 STAT Register (Offset = 6814h) [Reset = 00000000h]

STAT is shown in Figure 27-34 and described in Table 27-28.

Return to the Summary Table.

peripheral enable and reset status

Figure 27-34. STAT Register

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

Table 27-36. STAT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-17	RESERVED	R	0h	
16	RESETSTKY	R	0h	This bit indicates, if the peripheral was reset, since this bit was cleared by RESETSTKYCLR in the RSTCTL register 0h = The peripheral has not been reset since this bit was last cleared by RESETSTKYCLR in the RSTCTL register 1h = The peripheral was reset since the last bit clear
15-0	RESERVED	R	0h	

9 MCAN_CREL Register (Offset = 7000h) [Reset = 32380608h]

MCAN_CREL is shown in Figure 27-35 and described in Table 27-29.

Return to the Summary Table.

MCAN Core Release Register

Figure 27-35. MCAN_CREL Register

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

Table 27-38. MCAN_CREL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	REL	R	3h	Core Release. One digit, BCD-coded.
27-24	STEP	R	2h	Step of Core Release. One digit, BCD-coded.
23-20	SUBSTEP	R	3h	Sub-Step of Core Release. One digit, BCD-coded.
19-16	YEAR	R	8h	Time Stamp Year. One digit, BCD-coded.
15-8	MON	R	6h	Time Stamp Month. Two digits, BCD-coded.
7-0	DAY	R	8h	Time Stamp Day. Two digits, BCD-coded.

10 MCAN_ENDN Register (Offset = 7004h) [Reset = 87654321h]

MCAN_ENDN is shown in Figure 27-36 and described in Table 27-30.

Return to the Summary Table.

MCAN Endian Register

Figure 27-36. MCAN_ENDN Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ETV																															
R-87654321h																															

Table 27-40. MCAN_ENDN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ETV	R	87654321h	Endianess Test Value. Reading the constant value maintained in this register allows software to determine the endianess of the host CPU.

ADVANCE INFORMATION

11 MCAN_DBTP Register (Offset = 700Ch) [Reset = 00000A33h]

MCAN_DBTP is shown in Figure 27-37 and described in Table 27-31.

Return to the Summary Table.

This register is only writable if bits CCCR.CCE and CCCR.INIT are set. The CAN bit time may be programed in the range of 4 to 49 time quanta. The CAN time quantum may be programmed in the range of 1 to 32 m_can_cclk periods. tq = (DBRP + 1) mtq.
DTSEG1 is the sum of Prop_Seg and Phase_Seg1. DTSEG2 is Phase_Seg2.
Therefore the length of the bit time is (programmed values) (DTSEG1 + DTSEG2 + 3) tq or (functional values) (Sync_Seg + Prop_Seg + Phase_Seg1 + Phase_Seg2) tq.
The Information Processing Time (IPT) is zero, meaning the data for the next bit is available at the first clock edge after the sample point.

Figure 27-37. MCAN_DBTP Register

31	30	29	28	27	26	25	24
RESERVED							
R/W-0h							
23	22	21	20	19	18	17	16
TDC	RESERVED			DBRP			
R/WQ-0h	R/W-0h			R/WQ-0h			
15	14	13	12	11	10	9	8
RESERVED			DTSEG1				
R/W-0h			R/WQ-Ah				
7	6	5	4	3	2	1	0
DTSEG2				DSJW			
R/WQ-3h				R/WQ-3h			

Table 27-42. MCAN_DBTP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	RESERVED	R/W	0h	
23	TDC	R/WQ	0h	Transmitter Delay Compensation 0 Transmitter Delay Compensation disabled 1 Transmitter Delay Compensation enabled
22-21	RESERVED	R/W	0h	
20-16	DBRP	R/WQ	0h	Data Bit Rate Prescaler. The value by which the oscillator frequency is divided for generating the bit time quanta. The bit time is built up from a multiple of this quanta. Valid values for the Bit Rate Prescaler are 0 to 31. The actual interpretation by the hardware of this value is such that one more than the value programmed here is used. Qualified Write is possible only with CCCR.CCE='1' and CCCR.INIT='1'.
15-13	RESERVED	R/W	0h	
12-8	DTSEG1	R/WQ	Ah	Data Time Segment Before Sample Point. Valid values are 0 to 31. The actual interpretation by the hardware of this value is such that one more than the programmed value is used. Qualified Write is possible only with CCCR.CCE='1' and CCCR.INIT='1'.
7-4	DTSEG2	R/WQ	3h	Data Time Segment After Sample Point. Valid values are 0 to 15. The actual interpretation by the hardware of this value is such that one more than the programmed value is used. Qualified Write is possible only with CCCR.CCE='1' and CCCR.INIT='1'.

Table 27-42. MCAN_DBTP Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3-0	DSJW	R/WQ	3h	Data Resynchronization Jump Width. Valid values are 0 to 15. The actual interpretation by the hardware of this value is such that one more than the value programmed here is used. Qualified Write is possible only with CCCR.CCE='1' and CCCR.INIT='1'.

12 MCAN_TEST Register (Offset = 7010h) [Reset = 000000X0h]

MCAN_TEST is shown in Figure 27-38 and described in Table 27-32.

Return to the Summary Table.

Write access to the Test Register has to be enabled by setting bit CCCR.TEST to '1'. All Test Register functions are set to their reset values when bit CCCR.TEST is reset.
Loop Back Mode and software control of the internal CAN TX pin are hardware test modes.

Figure 27-38. MCAN_TEST Register

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

Table 27-44. MCAN_TEST Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R/W	0h	
7	RX	R	Xh	Receive Pin. Monitors the actual value of the CAN receive pin. 0h = DOMINANT : The CAN bus is dominant (CAN RX pin = '0') 1h = RECESSIVE : The CAN bus is recessive (CAN RX pin = '1')
6-5	TX	R/WQ	0h	Control of Transmit Pin 00 CAN TX pin controlled by the CAN Core, updated at the end of the CAN bit time 01 Sample Point can be monitored at CAN TX pin 10 Dominant ('0') level at CAN TX pin 11 Recessive ('1') at CAN TX pin Qualified Write is possible only with CCCR.CCE='1' and CCCR.INIT='1'.
4	LBCK	R/WQ	0h	Loop Back Mode. Qualified Write is possible only with CCCR.CCE='1' and CCCR.INIT='1'. 0h = DISABLE : Reset value, Loop Back Mode is disabled 1h = ENABLE : Loop Back Mode is enabled
3-0	RESERVED	R/W	0h	

13 MCAN_RWD Register (Offset = 7014h) [Reset = 00000000h]

MCAN_RWD is shown in Figure 27-39 and described in Table 27-33.

Return to the Summary Table.

MCAN RAM Watchdog

Figure 27-39. MCAN_RWD Register

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

Table 27-46. MCAN_RWD Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R/W	0h	
15-8	WDV	R	0h	Watchdog Value. Acutal Message RAM Watchdog Counter Value. The RAM Watchdog monitors the READY output of the Message RAM. A Message RAM access via the MCAN's Generic Master Interface starts the Message RAM Watchdog Counter with the value configured by the WDC field. The counter is reloaded with WDC when the Message RAM signals successful completion by activating its READY output. In case there is no response from the Message RAM until the counter has counted down to zero, the counter stops and interrupt flag MCAN_IR.WDI is set. The RAM Watchdog Counter is clocked by the host (system) clock.
7-0	WDC	R/WQ	0h	Watchdog Configuration. Start value of the Message RAM Watchdog Counter. With the reset value of "00" the counter is disabled. Qualified Write is possible only with CCCR.CCE='1' and CCCR.INIT='1'.

14 MCAN_CCCR Register (Offset = 7018h) [Reset = 00000001h]

MCAN_CCCR is shown in Figure 27-40 and described in Table 27-34.

Return to the Summary Table.

MCAN CC Control Register

Figure 27-40. MCAN_CCCR Register

31	30	29	28	27	26	25	24
RESERVED							
R/W-0h							
23	22	21	20	19	18	17	16
RESERVED							
R/W-0h							
15	14	13	12	11	10	9	8
NISO	TXP	EFBI	PXHD	RESERVED		BRSE	FDOE
R/WQ-0h	R/WQ-0h	R/WQ-0h	R/WQ-0h	R/W-0h		R/WQ-0h	R/WQ-0h
7	6	5	4	3	2	1	0
TEST	DAR	MON	CSR	CSA	ASM	CCE	INIT
R/W1SQ-0h	R/WQ-0h	R/W1SQ-0h	R/W-0h	R-0h	R/W1SQ-0h	R/WQ-0h	R/W-1h

Table 27-48. MCAN_CCCR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R/W	0h	
15	NISO	R/WQ	0h	Non ISO Operation. If this bit is set, the MCAN uses the CAN FD frame format as specified by the Bosch CAN FD Specification V1.0. 0 CAN FD frame format according to ISO 11898-1:2015 1 CAN FD frame format according to Bosch CAN FD Specification V1.0
14	TXP	R/WQ	0h	Transmit Pause. If this bit is set, the MCAN pauses for two CAN bit times before starting the next transmission after itself has successfully transmitted a frame. 0 Transmit pause disabled 1 Transmit pause enabled Qualified Write is possible only with CCCR.CCE='1' and CCCR.INIT='1'.
13	EFBI	R/WQ	0h	Edge Filtering during Bus Integration 0 Edge filtering disabled 1 Two consecutive dominant tq required to detect an edge for hard synchronization Qualified Write is possible only with CCCR.CCE='1' and CCCR.INIT='1'.
12	PXHD	R/WQ	0h	Protocol Exception Handling Disable 0 Protocol exception handling enabled 1 Protocol exception handling disabled Note: When protocol exception handling is disabled, the MCAN will transmit an error frame when it detects a protocol exception condition. Qualified Write is possible only with CCCR.CCE='1' and CCCR.INIT='1'.
11-10	RESERVED	R/W	0h	

Table 27-48. MCAN_CCCR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
9	BRSE	R/WQ	0h	Bit Rate Switch Enable 0 Bit rate switching for transmissions disabled 1 Bit rate switching for transmissions enabled Note: When CAN FD operation is disabled FDOE = '0', BRSE is not evaluated. Qualified Write is possible only with CCCR.CCE='1' and CCCR.INIT='1'.
8	FDOE	R/WQ	0h	Flexible Datarate Operation Enable 0 FD operation disabled 1 FD operation enabled Qualified Write is possible only with CCCR.CCE='1' and CCCR.INIT='1'.
7	TEST	R/W1SQ	0h	Test Mode Enable 0 Normal operation, register TEST holds reset values 1 Test Mode, write access to register TEST enabled Qualified Write 1 to Set is possible only with CCCR.CCE='1' and CCCR.INIT='1'.
6	DAR	R/WQ	0h	Disable Automatic Retransmission 0 Automatic retransmission of messages not transmitted successfully enabled 1 Automatic retransmission disabled Qualified Write is possible only with CCCR.CCE='1' and CCCR.INIT='1'.
5	MON	R/W1SQ	0h	Bus Monitoring Mode. Bit MON can only be set by SW when both CCE and INIT are set to '1'. The bit can be reset by SW at any time. 0 Bus Monitoring Mode is disabled 1 Bus Monitoring Mode is enabled Qualified Write 1 to Set is possible only with CCCR.CCE='1' and CCCR.INIT='1'.
4	CSR	R/W	0h	Clock Stop Request 0 No clock stop is requested 1 Clock stop requested. When clock stop is requested, first INIT and then CSA will be set after all pending transfer requests have been completed and the CAN bus reached idle.
3	CSA	R	0h	Clock Stop Acknowledge 0 No clock stop acknowledged 1 MCAN may be set in power down by stopping the Host and CAN clocks
2	ASM	R/W1SQ	0h	Restricted Operation Mode. Bit ASM can only be set by SW when both CCE and INIT are set to '1'. The bit can be reset by SW at any time. 0 Normal CAN operation 1 Restricted Operation Mode active Qualified Write 1 to Set is possible only with CCCR.CCE='1' and CCCR.INIT='1'.
1	CCE	R/WQ	0h	Configuration Change Enable 0 The CPU has no write access to the protected configuration registers 1 The CPU has write access to the protected configuration registers (while CCCR.INIT = '1') Qualified Write is possible only with CCCR.CCE='1' and CCCR.INIT='1'.
0	INIT	R/W	1h	Initialization 0 Normal Operation 1 Initialization is started Note: Due to the synchronization mechanism between the two clock domains, there may be a delay until the value written to INIT can be read back. Therefore the programmer has to assure that the previous value written to INIT has been accepted by reading INIT before setting INIT to a new value.

15 MCAN_NBTP Register (Offset = 701Ch) [Reset = 06000A03h]

MCAN_NBTP is shown in Figure 27-41 and described in Table 27-35.

Return to the Summary Table.

This register is only writable if bits CCCR.CCE and CCCR.INIT are set. The CAN bit time may be programed in the range of 4 to 385 time quanta. The CAN time quantum may be programmed in the range of 1 to 512 m_can_cclk periods. tq = (NBRP + 1) mtq.
NTSEG1 is the sum of Prop_Seg and Phase_Seg1. NTSEG2 is Phase_Seg2.
Therefore the length of the bit time is (programmed values) (NTSEG1 + NTSEG2 + 3) tq or (functional values) (Sync_Seg + Prop_Seg + Phase_Seg1 + Phase_Seg2) tq.
The Information Processing Time (IPT) is zero, meaning the data for the next bit is available at the first clock edge after the sample point.
Note: With a CAN clock of 8 MHz, the reset value of 0x06000A03 configures the MCAN for a bit rate of 500 kBit/s.

Figure 27-41. MCAN_NBTP Register

31	30	29	28	27	26	25	24
NSJW							NBRP
R/WQ-3h							R/WQ-0h
23	22	21	20	19	18	17	16
NBRP							
R/WQ-0h							
15	14	13	12	11	10	9	8
NTSEG1							
R/WQ-Ah							
7	6	5	4	3	2	1	0
RESERVED	NTSEG2						
R/W-0h	R/WQ-3h						

Table 27-50. MCAN_NBTP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-25	NSJW	R/WQ	3h	Nominal (Re)Synchronization Jump Width. Valid values are 0 to 127. The actual interpretation by the hardware of this value is such that one more than the value programmed here is used. Qualified Write is possible only with CCCR.CCE='1' and CCCR.INIT='1'.
24-16	NBRP	R/WQ	0h	Nominal Bit Rate Prescaler. The value by which the oscillator frequency is divided for generating the bit time quanta. The bit time is built up from a multiple of this quanta. Valid values for the Bit Rate Prescaler are 0 to 511. The actual interpretation by the hardware of this value is such that one more than the value programmed here is used. Qualified Write is possible only with CCCR.CCE='1' and CCCR.INIT='1'.
15-8	NTSEG1	R/WQ	Ah	Nominal Time Segment Before Sample Point. Valid values are 1 to 255. The actual interpretation by the hardware of this value is such that one more than the programmed value is used. Qualified Write is possible only with CCCR.CCE='1' and CCCR.INIT='1'.
7	RESERVED	R/W	0h	

ADVANCE INFORMATION

Table 27-50. MCAN_NBTP Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
6-0	NTSEG2	R/WQ	3h	Nominal Time Segment After Sample Point. Valid values are 1 to 127. The actual interpretation by the hardware of this value is such that one more than the programmed value is used. Qualified Write is possible only with CCCR.CCE='1' and CCCR.INIT='1'.

16 MCAN_TSCC Register (Offset = 7020h) [Reset = 00000000h]

MCAN_TSCC is shown in Figure 27-42 and described in Table 27-36.

Return to the Summary Table.

MCAN Timestamp Counter Configuration

Figure 27-42. MCAN_TSCC Register

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

Table 27-52. MCAN_TSCC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-20	RESERVED	R/W	0h	
19-16	TCP	R/WQ	0h	Timestamp Counter Prescaler. Configures the timestamp and timeout counters time unit in multiples of CAN bit times. Valid values are 0 to 15. The actual interpretation by the hardware of this value is such that one more than the value programmed here is used. Note: With CAN FD an external counter is required for timestamp generation (TSS = "10"). Qualified Write is possible only with CCCR.CCE='1' and CCCR.INIT='1'.
15-2	RESERVED	R/W	0h	
1-0	TSS	R/WQ	0h	Timestamp Select 00 Timestamp counter value always 0x0000 01 Timestamp counter value incremented according to TCP 10 External timestamp counter value used 11 Same as "00" Qualified Write is possible only with CCCR.CCE='1' and CCCR.INIT='1'.

17 MCAN_TSCV Register (Offset = 7024h) [Reset = 00000000h]

MCAN_TSCV is shown in Figure 27-43 and described in Table 27-37.

Return to the Summary Table.

MCAN Timestamp Counter Value

Figure 27-43. MCAN_TSCV Register

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

Table 27-54. MCAN_TSCV Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R/W	0h	
15-0	TSC	R/W	0h	Timestamp Counter. The internal/external Timestamp Counter value is captured on start of frame (both Rx and Tx). When TSCC.TSS = "01", the Timestamp Counter is incremented in multiples of CAN bit times, (1...16), depending on the configuration of TSCC.TCP. A wrap around sets interrupt flag IR.TSW. Write access resets the counter to zero. When TSCC.TSS = "10", TSC reflects the External Timestamp Counter value, and a write access has no impact. Note: A "wrap around" is a change of the Timestamp Counter value from non-zero to zero not caused by write access to MCAN_TSCV.

18 MCAN_TOCC Register (Offset = 7028h) [Reset = FFFF0000h]

MCAN_TOCC is shown in Figure 27-44 and described in Table 27-38.

Return to the Summary Table.

MCAN Timeout Counter Configuration

Figure 27-44. MCAN_TOCC Register

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

Table 27-56. MCAN_TOCC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	TOP	R/WQ	FFFFh	Timeout Period. Start value of the Timeout Counter (down-counter). Configures the Timeout Period. Qualified Write is possible only with CCCR.CCE='1' and CCCR.INIT='1'.
15-3	RESERVED	R/W	0h	
2-1	TOS	R/WQ	0h	Timeout Select. When operating in Continuous mode, a write to TOCV presets the counter to the value configured by TOCC.TOP and continues down-counting. When the Timeout Counter is controlled by one of the FIFOs, an empty FIFO presets the counter to the value configured by TOCC.TOP. Down-counting is started when the first FIFO element is stored. 00 Continuous operation 01 Timeout controlled by Tx Event FIFO 10 Timeout controlled by Rx FIFO 0 11 Timeout controlled by Rx FIFO 1 Qualified Write is possible only with CCCR.CCE='1' and CCCR.INIT='1'.
0	ETOC	R/WQ	0h	Enable Timeout Counter 0 Timeout Counter disabled 1 Timeout Counter enabled Qualified Write is possible only with CCCR.CCE='1' and CCCR.INIT='1'.

ADVANCE INFORMATION

19 MCAN_TOCV Register (Offset = 702Ch) [Reset = 0000FFFFh]

MCAN_TOCV is shown in Figure 27-45 and described in Table 27-39.

Return to the Summary Table.

MCAN Timeout Counter Value

Figure 27-45. MCAN_TOCV Register

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

Table 27-58. MCAN_TOCV Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R/W	0h	
15-0	TOC	R/W	FFFFh	Timeout Counter. The Timeout Counter is decremented in multiples of CAN bit times, (1...16), depending on the configuration of TSCC.TCP. When decremented to zero, interrupt flag IR.TOO is set and the Timeout Counter is stopped. Start and reset/restart conditions are configured via TOCC.TOS.

20 MCAN_ECR Register (Offset = 7040h) [Reset = 00000000h]

MCAN_ECR is shown in Figure 27-46 and described in Table 27-40.

Return to the Summary Table.

MCAN Error Counter Register

Figure 27-46. MCAN_ECR Register

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

Table 27-60. MCAN_ECR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	RESERVED	R	0h	
23-16	CEL	RC	0h	CAN Error Logging. The counter is incremented each time when a CAN protocol error causes the Transmit Error Counter or the Receive Error Counter to be incremented. It is reset by read access to CEL. The counter stops at 0xFF the next increment of TEC or REC sets interrupt flag IR.ELO. Note: When CCCR.ASM is set, the CAN protocol controller does not increment TEC and REC when a CAN protocol error is detected, but CEL is still incremented.
15	RP	R	0h	Receive Error Passive 0 The Receive Error Counter is below the error passive level of 128 1 The Receive Error Counter has reached the error passive level of 128
14-8	REC	R	0h	Receive Error Counter. Actual state of the Receive Error Counter, values between 0 and 127. Note: When CCCR.ASM is set, the CAN protocol controller does not increment TEC and REC when a CAN protocol error is detected, but CEL is still incremented.
7-0	TEC	R	0h	Transmit Error Counter. Actual state of the Transmit Error Counter, values between 0 and 255. Note: When CCCR.ASM is set, the CAN protocol controller does not increment TEC and REC when a CAN protocol error is detected, but CEL is still incremented.

21 MCAN_PSR Register (Offset = 7044h) [Reset = 00000707h]

MCAN_PSR is shown in Figure 27-47 and described in Table 27-41.

Return to the Summary Table.

MCAN Protocol Status Register

Figure 27-47. MCAN_PSR Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED	TDCV						
R-0h	R-0h						
15	14	13	12	11	10	9	8
RESERVED	PXE	RFDF	RBRS	RESI	DLEC		
R-0h	RC-0h	RC-0h	RC-0h	RC-0h	RS-7h		
7	6	5	4	3	2	1	0
BO	EW	EP	ACT		LEC		
R-0h	R-0h	R-0h	R-0h		RS-7h		

Table 27-62. MCAN_PSR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-23	RESERVED	R	0h	
22-16	TDCV	R	0h	Transmitter Delay Compensation Value. Position of the secondary sample point, defined by the sum of the measured delay from the internal CAN TX signal to the internal CAN RX signal and TDCR.TDCO. The SSP position is, in the data phase, the number of mtq between the start of the transmitted bit and the secondary sample point. Valid values are 0 to 127 mtq.
15	RESERVED	R	0h	
14	PXE	RC	0h	Protocol Exception Event 0 No protocol exception event occurred since last read access 1 Protocol exception event occurred
13	RFDF	RC	0h	Received a CAN FD Message. This bit is set independent of acceptance filtering. 0 Since this bit was reset by the CPU, no CAN FD message has been received 1 Message in CAN FD format with FDF flag set has been received
12	RBRS	RC	0h	BRS Flag of Last Received CAN FD Message. This bit is set together with RFDF, independent of acceptance filtering. 0 Last received CAN FD message did not have its BRS flag set 1 Last received CAN FD message had its BRS flag set
11	RESI	RC	0h	ESI Flag of Last Received CAN FD Message. This bit is set together with RFDF, independent of acceptance filtering. 0 Last received CAN FD message did not have its ESI flag set 1 Last received CAN FD message had its ESI flag set
10-8	DLEC	RS	7h	Data Phase Last Error Code. Type of last error that occurred in the data phase of a CAN FD format frame with its BRS flag set. Coding is the same as for LEC. This field will be cleared to zero when a CAN FD format frame with its BRS flag set has been transferred (reception or transmission) without error.
7	BO	R	0h	Bus_Off Status 0 The M_CAN is not Bus_Off 1 The M_CAN is in Bus_Off state

Table 27-62. MCAN_PSR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
6	EW	R	0h	Warning Status 0 Both error counters are below the Error_Warning limit of 96 1 At least one of error counter has reached the Error_Warning limit of 96
5	EP	R	0h	Error Passive 0 The M_CAN is in the Error_Active state. It normally takes part in bus communication and sends an active error flag when an error has been detected 1 The M_CAN is in the Error_Passive state
4-3	ACT	R	0h	Node Activity. Monitors the module's CAN communication state. 00 Synchronizing - node is synchronizing on CAN communication 01 Idle - node is neither receiver nor transmitter 10 Receiver - node is operating as receiver 11 Transmitter - node is operating as transmitter Note: ACT is set to "00" by a Protocol Exception Event.
2-0	LEC	RS	7h	Last Error Code. The LEC indicates the type of the last error to occur on the CAN bus. This field will be cleared to '0' when a message has been transferred (reception or transmission) without error. 0 No Error: No error occurred since LEC has been reset by successful reception or transmission. 1 Stuff Error: More than 5 equal bits in a sequence have occurred in a part of a received message where this is not allowed. 2 Form Error: A fixed format part of a received frame has the wrong format. 3 AckError: The message transmitted by the MCAN was not acknowledged by another node. 4 Bit1Error: During the transmission of a message (with the exception of the arbitration field), the device wanted to send a recessive level (bit of logical value '1'), but the monitored bus value was dominant. 5 Bit0Error: During the transmission of a message (or acknowledge bit, or active error flag, or overload flag), the device wanted to send a dominant level (data or identifier bit logical value '0'), but the monitored bus value was recessive. During Bus_Off recovery this status is set each time a sequence of 11 recessive bits has been monitored. This enables the CPU to monitor the proceeding of the Bus_Off recovery sequence (indicating the bus is not stuck at dominant or continuously disturbed). 6 CRCError: The CRC check sum of a received message was incorrect. The CRC of an incoming message does not match with the CRC calculated from the received data. 7 NoChange: Any read access to the Protocol Status Register re-initializes the LEC to '7'. When the LEC shows the value '7', no CAN bus event was detected since the last CPU read access to the Protocol Status Register. Note: When a frame in CAN FD format has reached the data phase with BRS flag set, the next CAN event (error or valid frame) will be shown in DLEC instead of LEC. An error in a fixed stuff bit of a CAN FD CRC sequence will be shown as a Form Error, not Stuff Error. Note: The Bus_Off recovery sequence (see ISO 11898-1:2015) cannot be shortened by setting or resetting CCCR.INIT. If the device goes Bus_Off, it will set CCCR.INIT of its own accord, stopping all bus activities. Once CCCR.INIT has been cleared by the CPU, the device will then wait for 129 occurrences of Bus Idle (129 * 11 consecutive recessive bits) before resuming normal operation. At the end of the Bus_Off recovery sequence, the Error Management Counters will be reset. During the waiting time after the resetting of CCCR.INIT, each time a sequence of 11 recessive bits has been monitored, a Bit0Error code is written to PSR.LEC, enabling the CPU to readily check up whether the CAN bus is stuck at dominant or continuously disturbed and to monitor the Bus_Off recovery sequence. ECR.REC is used to count these sequences.

22 MCAN_TDCR Register (Offset = 7048h) [Reset = 00000000h]

MCAN_TDCR is shown in Figure 27-48 and described in Table 27-42.

Return to the Summary Table.

MCAN Transmitter Delay Compensation Register

Figure 27-48. MCAN_TDCR Register

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

Table 27-64. MCAN_TDCR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-15	RESERVED	R/W	0h	
14-8	TDCO	R/WQ	0h	Transmitter Delay Compensation Offset. Offset value defining the distance between the measured delay from the internal CAN TX signal to the internal CAN RX signal and the secondary sample point. Valid values are 0 to 127 mtq. Qualified Write is possible only with CCCR.CCE='1' and CCCR.INIT='1'.
7	RESERVED	R/W	0h	
6-0	TDCF	R/WQ	0h	Transmitter Delay Compensation Filter Window Length. Defines the minimum value for the SSP position, dominant edges on the internal CAN RX signal that would result in an earlier SSP position are ignored for transmitter delay measurement. The feature is enabled when TDCF is configured to a value greater than TDCO. Valid values are 0 to 127 mtq. Qualified Write is possible only with CCCR.CCE='1' and CCCR.INIT='1'.

23 MCAN_IR Register (Offset = 7050h) [Reset = 80000000h]

MCAN_IR is shown in Figure 27-49 and described in Table 27-43.

Return to the Summary Table.

The flags are set when one of the listed conditions is detected (edge-sensitive). The flags remain set until the Host clears them. A flag is cleared by writing a '1' to the corresponding bit position. Writing a '0' has no effect. A hard reset will clear the register. The configuration of IE controls whether an interrupt is generated. The configuration of ILS controls on which interrupt line an interrupt is signalled.

Figure 27-49. MCAN_IR Register

31	30	29	28	27	26	25	24
RESERVED		ARA	PED	PEA	WDI	BO	EW
R/W-2h		R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h
23	22	21	20	19	18	17	16
EP	ELO	BEU	RESERVED	DRX	TOO	MRAF	TSW
R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h
15	14	13	12	11	10	9	8
TEFL	TEFF	TEFW	TEFN	TFE	TCF	TC	HPM
R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h
7	6	5	4	3	2	1	0
RF1L	RF1F	RF1W	RF1N	RF0L	RF0F	RF0W	RF0N
R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h

Table 27-66. MCAN_IR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	RESERVED	R/W	2h	
29	ARA	R/W1C	0h	Access to Reserved Address 0 No access to reserved address occurred 1 Access to reserved address occurred
28	PED	R/W1C	0h	Protocol Error in Data Phase (Data Bit Time is used) 0 No protocol error in data phase 1 Protocol error in data phase detected (PSR.DLEC ? 0,7)
27	PEA	R/W1C	0h	Protocol Error in Arbitration Phase (Nominal Bit Time is used) 0 No protocol error in arbitration phase 1 Protocol error in arbitration phase detected (PSR.LEC ? 0,7)
26	WDI	R/W1C	0h	Watchdog Interrupt 0 No Message RAM Watchdog event occurred 1 Message RAM Watchdog event due to missing READY
25	BO	R/W1C	0h	Bus_Off Status 0 Bus_Off status unchanged 1 Bus_Off status changed
24	EW	R/W1C	0h	Warning Status 0 Error_Warning status unchanged 1 Error_Warning status changed
23	EP	R/W1C	0h	Error Passive 0 Error_Passive status unchanged 1 Error_Passive status changed
22	ELO	R/W1C	0h	Error Logging Overflow 0 CAN Error Logging Counter did not overflow 1 Overflow of CAN Error Logging Counter occurred

Table 27-66. MCAN_IR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
21	BEU	R/W1C	0h	Bit Error Uncorrected. Message RAM bit error detected, uncorrected. This bit is set when a double bit error is detected by the ECC aggregator attached to the Message RAM. An uncorrected Message RAM bit error sets CCCR.INIT to '1'. This is done to avoid transmission of corrupted data. 0 No bit error detected when reading from Message RAM 1 Bit error detected, uncorrected (e.g. parity logic)
20	RESERVED	R/W	0h	
19	DRX	R/W1C	0h	Message Stored to Dedicated Rx Buffer. The flag is set whenever a received message has been stored into a dedicated Rx Buffer. 0 No Rx Buffer updated 1 At least one received message stored into an Rx Buffer
18	TOO	R/W1C	0h	Timeout Occurred 0 No timeout 1 Timeout reached
17	MRAF	R/W1C	0h	Message RAM Access Failure. The flag is set, when the Rx Handler: - has not completed acceptance filtering or storage of an accepted message until the arbitration field of the following message has been received. In this case acceptance filtering or message storage is aborted and the Rx Handler starts processing of the following message. - was not able to write a message to the Message RAM. In this case message storage is aborted. In both cases the FIFO put index is not updated resp. the New Data flag for a dedicated Rx Buffer is not set, a partly stored message is overwritten when the next message is stored to this location. The flag is also set when the Tx Handler was not able to read a message from the Message RAM in time. In this case message transmission is aborted. In case of a Tx Handler access failure the MCAN is switched into Restricted Operation Mode. To leave Restricted Operation Mode, the Host CPU has to reset CCCR.ASM. 0 No Message RAM access failure occurred 1 Message RAM access failure occurred
16	TSW	R/W1C	0h	Timestamp Wraparound 0 No timestamp counter wrap-around 1 Timestamp counter wrapped around
15	TEFL	R/W1C	0h	Tx Event FIFO Element Lost 0 No Tx Event FIFO element lost 1 Tx Event FIFO element lost, also set after write attempt to Tx Event FIFO of size zero
14	TEFF	R/W1C	0h	Tx Event FIFO Full 0 Tx Event FIFO not full 1 Tx Event FIFO full
13	TEFW	R/W1C	0h	Tx Event FIFO Watermark Reached 0 Tx Event FIFO fill level below watermark 1 Tx Event FIFO fill level reached watermark
12	TEFN	R/W1C	0h	Tx Event FIFO New Entry 0 Tx Event FIFO unchanged 1 Tx Handler wrote Tx Event FIFO element
11	TFE	R/W1C	0h	Tx FIFO Empty 0 Tx FIFO non-empty 1 Tx FIFO empty
10	TCF	R/W1C	0h	Transmission Cancellation Finished 0 No transmission cancellation finished 1 Transmission cancellation finished
9	TC	R/W1C	0h	Transmission Completed 0 No transmission completed 1 Transmission completed

Table 27-66. MCAN_IR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
8	HPM	R/W1C	0h	High Priority Message 0 No high priority message received 1 High priority message received
7	RF1L	R/W1C	0h	Rx FIFO 1 Message Lost 0 No Rx FIFO 1 message lost 1 Rx FIFO 1 message lost, also set after write attempt to Rx FIFO 1 of size zero
6	RF1F	R/W1C	0h	Rx FIFO 1 Full 0 Rx FIFO 1 not full 1 Rx FIFO 1 full
5	RF1W	R/W1C	0h	Rx FIFO 1 Watermark Reached 0 Rx FIFO 1 fill level below watermark 1 Rx FIFO 1 fill level reached watermark
4	RF1N	R/W1C	0h	Rx FIFO 1 New Message 0 No new message written to Rx FIFO 1 1 New message written to Rx FIFO 1
3	RF0L	R/W1C	0h	Rx FIFO 0 Message Lost 0 No Rx FIFO 0 message lost 1 Rx FIFO 0 message lost, also set after write attempt to Rx FIFO 0 of size zero
2	RF0F	R/W1C	0h	Rx FIFO 0 Full 0 Rx FIFO 0 not full 1 Rx FIFO 0 full
1	RF0W	R/W1C	0h	Rx FIFO 0 Watermark Reached 0 Rx FIFO 0 fill level below watermark 1 Rx FIFO 0 fill level reached watermark
0	RF0N	R/W1C	0h	Rx FIFO 0 New Message 0 No new message written to Rx FIFO 0 1 New message written to Rx FIFO 0

24 MCAN_IE Register (Offset = 7054h) [Reset = 00000000h]

MCAN_IE is shown in Figure 27-50 and described in Table 27-44.

Return to the Summary Table.

MCAN Interrupt Enable

Figure 27-50. MCAN_IE Register

31	30	29	28	27	26	25	24
RESERVED		ARAE	PEDE	PEAE	WDIE	BOE	EWE
R/W-0h		R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
EPE	ELOE	BEUE	BECE	DRXE	TOOE	MRAFE	TSWE
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
TEFLE	TEFFE	TEFWE	TEFNE	TFEE	TCFE	TCE	HPME
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
RF1LE	RF1FE	RF1WE	RF1NE	RF0LE	RF0FE	RF0WE	RF0NE
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 27-68. MCAN_IE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	RESERVED	R/W	0h	
29	ARAE	R/W	0h	Access to Reserved Address Enable
28	PEDE	R/W	0h	Protocol Error in Data Phase Enable
27	PEAE	R/W	0h	Protocol Error in Arbitration Phase Enable
26	WDIE	R/W	0h	Watchdog Interrupt Enable
25	BOE	R/W	0h	Bus_Off Status Enable
24	EWE	R/W	0h	Warning Status Enable
23	EPE	R/W	0h	Error Passive Enable
22	ELOE	R/W	0h	Error Logging Overflow Enable
21	BEUE	R/W	0h	Bit Error Uncorrected Enable
20	BECE	R/W	0h	Bit Error Corrected Enable A separate interrupt line reserved for corrected bit errors is provided via the MCAN_ERROR_REGS. It advised for the user to use these registers and leave this bit cleared to '0'.
19	DRXE	R/W	0h	Message Stored to Dedicated Rx Buffer Enable
18	TOOE	R/W	0h	Timeout Occurred Enable
17	MRAFE	R/W	0h	Message RAM Access Failure Enable
16	TSWE	R/W	0h	Timestamp Wraparound Enable
15	TEFLE	R/W	0h	Tx Event FIFO Element Lost Enable
14	TEFFE	R/W	0h	Tx Event FIFO Full Enable
13	TEFWE	R/W	0h	Tx Event FIFO Watermark Reached Enable
12	TEFNE	R/W	0h	Tx Event FIFO New Entry Enable
11	TFEE	R/W	0h	Tx FIFO Empty Enable
10	TCFE	R/W	0h	Transmission Cancellation Finished Enable
9	TCE	R/W	0h	Transmission Completed Enable
8	HPME	R/W	0h	High Priority Message Enable

Table 27-68. MCAN_IE Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
7	RF1LE	R/W	0h	Rx FIFO 1 Message Lost Enable
6	RF1FE	R/W	0h	Rx FIFO 1 Full Enable
5	RF1WE	R/W	0h	Rx FIFO 1 Watermark Reached Enable
4	RF1NE	R/W	0h	Rx FIFO 1 New Message Enable
3	RF0LE	R/W	0h	Rx FIFO 0 Message Lost Enable
2	RF0FE	R/W	0h	Rx FIFO 0 Full Enable
1	RF0WE	R/W	0h	Rx FIFO 0 Watermark Reached Enable
0	RF0NE	R/W	0h	Rx FIFO 0 New Message Enable

25 MCAN_ILS Register (Offset = 7058h) [Reset = 00000000h]

MCAN_ILS is shown in Figure 27-51 and described in Table 27-45.

Return to the Summary Table.

The Interrupt Line Select register assigns an interrupt generated by a specific interrupt flag from the Interrupt Register to one of the two module interrupt lines. For interrupt generation the respective interrupt line has to be enabled via ILE.EINT0 and ILE.EINT1.

Figure 27-51. MCAN_ILS Register

31	30	29	28	27	26	25	24
RESERVED		ARAL	PEDL	PEAL	WDIL	BOL	EWL
R/W-0h		R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
EPL	ELOL	BEUL	BECL	DRXL	TOOL	MRAFL	TSWL
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
TEFLL	TEFFL	TEFWL	TEFNL	TFEL	TCFL	TCL	HPML
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
RF1LL	RF1FL	RF1WL	RF1NL	RF0LL	RF0FL	RF0WL	RF0NL
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 27-70. MCAN_ILS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	RESERVED	R/W	0h	
29	ARAL	R/W	0h	Access to Reserved Address Line 0 Interrupt source is assigned to Interrupt Line 0 1 Interrupt source is assigned to Interrupt Line 1
28	PEDL	R/W	0h	Protocol Error in Data Phase Line 0 Interrupt source is assigned to Interrupt Line 0 1 Interrupt source is assigned to Interrupt Line 1
27	PEAL	R/W	0h	Protocol Error in Arbitration Phase Line 0 Interrupt source is assigned to Interrupt Line 0 1 Interrupt source is assigned to Interrupt Line 1
26	WDIL	R/W	0h	Watchdog Interrupt Line 0 Interrupt source is assigned to Interrupt Line 0 1 Interrupt source is assigned to Interrupt Line 1
25	BOL	R/W	0h	Bus_Off Status Line 0 Interrupt source is assigned to Interrupt Line 0 1 Interrupt source is assigned to Interrupt Line 1
24	EWL	R/W	0h	Warning Status Line 0 Interrupt source is assigned to Interrupt Line 0 1 Interrupt source is assigned to Interrupt Line 1
23	EPL	R/W	0h	Error Passive Line 0 Interrupt source is assigned to Interrupt Line 0 1 Interrupt source is assigned to Interrupt Line 1
22	ELOL	R/W	0h	Error Logging Overflow Line 0 Interrupt source is assigned to Interrupt Line 0 1 Interrupt source is assigned to Interrupt Line 1
21	BEUL	R/W	0h	Bit Error Uncorrected Line 0 Interrupt source is assigned to Interrupt Line 0 1 Interrupt source is assigned to Interrupt Line 1

Table 27-70. MCAN_ILS Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
20	BECL	R/W	0h	Bit Error Corrected Line A separate interrupt line reserved for corrected bit errors is provided via the MCAN_ERROR_REGS. It advised for the user to use these registers and leave the MCAN_IE.BECE bit cleared to '0' (disabled), thereby relegating this bit to not applicable.
19	DRXL	R/W	0h	Message Stored to Dedicated Rx Buffer Line 0 Interrupt source is assigned to Interrupt Line 0 1 Interrupt source is assigned to Interrupt Line 1
18	TOOL	R/W	0h	Timeout Occurred Line 0 Interrupt source is assigned to Interrupt Line 0 1 Interrupt source is assigned to Interrupt Line 1
17	MRAFL	R/W	0h	Message RAM Access Failure Line 0 Interrupt source is assigned to Interrupt Line 0 1 Interrupt source is assigned to Interrupt Line 1
16	TSWL	R/W	0h	Timestamp Wraparound Line 0 Interrupt source is assigned to Interrupt Line 0 1 Interrupt source is assigned to Interrupt Line 1
15	TEFLL	R/W	0h	Tx Event FIFO Element Lost Line 0 Interrupt source is assigned to Interrupt Line 0 1 Interrupt source is assigned to Interrupt Line 1
14	TEFFL	R/W	0h	Tx Event FIFO Full Line 0 Interrupt source is assigned to Interrupt Line 0 1 Interrupt source is assigned to Interrupt Line 1
13	TEFWL	R/W	0h	Tx Event FIFO Watermark Reached Line 0 Interrupt source is assigned to Interrupt Line 0 1 Interrupt source is assigned to Interrupt Line 1
12	TEFNL	R/W	0h	Tx Event FIFO New Entry Line 0 Interrupt source is assigned to Interrupt Line 0 1 Interrupt source is assigned to Interrupt Line 1
11	TFEL	R/W	0h	Tx FIFO Empty Line 0 Interrupt source is assigned to Interrupt Line 0 1 Interrupt source is assigned to Interrupt Line 1
10	TCFL	R/W	0h	Transmission Cancellation Finished Line 0 Interrupt source is assigned to Interrupt Line 0 1 Interrupt source is assigned to Interrupt Line 1
9	TCL	R/W	0h	Transmission Completed Line 0 Interrupt source is assigned to Interrupt Line 0 1 Interrupt source is assigned to Interrupt Line 1
8	HPML	R/W	0h	High Priority Message Line 0 Interrupt source is assigned to Interrupt Line 0 1 Interrupt source is assigned to Interrupt Line 1
7	RF1LL	R/W	0h	Rx FIFO 1 Message Lost Line 0 Interrupt source is assigned to Interrupt Line 0 1 Interrupt source is assigned to Interrupt Line 1
6	RF1FL	R/W	0h	Rx FIFO 1 Full Line 0 Interrupt source is assigned to Interrupt Line 0 1 Interrupt source is assigned to Interrupt Line 1
5	RF1WL	R/W	0h	Rx FIFO 1 Watermark Reached Line 0 Interrupt source is assigned to Interrupt Line 0 1 Interrupt source is assigned to Interrupt Line 1
4	RF1NL	R/W	0h	Rx FIFO 1 New Message Line 0 Interrupt source is assigned to Interrupt Line 0 1 Interrupt source is assigned to Interrupt Line 1
3	RF0LL	R/W	0h	Rx FIFO 0 Message Lost Line 0 Interrupt source is assigned to Interrupt Line 0 1 Interrupt source is assigned to Interrupt Line 1

Table 27-70. MCAN_ILS Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
2	RF0FL	R/W	0h	Rx FIFO 0 Full Line 0 Interrupt source is assigned to Interrupt Line 0 1 Interrupt source is assigned to Interrupt Line 1
1	RF0WL	R/W	0h	Rx FIFO 0 Watermark Reached Line 0 Interrupt source is assigned to Interrupt Line 0 1 Interrupt source is assigned to Interrupt Line 1
0	RF0NL	R/W	0h	Rx FIFO 0 New Message Line 0 Interrupt source is assigned to Interrupt Line 0 1 Interrupt source is assigned to Interrupt Line 1

26 MCAN_ILE Register (Offset = 705Ch) [Reset = 00000000h]

MCAN_ILE is shown in Figure 27-52 and described in Table 27-46.

Return to the Summary Table.

MCAN Interrupt Line Enable

Figure 27-52. MCAN_ILE Register

31	30	29	28	27	26	25	24
RESERVED							
R/W-0h							
23	22	21	20	19	18	17	16
RESERVED							
R/W-0h							
15	14	13	12	11	10	9	8
RESERVED							
R/W-0h							
7	6	5	4	3	2	1	0
RESERVED						EINT1	EINT0
R/W-0h						R/W-0h	R/W-0h

Table 27-72. MCAN_ILE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R/W	0h	
1	EINT1	R/W	0h	Enable Interrupt Line 1 0 Interrupt Line 1 is disabled 1 Interrupt Line 1 is enabled
0	EINT0	R/W	0h	Enable Interrupt Line 0 0 Interrupt Line 0 is disabled 1 Interrupt Line 0 is enabled

ADVANCE INFORMATION

27 MCAN_GFC Register (Offset = 7080h) [Reset = 00000000h]

MCAN_GFC is shown in Figure 27-53 and described in Table 27-47.

Return to the Summary Table.

MCAN Global Filter Configuration

Figure 27-53. MCAN_GFC Register

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

Table 27-74. MCAN_GFC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-6	RESERVED	R/W	0h	
5-4	ANFS	R/WQ	0h	Accept Non-matching Frames Standard. Defines how received messages with 11-bit IDs that do not match any element of the filter list are treated. 00 Accept in Rx FIFO 0 01 Accept in Rx FIFO 1 10 Reject 11 Reject Qualified Write is possible only with CCCR.CCE='1' and CCCR.INIT='1'.
3-2	ANFE	R/WQ	0h	Accept Non-matching Frames Extended. Defines how received messages with 29-bit IDs that do not match any element of the filter list are treated. 00 Accept in Rx FIFO 0 01 Accept in Rx FIFO 1 10 Reject 11 Reject Qualified Write is possible only with CCCR.CCE='1' and CCCR.INIT='1'.
1	RRFS	R/WQ	0h	Reject Remote Frames Standard 0 Filter remote frames with 11-bit standard IDs 1 Reject all remote frames with 11-bit standard IDs Qualified Write is possible only with CCCR.CCE='1' and CCCR.INIT='1'.
0	RRFE	R/WQ	0h	Reject Remote Frames Extended 0 Filter remote frames with 29-bit extended IDs 1 Reject all remote frames with 29-bit extended IDs Qualified Write is possible only with CCCR.CCE='1' and CCCR.INIT='1'.

28 MCAN_SIDFC Register (Offset = 7084h) [Reset = 00000000h]

MCAN_SIDFC is shown in Figure 27-54 and described in Table 27-48.

Return to the Summary Table.

MCAN Standard ID Filter Configuration

Figure 27-54. MCAN_SIDFC Register

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

Table 27-76. MCAN_SIDFC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	RESERVED	R/W	0h	
23-16	LSS	R/WQ	0h	List Size Standard 0 No standard Message ID filter 1-128 Number of standard Message ID filter elements >128 Values greater than 128 are interpreted as 128
15-2	FLSSA	R/WQ	0h	Filter List Standard Start Address. Start address of standard Message ID filter list (32-bit word address).
1-0	RESERVED	R/W	0h	

29 MCAN_XIDFC Register (Offset = 7088h) [Reset = 00000000h]

MCAN_XIDFC is shown in Figure 27-55 and described in Table 27-49.

Return to the Summary Table.

MCAN Extended ID Filter Configuration

Figure 27-55. MCAN_XIDFC Register

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

Table 27-78. MCAN_XIDFC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-23	RESERVED	R/W	0h	
22-16	LSE	R/WQ	0h	Filter List Extended Start Address. Start address of extended Message ID filter list (32-bit word address). Qualified Write is possible only with CCCR.CCE='1' and CCCR.INIT='1'.
15-2	FLESA	R/WQ	0h	List Size Extended 0 No extended Message ID filter 1-64 Number of extended Message ID filter elements >64 Values greater than 64 are interpreted as 64 Qualified Write is possible only with CCCR.CCE='1' and CCCR.INIT='1'.
1-0	RESERVED	R/W	0h	

30 MCAN_XIDAM Register (Offset = 7090h) [Reset = 1FFFFFFFh]

MCAN_XIDAM is shown in Figure 27-56 and described in Table 27-50.

Return to the Summary Table.

MCAN Extended ID and Mask

Figure 27-56. MCAN_XIDAM Register

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

Table 27-80. MCAN_XIDAM Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R/W	0h	
28-0	EIDM	R/WQ	1FFFFFFFh	Extended ID Mask. For acceptance filtering of extended frames the Extended ID AND Mask is ANDed with the Message ID of a received frame. Intended for masking of 29-bit IDs in SAE J1939. With the reset value of all bits set to one the mask is not active. Qualified Write is possible only with CCCR.CCE='1' and CCCR.INIT='1'.

31 MCAN_HPMS Register (Offset = 7094h) [Reset = 00000000h]

MCAN_HPMS is shown in Figure 27-57 and described in Table 27-51.

Return to the Summary Table.

This register is updated every time a Message ID filter element configured to generate a priority event matches. This can be used to monitor the status of incoming high priority messages and to enable fast access to these messages.

Figure 27-57. MCAN_HPMS Register

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

Table 27-82. MCAN_HPMS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	
15	FLST	R	0h	Filter List. Indicates the filter list of the matching filter element. 0 Standard Filter List 1 Extended Filter List
14-8	FIDX	R	0h	Filter Index. Index of matching filter element. Range is 0 to SIDFC.LSS - 1 resp. XIDFC.LSE - 1.
7-6	MSI	R	0h	Message Storage Indicator 00 No FIFO selected 01 FIFO message lost 10 Message stored in FIFO 0 11 Message stored in FIFO 1
5-0	BIDX	R	0h	Buffer Index. Index of Rx FIFO element to which the message was stored. Only valid when MSI(1) = '1'.

32 MCAN_NDAT1 Register (Offset = 7098h) [Reset = 00000000h]

MCAN_NDAT1 is shown in Figure 27-58 and described in Table 27-52.

Return to the Summary Table.

MCAN New Data 1

Figure 27-58. MCAN_NDAT1 Register

31	30	29	28	27	26	25	24
ND31	ND30	ND29	ND28	ND27	ND26	ND25	ND24
R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h
23	22	21	20	19	18	17	16
ND23	ND22	ND21	ND20	ND19	ND18	ND17	ND16
R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h
15	14	13	12	11	10	9	8
ND15	ND14	ND13	ND12	ND11	ND10	ND9	ND8
R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h
7	6	5	4	3	2	1	0
ND7	ND6	ND5	ND4	ND3	ND2	ND1	ND0
R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h

Table 27-84. MCAN_NDAT1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	ND31	R/W1C	0h	New Data RX Buffer 31 0 Rx Buffer not updated 1 Rx Buffer updated from new message
30	ND30	R/W1C	0h	New Data RX Buffer 30 0 Rx Buffer not updated 1 Rx Buffer updated from new message
29	ND29	R/W1C	0h	New Data RX Buffer 29 0 Rx Buffer not updated 1 Rx Buffer updated from new message
28	ND28	R/W1C	0h	New Data RX Buffer 28 0 Rx Buffer not updated 1 Rx Buffer updated from new message
27	ND27	R/W1C	0h	New Data RX Buffer 27 0 Rx Buffer not updated 1 Rx Buffer updated from new message
26	ND26	R/W1C	0h	New Data RX Buffer 26 0 Rx Buffer not updated 1 Rx Buffer updated from new message
25	ND25	R/W1C	0h	New Data RX Buffer 25 0 Rx Buffer not updated 1 Rx Buffer updated from new message
24	ND24	R/W1C	0h	New Data RX Buffer 24 0 Rx Buffer not updated 1 Rx Buffer updated from new message
23	ND23	R/W1C	0h	New Data RX Buffer 23 0 Rx Buffer not updated 1 Rx Buffer updated from new message
22	ND22	R/W1C	0h	New Data RX Buffer 22 0 Rx Buffer not updated 1 Rx Buffer updated from new message

Table 27-84. MCAN_NDAT1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
21	ND21	R/W1C	0h	New Data RX Buffer 21 0 Rx Buffer not updated 1 Rx Buffer updated from new message
20	ND20	R/W1C	0h	New Data RX Buffer 20 0 Rx Buffer not updated 1 Rx Buffer updated from new message
19	ND19	R/W1C	0h	New Data RX Buffer 19 0 Rx Buffer not updated 1 Rx Buffer updated from new message
18	ND18	R/W1C	0h	New Data RX Buffer 18 0 Rx Buffer not updated 1 Rx Buffer updated from new message
17	ND17	R/W1C	0h	New Data RX Buffer 17 0 Rx Buffer not updated 1 Rx Buffer updated from new message
16	ND16	R/W1C	0h	New Data RX Buffer 16 0 Rx Buffer not updated 1 Rx Buffer updated from new message
15	ND15	R/W1C	0h	New Data RX Buffer 15 0 Rx Buffer not updated 1 Rx Buffer updated from new message
14	ND14	R/W1C	0h	New Data RX Buffer 14 0 Rx Buffer not updated 1 Rx Buffer updated from new message
13	ND13	R/W1C	0h	New Data RX Buffer 13 0 Rx Buffer not updated 1 Rx Buffer updated from new message
12	ND12	R/W1C	0h	New Data RX Buffer 12 0 Rx Buffer not updated 1 Rx Buffer updated from new message
11	ND11	R/W1C	0h	New Data RX Buffer 11 0 Rx Buffer not updated 1 Rx Buffer updated from new message
10	ND10	R/W1C	0h	New Data RX Buffer 10 0 Rx Buffer not updated 1 Rx Buffer updated from new message
9	ND9	R/W1C	0h	New Data RX Buffer 9 0 Rx Buffer not updated 1 Rx Buffer updated from new message
8	ND8	R/W1C	0h	New Data RX Buffer 8 0 Rx Buffer not updated 1 Rx Buffer updated from new message
7	ND7	R/W1C	0h	New Data RX Buffer 7 0 Rx Buffer not updated 1 Rx Buffer updated from new message
6	ND6	R/W1C	0h	New Data RX Buffer 6 0 Rx Buffer not updated 1 Rx Buffer updated from new message
5	ND5	R/W1C	0h	New Data RX Buffer 5 0 Rx Buffer not updated 1 Rx Buffer updated from new message
4	ND4	R/W1C	0h	New Data RX Buffer 4 0 Rx Buffer not updated 1 Rx Buffer updated from new message
3	ND3	R/W1C	0h	New Data RX Buffer 3 0 Rx Buffer not updated 1 Rx Buffer updated from new message

Table 27-84. MCAN_NDAT1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
2	ND2	R/W1C	0h	New Data RX Buffer 2 0 Rx Buffer not updated 1 Rx Buffer updated from new message
1	ND1	R/W1C	0h	New Data RX Buffer 1 0 Rx Buffer not updated 1 Rx Buffer updated from new message
0	ND0	R/W1C	0h	New Data RX Buffer 0 0 Rx Buffer not updated 1 Rx Buffer updated from new message

33 MCAN_NDAT2 Register (Offset = 709Ch) [Reset = 00000000h]

MCAN_NDAT2 is shown in Figure 27-59 and described in Table 27-53.

Return to the Summary Table.

MCAN New Data 2

Figure 27-59. MCAN_NDAT2 Register

31	30	29	28	27	26	25	24
ND63	ND62	ND61	ND60	ND59	ND58	ND57	ND56
R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h
23	22	21	20	19	18	17	16
ND55	ND54	ND53	ND52	ND51	ND50	ND49	ND48
R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h
15	14	13	12	11	10	9	8
ND47	ND46	ND45	ND44	ND43	ND42	ND41	ND40
R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h
7	6	5	4	3	2	1	0
ND39	ND38	ND37	ND36	ND35	ND34	ND33	ND32
R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h

Table 27-86. MCAN_NDAT2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	ND63	R/W1C	0h	New Data RX Buffer 63 0 Rx Buffer not updated 1 Rx Buffer updated from new message
30	ND62	R/W1C	0h	New Data RX Buffer 62 0 Rx Buffer not updated 1 Rx Buffer updated from new message
29	ND61	R/W1C	0h	New Data RX Buffer 61 0 Rx Buffer not updated 1 Rx Buffer updated from new message
28	ND60	R/W1C	0h	New Data RX Buffer 60 0 Rx Buffer not updated 1 Rx Buffer updated from new message
27	ND59	R/W1C	0h	New Data RX Buffer 59 0 Rx Buffer not updated 1 Rx Buffer updated from new message
26	ND58	R/W1C	0h	New Data RX Buffer 58 0 Rx Buffer not updated 1 Rx Buffer updated from new message
25	ND57	R/W1C	0h	New Data RX Buffer 57 0 Rx Buffer not updated 1 Rx Buffer updated from new message
24	ND56	R/W1C	0h	New Data RX Buffer 56 0 Rx Buffer not updated 1 Rx Buffer updated from new message
23	ND55	R/W1C	0h	New Data RX Buffer 55 0 Rx Buffer not updated 1 Rx Buffer updated from new message
22	ND54	R/W1C	0h	New Data RX Buffer 54 0 Rx Buffer not updated 1 Rx Buffer updated from new message

Table 27-86. MCAN_NDAT2 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
21	ND53	R/W1C	0h	New Data RX Buffer 53 0 Rx Buffer not updated 1 Rx Buffer updated from new message
20	ND52	R/W1C	0h	New Data RX Buffer 52 0 Rx Buffer not updated 1 Rx Buffer updated from new message
19	ND51	R/W1C	0h	New Data RX Buffer 51 0 Rx Buffer not updated 1 Rx Buffer updated from new message
18	ND50	R/W1C	0h	New Data RX Buffer 50 0 Rx Buffer not updated 1 Rx Buffer updated from new message
17	ND49	R/W1C	0h	New Data RX Buffer 49 0 Rx Buffer not updated 1 Rx Buffer updated from new message
16	ND48	R/W1C	0h	New Data RX Buffer 48 0 Rx Buffer not updated 1 Rx Buffer updated from new message
15	ND47	R/W1C	0h	New Data RX Buffer 47 0 Rx Buffer not updated 1 Rx Buffer updated from new message
14	ND46	R/W1C	0h	New Data RX Buffer 46 0 Rx Buffer not updated 1 Rx Buffer updated from new message
13	ND45	R/W1C	0h	New Data RX Buffer 45 0 Rx Buffer not updated 1 Rx Buffer updated from new message
12	ND44	R/W1C	0h	New Data RX Buffer 44 0 Rx Buffer not updated 1 Rx Buffer updated from new message
11	ND43	R/W1C	0h	New Data RX Buffer 43 0 Rx Buffer not updated 1 Rx Buffer updated from new message
10	ND42	R/W1C	0h	New Data RX Buffer 42 0 Rx Buffer not updated 1 Rx Buffer updated from new message
9	ND41	R/W1C	0h	New Data RX Buffer 41 0 Rx Buffer not updated 1 Rx Buffer updated from new message
8	ND40	R/W1C	0h	New Data RX Buffer 40 0 Rx Buffer not updated 1 Rx Buffer updated from new message
7	ND39	R/W1C	0h	New Data RX Buffer 39 0 Rx Buffer not updated 1 Rx Buffer updated from new message
6	ND38	R/W1C	0h	New Data RX Buffer 38 0 Rx Buffer not updated 1 Rx Buffer updated from new message
5	ND37	R/W1C	0h	New Data RX Buffer 37 0 Rx Buffer not updated 1 Rx Buffer updated from new message
4	ND36	R/W1C	0h	New Data RX Buffer 36 0 Rx Buffer not updated 1 Rx Buffer updated from new message
3	ND35	R/W1C	0h	New Data RX Buffer 35 0 Rx Buffer not updated 1 Rx Buffer updated from new message

Table 27-86. MCAN_NDAT2 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
2	ND34	R/W1C	0h	New Data RX Buffer 34 0 Rx Buffer not updated 1 Rx Buffer updated from new message
1	ND33	R/W1C	0h	New Data RX Buffer 33 0 Rx Buffer not updated 1 Rx Buffer updated from new message
0	ND32	R/W1C	0h	New Data RX Buffer 32 0 Rx Buffer not updated 1 Rx Buffer updated from new message

34 MCAN_RXF0C Register (Offset = 70A0h) [Reset = 00000000h]

MCAN_RXF0C is shown in Figure 27-60 and described in Table 27-54.

Return to the Summary Table.

MCAN Rx FIFO 0 Configuration

Figure 27-60. MCAN_RXF0C Register

31	30	29	28	27	26	25	24
F0OM	F0WM						
R/WQ-0h	R/WQ-0h						
23	22	21	20	19	18	17	16
RESERVED	F0S						
R/W-0h	R/WQ-0h						
15	14	13	12	11	10	9	8
F0SA							
R/WQ-0h							
7	6	5	4	3	2	1	0
F0SA						RESERVED	
R/WQ-0h						R/W-0h	

Table 27-88. MCAN_RXF0C Register Field Descriptions

Bit	Field	Type	Reset	Description
31	F0OM	R/WQ	0h	FIFO 0 Operation Mode. FIFO 0 can be operated in blocking or in overwrite mode. 0 FIFO 0 blocking mode 1 FIFO 0 overwrite mode Qualified Write is possible only with CCCR.CCE='1' and CCCR.INIT='1'.
30-24	F0WM	R/WQ	0h	Rx FIFO 0 Watermark 0 Watermark interrupt disabled 1-64 Level for Rx FIFO 0 watermark interrupt (IR.RF0W) >64 Watermark interrupt disabled Qualified Write is possible only with CCCR.CCE='1' and CCCR.INIT='1'.
23	RESERVED	R/W	0h	
22-16	F0S	R/WQ	0h	Rx FIFO 0 Size. The Rx FIFO 0 elements are indexed from 0 to F0S-1. 0 No Rx FIFO 0 1-64 Number of Rx FIFO 0 elements >64 Values greater than 64 are interpreted as 64 Qualified Write is possible only with CCCR.CCE='1' and CCCR.INIT='1'.
15-2	F0SA	R/WQ	0h	Rx FIFO 0 Start Address. Start address of Rx FIFO 0 in Message RAM (32-bit word address). Qualified Write is possible only with CCCR.CCE='1' and CCCR.INIT='1'.
1-0	RESERVED	R/W	0h	

35 MCAN_RXF0S Register (Offset = 70A4h) [Reset = 00000000h]

MCAN_RXF0S is shown in Figure 27-61 and described in Table 27-55.

Return to the Summary Table.

MCAN Rx FIFO 0 Status

Figure 27-61. MCAN_RXF0S Register

31	30	29	28	27	26	25	24
RESERVED						RF0L	F0F
R-0h						R-0h	R-0h
23	22	21	20	19	18	17	16
RESERVED		F0PI					
R-0h		R-0h					
15	14	13	12	11	10	9	8
RESERVED		F0GI					
R-0h		R-0h					
7	6	5	4	3	2	1	0
RESERVED	F0FL						
R-0h	R-0h						

Table 27-90. MCAN_RXF0S Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	RESERVED	R	0h	
25	RF0L	R	0h	Rx FIFO 0 Message Lost. This bit is a copy of interrupt flag IR.RF0L. When IR.RF0L is reset, this bit is also reset. 0 No Rx FIFO 0 message lost 1 Rx FIFO 0 message lost, also set after write attempt to Rx FIFO 0 of size zero Note: Overwriting the oldest message when RXF0C.F0OM = '1' will not set this flag.
24	F0F	R	0h	Rx FIFO 0 Full 0 Rx FIFO 0 not full 1 Rx FIFO 0 full
23-22	RESERVED	R	0h	
21-16	F0PI	R	0h	Rx FIFO 0 Put Index. Rx FIFO 0 write index pointer, range 0 to 63.
15-14	RESERVED	R	0h	
13-8	F0GI	R	0h	Rx FIFO 0 Get Index. Rx FIFO 0 read index pointer, range 0 to 63.
7	RESERVED	R	0h	
6-0	F0FL	R	0h	Rx FIFO 0 Fill Level. Number of elements stored in Rx FIFO 0, range 0 to 64.

36 MCAN_RXF0A Register (Offset = 70A8h) [Reset = 00000000h]

MCAN_RXF0A is shown in Figure 27-62 and described in Table 27-56.

Return to the Summary Table.

MCAN Rx FIFO 0 Acknowledge

Figure 27-62. MCAN_RXF0A Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																										F0AI					
R/W-0h																										R/W-0h					

Table 27-92. MCAN_RXF0A Register Field Descriptions

Bit	Field	Type	Reset	Description
31-6	RESERVED	R/W	0h	
5-0	F0AI	R/W	0h	Rx FIFO 0 Acknowledge Index. After the Host has read a message or a sequence of messages from Rx FIFO 0 it has to write the buffer index of the last element read from Rx FIFO 0 to F0AI. This will set the Rx FIFO 0 Get Index RXF0S.F0GI to F0AI + 1 and update the FIFO 0 Fill Level RXF0S.F0FL.

37 MCAN_RXBC Register (Offset = 70ACh) [Reset = 00000000h]

MCAN_RXBC is shown in Figure 27-63 and described in Table 27-57.

Return to the Summary Table.

MCAN Rx Buffer Configuration

Figure 27-63. MCAN_RXBC Register

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

Table 27-94. MCAN_RXBC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R/W	0h	
15-2	RBSA	R/WQ	0h	Rx Buffer Start Address. Configures the start address of the Rx Buffers section in the Message RAM (32-bit word address).
1-0	RESERVED	R/W	0h	

38 MCAN_RXF1C Register (Offset = 70B0h) [Reset = 00000000h]

MCAN_RXF1C is shown in Figure 27-64 and described in Table 27-58.

Return to the Summary Table.

MCAN Rx FIFO 1 Configuration

Figure 27-64. MCAN_RXF1C Register

31	30	29	28	27	26	25	24
F1OM	F1WM						
R/WQ-0h	R/WQ-0h						
23	22	21	20	19	18	17	16
RESERVED	F1S						
R/W-0h	R/WQ-0h						
15	14	13	12	11	10	9	8
F1SA							
R/WQ-0h							
7	6	5	4	3	2	1	0
F1SA						RESERVED	
R/WQ-0h						R/W-0h	

Table 27-96. MCAN_RXF1C Register Field Descriptions

Bit	Field	Type	Reset	Description
31	F1OM	R/WQ	0h	FIFO 1 Operation Mode. FIFO 1 can be operated in blocking or in overwrite mode. 0 FIFO 1 blocking mode 1 FIFO 1 overwrite mode Qualified Write is possible only with CCCR.CCE='1' and CCCR.INIT='1'.
30-24	F1WM	R/WQ	0h	Rx FIFO 1 Watermark 0 Watermark interrupt disabled 1-64 Level for Rx FIFO 1 watermark interrupt (IR.RF1W) >64 Watermark interrupt disabled Qualified Write is possible only with CCCR.CCE='1' and CCCR.INIT='1'.
23	RESERVED	R/W	0h	
22-16	F1S	R/WQ	0h	Rx FIFO 1 Size. The Rx FIFO 1 elements are indexed from 0 to F1S - 1. 0 No Rx FIFO 1 1-64 Number of Rx FIFO 1 elements >64 Values greater than 64 are interpreted as 64 Qualified Write is possible only with CCCR.CCE='1' and CCCR.INIT='1'.
15-2	F1SA	R/WQ	0h	Rx FIFO 1 Start Address Start address of Rx FIFO 1 in Message RAM (32-bit word address).
1-0	RESERVED	R/W	0h	

39 MCAN_RXF1S Register (Offset = 70B4h) [Reset = 00000000h]

MCAN_RXF1S is shown in Figure 27-65 and described in Table 27-59.

Return to the Summary Table.

MCAN Rx FIFO 1 Status

Figure 27-65. MCAN_RXF1S Register

31	30	29	28	27	26	25	24
DMS		RESERVED				RF1L	F1F
R-0h		R-0h				R-0h	R-0h
23	22	21	20	19	18	17	16
RESERVED		F1PI					
R-0h		R-0h					
15	14	13	12	11	10	9	8
RESERVED		F1GI					
R-0h		R-0h					
7	6	5	4	3	2	1	0
RESERVED	F1FL						
R-0h	R-0h						

Table 27-98. MCAN_RXF1S Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	DMS	R	0h	Debug Message Status 00 Idle state, wait for reception of debug messages 01 Debug message A received 10 Debug messages A, B received 11 Debug messages A, B, C received
29-26	RESERVED	R	0h	
25	RF1L	R	0h	Rx FIFO 1 Message Lost. This bit is a copy of interrupt flag IR.RF1L. When IR.RF1L is reset, this bit is also reset. 0 No Rx FIFO 1 message lost 1 Rx FIFO 1 message lost, also set after write attempt to Rx FIFO 1 of size zero Note: Overwriting the oldest message when RXF1C.F1OM = '1' will not set this flag.
24	F1F	R	0h	Rx FIFO 1 Full 0 Rx FIFO 1 not full 1 Rx FIFO 1 full
23-22	RESERVED	R	0h	
21-16	F1PI	R	0h	Rx FIFO 1 Put Index. Rx FIFO 1 write index pointer, range 0 to 63.
15-14	RESERVED	R	0h	
13-8	F1GI	R	0h	Rx FIFO 1 Get Index. Rx FIFO 1 read index pointer, range 0 to 63.
7	RESERVED	R	0h	
6-0	F1FL	R	0h	Rx FIFO 1 Fill Level. Number of elements stored in Rx FIFO 1, range 0 to 64.

40 MCAN_RXF1A Register (Offset = 70B8h) [Reset = 00000000h]

MCAN_RXF1A is shown in Figure 27-66 and described in Table 27-60.

Return to the Summary Table.

MCAN Rx FIFO 1 Acknowledge

Figure 27-66. MCAN_RXF1A Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																										F1AI					
R/W-0h																										R/W-0h					

Table 27-100. MCAN_RXF1A Register Field Descriptions

Bit	Field	Type	Reset	Description
31-6	RESERVED	R/W	0h	
5-0	F1AI	R/W	0h	Rx FIFO 1 Acknowledge Index. After the Host has read a message or a sequence of messages from Rx FIFO 1 it has to write the buffer index of the last element read from Rx FIFO 1 to F1AI. This will set the Rx FIFO 1 Get Index RXF1S.F1GI to F1AI + 1 and update the FIFO 1 Fill Level RXF1S.F1FL.

41 MCAN_RXESC Register (Offset = 70BCh) [Reset = 00000000h]

MCAN_RXESC is shown in Figure 27-67 and described in Table 27-61.

Return to the Summary Table.

Configures the number of data bytes belonging to an Rx Buffer / Rx FIFO element. Data field sizes >8 bytes are intended for CAN FD operation only.

Figure 27-67. MCAN_RXESC Register

31	30	29	28	27	26	25	24
RESERVED							
R/W-0h							
23	22	21	20	19	18	17	16
RESERVED							
R/W-0h							
15	14	13	12	11	10	9	8
RESERVED					RBDS		
R/W-0h					R/WQ-0h		
7	6	5	4	3	2	1	0
RESERVED	F1DS			RESERVED	F0DS		
R/W-0h	R/WQ-0h			R/W-0h	R/WQ-0h		

Table 27-102. MCAN_RXESC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-11	RESERVED	R/W	0h	
10-8	RBDS	R/WQ	0h	Rx Buffer Data Field Size 000 8 byte data field 001 12 byte data field 010 16 byte data field 011 20 byte data field 100 24 byte data field 101 32 byte data field 110 48 byte data field 111 64 byte data field Note: In case the data field size of an accepted CAN frame exceeds the data field size configured for the matching Rx Buffer or Rx FIFO, only the number of bytes as configured by RXESC are stored to the Rx Buffer resp. Rx FIFO element. The rest of the frame's data field is ignored. Qualified Write is possible only with CCCR.CCE='1' and CCCR.INIT='1'.
7	RESERVED	R/W	0h	
6-4	F1DS	R/WQ	0h	Rx FIFO 1 Data Field Size 000 8 byte data field 001 12 byte data field 010 16 byte data field 011 20 byte data field 100 24 byte data field 101 32 byte data field 110 48 byte data field 111 64 byte data field Note: In case the data field size of an accepted CAN frame exceeds the data field size configured for the matching Rx Buffer or Rx FIFO, only the number of bytes as configured by RXESC are stored to the Rx Buffer resp. Rx FIFO element. The rest of the frame's data field is ignored. Qualified Write is possible only with CCCR.CCE='1' and CCCR.INIT='1'.

Table 27-102. MCAN_RXESC Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3	RESERVED	R/W	0h	
2-0	F0DS	R/WQ	0h	Rx FIFO 0 Data Field Size 000 8 byte data field 001 12 byte data field 010 16 byte data field 011 20 byte data field 100 24 byte data field 101 32 byte data field 110 48 byte data field 111 64 byte data field Note: In case the data field size of an accepted CAN frame exceeds the data field size configured for the matching Rx Buffer or Rx FIFO, only the number of bytes as configured by RXESC are stored to the Rx Buffer resp. Rx FIFO element. The rest of the frame's data field is ignored. Qualified Write is possible only with CCCR.CCE='1' and CCCR.INIT='1'.

42 MCAN_TXBC Register (Offset = 70C0h) [Reset = 00000000h]

MCAN_TXBC is shown in Figure 27-68 and described in Table 27-62.

Return to the Summary Table.

MCAN Tx Buffer Configuration

Figure 27-68. MCAN_TXBC Register

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

Table 27-104. MCAN_TXBC Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	
30	TFQM	R/WQ	0h	Tx FIFO/Queue Mode 0 Tx FIFO operation 1 Tx Queue operation Qualified Write is possible only with CCCR.CCE='1' and CCCR.INIT='1'.
29-24	TFQS	R/WQ	0h	Transmit FIFO/Queue Size 0 No Tx FIFO/Queue 1-32 Number of Tx Buffers used for Tx FIFO/Queue >32 Values greater than 32 are interpreted as 32 Note: Be aware that the sum of TFQS and NDTB may be not greater than 32. There is no check for erroneous configurations. The Tx Buffers section in the Message RAM starts with the dedicated Tx Buffers. Qualified Write is possible only with CCCR.CCE='1' and CCCR.INIT='1'.
23-22	RESERVED	R/W	0h	
21-16	NDTB	R/WQ	0h	Number of Dedicated Transmit Buffers 0 No Dedicated Tx Buffers 1-32 Number of Dedicated Tx Buffers >32 Values greater than 32 are interpreted as 32 Note: Be aware that the sum of TFQS and NDTB may be not greater than 32. There is no check for erroneous configurations. The Tx Buffers section in the Message RAM starts with the dedicated Tx Buffers. Qualified Write is possible only with CCCR.CCE='1' and CCCR.INIT='1'.
15-2	TBSA	R/WQ	0h	Tx Buffers Start Address. Start address of Tx Buffers section in Message RAM (32-bit word address). Qualified Write is possible only with CCCR.CCE='1' and CCCR.INIT='1'.

Table 27-104. MCAN_TXBC Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
1-0	RESERVED	R/W	0h	

43 MCAN_TXFQS Register (Offset = 70C4h) [Reset = 00000000h]

MCAN_TXFQS is shown in Figure 27-69 and described in Table 27-63.

Return to the Summary Table.

The Tx FIFO/Queue status is related to the pending Tx requests listed in register TXBRP. Therefore the effect of Add/Cancellation requests may be delayed due to a running Tx scan (TXBRP not yet updated).

Figure 27-69. MCAN_TXFQS Register

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

Table 27-106. MCAN_TXFQS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-22	RESERVED	R	0h	
21	TFQF	R	0h	Tx FIFO/Queue Full 0 Tx FIFO/Queue not full 1 Tx FIFO/Queue full
20-16	TFQP	R	0h	Tx FIFO/Queue Put Index. Tx FIFO/Queue write index pointer, range 0 to 31. Note: In case of mixed configurations where dedicated Tx Buffers are combined with a Tx FIFO or a Tx Queue, the Put and Get Indices indicate the number of the Tx Buffer starting with the first dedicated Tx Buffers. Example: For a configuration of 12 dedicated Tx Buffers and a Tx FIFO of 20 Buffers a Put Index of 15 points to the fourth buffer of the Tx FIFO.
15-13	RESERVED	R	0h	
12-8	TFGI	R	0h	Tx FIFO Get Index. Tx FIFO read index pointer, range 0 to 31. Read as zero when Tx Queue operation is configured (TXBC.TFQM = '1'). Note: In case of mixed configurations where dedicated Tx Buffers are combined with a Tx FIFO or a Tx Queue, the Put and Get Indices indicate the number of the Tx Buffer starting with the first dedicated Tx Buffers. Example: For a configuration of 12 dedicated Tx Buffers and a Tx FIFO of 20 Buffers a Put Index of 15 points to the fourth buffer of the Tx FIFO.
7-6	RESERVED	R	0h	
5-0	TFFL	R	0h	Tx FIFO Free Level. Number of consecutive free Tx FIFO elements starting from TFGI, range 0 to 32. Read as zero when Tx Queue operation is configured (TXBC.TFQM = '1').

44 MCAN_TXESC Register (Offset = 70C8h) [Reset = 00000000h]

MCAN_TXESC is shown in Figure 27-70 and described in Table 27-64.

Return to the Summary Table.

Configures the number of data bytes belonging to a Tx Buffer element. Data field sizes > 8 bytes are intended for CAN FD operation only.

Figure 27-70. MCAN_TXESC Register

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

Table 27-108. MCAN_TXESC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R/W	0h	
2-0	TBDS	R/WQ	0h	Tx Buffer Data Field Size 000 8 byte data field 001 12 byte data field 010 16 byte data field 011 20 byte data field 100 24 byte data field 101 32 byte data field 110 48 byte data field 111 64 byte data field Note: In case the data length code DLC of a Tx Buffer element is configured to a value higher than the Tx Buffer data field size TXESC.TBDS, the bytes not defined by the Tx Buffer are transmitted as "0xCC" (padding bytes). Qualified Write is possible only with CCCR.CCE='1' and CCCR.INIT='1'.

45 MCAN_TXBRP Register (Offset = 70CCh) [Reset = 00000000h]

MCAN_TXBRP is shown in Figure 27-71 and described in Table 27-65.

Return to the Summary Table.

MCAN Tx Buffer Request Pending

Figure 27-71. MCAN_TXBRP Register

31	30	29	28	27	26	25	24
TRP31	TRP30	TRP29	TRP28	TRP27	TRP26	TRP25	TRP24
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
23	22	21	20	19	18	17	16
TRP23	TRP22	TRP21	TRP20	TRP19	TRP18	TRP17	TRP16
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
15	14	13	12	11	10	9	8
TRP15	TRP14	TRP13	TRP12	TRP11	TRP10	TRP9	TRP8
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
7	6	5	4	3	2	1	0
TRP7	TRP6	TRP5	TRP4	TRP3	TRP2	TRP1	TRP0
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

Table 27-110. MCAN_TXBRP Register Field Descriptions

Bit	Field	Type	Reset	Description
31	TRP31	R	0h	Transmission Request Pending 31. See description for bit 0.
30	TRP30	R	0h	Transmission Request Pending 30. See description for bit 0.
29	TRP29	R	0h	Transmission Request Pending 29. See description for bit 0.
28	TRP28	R	0h	Transmission Request Pending 28. See description for bit 0.
27	TRP27	R	0h	Transmission Request Pending 27. See description for bit 0.
26	TRP26	R	0h	Transmission Request Pending 26. See description for bit 0.
25	TRP25	R	0h	Transmission Request Pending 25. See description for bit 0.
24	TRP24	R	0h	Transmission Request Pending 24. See description for bit 0.
23	TRP23	R	0h	Transmission Request Pending 23. See description for bit 0.
22	TRP22	R	0h	Transmission Request Pending 22. See description for bit 0.
21	TRP21	R	0h	Transmission Request Pending 21. See description for bit 0.
20	TRP20	R	0h	Transmission Request Pending 20. See description for bit 0.
19	TRP19	R	0h	Transmission Request Pending 19. See description for bit 0.
18	TRP18	R	0h	Transmission Request Pending 18. See description for bit 0.
17	TRP17	R	0h	Transmission Request Pending 17. See description for bit 0.
16	TRP16	R	0h	Transmission Request Pending 16. See description for bit 0.
15	TRP15	R	0h	Transmission Request Pending 15. See description for bit 0.
14	TRP14	R	0h	Transmission Request Pending 14. See description for bit 0.
13	TRP13	R	0h	Transmission Request Pending 13. See description for bit 0.
12	TRP12	R	0h	Transmission Request Pending 12. See description for bit 0.
11	TRP11	R	0h	Transmission Request Pending 11. See description for bit 0.
10	TRP10	R	0h	Transmission Request Pending 10. See description for bit 0.
9	TRP9	R	0h	Transmission Request Pending 9. See description for bit 0.
8	TRP8	R	0h	Transmission Request Pending 8. See description for bit 0.
7	TRP7	R	0h	Transmission Request Pending 7. See description for bit 0.

Table 27-110. MCAN_TXBRP Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
6	TRP6	R	0h	Transmission Request Pending 6. See description for bit 0.
5	TRP5	R	0h	Transmission Request Pending 5. See description for bit 0.
4	TRP4	R	0h	Transmission Request Pending 4. See description for bit 0.
3	TRP3	R	0h	Transmission Request Pending 3. See description for bit 0.
2	TRP2	R	0h	Transmission Request Pending 2. See description for bit 0.
1	TRP1	R	0h	Transmission Request Pending 1. See description for bit 0.
0	TRP0	R	0h	Transmission Request Pending 0. Each Tx Buffer has its own Transmission Request Pending bit. The bits are set via register TXBAR. The bits are reset after a requested transmission has completed or has been cancelled via register TXBCR. TXBRP bits are set only for those Tx Buffers configured via TXBC. After a TXBRP bit has been set, a Tx scan is started to check for the pending Tx request with the highest priority (Tx Buffer with lowest Message ID). A cancellation request resets the corresponding transmission request pending bit of register TXBRP. In case a transmission has already been started when a cancellation is requested, this is done at the end of the transmission, regardless whether the transmission was successful or not. The cancellation request bits are reset directly after the corresponding TXBRP bit has been reset. After a cancellation has been requested, a finished cancellation is signalled via TXBCF - after successful transmission together with the corresponding TXBTO bit - when the transmission has not yet been started at the point of cancellation - when the transmission has been aborted due to lost arbitration - when an error occurred during frame transmission In DAR mode all transmissions are automatically cancelled if they are not successful. The corresponding TXBCF bit is set for all unsuccessful transmissions. 0 No transmission request pending 1 Transmission request pending Note: TXBRP bits which are set while a Tx scan is in progress are not considered during this particular Tx scan. In case a cancellation is requested for such a Tx Buffer, this Add Request is cancelled immediately, the corresponding TXBRP bit is reset.

46 MCAN_TXBAR Register (Offset = 70D0h) [Reset = 00000000h]

MCAN_TXBAR is shown in Figure 27-72 and described in Table 27-66.

Return to the Summary Table.

MCAN Tx Buffer Add Request

Figure 27-72. MCAN_TXBAR Register

31	30	29	28	27	26	25	24
AR31	AR30	AR29	AR28	AR27	AR26	AR25	AR24
R/WQ-0h	R/WQ-0h	R/WQ-0h	R/WQ-0h	R/WQ-0h	R/WQ-0h	R/WQ-0h	R/WQ-0h
23	22	21	20	19	18	17	16
AR23	AR22	AR21	AR20	AR19	AR18	AR17	AR16
R/WQ-0h	R/WQ-0h	R/WQ-0h	R/WQ-0h	R/WQ-0h	R/WQ-0h	R/WQ-0h	R/WQ-0h
15	14	13	12	11	10	9	8
AR15	AR14	AR13	AR12	AR11	AR10	AR9	AR8
R/WQ-0h	R/WQ-0h	R/WQ-0h	R/WQ-0h	R/WQ-0h	R/WQ-0h	R/WQ-0h	R/WQ-0h
7	6	5	4	3	2	1	0
AR7	AR6	AR5	AR4	AR3	AR2	AR1	AR0
R/WQ-0h	R/WQ-0h	R/WQ-0h	R/WQ-0h	R/WQ-0h	R/WQ-0h	R/WQ-0h	R/WQ-0h

Table 27-112. MCAN_TXBAR Register Field Descriptions

Bit	Field	Type	Reset	Description
31	AR31	R/WQ	0h	Add Request 31. See description for bit 0.
30	AR30	R/WQ	0h	Add Request 30. See description for bit 0.
29	AR29	R/WQ	0h	Add Request 29. See description for bit 0.
28	AR28	R/WQ	0h	Add Request 28. See description for bit 0.
27	AR27	R/WQ	0h	Add Request 27. See description for bit 0.
26	AR26	R/WQ	0h	Add Request 26. See description for bit 0.
25	AR25	R/WQ	0h	Add Request 25. See description for bit 0.
24	AR24	R/WQ	0h	Add Request 24. See description for bit 0.
23	AR23	R/WQ	0h	Add Request 23. See description for bit 0.
22	AR22	R/WQ	0h	Add Request 22. See description for bit 0.
21	AR21	R/WQ	0h	Add Request 21. See description for bit 0.
20	AR20	R/WQ	0h	Add Request 20. See description for bit 0.
19	AR19	R/WQ	0h	Add Request 19. See description for bit 0.
18	AR18	R/WQ	0h	Add Request 18. See description for bit 0.
17	AR17	R/WQ	0h	Add Request 17. See description for bit 0.
16	AR16	R/WQ	0h	Add Request 16. See description for bit 0.
15	AR15	R/WQ	0h	Add Request 15. See description for bit 0.
14	AR14	R/WQ	0h	Add Request 14. See description for bit 0.
13	AR13	R/WQ	0h	Add Request 13. See description for bit 0.
12	AR12	R/WQ	0h	Add Request 12. See description for bit 0.
11	AR11	R/WQ	0h	Add Request 11. See description for bit 0.
10	AR10	R/WQ	0h	Add Request 10. See description for bit 0.
9	AR9	R/WQ	0h	Add Request 9. See description for bit 0.
8	AR8	R/WQ	0h	Add Request 8. See description for bit 0.
7	AR7	R/WQ	0h	Add Request 7. See description for bit 0.

Table 27-112. MCAN_TXBAR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
6	AR6	R/WQ	0h	Add Request 6. See description for bit 0.
5	AR5	R/WQ	0h	Add Request 5. See description for bit 0.
4	AR4	R/WQ	0h	Add Request 4. See description for bit 0.
3	AR3	R/WQ	0h	Add Request 3. See description for bit 0.
2	AR2	R/WQ	0h	Add Request 2. See description for bit 0.
1	AR1	R/WQ	0h	Add Request 1. See description for bit 0.
0	AR0	R/WQ	0h	Add Request 0. Each Tx Buffer has its own Add Request bit. Writing a '1' will set the corresponding Add Request bit writing a '0' has no impact. This enables the Host to set transmission requests for multiple Tx Buffers with one write to TXBAR. TXBAR bits are set only for those Tx Buffers configured via TXBC. When no Tx scan is running, the bits are reset immediately, else the bits remain set until the Tx scan process has completed. 0 No transmission request added 1 Transmission requested added Note: If an add request is applied for a Tx Buffer with pending transmission request (corresponding TXBRP bit already set), this add request is ignored. Qualified Write is possible only with CCCR.CCE='0'

47 MCAN_TXBCR Register (Offset = 70D4h) [Reset = 00000000h]

MCAN_TXBCR is shown in Figure 27-73 and described in Table 27-67.

Return to the Summary Table.

MCAN Tx Buffer Cancellation Request

Figure 27-73. MCAN_TXBCR Register

31	30	29	28	27	26	25	24
CR31	CR30	CR29	CR28	CR27	CR26	CR25	CR24
R/WQ-0h	R/WQ-0h	R/WQ-0h	R/WQ-0h	R/WQ-0h	R/WQ-0h	R/WQ-0h	R/WQ-0h
23	22	21	20	19	18	17	16
CR23	CR22	CR21	CR20	CR19	CR18	CR17	CR16
R/WQ-0h	R/WQ-0h	R/WQ-0h	R/WQ-0h	R/WQ-0h	R/WQ-0h	R/WQ-0h	R/WQ-0h
15	14	13	12	11	10	9	8
CR15	CR14	CR13	CR12	CR11	CR10	CR9	CR8
R/WQ-0h	R/WQ-0h	R/WQ-0h	R/WQ-0h	R/WQ-0h	R/WQ-0h	R/WQ-0h	R/WQ-0h
7	6	5	4	3	2	1	0
CR7	CR6	CR5	CR4	CR3	CR2	CR1	CR0
R/WQ-0h	R/WQ-0h	R/WQ-0h	R/WQ-0h	R/WQ-0h	R/WQ-0h	R/WQ-0h	R/WQ-0h

Table 27-114. MCAN_TXBCR Register Field Descriptions

Bit	Field	Type	Reset	Description
31	CR31	R/WQ	0h	Cancellation Request 31. See description for bit 0.
30	CR30	R/WQ	0h	Cancellation Request 30. See description for bit 0.
29	CR29	R/WQ	0h	Cancellation Request 29. See description for bit 0.
28	CR28	R/WQ	0h	Cancellation Request 28. See description for bit 0.
27	CR27	R/WQ	0h	Cancellation Request 27. See description for bit 0.
26	CR26	R/WQ	0h	Cancellation Request 26. See description for bit 0.
25	CR25	R/WQ	0h	Cancellation Request 25. See description for bit 0.
24	CR24	R/WQ	0h	Cancellation Request 24. See description for bit 0.
23	CR23	R/WQ	0h	Cancellation Request 23. See description for bit 0.
22	CR22	R/WQ	0h	Cancellation Request 22. See description for bit 0.
21	CR21	R/WQ	0h	Cancellation Request 21. See description for bit 0.
20	CR20	R/WQ	0h	Cancellation Request 20. See description for bit 0.
19	CR19	R/WQ	0h	Cancellation Request 19. See description for bit 0.
18	CR18	R/WQ	0h	Cancellation Request 18. See description for bit 0.
17	CR17	R/WQ	0h	Cancellation Request 17. See description for bit 0.
16	CR16	R/WQ	0h	Cancellation Request 16. See description for bit 0.
15	CR15	R/WQ	0h	Cancellation Request 15. See description for bit 0.
14	CR14	R/WQ	0h	Cancellation Request 14. See description for bit 0.
13	CR13	R/WQ	0h	Cancellation Request 13. See description for bit 0.
12	CR12	R/WQ	0h	Cancellation Request 12. See description for bit 0.
11	CR11	R/WQ	0h	Cancellation Request 11. See description for bit 0.
10	CR10	R/WQ	0h	Cancellation Request 10. See description for bit 0.
9	CR9	R/WQ	0h	Cancellation Request 9. See description for bit 0.
8	CR8	R/WQ	0h	Cancellation Request 8. See description for bit 0.
7	CR7	R/WQ	0h	Cancellation Request 7. See description for bit 0.

Table 27-114. MCAN_TXBCR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
6	CR6	R/WQ	0h	Cancellation Request 6. See description for bit 0.
5	CR5	R/WQ	0h	Cancellation Request 5. See description for bit 0.
4	CR4	R/WQ	0h	Cancellation Request 4. See description for bit 0.
3	CR3	R/WQ	0h	Cancellation Request 3. See description for bit 0.
2	CR2	R/WQ	0h	Cancellation Request 2. See description for bit 0.
1	CR1	R/WQ	0h	Cancellation Request 1. See description for bit 0.
0	CR0	R/WQ	0h	Cancellation Request 0. Each Tx Buffer has its own Cancellation Request bit. Writing a '1' will set the corresponding Cancellation Request bit writing a '0' has no impact. This enables the Host to set cancellation requests for multiple Tx Buffers with one write to TXBCR. TXBCR bits are set only for those Tx Buffers configured via TXBC. The bits remain set until the corresponding bit of TXBRP is reset. 0 No cancellation pending 1 Cancellation pending Qualified Write is possible only with CCCR.CCE='0'

48 MCAN_TXBTO Register (Offset = 70D8h) [Reset = 00000000h]

MCAN_TXBTO is shown in Figure 27-74 and described in Table 27-68.

Return to the Summary Table.

MCAN Tx Buffer Transmission Occurred

Figure 27-74. MCAN_TXBTO Register

31	30	29	28	27	26	25	24
TO31	TO30	TO29	TO28	TO27	TO26	TO25	TO24
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
23	22	21	20	19	18	17	16
TO23	TO22	TO21	TO20	TO19	TO18	TO17	TO16
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
15	14	13	12	11	10	9	8
TO15	TO14	TO13	TO12	TO11	TO10	TO9	TO8
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
7	6	5	4	3	2	1	0
TO7	TO6	TO5	TO4	TO3	TO2	TO1	TO0
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

Table 27-116. MCAN_TXBTO Register Field Descriptions

Bit	Field	Type	Reset	Description
31	TO31	R	0h	Transmission Occurred 31. See description for bit 0.
30	TO30	R	0h	Transmission Occurred 30. See description for bit 0.
29	TO29	R	0h	Transmission Occurred 29. See description for bit 0.
28	TO28	R	0h	Transmission Occurred 28. See description for bit 0.
27	TO27	R	0h	Transmission Occurred 27. See description for bit 0.
26	TO26	R	0h	Transmission Occurred 26. See description for bit 0.
25	TO25	R	0h	Transmission Occurred 25. See description for bit 0.
24	TO24	R	0h	Transmission Occurred 24. See description for bit 0.
23	TO23	R	0h	Transmission Occurred 23. See description for bit 0.
22	TO22	R	0h	Transmission Occurred 22. See description for bit 0.
21	TO21	R	0h	Transmission Occurred 21. See description for bit 0.
20	TO20	R	0h	Transmission Occurred 20. See description for bit 0.
19	TO19	R	0h	Transmission Occurred 19. See description for bit 0.
18	TO18	R	0h	Transmission Occurred 18. See description for bit 0.
17	TO17	R	0h	Transmission Occurred 17. See description for bit 0.
16	TO16	R	0h	Transmission Occurred 16. See description for bit 0.
15	TO15	R	0h	Transmission Occurred 15. See description for bit 0.
14	TO14	R	0h	Transmission Occurred 14. See description for bit 0.
13	TO13	R	0h	Transmission Occurred 13. See description for bit 0.
12	TO12	R	0h	Transmission Occurred 12. See description for bit 0.
11	TO11	R	0h	Transmission Occurred 11. See description for bit 0.
10	TO10	R	0h	Transmission Occurred 10. See description for bit 0.
9	TO9	R	0h	Transmission Occurred 9. See description for bit 0.
8	TO8	R	0h	Transmission Occurred 8. See description for bit 0.
7	TO7	R	0h	Transmission Occurred 7. See description for bit 0.

Table 27-116. MCAN_TXBTO Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
6	TO6	R	0h	Transmission Occurred 6. See description for bit 0.
5	TO5	R	0h	Transmission Occurred 5. See description for bit 0.
4	TO4	R	0h	Transmission Occurred 4. See description for bit 0.
3	TO3	R	0h	Transmission Occurred 3. See description for bit 0.
2	TO2	R	0h	Transmission Occurred 2. See description for bit 0.
1	TO1	R	0h	Transmission Occurred 1. See description for bit 0.
0	TO0	R	0h	Transmission Occurred 0. Each Tx Buffer has its own Transmission Occurred bit. The bits are set when the corresponding TXBRP bit is cleared after a successful transmission. The bits are reset when a new transmission is requested by writing a '1' to the corresponding bit of register TXBAR. 0 No transmission occurred 1 Transmission occurred

49 MCAN_TXBCF Register (Offset = 70DCh) [Reset = 00000000h]

MCAN_TXBCF is shown in Figure 27-75 and described in Table 27-69.

Return to the Summary Table.

MCAN Tx Buffer Cancellation Finished

Figure 27-75. MCAN_TXBCF Register

31	30	29	28	27	26	25	24
CF31	CF30	CF29	CF28	CF27	CF26	CF25	CF24
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
23	22	21	20	19	18	17	16
CF23	CF22	CF21	CF20	CF19	CF18	CF17	CF16
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
15	14	13	12	11	10	9	8
CF15	CF14	CF13	CF12	CF11	CF10	CF9	CF8
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
7	6	5	4	3	2	1	0
CF7	CF6	CF5	CF4	CF3	CF2	CF1	CF0
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

Table 27-118. MCAN_TXBCF Register Field Descriptions

Bit	Field	Type	Reset	Description
31	CF31	R	0h	Cancellation Finished 31. See description for bit 0.
30	CF30	R	0h	Cancellation Finished 30. See description for bit 0.
29	CF29	R	0h	Cancellation Finished 29. See description for bit 0.
28	CF28	R	0h	Cancellation Finished 28. See description for bit 0.
27	CF27	R	0h	Cancellation Finished 27. See description for bit 0.
26	CF26	R	0h	Cancellation Finished 26. See description for bit 0.
25	CF25	R	0h	Cancellation Finished 25. See description for bit 0.
24	CF24	R	0h	Cancellation Finished 24. See description for bit 0.
23	CF23	R	0h	Cancellation Finished 23. See description for bit 0.
22	CF22	R	0h	Cancellation Finished 22. See description for bit 0.
21	CF21	R	0h	Cancellation Finished 21. See description for bit 0.
20	CF20	R	0h	Cancellation Finished 20. See description for bit 0.
19	CF19	R	0h	Cancellation Finished 19. See description for bit 0.
18	CF18	R	0h	Cancellation Finished 18. See description for bit 0.
17	CF17	R	0h	Cancellation Finished 17. See description for bit 0.
16	CF16	R	0h	Cancellation Finished 16. See description for bit 0.
15	CF15	R	0h	Cancellation Finished 15. See description for bit 0.
14	CF14	R	0h	Cancellation Finished 14. See description for bit 0.
13	CF13	R	0h	Cancellation Finished 13. See description for bit 0.
12	CF12	R	0h	Cancellation Finished 12. See description for bit 0.
11	CF11	R	0h	Cancellation Finished 11. See description for bit 0.
10	CF10	R	0h	Cancellation Finished 10. See description for bit 0.
9	CF9	R	0h	Cancellation Finished 9. See description for bit 0.
8	CF8	R	0h	Cancellation Finished 8. See description for bit 0.
7	CF7	R	0h	Cancellation Finished 7. See description for bit 0.

Table 27-118. MCAN_TXBCF Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
6	CF6	R	0h	Cancellation Finished 6. See description for bit 0.
5	CF5	R	0h	Cancellation Finished 5. See description for bit 0.
4	CF4	R	0h	Cancellation Finished 4. See description for bit 0.
3	CF3	R	0h	Cancellation Finished 3. See description for bit 0.
2	CF2	R	0h	Cancellation Finished 2. See description for bit 0.
1	CF1	R	0h	Cancellation Finished 1. See description for bit 0.
0	CF0	R	0h	Cancellation Finished 0. Each Tx Buffer has its own Cancellation Finished bit. The bits are set when the corresponding TXBRP bit is cleared after a cancellation was requested via TXBCR. In case the corresponding TXBRP bit was not set at the point of cancellation, CF is set immediately. The bits are reset when a new transmission is requested by writing a '1' to the corresponding bit of register TXBAR. 0 No transmit buffer cancellation 1 Transmit buffer cancellation finished

50 MCAN_TXBTIE Register (Offset = 70E0h) [Reset = 00000000h]

MCAN_TXBTIE is shown in Figure 27-76 and described in Table 27-70.

Return to the Summary Table.

MCAN Tx Buffer Transmission Interrupt Enable

Figure 27-76. MCAN_TXBTIE Register

31	30	29	28	27	26	25	24
TIE31	TIE30	TIE29	TIE28	TIE27	TIE26	TIE25	TIE24
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
TIE23	TIE22	TIE21	TIE20	TIE19	TIE18	TIE17	TIE16
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
TIE15	TIE14	TIE13	TIE12	TIE11	TIE10	TIE9	TIE8
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
TIE7	TIE6	TIE5	TIE4	TIE3	TIE2	TIE1	TIE0
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 27-120. MCAN_TXBTIE Register Field Descriptions

Bit	Field	Type	Reset	Description
31	TIE31	R/W	0h	Transmission Interrupt Enable 31. Each Tx Buffer has its own Transmission Interrupt Enable bit. 0 Transmission interrupt disabled 1 Transmission interrupt enable
30	TIE30	R/W	0h	Transmission Interrupt Enable 30. Each Tx Buffer has its own Transmission Interrupt Enable bit. 0 Transmission interrupt disabled 1 Transmission interrupt enable
29	TIE29	R/W	0h	Transmission Interrupt Enable 29. Each Tx Buffer has its own Transmission Interrupt Enable bit. 0 Transmission interrupt disabled 1 Transmission interrupt enable
28	TIE28	R/W	0h	Transmission Interrupt Enable 28. Each Tx Buffer has its own Transmission Interrupt Enable bit. 0 Transmission interrupt disabled 1 Transmission interrupt enable
27	TIE27	R/W	0h	Transmission Interrupt Enable 27. Each Tx Buffer has its own Transmission Interrupt Enable bit. 0 Transmission interrupt disabled 1 Transmission interrupt enable
26	TIE26	R/W	0h	Transmission Interrupt Enable 26. Each Tx Buffer has its own Transmission Interrupt Enable bit. 0 Transmission interrupt disabled 1 Transmission interrupt enable
25	TIE25	R/W	0h	Transmission Interrupt Enable 25. Each Tx Buffer has its own Transmission Interrupt Enable bit. 0 Transmission interrupt disabled 1 Transmission interrupt enable
24	TIE24	R/W	0h	Transmission Interrupt Enable 24. Each Tx Buffer has its own Transmission Interrupt Enable bit. 0 Transmission interrupt disabled 1 Transmission interrupt enable

Table 27-120. MCAN_TXBTIE Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
23	TIE23	R/W	0h	Transmission Interrupt Enable 23. Each Tx Buffer has its own Transmission Interrupt Enable bit. 0 Transmission interrupt disabled 1 Transmission interrupt enable
22	TIE22	R/W	0h	Transmission Interrupt Enable 22. Each Tx Buffer has its own Transmission Interrupt Enable bit. 0 Transmission interrupt disabled 1 Transmission interrupt enable
21	TIE21	R/W	0h	Transmission Interrupt Enable 21. Each Tx Buffer has its own Transmission Interrupt Enable bit. 0 Transmission interrupt disabled 1 Transmission interrupt enable
20	TIE20	R/W	0h	Transmission Interrupt Enable 20. Each Tx Buffer has its own Transmission Interrupt Enable bit. 0 Transmission interrupt disabled 1 Transmission interrupt enable
19	TIE19	R/W	0h	Transmission Interrupt Enable 19. Each Tx Buffer has its own Transmission Interrupt Enable bit. 0 Transmission interrupt disabled 1 Transmission interrupt enable
18	TIE18	R/W	0h	Transmission Interrupt Enable 18. Each Tx Buffer has its own Transmission Interrupt Enable bit. 0 Transmission interrupt disabled 1 Transmission interrupt enable
17	TIE17	R/W	0h	Transmission Interrupt Enable 17. Each Tx Buffer has its own Transmission Interrupt Enable bit. 0 Transmission interrupt disabled 1 Transmission interrupt enable
16	TIE16	R/W	0h	Transmission Interrupt Enable 16. Each Tx Buffer has its own Transmission Interrupt Enable bit. 0 Transmission interrupt disabled 1 Transmission interrupt enable
15	TIE15	R/W	0h	Transmission Interrupt Enable 15. Each Tx Buffer has its own Transmission Interrupt Enable bit. 0 Transmission interrupt disabled 1 Transmission interrupt enable
14	TIE14	R/W	0h	Transmission Interrupt Enable 14. Each Tx Buffer has its own Transmission Interrupt Enable bit. 0 Transmission interrupt disabled 1 Transmission interrupt enable
13	TIE13	R/W	0h	Transmission Interrupt Enable 13. Each Tx Buffer has its own Transmission Interrupt Enable bit. 0 Transmission interrupt disabled 1 Transmission interrupt enable
12	TIE12	R/W	0h	Transmission Interrupt Enable 12. Each Tx Buffer has its own Transmission Interrupt Enable bit. 0 Transmission interrupt disabled 1 Transmission interrupt enable
11	TIE11	R/W	0h	Transmission Interrupt Enable 11. Each Tx Buffer has its own Transmission Interrupt Enable bit. 0 Transmission interrupt disabled 1 Transmission interrupt enable
10	TIE10	R/W	0h	Transmission Interrupt Enable 10. Each Tx Buffer has its own Transmission Interrupt Enable bit. 0 Transmission interrupt disabled 1 Transmission interrupt enable

Table 27-120. MCAN_TXBTIE Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
9	TIE9	R/W	0h	Transmission Interrupt Enable 9. Each Tx Buffer has its own Transmission Interrupt Enable bit. 0 Transmission interrupt disabled 1 Transmission interrupt enable
8	TIE8	R/W	0h	Transmission Interrupt Enable 8. Each Tx Buffer has its own Transmission Interrupt Enable bit. 0 Transmission interrupt disabled 1 Transmission interrupt enable
7	TIE7	R/W	0h	Transmission Interrupt Enable 7. Each Tx Buffer has its own Transmission Interrupt Enable bit. 0 Transmission interrupt disabled 1 Transmission interrupt enable
6	TIE6	R/W	0h	Transmission Interrupt Enable 6. Each Tx Buffer has its own Transmission Interrupt Enable bit. 0 Transmission interrupt disabled 1 Transmission interrupt enable
5	TIE5	R/W	0h	Transmission Interrupt Enable 5. Each Tx Buffer has its own Transmission Interrupt Enable bit. 0 Transmission interrupt disabled 1 Transmission interrupt enable
4	TIE4	R/W	0h	Transmission Interrupt Enable 4. Each Tx Buffer has its own Transmission Interrupt Enable bit. 0 Transmission interrupt disabled 1 Transmission interrupt enable
3	TIE3	R/W	0h	Transmission Interrupt Enable 3. Each Tx Buffer has its own Transmission Interrupt Enable bit. 0 Transmission interrupt disabled 1 Transmission interrupt enable
2	TIE2	R/W	0h	Transmission Interrupt Enable 2. Each Tx Buffer has its own Transmission Interrupt Enable bit. 0 Transmission interrupt disabled 1 Transmission interrupt enable
1	TIE1	R/W	0h	Transmission Interrupt Enable 1. Each Tx Buffer has its own Transmission Interrupt Enable bit. 0 Transmission interrupt disabled 1 Transmission interrupt enable
0	TIE0	R/W	0h	Transmission Interrupt Enable 0. Each Tx Buffer has its own Transmission Interrupt Enable bit. 0 Transmission interrupt disabled 1 Transmission interrupt enable

51 MCAN_TXBCIE Register (Offset = 70E4h) [Reset = 00000000h]

MCAN_TXBCIE is shown in Figure 27-77 and described in Table 27-71.

Return to the Summary Table.

MCAN Tx Buffer Cancellation Finished Interrupt Enable

Figure 27-77. MCAN_TXBCIE Register

31	30	29	28	27	26	25	24
CFIE31	CFIE30	CFIE29	CFIE28	CFIE27	CFIE26	CFIE25	CFIE24
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
CFIE23	CFIE22	CFIE21	CFIE20	CFIE19	CFIE18	CFIE17	CFIE16
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
CFIE15	CFIE14	CFIE13	CFIE12	CFIE11	CFIE10	CFIE9	CFIE8
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
CFIE7	CFIE6	CFIE5	CFIE4	CFIE3	CFIE2	CFIE1	CFIE0
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 27-122. MCAN_TXBCIE Register Field Descriptions

Bit	Field	Type	Reset	Description
31	CFIE31	R/W	0h	Cancellation Finished Interrupt Enable 31. Each Tx Buffer has its own Cancellation Finished Interrupt Enable bit. 0 Cancellation finished interrupt disabled 1 Cancellation finished interrupt enabled
30	CFIE30	R/W	0h	Cancellation Finished Interrupt Enable 30. Each Tx Buffer has its own Cancellation Finished Interrupt Enable bit. 0 Cancellation finished interrupt disabled 1 Cancellation finished interrupt enabled
29	CFIE29	R/W	0h	Cancellation Finished Interrupt Enable 29. Each Tx Buffer has its own Cancellation Finished Interrupt Enable bit. 0 Cancellation finished interrupt disabled 1 Cancellation finished interrupt enabled
28	CFIE28	R/W	0h	Cancellation Finished Interrupt Enable 28. Each Tx Buffer has its own Cancellation Finished Interrupt Enable bit. 0 Cancellation finished interrupt disabled 1 Cancellation finished interrupt enabled
27	CFIE27	R/W	0h	Cancellation Finished Interrupt Enable 27. Each Tx Buffer has its own Cancellation Finished Interrupt Enable bit. 0 Cancellation finished interrupt disabled 1 Cancellation finished interrupt enabled
26	CFIE26	R/W	0h	Cancellation Finished Interrupt Enable 26. Each Tx Buffer has its own Cancellation Finished Interrupt Enable bit. 0 Cancellation finished interrupt disabled 1 Cancellation finished interrupt enabled
25	CFIE25	R/W	0h	Cancellation Finished Interrupt Enable 25. Each Tx Buffer has its own Cancellation Finished Interrupt Enable bit. 0 Cancellation finished interrupt disabled 1 Cancellation finished interrupt enabled
24	CFIE24	R/W	0h	Cancellation Finished Interrupt Enable 24. Each Tx Buffer has its own Cancellation Finished Interrupt Enable bit. 0 Cancellation finished interrupt disabled 1 Cancellation finished interrupt enabled

Table 27-122. MCAN_TXBCIE Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
23	CFIE23	R/W	0h	Cancellation Finished Interrupt Enable 23. Each Tx Buffer has its own Cancellation Finished Interrupt Enable bit. 0 Cancellation finished interrupt disabled 1 Cancellation finished interrupt enabled
22	CFIE22	R/W	0h	Cancellation Finished Interrupt Enable 22. Each Tx Buffer has its own Cancellation Finished Interrupt Enable bit. 0 Cancellation finished interrupt disabled 1 Cancellation finished interrupt enabled
21	CFIE21	R/W	0h	Cancellation Finished Interrupt Enable 21. Each Tx Buffer has its own Cancellation Finished Interrupt Enable bit. 0 Cancellation finished interrupt disabled 1 Cancellation finished interrupt enabled
20	CFIE20	R/W	0h	Cancellation Finished Interrupt Enable 20. Each Tx Buffer has its own Cancellation Finished Interrupt Enable bit. 0 Cancellation finished interrupt disabled 1 Cancellation finished interrupt enabled
19	CFIE19	R/W	0h	Cancellation Finished Interrupt Enable 19. Each Tx Buffer has its own Cancellation Finished Interrupt Enable bit. 0 Cancellation finished interrupt disabled 1 Cancellation finished interrupt enabled
18	CFIE18	R/W	0h	Cancellation Finished Interrupt Enable 18. Each Tx Buffer has its own Cancellation Finished Interrupt Enable bit. 0 Cancellation finished interrupt disabled 1 Cancellation finished interrupt enabled
17	CFIE17	R/W	0h	Cancellation Finished Interrupt Enable 17. Each Tx Buffer has its own Cancellation Finished Interrupt Enable bit. 0 Cancellation finished interrupt disabled 1 Cancellation finished interrupt enabled
16	CFIE16	R/W	0h	Cancellation Finished Interrupt Enable 16. Each Tx Buffer has its own Cancellation Finished Interrupt Enable bit. 0 Cancellation finished interrupt disabled 1 Cancellation finished interrupt enabled
15	CFIE15	R/W	0h	Cancellation Finished Interrupt Enable 15. Each Tx Buffer has its own Cancellation Finished Interrupt Enable bit. 0 Cancellation finished interrupt disabled 1 Cancellation finished interrupt enabled
14	CFIE14	R/W	0h	Cancellation Finished Interrupt Enable 14. Each Tx Buffer has its own Cancellation Finished Interrupt Enable bit. 0 Cancellation finished interrupt disabled 1 Cancellation finished interrupt enabled
13	CFIE13	R/W	0h	Cancellation Finished Interrupt Enable 13. Each Tx Buffer has its own Cancellation Finished Interrupt Enable bit. 0 Cancellation finished interrupt disabled 1 Cancellation finished interrupt enabled
12	CFIE12	R/W	0h	Cancellation Finished Interrupt Enable 12. Each Tx Buffer has its own Cancellation Finished Interrupt Enable bit. 0 Cancellation finished interrupt disabled 1 Cancellation finished interrupt enabled
11	CFIE11	R/W	0h	Cancellation Finished Interrupt Enable 11. Each Tx Buffer has its own Cancellation Finished Interrupt Enable bit. 0 Cancellation finished interrupt disabled 1 Cancellation finished interrupt enabled
10	CFIE10	R/W	0h	Cancellation Finished Interrupt Enable 10. Each Tx Buffer has its own Cancellation Finished Interrupt Enable bit. 0 Cancellation finished interrupt disabled 1 Cancellation finished interrupt enabled

Table 27-122. MCAN_TXBCIE Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
9	CFIE9	R/W	0h	Cancellation Finished Interrupt Enable 9. Each Tx Buffer has its own Cancellation Finished Interrupt Enable bit. 0 Cancellation finished interrupt disabled 1 Cancellation finished interrupt enabled
8	CFIE8	R/W	0h	Cancellation Finished Interrupt Enable 8. Each Tx Buffer has its own Cancellation Finished Interrupt Enable bit. 0 Cancellation finished interrupt disabled 1 Cancellation finished interrupt enabled
7	CFIE7	R/W	0h	Cancellation Finished Interrupt Enable 7. Each Tx Buffer has its own Cancellation Finished Interrupt Enable bit. 0 Cancellation finished interrupt disabled 1 Cancellation finished interrupt enabled
6	CFIE6	R/W	0h	Cancellation Finished Interrupt Enable 6. Each Tx Buffer has its own Cancellation Finished Interrupt Enable bit. 0 Cancellation finished interrupt disabled 1 Cancellation finished interrupt enabled
5	CFIE5	R/W	0h	Cancellation Finished Interrupt Enable 5. Each Tx Buffer has its own Cancellation Finished Interrupt Enable bit. 0 Cancellation finished interrupt disabled 1 Cancellation finished interrupt enabled
4	CFIE4	R/W	0h	Cancellation Finished Interrupt Enable 4. Each Tx Buffer has its own Cancellation Finished Interrupt Enable bit. 0 Cancellation finished interrupt disabled 1 Cancellation finished interrupt enabled
3	CFIE3	R/W	0h	Cancellation Finished Interrupt Enable 3. Each Tx Buffer has its own Cancellation Finished Interrupt Enable bit. 0 Cancellation finished interrupt disabled 1 Cancellation finished interrupt enabled
2	CFIE2	R/W	0h	Cancellation Finished Interrupt Enable 2. Each Tx Buffer has its own Cancellation Finished Interrupt Enable bit. 0 Cancellation finished interrupt disabled 1 Cancellation finished interrupt enabled
1	CFIE1	R/W	0h	Cancellation Finished Interrupt Enable 1. Each Tx Buffer has its own Cancellation Finished Interrupt Enable bit. 0 Cancellation finished interrupt disabled 1 Cancellation finished interrupt enabled
0	CFIE0	R/W	0h	Cancellation Finished Interrupt Enable 0. Each Tx Buffer has its own Cancellation Finished Interrupt Enable bit. 0 Cancellation finished interrupt disabled 1 Cancellation finished interrupt enabled

52 MCAN_TXEFC Register (Offset = 70F0h) [Reset = 00000000h]

MCAN_TXEFC is shown in Figure 27-78 and described in Table 27-72.

Return to the Summary Table.

MCAN Tx Event FIFO Configuration

Figure 27-78. MCAN_TXEFC Register

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

Table 27-124. MCAN_TXEFC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	RESERVED	R/W	0h	
29-24	EFWM	R/WQ	0h	Event FIFO Watermark 0 Watermark interrupt disabled 1-32 Level for Tx Event FIFO watermark interrupt (IR.TEFW) >32 Watermark interrupt disabled
23-22	RESERVED	R/W	0h	
21-16	EFS	R/WQ	0h	Event FIFO Size. The Tx Event FIFO elements are indexed from 0 to EFS - 1. 0 Tx Event FIFO disabled 1-32 Number of Tx Event FIFO elements >32 Values greater than 32 are interpreted as 32
15-2	EFSA	R/WQ	0h	Event FIFO Start Address. Start address of Tx Event FIFO in Message RAM (32-bit word address).
1-0	RESERVED	R/W	0h	

53 MCAN_TXEFS Register (Offset = 70F4h) [Reset = 00000000h]

MCAN_TXEFS is shown in Figure 27-79 and described in Table 27-73.

Return to the Summary Table.

MCAN Tx Event FIFO Status

Figure 27-79. MCAN_TXEFS Register

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

Table 27-126. MCAN_TXEFS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	RESERVED	R	0h	
25	TEFL	R	0h	Tx Event FIFO Element Lost. This bit is a copy of interrupt flag IR.TEFL. When IR.TEFL is reset, this bit is also reset. 0 No Tx Event FIFO element lost 1 Tx Event FIFO element lost, also set after write attempt to Tx Event FIFO of size zero.
24	EFF	R	0h	Event FIFO Full 0 Tx Event FIFO not full 1 Tx Event FIFO full
23-21	RESERVED	R	0h	
20-16	EFPI	R	0h	Event FIFO Put Index.Tx Event FIFO write index pointer, range 0 to 31.
15-13	RESERVED	R	0h	
12-8	EFGI	R	0h	Event FIFO Get Index. Tx Event FIFO read index pointer, range 0 to 31.
7-6	RESERVED	R	0h	
5-0	EFFL	R	0h	Event FIFO Fill Level. Number of elements stored in Tx Event FIFO, range 0 to 32.

54 MCAN_TXEFA Register (Offset = 70F8h) [Reset = 00000000h]

MCAN_TXEFA is shown in Figure 27-80 and described in Table 27-74.

Return to the Summary Table.

MCAN Tx Event FIFO Acknowledge

Figure 27-80. MCAN_TXEFA Register

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

Table 27-128. MCAN_TXEFA Register Field Descriptions

Bit	Field	Type	Reset	Description
31-5	RESERVED	R/W	0h	
4-0	EFAI	R/W	0h	Event FIFO Acknowledge Index. After the Host has read an element or a sequence of elements from the Tx Event FIFO it has to write the index of the last element read from Tx Event FIFO to EFAI. This will set the Tx Event FIFO Get Index TXEFS.EFGI to EFAI + 1 and update the Event FIFO Fill Level TXEFS.EFFL.

55 MCANSS_PID Register (Offset = 7200h) [Reset = 68E04901h]

MCANSS_PID is shown in Figure 27-81 and described in Table 27-75.

Return to the Summary Table.

MCAN Subsystem Revision Register

Figure 27-81. MCANSS_PID Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SCHEME		RESERVED		MODULE_ID											
R-1h		R-2h		R-8E0h											
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED					MAJOR			RESERVED		MINOR					
R-9h					R-1h			R-0h		R-1h					

Table 27-130. MCANSS_PID Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	SCHEME	R	1h	PID Register Scheme
29-28	RESERVED	R	2h	Reserved
27-16	MODULE_ID	R	8E0h	Module Identification Number
15-11	RESERVED	R	9h	Reserved
10-8	MAJOR	R	1h	Major Revision of the MCAN Subsystem
7-6	RESERVED	R	0h	Reserved
5-0	MINOR	R	1h	Minor Revision of the MCAN Subsystem

56 MCANSS_CTRL Register (Offset = 7204h) [Reset = 00000008h]

MCANSS_CTRL is shown in Figure 27-82 and described in Table 27-76.

Return to the Summary Table.

MCAN Subsystem Control Register

Figure 27-82. MCANSS_CTRL Register

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

Table 27-132. MCANSS_CTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-7	RESERVED	R/W	0h	
6	EXT_TS_CNTR_EN	R/W	0h	External Timestamp Counter Enable. 0 External timestamp counter disabled 1 External timestamp counter enabled
5	AUTOWAKEUP	R/W	0h	Automatic Wakeup Enable. Enables the MCANSS to automatically clear the MCAN CCCR.INIT bit, fully waking the MCAN up, on an enabled wakeup request. 0 Disable the automatic write to CCCR.INIT 1 Enable the automatic write to CCCR.INIT
4	WAKEUPREQEN	R/W	0h	Wakeup Request Enable. Enables the MCANSS to wakeup on CAN RXD activity. 0 Disable wakeup request 1 Enables wakeup request
3	DBGSUSP_FREE	R/W	1h	Debug Suspend Free Bit. Enables debug suspend. 0 Honor debug suspend 1 Disregard debug suspend
2-0	RESERVED	R/W	0h	

57 MCANSS_STAT Register (Offset = 7208h) [Reset = 0000000Xh]

MCANSS_STAT is shown in Figure 27-83 and described in Table 27-77.

Return to the Summary Table.

MCAN Subsystem Status Register

Figure 27-83. MCANSS_STAT Register

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

Table 27-134. MCANSS_STAT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R	0h	
2	ENABLE_FDOE	R	Xh	Reflects the value of mcanss_enable_fdoe configuration port
1	MEM_INIT_DONE	R	0h	Memory Initialization Done. 0 Message RAM initialization is in progress 1 Message RAM is initialized for use
0	RESET	R	0h	Soft Reset Status. 0 Not in reset 1 Reset is in progress

58 MCANSS_ICS Register (Offset = 720Ch) [Reset = 00000000h]

MCANSS_ICS is shown in Figure 27-84 and described in Table 27-78.

Return to the Summary Table.

MCAN Subsystem Interrupt Clear Shadow Register

Figure 27-84. MCANSS_ICS Register

31	30	29	28	27	26	25	24
RESERVED							
R/W-0h							
23	22	21	20	19	18	17	16
RESERVED							
R/W-0h							
15	14	13	12	11	10	9	8
RESERVED							
R/W-0h							
7	6	5	4	3	2	1	0
RESERVED							EXT_TS_CNTR_OVFL
R/W-0h							R-0/W1C-0h

Table 27-136. MCANSS_ICS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R/W	0h	
0	EXT_TS_CNTR_OVFL	R-0/W1C	0h	External Timestamp Counter Overflow Interrupt Status Clear. Reads always return a 0. 0 Write of '0' has no effect 1 Write of '1' clears the MCANSS_IRS.EXT_TS_CNTR_OVFL bit

ADVANCE INFORMATION

59 MCANSS_IRS Register (Offset = 7210h) [Reset = 00000000h]

MCANSS_IRS is shown in Figure 27-85 and described in Table 27-79.

Return to the Summary Table.

MCAN Subsystem Interrupt Raw Satus Register

Figure 27-85. MCANSS_IRS Register

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

Table 27-138. MCANSS_IRS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R/W	0h	
0	EXT_TS_CNTR_OVFL	R/W1S	0h	External Timestamp Counter Overflow Interrupt Status. This bit is set by HW or by a SW write of '1'. To clear, use the MCANSS_ICS.EXT_TS_CNTR_OVFL bit. 0 External timestamp counter has not overflowed 1 External timestamp counter has overflowed When this bit is set to '1' by HW or SW, the MCANSS_EXT_TS_UNSERVICED_INTR_CNTR.EXT_TS_INTR_CNTR bit field will increment by 1.

ADVANCE INFORMATION

60 MCANSS_IECS Register (Offset = 7214h) [Reset = 00000000h]

MCANSS_IECS is shown in Figure 27-86 and described in Table 27-80.

Return to the Summary Table.

MCAN Subsystem Interrupt Enable Clear Shadow Register

Figure 27-86. MCANSS_IECS Register

31	30	29	28	27	26	25	24
RESERVED							
R/W-0h							
23	22	21	20	19	18	17	16
RESERVED							
R/W-0h							
15	14	13	12	11	10	9	8
RESERVED							
R/W-0h							
7	6	5	4	3	2	1	0
RESERVED							EXT_TS_CNTR_OVFL
R/W-0h							R-0/W1C-0h

Table 27-140. MCANSS_IECS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R/W	0h	
0	EXT_TS_CNTR_OVFL	R-0/W1C	0h	External Timestamp Counter Overflow Interrupt Enable Clear. Reads always return a 0. 0 Write of '0' has no effect 1 Write of '1' clears the MCANSS_IES.EXT_TS_CNTR_OVFL bit

ADVANCE INFORMATION

61 MCANSS_IE Register (Offset = 7218h) [Reset = 00000000h]

MCANSS_IE is shown in Figure 27-87 and described in Table 27-81.

Return to the Summary Table.

MCAN Subsystem Interrupt Enable Register

Figure 27-87. MCANSS_IE Register

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

Table 27-142. MCANSS_IE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R/W	0h	
0	EXT_TS_CNTR_OVFL	R/W1S	0h	External Timestamp Counter Overflow Interrupt Enable. A write of '0' has no effect. A write of '1' sets the MCANSS_IES.EXT_TS_CNTR_OVFL bit.

ADVANCE INFORMATION

62 MCANSS_IES Register (Offset = 721Ch) [Reset = 00000000h]

MCANSS_IES is shown in Figure 27-88 and described in Table 27-82.

Return to the Summary Table.

MCAN Subsystem Interrupt Enable Status

Figure 27-88. MCANSS_IES Register

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

Table 27-144. MCANSS_IES Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	
0	EXT_TS_CNTR_OVFL	R	0h	External Timestamp Counter Overflow Interrupt Enable Status. To set, use the CANSS_IE.EXT_TS_CNTR_OVFL bit. To clear, use the MCANSS_IECS.EXT_TS_CNTR_OVFL bit. 0 External timestamp counter overflow interrupt is not enabled 1 External timestamp counter overflow interrupt is enabled

63 MCANSS_EOI Register (Offset = 7220h) [Reset = 00000000h]

MCANSS_EOI is shown in Figure 27-89 and described in Table 27-83.

Return to the Summary Table.

MCAN Subsystem End of Interrupt

Figure 27-89. MCANSS_EOI Register

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

Table 27-146. MCANSS_EOI Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R/W	0h	
7-0	EOI	R-0/W1S	0h	End of Interrupt. A write to this register will clear the associated interrupt. If the unserviced interrupt counter is > 1, another interrupt is generated. 0x00 External TS Interrupt is cleared 0x01 MCAN(0) interrupt is cleared 0x02 MCAN(1) interrupt is cleared Other writes are ignored.

64 MCANSS_EXT_TS_PRESCALER Register (Offset = 7224h) [Reset = 00000000h]

MCANSS_EXT_TS_PRESCALER is shown in Figure 27-90 and described in Table 27-84.

Return to the Summary Table.

MCAN Subsystem External Timestamp Prescaler 0

Figure 27-90. MCANSS_EXT_TS_PRESCALER Register

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

Table 27-148. MCANSS_EXT_TS_PRESCALER Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	RESERVED	R/W	0h	
23-0	PRESCALER	R/W	0h	External Timestamp Prescaler Reload Value. The external timestamp count rate is the host (system) clock rate divided by this value, except in the case of 0. A zero value in this bit field will act identically to a value of 0x000001.

65 MCANSS_EXT_TS_UNSERVICED_INTR_CNTR Register (Offset = 7228h) [Reset = 00000000h]

MCANSS_EXT_TS_UNSERVICED_INTR_CNTR is shown in Figure 27-91 and described in Table 27-85.

Return to the Summary Table.

MCAN Subsystem External Timestamp Unserviced Interrupts Counter

Figure 27-91. MCANSS_EXT_TS_UNSERVICED_INTR_CNTR Register

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

Table 27-150. MCANSS_EXT_TS_UNSERVICED_INTR_CNTR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-5	RESERVED	R	0h	
4-0	EXT_TS_INTR_CNTR	R	0h	External Timestamp Counter Unserviced Rollover Interrupts. If this value is > 1, an MCANSS_EOI write of '1' to bit 0 will issue another interrupt. The status of this bit field is affected by the MCANSS_IRS.EXT_TS_CNTR_OVFL bit field.

66 MCANERR_REV Register (Offset = 7400h) [Reset = 66A0EA00h]

MCANERR_REV is shown in Figure 27-92 and described in Table 27-86.

Return to the Summary Table.

MCAN Error Aggregator Revision Register

Figure 27-92. MCANERR_REV Register

31	30	29	28	27	26	25	24
SCHEME		RESERVED		MODULE_ID			
R-1h		R-2h		R-6A0h			
23	22	21	20	19	18	17	16
MODULE_ID							
R-6A0h							
15	14	13	12	11	10	9	8
RESERVED					REVMAJ		
R-1Dh					R-2h		
7	6	5	4	3	2	1	0
RESERVED		REVMIN					
R-0h		R-0h					

Table 27-152. MCANERR_REV Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	SCHEME	R	1h	PID Register Scheme
29-28	RESERVED	R	2h	Reserved
27-16	MODULE_ID	R	6A0h	Module Identification Number
15-11	RESERVED	R	1Dh	Reserved
10-8	REVMAJ	R	2h	Major Revision of the Error Aggregator
7-6	RESERVED	R	0h	Reserved
5-0	REVMIN	R	0h	Minor Revision of the Error Aggregator

67 MCANERR_VECTOR Register (Offset = 7408h) [Reset = 00000000h]

MCANERR_VECTOR is shown in Figure 27-93 and described in Table 27-87.

Return to the Summary Table.

Each error detection and correction (EDC) controller has a bank of error registers (offsets 0x10 - 0x3B) associated with it. These registers are accessed via an internal serial bus (SVBUS). To access them through the ECC aggregator the controller ID desired must be written to the ECC_VECTOR field, together with the RD_SVBUS trigger and RD_SVBUS_ADDRESS bit field. This initiates the serial read which consummates by setting the RD_SVBUS_DONE bit. At this point the addressed register may be read by a normal CPU read of the appropriate offset address.

Figure 27-93. MCANERR_VECTOR Register

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

Table 27-154. MCANERR_VECTOR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-25	RESERVED	R/W	0h	
24	RD_SVBUS_DONE	R	0h	Read Completion Flag
23-16	RD_SVBUS_ADDRESS	R/W	0h	Read Address Offset
15	RD_SVBUS	R-0/W1S	0h	Read Trigger
14-11	RESERVED	R/W	0h	
10-0	ECC_VECTOR	R/W	0h	ECC RAM ID. Each error detection and correction (EDC) controller has a bank of error registers (offsets 0x10 - 0x3B) associated with it. These registers are accessed via an internal serial bus (SVBUS). To access them through the ECC aggregator the controller ID desired must be written to the ECC_VECTOR field, together with the RD_SVBUS trigger and RD_SVBUS_ADDRESS bit field. This initiates the serial read which consummates by setting the RD_SVBUS_DONE bit. At this point the addressed register may be read by a normal CPU read of the appropriate offset address. 0x000 Message RAM ECC controller is selected Others Reserved (do not use) Subsequent writes through the SVBUS (offsets 0x10 - 0x3B) have a delayed completion. To avoid conflicts, perform a read back of a register within this range after writing.

68 MCANERR_STAT Register (Offset = 740Ch) [Reset = 00000002h]

MCANERR_STAT is shown in Figure 27-94 and described in Table 27-88.

Return to the Summary Table.

MCAN Error Misc Status

Figure 27-94. MCANERR_STAT Register

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

Table 27-156. MCANERR_STAT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-11	RESERVED	R	0h	
10-0	NUM_RAMS	R	2h	Number of RAMs. Number of ECC RAMs serviced by the aggregator.

69 MCANERR_WRAP_REV Register (Offset = 7410h) [Reset = 66A46A02h]

MCANERR_WRAP_REV is shown in Figure 27-95 and described in Table 27-89.

Return to the Summary Table.

This register is accessed through the ECC aggregator via an internal serial bus. To access, the appropriate procedure must be first followed in the MCAN ECC Vector Register.

Figure 27-95. MCANERR_WRAP_REV Register

31	30	29	28	27	26	25	24
SCHEME		RESERVED		MODULE_ID			
R-1h		R-2h		R-6A4h			
23	22	21	20	19	18	17	16
MODULE_ID							
R-6A4h							
15	14	13	12	11	10	9	8
RESERVED					REVMAJ		
R-Dh					R-2h		
7	6	5	4	3	2	1	0
RESERVED		REVMIN					
R-0h		R-2h					

Table 27-158. MCANERR_WRAP_REV Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	SCHEME	R	1h	PID Register Scheme
29-28	RESERVED	R	2h	Reserved
27-16	MODULE_ID	R	6A4h	Module Identification Number
15-11	RESERVED	R	Dh	Reserved
10-8	REVMAJ	R	2h	Major Revision of the Error Aggregator
7-6	RESERVED	R	0h	Reserved
5-0	REVMIN	R	2h	Minor Revision of the Error Aggregator

70 MCANERR_CTRL Register (Offset = 7414h) [Reset = 00000187h]

MCANERR_CTRL is shown in Figure 27-96 and described in Table 27-90.

Return to the Summary Table.

This register is accessed through the ECC aggregator via an internal serial bus. To access, the appropriate procedure must be first followed in the MCAN ECC Vector Register.

Figure 27-96. MCANERR_CTRL Register

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

Table 27-160. MCANERR_CTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	RESERVED	R/W	0h	
8	CHECK_SVBUS_TIMEOUT	R/W	1h	Enables Serial VBUS timeout mechanism
7	RESERVED	R/W	1h	
6	ERROR_ONCE	R/W	0h	If this bit is set, the FORCE_SEC/FORCE_DED will inject an error to the specified row only once. The FORCE_SEC bit will be cleared once a writeback happens. If writeback is not enabled, this error will be cleared the cycle following the read when the data is corrected. For double-bit errors, the FORCE_DED bit will be cleared the cycle following the double-bit error. Any subsequent reads will not force an error.
5	FORCE_N_ROW	R/W	0h	Enable single/double-bit error on the next RAM read, regardless of the MCANERR_ERR_CTRL1.ECC_ROW setting. For write through mode, this applies to writes as well as reads.
4	FORCE_DED	R/W	0h	Force double-bit error. Cleared the cycle following the error if ERROR_ONCE is asserted. For write through mode, this applies to writes as well as reads. MCANERR_ERR_CTRL1 and MCANERR_ERR_CTRL2 should be configured prior to setting this bit.
3	FORCE_SEC	R/W	0h	Force single-bit error. Cleared on a writeback or the cycle following the error if ERROR_ONCE is asserted. For write through mode, this applies to writes as well as reads. MCANERR_ERR_CTRL1 and MCANERR_ERR_CTRL2 should be configured prior to setting this bit.
2	ENABLE_RMW	R/W	1h	Enable read-modify-write on partial word writes
1	ECC_CHECK	R/W	1h	Enable ECC Check. ECC is completely bypassed if both ECC_ENABLE and ECC_CHECK are '0'.
0	ECC_ENABLE	R/W	1h	Enable ECC Generation

71 MCANERR_ERR_CTRL1 Register (Offset = 7418h) [Reset = 00000000h]

MCANERR_ERR_CTRL1 is shown in Figure 27-97 and described in Table 27-91.

Return to the Summary Table.

This register is accessed through the ECC aggregator via an internal serial bus. To access, the appropriate procedure must be first followed in the MCAN ECC Vector Register.

Figure 27-97. MCANERR_ERR_CTRL1 Register

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

Table 27-162. MCANERR_ERR_CTRL1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ECC_ROW	R/W	0h	Row address where FORCE_SEC or FORCE_DED needs to be applied. This is ignored if FORCE_N_ROW is set.

ADVANCE INFORMATION

72 MCANERR_ERR_CTRL2 Register (Offset = 741Ch) [Reset = 00000000h]

MCANERR_ERR_CTRL2 is shown in Figure 27-98 and described in Table 27-92.

Return to the Summary Table.

This register is accessed through the ECC aggregator via an internal serial bus. To access, the appropriate procedure must be first followed in the MCAN ECC Vector Register.

Figure 27-98. MCANERR_ERR_CTRL2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ECC_BIT2																ECC_BIT1															
R/W-0h																R/W-0h															

Table 27-164. MCANERR_ERR_CTRL2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	ECC_BIT2	R/W	0h	Second column/data bit that needs to be flipped when FORCE_DED is set
15-0	ECC_BIT1	R/W	0h	Column/Data bit that needs to be flipped when FORCE_SEC or FORCE_DED is set

73 MCANERR_ERR_STAT1 Register (Offset = 7420h) [Reset = 00000000h]

MCANERR_ERR_STAT1 is shown in Figure 27-99 and described in Table 27-93.

Return to the Summary Table.

This register is accessed through the ECC aggregator via an internal serial bus. To access, the appropriate procedure must be first followed in the MCAN ECC Vector Register.

Figure 27-99. MCANERR_ERR_STAT1 Register

31	30	29	28	27	26	25	24
ECC_BIT1							
R-0h							
23	22	21	20	19	18	17	16
ECC_BIT1							
R-0h							
15	14	13	12	11	10	9	8
CLR_CTRL_REG_ERROR	RESERVED		CLR_ECC_OTHER	CLR_ECC_DED		CLR_ECC_SEC	
R/W1S-0h	R/W-0h		R/W1C-0h	R/WD-0h		R/WD-0h	
7	6	5	4	3	2	1	0
CTRL_REG_ERROR	RESERVED		ECC_OTHER	ECC_DED		ECC_SEC	
R/W1S-0h	R/W-0h		R/W1S-0h	R/WI-0h		R/WI-0h	

Table 27-166. MCANERR_ERR_STAT1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	ECC_BIT1	R	0h	ECC Error Bit Position. Indicates the bit position in the RAM data that is in error on an SEC error. Only valid on an SEC error. 0 Bit 0 is in error 1 Bit 1 is in error 2 Bit 2 is in error 3 Bit 3 is in error ... 31 Bit 31 is in error >32 Invalid
15	CLR_CTRL_REG_ERROR	R/W1S	0h	Writing a '1' clears the CTRL_REG_ERROR bit
14-13	RESERVED	R/W	0h	
12	CLR_ECC_OTHER	R/W1C	0h	Writing a '1' clears the ECC_OTHER bit.
11-10	CLR_ECC_DED	R/WD	0h	Clear ECC_DED. A write of a non-zero value to this bit field decrements the ECC_DED bit field by the value provided.
9-8	CLR_ECC_SEC	R/WD	0h	Clear ECC_SEC. A write of a non-zero value to this bit field decrements the ECC_SEC bit field by the value provided.
7	CTRL_REG_ERROR	R/W1S	0h	Control Register Error. A bit field in the control register is in an ambiguous state. This means that the redundancy registers have detected a state where not all values are the same and has defaulted to the reset state. S/W needs to re-write these registers to a known state. A write of 1 will set this interrupt flag.
6-5	RESERVED	R/W	0h	
4	ECC_OTHER	R/W1S	0h	SEC While Writeback Error Status 0 No SEC error while writeback pending 1 Indicates that successive single-bit errors have occurred while a writeback is still pending

Table 27-166. MCANERR_ERR_STAT1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3-2	ECC_DED	R/WI	0h	Double Bit Error Detected Status. A 2-bit saturating counter of the number of DED errors that have occurred since last cleared. 0 No double-bit error detected 1 One double-bit error was detected 2 Two double-bit errors were detected 3 Three double-bit errors were detected A write of a non-zero value to this bit field increments it by the value provided.
1-0	ECC_SEC	R/WI	0h	Single Bit Error Corrected Status. A 2-bit saturating counter of the number of SEC errors that have occurred since last cleared. 0 No single-bit error detected 1 One single-bit error was detected and corrected 2 Two single-bit errors were detected and corrected 3 Three single-bit errors were detected and corrected A write of a non-zero value to this bit field increments it by the value provided.

74 MCANERR_ERR_STAT2 Register (Offset = 7424h) [Reset = 00000000h]

MCANERR_ERR_STAT2 is shown in Figure 27-100 and described in Table 27-94.

Return to the Summary Table.

This register is accessed through the ECC aggregator via an internal serial bus. To access, the appropriate procedure must be first followed in the MCAN ECC Vector Register.

Figure 27-100. MCANERR_ERR_STAT2 Register

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

Table 27-168. MCANERR_ERR_STAT2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ECC_ROW	R	0h	Indicates the row address where the single or double-bit error occurred. This value is address offset/4.

75 MCANERR_ERR_STAT3 Register (Offset = 7428h) [Reset = 00000000h]

MCANERR_ERR_STAT3 is shown in Figure 27-101 and described in Table 27-95.

Return to the Summary Table.

This register is accessed through the ECC aggregator via an internal serial bus. To access, the appropriate procedure must be first followed in the MCAN ECC Vector Register.

Figure 27-101. MCANERR_ERR_STAT3 Register

31	30	29	28	27	26	25	24
RESERVED							
R/W-0h							
23	22	21	20	19	18	17	16
RESERVED							
R/W-0h							
15	14	13	12	11	10	9	8
RESERVED						CLR_SVBUS_TIMEOUT	RESERVED
R/W-0h						R-0/W1C-0h	R/W-0h
7	6	5	4	3	2	1	0
RESERVED						SVBUS_TIMEOUT	WB_PEND
R/W-0h						R-0/W1S-0h	R-0h

Table 27-170. MCANERR_ERR_STAT3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-10	RESERVED	R/W	0h	
9	CLR_SVBUS_TIMEOUT	R-0/W1C	0h	Write 1 to clear the Serial VBUS Timeout Flag
8-2	RESERVED	R/W	0h	
1	SVBUS_TIMEOUT	R-0/W1S	0h	Serial VBUS Timeout Flag. Write 1 to set.
0	WB_PEND	R	0h	Delayed Write Back Pending Status 0 No write back pending 1 An ECC data correction write back is pending

ADVANCE INFORMATION

76 MCANERR_SEC_EOI Register (Offset = 743Ch) [Reset = 00000000h]

MCANERR_SEC_EOI is shown in Figure 27-102 and described in Table 27-96.

Return to the Summary Table.

MCAN Single Error Corrected End of Interrupt Register

Figure 27-102. MCANERR_SEC_EOI Register

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

Table 27-172. MCANERR_SEC_EOI Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R/W	0h	
0	EOI_WR	R-0/W1S	0h	Write to this register indicates that software has acknowledged the pending interrupt and the next interrupt can be sent to the host. Note that a write to the MCANERR_ERR_STAT1.CLR_ECC_SEC goes through the SVBUS and has a delayed completion. To avoid an additional interrupt, read the MCANERR_ERR_STAT1 register back prior to writing to this bit field.

77 MCANERR_SEC_STATUS Register (Offset = 7440h) [Reset = 00000000h]

MCANERR_SEC_STATUS is shown in Figure 27-103 and described in Table 27-97.

Return to the Summary Table.

MCAN Single Error Corrected Interrupt Status Register

Figure 27-103. MCANERR_SEC_STATUS Register

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

Table 27-174. MCANERR_SEC_STATUS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R/W	0h	
0	MSGMEM_PEND	R-0	0h	Message RAM SEC Interrupt Pending 0 No SEC interrupt is pending 1 SEC interrupt is pending

ADVANCE INFORMATION

78 MCANERR_SEC_ENABLE_SET Register (Offset = 7480h) [Reset = 00000000h]

MCANERR_SEC_ENABLE_SET is shown in Figure 27-104 and described in Table 27-98.

Return to the Summary Table.

MCAN Single Error Corrected Interrupt Enable Set Register

Figure 27-104. MCANERR_SEC_ENABLE_SET Register

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

Table 27-176. MCANERR_SEC_ENABLE_SET Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R/W	0h	
0	MSGMEM_ENABLE_SET	R/W1S	0h	Message RAM SEC Interrupt Pending Enable Set. Writing a 1 to this bit enables the Message RAM SEC error interrupts. Writing a 0 has no effect. Reads return the corresponding enable bit's current value.

ADVANCE INFORMATION

79 MCANERR_SEC_ENABLE_CLR Register (Offset = 74C0h) [Reset = 00000000h]

MCANERR_SEC_ENABLE_CLR is shown in Figure 27-105 and described in Table 27-99.

Return to the Summary Table.

MCAN Single Error Corrected Interrupt Enable Clear Register

Figure 27-105. MCANERR_SEC_ENABLE_CLR Register

31	30	29	28	27	26	25	24
RESERVED							
R/W-0h							
23	22	21	20	19	18	17	16
RESERVED							
R/W-0h							
15	14	13	12	11	10	9	8
RESERVED							
R/W-0h							
7	6	5	4	3	2	1	0
RESERVED							MSGMEM_ENABLE_CLR
R/W-0h							R/W1C-0h

Table 27-178. MCANERR_SEC_ENABLE_CLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R/W	0h	
0	MSGMEM_ENABLE_CLR	R/W1C	0h	Message RAM SEC Interrupt Pending Enable Clear. Writing a 1 to this bit disables the Message RAM SEC error interrupts. Writing a 0 has no effect. Reads return the corresponding enable bit's current value.

80 MCANERR_DED_EOI Register (Offset = 753Ch) [Reset = 00000000h]

MCANERR_DED_EOI is shown in Figure 27-106 and described in Table 27-100.

Return to the Summary Table.

MCAN Double Error Detected End of Interrupt Register

Figure 27-106. MCANERR_DED_EOI Register

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

Table 27-180. MCANERR_DED_EOI Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R/W	0h	
0	EOI_WR	R-0/W1S	0h	Write to this register indicates that software has acknowledged the pending interrupt and the next interrupt can be sent to the host. Note that a write to the MCANERR_ERR_STAT1.CLR_ECC_DED goes through the SVBUS and has a delayed completion. To avoid an additional interrupt, read the MCANERR_ERR_STAT1 register back prior to writing to this bit field.

81 MCANERR_DED_STATUS Register (Offset = 7540h) [Reset = 00000000h]

MCANERR_DED_STATUS is shown in Figure 27-107 and described in Table 27-101.

Return to the Summary Table.

MCAN Double Error Detected Interrupt Status Register

Figure 27-107. MCANERR_DED_STATUS Register

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

Table 27-182. MCANERR_DED_STATUS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R/W	0h	
0	MSGMEM_PEND	R-0	0h	Message RAM DED Interrupt Pending 0 No DED interrupt is pending 1 DED interrupt is pending

82 MCANERR_DED_ENABLE_SET Register (Offset = 7580h) [Reset = 00000000h]

MCANERR_DED_ENABLE_SET is shown in Figure 27-108 and described in Table 27-102.

Return to the Summary Table.

MCAN Double Error Detected Interrupt Enable Set Register

Figure 27-108. MCANERR_DED_ENABLE_SET Register

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

Table 27-184. MCANERR_DED_ENABLE_SET Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R/W	0h	
0	MSGMEM_ENABLE_SET	R/W1S	0h	Message RAM DED Interrupt Pending Enable Set. Writing a 1 to this bit enables the Message RAM DED error interrupts. Writing a 0 has no effect. Reads return the corresponding enable bit's current value.

ADVANCE INFORMATION

83 MCANERR_DED_ENABLE_CLR Register (Offset = 75C0h) [Reset = 00000000h]

MCANERR_DED_ENABLE_CLR is shown in Figure 27-109 and described in Table 27-103.

Return to the Summary Table.

MCAN Double Error Detected Interrupt Enable Clear Register

Figure 27-109. MCANERR_DED_ENABLE_CLR Register

31	30	29	28	27	26	25	24
RESERVED							
R/W-0h							
23	22	21	20	19	18	17	16
RESERVED							
R/W-0h							
15	14	13	12	11	10	9	8
RESERVED							
R/W-0h							
7	6	5	4	3	2	1	0
RESERVED							MSGMEM_ENABLE_CLR
R/W-0h							R/W1C-0h

Table 27-186. MCANERR_DED_ENABLE_CLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R/W	0h	
0	MSGMEM_ENABLE_CLR	R/W1C	0h	Message RAM DED Interrupt Pending Enable Clear. Writing a 1 to this bit disables the Message RAM DED error interrupts. Writing a 0 has no effect. Reads return the corresponding enable bit's current value.

84 MCANERR_AGGR_ENABLE_SET Register (Offset = 7600h) [Reset = 00000000h]

MCANERR_AGGR_ENABLE_SET is shown in Figure 27-110 and described in Table 27-104.

Return to the Summary Table.

MCAN Error Aggregator Enable Set Register

Figure 27-110. MCANERR_AGGR_ENABLE_SET Register

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

Table 27-188. MCANERR_AGGR_ENABLE_SET Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R/W	0h	
1	ENABLE_TIMEOUT_SET	R/W1S	0h	Write 1 to enable timeout errors. Reads return the corresponding enable bit's current value.
0	ENABLE_PARITY_SET	R/W1S	0h	Write 1 to enable parity errors. Reads return the corresponding enable bit's current value.

85 MCANERR_AGGR_ENABLE_CLR Register (Offset = 7604h) [Reset = 00000000h]

MCANERR_AGGR_ENABLE_CLR is shown in Figure 27-111 and described in Table 27-105.

Return to the Summary Table.

MCAN Error Aggregator Enable Clear Register

Figure 27-111. MCANERR_AGGR_ENABLE_CLR Register

31	30	29	28	27	26	25	24
RESERVED							
R/W-0h							
23	22	21	20	19	18	17	16
RESERVED							
R/W-0h							
15	14	13	12	11	10	9	8
RESERVED							
R/W-0h							
7	6	5	4	3	2	1	0
RESERVED						ENABLE_TIMEOUT_CLR	ENABLE_PARITY_CLR
R/W-0h						R/W1C-0h	R/W1C-0h

Table 27-190. MCANERR_AGGR_ENABLE_CLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R/W	0h	
1	ENABLE_TIMEOUT_CLR	R/W1C	0h	Write 1 to disable timeout errors. Reads return the corresponding enable bit's current value.
0	ENABLE_PARITY_CLR	R/W1C	0h	Write 1 to disable parity errors. Reads return the corresponding enable bit's current value.

86 MCANERR_AGGR_STATUS_SET Register (Offset = 7608h) [Reset = 00000000h]

MCANERR_AGGR_STATUS_SET is shown in Figure 27-112 and described in Table 27-106.

Return to the Summary Table.

MCAN Error Aggregator Status Set Register

Figure 27-112. MCANERR_AGGR_STATUS_SET Register

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

Table 27-192. MCANERR_AGGR_STATUS_SET Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R/W	0h	
3-2	SVBUS_TIMEOUT	R/WI	0h	Aggregator Serial VBUS Timeout Error Status 2-bit saturating counter of the number of SVBUS timeout errors that have occurred since last cleared. 0 No timeout errors have occurred 1 One timeout error has occurred 2 Two timeout errors have occurred 3 Three timeout errors have occurred A write of a non-zero value to this bit field increments it by the value provided.
1-0	AGGR_PARITY_ERR	R/WI	0h	Aggregator Parity Error Status 2-bit saturating counter of the number of parity errors that have occurred since last cleared. 0 No parity errors have occurred 1 One parity error has occurred 2 Two parity errors have occurred 3 Three parity errors have occurred A write of a non-zero value to this bit field increments it by the value provided.

87 MCANERR_AGGR_STATUS_CLR Register (Offset = 760Ch) [Reset = 00000000h]

MCANERR_AGGR_STATUS_CLR is shown in Figure 27-113 and described in Table 27-107.

Return to the Summary Table.

MCAN Error Aggregator Status Clear Register

Figure 27-113. MCANERR_AGGR_STATUS_CLR Register

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

Table 27-194. MCANERR_AGGR_STATUS_CLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R/W	0h	
3-2	SVBUS_TIMEOUT	R/WD	0h	Aggregator Serial VBUS Timeout Error Status 2-bit saturating counter of the number of SVBUS timeout errors that have occurred since last cleared. 0 No timeout errors have occurred 1 One timeout error has occurred 2 Two timeout errors have occurred 3 Three timeout errors have occurred A write of a non-zero value to this bit field decrements it by the value provided.
1-0	AGGR_PARITY_ERR	R/WD	0h	Aggregator Parity Error Status 2-bit saturating counter of the number of parity errors that have occurred since last cleared. 0 No parity errors have occurred 1 One parity error has occurred 2 Two parity errors have occurred 3 Three parity errors have occurred A write of a non-zero value to this bit field decrements it by the value provided.

88 IIDX Register (Offset = 7820h) [Reset = 00000000h]

IIDX is shown in Figure 27-114 and described in Table 27-108.

Return to the Summary Table.

This register provides the highest priority enabled interrupt index. Value 0x00 means no event pending. Interrupt 1 is the highest priority, IIDX next highest, 4, 8, ... IIDX^31 is the least priority. That is, the least bit position that is set to 1 denotes the highest priority pending interrupt. The priority order is fixed. However, users can implement their own prioritization schemes using other registers that expose the full set of interrupts that have occurred. On each read, only one interrupt is indicated. On a read, the current interrupt (highest priority) is automatically cleared by the hardware and the corresponding interrupt flag in [RIS] and [MIS] are cleared as well. After a read from the CPU (not from the debug interface), the register is updated with the next highest priority interrupt, if none are pending, then it should display 0x0.

Figure 27-114. IIDX Register

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

Table 27-196. IIDX Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	
7-0	STAT	R	0h	Interrupt index status 00h = No interrupt pending. 1h = MCAN Interrupt Line 0 interrupt pending. 2h = MCAN Interrupt Line 1 interrupt pending. 3h = Message RAM SEC (Single Error Correction) interrupt pending. 4h = Message RAM DED (Double Error Detection) interrupt pending. 5h = External Timestamp Counter Overflow interrupt pending. 6h = Clock Stop Wake Up interrupt pending.

89 IMASK Register (Offset = 7828h) [Reset = 00000000h]

IMASK is shown in Figure 27-115 and described in Table 27-109.

Return to the Summary Table.

Interrupt Mask. If a bit is set, then corresponding interrupt is un-masked. Un-masking the interrupt causes the raw interrupt to be visible in IIDX, as well as MIS.

Figure 27-115. IMASK Register

31	30	29	28	27	26	25	24
RESERVED							
R/W-0h							
23	22	21	20	19	18	17	16
RESERVED							
R/W-0h							
15	14	13	12	11	10	9	8
RESERVED							
R/W-0h							
7	6	5	4	3	2	1	0
RESERVED		WAKEUP	EXT_TS_CNTR_OVFL	DED	SEC	INTL1	INTL0
R/W-0h		R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 27-198. IMASK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-6	RESERVED	R/W	0h	
5	WAKEUP	R/W	0h	Clock Stop Wake Up interrupt mask. 0h = Clear Interrupt Mask 1h = Set Interrrupt Mask
4	EXT_TS_CNTR_OVFL	R/W	0h	External Timestamp Counter Overflow interrupt mask. 0h = Clear Interrupt Mask 1h = Set Interrrupt Mask
3	DED	R/W	0h	Massage RAM DED interrupt mask. 0h = Clear Interrupt Mask 1h = Set Interrrupt Mask
2	SEC	R/W	0h	Message RAM SEC interrupt mask. 0h = Clear Interrupt Mask 1h = Set Interrrupt Mask
1	INTL1	R/W	0h	MCAN Interrupt Line 1 mask. 0h = Clear Interrupt Mask 1h = Set Interrrupt Mask
0	INTL0	R/W	0h	MCAN Interrupt Line 0 mask. 0h = Clear Interrupt Mask 1h = Set Interrrupt Mask

90 RIS Register (Offset = 7830h) [Reset = 00000000h]

RIS is shown in Figure 27-116 and described in Table 27-110.

Return to the Summary Table.

Raw interrupt status. Reflects all pending interrupts, regardless of masking. The RIS register allows the user to implement a poll scheme. A flag set in this register can be cleared by writing 1 to the ICLR register bit even if the corresponding IMASK bit is not enabled.

Figure 27-116. RIS Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED		WAKEUP	EXT_TS_CNTR_OVFL	DED	SEC	INTL1	INTL0
R-0h		R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

Table 27-200. RIS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-6	RESERVED	R	0h	
5	WAKEUP	R	0h	Clock Stop Wake Up interrupt. 0h = Interrupt did not occur 1h = Interrupt occured
4	EXT_TS_CNTR_OVFL	R	0h	External Timestamp Counter Overflow interrupt. 0h = Interrupt did not occur 1h = Interrupt occured
3	DED	R	0h	Message RAM DED interrupt. 0h = Interrupt did not occur 1h = Interrupt occured
2	SEC	R	0h	Message RAM SEC interrupt. 0h = Interrupt did not occur 1h = Interrupt occured
1	INTL1	R	0h	MCAN Interrupt Line 1. 0h = Interrupt did not occur 1h = Interrupt occured
0	INTL0	R	0h	MCAN Interrupt Line 0. 0h = Interrupt did not occur 1h = Interrupt occured

91 MIS Register (Offset = 7838h) [Reset = 00000000h]

MIS is shown in Figure 27-117 and described in Table 27-111.

Return to the Summary Table.

Masked interrupt status. This is an AND of the IMASK and RIS registers.

Figure 27-117. MIS Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED		WAKEUP	EXT_TS_CNTR_OVFL	DED	SEC	INTL1	INTL0
R-0h		R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

Table 27-202. MIS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-6	RESERVED	R	0h	
5	WAKEUP	R	0h	Masked Clock Stop Wake Up interrupt. 0h = Interrupt did not occur 1h = Interrupt occured
4	EXT_TS_CNTR_OVFL	R	0h	Masked External Timestamp Counter Overflow interrupt. 0h = Interrupt did not occur 1h = Interrupt occured
3	DED	R	0h	Masked Message RAM DED interrupt. 0h = Interrupt did not occur 1h = Interrupt occured
2	SEC	R	0h	Masked Message RAM SEC interrupt. 0h = Interrupt did not occur 1h = Interrupt occured
1	INTL1	R	0h	Masked MCAN Interrupt Line 1. 0h = Interrupt did not occur 1h = Interrupt occured
0	INTL0	R	0h	Masked MCAN Interrupt Line 0. 0h = Interrupt did not occur 1h = Interrupt occured

92 ISET Register (Offset = 7840h) [Reset = 00000000h]

ISET is shown in Figure 27-118 and described in Table 27-112.

Return to the Summary Table.

Interrupt set. Allows interrupts to be set by software (useful in diagnostics and safety checks). Writing a 1 to a bit in ISET will set the event and therefore the related RIS bit also gets set. If the interrupt is enabled through the mask, then the corresponding MIS bit is also set.

Figure 27-118. ISET Register

31	30	29	28	27	26	25	24
RESERVED							
W-0h							
23	22	21	20	19	18	17	16
RESERVED							
W-0h							
15	14	13	12	11	10	9	8
RESERVED							
W-0h							
7	6	5	4	3	2	1	0
RESERVED		WAKEUP	EXT_TS_CNTR_OVFL	DED	SEC	INTL1	INTL0
W-0h		W-0h	W-0h	W-0h	W-0h	W-0h	W-0h

Table 27-204. ISET Register Field Descriptions

Bit	Field	Type	Reset	Description
31-6	RESERVED	W	0h	
5	WAKEUP	W	0h	Set Clock Stop Wake Up interrupt. 0h = Writing 0 has no effect 1h = Set Interrupt
4	EXT_TS_CNTR_OVFL	W	0h	Set External Timestamp Counter Overflow interrupt. 0h = Writing 0 has no effect 1h = Set Interrupt
3	DED	W	0h	Set Message RAM DED interrupt. 0h = Writing 0 has no effect 1h = Set Interrupt
2	SEC	W	0h	Set Message RAM SEC interrupt. 0h = Writing 0 has no effect 1h = Set Interrupt
1	INTL1	W	0h	Set MCAN Interrupt Line 1. 0h = Writing 0 has no effect 1h = Set Interrupt
0	INTL0	W	0h	Set MCAN Interrupt Line 0. 0h = Writing 0 has no effect 1h = Set Interrupt

93 ICLR Register (Offset = 7848h) [Reset = 00000000h]

ICLR is shown in Figure 27-119 and described in Table 27-113.

Return to the Summary Table.

Interrupt clear. Write a 1 to clear corresponding Interrupt.

Figure 27-119. ICLR Register

31	30	29	28	27	26	25	24
RESERVED							
W-0h							
23	22	21	20	19	18	17	16
RESERVED							
W-0h							
15	14	13	12	11	10	9	8
RESERVED							
W-0h							
7	6	5	4	3	2	1	0
RESERVED		WAKEUP	EXT_TS_CNTR_OVFL	DED	SEC	INTL1	INTL0
W-0h		W-0h	W-0h	W-0h	W-0h	W-0h	W-0h

Table 27-206. ICLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-6	RESERVED	W	0h	
5	WAKEUP	W	0h	Clear Clock Stop Wake Up interrupt. 0h = Writing 0 has no effect 1h = Clear Interrupt
4	EXT_TS_CNTR_OVFL	W	0h	Clear External Timestamp Counter Overflow interrupt. 0h = Writing 0 has no effect 1h = Clear Interrupt
3	DED	W	0h	Clear Message RAM DED interrupt. 0h = Writing 0 has no effect 1h = Clear Interrupt
2	SEC	W	0h	Clear Message RAM SEC interrupt. 0h = Writing 0 has no effect 1h = Clear Interrupt
1	INTL1	W	0h	Clear MCAN Interrupt Line 1. 0h = Writing 0 has no effect 1h = Clear Interrupt
0	INTL0	W	0h	Clear MCAN Interrupt Line 0. 0h = Writing 0 has no effect 1h = Clear Interrupt

94 EVT_MODE Register (Offset = 78E0h) [Reset = 00000000h]

EVT_MODE is shown in Figure 27-120 and described in Table 27-114.

Return to the Summary Table.

Event mode register. It is used to select whether each line is disabled, in software mode (software clears the RIS) or in hardware mode (hardware clears the RIS)

Figure 27-120. EVT_MODE Register

31	30	29	28	27	26	25	24
RESERVED							
R/W-0h							
23	22	21	20	19	18	17	16
RESERVED							
R/W-0h							
15	14	13	12	11	10	9	8
RESERVED							
R/W-0h							
7	6	5	4	3	2	1	0
RESERVED						INT0_CFG	
R/W-0h						R-0h	

Table 27-208. EVT_MODE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R/W	0h	
1-0	INT0_CFG	R	1h	Event line mode select for event corresponding to [IPSTANDARD.CPU_INT] 0h = The interrupt or event line is disabled. 1h = The interrupt or event line is in software mode. Software must clear the RIS. 2h = The interrupt or event line is in hardware mode. The hardware (another module) clears automatically the associated RIS flag.

95 DESC Register (Offset = 78FCh) [Reset = 00000000h]

DESC is shown in Figure 27-121 and described in Table 27-115.

Return to the Summary Table.

This register identifies the peripheral and its exact version.

Figure 27-121. DESC Register

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

Table 27-210. DESC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	MODULEID	R	940h	Module identification contains a unique peripheral identification number. The assignments are maintained in a central database for all of the platform modules to ensure uniqueness. 0h = Smallest value FFFFh = Highest possible value
15-12	FEATUREVER	R	0h	Feature Set for the module *instance* 0h = MCAN module with CAN-FD mode enabled 1h = MCAN module with CAN-FD mode disabled
11-8	RESERVED	R	0h	
7-4	MAJREV	R	0h	Major rev of the IP 0h = Smallest value Fh = Highest possible value
3-0	MINREV	R	0h	Minor rev of the IP 0h = Smallest value Fh = Highest possible value

96 MCANSS_CLKEN Register (Offset = 7900h) [Reset = 00000000h]

MCANSS_CLKEN is shown in Figure 27-122 and described in Table 27-116.

Return to the Summary Table.

MCAN module clock (functional clock and Vbusp to access MCAN module MMRs) enable register

Figure 27-122. MCANSS_CLKEN Register

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

Table 27-212. MCANSS_CLKEN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	
0	CLK_REQEN	0h	MCAN functional and MCAN/ MCANSS MMR clock request enable bit 0h = MCAN module functional clock and Vbusp is not requested. These clocks are gated to the MCAN module. 1h = Setting this bit requests MCAN module functional clock and Vbusp. These clocks are not gated to MCAN module.	

97 MCANSS_CLKDIV Register (Offset = 7904h) [Reset = 00000000h]

MCANSS_CLKDIV is shown in Figure 27-123 and described in Table 27-117.

Return to the Summary Table.

Needs to go to the Management apperture once available

Figure 27-123. MCANSS_CLKDIV Register

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

Table 27-214. MCANSS_CLKDIV Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R/W	0h	
1-0	RATIO	R/W	0h	Clock divide ratio specification. Enables configuring clock divide settings for the MCAN functional clock input to the MCAN-SS. 0h (R/W) = Divides input clock by 1 1h (R/W) = Divides input clock by 2 2h (R/W) = Divides input clock by 4 3h (R/W) = Divides input clock by 1

98 MCANSS_CLKCTL Register (Offset = 7908h) [Reset = 00000000h]

MCANSS_CLKCTL is shown in Figure 27-124 and described in Table 27-118.

Return to the Summary Table.

MCANSS clock stop control MMR.
Bus clock for the wrapper MMRs (including this MMR) is not gated by this register settings.

Figure 27-124. MCANSS_CLKCTL Register

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

Table 27-216. MCANSS_CLKCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	RESERVED	R/W	0h	
8	WKUP_GLTFLT_EN	R/W	0h	Setting this bit enables the glitch filter on MCAN RXD input, which wakes up the MCAN controller to exit clock gating. 0h = Disable glitch filter enable on RXD input when MCAN is in clock stop mode (waiting for event on RXD input for clock stop wakeup). 1h = Enable glitch filter enable on RXD input when MCAN is in clock stop mode (waiting for event on RXD input for clock stop wakeup).
7-5	RESERVED	R/W	0h	
4	WAKEUP_INT_EN	R/W	0h	This bit contols enabling or disabling the MCAN IP clock stop wakeup interrupt (when MCANSS_CTRL.WAKEUPREQEN wakeup request is enabled to wakeup MCAN IP upon CAN RXD activity) 0h = Disable MCAN IP clock stop wakeup interrupt 1h = Enable MCAN IP clock stop wakeup interrupt
3-1	RESERVED	R/W	0h	
0	STOPREQ	R/W	0h	This bit is used to enable/disable MCAN clock (both host clock and functional clock) gating request. Note: This bit can be reset by HW by Clock-Stop Wake-up via CAN RX Activity. See spec for more details. 0h = Disable MCAN-SS clock stop request 1h = Enable MCAN-SS clock stop request

ADVANCE INFORMATION

99 MCANSS_CLKSTS Register (Offset = 790Ch) [Reset = 00000000h]

MCANSS_CLKSTS is shown in Figure 27-125 and described in Table 27-119.

Return to the Summary Table.

MCANSS clock stop status register to indicate status of clock stop mechanism

Figure 27-125. MCANSS_CLKSTS Register

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

Table 27-218. MCANSS_CLKSTS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	RESERVED	R	0h	
8	CCLKDONE	R	0h	This bit indicates the status of MCAN contoller clock request from GPRCM. 0h = MCAN controller clock is not available to the MCAN IP. 1h = MCAN controller clock is enabled and available to the MCAN IP.
7-5	RESERVED	R	0h	
4	STOPREQ_HW_OVR	R	0h	MCANSS clock stop HW override status bit. This bit indicates when the MCANSS_CLKCTL.STOPREQ bit has been cleared by HW when a clock-stop wake-up event via CAN RX activity is triggered. 0h = MCANSS_CLKCTL.STOPREQ bit has not been cleared by HW. 1h = MCANSS_CLKCTL.STOPREQ bit has been cleared by HW.
3-1	RESERVED	R	0h	
0	CLKSTOP_ACKSTS	R	0h	Clock stop acknowledge status from MCAN IP 0h = No clock stop acknowledged. 1h = Clock stop has been acknowledged by MCAN IP MCAN-SS may be clock gated by stopping both the CAN host and functional clocks.

28.1 External Peripheral Interface (EPI)

The External Peripheral Interface is a high-speed parallel bus for external peripherals or memory. It has several modes of operation to interface gluelessly to many types of external devices. Enhanced capabilities include DMA support, clocking control, and support for external FIFO buffers.

28.1.1 Introduction

The EPI has the following features:

- 8-, 16-, or 32-bit dedicated parallel bus for external peripherals and memory
- Memory interface supports contiguous memory access independent of data bus width, thus enabling code execution directly from synchronous dynamic random access memory (SDRAM), synchronous random access memory (SRAM), and flash memory
- Blocking and nonblocking reads
- Offloads the processor from the timing details of the parallel interface through use of an internal write FIFO
- Efficient transfers using Direct Memory Access Controller (DMA)

The EPI supports three primary functional modes: SDRAM mode, traditional host-bus mode, and general-purpose mode. The EPI module also provides a custom GPIO interface that enables fast parallel interfaces using a FIFO with speed control that uses an internal bit clock.

- SDRAM mode
  - Supports x16 (single data rate) SDRAM at up to 50MHz
  - Supports low-cost SDRAMs up to 64MB (512 megabits)
  - Includes automatic refresh and access to all banks and rows
  - Includes a sleep or standby mode to keep contents active with minimal power draw
  - Multiplexed address/data interface for reduced pin count
- Host-bus mode
  - Traditional x8 and x16 MCU bus interface capabilities
  - Access to SRAM, NOR flash memory, and other devices, with up to 1MB of addressing in nonmultiplexed mode and 256MB in multiplexed mode (512MB in host-bus 16 mode with no byte selects)
  - Support for up to 512Mb PSRAM in quad chip select mode, with dedicated configuration register read and write enable
  - Support of both muxed and demuxed address and data
  - Access to a range of devices supporting the nonaddress FIFO x8 and x16 interface variant, with support for external FIFO (XFIFO) empty and full signals
  - Speed controlled, with read and write data wait-state counters
  - Support for read or write burst mode to host bus
  - Multiple chip select modes
  - External iRDY signal provided for stall capability of reads and writes
  - Manual chip-enable (or use extra address pins)
- General-purpose mode
  - Wide parallel interfaces for fast communications with CPLDs and FPGAs
  - Data widths up to 32 bits
  - Data rates up to 125 MB/second
  - Optional "address" sizes from 4 bits to 20 bits
  - Optional clock output, read/write strobes, framing (with counter-based size), and clock-enable input
- General parallel GPIO
  - 1 to 32 bits, FIFOed with speed control
  - Useful for custom peripherals or for digital data acquisition and actuator controls

28.1.2 EPI Block Diagram

Figure 28-1 shows the block diagram of the EPI module.

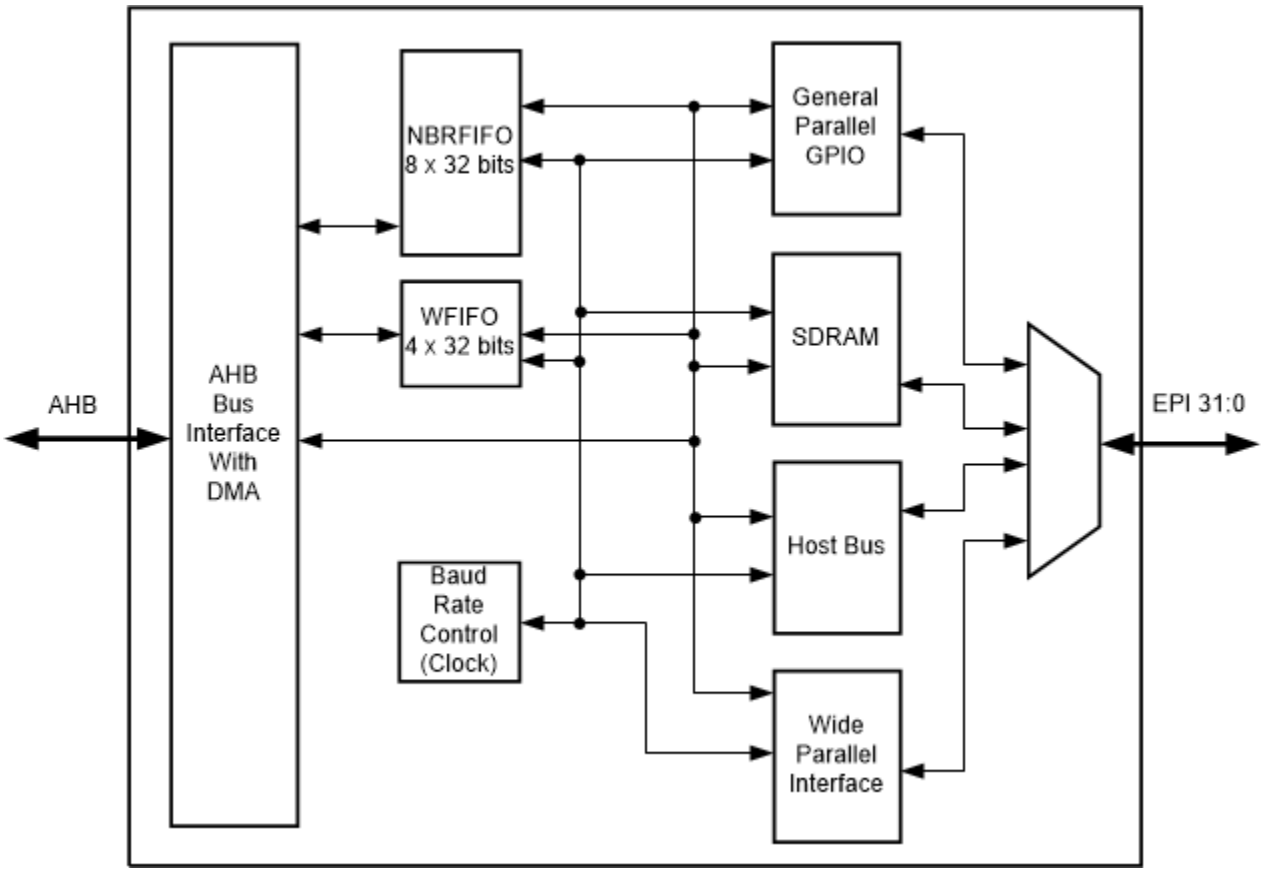


Figure 28-1. EPI Block Diagram

ADVANCE INFORMATION

28.1.3 Functional Description

The EPI controller provides a glueless, programmable interface to a variety of common external peripherals such as SDRAM x16, host bus x8 and x16 devices, RAM, NOR flash memory, CPLDs, and FPGAs. In addition, the EPI controller provides custom GPIOs that can provide a fast speed-controlled parallel interface using either the internal write FIFO (WFIFO) or the nonblocking read FIFO (NBRFIFO) based on the amount of data in the FIFO.

The WFIFO can hold four words of data that are written to the external interface at the rate controlled by the EPI Main Baud Rate (EPIBAUD) registers. The NBRFIFO can hold 8 words of data and samples at the rate controlled by the EPIBAUD register. The EPI controller provides predictable operation and thus has an advantage over regular GPIOs, which have more variable timing due to on-chip bus arbitration and delays across bus bridges. Blocking reads stall the CPU until the transaction completes. Nonblocking reads are performed in the background and allow the processor to continue operation. In addition, write data can also be stored in the WFIFO to allow multiple writes with no stalls.

Note

Poll both the WTAV bit field in the EPIWFIFOCNT register and the WBUSY bit in the EPISTAT register to determine if there is a current write transaction from the WFIFO. If both of these bits are clear, then a new bus access can begin.

Main read and write operations can be performed in subsets of the range 0x8000.0000 to 0x9FFF.FFFF. A read from an address-mapped location uses the offset and size to control the address and size of the external operation. When performing a multiple-value load, the read is done as a burst (when available) to maximize performance. A write to an address-mapped location uses the offset and size to control the address and size of the external operation. When performing a multiple-value store, the write is done as a burst (when available) to maximize performance.

28.1.3.1 Controller Access to EPI

These bus controllers have access to the EPI:

- CPU
- DMA

28.1.3.2 Nonblocking Reads

The EPI Controller supports a special kind of read called a nonblocking read, also referred to as a posted read. Where a normal read stalls the processor or DMA until the data is returned, a nonblocking read is performed in the background.

A nonblocking read is configured by writing the start address into a EPIRADDRn register, the size per transaction into a EPIRSIZEn register, and then the count of operations into a EPIRPSTDn register. After each read is completed, the result is written into the NBRFIFO and the EPIRADDRn register is incremented by the size (1, 2, or 4). The 3 most-significant bits of the EPIRADDRn register are only relevant in the host bus multi-chip select mode when the bits are used to enable the different chip selects.

If the NBRFIFO is filled, then the reads pause until space is made available. The NBRFIFO can be configured to interrupt the processor based on the amount of data in the FIFO using the EPIFIFOLVL register. By using the trigger and interrupt method, the processor can keep space available in the NBRFIFO and allow the reads to continue unimpeded.

When performing nonblocking reads, the SDRAM controller issues two additional read transactions after the burst request is terminated. The data for these additional transfers is discarded. This situation is transparent to the user other than the additional EPI bus activity and can safely be ignored.

Two nonblocking read register sets are available to allow sequencing and ping-pong use. When one completes, the other then activates. So, for example, if 20 words are to be read from 0x100 and 10 words from 0x200, the EPIRPSTD0 register can be set up with the read from 0x100 (with a count of 20), and the EPIRPSTD1 register can be set up with the read from 0x200 (with a count of 10). When EPIRPSTD0 finishes (count goes to 0), the EPIRPSTD1 register then starts an operation. The NBRFIFO has then passed 30 values. It is also possible to reload the EPIRPSTD0 register when it is finished (and the EPIRPSTD1 register is active); thereby, keeping the interface constantly busy.

To cancel a nonblocking read, the EPIRPSTDn register is cleared. Care must be taken, however if the register set was active to drain away any values read into the NBRFIFO and make sure that any read in progress is allowed to complete.

To make sure that the cancel is complete, the following algorithm is used (using the EPIRPSTD0 register for example):

```
EPIRPSTD0 = 0;
while ((EPISTAT & 0x11) == 0x10)
; // we are active and busy
// if here, then other one is active or interface no longer busy
cnt = (EPIRADDR0 - original_address) / EPIRSIZE0; // count of values read
cnt -= values_read_so_far;
// cnt is now number left in FIFO
while (cnt--)
value = EPIREADFIFO; // drain
ISB
```

Note that in the provided algorithm, the important point is to wait for the cancel to complete. This avoids situations where the external interface is in the process of reading a value when the cancel request is received, and it allows the process to complete.

28.1.3.3 DMA Operation

The DMA can be used to achieve maximum transfer rates on the EPI through the NBRFIFO and the WFIFO. The DMA has one channel for write and one for read. For writes, the EPI DMA Transmit Count (EPIDMATXCNT) register is programmed with the total number of transfers by the DMA. An equivalent value is programmed

into the DMA Channel Control Word (DMACHCTL) register of the DMA at offset 0x008. A DMA request is asserted by the EPI WRFIFO when the TXCNT value of the EPIDMATXCNT register is greater than zero and the WTAV bit field of the EPIWFIFOCNT register is less than the programmed threshold trigger, WRFIFO, of the EPIFIFOLVL register. The write channel continues to write data until the TXCNT value in the EPIDMATXCNT register is zero.

Note

When the WRFIFO bit in the EPIFIFOLVL register is set to 0x4 and the application bursts four words to an empty FIFO, the WRFIFO trigger can or can not deassert depending on if all four words were written to the WRFIFO or if the first word is passed immediately to the function requiring WRFIFO. Thus, the application can not see the WRRIS bit in the EPIRIS register clear on a burst of four words.

The nonblocking read channel copies values from the NBRFIFO when the NBRFIFO is at the level specified by the EPIFIFOLVL register. For nonblocking reads, the start address, the size per transaction, and the count of elements must be programmed in the DMA. Both nonblocking read register sets can be used, and the registers fill the NBRFIFO such that one runs to completion, then the next one starts (the registers do not interleave). Using the NBRFIFO provides the best possible transfer rate.

For blocking reads, the DMA software channel (or another unused channel) is used for memory-to-memory transfers (or memory to peripheral, where some other peripheral is used). In this situation, the DMA stalls until the read is complete and is not able to service another channel until the read is done. As a result, the arbitration size must normally be programmed to one access at a time. The DMA controller can also transfer from and to the NBRFIFO and the WFIFO using the DMA software channel in memory mode, however, the DMA is stalled once the NBRFIFO is empty or the WFIFO is full. When the DMA controller is stalled, the core continues operation. For more information on configuring the DMA, see *Direct Memory Access (DMA)*.

The size of the FIFOs must be taken into consideration when configuring the DMA to transfer data to and from the EPI. The arbitration size must be 4 or less when writing to EPI address space and 8 or less when reading from EPI address space.

28.1.4 Initialization and Configuration

To enable and initialize the EPI controller, the following steps are necessary:

1. Enable the EPI module using the PWREN_SYSPERIPH register, see SysCtrl RCGCEPT Register.
2. If desired, enable the input sync to the appropriate GPIO module using the ULPCLK registers (see ULPCLK RCGCGPIO Register).
3. Configure the PINCM fields in the GPIOPCTL register to assign the EPI signals to the appropriate pins (see PINCM GPIOCTL Register and the device-specific data sheet).
4. Select the mode for the EPI block to SDRAM, HB8, HB16, or general parallel use, using the MODE field in the EPI Configuration (EPICFG) register. Set the mode-specific details (if needed) using the appropriate mode configuration EPI Host Bus Configuration (EPIHBnCFGn) registers for the desired chip-select configuration. Set the EPI Main Baud Rate (EPIBAUD) and EPI Main Baud Rate 2 (EPIBAUD2) register if the baud rate must be slower than the system clock rate.
5. Configure the address mapping using the EPI Address Map (EPIADDRMAP) register. The selected start address and range is dependent on the type of external device and maximum address (as appropriate). For example, for a 512-megabit SDRAM, program the ERADR field to 0x1 for address 0x8000.0000 or 0x2 for address 0x9000.0000; and program the ERSZ field to 0x3 for 256MB. If using General-Purpose mode and no address at all, program the EPADR field to 0x1 for address 0x8000.0000 or 0x2 for address 0x9000.0000; and program the EPSZ field to 0x0 for 256 bytes.
6. To read or write directly, use the mapped address area (configured with the EPIADDRMAP register). Up to 4 or 5 writes can be performed at once without blocking. Each read is blocked until the value is retrieved.
7. To perform a nonblocking read, see Section 28.1.3.2 .

Note

The application must not attempt external access until 8 system clock cycles after the EPI has been fully configured.

When a MODE field has been programmed in the EPICFG register, the application must reset all configuration registers before programming to a new MODE value.

The following subsections describe the initialization and configuration for each of the modes of operation. Precisely follow the initialization instructions for the desired configuration to ensure correct operation. Control of the GPIO states is also important, as changes can cause the external device to interpret pin states as actions or commands. Normally, a pullup or pulldown is needed on the board to at least control the chip-select or chip-enable as the GPIOs come out of reset in high-impedance.

28.1.4.1 EPI Interface Options

There are a variety of memories and peripherals that can interface to the EPI module. Table 28-1 shows the various configurations with the maximum performance.

Table 28-1. EPI Interface Options

Interface	Maximum Frequency
Single SDRAM	50MHz
Single SRAM	50MHz
Single PSRAM without iRDY signal use	50MHz
Single PSRAM with iRDY signal use	50MHz
FPGAs, CPLDs, and others using general-purpose mode	50MHz
Memory configurations with 2 chip selects	50MHz
Memory configurations with 4 chip selects	50MHz

28.1.4.2 SDRAM Mode

When activating the SDRAM mode, it is important to consider a few points:

- Generally, it takes over 100µs from when the mode is activated to when the first operation is allowed. The SDRAM controller begins the SDRAM initialization sequence as soon as the mode is selected and enabled using the EPICFG register. It is important that the GPIOs are properly configured before the SDRAM mode is enabled, as the EPI controller is relying on the GPIO block's ability to drive the pins immediately. As part of the initialization sequence, the LOAD MODE REGISTER command is automatically sent to the SDRAM with a value of 0x27, which sets a CAS latency of 2 and a full page burst length.
- The INITSEQ bit in the EPI Status (EPISTAT) register can be checked to determine when the initialization sequence is complete.
- When using a frequency range and/or refresh value other than the default value, it is important to configure the FREQ and RFSH fields in the EPI SDRAM Configuration (EPISDRAMCFG) register shortly after activating the mode. After the 100µs startup time, the EPI block must be configured properly to keep the SDRAM contents stable.
- The SLEEP bit in the EPISDRAMCFG register may be configured to put the SDRAM into a low-power self-refreshing state. It is important to note that the SDRAM mode must not be disabled once enabled, or else the SDRAM is no longer clocked and the contents are lost.
- Before entering SLEEP mode, make sure all nonblocking reads and normal reads and writes have completed. If the system is running at 30MHz to 50MHz, wait 2 EPI clock cycles after clearing the SLEEP bit before executing nonblocking reads, or normal reads and writes. If the system is configured to greater than 50MHz, wait 5 EPI clock cycles before read and write transactions. For all other configurations, wait 1 EPI clock.

The SIZE field of the EPISDRAMCFG register must be configured correctly based on the amount of SDRAM in the system.

The FREQ field must be configured according to the value that represents the range being used. Based on the range selected, the number of external clocks used between certain operations (for example, PRECHARGE or ACTIVATE) is determined. If a higher frequency is given than is used, then the only downside is that the peripheral is slower (uses more cycles for these delays). If a lower frequency is given, incorrect operation occurs.

For timing details for the SDRAM mode, see the device-specific data sheet.

28.1.4.2.1 External Signal Connections

Table 28-2 defines how EPI module signals must be connected to SDRAMs. The table applies when using a x16 SDRAM up to 512 megabits. Any unused EPI controller signals can be used as GPIOs or another alternate function.

Table 28-2. EPI SDRAM x16 Signal Connections

EPI Signal	SDRAM Signal (1)	
EPI0S0	A0	D0
EPI0S1	A1	D1
EPI0S2	A2	D2
EPI0S3	A3	D3
EPI0S4	A4	D4
EPI0S5	A5	D5
EPI0S6	A6	D6
EPI0S7	A7	D7
EPI0S8	A8	D8
EPI0S9	A9	D9
EPI0S10	A10	D10
EPI0S11	A11	D11
EPI0S12	A12 (2)	D12
EPI0S13	BA0	D13
EPI0S14	BA1	D14
EPI0S15	D15	
EPI0S16	DQML	
EPI0S17	DQMH	
EPI0S18	CASn	
EPI0S19	RASn	
EPI0S20-EPI0S27	Not used	
EPI0S28	WEn	
EPI0S29	CSn	
EPI0S30	CKE	
EPI0S31	CLK	

(1) If two signals are listed, connect the EPI signal to both pins.
(2) Only for 256- or 512-megabit SDRAMs

ADVANCE INFORMATION

28.1.4.2.2 Refresh Configuration

The refresh count is based on the external clock speed and the number of rows per bank as well as the refresh period. The RFSH field represents how many external clock cycles remain before an AUTO-REFRESH is required. The normal formula is:

$$RFSH \leq \frac{t_{refresh\_us}/number\_rows}{ext\_clock\_period\_us}$$

RFSH ≤ ($t_{Refresh\_us}$/ number_rows) / ext_clock_period_µs

A refresh period is normally 64ms, or 64000µs. The number of rows is normally 4096 or 8192. The ext_clock_period is a value expressed in µs and is derived by dividing 1000 by the clock speed expressed in MHz. So, 50MHz is 1000 / 50 = 20ns, or 0.02µs. A typical SDRAM is 4096 rows per bank if the system clock is running at 50MHz with an EPIBAUD register value of 0:

$$RFSH = \frac{64000 \ / \ 4096}{0.02} = \frac{15.625\mu s}{0.02\mu s} = 781.25$$

RFSH = (64000 / 4096) / 0.02 = 15.625µs / 0.02µs = 781.25.

The default value in the RFSH field is 750 decimal or 0x2EE to allow for a margin of safety and providing 15µs per refresh. It is important to note that this number must always be smaller or equal to what is required by the above equation. For example, if running the external clock at 25MHz (40ns per clock period), 390 is the highest number that can be used. The external clock can be 25MHz when the system clock is 25MHz or when the system clock is 50MHz and configuring the COUNT0 field in the EPIBAUD register to 1 (divide by 2).

If a number larger than allowed is used, the SDRAM is not refreshed often enough, and data is lost.

28.1.4.2.3 Bus Interface Speed

The EPI Controller SDRAM interface can operate up to 60 MHz. The COUNT0 field in the EPIBAUD register configures the speed of the EPI clock. For main clock (MCLK) speeds up to 60 MHz, the COUNT0 field can be 0x0000, and the SDRAM interface can run at the same speed as MCLK. However, if MCLK is running at higher speeds, the bus interface can run only as fast as half speed, and the COUNT0 field must be configured to at least 0x0001.

28.1.4.2.4 Nonblocking Read Cycle

Figure 28-2 shows a nonblocking read cycle of n halfwords; n can be any number greater than or equal to 1. The cycle begins with the Activate command and the row address on the EPI0S[15:0] signals. With the programmed CAS latency of 2, the Read command with the column address on the EPI0S[15:0] signals follows after 2 clock cycles. Following one more NOP cycle, data is read in on the EPI0S[15:0] signals on every rising clock edge. The Burst Terminate command is issued during the cycle when the next-to-last halfword is read in. The DQMH and DQML signals are deasserted after the last halfword of data is received; the CSn signal deasserts on the following clock cycle, signaling the end of the read cycle. At least one clock period of inactivity separates any two SDRAM cycles.

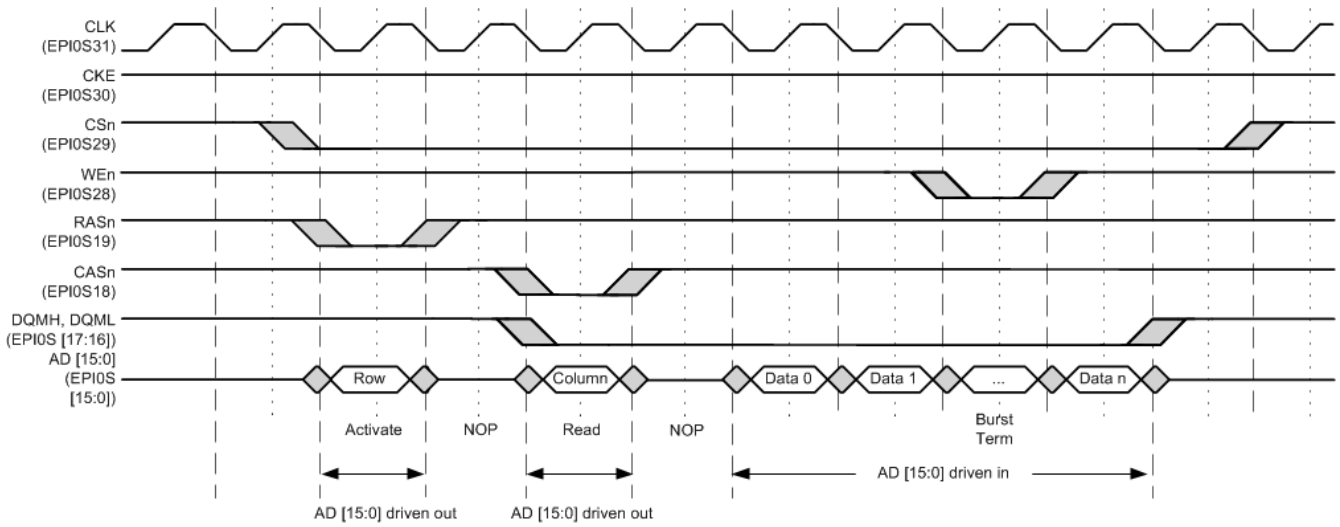


Figure 28-2. SDRAM Nonblocking Read Cycle

28.1.4.2.5 Normal Read Cycle

Figure 28-3 shows a normal read cycle of n halfwords; n can be 1 or 2. The cycle begins with the Activate command and the row address on the EPI0S[15:0] signals. With the programmed CAS latency of 2, the Read command with the column address on the EPI0S[15:0] signals follows after 2 clock cycles. Following one more NOP cycle, data is read in on the EPI0S[15:0] signals on every rising clock edge. The DQMH, DQML, and CSn signals are deasserted after the last halfword of data is received, signaling the end of the cycle. At least one clock period of inactivity separates any two SDRAM cycles.

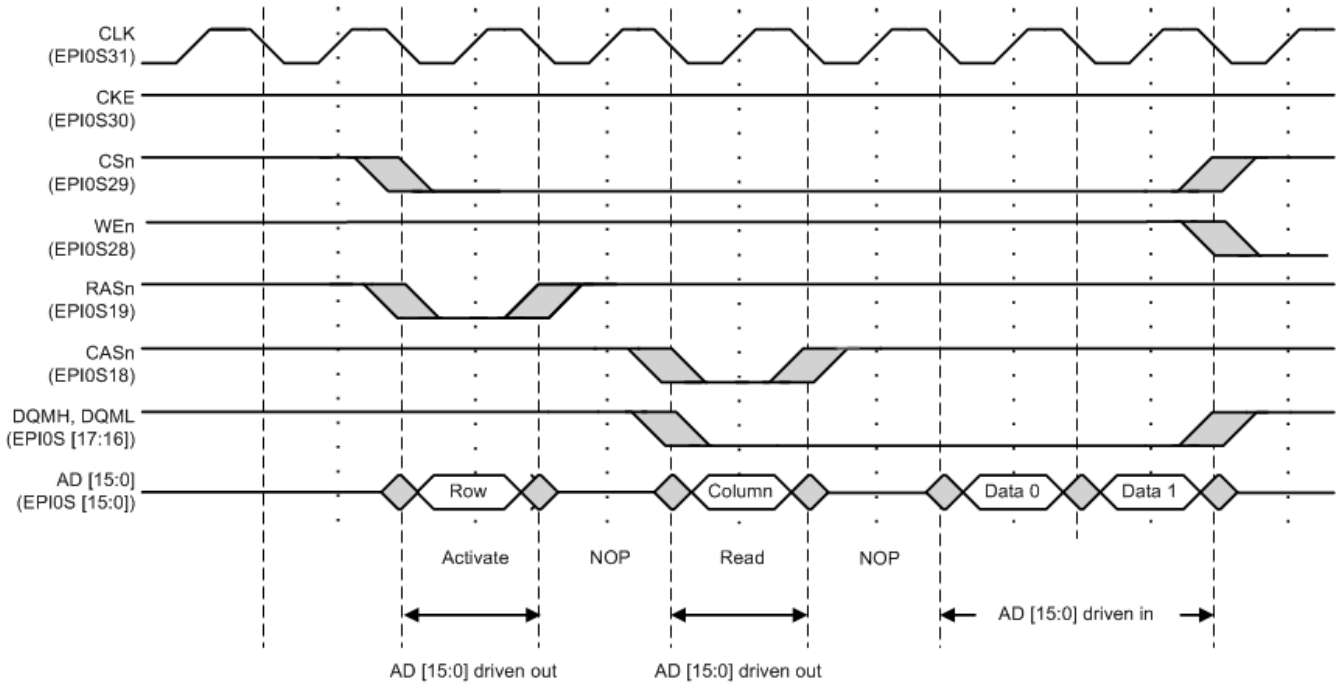


Figure 28-3. SDRAM Normal Read Cycle

28.1.4.2.6 Write Cycle

Figure 28-4 shows a write cycle of n halfwords; n can be any number greater than or equal to 1. The cycle begins with the Activate command and the row address on the EPI0S[15:0] signals. With the programmed CAS latency of 2, the Write command with the column address on the EPI0S[15:0] signals follows after 2 clock cycles. When writing to SDRAMs, the Write command is presented with the first halfword of data. Because the address lines and the data lines are multiplexed, the column address is modified to be (programmed address -1). During the Write command, the DQMH and DQML signals are high, so no data is written to the SDRAM. On the next clock, the DQMH and DQML signals are asserted, and the data associated with the programmed address is written. The Burst Terminate command occurs during the clock cycle following the write of the last halfword of data. The WEn, DQMH, DQML, and CSn signals are deasserted after the last halfword of data is received, signaling the end of the access. At least one clock period of inactivity separates any two SDRAM cycles.

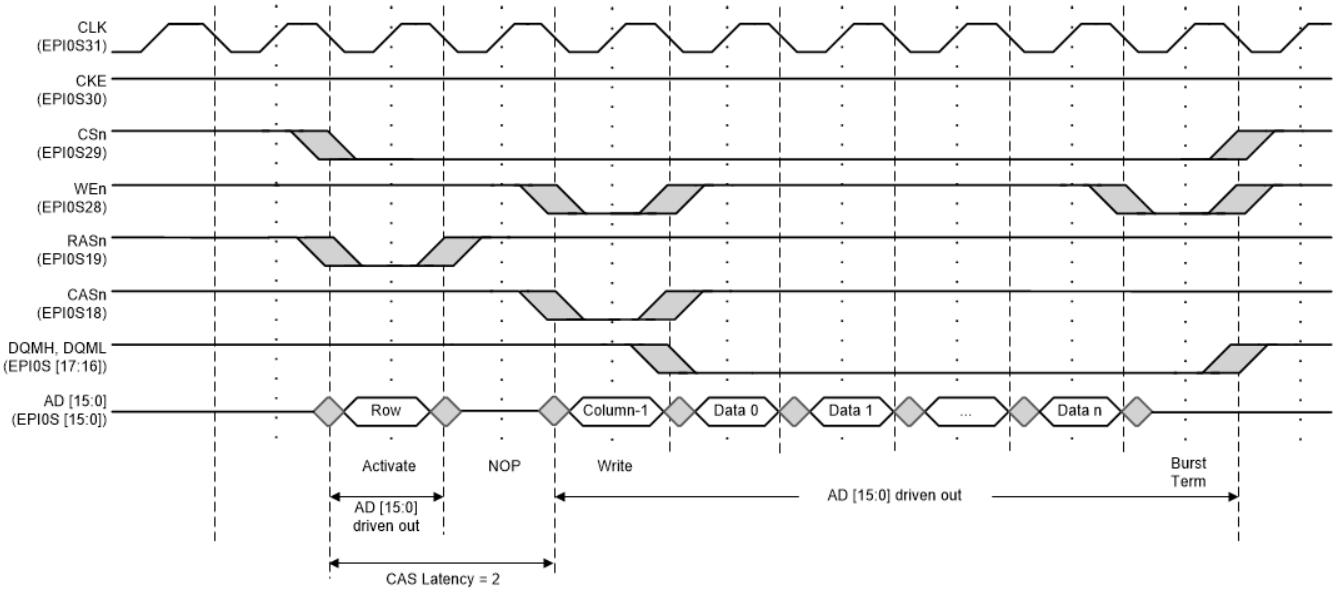


Figure 28-4. SDRAM Write Cycle

28.1.4.3 Host Bus Mode

Host Bus supports the traditional 8-bit and 16-bit interfaces popularized by the 8051 devices and SRAM devices, as well as PSRAM and NOR Flash memory. This interface is asynchronous and uses strobe pins to control activity. Addressable memory can be doubled using Host Bus-16 mode as it performs halfword accesses. The EPI0S0 is the LSB of the address and is equivalent to the internal Cortex-M33 A1 address. EPI0S0 should be connected to A0 of 16-bit memories.

28.1.4.3.1 Control Pins

The main three strobes are Address Latch Enable (ALE), Write (WRn), and Read (RDn, sometimes called OEn).The polarity of the read and write strobes can be active-high or active-low by clearing or setting the RDHIGH and WRHIGH bits in the EPI Host-Bus n Configuration (EPIHBnCFGn) register.

The ALE can be changed to an active-low chip select signal, CSn, through the EPIHBnCFGn register. The ALE is best used for Host-Bus muxed mode in which EPI address and data pins are shared. All Host-Bus accesses have an address phase followed by a data phase. The ALE indicates to an external latch to capture the address then hold it until the data phase. The polarity of the ALE can be active High or Low by clearing or setting the ALEHIGH bit in the EPI Host-Bus n Configuration (EPIHBnCFGn) register. CSn is best used for Host-Bus unmuxed mode in which EPI address and data pins are separate. The CSn indicates when the address and data phases of a read or write access are occurring. Both the ALE and the CSn modes can be enhanced to access four external devices using settings in the EPIHBnCFGn register. PSRAM accesses must use both ALE and CSn. Wait states can be added to the data phase of the access using the WRWS and RDWS bits

in the EPIHBnCFGn register. Additionally, within these wait state options, the WRWSM and RDWSM bit of the EPIHBnTIMEn register can be set to reduce the given wait states by 1 EPI clock cycle for finer granularity.

For FIFO mode, the ALE is not used, and two input holds are optionally supported to gate input and output to what the XFIFO can handle. FIFO mode is only applicable in EPI asynchronous mode.

Host-bus 8 and host-bus 16 modes are very configurable. The user has the ability to connect 1, 2, or 4 external devices to the EPI signals, as well as control whether byte select signals are provided in HB16 mode. These capabilities depend on the configuration of the MODE field in the EPIHBnCFG register, the CSCFG field and the CSCFGEXT bit in the EPIHBnCFGn register, and the BSEL bit in the EPIHB16CFG register. The CSCFGEXT bit extends the chip select configuration possibilities by providing the most significant bit of the CSCFGEXT field. For the possible ALE and chip select options that can be programmed by the combination of the CSCFGEXT and CSCFGEXT bits, see Table 28-3. CSCFGEXT is the most significant bit.

Table 28-3. CSCFGEXT and CSCFG Encodings

Value	Description
0x0	ALE configuration EPI0S30 is used as an address latch (ALE). The ALE signal is generally used when the address and data are muxed (MODE field in the EPIHB8CFG register is 0x0). The ALE signal is used by an external latch to hold the address through the bus cycle.
0x1	CSn Configuration EPI0S30 is used as a chip select (CSn). When using this mode, the address and data are generally not muxed (MODE field in the EPIHB8CFG register is 0x1). However, if address and data muxing is needed, the WR signal (EPI0S29) and the RD signal (EPI0S28) can be used to latch the address when CSn is low.
0x2	Dual CSn configuration EPI0S30 is used as CS0n and EPI0S27 is used as CS1n. Whether CS0n or CS1n is asserted is determined by the most significant address bit for a respective external address map. This configuration can be used for a RAM bank split between 2 devices and when using both an external RAM and an external peripheral.
0x3	ALE with dual CSn configuration EPI0S30 is used as address latch (ALE), EPI0S27 is used as CS1n, and EPI0S26 is used as CS0n. Whether CS0n or CS1n is asserted is determined by the most significant address bit for a respective external address map.
0x4	ALE with single CSn configuration EPI0S30 is used as address latch (ALE) and EPI0S27 is used as CSn.
0x5	Tri CSn configuration EPI0S30 is used as CS0n, EPI0S27 is used as CS1n, EPI0S34 is used as CS2n.
0x6	ALE with tri CSn configuration EPI0S30 is used as ALE, EPI0S26 is used as CS0n, EPI0S27 is used as CS1n, EPI0S34 is used as CS2n.
0x7	Reserved

If one of the dual-chip-select modes is selected (CSCFGEXT is 0x0 and CSCFG is 0x2 or 0x3 in the EPIHBnCFGn register), both chip selects can share the peripheral, code, or the memory space, or one chip select can use the peripheral space and the other can use the memory or code space. In the EPIADDRMAP register, if the EPADR field is not 0x0, the ECADR field is 0x0, and the ERADR field is 0x0, then the address specified by EPADR is used for both chip selects, with CS0n being asserted when the MSB of the address range is 0 and CS1n being asserted when the MSB of the address range is 1. If the ERADR field is not 0x0, the ECADR field is 0x0, and the EPADR field is 0x0, then the address specified by ERADR is used for both chip selects, with the MSB performing the same delineation. If both the EPADR and the ERADR are not 0x0, and the ECADR field is 0x0 and the EPI is configured for dual-chip selects, then CS0n is asserted for either address range defined by EPADR and CS1n is asserted for either address range defined by ERADR. The two chip selects can also be shared between the code space and memory or peripheral space. If the ECADR field is 0x1, ERADR field is 0x0, and the EPADR field is not 0x0, then CS0n is asserted for the address range defined by ECADR and CS1n is asserted for either address range defined by EPADR. If the ECADR field is 0x1, EPADR field is 0x0, and the ERADR field is not 0x0, then CS0n is asserted for the address range defined by ECADR and CS1n is asserted for either address range defined by ERADR.

In tri chip select mode (CSCFGEXT is 0x1 and CSCFG is 0x1 or 0x2 in the EPIHBnCFG2 register), both the peripheral and the memory space must be enabled. In the EPIADDRMAP register, the EPADR field is 0x3, the ERADR field is 0x3, and the ECADR field is 0x0. With this configuration, CS0n asserts for the address range beginning at 0x8000.0000, CS1n asserts for 0xA000.0000, and CS2n for 0xC000.0000. Table 28-4 gives a detailed explanation of chip select address range mappings based on combinations of enabled peripheral and memory space.

Note

Only one memory area can be mapped to a single chip select. Enabling multiple memory areas for one chip select can produce unexpected results.

Table 28-4. Dual-Chip and Tri-Chip Select Address Mappings

Chip Select Mode	ERADR	EPADR	ECADR	CS0 (1)	CS1 (1)	CS2	CS3
Dual-chip select	0x0	0x1 or 0x2	0x0	EPADR defined address range (0xA000.000 or 0xC000.0000)	EPADR defined address range (0xA000.000 or 0xC000.0000)	N/A	N/A
Dual-chip select	0x1 or 0x2	0x0	0x0	ERADR defined address range (0x6000.000 or 0x8000.000)	ERADR defined address range (0x6000.000 or 0x8000.000)	N/A	N/A
Dual-chip select	0x1 or 0x2	0x1 or 0x2	0x0	EPADR defined address range (0xA000.000 or 0xC000.0000)	ERADR defined address range (0x6000.000 or 0x8000.000)	N/A	N/A

(1) When CS0 and CS1 share address space, CS0 asserts when the MSB of the address is 0 and CS1 asserts when the MSB of the address is 1.

The MODE field of the EPIHBnCFGn registers configure the interface for the chip selects, which support ADMUX, where data and address are muxed, or ADNONMUX, where data and address or separate. See Table 28-5 for details on which configuration register controls each chip select. If the CSBAUD bit is clear, all chip selects are configured by the MODE bit field of the EPIHBnCFG register.

If the CSBAUD bit in the EPIHBnCFG2 register is set in Dual-chip select mode, the 2 chip selects can use different clock frequencies, wait states and strobe polarity. If the CSBAUD bit is clear, both chip selects use the clock frequency, wait states, and strobe polarity defined for CS0n. Additionally, if the CSBAUD bit is set, the two chip selects can use different interface modes. If any interface modes are programmed to ADMUX, then dual chip select mode must include the ALE capability. In quad chip select mode, if the CSBAUD bit in the EPIHBnCFG2 register is set, the 4 chip selects can use different clock frequencies, wait states and strobe polarity. If the CSBAUD bit is clear, all chip selects use the clock frequency, wait states, and strobe polarity defined for CS0n. If the CSBAUD bit is set, the four chip selects can use different interface modes.

Table 28-5. Chip Select Configuration Register Assignment

Configuration Register (1)	Corresponding Chip Select
EPIHBnCFG	CS0n
EPIHBnCFG2	CS1n
EPIHBnCFG3	CS2n
EPIHBnCFG4	CS3n

(1) If the CSBAUD bit in the EPIHBnCFG2 register is clear and multiple chip selects are enabled, then all chip selects are configured by the MODE bit field in the EPIHBnCFG register.

Multiple chip select modes do not allow the intermixing of Host-Bus 8 and Host-Bus16 modes.

When BSEL = 1 in the EPIHB16CFG register, byte select signals are provided, so byte-sized data can be read and written at any address, however these signals reduce the available address width by 2 pins. The byte select signals are active Low. BSEL0n corresponds to the LSB of the halfword, and BSEL1n corresponds to the MSB of the halfword.

When BSEL = 0, byte reads and writes at odd addresses only act on the even byte, and byte writes at even addresses write invalid values into the odd byte. As a result, accesses must be made as halfwords (16-bits) or words (32-bits). In C/C++, programmers must use only short int and long int for accesses. Also, because data accesses in HB16 mode with no byte selects are on 2-byte boundaries, the available address space is doubled. For example, 28 bits of address accesses 512MB in this mode. Table 28-6 shows the capabilities of the HB8 and HB16 modes as well as the available address bits with the possible combinations of these bits.

Although the EPI0S31 signal can be configured for the EPI clock signal in Host-Bus mode, it is not required and must be configured as a GPIO to reduce EMI in the system.

Table 28-6. Capabilities of Host Bus 8 and Host Bus 16 Modes

Host Bus Type	MODE	CSCFGEXT	CSCFG	Maximum No. of External Devices	BSEL	Byte Access	Available Address	Addressable Memory
HB8	0x0	0	0x0, 0x1	1	N/A	Always	28 bits	256MB
HB8	0x0	0	0x2	2	N/A	Always	27 bits	128MB
HB8	0x0	0	0x3	2	N/A	Always	26 bits	64MB
HB8	0x0	1	0x0	1	N/A	Always	27 bits	128MB
HB8	0x0	1	0x1	3	N/A	Always	27 bits	128MB
HB8	0x0	1	0x2	3	N/A	Always	26 bits	64MB
HB8	0x1	0	0x0, 0x1	1	N/A	Always	20 bits	1MB
HB8	0x1	0	0x2	2	N/A	Always	19 bits	512KB
HB8	0x1	0	0x3	2	N/A	Always	18 bits	256KB
HB8	0x1	1	0x0	1	N/A	Always	19 bits	512KB
HB8	0x1	1	0x1	3	N/A	Always	19 bits	512MB
HB8	0x1	1	0x2	3	N/A	Always	18 bits	256KB
HB8	0x2	0	0x1	1	N/A	Always	20 bits	1MB
HB8	0x3	0	0x1	1	N/A	Always	none	–
HB8	0x3	0	0x3	2	N/A	Always	none	–
HB8	0x3	1	0x0	1	N/A	Always	none	–
HB8	0x3	1	0x1	3	N/A	Always	none	–

ADVANCE INFORMATION

Table 28-6. Capabilities of Host Bus 8 and Host Bus 16 Modes (continued)

Host Bus Type	MODE	CSCFGEXT	CSCFG	Maximum No. of External Devices	BSEL	Byte Access	Available Address	Addressable Memory
HB8	0x3	1	0x2	3	N/A	Always	none	–
HB16	0x0	0	0x0, 0x1	1	0	No	28 bits (1)	512MB
HB16	0x0	0	0x0, 0x1	1	1	Yes	26 bits (2)	128MB
HB16	0x0	0	0x2	2	0	No	27 bits (1)	256MB
HB16	0x0	0	0x2	2	1	Yes	25 bits (2)	64MB
HB16	0x0	0	0x3	2	0	No	26 bits (1)	128MB
HB16	0x0	0	0x3	2	1	Yes	24 bits (2)	32MB
HB16	0x0	1	0x0	1	0	No	27 bits (1)	256MB
HB16	0x0	1	0x0	1	1	Yes	25 bits (2)	128MB
HB16	0x0	1	0x1	3	0	No	27 bits (1)	256MB
HB16	0x0	1	0x1	3	1	Yes	25 bits (2)	64MB
HB16	0x0	1	0x2	3	0	No	26 bits (1)	128MB
HB16	0x0	1	0x2	3	1	Yes	24 bits (2)	32MB
HB16	0x1	0	0x0, 0x1	1	0	No	12 bits (1)	8KB
HB16	0x1	0	0x0, 0x1	1	1	Yes	10 bits (2)	2KB
HB16	0x1	0	0x2	2	0	No	11 bits (1)	4KB
HB16	0x1	0	0x2	2	1	Yes	9 bits (2)	1KB
HB16	0x1	0	0x3	2	0	No	10 bits (1)	2KB
HB16	0x1	0	0x3	2	1	Yes	8 bits (2)	512 B
HB16	0x1	1	0x0	1	0	No	11 bits (1)	4KB
HB16	0x1	1	0x0	1	1	Yes	9 bits (2)	1KB
HB16	0x1	1	0x1	3	0	No	11 bits (1)	4KB
HB16	0x1	1	0x1	3	1	Yes	9 bits (2)	1KB
HB16	0x1	1	0x2	3	0	No	10 bits (1)	2KB
HB16	0x1	1	0x2	3	1	Yes	8 bits (2)	512 B
HB16	0x3	0	0x1	1	0	No	none	–
HB16	0x3	0	0x1	1	1	Yes	none	–
HB16	0x3	0	0x3	2	0	No	none	–
HB16	0x3	0	0x3	2	1	Yes	none	–
HB16	0x3	1	0x0	1	0	No	none	–
HB16	0x3	1	0x0	1	1	Yes	none	–
HB16	0x3	1	0x1	3	0	No	none	–
HB16	0x3	1	0x1	3	1	Yes	none	–
HB16	0x3	1	0x2	3	0	No	none	–

ADVANCE INFORMATION

Table 28-6. Capabilities of Host Bus 8 and Host Bus 16 Modes (continued)

Host Bus Type	MODE	CSCFGEXT	CSCFG	Maximum No. of External Devices	BSEL	Byte Access	Available Address	Addressable Memory
HB16	0x3	1	0x2	3	1	Yes	none	–

(1) If byte selects are not used, data accesses are on 2-byte boundaries. As a result, the available address space is doubled.
(2) Two EPI signals are used for byte selects, reducing the available address space by two bits.

Table 28-7 shows how the EPI[31:0] signals function while in Host-Bus 8 mode. Notice that the signal configuration changes based on the address/data mode selected by the MODE field in the EPIHB8CFGn register and on the chip select configuration selected by the CSCFG and CSCFGEXT field in the EPIHB8CFG2 register.

Although the EPI0S31 signal can be configured for the EPI clock signal in Host-Bus mode, it is not required and must be configured as a GPIO to reduce EMI in the system. Any unused EPI controller signals can be used as GPIOs or another alternate function.

Table 28-7. EPI Host-Bus 8 Signal Connections

EPI Signal	CSCFG (1)	HB8 Signal (MODE = ADMUX) (1)	HB8 Signal (MODE = ADNOMUX (Cont. Read)) (1)	HB8 Signal (MODE = XFIFO) (1)
EPI0S0	X	AD0	D0	D0
EPI0S1	X	AD1	D1	D1
EPI0S2	X	AD2	D2	D2
EPI0S3	X	AD3	D3	D3
EPI0S4	X	AD4	D4	D4
EPI0S5	X	AD5	D5	D5
EPI0S6	X	AD6	D6	D6
EPI0S7	X	AD7	D7	D7
EPI0S8	X	A8	A0	–
EPI0S9	X	A9	A1	–
EPI0S10	X	A10	A2	–
EPI0S11	X	A11	A3	–
EPI0S12	X	A12	A4	–
EPI0S13	X	A13	A5	–
EPI0S14	X	A14	A6	–
EPI0S15	X	A15	A7	–
EPI0S16	X	A16	A8	–
EPI0S17	X	A17	A9	–
EPI0S18	X	A18	A10	–
EPI0S19	X	A19	A11	–
EPI0S20	X	A20	A12	–
EPI0S21	X	A21	A13	–
EPI0S22	X	A22	A14	–
EPI0S23	X	A23	A15	–
EPI0S24	X	A24	A16	–
EPI0S25	0x0	A25 (2)	A17	–
	0x1			–
	0x2			CS1n
	0x3			–
	0x4			–
	0x5			–
	0x6			–
EPI0S26	0x0	A26	A18	FEMPTY
	0x1			
	0x2			
	0x3	CS0n	CS0n	

ADVANCE INFORMATION

Table 28-7. EPI Host-Bus 8 Signal Connections (continued)

EPI Signal	CSCFG [1]	HB8 Signal (MODE = ADMUX) [1]	HB8 Signal (MODE = ADNOMUX (Cont. Read)) [1]	HB8 Signal (MODE = XFIFO) [1]
	0x4	A26	A18	
	0x5			
	0x6	CS0n	CS0n	
EPI0S27	0x0	A27	A19	FFULL
	0x1			
	0x2	CS1n	CS1n	
	0x3			
	0x4	CS0n	CS0n	
	0x5	CS1n	CS1n	
	0x6			
EPI0S28	X	RDn/OEn	RDn/OEn	RDn
EPI0S29	X	WRn	WRn	WRn
EPI0S30	0x0	ALE	ALE	–
	0x1	CSn	CSn	CSn
	0x2	CS0n	CS0n	CS0n
	0x3	ALE	ALE	–
	0x4			–
	0x5	CS0n	CS0n	–
	0x6	ALE	ALE	–
EPI0S31	X	Clock [3]		Clock [3]
EPI0S32	X	iRDY	iRDY	iRDY
EPI0S33	0x0	X	X	X
	0x1	X	X	X
	0x2	X	X	X
	0x3	X	X	X
	0x4	X	X	X
	0x5	CS3n	CS3n	X
	0x6			X
EPI0S34	0x0	X	X	X
	0x1	X	X	X
	0x2	X	X	X
	0x3	X	X	X
	0x4	X	X	X
	0x5	CS2n	CS2n	X
	0x6			X
EPI0S35	0x0	X	X	X
	0x1	X	X	X
	0x2	X	X	X
	0x3	X	X	X
	0x4	X	X	X
	0x5	CRE	CRE	X

ADVANCE INFORMATION

Table 28-7. EPI Host-Bus 8 Signal Connections (continued)

EPI Signal	CSCFG (1)	HB8 Signal (MODE = ADMUX) (1)	HB8 Signal (MODE = ADNOMUX (Cont. Read)) (1)	HB8 Signal (MODE = XFIFO) (1)
	0x6			X

(1) "X" indicates the state of this field is a don't care.
(2) When an entry straddles several row, the signal configuration is the same for all rows.
(3) The clock signal is not required for this mode.

Table 28-8 shows how the EPI[31:0] signals function while in host-bus 16 mode. The signal configuration changes based on the address/data mode selected by the MODE field in the EPIHB16CFGn register, on the chip select configuration selected by the CSCFG and CSCFGEXT field in the same register, and on whether byte selects are used as configured by the BSEL bit in the EPIHB16CFG register.

Although the EPI0S31 signal can be configured for the EPI clock signal in host-bus mode, it is not required and must be configured as a GPIO to reduce EMI in the system. Any unused EPI controller signals can be used as GPIOs or another alternate function.

Table 28-8. EPI Host-Bus 16 Signal Connections

EPI Signal	CSCFG (1)	BSEL (1)	HB16 Signal (MODE = ADMUX)	HB16 Signal (MODE = ADNOMUX (Cont. Read))	HB16 Signal (MODE = XFIFO) (1)
EPI0S0	X	X	AD0 (2)	D0	D0
EPI0S1	X	X	AD1	D1	D1
EPI0S2	X	X	AD2	D2	D2
EPI0S3	X	X	AD3	D3	D3
EPI0S4	X	X	AD4	D4	D4
EPI0S5	X	X	AD5	D5	D5
EPI0S6	X	X	AD6	D6	D6
EPI0S7	X	X	AD7	D7	D7
EPI0S8	X	X	AD8	D8	D8
EPI0S9	X	X	AD9	D9	D9
EPI0S10	X	X	AD10	D10	D10
EPI0S11	X	X	AD11	D11	D11
EPI0S12	X	X	AD12	D12	D12
EPI0S13	X	X	AD13	D13	D13
EPI0S14	X	X	AD14	D14	D14
EPI0S15	X	X	AD15	D15	D15
EPI0S16	X	X	A16	A0 (2)	–
EPI0S17	X	X	A17	A1	–
EPI0S18	X	X	A18	A2	–
EPI0S19	X	X	A19	A3	–
EPI0S20	X	X	A20	A4	–
EPI0S21	X	X	A21	A5	–
EPI0S22	X	X	A22	A6	–
EPI0S23	X(3)	0	A23	A7	–
		1			
EPI0S24	0x0	0	A24	A8	–
		1			

ADVANCE INFORMATION

Table 28-8. EPI Host-Bus 16 Signal Connections (continued)

EPI Signal	CSCFG [1]	BSEL [1]	HB16 Signal (MODE = ADMUX)	HB16 Signal (MODE = ADNOMUX (Cont. Read))	HB16 Signal (MODE = XFIFO) [1]
	0x1	0			
		1			
	0x2	0			
		1			
	0x3	0			
		1	BSEL0n	BSEL0n	
	0x4	0	A24	A8	
		1			
	0x5	0			
		1			
	0x6	0			
		1	BSEL0n	BSEL0n	
EPI0S25	0x0	X	A25	A9	–
	0x1				
	0x2	0	A25	A9	CS1n
		1	BSEL0n	BSEL0n	
	0x3	0	A25	A9	–
		1	BSEL1n	BSEL1n	
	0x4	0	A25	A9	–
		1	BSEL0n	BSEL0n	
	0x5	0	A25	A9	–
		1	BSEL0n	BSEL0n	
	0x6	0	A25	A9	–
		1	BSEL1n	BSEL1n	
EPI0S26	0x0	0	A26	A10	FEMPTY
		1	BSEL0n	BSEL0n	
	0x1	0	A26	A10	
		1	BSEL0n	BSEL0n	
	0x2	0	A26	A10	
		1	BSEL1n	BSEL1n	
	0x3	X	CS0n	CS0n	
	0x4	0	A26	A10	–
		1	BSEL1n	BSEL1n	
	0x5	0	A26	A10	–
		1	BSEL1n	BSEL1n	
	0x6	0	CS0n	CS0n	–
		1			
EPI0S27	0x0	0	A27	A11	FFULL
		1	BSEL1n	BSEL1n	
	0x1	0	A27	A11	
		1	BSEL1n	BSEL1n	
	0x2	X	CS1n	CS1n	
	0x3	X	CS1n	CS1n	
	0x4	X	CS0n	CS0n	–

ADVANCE INFORMATION

Table 28-8. EPI Host-Bus 16 Signal Connections (continued)

EPI Signal	CSCFG (1)	BSEL (1)	HB16 Signal (MODE = ADMUX)	HB16 Signal (MODE = ADNOMUX (Cont. Read))	HB16 Signal (MODE = XFIFO) (1)
	0x5	X	CS1n	CS1n	–
	0x6	X	CS1n	CS1n	–
EPI0S28	X	X	RDn/OEn	RDn/OEn	RDn
EPI0S29	X	X	WRn	WRn	WRn
EPI0S30	0x0	X	ALE	ALE	–
	0x1	X	CSn	CSn	CSn
	0x2	X	CS0n	CS0n	CS0n
	0x3	X	ALE	ALE	–
	0x4	X	ALE	ALE	–
	0x5	X	CS0n	CS0n	–
	0x6	X	ALE	ALE	–
EPI0S31	X	X	Clock (4)	Clock (4)	Clock (4)
EPI0S32	X	X	iRDY	iRDY	iRDY
EPI0S33	X	X	CS3n	CS3n	X
EPI0S34	X	X	CS2n	CS2n	X
EPI0S35	X	X	CRE	CRE	X

(1) X = don't care
(2) In this mode, halfword accesses are used. A0 is the LSB of the address and is equivalent to the internal Cortex-M33 A1 address. This pin must be connected to A0 of 16-bit memories.
(3) When an entry straddles several rows, the signal configuration is the same for all rows.
(4) The clock signal is not required for this mode.

The RDYEN in the EPIHBnCFG enables the monitoring of the external iRDY pin to stall accesses. On the rising edge of EPI clock, if iRDY is low, access is stalled. The IRDYDLY can program the number of EPI clock cycles in advance to the stall (1, 2, or 3) (see Figure 28-5). This is a conceptual timing diagram of how the iRDY signal works with different IRDYDLY configurations. When enabled, the iRDY stalls the internal states of the EPI, while IRDYDLY controls the delay pipeline when this stall takes affect. The iRDY signal can be connected to multiple devices with a pullup resistor (see Figure 28-6). When multiple PSRAMs are connected to iRDY, the EPIHPnCFG registers must be programmed to the same iRDY signal polarity through the IRDYINV bit. When connected to a PSRAM, iRDY is used to control the address to data latency.

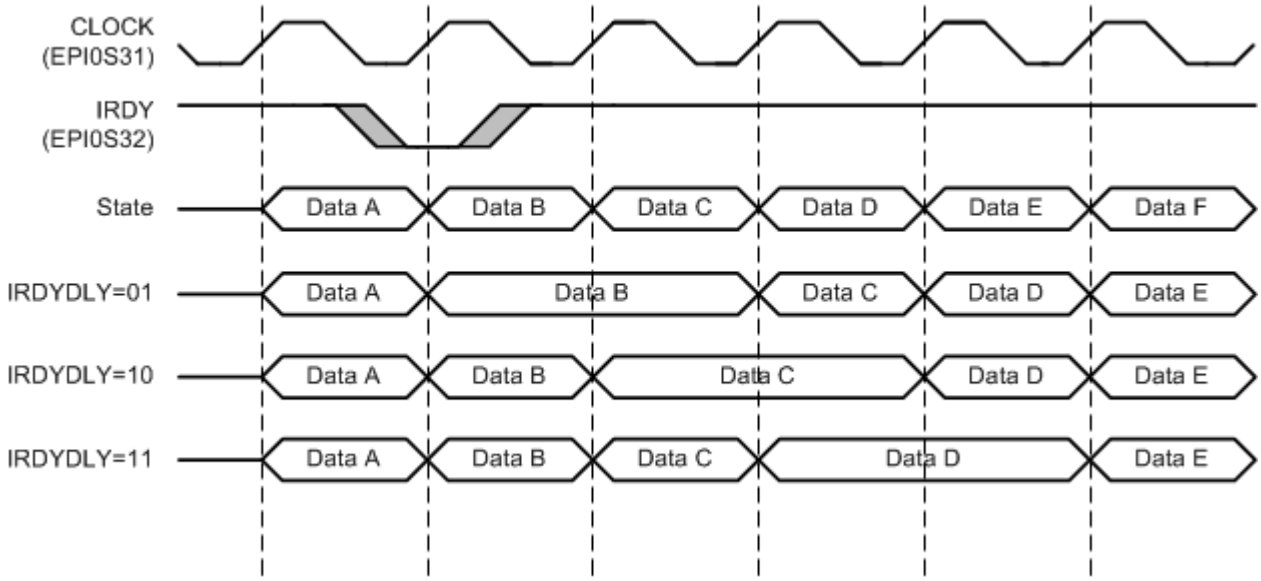


Figure 28-5. iRDY Access Stalls, IRDYDLY = 01, 10, 11

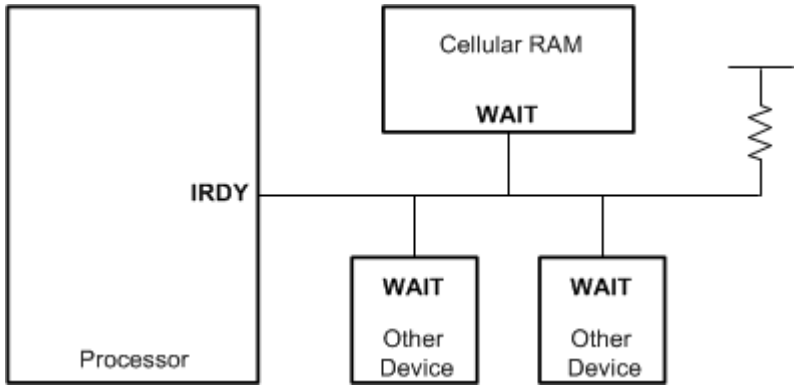


Figure 28-6. iRDY Signal Connection

28.1.4.3.2 PSRAM Support

The EPI host bus supports both a synchronous and asynchronous interface to PSRAM memory when configured in 16-bit bus multiplexed mode. The EPIHBPSRAM register holds the values for the PSRAM bus configuration (CR) registers. The contents of the EPIHBPSRAM register can be sent to different memories depending on which WRCRE or RDCRE bit is set in the various EPIHB16CFGn registers. For example, if the WRCRE bit is enabled in EPIHB16CFG, then the CRE signal asserts and the contents are sent to the memory enabled by CS0. Enabling the WRCRE or RDCRE bit in EPIHB16CFG2 register activates CS1n during a PSRAM configuration register write or read. The WRCRE and RDCRE bit in EBIHB16CFG3 corresponds to CS2n and EPIHB16CFG4, to CS3n. The WRCRE bit clears when the transfer is done. There must not be any system access or nonblocking read activity during the CRE read or write-enable transfer. During a write to the PSRAM's CR, the configuration data is written out on data pins [20:0] of the EPI bus. For a PSRAM configuration read access, the RDCRE bit in the EPIHB16CFG register is set to signal that the next access is a read of the PSRAM configuration register (CR). The address for the CR is written to bits CR[19:18] of the EPIHBPSRAM register. The read data is returned at CR bits [15:0] of the EPIHBPSRAM register.

Note

- CRE read and write operations can only occur in asynchronous mode. During synchronous mode the CRE bit must be disabled. Setting the CRE bit during synchronous PSRAM accesses can lead to unpredictable behavior.
- When the chip select is programmed to access the PSRAM, the MODE bit of the EPIHBnCFGn register must be programmed to enable address and data muxed (ADMUX). Page mode accesses are not supported by the EPI.
- BURST is optimized for word-length bursting for SDRAM and PSRAM accesses.

These steps initialize the PSRAM interface:

1. Perform the EPI initialization steps in Section 28.1.4 .
2. Enable host bus 16 mode by setting the MODE bits in the EPICFG register to 0x13. Choose between an integer or formula clock divide for the baud rate by configuring the INTDIV bit in the EPICFG register.
3. Configure the EPIBAUD register to the desired baud rate.
4. Because the EPI module supports only asynchronous programming of the configuration registers, clock gate the EPI clock by programming both the CLKGATE and CLKGATEI bits in the EPIHB16CFG register to 0.
5. Prepare for writing the PSRAM Bus Configuration register by setting the ALEHIGH = 1 and MODE = 0x0 in the EPIHB16CFG register.
6. Program the EPIHBPSRAM register to be loaded into the CR register of the PSRAM by configuring bits [21:0].
   a. CR[20:19] = 0x0, reserved
   b. CR[19:18] = 0x2 to enable configuring of the CR register
   c. CR[15]= 0x1 to enable asynchronous access
   d. CR [14] = 0 if the iRDY signal is used for memory transfers; if the design does not use the iRDY signal CR[14] must be cleared.
   e. CR[13:11] must be programmed to have a matching read and write wait state configuration as is programmed in the EPIHB16CFG and EPIHB16TIME register.
   f. CR[10] configures the polarity of the WAIT signal and must match the configuration of the IRDYINV bit in the EPIHB16CFG register.
   g. CR[8]= 0x1 to configure the appropriate wait configuration of the data
   h. CR[2:0]= 0x7 since the EPI interface in PSRAM mode is a continuous burst access.
7. Set the WRCRE bit in the EPIHB16CFGn register to initiate a write from the EPIHBPSRAM register to the PSRAM CR register.

Note

If the PSRAM CR register must be reprogrammed after initialization, the application must allow the previous transfer to complete before beginning configuration to make sure proper PSRAM functionality.

Table 28-9. PSRAM Fixed Latency Wait State Configuration

Latency Counter	Latency in Clocks	RDWS[1:0]/WRWS[1:0]	RWSM/WRWSM
BCR Code 2	3	0x0	0
BCR Code 3	4	0x1	1
BCR Code 4	5	0x1	0
BCR Code 5	6	0x2	1
BCR Code 6	7	0x2	0
BCR Code 8	9	0x3	0

In variable initial latency mode, the memory's WAIT (iRDY) pin guides the EPI module when to read and write. The WAIT (iRDY) pin stalls the access for the duration of the latency and adds cycles if there is a refresh collision. To get the best performance, set CR[13:11] = 0x2, the WRWS field of the EPIHB16CFG register to 0x0, and the WRWSM and RDWSM bit of the EPI16TIMEn register to 0. For the WAIT pin to be recognized correctly, set the IRDYDLY bit in the EPI16TIMEn register to 1 and the CR[8] = 1 in the EPIHBPSRAM register.

Note

Wait state latency works differently in PSRAM Burst mode than in other modes. In PSRAM Burst mode, the RDWS and WRWS bit fields define the latency for only the first access of the write or read cycle. Every access after that is a single access.

Figure 28-7 and Figure 28-8 depict a PSRAM burst read and write.

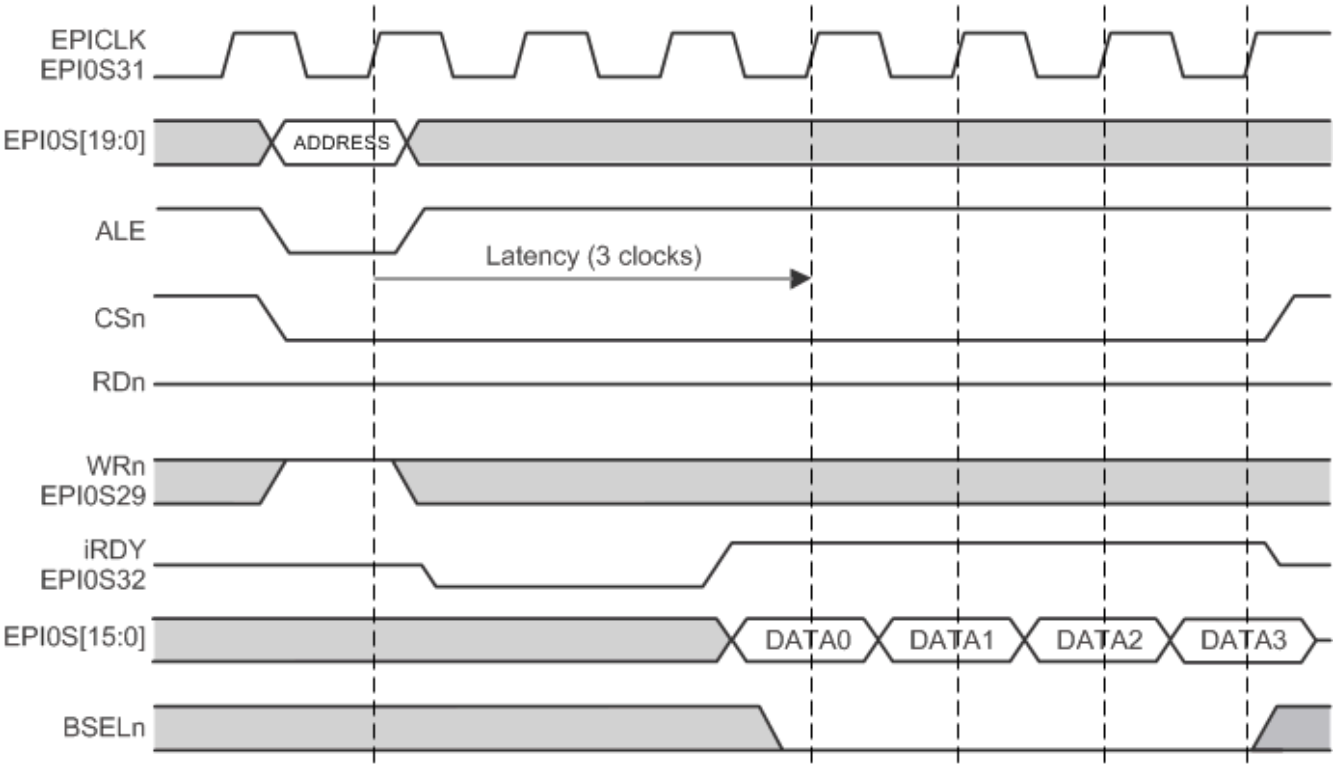


Figure 28-7. PSRAM Burst Read

ADVANCE INFORMATION

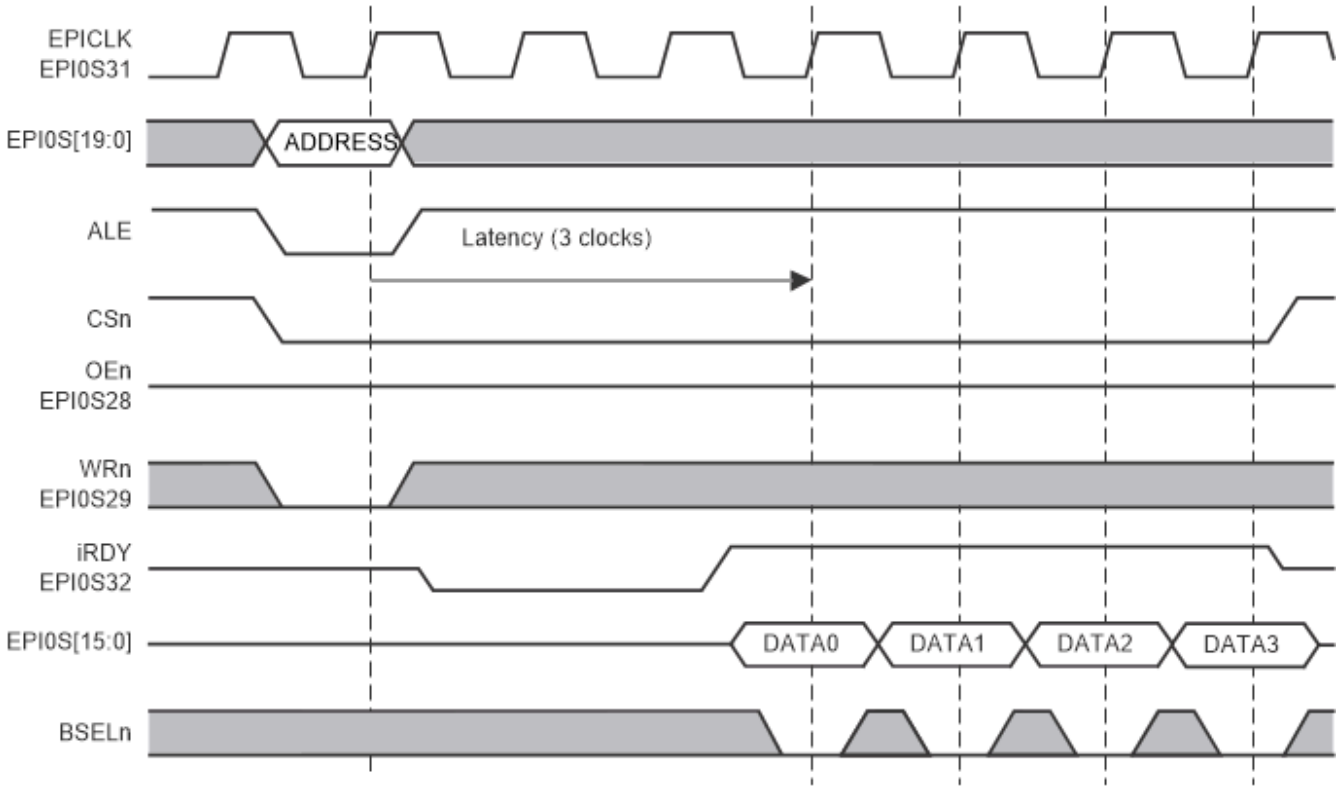


Figure 28-8. PSRAM Burst Write

If a read or write transfer attempts to begin during a refresh event, the transfer is held off by the assertion of the iRDY pin by the memory to the EPI module. Figure 28-9 and Figure 28-10 depict the delay in data transfer during a refresh collision.

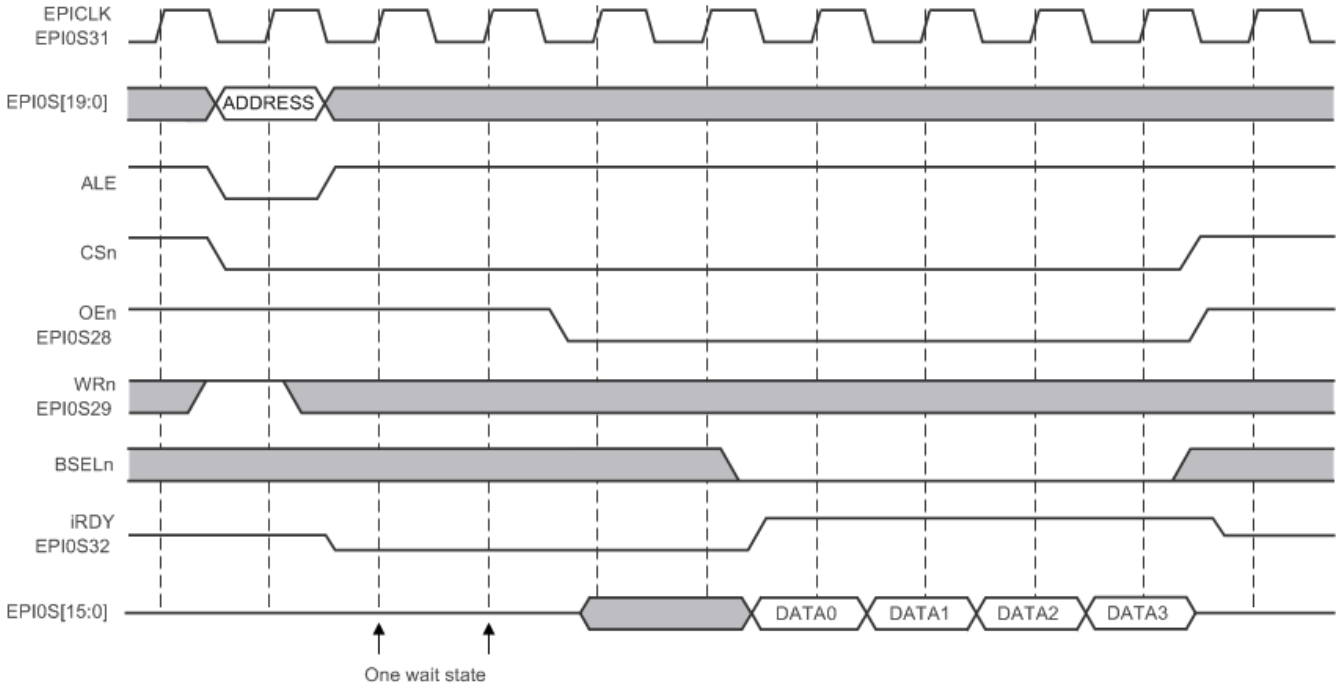


Figure 28-9. Read Delay During Refresh Event

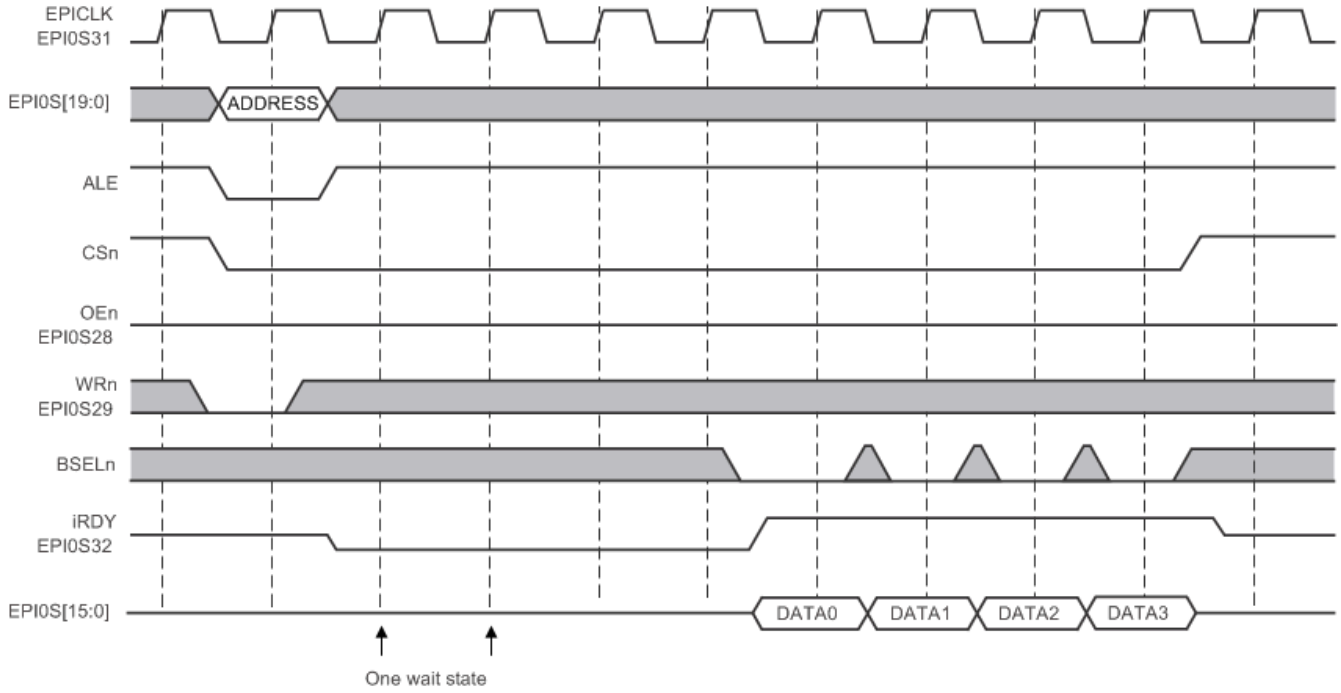


Figure 28-10. Write Delay During Refresh Event

28.1.4.3.3 Host Bus 16-Bit Muxed Interface

Figure 28-11 shows how to connect the EPI signals to a 16-bit SRAM and a 16-bit flash memory with muxed address and memory using byte selects and dual chip selects with ALE. This schematic is just an example of how to connect the signals; timing and loading have not been analyzed. In addition, not all bypass capacitors are shown.

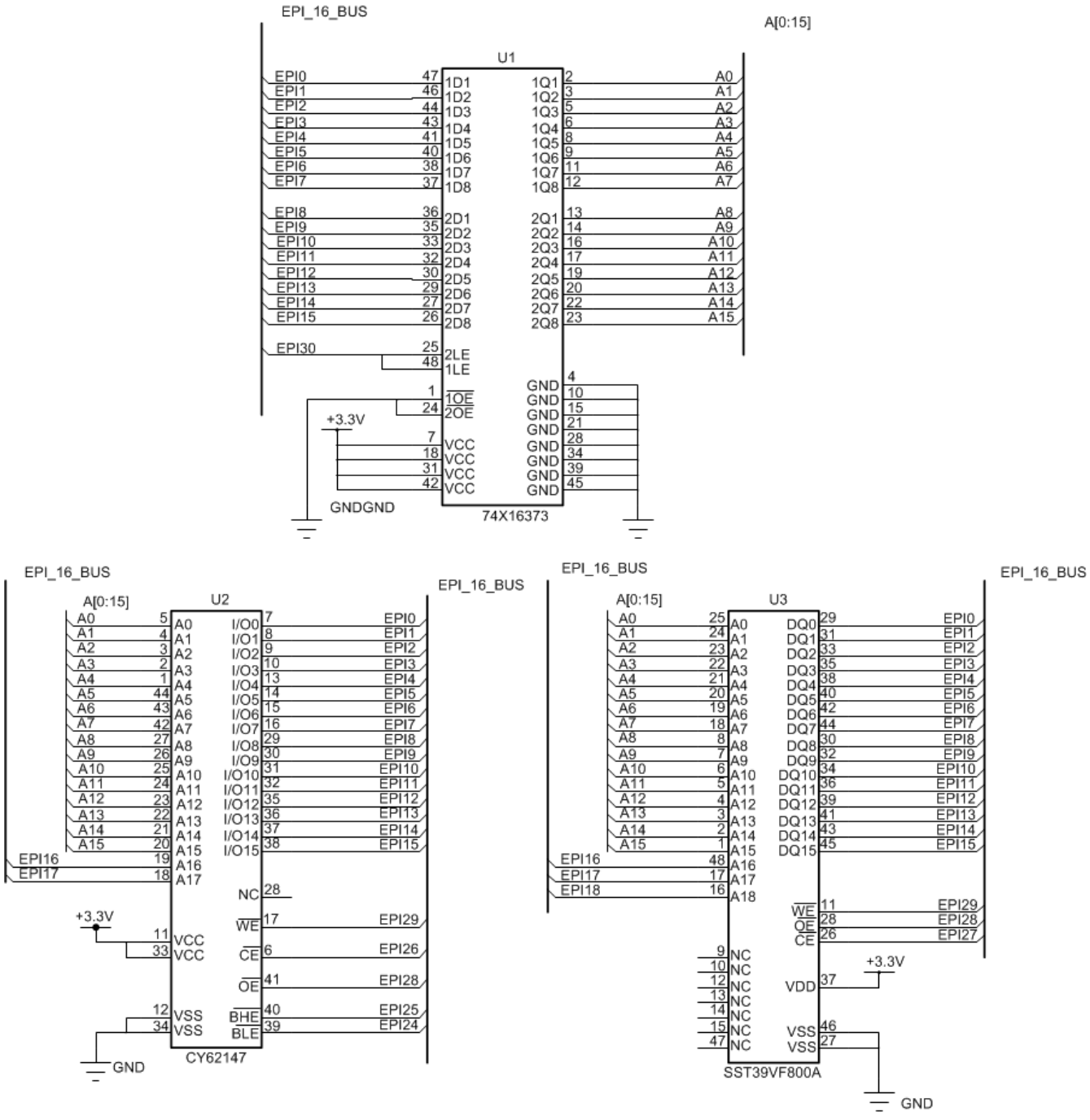


Figure 28-11. Example Schematic for Muxed Host-Bus 16 Mode

28.1.4.3.4 Speed of Transactions

The COUNT0 field in the EPIBAUD register must be configured to set the main transaction rate based on what the slave device can support (including wiring considerations). The main control transitions are normally 1/2 the baud rate (COUNT0 = 1) because the EPI block forces data versus control to change on alternating clocks. When using dual chip selects, each chip select can access the bus using differing baud rates by setting the CSBAUD bit in the EPIHBnCFG2 register. In this case, the COUNT0 field controls the CS0n transactions, and the COUNT1 field controls the CS1n transactions. When using quad chip select mode, the COUNT0 bit field of the EPIBAUD2 register controls the baud rate of CS2n and the COUNT1 bit field is programmed to control the baud rate of CS3n.

Additionally, the Host-Bus mode provides read and write wait states for the data portion to support different classes of device. These wait states stretch the data period (hold the rising edge of data strobe) and may be used in all four sub-modes. The wait states are set using the WRWS and RDWS bits in the EPI Host-Bus n Configuration (EPIHBnCFGn) register. The WRWS and RDWS bits are enhanced with more precision by WRWSM and RDWSM bits in the EPIHBnTIMEn registers. Note none of the wait state configuration bits can be set concurrently with the BURST bit in the same EPIHBnCFGn register. See Table 28-10 for programming information.

Table 28-10. Data Phase Wait State Programming

RDWS or WRWS Encoding in EPIHBnCFGn Register	RDWSM or WRWSM Encoding in EPIHBnTIMEn Registers	Data Phase Wait States
0x0	1	1 EPI clock cycle
0x0	0	2 EPI clock cycles
0x1	1	3 EPI clock cycles
0x1	0	4 EPI clock cycles
0x2	1	5 EPI clock cycles
0x2	0	6 EPI clock cycles
0x3	1	7 EPI clock cycles
0x3	0	8 EPI clock cycles

The CAPWIDTH bit in EPIHBnTIMEn registers controls the delay between Host-Bus transfers. When the CSBAUD bit is set and multiple chip selects have been configured in the EPIHBnCFG2 registers, delay takes an additional clock cycle to adjust the clock rate of different chip selects.

Word read and write transactions can be enhanced through the enabling of the BURST bit in the EPIHB16CFGn registers.

28.1.4.3.5 Sub-Modes of Host Bus 8 and 16

The EPI controller supports four variants of the host-bus model using 8 or 16 bits of data in all four cases. The four sub-modes are selected using the MODE bits in the EPIHBnCFG register, and are:

1. Address and data are muxed. This scheme is used by many 8051 devices, some Microchip PIC parts, and some ATmega parts. When used for standard SRAMs, a latch must be used between the microcontroller and the SRAM. This sub-mode is provided for compatibility with existing devices that support data transfers without a latch (that is, CPLDs). In general, the de-muxed sub-mode must normally be used. The ALE configuration must be used in this mode, as all Host-Bus accesses have an address phase followed by a data phase. The ALE indicates to an external latch to capture the address then hold until the data phase. The ALE configuration is controlled by configuring the CSCFG and CSCFGEXT field to be 0x0 in the EPIHBnCFG2 register. The ALE can be enhanced to access two or four external devices with four separate CSn signals. By configuring the CSCFG field to be 0x3 and the CSCFGEXT bit to be 0 in the EPIHBnCFG2 register, EPI0S30 functions as ALE, EPI0S27 functions as CS1n, and EPI0S26 functions as CS0n. When the CSCFG field is set to 0x0 and the CSCFGEXT bit is set to 1 in the EPIHBnCFG2 register, EPI0S30 functions as ALE, EPIOS33 functions as CS3n, EPIOS34 functions as CS2n, EPI0S27 functions as CS1n, and EPI0S26 functions as CS0n. The CSn is best used for Host-Bus unmuxed mode, in which EPI address

and data pins are separate. The CSn indicates when the address and data phases of a read or write access are occurring.

2. Address and data are separate with 8 or 16 bits of data and up to 20 bits of address (1MB). This scheme is used by more modern 8051 devices, as well as some PIC and ATmega parts. This mode is generally used with SRAMs in continuous read modes, many EEPROMs, and many NOR Flash memory devices. There is no hardware command write support for flash memory devices; this mode must only be used for Flash memory devices programmed at manufacturing time. If a Flash memory device must be written and does not support a direct programming model, the command mechanism must be performed in software. The CSn configuration must be used in this mode. The CSn signal indicates when the address and data phases of a read or write access is occurring. The CSn configuration is controlled by configuring the CSCFG field to be 0x1 and the CSCFGEXT bit to be 0 in the EPIHBnCFG2 register.
3. Continuous read mode where address and data are separate. This read sub-mode is used by some SRAMs and can read more quickly by only changing the address (and not using RDn/OEn strobing). In this sub-mode, reads are performed by keeping the read mode selected (output enable is asserted) and then changing the address pins. The data pins are changed by the SRAM after the address pins change. For example, to read data from address 0x100 and then 0x101, the EPI controller asserts the output-enable signal and then configures the address pins to 0x100; the EPI controller then captures what is on the data pins and increments A0 to 1 (so the address is now 0x101); the EPI controller then captures what is on the data pins. This mode consumes higher power because the SRAM must continuously drive the data pins. This mode is not practical in HB16 mode for normal SRAMs because there are generally not enough address bits available. Writes are not permitted in this mode.
4. FIFO mode uses 8 or 16 bits of data, removes ALE and address pins and optionally adds external XFIFO FULL/EMPTY flag inputs. This scheme is used by many devices, such as radios, communication devices (including USB2 devices), and some FPGA configurations (FIFO through block RAM). This sub-mode provides the data side of the normal Host-Bus interface, but is paced by the FIFO control signals. It is important to consider that the XFIFO FULL/EMPTY control signals can stall the interface and can have an impact on blocking read latency from the processor or DMA. The EPI FIFO can only be used in asynchronous mode.

For the three modes (1, 2, and 4) that the Host-Bus 16 mode supports, byte select signals can be optionally implemented by setting the BSEL bit in the EPIHB16CFG register.

Note

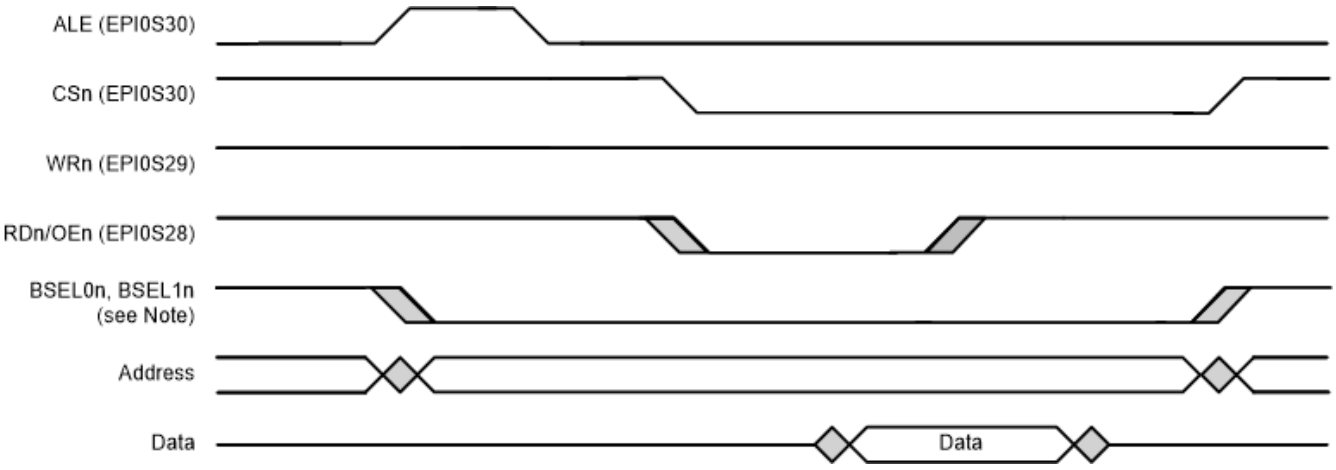
Byte accesses must not be attempted if the BSEL bit has not been enabled in Host-Bus 16 Mode.

For timing details for the Host-Bus mode, see the device-specific data sheet.

28.1.4.3.6 Bus Operation

Bus operation is the same in Host-Bus 8 and Host-Bus 16 modes and is asynchronous. Timing diagrams show both ALE and CSn operation. The optional HB16 byte select signals have the same timing as the address signals. If wait states are required in the bus access, the wait states can be inserted during the data phase of the access using the WRWS and RDWS bits in the EPIHBnCFG2 register. Each wait state adds 2 EPI clock cycles to the duration of the WRn or RDn strobe. During idle cycles, the address and muxed address data signals maintain the state of the last cycle.

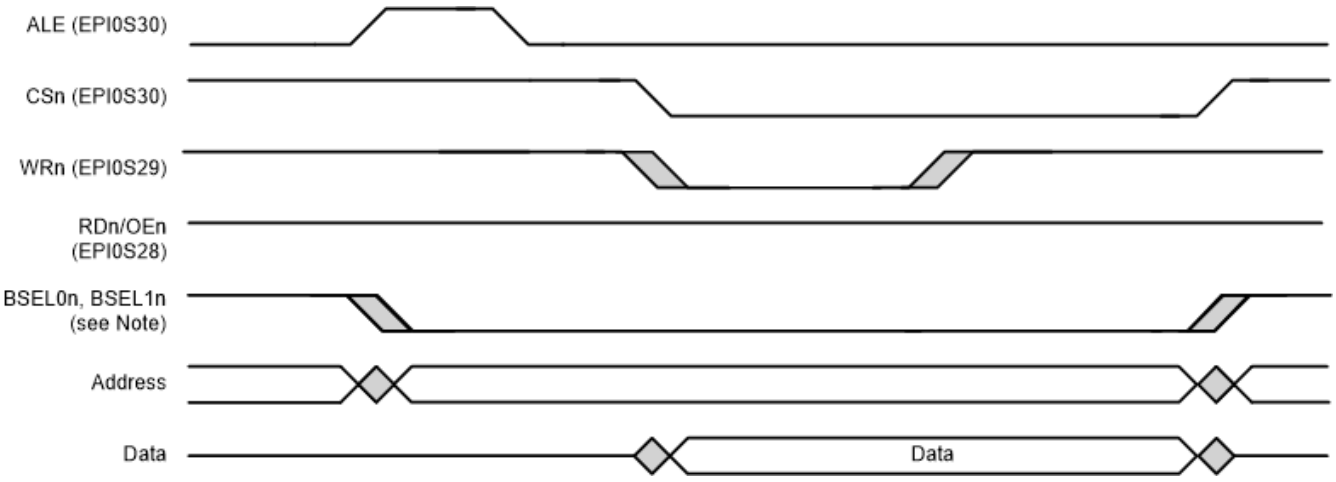
Figure 28-12 shows a basic Host-Bus read cycle. Figure 28-13 shows a basic Host-Bus write cycle. Both of these figures show address and data signals in the non-multiplexed mode (MODE field ix 0x1 in the EPIHBnCFG register).



Note

BSEL0n and BSEL1n are available in Host-Bus 16 mode only.

Figure 28-12. Host-Bus Read Cycle, MODE = 0x1, WRHIGH = 0, RDHIGH = 0



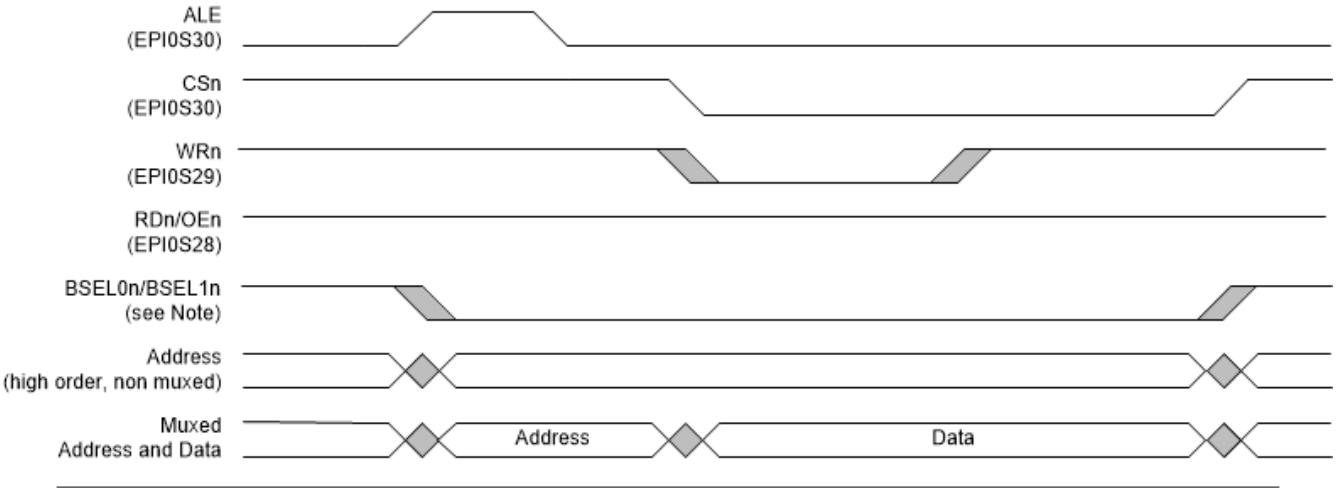
Note

BSEL0n and BSEL1n are available in Host-Bus 16 mode only.

Figure 28-13. Host-Bus Write Cycle, MODE = 0x1, WRHIGH = 0, RDHIGH = 0

ADVANCE INFORMATION

Figure 28-14 shows a write cycle with the address and data signals multiplexed (MODE field is 0x0 in the EPIHBnCFG register). A read cycle can look similar, with the RDn strobe being asserted along with CSn and data being latched on the rising edge of RDn.

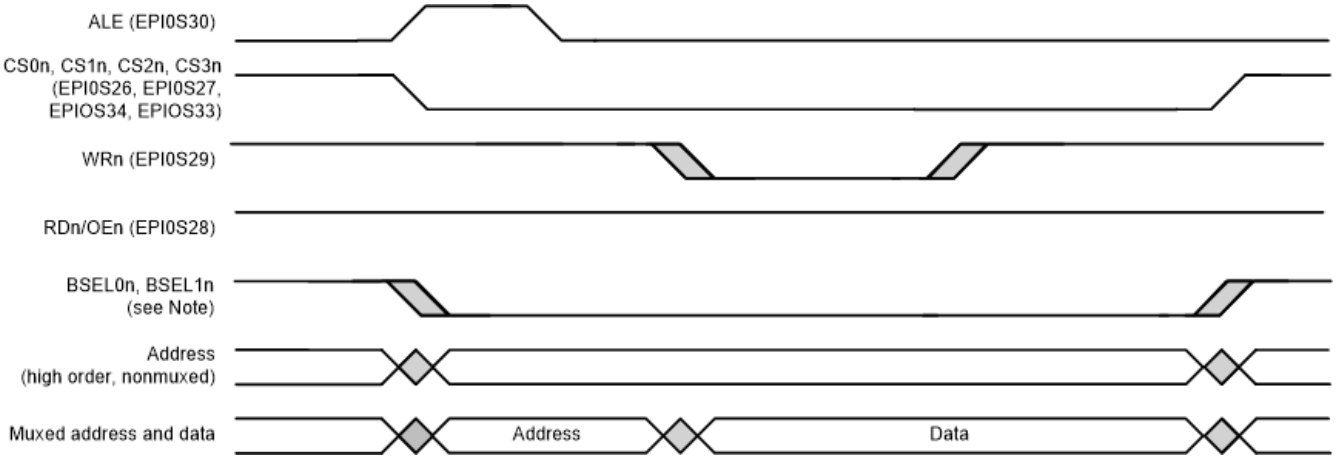


Note

BSEL0n and BSEL1n are available in Host-Bus 16 mode only.

Figure 28-14. Host-Bus Write Cycle with Multiplexed Address and Data, MODE = 0x0, WRHIGH = 0, RDHIGH = 0

When using ALE with dual CSn configuration (CSCFGEXT bit is 0 and the CSCFG field is 0x3 in the EPIHBnCFG2 register) or quad chip select (CSCFGEXT bit is 1 and CSCSFG is 0x2), the appropriate CSn signal is asserted at the same time as ALE, as shown in Figure 28-15.



Note

BSEL0n and BSEL1n are available in Host-Bus 16 mode only.

Figure 28-15. Host-Bus Write Cycle with Multiplexed Address and Data and ALE With Dual or Quad CSn

Figure 28-16 shows continuous read mode accesses. In this mode, reads are performed by keeping the read mode selected (output enable is asserted) and then changing the address pins. The data pins are changed by the SRAM after the address pins change.

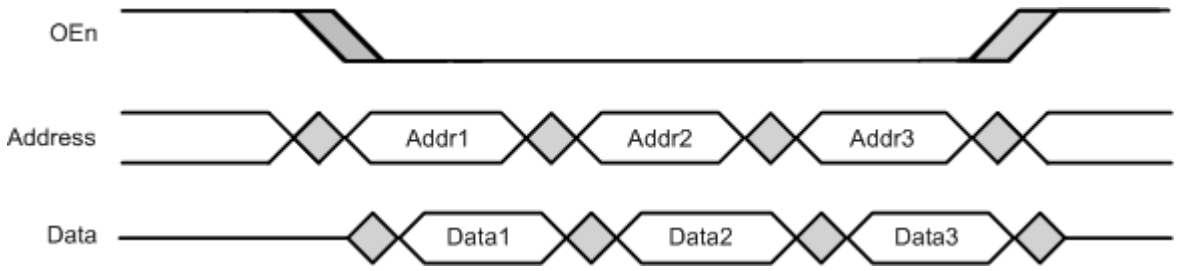


Figure 28-16. Continuous Read Mode Accesses

FIFO mode accesses are the same as normal read and write accesses, except that the ALE signal and address pins are not present. Two input signals can be used to indicate when the XFIFO is full or empty to gate transactions and avoid overruns and underruns. The FFULL and FEMPTY signals are synchronized and must be recognized as asserted by the microcontroller for 2 system clocks before the signals affect transaction status. The MAXWAIT field in the EPIHBnCFG register defines the maximum number of EPI clock cycles to wait while the FEMPTY or FFULL signal is holding off a transaction. Figure 28-17 shows how the FEMPTY signal responds to a write and read from the XFIFO. Figure 28-18 shows how the FEMPTY and FFULL signals respond to 2 writes and 1 read from an external FIFO that contains two entries.

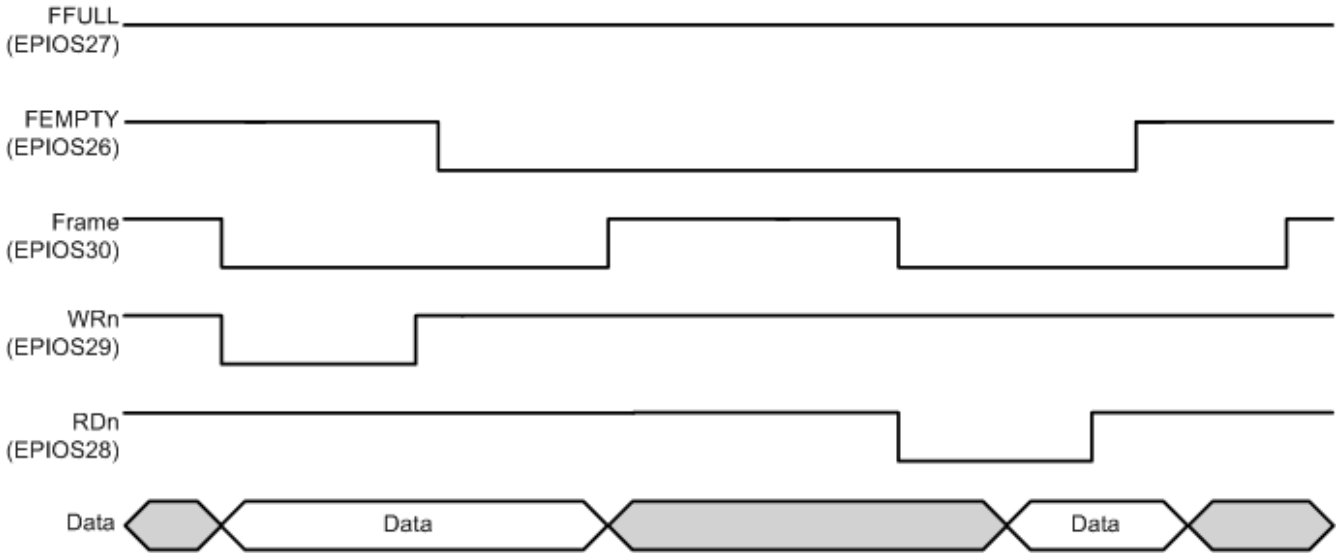


Figure 28-17. Write Followed by Read to External FIFO

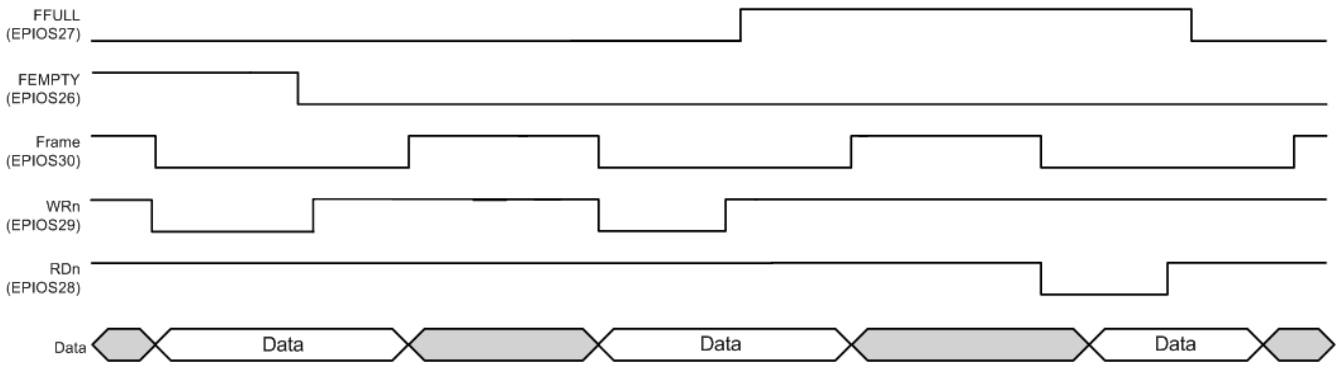


Figure 28-18. Two-Entry FIFO

ADVANCE INFORMATION

28.1.4.4 General-Purpose Mode

The General-Purpose Mode Configuration (EPIGPCFG) register is used to configure the control, data, and address pins, if used. Any unused EPI controller signals can be used as GPIOs or another alternate function. The general-purpose configuration can be used for custom interfaces with FPGAs, CPLDs, and digital data acquisition and actuator control.

General-Purpose mode is designed for three general types of use:

- Extremely high-speed clocked interfaces to FPGAs and CPLDs. Three sizes of data and optional address are supported. Framing and clock-enable functions permit more optimized interfaces.
- General parallel GPIO. From 1 to 32 pins can be written or read, with the speed precisely controlled by the EPIBAUD register baud rate (when used with the WFIFO and/or the NBRFIFO) or by the rate of accesses from software or DMA. Examples of this type of use include:
  - Reading 20 sensors at fixed time periods by configuring 20 pins to be inputs, configuring the COUNT0 field in the EPIBAUD register to some divider, and then using nonblocking reads.
  - Implementing a very wide ganged PWM/PCM with fixed frequency for driving actuators or LEDs.
- General custom interfaces of any speed.

The configuration allows for choice of an output clock (free-running or gated), a framing signal (with frame size), a ready input (to stretch transactions), an address (of varying sizes), and data (of varying sizes). Additionally, provisions are made for separating data and address phases.

The interface has the following optional features:

- Use of the EPI clock output is controlled by the CLKPIN bit in the EPIGPCFG register. Unclocked uses include general-purpose I/O and asynchronous interfaces (optionally using RD and WR strobes). Clocked interfaces allow for higher speeds and are much easier to connect to FPGAs and CPLDs (which usually include input clocks).
- EPI clock, if used, can be free running or gated depending on the CLKGATE bit in the EPIGPCFG register. A free-running EPI clock requires another method for determining when data is live, such as the frame pin or RD/WR strobes. A gated clock approach uses a setup-time model in which the EPI clock controls when transactions are starting and stopping. The gated clock is held high until a new transaction is started and goes high at the end of the cycle where RD/WR/FRAME and address (and data if write) are emitted.
- Use of the RD and WR outputs is controlled by the RW bit in the EPIGPCFG register. For interfaces where the direction is known (in advance, related to frame size, or other means), these strobes are not needed. For most other interfaces, RD and WR are used so the external peripheral knows what transaction is taking place, and if any transaction is taking place.
- Separation of address/request and data phases can be used on writes using the WR2CYC bit in the EPIGPCFG register. This configuration allows the external peripheral extra time to act. Address and data phases must be separated on reads. When configured to use an address as specified by the ASIZE field in the EPIGPCFG register, the address is emitted on the with the RD strobe (first cycle) and data is expected to be returned on the next cycle (when RD is not asserted). If no address is used, then RD is asserted on the first cycle and data is captured on the second cycle (when RD is not asserted), allowing more setup time for data.

  Note

  When WR2CYC = 0, write data is valid when the WR strobe is asserted (High). When WR2CYC = 1, write data is valid when the WR strobe is Low after being asserted (High).

  For writes, the output can be in one or two cycles. In the two-cycle case, the address (if any) is emitted on the first cycle with the WR strobe and the data is emitted on the second cycle (with WR not asserted). Although split address and write data phases are not normally needed for logic reasons, it can be useful to make read and write timings match. If 2-cycle reads or writes are used, the RW bit is automatically set.
- Address can be emitted (controlled by the ASIZE field in the EPIGPCFG register). The address can be up to 4 bits (16 possible values), up to 12 bits (4096 possible values), or up to 20 bits (1M possible values). Size of address limits size of data, for example, 4 bits of address support up to 24 bits data. 4-bit address uses

ADVANCE INFORMATION

EPI0S[27:24]; 12-bit address uses EPI0S[27:16]; 20-bit address uses EPI0S[27:8]. The address signals can be used by the external peripheral as an address, code (command), or for other unrelated uses (such as a chip enable). If the chosen address/data combination does not use all of the EPI signals, the unused pins can be used as GPIOs or for other functions. For example, when using a 4-bit address with an 8-bit data, the pins assigned to EPIS0[23:8] can be assigned to other functions.

- Data can be 8 bits, 16 bits, 24 bits, or 32 bits (controlled by the DSIZE field in the EPIGPCFG register). By default, the EPI controller uses data bits [7:0] when the DSIZE field in the EPIGPCFG register is 0x0; data bits [15:0] when the DSIZE field is 0x1; data bits [23:0] when the DSIZE field is 0x2; and data bits [31:0] when the DSIZE field is 0x3.32-bit data cannot be used with address or EPI clock or any other signal. 24-bit data can only be used with 4-bit address or no address.
- When using the EPI controller as a GPIO interface, writes are FIFOed (up to 4 can be held at any time), and up to 32 pins are changed using the EPIBAUD clock rate specified by COUNT0. As a result, output pin control can be very precisely controlled as a function of time. By contrast, when writing to normal GPIOs, writes can only occur 8-bits at a time and take up to 2 clock cycles to complete. In addition, the write can be further delayed by the bus due to DMA or draining of a previous write. With both GPIO and the EPI controller, reads can be performed directly, in which case the current pin states are read back. With the EPI controller, the nonblocking interface can also be used to perform reads based on a fixed time rule using the EPIBAUD clock rate.

Table 28-11 shows how the EPI0S[31:0] signals function while in General-Purpose mode. Notice that the address connections vary depending on the data-width restrictions of the external peripheral.

Table 28-11. EPI General-Purpose Signal Connections

EPI Signal	General-Purpose Signal (D8, A20)	General-Purpose Signal (D16, A12)	General-Purpose Signal (D24, A4)	General-Purpose Signal (D32)
EPI0S0	D0	D0	D0	D0
EPI0S1	D1	D1	D1	D1
EPI0S2	D2	D2	D2	D2
EPI0S3	D3	D3	D3	D3
EPI0S4	D4	D4	D4	D4
EPI0S5	D5	D5	D5	D5
EPI0S6	D6	D6	D6	D6
EPI0S7	D7	D7	D7	D7
EPI0S8	A0	D8	D8	D8
EPI0S9	A1	D9	D9	D9
EPI0S10	A2	D10	D10	D10
EPI0S11	A3	D11	D11	D11
EPI0S12	A4	D12	D12	D12
EPI0S13	A5	D13	D13	D13
EPI0S14	A6	D14	D14	D14
EPI0S15	A7	D15	D15	D15
EPI0S16	A8	A0 (1)	D16	D16
EPI0S17	A9	A1	D17	D17
EPI0S18	A10	A2	D18	D18
EPI0S19	A11	A3	D19	D19
EPI0S20	A12	A4	D20	D20
EPI0S21	A13	A5	D21	D21
EPI0S22	A14	A6	D22	D22
EPI0S23	A15	A7	D23	D23
EPI0S24	A16	A8	A0 (2)	D24
EPI0S25	A17	A9	A1	D25

Table 28-11. EPI General-Purpose Signal Connections (continued)

EPI Signal	General-Purpose Signal (D8, A20)	General-Purpose Signal (D16, A12)	General-Purpose Signal (D24, A4)	General-Purpose Signal (D32)
EPI0S26	A18	A10	A2	D26
EPI0S27	A19	A11	A3	D27
EPI0S28	WR	WR	WR	D28
EPI0S29	RD	RD	RD	D29
EPI0S30	Frame	Frame	Frame	D30
EPI0S31	Clock	Clock	Clock	D31

(1) In this mode, halfword accesses are used. AO is the LSB of the address and is equivalent to the system A1 address.
(2) In this mode, word accesses are used. AO is the LSB of the address and is equivalent to the system A2 address.

28.1.4.4.1 Bus Operation

A basic access is 1 EPI clock for write cycles and 2 EPI clock cycles for read cycles. An additional EPI clock can be inserted into a write cycle by setting the WR2CYC bit in the EPIGPCFG register.

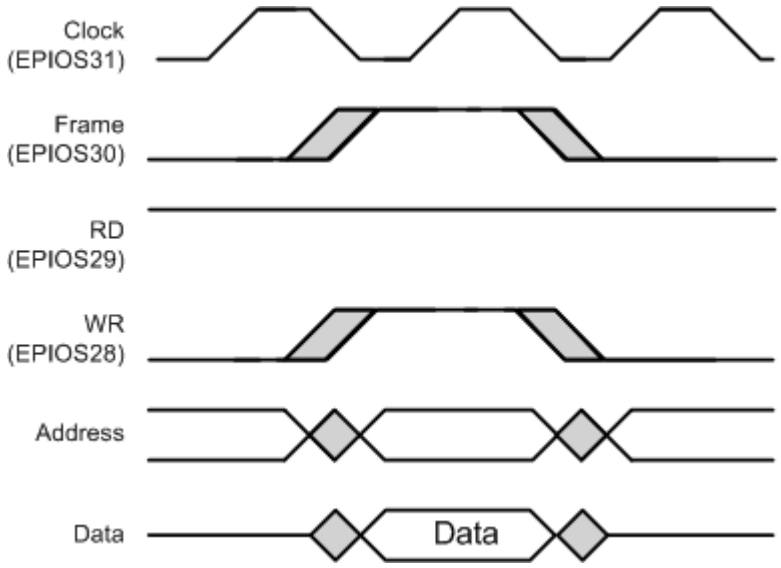


Figure 28-19. Single-Cycle Single Write Access, FRM50 = 0, FRMCNT = 0, WR2CYC = 0

ADVANCE INFORMATION

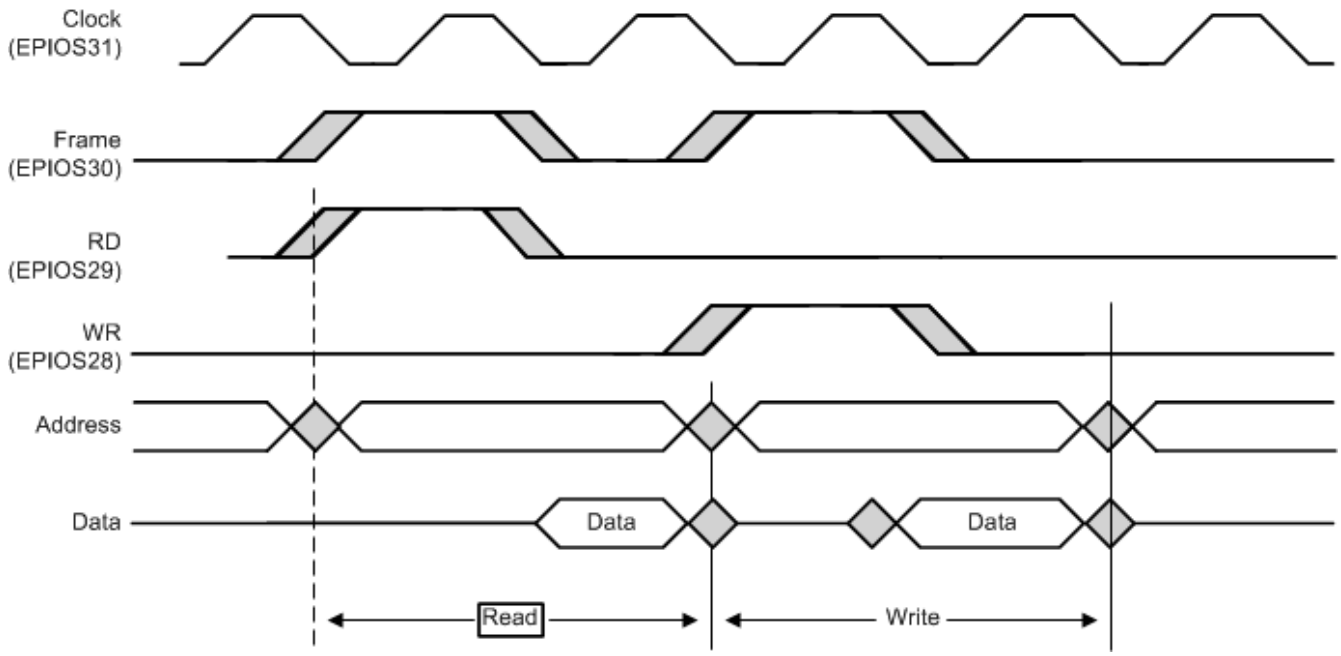


Figure 28-20. Two-Cycle Read, Write Accesses, FRM50 = 0, FRMCNT = 0, WR2CYC = 1

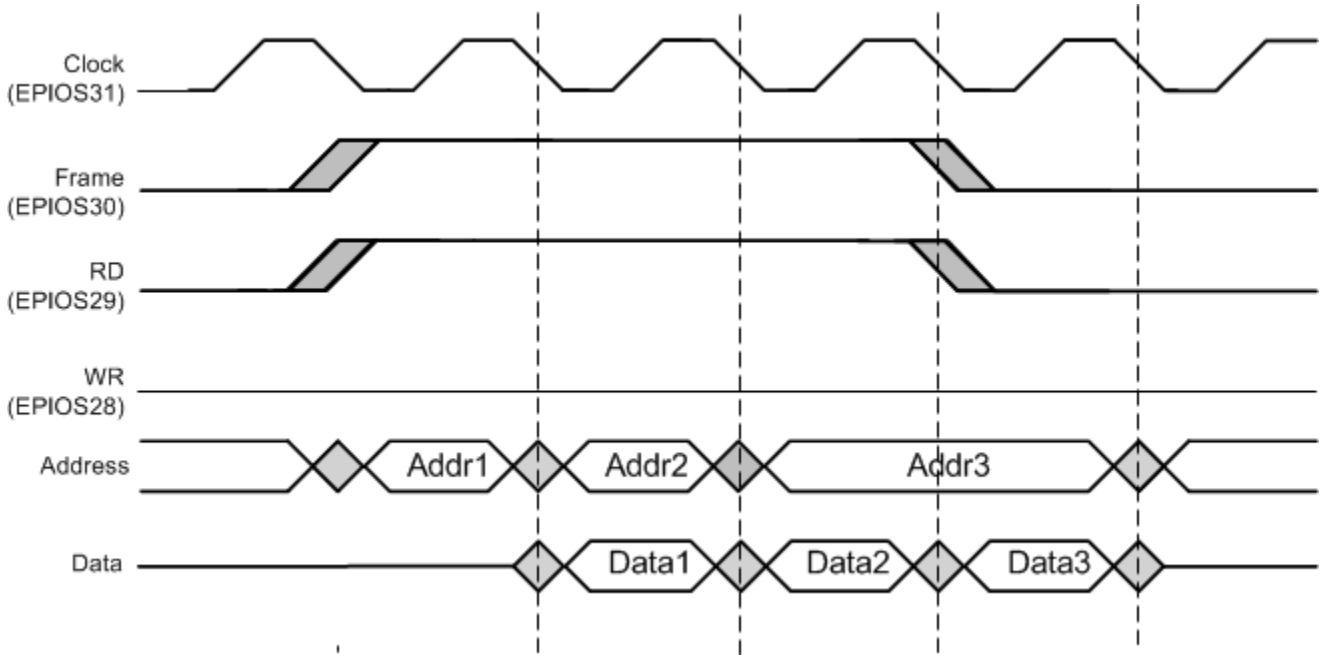


Figure 28-21. Read Accesses, FRM50 = 0, FRMCNT = 0

ADVANCE INFORMATION

28.1.4.4.1.1 FRAME Signal Operation

The operation of the FRAME signal is controlled by the FRMCNT and FRM50 bits. When FRM50 is clear, the FRAME signal is high whenever the WR or RD strobe is high. When FRMCNT is clear, the FRAME signal is simply the logical OR of the WR and RD strobes so the FRAME signal is high during every read or write access, see Figure 28-22.

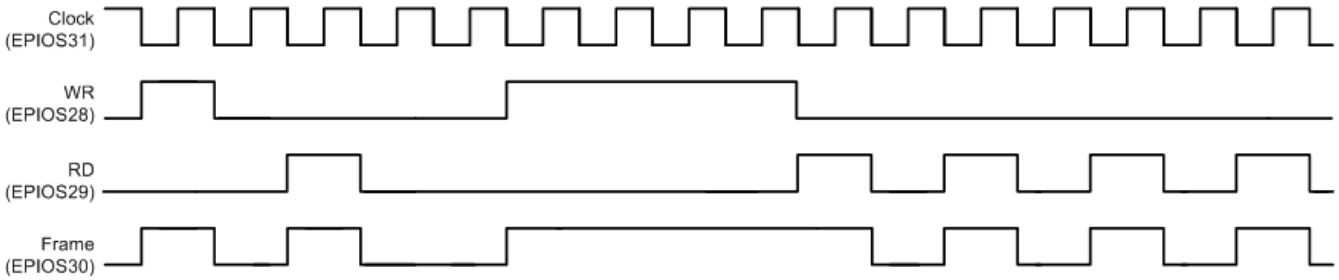


Figure 28-22. FRAME Signal Operation, FRM50 = 0 and FRMCNT = 0

If the FRMCNT field is 0x1, then the FRAME signal pulses high during every other read or write access, see Figure 28-23.

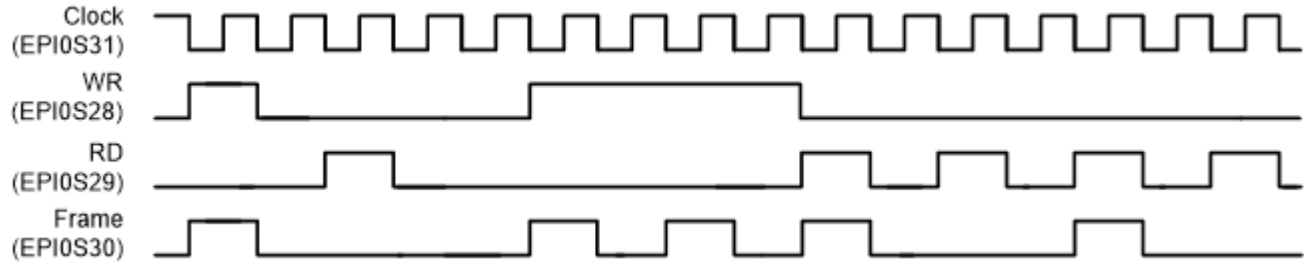


Figure 28-23. FRAME Signal Operation, FRM50 = 0 and FRMCNT = 1

If the FRMCNT field is 0x2 and FRM50 is clear, then the FRAME signal pulses high during every third access, and so on for every value of FRMCNT, see Figure 28-24.

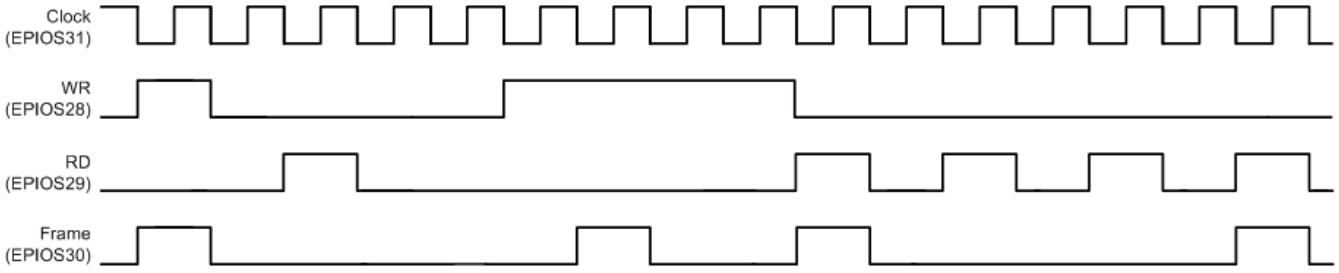


Figure 28-24. FRAME Signal Operation, FRM50 = 0 and FRMCNT = 2

ADVANCE INFORMATION

When FRM50 is set, the FRAME signal transitions on the rising edge of either the WR or RD strobes. When FRMCNT = 0, the FRAME signal transitions on the rising edge of WR or RD for every access, see Figure 28-25.

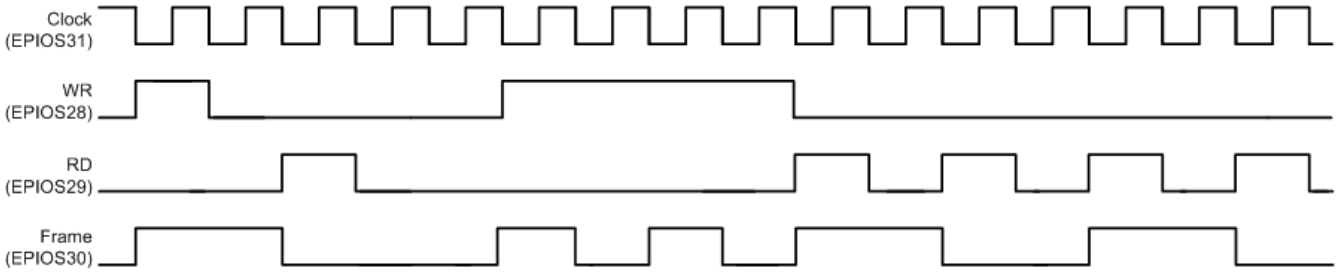


Figure 28-25. FRAME Signal Operation, FRM50 = 1 and FRMCNT = 0

When FRMCNT = 1, the FRAME signal transitions on the rising edge of the WR or RD strobes for every other access, see Figure 28-26.

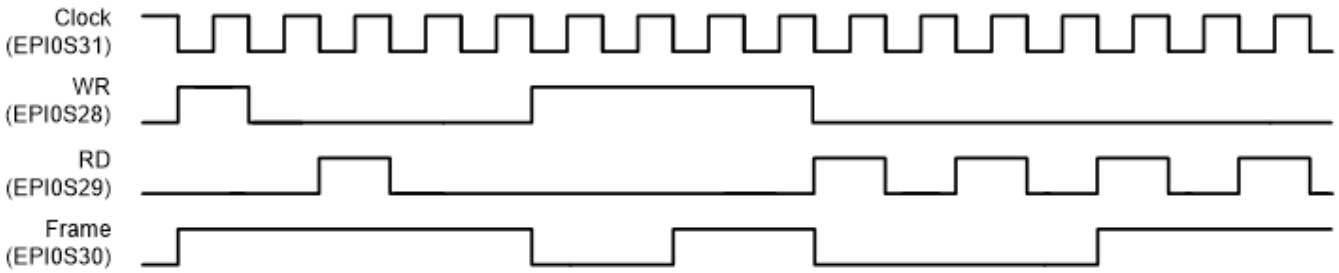


Figure 28-26. FRAME Signal Operation, FRM50 = 1 and FRMCNT = 1

When FRMCNT = 2, the FRAME signal transitions the rising edge of the WR or RD strobes for every third access, and so on for every value of FRMCNT, see Figure 28-27.

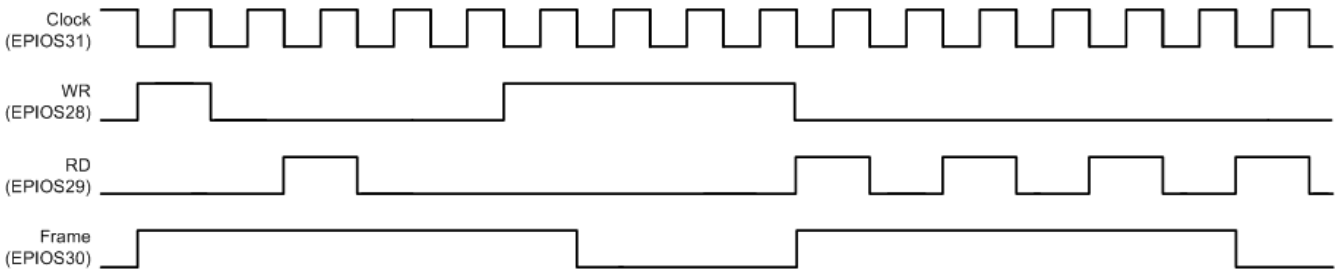


Figure 28-27. FRAME Signal Operation, FRM50 = 1 and FRMCNT = 2

ADVANCE INFORMATION

28.1.4.4.1.2 EPI Clock Operation

If the CLKGATE bit in the EPIGPCFG register is clear, the EPI clock always toggles when General-purpose mode is enabled. If CLKGATE is set, the clock is output only when a transaction is occurring, otherwise the clock is held high. If the WR2CYC bit is clear, the EPI clock begins toggling 1 cycle before the WR strobe goes High. If the WR2CYC bit is set, the EPI clock begins toggling when the WR strobe goes High. The clock stops toggling after the first rising edge after the WR strobe is deasserted. The RD strobe operates in the same manner as the WR strobe when the WR2CYC bit is set. See Figure 28-28 and Figure 28-29.

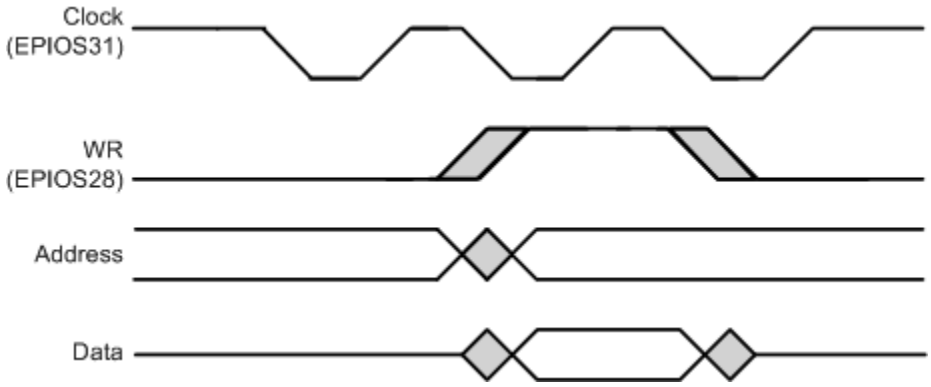


Figure 28-28. EPI Clock Operation, CLKGATE = 1, WR2CYC = 0

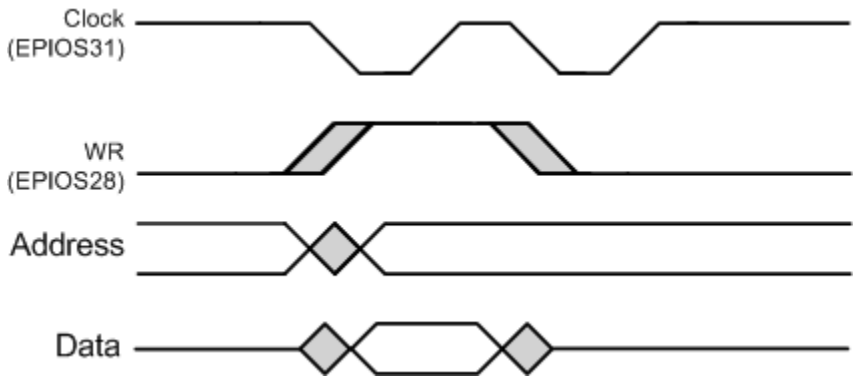


Figure 28-29. EPI Clock Operation, CLKGATE = 1, WR2CYC = 1

28.2 EPI Registers

This Section describes the EPI Registers.

28.2.1 EPI Base Address Table

Table 28-12. EPI Base Address Table

Bit Field Name		DriverLib Name	Base Address
Instance	Structure		
Epi0Regs, Epi0SdramRegs, Epi0Hb8Regs, Epi0Hb16Regs	EPI_REGS_GPCFG,EPI_REGS_SDRAMCFG,EPI_REGS_HB8CFG	EPI0, EPI0SDRAM, EPI0HB8, EPI0HB16	0x4001_A000

ADVANCE INFORMATION

28.2.2 EPI_REGS_GPCFG Registers

Table 28-13 lists the memory-mapped registers for the EPI_REGS_GPCFG registers. All register offset addresses not listed in Table 28-13 should be considered as reserved locations and the register contents should not be modified.

Table 28-13. EPI_REGS_GPCFG Registers

Offset	Acronym	Register Name	Write Protection	Section
0h	EPICFG	EPI Configuration		Go
4h	EPIBAUD	EPI Main Baud Rate		Go
8h	EPIBAUD2	EPI Main Baud Rate		Go
10h	EPIGPCFG	EPI General-Purpose Configuration		Go
1Ch	EPIADDRMAP	EPI Address Map		Go
20h	EPIRSIZE0	EPI Read Size 0		Go
24h	EPIRADDR0	EPI Read Address 0		Go
28h	EPIRPSTD0	EPI Non-Blocking Read Data 0		Go
30h	EPIRSIZE1	EPI Read Size 1		Go
34h	EPIRADDR1	EPI Read Address 1		Go
38h	EPIRPSTD1	EPI Non-Blocking Read Data 1		Go
60h	EPISTAT	EPI Status		Go
6Ch	EPIRFIFOCNT	EPI Read FIFO Count		Go
70h	EPIREADFIFO0	EPI Read FIFO 0		Go
74h	EPIREADFIFO1	EPI Read FIFO 1		Go
78h	EPIREADFIFO2	EPI Read FIFO 2		Go
7Ch	EPIREADFIFO3	EPI Read FIFO 3		Go
80h	EPIREADFIFO4	EPI Read FIFO 4		Go
84h	EPIREADFIFO5	EPI Read FIFO 5		Go
88h	EPIREADFIFO6	EPI Read FIFO 6		Go
8Ch	EPIREADFIFO7	EPI Read FIFO 7		Go
200h	EPIFIFOLVL	EPI FIFO Level Selects		Go
204h	EPIWFIFOCNT	EPI Write FIFO Count		Go
208h	EPIDMATXCNT	EPI DMA Transmit Count		Go
210h	EPIIM	EPI Interrupt Mask		Go
214h	EPIRIS	EPI Raw Interrupt Status		Go
218h	EPIMIS	EPI Masked Interrupt Status		Go
21Ch	EPIEISC	EPI Error and Interrupt Status and Clear		Go

Complex bit access types are encoded to fit into small table cells. Table 28-14 shows the codes that are used for access types in this section.

Table 28-14. EPI_REGS_GPCFG Access Type Codes

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

ADVANCE INFORMATION

Table 28-14. EPI_REGS_GPCFG Access Type Codes (continued)

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

1 EPICFG Register (Offset = 0h) [Reset = 00000000h]

EPICFG is shown in Figure 28-30 and described in Table 28-15.

Return to the Summary Table.

EPI Configuration

Figure 28-30. EPICFG Register

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

Table 28-16. EPICFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	RESERVED	R	0h	Reserved Reset type: SYSRSn
8	INTDIV	R/W	0h	Integer Clock Divider Enable Reset type: SYSRSn 0h (R/W) = EPIBAUD register values create formula clock divide. 1h (R/W) = EPIBAUD register values create integer clock divide.
7-5	RESERVED	R	0h	Reserved Reset type: SYSRSn
4	BLKEN	R/W	0h	Block Enable Reset type: SYSRSn 0h (R/W) = The EPI controller is disabled. 1h (R/W) = The EPI controller is enabled.
3-0	MODE	R/W	0h	Mode Select Reset type: SYSRSn 0h (R/W) = General Purpose. General-Purpose mode. Control, address, and data pins are configured using the EPIGPCFG and EPIGPCFG2 registers. 1h (R/W) = SDRAM. Supports SDR SDRAM. Control, address, and data pins are configured using the EPISDRAMCFG register. 2h (R/W) = 8-Bit Host-Bus (HB8). Host-bus 8-bit interface (also known as the MCU interface). Control, address, and data pins are configured using the EPIHB8CFG and EPIHB8CFG2 registers. 3h (R/W) = 16-Bit Host-Bus (HB16). Host-bus 16-bit interface (standard SRAM). Control, address, and data pins are configured using the EPIHB16CFG and EPIHB16CFG2 registers.

2 EPIBAUD Register (Offset = 4h) [Reset = 00000000h]

EPIBAUD is shown in Figure 28-31 and described in Table 28-16.

Return to the Summary Table.

EPI Main Baud Rate

Figure 28-31. EPIBAUD Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
COUNT1																COUNT0															
R/W-0h																R/W-0h															

Table 28-18. EPIBAUD Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	COUNT1	R/W	0h	This bit field is only valid with multiple chip selects which are enabled when the CSCFG field is 0x2 or 0x3 or the CSCFGEXT field is set to 1, with CSCFG field as 0x1 or 0x2 and the CSBAUD bit is set in the EPIHBnCFG2 register. This bit field contains a counter used to divide the system clock by the count. A count of 0 means the system clock is used as is. Reset type: SYSRSn
15-0	COUNT0	R/W	0h	This bit field contains a counter used to divide the system clock by the count. A count of 0 means the system clock is used as is. Reset type: SYSRSn

3 EPIBAUD2 Register (Offset = 8h) [Reset = 00000000h]

EPIBAUD2 is shown in Figure 28-32 and described in Table 28-17.

Return to the Summary Table.

EPI Main Baud Rate

Figure 28-32. EPIBAUD2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
COUNT1																COUNT0															
R/W-0h																R/W-0h															

Table 28-20. EPIBAUD2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	COUNT1	R/W	0h	This bit field contains a counter used to divide the system clock by the count. A count of 0 means the system clock is unchanged. This bit field is only valid when quad chip selects are enabled by setting the CSCFGEXT bit to 1 and the CSCFG field to 0x1 or 0x2. In addition, the CSBAUD bit must be set in the EPIHBnCFG2 register. Reset type: SYSRSn
15-0	COUNT0	R/W	0h	This bit field contains a counter used to divide the system clock by the count. A count of 0 means the system clock is unchanged. This bit field is only valid when quad chip selects are enabled by setting the CSCFGEXT to 1 and the CSCFG field to 0x1 or 0x2. In addition, the CSBAUD bit must be set in the EPIHBnCFG2 register. Reset type: SYSRSn

4 EPIGPCFG Register (Offset = 10h) [Reset = 0000FF00h]

EPIGPCFG is shown in Figure 28-33 and described in Table 28-18.

Return to the Summary Table.

EPI General-Purpose Configuration

Figure 28-33. EPIGPCFG Register

31	30	29	28	27	26	25	24
CLKPIN	CLKGATE	RESERVED	RDYEN	RESERVED	FRM50	FRMCNT	
R/W-0h	R/W-0h	R-0h	R/W-0h	R-0h	R/W-0h	R/W-0h	
23	22	21	20	19	18	17	16
FRMCNT		RESERVED		WR2CYC	RD2CYC	RESERVED	
R/W-0h		R-0h		R/W-0h	R/W-0h	R-0h	
15	14	13	12	11	10	9	8
RESERVED							
R/W-FFh							
7	6	5	4	3	2	1	0
RESERVED		ASIZE		RESERVED		DSIZE	
R-0h		R/W-0h		R-0h		R/W-0h	

Table 28-22. EPIGPCFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31	CLKPIN	R/W	0h	The EPI clock is generated from the COUNT0 field in the EPIBAUD register (as is the system clock which is divided down from it). Reset type: SYSRSn 0h (R/W) = No clock output. 1h (R/W) = EPI0S31 functions as the EPI clock output.
30	CLKGATE	R/W	0h	CLKGATE is ignored if CLKPIN is 0 or if the COUNT0 field in the EPIBAUD register is cleared. Reset type: SYSRSn 0h (R/W) = The EPI clock is free running. 1h (R/W) = The EPI clock is output only when there is data to write or read (current transaction) otherwise the EPI clock is held low.
29	RESERVED	R	0h	Reserved Reset type: SYSRSn
28	RDYEN	R/W	0h	The ready enable signal may only be used with a free-running EPI clock(CLKGATE=0).The external iRDY signal is sampled on the falling edge of the EPI clock. Setup and hold times must be met to ensure registration on the next falling EPI clock edge. This bit is ignored if CLKPIN is 0 or CLKGATE is 1. Reset type: SYSRSn 0h (R/W) = The external peripheral does not drive an iRDY signal and is assumed to be ready always. 1h (R/W) = The external peripheral drives an iRDY signal into pin EPI0S27.
27	RESERVED	R	0h	Reserved Reset type: SYSRSn

Table 28-22. EPIGPCFG Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
26	FRM50	R/W	0h	50/50 Frame Reset type: SYSRSn 0h (R/W) = The FRAME signal is output as a single pulse, and then held low for the count. 1h (R/W) = The FRAME signal is output as 50/50 duty cycle using count (see FRMCNT).
25-22	FRMCNT	R/W	0h	This field specifies the size of the frame in EPI clocks. The frame counter is used to determine the frame size. The count is FRMCNT +1. So, a FRMCNT of 0 forms a pure transaction valid signal (held high during transactions, low otherwise). A FRMCNT of 0 with FRM50 set inverts the FRAME signal on each transaction. A FRMCNT of 1 means the FRAME signal is inverted every other transaction a value of 15 means every sixteenth transaction. If FRM50 is set, the frame is held high for FRMCNT +1 transactions, then held low for that many transactions, and so on. If FRM50 is clear, the frame is pulsed high for one EPI clock and then low for FRMCNT EPI clocks. Reset type: SYSRSn
21-20	RESERVED	R	0h	Reserved Reset type: SYSRSn
19	WR2CYC	R/W	0h	When this bit is set, then the RW bit is forced to be set. Reset type: SYSRSn 0h (R/W) = Data is output on the same EPI clock cycle as the address. 1h (R/W) = Writes are two EPI clock cycles long, with address on one EPI clock cycle (with the WR strobe asserted) and data written on the following EPI clock cycle (with WR strobe de-asserted). The next address (if any) is in the cycle following.
18	RD2CYC	R/W	0h	When this bit is set, then the RW bit is forced to be set. This bit must be set at all times in General-Purpose mode to ensure proper operation. Reset type: SYSRSn 0h (R/W) = Data is captured on the EPI clock cycle with READ strobe asserted. 1h (R/W) = Reads are two EPI clock cycles, with address on one EPI clock cycle (with the RD strobe asserted) and data captured on the following EPI clock cycle (with the RD strobe de-asserted). The next address (if any) is in the cycle following.
17-16	RESERVED	R	0h	Reserved Reset type: SYSRSn
15-8	RESERVED	R/W	FFh	Reserved
7-6	RESERVED	R	0h	Reserved Reset type: SYSRSn

Table 28-22. EPIGPCFG Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
5-4	ASIZE	R/W	0h	This field defines the size of the address bus. The address can be up to 4-bits wide with a 24-bit data bus, up to 12-bits wide with a 16-bit data bus, and up to 20-bits wide with an 8- bit data bus. If the full address bus is not used, use the least significant address bits. Any unused address bits can be used as GPIOs by clearing the AFSEL bit for the corresponding GPIOs. Reset type: SYSRSn 0h (R/W) = No address 1h (R/W) = Up to 4 bits wide. 2h (R/W) = Up to 12 bits wide. This size cannot be used with 24-bit data. 3h (R/W) = Up to 20 bits wide. This size cannot be used with data sizes other than 8.
3-2	RESERVED	R	0h	Reserved Reset type: SYSRSn
1-0	DSIZE	R/W	0h	This field defines the size of the data bus (starting at EPI0S0). Subsets of these numbers can be created by clearing the AFSEL bit for the corresponding GPIOs. Size 32 may not be used with clock, frame, address, or other control. Reset type: SYSRSn 0h (R/W) = 8 bits wide (EPI0S0 to EPI0S7) 1h (R/W) = 16 bits wide (EPI0S0 to EPI0S15) 2h (R/W) = 24 bits wide (EPI0S0 to EPI0S23) 3h (R/W) = 32 bits wide (EPI0S0 to EPI0S31)This size may not be used with an EPI clock. This value is normally used for acquisition input and actuator control as well as other general-purpose uses that require 32 bits per direction.

5 EPIADDRMAP Register (Offset = 1Ch) [Reset = 00000000h]

EPIADDRMAP is shown in Figure 28-34 and described in Table 28-19.

Return to the Summary Table.

EPI Address Map

Figure 28-34. EPIADDRMAP Register

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

Table 28-24. EPIADDRMAP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-12	RESERVED	R	0h	Reserved Reset type: SYSRSn
11-10	ECSZ	R/W	0h	This field selects the size of the external code. If the size of the external code is larger, a bus fault occurs. If the size of the external peripheral is smaller, it wraps (upper address bits unused). When not using byte selects in Host-Bus 16, data is accessed on 2-byte boundaries. As a result, the available address space is double the amount shown below. Reset type: SYSRSn 0h (R/W) = 256 bytes lower address range: 0x00 to 0xFF 1h (R/W) = 64 KB lower address range: 0x0000 to 0xFFFF 2h (R/W) = 16 MB lower address range: 0x000000 to 0xFFFFFF 3h (R/W) = 256MB lower address range: 0x0000000 to 0x0FFFFFFF
9-8	ECADR	R/W	0h	This field selects address mapping for the external code area. Reset type: SYSRSn 0h (R/W) = Not mapped 1h (R/W) = At 0x10000000 2h (R/W) = reserved 3h (R/W) = reserved
7-6	EPSZ	R/W	0h	This field selects the size of the external peripheral. If the size of the external peripheral is larger, a bus fault occurs. If the size of the external peripheral is smaller, it wraps (upper address bits unused). When not using byte selects in Host-Bus 16, data is accessed on 2-byte boundaries. As a result, the available address space is double the amount shown below. Reset type: SYSRSn 0h (R/W) = 256 bytes lower address range: 0x00 to 0xFF 1h (R/W) = 64 KB lower address range: 0x0000 to 0xFFFF 2h (R/W) = 16 MB lower address range: 0x000000 to 0xFFFFFF 3h (R/W) = 256 MB lower address range: 0x0000000 to 0xFFFFFFF

Table 28-24. EPIADDRMAP Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
5-4	EPADR	R/W	0h	This field selects address mapping for the external peripheral area. Reset type: SYSRSn 0h (R/W) = Not mapped 1h (R/W) = At 0xA0000000 2h (R/W) = At 0xC0000000 3h (R/W) = Only to be used with Host Bus quad chip select. In quad chip select mode, CS2n maps to 0xA0000000 and CS3n maps to 0xC0000000.
3-2	ERSZ	R/W	0h	This field selects the size of mapped RAM. If the size of the external memory is larger, a bus fault occurs. If the size of the external memory is smaller, it wraps (upper address bits unused): Reset type: SYSRSn 0h (R/W) = 256 bytes lower address range: 0x00 to 0xFF 1h (R/W) = 64 KB lower address range: 0x0000 to 0xFFFF 2h (R/W) = 16 MB lower address range: 0x000000 to 0xFFFFFF 3h (R/W) = 256 MB lower address range: 0x0000000 to 0xFFFFFFF
1-0	ERADR	R/W	0h	Selects address mapping for external RAM area: Reset type: SYSRSn 0h (R/W) = Not mapped 1h (R/W) = At 0x60000000 2h (R/W) = At 0x80000000 3h (R/W) = Only to be used with Host Bus quad chip select. In quad chip select mode, CS0n maps to 0x60000000 and CS1n maps to 0x80000000.

6 EPIRSIZE0 Register (Offset = 20h) [Reset = 00000003h]

EPIRSIZE0 is shown in Figure 28-35 and described in Table 28-20.

Return to the Summary Table.

EPI Read Size 0

Figure 28-35. EPIRSIZE0 Register

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

Table 28-26. EPIRSIZE0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R	0h	Reserved Reset type: SYSRSn
1-0	SIZE	R/W	3h	Current Size Reset type: SYSRSn 0h (R/W) = reserved 1h (R/W) = Byte (8 bits) 2h (R/W) = Half-word (16 bits) 3h (R/W) = Word (32 bits)

7 EPIRADDR0 Register (Offset = 24h) [Reset = 00000000h]

EPIRADDR0 is shown in Figure 28-36 and described in Table 28-21.

Return to the Summary Table.

EPI Read Address 0

Figure 28-36. EPIRADDR0 Register

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

Table 28-28. EPIRADDR0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ADDR	R/W	0h	Next address to read. Reset type: SYSRSn

ADVANCE INFORMATION

8 EPIRPSTD0 Register (Offset = 28h) [Reset = 00000000h]

EPIRPSTD0 is shown in Figure 28-37 and described in Table 28-22.

Return to the Summary Table.

EPI Non-Blocking Read Data 0

Figure 28-37. EPIRPSTD0 Register

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

Table 28-30. EPIRPSTD0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-13	RESERVED	R	0h	Reserved Reset type: SYSRSn
12-0	POSTCNT	R/W	0h	A write of a non-zero value starts a read operation for that count. Note that it is the software's responsibility to handle address wrap-around. Reading this register provides the current count. A write of 0 cancels a non-blocking read (whether active now or pending). Prior to writing a non-zero value, this register must first be cleared. Reset type: SYSRSn

9 EPIRSIZE1 Register (Offset = 30h) [Reset = 00000003h]

EPIRSIZE1 is shown in Figure 28-38 and described in Table 28-23.

Return to the Summary Table.

EPI Read Size 1

Figure 28-38. EPIRSIZE1 Register

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

Table 28-32. EPIRSIZE1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R	0h	Reserved Reset type: SYSRSn
1-0	SIZE	R/W	3h	Current Size Reset type: SYSRSn 0h (R/W) = reserved 1h (R/W) = Byte (8 bits) 2h (R/W) = Half-word (16 bits) 3h (R/W) = Word (32 bits)

ADVANCE INFORMATION

10 EPIRADDR1 Register (Offset = 34h) [Reset = 00000000h]

EPIRADDR1 is shown in Figure 28-39 and described in Table 28-24.

Return to the Summary Table.

EPI Read Address 1

Figure 28-39. EPIRADDR1 Register

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

Table 28-34. EPIRADDR1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ADDR	R/W	0h	Next address to read. Reset type: SYSRSn

ADVANCE INFORMATION

11 EPIRPSTD1 Register (Offset = 38h) [Reset = 00000000h]

EPIRPSTD1 is shown in Figure 28-40 and described in Table 28-25.

Return to the Summary Table.

EPI Non-Blocking Read Data 1

Figure 28-40. EPIRPSTD1 Register

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

Table 28-36. EPIRPSTD1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-13	RESERVED	R	0h	Reserved Reset type: SYSRSn
12-0	POSTCNT	R/W	0h	A write of a non-zero value starts a read operation for that count. Note that it is the software's responsibility to handle address wrap-around. Reading this register provides the current count. A write of 0 cancels a non-blocking read (whether active now or pending). Prior to writing a non-zero value, this register must first be cleared. Reset type: SYSRSn

12 EPISTAT Register (Offset = 60h) [Reset = 00000000h]

EPISTAT is shown in Figure 28-41 and described in Table 28-26.

Return to the Summary Table.

EPI Status

Figure 28-41. EPISTAT Register

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

Table 28-38. EPISTAT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-10	RESERVED	R	0h	Reserved Reset type: SYSRSn
9	CELOW	R	0h	This bit provides information on the clock status when in general-purpose mode and the RDYEN bit is set. Reset type: SYSRSn 0h (R/W) = The external device is not gating the clock. 1h (R/W) = The external device is gating the clock (iRDY is low). Attempts to read or write in this situation are stalled until the clock is enabled or the counter times out as specified by the MAXWAIT field.
8	XFFULL	R	0h	This bit provides information on the XFIFO when in the FIFO sub-mode of the Host Bus n mode with the XFFEN bit set in the EPIHBnCFG register. The EPI0S26 signal reflects the status of this bit. Reset type: SYSRSn 0h (R/W) = The XFIFO is not signaling as full. 1h (R/W) = The XFIFO is signaling as full (the FIFO full signal is high). Attempts to write in this case are stalled until the XFIFO full signal goes low or the counter times out as specified by the MAXWAIT field.
7	XFEMPTY	R	0h	This bit provides information on the XFIFO when in the FIFO sub-mode of the Host Bus n mode with the XFEEN bit set in the EPIHBnCFG register. The EPI0S27 signal reflects the status of this bit. Reset type: SYSRSn 0h (R/W) = The XFIFO is not signaling as empty. 1h (R/W) = The XFIFO is signaling as empty (the FIFO empty signal is high). Attempts to read in this case are stalled until the XFIFO empty signal goes low or the counter times out as specified by the MAXWAIT field.

Table 28-38. EPISTAT Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
6	INITSEQ	R	0h	Initialization Sequence Reset type: SYSRSn 0h (R/W) = The SDRAM interface is not in the wakeup period. 1h (R/W) = The SDRAM interface is running through the wakeup period (greater than 100 microseconds).If an attempt is made to read or write the SDRAM during this period, the access is held off until the wakeup period is complete.
5	WBUSY	R	0h	Write Busy Reset type: SYSRSn 0h (R/W) = The external interface is not performing a write. 1h (R/W) = The external interface is performing a write.
4	NBRBUSY	R	0h	Non-Blocking Read Busy Reset type: SYSRSn 0h (R/W) = The external interface is not performing a non-blocking read. 1h (R/W) = The external interface is performing a non-blocking read, or if the non-blocking read is paused due to a write.
3-1	RESERVED	R	0h	Reserved Reset type: SYSRSn
0	ACTIVE	R	0h	Register Active Reset type: SYSRSn 0h (R/W) = If NBRBUSY is set, the EPIRPSTD0 register is active.If the NBRBUSY bit is clear, then neither EPIRPSTDx register is active. 1h (R/W) = The EPIRPSTD1 register is active.

13 EPIRFIFOCNT Register (Offset = 6Ch) [Reset = 0000000Xh]

EPIRFIFOCNT is shown in Figure 28-42 and described in Table 28-27.

Return to the Summary Table.

EPI Read FIFO Count

Figure 28-42. EPIRFIFOCNT Register

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

Table 28-40. EPIRFIFOCNT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R	0h	Reserved Reset type: SYSRSn
3-0	COUNT	R	Xh	Number of filled entries in the NBRFIFO. Reset type: SYSRSn

14 EPIREADFIFO0 Register (Offset = 70h) [Reset = X0000000h]

EPIREADFIFO0 is shown in Figure 28-43 and described in Table 28-28.

Return to the Summary Table.

EPI Read FIFO 0

Figure 28-43. EPIREADFIFO0 Register

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

Table 28-42. EPIREADFIFO0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DATA	R	Xh	This field contains the data that is at the top of the NBRFIFO. After being read, the NBRFIFO entry is removed. Reset type: SYSRSn

15 EPIREADFIFO1 Register (Offset = 74h) [Reset = X0000000h]

EPIREADFIFO1 is shown in Figure 28-44 and described in Table 28-29.

Return to the Summary Table.

EPI Read FIFO 1

Figure 28-44. EPIREADFIFO1 Register

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

Table 28-44. EPIREADFIFO1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DATA	R	Xh	This field contains the data that is at the top of the NBRFIFO. After being read, the NBRFIFO entry is removed. Reset type: SYSRSn

16 EPIREADFIFO2 Register (Offset = 78h) [Reset = X0000000h]

EPIREADFIFO2 is shown in Figure 28-45 and described in Table 28-30.

Return to the Summary Table.

EPI Read FIFO 2

Figure 28-45. EPIREADFIFO2 Register

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

Table 28-46. EPIREADFIFO2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DATA	R	Xh	This field contains the data that is at the top of the NBRFIFO. After being read, the NBRFIFO entry is removed. Reset type: SYSRSn

ADVANCE INFORMATION

17 EPIREADFIFO3 Register (Offset = 7Ch) [Reset = X0000000h]

EPIREADFIFO3 is shown in Figure 28-46 and described in Table 28-31.

Return to the Summary Table.

EPI Read FIFO 3

Figure 28-46. EPIREADFIFO3 Register

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

Table 28-48. EPIREADFIFO3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DATA	R	Xh	This field contains the data that is at the top of the NBRFIFO. After being read, the NBRFIFO entry is removed. Reset type: SYSRSn

18 EPIREADFIFO4 Register (Offset = 80h) [Reset = X0000000h]

EPIREADFIFO4 is shown in Figure 28-47 and described in Table 28-32.

Return to the Summary Table.

EPI Read FIFO 4

Figure 28-47. EPIREADFIFO4 Register

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

Table 28-50. EPIREADFIFO4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DATA	R	Xh	This field contains the data that is at the top of the NBRFIFO. After being read, the NBRFIFO entry is removed. Reset type: SYSRSn

19 EPIREADFIFO5 Register (Offset = 84h) [Reset = X0000000h]

EPIREADFIFO5 is shown in Figure 28-48 and described in Table 28-33.

Return to the Summary Table.

EPI Read FIFO 5

Figure 28-48. EPIREADFIFO5 Register

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

Table 28-52. EPIREADFIFO5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DATA	R	Xh	This field contains the data that is at the top of the NBRFIFO. After being read, the NBRFIFO entry is removed. Reset type: SYSRSn

20 EPIREADFIFO6 Register (Offset = 88h) [Reset = X0000000h]

EPIREADFIFO6 is shown in Figure 28-49 and described in Table 28-34.

Return to the Summary Table.

EPI Read FIFO 6

Figure 28-49. EPIREADFIFO6 Register

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

Table 28-54. EPIREADFIFO6 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DATA	R	Xh	This field contains the data that is at the top of the NBRFIFO. After being read, the NBRFIFO entry is removed. Reset type: SYSRSn

21 EPIREADFIFO7 Register (Offset = 8Ch) [Reset = X0000000h]

EPIREADFIFO7 is shown in Figure 28-50 and described in Table 28-35.

Return to the Summary Table.

EPI Read FIFO 7

Figure 28-50. EPIREADFIFO7 Register

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

Table 28-56. EPIREADFIFO7 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DATA	R	Xh	This field contains the data that is at the top of the NBRFIFO. After being read, the NBRFIFO entry is removed. Reset type: SYSRSn

22 EPIFIFOLVL Register (Offset = 200h) [Reset = 00000033h]

EPIFIFOLVL is shown in Figure 28-51 and described in Table 28-36.

Return to the Summary Table.

EPI FIFO Level Selects

Figure 28-51. EPIFIFOLVL Register

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

Table 28-58. EPIFIFOLVL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	RESERVED	R	0h	Reserved Reset type: SYSRSn
17	WFERR	R/W	0h	Write Full Error Reset type: SYSRSn 0h (R/W) = The Write Full error interrupt is disabled. Writes are stalled when the WFIFO is full until a space becomes available but an error is not generated. Note that the Cortex-M4 write buffer may hide that stall if no other memory transactions are attempted during that time. 1h (R/W) = This bit enables the Write Full error interrupt (WTFULL in the EPIEISC register) to be generated when a write is attempted and the WFIFO is full. The write stalls until a WFIFO entry becomes available.
16	RSERR	R/W	0h	Note that the configuration of this bit has no effect on non-blocking reads. Reset type: SYSRSn 0h (R/W) = The Read Stalled error interrupt is disabled. Reads behave as normal and are stalled until any preceding writes have completed and the read has returned a result. 1h (R/W) = This bit enables the Read Stalled error interrupt (RSTALL in the EPIEISC register) to be generated when a read is attempted and the WFIFO is not empty. The read is still stalled during the time the WFIFO drains, but this error notifies the application that this excess delay has occurred.
15-7	RESERVED	R	0h	Reserved Reset type: SYSRSn

Table 28-58. EPIFIFOLVL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
6-4	WRFIFO	R/W	3h	Write FIFO Reset type: SYSRSn 0h (R/W) = Interrupt is triggered while WRFIFO is empty. It will be deasserted when not empty. This encoding is optimized for burst of 4 writes. 1h (R/W) = reserved 2h (R/W) = Interrupt is triggered until there are only two slots available. Thus, trigger is deasserted when there are two WRFIFO entries present. This configuration is optimized for bursts of 2. 3h (R/W) = Interrupt is triggered until there is one WRFIFO entry available. This configuration expects only single writes. 4h (R/W) = Trigger interrupt when WRFIFO is not full, meaning trigger will continue to assert until there are four entries in the WRFIFO.
3	RESERVED	R	0h	Reserved Reset type: SYSRSn
2-0	RDFIFO	R/W	3h	This field configures the trigger point for the NBRFIFO. Reset type: SYSRSn 0h (R/W) = reserved 1h (R/W) = Trigger when there are 1 or more entries in the NBRFIFO. 2h (R/W) = Trigger when there are 2 or more entries in the NBRFIFO. 3h (R/W) = Trigger when there are 4 or more entries in the NBRFIFO. 4h (R/W) = Trigger when there are 6 or more entries in the NBRFIFO. 5h (R/W) = Trigger when there are 7 or more entries in the NBRFIFO. 6h (R/W) = Trigger when there are 8 entries in the NBRFIFO. 7h (R/W) = reserved

23 EPIWFIFOCNT Register (Offset = 204h) [Reset = 00000004h]

EPIWFIFOCNT is shown in Figure 28-52 and described in Table 28-37.

Return to the Summary Table.

EPI Write FIFO Count

Figure 28-52. EPIWFIFOCNT Register

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

Table 28-60. EPIWFIFOCNT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R	0h	Reserved Reset type: SYSRSn
2-0	WTAV	R	4h	The number of write transactions available in the WFIFO. When clear, a write is stalled waiting for a slot to become free (from a preceding write completing). Reset type: SYSRSn

ADVANCE INFORMATION

24 EPIDMATXCNT Register (Offset = 208h) [Reset = 00000000h]

EPIDMATXCNT is shown in Figure 28-53 and described in Table 28-38.

Return to the Summary Table.

EPI DMA Transmit Count

Figure 28-53. EPIDMATXCNT Register

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

Table 28-62. EPIDMATXCNT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	Reserved Reset type: SYSRSn
15-0	TXCNT	R/W	0h	This field is used to program the total number of transfers (byte, halfword or word) from the uDMA to the EPI WRFIFO. Reset type: SYSRSn

ADVANCE INFORMATION

25 EPIIM Register (Offset = 210h) [Reset = 00000000h]

EPIIM is shown in Figure 28-54 and described in Table 28-39.

Return to the Summary Table.

EPI Interrupt Mask

Figure 28-54. EPIIM Register

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

Table 28-64. EPIIM Register Field Descriptions

Bit	Field	Type	Reset	Description
31-5	RESERVED	R	0h	Reserved Reset type: SYSRSn
4	DMAWRIM	R/W	0h	Write uDMA Interrupt Mask Reset type: SYSRSn 0h (R/W) = DMAWRRIS in the EPIRIS register is masked and does not cause an interrupt. 1h (R/W) = DMAWRRIS in the EPIRIS register is not masked and can trigger an interrupt to the interrupt controller.
3	DMARDIM	R/W	0h	Read uDMA Interrupt Mask Reset type: SYSRSn 0h (R/W) = DMARDRIS in the EPIRIS register is masked and does not cause an interrupt. 1h (R/W) = DMARDRIS in the EPIRIS register is not masked and can trigger an interrupt to the interrupt controller.
2	WRIM	R/W	0h	Write FIFO Empty Interrupt Mask Reset type: SYSRSn 0h (R/W) = WRRIS in the EPIRIS register is masked and does not cause an interrupt. 1h (R/W) = WRRIS in the EPIRIS register is not masked and can trigger an interrupt to the interrupt controller.
1	RDIM	R/W	0h	Read FIFO Full Interrupt Mask Reset type: SYSRSn 0h (R/W) = RDRIS in the EPIRIS register is masked and does not cause an interrupt. 1h (R/W) = RDRIS in the EPIRIS register is not masked and can trigger an interrupt to the interrupt controller.

Table 28-64. EPIIM Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
0	ERRIM	R/W	0h	Error Interrupt Mask Reset type: SYSRSn 0h (R/W) = ERRIS in the EPIRIS register is masked and does not cause an interrupt. 1h (R/W) = ERRIS in the EPIRIS register is not masked and can trigger an interrupt to the interrupt controller.

26 EPIRIS Register (Offset = 214h) [Reset = 00000004h]

EPIRIS is shown in Figure 28-55 and described in Table 28-40.

Return to the Summary Table.

EPI Raw Interrupt Status

Figure 28-55. EPIRIS Register

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

Table 28-66. EPIRIS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-5	RESERVED	R	0h	Reserved Reset type: SYSRSn
4	DMAWRRIS	R	0h	This bit is cleared by writing a 1 to the DMAWRIC bit in the EPIEISC register. Reset type: SYSRSn 0h (R/W) = The write uDMA has not completed. 1h (R/W) = The write uDMA has completed.
3	DMARDRIS	R	0h	This bit is cleared by writing a 1 to the DMARDIC bit in the EPIEISC register. Reset type: SYSRSn 0h (R/W) = The read uDMA has not completed. 1h (R/W) = The read uDMA has completed.
2	WRRIS	R	1h	This bit is cleared when the level in the WFIFO is above the trigger point programmed by the WRFIFO field. Reset type: SYSRSn 0h (R/W) = The number of available entries in the WFIFO is above the range specified by the WRFIFO field in the EPIFIFOLVL register. 1h (R/W) = The number of available entries in the WFIFO is within the trigger range specified by the WRFIFO field in the EPIFIFOLVL register.

Table 28-66. EPIRIS Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
1	RDRIS	R	0h	This bit is cleared when the level in the NBRFIFO is below the trigger point programmed by the RDFIFO field. Reset type: SYSRSn 0h (R/W) = The number of valid entries in the NBRFIFO is below the trigger range specified by the RDFIFO field in the EPIFIFOLVL register. 1h (R/W) = The number of valid entries in the NBRFIFO is in the trigger range specified by the RDFIFO field in the EPIFIFOLVL register.
0	ERRRIS	R	0h	The error interrupt occurs in the following situations: Reset type: SYSRSn 0h (R/W) = An error has not occurred. 1h (R/W) = A WFIFO Full, a Read Stalled, or a Timeout error has occurred. To determine which error occurred, read the status of the EPI Error Interrupt Status and Clear (EPIEISC) register. This bit is cleared by writing a 1 to the bit in the EPIEISC register that caused the interrupt.

27 EPIMIS Register (Offset = 218h) [Reset = 00000000h]

EPIMIS is shown in Figure 28-56 and described in Table 28-41.

Return to the Summary Table.

EPI Masked Interrupt Status

Figure 28-56. EPIMIS Register

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

Table 28-68. EPIMIS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-5	RESERVED	R	0h	Reserved Reset type: SYSRSn
4	DMAWRMIS	R	0h	This bit is cleared by writing a 1 to the DMAWRIC bit in the EPIEISC register. Reset type: SYSRSn 0h (R/W) = The write uDMA has not completed or the interrupt is masked. 1h (R/W) = The write uDMA has completed and the DMAWRIM bit in the EPIIM register is set, triggering an interrupt to the interrupt controller.
3	DMARDMIS	R	0h	This bit is cleared by writing a 1 to the DMARDIC bit in the EPIEISC register. Reset type: SYSRSn 0h (R/W) = The read uDMA has not completed or the interrupt is masked. 1h (R/W) = The read uDMA has completed and the DMAWRIM bit in the EPIIM register is set, triggering an interrupt to the interrupt controller.
2	WRMIS	R	0h	Write Masked Interrupt Status Reset type: SYSRSn 0h (R/W) = The number of available entries in the WFIFO is above the range specified by the trigger level or the interrupt is masked. 1h (R/W) = The number of available entries in the WFIFO is within the range specified by the trigger level (the WRFIFO field in the EPIFIFOLVL register) and the WRIM bit in the EPIIM register is set, triggering an interrupt to the interrupt controller.

Table 28-68. EPIMIS Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
1	RDMIS	R	0h	Read Masked Interrupt Status Reset type: SYSRSn 0h (R/W) = The number of valid entries in the NBRFIFO is below the range specified by the trigger level or the interrupt is masked. 1h (R/W) = The number of valid entries in the NBRFIFO is within the range specified by the trigger level (the RDFIFO field in the EPIFIFOLVL register) and the RDIM bit in the EPIIM register is set, triggering an interrupt to the interrupt controller.
0	ERRMIS	R	0h	Error Masked Interrupt Status Reset type: SYSRSn 0h (R/W) = An error has not occurred or the interrupt is masked. 1h (R/W) = A WFIFO Full, a Read Stalled, or a Timeout error has occurred and the ERIM bit in the EPIIM register is set, triggering an interrupt to the interrupt controller.

28 EPIEISC Register (Offset = 21Ch) [Reset = 00000000h]

EPIEISC is shown in Figure 28-57 and described in Table 28-42.

Return to the Summary Table.

EPI Error and Interrupt Status and Clear

Figure 28-57. EPIEISC Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED			DMAWRIC	DMARDIC	WTFULL	RSTALL	TOUT
R-0h			W1C-0h	W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h

Table 28-70. EPIEISC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-5	RESERVED	R	0h	Reserved Reset type: SYSRSn
4	DMAWRIC	W1C	0h	Writing a 1 to this bit clears the DMAWRRIS bit in the EPIRIS register and the DMAWRMIS bit in the EPIMIS register. Reset type: SYSRSn
3	DMARDIC	W1C	0h	Writing a 1 to this bit clears the DMARDRIS bit in the EPIRIS register and the DMARDMIS bit in the EPIMIS register. Reset type: SYSRSn
2	WTFULL	R/W1C	0h	Writing a 1 to this bit clears it, as well as the ERRRIS and ERIM bits. Reset type: SYSRSn 0h (R/W) = The WFERR bit is not enabled or no writes are stalled. 1h (R/W) = The WFERR bit is enabled and a write is stalled due to the WFIFO being full.
1	RSTALL	R/W1C	0h	Writing a 1 to this bit clears it, as well as the ERRRIS and ERIM bits. Reset type: SYSRSn 0h (R/W) = The RSERR bit is not enabled or no pending reads are stalled. 1h (R/W) = The RSERR bit is enabled and a pending read is stalled due to writes in the WFIFO.
0	TOUT	R/W1C	0h	This bit is the timeout error source. The timeout error occurs when the XFIFO not-ready signals hold a transaction for more than the count in the MAXWAIT field (when not 0). Writing a 1 to this bit clears it, as well as the ERRRIS and ERIM bits. Reset type: SYSRSn 0h (R/W) = No timeout error has occurred. 1h (R/W) = A timeout error has occurred.

28.2.3 EPI_REGS_SDRAMCFG Registers

Table 28-43 lists the memory-mapped registers for the EPI_REGS_SDRAMCFG registers. All register offset addresses not listed in Table 28-43 should be considered as reserved locations and the register contents should not be modified.

Table 28-71. EPI_REGS_SDRAMCFG Registers

Offset	Acronym	Register Name	Write Protection	Section
10h	EPISDRAMCFG	EPI SDRAM Configuration		Go

Complex bit access types are encoded to fit into small table cells. Table 28-44 shows the codes that are used for access types in this section.

Table 28-72. EPI_REGS_SDRAMCFG Access Type Codes

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

1 EPISDRAMCFG Register (Offset = 10h) [Reset = 82EE0000h]

EPISDRAMCFG is shown in Figure 28-58 and described in Table 28-45.

Return to the Summary Table.

EPI SDRAM Configuration

Figure 28-58. EPISDRAMCFG Register

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

Table 28-74. EPISDRAMCFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	FREQ	R/W	2h	This field configures the frequency range used for delay references by internal counters. This EPI frequency is the system frequency with the divider programmed by the COUNT0 bit in the EPIBAUDn register bit. This field affects the power up, precharge, and auto refresh delays. This field does not affect the refresh counting, which is configured separately using the RFSH field (and is based on system clock rate and number of rows per bank). The ranges are: Reset type: SYSRSn 0h (R/W) = 0 to 15 MHz 1h (R/W) = 15 to 30 MHz 2h (R/W) = 30 to 50 MHz 3h (R/W) = 50 to 100 MHz
29-27	RESERVED	R	0h	Reserved Reset type: SYSRSn
26-16	RFSH	R/W	2EEh	This field contains the refresh counter in EPI clocks. The reset value of 0x2EE provides a refresh period of 64 ms when using a 50 MHz EPI clock. Reset type: SYSRSn
15-10	RESERVED	R	0h	Reserved Reset type: SYSRSn
9	SLEEP	R/W	0h	Sleep Mode Reset type: SYSRSn 0h (R/W) = No effect. 1h (R/W) = The SDRAM is put into low power state, but is self-refreshed.
8-2	RESERVED	R	0h	Reserved Reset type: SYSRSn

ADVANCE INFORMATION

Table 28-74. EPISDRAMCFG Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
1-0	SIZE	R/W	0h	The value of this field affects address pins and behavior. Reset type: SYSRSn 0h (R/W) = 64 megabits (8MB) 1h (R/W) = 128 megabits (16MB) 2h (R/W) = 256 megabits (32MB) 3h (R/W) = 512 megabits (64MB)

28.2.4 EPI_REGS_HB8CFG Registers

Table 28-46 lists the memory-mapped registers for the EPI_REGS_HB8CFG registers. All register offset addresses not listed in Table 28-46 should be considered as reserved locations and the register contents should not be modified.

Table 28-75. EPI_REGS_HB8CFG Registers

Offset	Acronym	Register Name	Write Protection	Section
10h	EPIHB8CFG	EPI Host-Bus 8 Configuration		Go
14h	EPIHB8CFG2	EPI Host-Bus 8 Configuration 2		Go
308h	EPIHB8CFG3	EPI Host-Bus 8 Configuration 3		Go
30Ch	EPIHB8CFG4	EPI Host-Bus 8 Configuration 4		Go
310h	EPIHB8TIME	EPI Host-Bus 8 Timing Extension		Go
314h	EPIHB8TIME2	EPI Host-Bus 8 Timing Extension		Go
318h	EPIHB8TIME3	EPI Host-Bus 8 Timing Extension		Go
31Ch	EPIHB8TIME4	EPI Host-Bus 8 Timing Extension		Go
360h	EPIHBPSRAM	EPI Host-Bus PSRAM		Go

Complex bit access types are encoded to fit into small table cells. Table 28-47 shows the codes that are used for access types in this section.

Table 28-76. EPI_REGS_HB8CFG Access Type Codes

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

1 EPIHB8CFG Register (Offset = 10h) [Reset = 0008FF00h]

EPIHB8CFG is shown in Figure 28-59 and described in Table 28-48.

Return to the Summary Table.

EPI Host-Bus 8 Configuration

Figure 28-59. EPIHB8CFG Register

31	30	29	28	27	26	25	24
CLKGATE	CLKGATEI	CLKINV	RDYEN	IRDYINV	RESERVED		
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R-0h		
23	22	21	20	19	18	17	16
XFFEN	XFEEN	WRHIGH	RDHIGH	ALEHIGH	RESERVED		RESERVED
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-1h	R/W-0h		R-0h
15	14	13	12	11	10	9	8
MAXWAIT							
R/W-FFh							
7	6	5	4	3	2	1	0
WRWS		RDWS		RESERVED		MODE	
R/W-0h		R/W-0h		R-0h		R/W-0h	

Table 28-78. EPIHB8CFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31	CLKGATE	R/W	0h	A software application should only set the CLKGATE bit when there are no pending transfers or no EPI register access has been issued. Reset type: SYSRSn 0h (R/W) = The EPI clock is free running. 1h (R/W) = The EPI clock is held low.
30	CLKGATEI	R/W	0h	Note that EPI0S32 is an iRDY signal if RDYEN is set. CLKGATEI is ignored if CLKPIN is 0 or if the COUNT0 field in the EPIBAUD register is cleared. Reset type: SYSRSn 0h (R/W) = The EPI clock is free running. 1h (R/W) = The EPI clock is output only when there is data to write or read (current transaction) otherwise the EPI clock is held low.
29	CLKINV	R/W	0h	Invert Output Clock Enable Reset type: SYSRSn 0h (R/W) = No effect. 1h (R/W) = Invert EPI clock to ensure the rising edge is centered for outbound signal's setup and hold. Inbound signal is captured on rising edge EPI clock.
28	RDYEN	R/W	0h	Input Ready Enable Reset type: SYSRSn 0h (R/W) = No effect. 1h (R/W) = An external ready can be used to control the continuation of the current access. If this bit is set and the iRDY signal (EPIS032) is low, the current access is stalled.

Table 28-78. EPIHB8CFG Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
27	IRDYINV	R/W	0h	Input Ready Invert Reset type: SYSRSn 0h (R/W) = No effect. 1h (R/W) = Invert the polarity of incoming external ready (iRDY signal). If this bit is set and the iRDY signal (EPIS032) is high the current access is stalled.
26-24	RESERVED	R	0h	Reserved Reset type: SYSRSn
23	XFFEN	R/W	0h	External FIFO FULL Enable Reset type: SYSRSn 0h (R/W) = No effect. 1h (R/W) = An external FIFO full signal can be used to control write cycles. If this bit is set and the FFULL full signal is high, XFIFO writes are stalled.
22	XFEEN	R/W	0h	External FIFO EMPTY Enable Reset type: SYSRSn 0h (R/W) = No effect. 1h (R/W) = An external FIFO empty signal can be used to control read cycles. If this bit is set and the FEMPTY signal is high, XFIFO reads are stalled.
21	WRHIGH	R/W	0h	WRITE Strobe Polarity Reset type: SYSRSn 0h (R/W) = The WRITE strobe for CS0n is WRn (active Low). 1h (R/W) = The WRITE strobe for CS0n is WR (active High).
20	RDHIGH	R/W	0h	READ Strobe Polarity Reset type: SYSRSn 0h (R/W) = The READ strobe for CS0n is RDn (active Low). 1h (R/W) = The READ strobe for CS0n is RD (active High).
19	ALEHIGH	R/W	1h	ALE Strobe Polarity Reset type: SYSRSn 0h (R/W) = The address latch strobe for CS0n accesses is ALEn (active Low). 1h (R/W) = The address latch strobe for CS0n accesses is ALE (active High).
18-17	RESERVED	R/W	0h	Reserved
16	RESERVED	R	0h	Reserved Reset type: SYSRSn
15-8	MAXWAIT	R/W	FFh	This field defines the maximum number of external clocks to wait while an external FIFO ready signal is holding off a transaction (FFULL and FEMPTY). When the MAXWAIT value is reached the ERRRIS interrupt status bit is set in the EPIRIS register. When this field is clear, the transaction can be held off forever without a system interrupt. When the MODE field is configured to be 0x2 and the BLKEN bit is set in the EPICFG register, enabling HB8 mode, this field defaults to 0xFF. Reset type: SYSRSn

Table 28-78. EPIHB8CFG Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
7-6	WRWS	R/W	0h	This field adds wait states to the data phase of CS0n (the address phase is not affected). The effect is to delay the rising edge of WRn (or the falling edge of WR). Each wait state adds 2 EPI clock cycles to the access time. The WRWSM bit in the EPIHB8TIME register can decrease the number of wait states by 1 EPI clock cycle for greater granularity. This field is not applicable in BURST mode. This field is used in conjunction with the EPIBAUD register. Reset type: SYSRSn 0h (R/W) = Active WRn is 2 EPI clocks. 1h (R/W) = Active WRn is 4 EPI clocks. 2h (R/W) = Active WRn is 6 EPI clocks. 3h (R/W) = Active WRn is 8 EPI clocks.
5-4	RDWS	R/W	0h	This field adds wait states to the data phase of CS0n (the address phase is not affected). The effect is to delay the rising edge of RDn/Oen (or the falling edge of RD). Each wait state adds 2 EPI clock cycles to the access time. The RDWSM bit in the EPIHB8TIME register can decrease the number of wait states by 1 EPI clock cycle for greater granularity. This field is not applicable in BURST mode. This field is used in conjunction with the EPIBAUD register Reset type: SYSRSn 0h (R/W) = Active RDn is 2 EPI clocks. 1h (R/W) = Active RDn is 4 EPI clocks. 2h (R/W) = Active RDn is 6 EPI clocks. 3h (R/W) = Active RDn is 8 EPI clocks.
3-2	RESERVED	R	0h	Reserved Reset type: SYSRSn
1-0	MODE	R/W	0h	This field determines which of four Host Bus 8 sub-modes to use. Sub-mode use is determined by the connected external peripheral. See for information on how this bit field affects the operation of the EPI signals. When used with multiple chip select option and the CSBAUD bit is set to 1 in the EPIHB8CFG2 register, this configuration is for CS0n. If the multiple chip select option is enabled and CSBAUD is clear, all chip-selects use the MODE encoding programmed in this register. Reset type: SYSRSn 0h (R/W) = ADMUX - AD[7:0]. Data and Address are muxed. 1h (R/W) = ADNONMUX - D[7:0]. Data and address are separate. 2h (R/W) = Continuous Read - D[7:0]. This mode is the same as ADNONMUX, but uses address switch for multiple reads instead of OEn strobing. 3h (R/W) = XFIFO - D[7:0]. This mode adds XFIFO controls with sense of XFIFO full and XFIFO empty. This mode uses no address or ALE. The XFIFO can only be used in asynchronous mode.

2 EPIHB8CFG2 Register (Offset = 14h) [Reset = 00080000h]

EPIHB8CFG2 is shown in Figure 28-60 and described in Table 28-49.

Return to the Summary Table.

EPI Host-Bus 8 Configuration 2

Figure 28-60. EPIHB8CFG2 Register

31	30	29	28	27	26	25	24
RESERVED				CSCFGEXT	CSBAUD	CSCFG	
R-0h				R/W-0h	R/W-0h	R/W-0h	
23	22	21	20	19	18	17	16
RESERVED		WRHIGH	RDHIGH	ALEHIGH	RESERVED		RESERVED
R-0h		R/W-0h	R/W-0h	R/W-1h	R/W-0h		R-0h
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
WRWS		RDWS		RESERVED	RESERVED	MODE	
R/W-0h		R/W-0h		R/W-0h	R-0h	R/W-0h	

Table 28-80. EPIHB8CFG2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	RESERVED	R	0h	Reserved Reset type: SYSRSn
27	CSCFGEXT	R/W	0h	This field is used in conjunction with CSCFG, to extend the chip select options, and ALE format. The values 0x0 through 0x3 are from the CSCFG field. The CSCFGEXT bit extends the values to 0x7. Reset type: SYSRSn 0h (R/W) = CSCFG bit field is used in chip select configuration. 1h (R/W) = The CSCFG bit field is extended with CSCFGEXT representing the MSB.
26	CSBAUD	R/W	0h	This bit is only valid when the CSCFGEXT + CSCFG field is programmed to 0x2 or 0x3, 0x5 or 0x6. This bit configures the baud rate settings for CS0n, CS1n, CS2n, and CS3n. This bit must also be set to allow different sub-mode configurations on chip-selects. If this bit is clear, all chip-select sub-modes are based on the MODE encoding defined in the EPI8HBCFG register. If the CSBAUD bit is set in the EPIHBnCFG2 register and dual- or quad-chip selects are enabled, then the individual chip selects can use different clock frequencies, wait states and strobe polarity. Reset type: SYSRSn 0h (R/W) = Same Baud Rate and Same Sub-ModeAll CSn use the baud rate for the external bus that is defined by the COUNT0 field in the EPIBAUD register and the sub-mode programmed in the MODE field of the EPIHB8CFG register. 1h (R/W) = Different Baud RatesCS0n uses the baud rate for the external bus that is defined by the COUNT0 field in the EPIBAUD register. CS1n uses the baud rate defined by the COUNT1 field in the EPIBAUD register.CS2n uses the baud rate for the external bus that is defined by the COUNT0 field in the EPIBAUD2 register. CS3n uses the baud rate defined by the COUNT1 field in the EPIBAUD2 register.In addition, the sub-modes for each chip select are individually programmed in their respective EPIHB8CFGn registers.

Table 28-80. EPIHB8CFG2 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
25-24	CSCFG	R/W	0h	This field controls the chip select options, including an ALE format, a single chip select, two chip selects, and an ALE combined with two chip selects. These bits are also used in combination with the CSCFGEXT bit for further configurations, including quad- chip select. Reset type: SYSRSn 0h (R/W) = ALE Configuration. EPI0S30 is used as an address latch (ALE). The ALE signal is generally used when the address and data are muxed (HB8MODE field in the EPIHB8CFG register is 0x0). The ALE signal is used by an external latch to hold the address through the bus cycle. 1h (R/W) = CSn Configuration. EPI0S30 is used as a Chip Select (CSn). When using this mode, the address and data are generally not muxed (HB8MODE field in the EPIHB8CFG register is 0x1). However, if address and data muxing is needed, the WR signal (EPI0S29) and the RD signal (EPI0S28) can be used to latch the address when CSn is low. 2h (R/W) = Dual CSn Configuration. EPI0S30 is used as CS0n and EPI0S27 is used as CS1n. Whether CS0n or CS1n is asserted is determined by two methods. If only external RAM or external PER is enabled in the address map, the most significant address bit for a respective external address map controls CS0n or CS1n. If both external RAM and external PER is enabled, CS0n is mapped to PER and CS1n is mapped to RAM. This configuration can be used for a RAM bank split between 2 devices as well as when using both an external RAM and an external peripheral. 3h (R/W) = ALE with Dual CSn Configuration. EPI0S30 is used as address latch (ALE), EPI0S27 is used as CS1n, and EPI0S26 is used as CS0n. Whether CS0n or CS1n is asserted is determined by the most significant address bit for a respective external address map.
23-22	RESERVED	R	0h	Reserved Reset type: SYSRSn
21	WRHIGH	R/W	0h	This field is used if the CSBAUD bit in the EPIHB8CFG2 register is enabled. Reset type: SYSRSn 0h (R/W) = The WRITE strobe for CS1n accesses is WRn (active Low). 1h (R/W) = The WRITE strobe for CS1n accesses is WR (active High).
20	RDHIGH	R/W	0h	This field is used if the CSBAUD bit in the EPIHB8CFG2 register is enabled. Reset type: SYSRSn 0h (R/W) = The READ strobe for CS1n accesses is RDn (active Low). 1h (R/W) = The READ strobe for CS1n accesses is RD (active High).
19	ALEHIGH	R/W	1h	This field is used if the CSBAUD bit in the EPIHB8CFG2 register is enabled. Reset type: SYSRSn 0h (R/W) = The address latch strobe for CS1n accesses is ALEn (active Low). 1h (R/W) = The address latch strobe for CS1n accesses is ALE (active High).
18-17	RESERVED	R/W	0h	Reserved
16-8	RESERVED	R	0h	Reserved Reset type: SYSRSn

ADVANCE INFORMATION

Table 28-80. EPIHB8CFG2 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
7-6	WRWS	R/W	0h	This field adds wait states to the data phase of CS1n accesses (the address phase is not affected). The effect is to delay the rising edge of WRn (or the falling edge of WR). Each wait state encoding adds 2 EPI clock cycles to the access time. The WRWSM bit in the EPIHB8TIME2 register can decrease the number of wait states by 1 EPI clock cycle for greater granularity. This field is used if the CSBAUD bit is enabled in the EPIHB8CFG2 register. This field is used in conjunction with the EPIBAUD register and is not applicable in BURST mode. Reset type: SYSRSn 0h (R/W) = Active WRn is 2 EPI clocks. 1h (R/W) = Active WRn is 4 EPI clocks 2h (R/W) = Active WRn is 6 EPI clocks 3h (R/W) = Active WRn is 8 EPI clocks
5-4	RDWS	R/W	0h	This field adds wait states to the data phase of CS1n accesses (the address phase is not affected). The effect is to delay the rising edge of RDn/Oen (or the falling edge of RD). Each wait state encoding adds 2 EPI clock cycles to the access time. The RDWSM bit in the EPIHB8TIME2 register can decrease the number of states by 1 EPI clock cycle for greater granularity. This field is used if the CSBAUD bit is enabled in the EPIHB8CFG2 register. This field is used in conjunction with the EPIBAUD register and is not applicable in BURST mode. Reset type: SYSRSn 0h (R/W) = Active RDn is 2 EPI clocks 1h (R/W) = Active RDn is 4 EPI clocks 2h (R/W) = Active RDn is 6 EPI clocks 3h (R/W) = Active RDn is 8 EPI clocks
3	RESERVED	R/W	0h	Reserved
2	RESERVED	R	0h	Reserved Reset type: SYSRSn
1-0	MODE	R/W	0h	This field determines which Host Bus 8 sub-mode to use for CS1. Sub-mode use is determined by the externally connected peripheral or memory. The CSBAUD bit must be set to enable this CS1 MODE field. If CSBAUD is clear, all chip- selects use the MODE configuration defined in the EPIHB8CFG register. Reset type: SYSRSn 0h (R/W) = ADMUX - AD[7:0] Data and Address are muxed. 1h (R/W) = ADNONMUX - D[7:0]Data and address are separate. 2h (R/W) = Continuous Read - D[7:0] This mode is the same as ADNONMUX, but uses address switch for multiple reads instead of OE strobing. 3h (R/W) = Reserved

3 EPIHB8CFG3 Register (Offset = 308h) [Reset = 00080000h]

EPIHB8CFG3 is shown in Figure 28-61 and described in Table 28-50.

Return to the Summary Table.

EPI Host-Bus 8 Configuration 3

Figure 28-61. EPIHB8CFG3 Register

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

Table 28-82. EPIHB8CFG3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-22	RESERVED	R	0h	Reserved Reset type: SYSRSn
21	WRHIGH	R/W	0h	This field is used if the CSBAUD bit is enabled in EPIHB8CFG2. Reset type: SYSRSn 0h (R/W) = The WRITE strobe for CS2n accesses is WRn (active Low). 1h (R/W) = The WRITE strobe for CS2n accesses is WR (active High).
20	RDHIGH	R/W	0h	This field is used if the CSBAUD bit is enabled in EPIHB8CFG2. Reset type: SYSRSn 0h (R/W) = The READ strobe for CS2n accesses is RDn (active Low). 1h (R/W) = The READ strobe for CS2n accesses is RD (active High).
19	ALEHIGH	R/W	1h	This field is used if the CSBAUD bit is enabled in EPIHB8CFG2. Reset type: SYSRSn 0h (R/W) = The address latch strobe for CS2n accesses is ADVn (active Low). 1h (R/W) = The address latch strobe for CS2n accesses is ALE (active High).
18-16	RESERVED	R/W	0h	Reserved
15-8	RESERVED	R	0h	Reserved Reset type: SYSRSn

Table 28-82. EPIHB8CFG3 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
7-6	WRWS	R/W	0h	This field adds wait states to the data phase of CS2n accesses (the address phase is not affected). The effect is to delay the rising edge of WRn (or the falling edge of WR). Each wait state adds 2 EPI clock cycles to the access time. The WRWSM bit in the EPIHB8TIME3 register can decrease the number of wait states by 1 EPI clock cycle for greater granularity. This field is used if the CSBAUD bit is enabled in the EPIHB8CFG2 register. This field is not applicable in BURST mode. This field is used in conjunction with the EPIBAUD2 register. Reset type: SYSRSn 0h (R/W) = Active WRn is 2 EPI clocks 1h (R/W) = Active WRn is 4 EPI clocks 2h (R/W) = Active WRn is 6 EPI clocks 3h (R/W) = Active WRn is 8 EPI clocks
5-4	RDWS	R/W	0h	This field adds wait states to the data phase of CS2n accesses (the address phase is not affected). The effect is to delay the rising edge of RDn/Oen (or the falling edge of RD). Each wait state adds 2 EPI clock cycles to the access time. The RDWSM bit in the EPIHB8TIME3 register can decrease the number of wait states by 1 EPI clock cycle for greater granularity. This field is used if the CSBAUD bit is enabled in the EPIHB8CFG2 register. This field is not applicable in BURST mode. This field is used in conjunction with the EPIBAUD2 register. Reset type: SYSRSn 0h (R/W) = Active RDn is 2 EPI clocks 1h (R/W) = Active RDn is 4 EPI clocks 2h (R/W) = Active RDn is 6 EPI clocks 3h (R/W) = Active RDn is 8 EPI clocks
3	RESERVED	R/W	0h	Reserved
2	RESERVED	R	0h	Reserved Reset type: SYSRSn
1-0	MODE	R/W	0h	This field determines which Host Bus 8 sub-mode to use for CS2n in multiple chip-select mode. Sub-mode use is determined by the connected external peripheral. See for information on how this bit field affects the operation of the EPI signals. The CSBAUD bit must be set to enable this CS2n MODE field. If CSBAUD is clear, all chip-selects use the MODE configuration defined in the EPIHB8CFG register. Reset type: SYSRSn 0h (R/W) = ADMUX - AD[7:0] Data and Address are muxed. 1h (R/W) = ADNONMUX - D[7:0]Data and address are separate. 2h (R/W) = Continuous Read - D[7:0] This mode is the same as ADNONMUX, but uses address switch for multiple reads instead of OE strobing. 3h (R/W) = Reserved

4 EPIHB8CFG4 Register (Offset = 30Ch) [Reset = 00080000h]

EPIHB8CFG4 is shown in Figure 28-62 and described in Table 28-51.

Return to the Summary Table.

EPI Host-Bus 8 Configuration 4

Figure 28-62. EPIHB8CFG4 Register

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

Table 28-84. EPIHB8CFG4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-22	RESERVED	R	0h	Reserved Reset type: SYSRSn
21	WRHIGH	R/W	0h	This field is used if the CSBAUD bit is enabled in EPIHB8CFG2. Reset type: SYSRSn 0h (R/W) = The WRITE strobe for CS3n accesses is WRn (active low). 1h (R/W) = The WRITE strobe for CS3n accesses is WR (active high).
20	RDHIGH	R/W	0h	This field is used if the CSBAUD bit is enabled in EPIHB8CFG2. Reset type: SYSRSn 0h (R/W) = The READ strobe for CS3n accesses is RDn (active low). 1h (R/W) = The READ strobe for CS3n accesses is RD (active high).
19	ALEHIGH	R/W	1h	This field is used if the CSBAUD bit is enabled in EPIHB8CFG2 Reset type: SYSRSn 0h (R/W) = The address latch strobe for CS3n accesses is ADVn (active low). 1h (R/W) = The address latch strobe for CS3n accesses is ALE (active high).
18-16	RESERVED	R/W	0h	Reserved
15-8	RESERVED	R	0h	Reserved Reset type: SYSRSn

Table 28-84. EPIHB8CFG4 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
7-6	WRWS	R/W	0h	This field adds wait states to the data phase of CS3n accesses (the address phase is not affected). The effect is to delay the rising edge of WRn (or the falling edge of WR). Each wait state adds 2 EPI clock cycles to the access time. The WRWSM bit in the EPIHB8TIME4 register can decrease the number of wait states by 1 EPI clock cycle for greater granularity. This field is used if the CSBAUD bit is enabled in the EPIHB8CFG2 register. This field is not applicable in BURST mode. This field is used in conjunction with the EPIBAUD2 register. Reset type: SYSRSn 0h (R/W) = Active WRn is 2 EPI clocks 1h (R/W) = Active WRn is 4 EPI clocks 2h (R/W) = Active WRn is 6 EPI clocks 3h (R/W) = Active WRn is 8 EPI clocks
5-4	RDWS	R/W	0h	This field adds wait states to the data phase of CS3n accesses (the address phase is not affected). The effect is to delay the rising edge of RDn/Oen (or the falling edge of RD). Each wait state adds 2 EPI clock cycles to the access time. The RDWSM bit in the EPIHB8TIME4 register can decrease the number of wait states by 1 EPI clock cycle for greater granularity. This field is used when the CSBAUD bit is set in the EPIHB8CFG2 register. This field is not applicable in BURST mode. This field is used in conjunction with the EPIBAUD2 register. Reset type: SYSRSn 0h (R/W) = Active RDn is 2 EPI clocks 1h (R/W) = Active RDn is 4 EPI clocks 2h (R/W) = Active RDn is 6 EPI clocks 3h (R/W) = Active RDn is 8 EPI clocks
3	RESERVED	R/W	0h	Reserved
2	RESERVED	R	0h	Reserved Reset type: SYSRSn
1-0	MODE	R/W	0h	This field determines which Host Bus 8 sub-mode to use for CS3n in multiple chip select mode. Sub-mode use is determined by the connected external peripheral. See for information on how this bit field affects the operation of the EPI signals. The CSBAUD bit must be set to enable this CS3n MODE field. If CSBAUD is clear, all chip-selects use the MODE configuration defined in the EPIHB8CFG register. Reset type: SYSRSn 0h (R/W) = ADMUX - AD[7:0] Data and Address are muxed. 1h (R/W) = ADNONMUX - D[7:0]Data and address are separate. 2h (R/W) = Continuous Read - D[7:0] This mode is the same as ADNONMUX, but uses address switch for multiple reads instead of OE strobing. 3h (R/W) = Reserved

5 EPIHB8TIME Register (Offset = 310h) [Reset = 00422000h]

EPIHB8TIME is shown in Figure 28-63 and described in Table 28-52.

Return to the Summary Table.

EPI Host-Bus 8 Timing Extension

Figure 28-63. EPIHB8TIME Register

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

Table 28-86. EPIHB8TIME Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	RESERVED	R	0h	Reserved Reset type: SYSRSn
25-24	IRDYDLY	R/W	0h	CS0n Input Ready Delay Reset type: SYSRSn 0h (R/W) = reserved 1h (R/W) = Stall begins one EPI clocks past iRDY low being sampled on the rising edge of EPIO clock. 2h (R/W) = Stall begins two EPI clocks past iRDY low being sampled on the rising edge of EPIO clock. 3h (R/W) = Stall begins three EPI clocks past iRDY low being sampled on the rising edge of EPIO clock.
23-19	RESERVED	R	8h	Reserved Reset type: SYSRSn
18-16	RESERVED	R/W	2h	Reserved
15-14	RESERVED	R	0h	Reserved Reset type: SYSRSn
13-12	CAPWIDTH	R/W	2h	Controls the delay between Host- Bus transfers. Reset type: SYSRSn 0h (R/W) = Reserved 1h (R/W) = 1 EPI clock. 2h (R/W) = 2 EPI clock. 3h (R/W) = Reserved
11-5	RESERVED	R	0h	Reserved Reset type: SYSRSn

ADVANCE INFORMATION

Table 28-86. EPIHB8TIME Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
4	WRWSM	R/W	0h	This bit is used with the WRWS field in EPIHB8CFG. This field is not applicable in BURST mode. Reset type: SYSRSn 0h (R/W) = No change in the number of wait state clock cycles programmed in the in WRWS field in EPIHB8CFG register. 1h (R/W) = Wait state value is now (WRWS - 1) where the WRWS field is programmed in EPIHB8CFG.
3-1	RESERVED	R	0h	Reserved Reset type: SYSRSn
0	RDWSM	R/W	0h	Use with RDWS field in the EPIHB8CFG register. This field is not applicable in BURST mode. Reset type: SYSRSn 0h (R/W) = No change in the number of wait state clock cycles programmed in the RDWS field of EPIHB8CFG. 1h (R/W) = Wait state value is now (RDWS - 1) where the RDWS field is programmed in EPIHB8CFG.

6 EPIHB8TIME2 Register (Offset = 314h) [Reset = 0002A000h]

EPIHB8TIME2 is shown in Figure 28-64 and described in Table 28-53.

Return to the Summary Table.

EPI Host-Bus 8 Timing Extension

Figure 28-64. EPIHB8TIME2 Register

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

Table 28-88. EPIHB8TIME2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	RESERVED	R	0h	Reserved Reset type: SYSRSn
25-24	IRDYDLY	R/W	0h	CS1n Input Ready Delay Reset type: SYSRSn 0h (R/W) = reserved 1h (R/W) = Stall begins one EPI clocks past iRDY low being sampled on the rising edge of EPIO clock. 2h (R/W) = Stall begins two EPI clocks past iRDY low being sampled on the rising edge of EPIO clock. 3h (R/W) = Stall begins three EPI clocks past iRDY low being sampled on the rising edge of EPIO clock.
23-19	RESERVED	R	0h	Reserved Reset type: SYSRSn
18-16	RESERVED	R/W	2h	Reserved
15-14	RESERVED	R	2h	Reserved Reset type: SYSRSn 3h (R/W) = Reserved
13-12	CAPWIDTH	R/W	2h	Controls the delay between Host- Bus transfers. Reset type: SYSRSn 0h (R/W) = Reserved 1h (R/W) = 1 EPI clock. 2h (R/W) = 2 EPI clock.
11-5	RESERVED	R	0h	Reserved Reset type: SYSRSn

Table 28-88. EPIHB8TIME2 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
4	WRWSM	R/W	0h	This bit is used with the WRWS field in EPIHB8CFG2. This field is not applicable in BURST mode. Reset type: SYSRSn 0h (R/W) = No change in the number of wait state clock cycles programmed in the in WRWS field in EPIHB8CFG register. 1h (R/W) = Wait state value is now (WRWS - 1) where the WRWS field is programmed in EPIHB8CFG.
3-1	RESERVED	R	0h	Reserved Reset type: SYSRSn
0	RDWSM	R/W	0h	This field is used with RDWS field in EPIHB8CFG2. This bit is not applicable in BURST mode. Reset type: SYSRSn 0h (R/W) = No change in the number of wait state clock cycles programmed in the RDWS field of EPIHB8CFG. 1h (R/W) = Wait state value is now (RDWS - 1) where the RDWS field is programmed in EPIHB8CFG.

7 EPIHB8TIME3 Register (Offset = 318h) [Reset = 0002A000h]

EPIHB8TIME3 is shown in Figure 28-65 and described in Table 28-54.

Return to the Summary Table.

EPI Host-Bus 8 Timing Extension

Figure 28-65. EPIHB8TIME3 Register

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

Table 28-90. EPIHB8TIME3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	RESERVED	R	0h	Reserved Reset type: SYSRSn
25-24	IRDYDLY	R/W	0h	CS2n Input Ready Delay Reset type: SYSRSn 0h (R/W) = reserved 1h (R/W) = Stall begins one EPI clocks past iRDY low being sampled on the rising edge of EPIO clock. 2h (R/W) = Stall begins two EPI clocks past iRDY low being sampled on the rising edge of EPIO clock. 3h (R/W) = Stall begins three EPI clocks past iRDY low being sampled on the rising edge of EPIO clock.
23-19	RESERVED	R	0h	Reserved Reset type: SYSRSn
18-16	RESERVED	R/W	2h	Reserved
15-14	RESERVED	R	2h	Reserved Reset type: SYSRSn
13-12	CAPWIDTH	R/W	2h	Controls the delay between Host- Bus transfers. Reset type: SYSRSn 0h (R/W) = Reserved 1h (R/W) = 1 EPI clock. 2h (R/W) = 2 EPI clock.
11-5	RESERVED	R	0h	Reserved Reset type: SYSRSn

ADVANCE INFORMATION

Table 28-90. EPIHB8TIME3 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
4	WRWSM	R/W	0h	This bit is used with the WRWS field in EPIHB8CFG3. This field is not applicable in BURST mode. Reset type: SYSRSn 0h (R/W) = No change in the number of wait state clock cycles programmed in the in WRWS field in EPIHB8CFG register. 1h (R/W) = Wait state value is now (WRWS - 1) where the WRWS field is programmed in EPIHB8CFG.
3-1	RESERVED	R	0h	Reserved Reset type: SYSRSn
0	RDWSM	R/W	0h	This field is used with RDWS field in EPIHB8CFG3. This bit is not applicable in BURST mode. Reset type: SYSRSn 0h (R/W) = No change in the number of wait state clock cycles programmed in the RDWS field of EPIHB8CFG. 1h (R/W) = Wait state value is now (RDWS - 1) where the RDWS field is programmed in EPIHB8CFG.

8 EPIHB8TIME4 Register (Offset = 31Ch) [Reset = 0002A000h]

EPIHB8TIME4 is shown in Figure 28-66 and described in Table 28-55.

Return to the Summary Table.

EPI Host-Bus 8 Timing Extension

Figure 28-66. EPIHB8TIME4 Register

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

Table 28-92. EPIHB8TIME4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	RESERVED	R	0h	Reserved Reset type: SYSRSn
25-24	IRDYDLY	R/W	0h	CS3n Input Ready Delay Reset type: SYSRSn 0h (R/W) = reserved 1h (R/W) = Stall begins one EPI clocks past iRDY low being sampled on the rising edge of EPIO clock. 2h (R/W) = Stall begins two EPI clocks past iRDY low being sampled on the rising edge of EPIO clock. 3h (R/W) = Stall begins three EPI clocks past iRDY low being sampled on the rising edge of EPIO clock.
23-19	RESERVED	R	0h	Bits [18:16] have the same RTL implementation as the HB16TIMEn register, even though this is not used in HB8 mode. Thus, the reset value of 0x2 is carried over from the PSRAMSZ bits of HB16TIMEn. Reset type: SYSRSn
18-16	RESERVED	R/W	2h	Reserved
15-14	RESERVED	R	2h	Bits [18:16] have the same RTL implementation as the HB16TIMEn register, even though this is not used in HB8 mode. Thus, the reset value of 0x2 is carried over from the PSRAMSZ bits of HB16TIMEn. Reset type: SYSRSn
13-12	CAPWIDTH	R/W	2h	Controls the delay between Host- Bus transfers. Reset type: SYSRSn 0h (R/W) = Reserved 1h (R/W) = 1 EPI clock. 2h (R/W) = 2 EPI clock.
11-5	RESERVED	R	0h	Reserved Reset type: SYSRSn

ADVANCE INFORMATION

Table 28-92. EPIHB8TIME4 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
4	WRWSM	R/W	0h	This bit is used with the WRWS field in EPIHB8CFG4. This field is not applicable in BURST mode. Reset type: SYSRSn 0h (R/W) = No change in the number of wait state clock cycles programmed in the in WRWS field in EPIHB8CFG register. 1h (R/W) = Wait state value is now (WRWS - 1) where the WRWS field is programmed in EPIHB8CFG.
3-1	RESERVED	R	0h	Reserved Reset type: SYSRSn
0	RDWSM	R/W	0h	This field is used with RDWS field in EPIHB8CFG4. This bit is not applicable in BURST mode. Reset type: SYSRSn 0h (R/W) = No change in the number of wait state clock cycles programmed in the RDWS field of EPIHB8CFG. 1h (R/W) = Wait state value is now (RDWS - 1) where the RDWS field is programmed in EPIHB8CFG.

9 EPIHBPSRAM Register (Offset = 360h) [Reset = 00000000h]

EPIHBPSRAM is shown in Figure 28-67 and described in Table 28-56.

Return to the Summary Table.

EPI Host-Bus PSRAM

Figure 28-67. EPIHBPSRAM Register

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

Table 28-94. EPIHBPSRAM Register Field Descriptions

Bit	Field	Type	Reset	Description
31-21	RESERVED	R	0h	Reserved Reset type: SYSRSn
20-0	CR	R/W	0h	During a configuration write, all 21 bits of the CR bit field are written to the PSRAM. During configuration reads, CR bits [15:0] of this register contain the configuration read of the PSRAM. CR[20:16] will not contain valid data. Reset type: SYSRSn

28.2.5 EPI_REGS_HB16CFG Registers

Table 28-57 lists the memory-mapped registers for the EPI_REGS_HB16CFG registers. All register offset addresses not listed in Table 28-57 should be considered as reserved locations and the register contents should not be modified.

Table 28-95. EPI_REGS_HB16CFG Registers

Offset	Acronym	Register Name	Write Protection	Section
10h	EPIHB16CFG	EPI Host-Bus 16 Configuration		Go
14h	EPIHB16CFG2	EPI Host-Bus 16 Configuration 2		Go
308h	EPIHB16CFG3	EPI Host-Bus 16 Configuration 3		Go
30Ch	EPIHB16CFG4	EPI Host-Bus 16 Configuration 4		Go
310h	EPIHB16TIME	EPI Host-Bus 16 Timing Extension		Go
314h	EPIHB16TIME2	EPI Host-Bus 16 Timing Extension		Go
318h	EPIHB16TIME3	EPI Host-Bus 16 Timing Extension		Go
31Ch	EPIHB16TIME4	EPI Host-Bus 16 Timing Extension		Go

Complex bit access types are encoded to fit into small table cells. Table 28-58 shows the codes that are used for access types in this section.

Table 28-96. EPI_REGS_HB16CFG Access Type Codes

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

ADVANCE INFORMATION

1 EPIHB16CFG Register (Offset = 10h) [Reset = 0008FF00h]

EPIHB16CFG is shown in Figure 28-68 and described in Table 28-59.

Return to the Summary Table.

EPI Host-Bus 16 Configuration

Figure 28-68. EPIHB16CFG Register

31	30	29	28	27	26	25	24
CLKGATE	CLKGATEI	CLKINV	RDYEN	IRDYINV	RESERVED		
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R-0h		
23	22	21	20	19	18	17	16
XFFEN	XFEEN	WRHIGH	RDHIGH	ALEHIGH	WRCRE	RDCRE	BURST
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-1h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
MAXWAIT							
R/W-FFh							
7	6	5	4	3	2	1	0
WRWS		RDWS		RESERVED	BSEL	MODE	
R/W-0h		R/W-0h		R/W-0h	R/W-0h	R/W-0h	

Table 28-98. EPIHB16CFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31	CLKGATE	R/W	0h	A software application should only set the CLKGATE bit when there are no pending transfers or no EPI register access has been issued. Reset type: SYSRSn 0h (R/W) = The EPI clock is free running. 1h (R/W) = The EPI clock is held low.
30	CLKGATEI	R/W	0h	Note that EPI0S32 is an iRDY signal if RDYEN is set. CLKGATEI is ignored if CLKPIN is 0 or if the COUNT0 field in the EPIBAUD register is cleared. Reset type: SYSRSn 0h (R/W) = The EPI clock is free running. 1h (R/W) = The EPI clock is output only when there is data to write or read (current transaction) otherwise the EPI clock is held low.
29	CLKINV	R/W	0h	If operating in asynchronous mode, CLKINV must be 0. Reset type: SYSRSn 0h (R/W) = No effect. 1h (R/W) = Invert EPI clock to ensure the rising edge is centered for outbound signal's setup and hold. Inbound signal is captured on rising edge EPI clock.
28	RDYEN	R/W	0h	Input Ready Enable Reset type: SYSRSn 0h (R/W) = No effect. 1h (R/W) = An external ready (iRDY) can be used to control the continuation of the current access. If this bit is set and the iRDY signal (EPIS032) is low, the current access is stalled.

Table 28-98. EPIHB16CFG Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
27	IRDYINV	R/W	0h	Input Ready Invert Reset type: SYSRSn 0h (R/W) = No effect. 1h (R/W) = Invert polarity of incoming external ready. If this bit is set and the iRDY signal (EPIS032) is high the current access is stalled.
26-24	RESERVED	R	0h	Reserved Reset type: SYSRSn
23	XFFEN	R/W	0h	External FIFO FULL Enable Reset type: SYSRSn 0h (R/W) = No effect. 1h (R/W) = An external FIFO full signal can be used to control write cycles. If this bit is set and the FFULL signal is high, XFIFO writes are stalled.
22	XFEEN	R/W	0h	External FIFO EMPTY Enable Reset type: SYSRSn 0h (R/W) = No effect. 1h (R/W) = An external FIFO empty signal can be used to control read cycles. If this bit is set and the FEMPTY signal is high, XFIFO reads are stalled.
21	WRHIGH	R/W	0h	WRITE Strobe Polarity Reset type: SYSRSn 0h (R/W) = The WRITE strobe for CS0n is WRn (active Low). 1h (R/W) = The WRITE strobe for CS0n is WR (active High).
20	RDHIGH	R/W	0h	READ Strobe Polarity Reset type: SYSRSn 0h (R/W) = The READ strobe for CS0n is RDn (active Low). 1h (R/W) = The READ strobe for CS0n is RD (active High).
19	ALEHIGH	R/W	1h	ALE Strobe Polarity Reset type: SYSRSn 0h (R/W) = The address latch strobe for CS0n is ALEn (active Low). 1h (R/W) = The address latch strobe for CS0n is ALE (active High).
18	WRCRE	R/W	0h	Used for PSRAM configuration registers. With WRCRE set, the next transaction by the EPI will be a write of the CR bit field in the EPIHBPSRAM register to the configuration register (CR) of the PSRAM. The WRCRE bit will self clear once the write-enabled CRE access is complete. Reset type: SYSRSn 0h (R/W) = No Action. 1h (R/W) = Start CRE write transaction for CS0n.
17	RDCRE	R/W	0h	Enables read of PSRAM configuration registers. With the RDCRE set, the next access is a read of the PSRAM's Configuration Register (CR). This bit self clears once the read-enabled CRE access is complete. The address for the CRE access is located at EPIHBPSRAM [19:18]. The read data is returned on EPIHBPSRAM [15:0]. Reset type: SYSRSn 0h (R/W) = No Action. 1h (R/W) = Start CRE read transaction for CS0n.

Table 28-98. EPIHB16CFG Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
16	BURST	R/W	0h	Burst mode must be used with an ALE-enabled interface. Burst mode must be used with ADMUX, which is configured by the MODE field in the EPIHB16CFG register. Burst mode is optimized for word-length accesses. Reset type: SYSRSn 0h (R/W) = Burst mode is disabled. 1h (R/W) = Burst mode is enabled for CS0n or single chip access.
15-8	MAXWAIT	R/W	FFh	This field defines the maximum number of external clocks to wait while an external FIFO ready signal is holding off a transaction (FFULL and FEMPTY). When this field is clear, the transaction can be held off forever without a system interrupt. When the MODE field is configured to be 0x3 and the BLKEN bit is set in the EPICFG register, enabling HB16 mode, this field defaults to 0xFF. Reset type: SYSRSn
7-6	WRWS	R/W	0h	This field adds wait states to the data phase of CS0n (the address phase is not affected). The effect is to delay the rising edge of WRn (or the falling edge of WR). Each wait state adds 2 EPI clock cycles to the access time. The WRWSM bit EPIHB16TIME register can decrease the number of wait states by 1 EPI clock cycle for greater granularity. This field is not applicable in BURST mode. This field is used in conjunction with the EPIBAUD register. Reset type: SYSRSn 0h (R/W) = Active WRn is 2 EPI clocks. 1h (R/W) = Active WRn is 4 EPI clocks. 2h (R/W) = Active WRn is 6 EPI clocks. 3h (R/W) = Active WRn is 8 EPI clocks.
5-4	RDWS	R/W	0h	This field adds wait states to the data phase of CS0n (the address phase is not affected). The effect is to delay the rising edge of RDn/Oen (or the falling edge of RD). Each wait state adds 2 EPI clock cycles to the access time. The RDWSM bit in the EPIHB16TIME register can decrease the number of wait states by 1 EPI clock cycle for greater granularity. This field is not applicable in BURST mode. This field is used in conjunction with the EPIBAUD register Reset type: SYSRSn 0h (R/W) = Active RDn is 2 EPI clocks. 1h (R/W) = Active RDn is 4 EPI clocks. 2h (R/W) = Active RDn is 6 EPI clocks. 3h (R/W) = Active RDn is 8 EPI clocks.
3	RESERVED	R/W	0h	Reserved
2	BSEL	R/W	0h	This bit enables byte select operation. If BSEL = 0, byte accesses cannot be executed. Reset type: SYSRSn 0h (R/W) = No Byte SelectsData is read and written as 16 bits. 1h (R/W) = Enable Byte SelectsTwo EPI signals function as byte select signals to allow 8-bit transfers. See for details on which EPI signals are used.

ADVANCE INFORMATION

Table 28-98. EPIHB16CFG Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
1-0	MODE	R/W	0h	This field determines which of three Host Bus 16 sub-modes to use. Submode use is determined by the connected external peripheral. See for information on how this bit field affects the operation of the EPI signals. When used with multiple chip select option and the CSBAUD bit is set to 1 in the EPIHB16CFG2 register, this configuration is for CS0n. If the multiple chip select option is enabled and CSBAUD is clear, all chip-selects use the MODE encoding programmed in this register. Reset type: SYSRSn 0h (R/W) = ADMUX - AD[15:0]Data and Address are muxed. 1h (R/W) = ADNONMUX - D[15:0]Data and address are separate. This mode is not practical in HB16 mode for normal peripherals because there are generally not enough address bits available. 2h (R/W) = Continuous Read - D[15:0]This mode is the same as ADNONMUX, but uses address switch for multiple reads instead of OEn strobing. This mode is not practical in HB16 mode for normal SRAMs because there are generally not enough address bits available. 3h (R/W) = XFIFO - D[15:0]This mode adds XFIFO controls with sense of XFIFO full and XFIFO empty. This mode uses no address or ALE.Note that the XFIFO can only be used in asynchronous mode.

2 EPIHB16CFG2 Register (Offset = 14h) [Reset = 00080000h]

EPIHB16CFG2 is shown in Figure 28-69 and described in Table 28-60.

Return to the Summary Table.

EPI Host-Bus 16 Configuration 2

Figure 28-69. EPIHB16CFG2 Register

31	30	29	28	27	26	25	24
RESERVED				CSCFGEXT	CSBAUD	CSCFG	
R-0h				R/W-0h	R/W-0h	R/W-0h	
23	22	21	20	19	18	17	16
RESERVED		WRHIGH	RDHIGH	ALEHIGH	WRCRE	RDCRE	BURST
R-0h		R/W-0h	R/W-0h	R/W-1h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
WRWS		RDWS		RESERVED	RESERVED	MODE	
R/W-0h		R/W-0h		R/W-0h	R-0h	R/W-0h	

Table 28-100. EPIHB16CFG2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	RESERVED	R	0h	Reserved Reset type: SYSRSn
27	CSCFGEXT	R/W	0h	This field is used in conjunction with CSCFG, to extend the chip select options, and ALE format. The values 0x0 through 0x3 are from the CSCFG field. The CSCFGEXT bit extends the values to 0x7. Reset type: SYSRSn 0h (R/W) = CSCFG bit field is used in chip select configuration. 1h (R/W) = The CSCFG bit field is extended with CSCFGEXT representing the MSB.
26	CSBAUD	R/W	0h	This bit is only valid when the CSCFGEXT + CSCFG field is programmed to 0x2 or 0x3, 0x5 or 0x6. This bit configures the baud rate settings for CS0n, CS1n, CS2n, and CS3n. This bit must also be set to allow different sub-mode configurations on chip-selects. If this bit is clear, all chip-select sub-modes are based on the MODE encoding defined in the EPI8HBCFG register. If the CSBAUD bit is set in the EPIHBnCFG2 register and dual- or quad-chip selects are enabled, then the individual chip selects can use different clock frequencies, wait states and strobe polarity. Reset type: SYSRSn 0h (R/W) = Same Baud Rate and Same Sub-Mode. All CSn use the baud rate for the external bus that is defined by the COUNT0 field in the EPIBAUD register and the sub-mode programmed in the MODE field of the EPIHB16CFG register. 1h (R/W) = Different Baud Rates. CS0n uses the baud rate for the external bus that is defined by the COUNT0 field in the EPIBAUD register. CS1n uses the baud rate defined by the COUNT1 field in the EPIBAUD register.CS2n uses the baud rate for the external bus that is defined by the COUNT0 field in the EPIBAUD2 register. CS3n uses the baud rate defined by the COUNT1 field in the EPIBAUD2 register.In addition, the sub-modes for each chip select are individually programmed in their respective EPIHB16CFGn registers.

Table 28-100. EPIHB16CFG2 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
25-24	CSCFG	R/W	0h	This field controls the chip select options, including an ALE format, a single chip select, two chip selects, and an ALE combined with two chip selects. These bits are also used in combination with the CSCFGEXT bit for further configurations, including quad- chip select. Reset type: SYSRSn 0h (R/W) = ALE Configuration. EPI0S30 is used as an address latch (ALE). When using this mode, the address and data should be muxed (HB16MODE field in the EPIHB16CFG register should be configured to 0x0). If needed, the address can be latched by external logic. 1h (R/W) = CSn Configuration. EPI0S30 is used as a Chip Select (CSn). When using this mode, the address and data should not be muxed (MODE field in the EPIHB16CFG register should be configured to 0x1). In this mode, the WR signal (EPI0S29) and the RD signal (EPI0S28) are used to latch the address when CSn is low. 2h (R/W) = Dual CSn Configuration. EPI0S30 is used as CS0n and EPI0S27 is used as CS1n. Whether CS0n or CS1n is asserted is determined by the most significant address bit for a respective external address map. This configuration can be used for a RAM bank split between 2 devices as well as when using both an external RAM and an external peripheral. 3h (R/W) = ALE with Dual CSn Configuration. EPI0S30 is used as address latch (ALE), EPI0S27 is used as CS1n, and EPI0S26 is used as CS0n. Whether CS0n or CS1n is asserted is determined by the most significant address bit for a respective external address map.
23-22	RESERVED	R	0h	Reserved Reset type: SYSRSn
21	WRHIGH	R/W	0h	This field is used if CSBAUD bit of the EPIHB16CFG2 register is enabled. Reset type: SYSRSn 0h (R/W) = The WRITE strobe for CS1n accesses is WRn (active Low). 1h (R/W) = The WRITE strobe for CS1n accesses is WR (active High).
20	RDHIGH	R/W	0h	This field is used if CSBAUD bit of the EPIHB16CFG2 register is enabled. Reset type: SYSRSn 0h (R/W) = The READ strobe for CS1n accesses is RDn (active Low). 1h (R/W) = The READ strobe for CS1n accesses is RD (active High).
19	ALEHIGH	R/W	1h	This field is used if CSBAUD bit of the EPIHB16CFG2 register is enabled. Reset type: SYSRSn 0h (R/W) = The address latch strobe for CS1n accesses is ALEn (active Low). 1h (R/W) = The address latch strobe for CS1n accesses is ALE (active High).
18	WRCRE	R/W	0h	Used for the PSRAM configuration registers (CR). With WRCRE set, the next transaction by the EPI is a write of the CR bit field in the EPIHBPSRAM register to the configuration register (CR) of the PSRAM. The WRCRE bit self clears once the write-enabled CRE access is complete. Reset type: SYSRSn 0h (R/W) = No Action. 1h (R/W) = Start CRE write transaction for CS1n.

Table 28-100. EPIHB16CFG2 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
17	RDCRE	R/W	0h	Used for the PSRAM configuration registers (CR). With the RDCRE set, the next access is a read of the PSRAM's Configuration Register (CR). This bit self clears once the CRE access is complete. The address for the CRE access is located at EPIHBPSRAM [19:18]. The read data is returned on EPIHBPSRAM [15:0]. Reset type: SYSRSn 0h (R/W) = No Action. 1h (R/W) = Start CRE read transaction for CS1n.
16	BURST	R/W	0h	Burst mode must be used with an ALE which is configured by programming the CSCFG and CSCFGEXT fields in the EPIHB16CFG2 register. Burst mode must be used in ADMUX, which is set by the MODE field in EPIHB16CFG2. Burst mode is optimized for word-length accesses. Reset type: SYSRSn 0h (R/W) = Burst mode is disabled. 1h (R/W) = Burst mode is enabled for CS1n.
15-8	RESERVED	R	0h	Reserved Reset type: SYSRSn
7-6	WRWS	R/W	0h	This field adds wait states to the data phase of CS1n accesses (the address phase is not affected). The effect is to delay the rising edge of WRn (or the falling edge of WR). Each wait state encoding adds 2 EPI clock cycles to the access time. The WRWSM bit in the EPIHB16TIME2 register can decrease the number of wait states by 1 EPI clock cycle for greater granularity. This field is used if the CSBAUD bit is enabled in the EPIHB16CFG2 register. This field is used in conjunction with the EPIBAUD register and is not applicable in BURST mode. Reset type: SYSRSn 0h (R/W) = Active WRn is 2 EPI clocks 1h (R/W) = Active WRn is 4 EPI clocks. 2h (R/W) = Active WRn is 6 EPI clocks 3h (R/W) = Active WRn is 8 EPI clocks
5-4	RDWS	R/W	0h	This field adds wait states to the data phase of CS1n accesses (the address phase is not affected). The effect is to delay the rising edge of RDn/Oen (or the falling edge of RD). Each wait state encoding adds 2 EPI clock cycles to the access time. The RDWSM bit in the EPIHB16TIME2 register can decrease the number of states by 1 EPI clock cycle for greater granularity. This field is used if the CSBAUD bit is enabled in the EPIHB16CFG2 register. This field is used in conjunction with the EPIBAUD register and is not applicable in BURST mode. Reset type: SYSRSn 0h (R/W) = Active RDn is 2 EPI clocks 1h (R/W) = Active RDn is 4 EPI clocks 2h (R/W) = Active RDn is 6 EPI clocks 3h (R/W) = Active RDn is 8 EPI clocks
3	RESERVED	R/W	0h	Reserved
2	RESERVED	R	0h	Reserved Reset type: SYSRSn

Table 28-100. EPIHB16CFG2 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
1-0	MODE	R/W	0h	This field determines which Host Bus 16 sub-mode to use for CS1n. Sub-mode use is determined by the connected external peripheral. See for information on how this bit field affects the operation of the EPI signals. When used with multiple chip select option this configuration is for CS1n. The CSBAUD bit must be set to enable this CS1n MODE field. If CSBAUD is clear, all chip-selects use the MODE configuration defined in the EPIHB16CFG register. Reset type: SYSRSn 0h (R/W) = reserved 1h (R/W) = ADNONMUX - D[15:0]Data and address are separate. This mode is not practical in HB16 mode for normal peripherals because there are generally not enough address bits available.

3 EPIHB16CFG3 Register (Offset = 308h) [Reset = 00080000h]

EPIHB16CFG3 is shown in Figure 28-70 and described in Table 28-61.

Return to the Summary Table.

EPI Host-Bus 16 Configuration 3

Figure 28-70. EPIHB16CFG3 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED		WRHIGH	RDHIGH	ALEHIGH	WRCRE	RDCRE	BURST
R-0h		R/W-0h	R/W-0h	R/W-1h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
WRWS		RDWS		RESERVED	RESERVED	MODE	
R/W-0h		R/W-0h		R/W-0h	R-0h	R/W-0h	

Table 28-102. EPIHB16CFG3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-22	RESERVED	R	0h	Reserved Reset type: SYSRSn
21	WRHIGH	R/W	0h	This field is used if the CSBAUD bit is enabled in EPIHB16CFG2. Reset type: SYSRSn 0h (R/W) = The WRITE strobe for CS2n accesses is WRn (active Low). 1h (R/W) = The WRITE strobe for CS2n accesses is WR (active High).
20	RDHIGH	R/W	0h	This field is used if the CSBAUD bit is enabled in EPIHB16CFG2. Reset type: SYSRSn 0h (R/W) = The READ strobe for CS2n accesses is RDn (active Low). 1h (R/W) = The READ strobe for CS2n accesses is RD (active High).
19	ALEHIGH	R/W	1h	This field is used if the CSBAUD bit is enabled in EPIHB16CFG2. Reset type: SYSRSn 0h (R/W) = The address latch strobe for CS2n accesses is ADVn (active Low). 1h (R/W) = The address latch strobe for CS2n accesses is ALE (active High).
18	WRCRE	R/W	0h	Used for PSRAM configuration registers. With WRCRE set, the next transaction by the EPI is a write of the CR bit field in the EPIHBPSRAM register to the configuration register (CR) of the PSRAM. The WRCRE bit self clears once the write-enabled CRE access is complete. Reset type: SYSRSn 0h (R/W) = No Action. 1h (R/W) = Start CRE write transaction for CS2n.

Table 28-102. EPIHB16CFG3 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
17	RDCRE	R/W	0h	Used for PSRAM configuration registers. With the RDCRE set, the next access is a read of the PSRAM's Configuration Register (CR). This bit self clears once the CRE access is complete. The address for the CRE access is located at EPIHBPSRAM [19:18]. The read data is returned on EPIHBPSRAM [15:0]. Reset type: SYSRSn 0h (R/W) = No Action. 1h (R/W) = Start CRE read transaction for CS2n.
16	BURST	R/W	0h	Burst mode must be used with an ALE, which is configured by programming the CSCFG and CSCFGEXT fields in the EPIHB16CFG2 register. Burst mode must be used in ADMUX, which is set by the MODE field in EPIHB16CFG3. Burst mode is optimized for word-length accesses. Reset type: SYSRSn 0h (R/W) = Burst mode is disabled. 1h (R/W) = Burst mode is enabled for CS2n.
15-8	RESERVED	R	0h	Reserved Reset type: SYSRSn
7-6	WRWS	R/W	0h	This field adds wait states to the data phase of CS2n accesses (the address phase is not affected). The effect is to delay the rising edge of WRn (or the falling edge of WR). Each wait state adds 2 EPI clock cycles to the access time. The WRWSM bit in the EPIHB16TIME3 register can decrease the number of wait states by 1 EPI clock cycle for greater granularity. This field is used if the EPIHB16CFG2 register. This field is not applicable in BURST mode. This field is used in conjunction with the EPIBAUD2 register. Reset type: SYSRSn 0h (R/W) = Active WRn is 2 EPI clocks 1h (R/W) = Active WRn is 4 EPI clocks 2h (R/W) = Active WRn is 6 EPI clocks 3h (R/W) = Active WRn is 8 EPI clocks
5-4	RDWS	R/W	0h	This field adds wait states to the data phase of CS2n accesses (the address phase is not affected). The effect is to delay the rising edge of RDn/Oen (or the falling edge of RD). Each wait state adds 2 EPI clock cycles to the access time. The RDWSM bit in the EPIHB16TIME3 register can decrease the number of wait states by 1 EPI clock cycle for greater granularity. This field is used if the CSBAUD bit is enabled in the EPIHB16CFG2 register. This field is not applicable in BURST mode. This field is used in conjunction with the EPIBAUD2 register. Reset type: SYSRSn 0h (R/W) = Active RDn is 2 EPI clocks 1h (R/W) = Active RDn is 4 EPI clocks 2h (R/W) = Active RDn is 6 EPI clocks 3h (R/W) = Active RDn is 8 EPI clocks
3	RESERVED	R/W	0h	Reserved
2	RESERVED	R	0h	Reserved Reset type: SYSRSn

ADVANCE INFORMATION

Table 28-102. EPIHB16CFG3 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
1-0	MODE	R/W	0h	This field determines which Host Bus 16 sub-mode to use for CS2n in multiple chip select mode. Sub-mode use is determined by the connected external peripheral. See for information on how this bit field affects the operation of the EPI signals. The CSBAUD bit must be set to enable this CS2n MODE field. If CSBAUD is clear, all chip-selects use the MODE configuration defined in the EPIHB16CFG register. Reset type: SYSRSn 0h (R/W) = ADMUX - AD[7:0] Data and Address are muxed. 1h (R/W) = ADNONMUX - D[7:0]Data and address are separate. 2h (R/W) = Continuous Read - D[7:0] This mode is the same as ADNONMUX, but uses address switch for multiple reads instead of OE strobing. 3h (R/W) = Reserved

4 EPIHB16CFG4 Register (Offset = 30Ch) [Reset = 00080000h]

EPIHB16CFG4 is shown in Figure 28-71 and described in Table 28-62.

Return to the Summary Table.

EPI Host-Bus 16 Configuration 4

Figure 28-71. EPIHB16CFG4 Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED		WRHIGH	RDHIGH	ALEHIGH	WRCRE	RDCRE	BURST
R-0h		R/W-0h	R/W-0h	R/W-1h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
WRWS		RDWS		RESERVED		MODE	
R/W-0h		R/W-0h		R-0h		R/W-0h	

Table 28-104. EPIHB16CFG4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-22	RESERVED	R	0h	Reserved Reset type: SYSRSn
21	WRHIGH	R/W	0h	This field is used if the CSBAUD bit is enabled in EPIHB16CFG2. Reset type: SYSRSn 0h (R/W) = The WRITE strobe for CS3n accesses is WRn (active Low). 1h (R/W) = The WRITE strobe for CS3n accesses is WR (active High).
20	RDHIGH	R/W	0h	This field is used if the CSBAUD bit is enabled in EPIHB16CFG2. Reset type: SYSRSn 0h (R/W) = The READ strobe for CS3n accesses is RDn (active Low). 1h (R/W) = The READ strobe for CS3n accesses is RD (active High).
19	ALEHIGH	R/W	1h	This field is used if the CSBAUD bit is enabled in EPIHB16CFG2. Reset type: SYSRSn 0h (R/W) = The address latch strobe for CS3n accesses is ADVn (active Low). 1h (R/W) = The address latch strobe for CS3n accesses is ALE (active High).
18	WRCRE	R/W	0h	Used for PSRAM configuration registers. With WRCRE set, the next transaction by the EPI will be a write of the CR bit field in the EPIHBPSRAM register to the configuration register (CR) of the PSRAM. The WRCRE bit will self clear once the write-enabled CRE access is complete. Reset type: SYSRSn 0h (R/W) = No Action. 1h (R/W) = Start CRE write transaction for CS3n.

Table 28-104. EPIHB16CFG4 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
17	RDCRE	R/W	0h	Used for PSRAM configuration registers. With the RDCRE set, the next access is a read of the PSRAM's Configuration Register (CR). This bit self clears once the CRE access is complete. The address for the CRE access is located at EPIHBPSRAM [19:18]. The read data is returned on EPIHBPSRAM [15:0]. Reset type: SYSRSn 0h (R/W) = No Action. 1h (R/W) = Start CRE read transaction for CS3n.
16	BURST	R/W	0h	Burst mode must be used with an ALE, which is configured by programming the CSCFG and CSCFGEXT fields in the EPIHB16CFG2 register. Burst mode must be used in ADMUX, which is set by the MODE field in EPIHB16CFG4. Burst mode is optimized for word-length accesses. Reset type: SYSRSn 0h (R/W) = Burst mode is disabled. 1h (R/W) = Burst mode is enabled for CS3n.
15-8	RESERVED	R	0h	Reserved Reset type: SYSRSn
7-6	WRWS	R/W	0h	This field adds wait states to the data phase of CS2n accesses (the address phase is not affected). The effect is to delay the rising edge of WRn (or the falling edge of WR). Each wait state adds 2 EPI clock cycles to the access time. The WRWSM bit in the EPIHB16TIME4 register can decrease the number of wait states by 1 EPI clock cycle for greater granularity. This field is used if the CSBAUD bit is set in the EPIHB16CFG2 register. This field is not applicable in BURST mode. This field is used in conjunction with the EPIBAUD2 register. Reset type: SYSRSn 0h (R/W) = Active WRn is 2 EPI clocks 1h (R/W) = Active WRn is 4 EPI clocks 2h (R/W) = Active WRn is 6 EPI clocks 3h (R/W) = Active WRn is 8 EPI clocks
5-4	RDWS	R/W	0h	This field adds wait states to the data phase of CS3n accesses (the address phase is not affected). The effect is to delay the rising edge of RDn/Oen (or the falling edge of RD). Each wait state adds 2 EPI clock cycles to the access time. The RDWSM bit in the EPIHB16TIME4 register can decrease the number of wait states by 1 EPI clock cycle for greater granularity. This field is used when the CSBAUD bit is set in the EPIHB16CFG2 register. This field is not applicable in BURST mode. This field is used in conjunction with the EPIBAUD2 register. Reset type: SYSRSn 0h (R/W) = Active RDn is 2 EPI clocks 1h (R/W) = Active RDn is 4 EPI clocks 2h (R/W) = Active RDn is 6 EPI clocks 3h (R/W) = Active RDn is 8 EPI clocks
3-2	RESERVED	R	0h	Reserved Reset type: SYSRSn

Table 28-104. EPIHB16CFG4 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
1-0	MODE	R/W	0h	This field determines which Host Bus 16 sub-mode to use for CS3n in multiple chip select mode. Sub-mode use is determined by the connected external peripheral. See for information on how this bit field affects the operation of the EPI signals. The CSBAUD bit must be set to enable this CS3n MODE field. If CSBAUD is clear, all chip-selects use the MODE configuration defined in the EPIHB16CFG register. Reset type: SYSRSn 0h (R/W) = ADMUX - AD[7:0] Data and Address are muxed. 1h (R/W) = ADNONMUX - D[7:0]Data and address are separate. 2h (R/W) = Continuous Read - D[7:0] This mode is the same as ADNONMUX, but uses address switch for multiple reads instead of OE strobing. 3h (R/W) = Reserved

5 EPIHB16TIME Register (Offset = 310h) [Reset = 00022000h]

EPIHB16TIME is shown in Figure 28-72 and described in Table 28-63.

Return to the Summary Table.

EPI Host-Bus 16 Timing Extension

Figure 28-72. EPIHB16TIME Register

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

Table 28-106. EPIHB16TIME Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	RESERVED	R	0h	Reserved Reset type: SYSRSn
25-24	IRDYDLY	R/W	0h	CS0n Input Ready Delay Reset type: SYSRSn 0h (R/W) = reserved 1h (R/W) = Stall begins one EPI clocks past iRDY low being sampled on the rising edge of EPIO clock. 2h (R/W) = Stall begins two EPI clocks past iRDY low being sampled on the rising edge of EPIO clock. 3h (R/W) = Stall begins three EPI clocks past iRDY low being sampled on the rising edge of EPIO clock.
23-19	RESERVED	R	0h	Reserved Reset type: SYSRSn
18-16	PSRAMSZ	R/W	2h	Defines the row size for the PSRAM controlled by CS0n Reset type: SYSRSn 0h (R/W) = No row size limitation 1h (R/W) = 128 B 2h (R/W) = 256 B 3h (R/W) = 512 B 4h (R/W) = 1024 B 5h (R/W) = 2048 B 6h (R/W) = 4096 B 7h (R/W) = 8192 B
15-14	RESERVED	R	0h	Reserved Reset type: SYSRSn

ADVANCE INFORMATION

Table 28-106. EPIHB16TIME Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
13-12	CAPWIDTH	R/W	2h	Controls the delay between Host- Bus transfers. Reset type: SYSRSn 0h (R/W) = Reserved 1h (R/W) = 1 EPI clock. 2h (R/W) = 2 EPI clock. 3h (R/W) = Reserved
11-5	RESERVED	R	0h	Reserved Reset type: SYSRSn
4	WRWSM	R/W	0h	This bit is used with the WRWS field in EPIHB16CFG. This field is not applicable in BURST mode. Reset type: SYSRSn 0h (R/W) = No change in the number of wait state clock cycles programmed in the in WRWS field in EPIHB16CFG register. 1h (R/W) = Wait state value is now (WRWS - 1) where the WRWS field is programmed in EPIHB16CFG.
3-1	RESERVED	R	0h	Reserved Reset type: SYSRSn
0	RDWSM	R/W	0h	Use with RDWS field in the EPIHB16CFG register. This field is not applicable in BURST mode. Reset type: SYSRSn 0h (R/W) = No change in the number of wait state clock cycles programmed in the RDWS field of EPIHB16CFG. 1h (R/W) = Wait state value is now (RDWS - 1) where the RDWS field is programmed in EPIHB16CFG.

6 EPIHB16TIME2 Register (Offset = 314h) [Reset = 00022000h]

EPIHB16TIME2 is shown in Figure 28-73 and described in Table 28-64.

Return to the Summary Table.

EPI Host-Bus 16 Timing Extension

Figure 28-73. EPIHB16TIME2 Register

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

Table 28-108. EPIHB16TIME2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	RESERVED	R	0h	Reserved Reset type: SYSRSn
25-24	IRDYDLY	R/W	0h	CS1n Input Ready Delay Reset type: SYSRSn 0h (R/W) = reserved 1h (R/W) = Stall begins one EPI clocks past iRDY low being sampled on the rising edge of EPIO clock. 2h (R/W) = Stall begins two EPI clocks past iRDY low being sampled on the rising edge of EPIO clock. 3h (R/W) = Stall begins three EPI clocks past iRDY low being sampled on the rising edge of EPIO clock.
23-19	RESERVED	R	0h	Reserved Reset type: SYSRSn
18-16	PSRAMSZ	R/W	2h	Defines the row size for the PSRAM controlled by CS1n Reset type: SYSRSn 0h (R/W) = No row size limitation 1h (R/W) = 128 B 2h (R/W) = 256 B 3h (R/W) = 512 B 4h (R/W) = 1024 B 5h (R/W) = 2048 B 6h (R/W) = 4096 B 7h (R/W) = 8192 B
15-14	RESERVED	R	0h	Reserved Reset type: SYSRSn

Table 28-108. EPIHB16TIME2 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
13-12	CAPWIDTH	R/W	2h	Controls the delay between Host- Bus transfers. Reset type: SYSRSn 0h (R/W) = Reserved 1h (R/W) = 1 EPI clock. 2h (R/W) = 2 EPI clock. 3h (R/W) = Reserved
11-5	RESERVED	R	0h	Reserved Reset type: SYSRSn
4	WRWSM	R/W	0h	This bit is used with the WRWS field in EPIHB16CFG2. This field is not applicable in BURST mode.. Reset type: SYSRSn 0h (R/W) = No change in the number of wait state clock cycles programmed in the in WRWS field in EPIHB16CFG register. 1h (R/W) = Wait state value is now (WRWS - 1) where the WRWS field is programmed in EPIHB16CFG.
3-1	RESERVED	R	0h	Reserved Reset type: SYSRSn
0	RDWSM	R/W	0h	This field is used with RDWS field in EPIHB16CFG2. This bit is not applicable in BURST mode. Reset type: SYSRSn 0h (R/W) = No change in the number of wait state clock cycles programmed in the RDWS field of EPIHB16CFG. 1h (R/W) = Wait state value is now (RDWS - 1) where the RDWS field is programmed in EPIHB16CFG.

7 EPIHB16TIME3 Register (Offset = 318h) [Reset = 00022000h]

EPIHB16TIME3 is shown in Figure 28-74 and described in Table 28-65.

Return to the Summary Table.

EPI Host-Bus 16 Timing Extension

Figure 28-74. EPIHB16TIME3 Register

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

Table 28-110. EPIHB16TIME3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	RESERVED	R	0h	Reserved Reset type: SYSRSn
25-24	IRDYDLY	R/W	0h	CS2n Input Ready Delay Reset type: SYSRSn 0h (R/W) = reserved 1h (R/W) = Stall begins one EPI clocks past iRDY low being sampled on the rising edge of EPIO clock. 2h (R/W) = Stall begins two EPI clocks past iRDY low being sampled on the rising edge of EPIO clock. 3h (R/W) = Stall begins three EPI clocks past iRDY low being sampled on the rising edge of EPIO clock.
23-19	RESERVED	R	0h	Reserved Reset type: SYSRSn
18-16	PSRAMSZ	R/W	2h	Defines the row size for the PSRAM controlled by CS2n Reset type: SYSRSn 0h (R/W) = No row size limitation 1h (R/W) = 128 B 2h (R/W) = 256 B 3h (R/W) = 512 B 4h (R/W) = 1024 B 5h (R/W) = 2048 B 6h (R/W) = 4096 B 7h (R/W) = 8192 B
15-14	RESERVED	R	0h	Reserved Reset type: SYSRSn

Table 28-110. EPIHB16TIME3 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
13-12	CAPWIDTH	R/W	2h	Controls the delay between Host- Bus transfers. Reset type: SYSRSn 0h (R/W) = Reserved 1h (R/W) = 1 EPI clock. 2h (R/W) = 2 EPI clock. 3h (R/W) = Reserved
11-5	RESERVED	R	0h	Reserved Reset type: SYSRSn
4	WRWSM	R/W	0h	This bit is used with the WRWS field in EPIHB16CFG3. This field is not applicable in BURST mode. Reset type: SYSRSn 0h (R/W) = No change in the number of wait state clock cycles programmed in the in WRWS field in EPIHB16CFG register. 1h (R/W) = Wait state value is now (WRWS - 1) where the WRWS field is programmed in EPIHB16CFG.
3-1	RESERVED	R	0h	Reserved Reset type: SYSRSn
0	RDWSM	R/W	0h	This field is used with RDWS field in EPIHB16CFG3. This bit is not applicable in BURST mode. Reset type: SYSRSn 0h (R/W) = No change in the number of wait state clock cycles programmed in the RDWS field of EPIHB16CFG. 1h (R/W) = Wait state value is now (RDWS - 1) where the RDWS field is programmed in EPIHB16CFG.

ADVANCE INFORMATION

8 EPIHB16TIME4 Register (Offset = 31Ch) [Reset = 00022000h]

EPIHB16TIME4 is shown in Figure 28-75 and described in Table 28-66.

Return to the Summary Table.

EPI Host-Bus 16 Timing Extension

Figure 28-75. EPIHB16TIME4 Register

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

Table 28-112. EPIHB16TIME4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	RESERVED	R	0h	Reserved Reset type: SYSRSn
25-24	IRDYDLY	R/W	0h	CS3n Input Ready Delay Reset type: SYSRSn 0h (R/W) = reserved 1h (R/W) = Stall begins one EPI clocks past iRDY low being sampled on the rising edge of EPIO clock. 2h (R/W) = Stall begins two EPI clocks past iRDY low being sampled on the rising edge of EPIO clock. 3h (R/W) = Stall begins three EPI clocks past iRDY low being sampled on the rising edge of EPIO clock.
23-19	RESERVED	R	0h	Reserved Reset type: SYSRSn
18-16	PSRAMSZ	R/W	2h	Defines the row size for the PSRAM controlled by CS3n Reset type: SYSRSn 0h (R/W) = No row size limitation 1h (R/W) = 128 B 2h (R/W) = 256 B 3h (R/W) = 512 B 4h (R/W) = 1024 B 5h (R/W) = 2048 B 6h (R/W) = 4096 B 7h (R/W) = 8192 B
15-14	RESERVED	R	0h	Reserved Reset type: SYSRSn

Table 28-112. EPIHB16TIME4 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
13-12	CAPWIDTH	R/W	2h	Controls the delay between Host- Bus transfers. Reset type: SYSRSn 0h (R/W) = Reserved 1h (R/W) = 1 EPI clock 2h (R/W) = 2 EPI clock 3h (R/W) = Reserved
11-5	RESERVED	R	0h	Reserved Reset type: SYSRSn
4	WRWSM	R/W	0h	This bit is used with the WRWS field in EPIHB16CFG4. This field is not applicable in BURST mode. Reset type: SYSRSn 0h (R/W) = No change in the number of wait state clock cycles programmed in the in WRWS field in EPIHB16CFG register. 1h (R/W) = Wait state value is now (WRWS - 1) where the WRWS field is programmed in EPIHB16CFG.
3-1	RESERVED	R	0h	Reserved Reset type: SYSRSn
0	RDWSM	R/W	0h	This field is used with RDWS field in EPIHB16CFG4. This bit is not applicable in BURST mode. Reset type: SYSRSn 0h (R/W) = No change in the number of wait state clock cycles programmed in the RDWS field of EPIHB16CFG. 1h (R/W) = Wait state value is now (RDWS - 1) where the RDWS field is programmed in EPIHB16CFG.

Chapter 29

Cyclic Redundancy Check (CRC)

This chapter describes the Cyclic Redundancy Check (CRC) accelerator.

29.1 CRC

The cyclic redundancy check (CRC) accelerator generates signatures for a given data sequence based on the CRC16-CCITT polynomial and CRC32-ISO3309 polynomial.

29.1.1 CRC Overview

The CRC-P accelerator produces CRC signatures for given sequences of data. The CRC16-CCITT and CRC32-ISO3309 polynomial functions are supported. Identical input data sequences result in identical CRC signatures when the CRC is initialized with a fixed seed value. Different sequences of input data, in general, result in different signatures for a given CRC function.

Table 29-1. CRC-P Key Features

Feature	CRC-P
Support for CRC16-CCITT	✓
Support for CRC32-ISO3309 Polynomial	✓
Fast single cycle computation of new CRC output for each data input (no wait states)	X
Support for input / output bit reversal	✓
Support for big endian and little endian operation	✓
Byte, Half-word, or word input to CRCIN	✓
512-word CRCIN_IDX input field in which all addresses are mapped to CRCIN, supporting use of a standard C-style memcpy() routine to load data into the CRC module for data lengths up to 2KB	✓
User-selectable CRC32 polynomial	✓

29.1.1.1 CRC16-CCITT

For CRC16-CCITT, the CRC signature is generated based on the polynomial given in the 16-bit CCITT standard, as shown in the equation below.

$$f(x)=x^{16}+x^{12}+x^{5}+1 \quad (38)$$

The CRC16-CCITT digest size is 16 bits (half-word).

29.1.1.2 CRC32-ISO3309

For CRC32-ISO3309, the CRC signature is generated based on the polynomial given in the ISO3309 Ethernet standard, as shown in the equation below.

$$f(x)=x^{32}+x^{26}+x^{23}+x^{22}+x^{16}+x^{12}+x^{11}+x^{10}+x^{8}+x^{7}+x^{5}+x^{4}+x^{2}+x+1 \quad (39)$$

The digest size for CRC32-ISO3309 is 32 bits.

29.1.2 CRC Operation

The CRC generator is initialized by configuring the desired mode of operation in the CRCCTRL register, followed by writing the seed value to the CRCSEED register. After the seed is loaded to the CRCSEED register, the CRCOUT register will reflect the SEED value loaded to CRCSEED.

Note

If the endianness mode is configured to big endian before the seed value is written to CRCSEED, then the byte order of the seed value written to CRCSEED is swapped when the seed value is loaded into the CRC module.

Once initialized, data can be input into the CRC generator by writing to the CRCIN register using byte (8-bit), half-word (16-bit), or word (32-bit) writes. The CPU or the DMA can be used to move input data into the CRC accelerator.

Note

Byte writes need not be word aligned; a byte write to any byte location in CRCIN will be interpreted the same way (adding the 8 written bits to the computed CRC). Half-word writes also need not be word aligned, but they must be half-word aligned. For example, a half-word can be written to BIT0-BIT15 or BIT16-BIT32 of CRCIN, but not to BIT8-BIT23.

When using the CRC generator to verify a data set, all data to be included in the CRC calculation must be written to the CRCIN register in the same order as was used to calculate the original CRC signature. When using the CRC generator to create a new signature to be used in the future for verification, be sure to load the data the same way and with the same settings when performing verification.

The current CRC output can be read at any time by reading the CRCOUT register.

29.1.2.1 CRC Generator Implementation

The CRC generator is implemented with a set of XOR trees. After a set of 8, 16, or 32 bits is provided to the CRC accelerator by writing to the CRCIN register, a calculation for the whole set of input bits is performed. When new data is written to CRCIN, the CRC generator updates the CRC output in a single cycle when using the CRC peripheral. CRC-P versions require additional cycles. Bus wait states are not required to load data back-to-back into the CRC generator.

29.1.2.2 Configuration

The CRC accelerator supports polynomial selection, bit reversal selection, and byte order (endianness) selection. This section describes these configuration aspects.

The CRC accelerator must be enabled before being used through the PWREN register (see peripheral power enable).

The CRC accelerator is in power domain 1 (PD1), and as such can only be active in RUN or SLEEP mode. If the CRC accelerator is configured to be enabled by application software, and the device enters STOP or STANDBY mode, SYSCTL forces the CRC into a disabled state until the device exists STOP or STANDBY mode. All CRC register contents are retained when the CRC is forced to a disabled state in STOP or STANDBY mode.

The CRC module only runs from the PD1 bus clock (MCLK).

29.1.2.2.1 Polynomial Selection

The desired polynomial (CRC16-CCITT or CRC32-ISO3309) is selected with the POLYSIZE bit in the CRCCTRL register. The polynomial to be used must be selected before loading the seed value and any data values.

When the CRC generator is configured for a 16-bit digest (CRC16-CCITT), the following conditions apply:

- The upper half-word (16 bits) of the CRCSEED register are ignored and only the lower half-word is used
- The upper half-word (16 bits) of the CRCOUT register read back as zero and only the lower half-word is valid

When the CRC generator is configured for CRC32-ISO3309, all 32 bits of CRCSEED and CRCOUT are valid.

29.1.2.2.2 Bit Order

The various CRC standards were defined in the era of main frame computers. At that time, BIT0 was treated as the MSB. In modern computing, BIT0 is typically the LSB.

The Arm Cortex-M33 CPU treats BIT0 as the LSB, as is typical in modern CPUs and MCUs. This sometimes causes confusion, because BIT0 has been treated as the LSB in some cases and as the MSB in other cases. Therefore, the CRC accelerator provides a bit order reversal capability to support both conventions.

Bit order reversal can be enabled by setting the BITREVERSE bit in the CRCCTRL register, giving the following behavior for input and output data:

- **Input data:** The bit order of each input byte written to the CRCIN register is reversed before it is passed to the CRC generator to be used for CRC calculation

- **Output data:** The bit order of the 16-bit or 32-bit CRC result is reversed when read from the CRCOUT register

Note

If input data must be provided bit-reversed, but the output is to be read not reversed, the BITREVERSE bit can be set before loading data to CRCIN and then cleared after all data is written to CRCIN but before the resulting signature is read from CRCOUT. Likewise, it is possible to load input data to CRCIN with BITREVERSE cleared (not reversed), and flip the output before reading it (by setting BITREVERSE before reading CRCOUT).

29.1.2.2.3 Byte Swap

The bit OUTPUT_BYTESWAP in the register CRCCTRL can be used to enable or disable CRC output byte swap. This bit controls whether the output is byte-swapped upon a read of the CRCOUT register. If CRCOUT is accessed as a half-word, and the OUTPUT_BYTESWAP is set to 1, then the two bytes in the 16-bit access are swapped and returned. Using B0, B1, B2 and B3 to identify Byte 0, Byte 1, Byte 2, Byte 3. B1 is returned as B0 and B0 is returned as B1. If CRCOUT is accessed as a word, and the OUTPUT_BYTESWAP is set to 1, then the four bytes in the 32-bit read are swapped. B3 is returned as B0, B2 is returned as B1, B1 is returned as B2 and B0 is returned as B3.

Note that if the CRC POLYSIZE is 16-bit and a 32-bit read of CRCOUT is performed with OUTPUT_BYTESWAP enabled, then the output is: MSB LSB 0x0 0x0 B0 B1. If the CRC POLYSIZE is 16-bit and a 32-bit read of CRCOUT is performed with OUTPUT_BYTESWAP disabled, then the output is: MSB LSB 0x0 0x0 B1 B0.

29.1.2.2.4 Byte Order

When working with half-word or word input data, the input byte order can be configured as either little endian or big endian. The default configuration is little endian. To reverse the byte order when using half-word or word inputs, set the INPUT_ENDIANNESS bit in the CRCCTRL register.

Reversing the endianness will cause the following translation for half-word and word writes:

Table 29-2. CRCIN Byte Order Translation

Endianness	Data Written to CRCIN	Data Applied to CRC Logic
0 (little)	0x1234	0x1234
1 (big)	0x1234	0x3412
0 (little)	0x12345678	0x12345678
1 (big)	0x12345678	0x78563412

Note

If the ENDIANNESS bit is set before the seed value is written to CRCSEED, then the byte order of the seed value written to CRCSEED is also reversed when it is loaded into the CRC, and the seed value will read back reversed when reading the CRCOUT register after writing to the CRCSEED register.

29.1.2.2.5 CRC C Library Compatibility

To simplify loading of data by software into the CRC, the CRC accelerator provides a 512-word (2KB) CRCIN_IDX region. Within the CRCIN_IDX region, a write to any word is re-mapped as, and functionally equivalent to, a write to the CRCIN register. This mechanism enables the use of the standard C library *memcpy()* routine to copy data from SRAM or flash into the CRC, provided that the source data is less than 2KB (2,048 bytes).

29.2 CRC Registers

This Section describes the CRC Registers.

29.2.1 CRC Base Address Table

Table 29-3. CRC Base Address Table

Bit Field Name		DriverLib Name	Base Address
Instance	Structure		
CrcRegs	CRCP_REGS	CRC	0x401B_2000

29.2.2 CRCP_REGS Registers

Table 29-4 lists the memory-mapped registers for the CRCP_REGS registers. All register offset addresses not listed in Table 29-4 should be considered as reserved locations and the register contents should not be modified.

Table 29-4. CRCP_REGS Registers

Offset	Acronym	Register Name	Section
800h	PWREN	Power enable	Go
804h	RSTCTL	Reset Control	Go
814h	STAT	Status Register	Go
1004h	CLKSEL	Clock Select	Go
10FCh	DESC	Module Description	Go
1100h	CRCCTRL	CRC Control Register	Go
1104h	CRCSEED	CRC Seed Register	Go
1108h	CRCIN	CRC Input Data Register	Go
110Ch	CRCOUT	CRC Output Result Register	Go
1110h	CRCPOLY	CRC Polynomial configuration register	Go
1800h + formula	CRCIN_IDX_y	CRC Input Data Array Register	Go

Complex bit access types are encoded to fit into small table cells. Table 29-5 shows the codes that are used for access types in this section.

Table 29-5. CRCP_REGS Access Type Codes

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

1 PWREN Register (Offset = 800h) [Reset = 00000000h]

PWREN is shown in Figure 29-1 and described in Table 29-6.

Return to the Summary Table.

Register to control the power state

Figure 29-1. PWREN Register

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

Table 29-7. PWREN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	KEY	W	0h	KEY to allow Power State Change 26h = KEY to allow write access to this register
23-1	RESERVED	R/W	0h	
0	ENABLE	R/WK	0h	Enable the power [EXT_GPRCM.GPRCM.PWREN.KEY] must be set to 26h to write to this bit. 0h = Disable Power 1h = Enable Power

2 RSTCTL Register (Offset = 804h) [Reset = 00000000h]

RSTCTL is shown in Figure 29-2 and described in Table 29-7.

Return to the Summary Table.

Register to control reset assertion and de-assertion

Figure 29-2. RSTCTL Register

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

Table 29-9. RSTCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	KEY	W	0h	Unlock key B1h = KEY to allow write access to this register
23-2	RESERVED	W	0h	
1	RESETSTKYCLR	WK	0h	Clear the RESETSTKY bit in the STAT register [EXT_GPRCM.GPRCM.RSTCTL.KEY] must be set to B1h to write to this bit. 0h = Writing 0 has no effect 1h = Clear reset sticky bit
0	RESETASSERT	WK	0h	Assert reset to the peripheral [EXT_GPRCM.GPRCM.RSTCTL.KEY] must be set to B1h to write to this bit. 0h = Writing 0 has no effect 1h = Assert reset

ADVANCE INFORMATION

3 STAT Register (Offset = 814h) [Reset = 00000000h]

STAT is shown in Figure 29-3 and described in Table 29-8.

Return to the Summary Table.

peripheral enable and reset status

Figure 29-3. STAT Register

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

Table 29-11. STAT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-17	RESERVED	R	0h	
16	RESETSTKY	R	0h	This bit indicates, if the peripheral was reset, since this bit was cleared by RESETSTKYCLR in the RSTCTL register 0h = The peripheral has not been reset since this bit was last cleared by RESETSTKYCLR in the RSTCTL register 1h = The peripheral was reset since the last bit clear
15-0	RESERVED	R	0h	

4 CLKSEL Register (Offset = 1004h) [Reset = 00000001h]

CLKSEL is shown in Figure 29-4 and described in Table 29-9.

Return to the Summary Table.

Clock source selection

Figure 29-4. CLKSEL Register

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

Table 29-13. CLKSEL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	
0	MCLK_SEL	R	1h	Selects main clock (MCLK) if enabled 0h = Does not select this clock as a source 1h = Select this clock as a source

ADVANCE INFORMATION

5 DESC Register (Offset = 10FCh) [Reset = 00000000h]

DESC is shown in Figure 29-5 and described in Table 29-10.

Return to the Summary Table.

This register identifies the peripheral and its exact version.

Figure 29-5. DESC Register

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

Table 29-15. DESC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	MODULEID	R	2011h	Module identification contains a unique peripheral identification number. The assignments are maintained in a central database for all of the platform modules to ensure uniqueness. 0h = Smallest value FFFFh = Highest possible value
15-12	FEATUREVER	R	8h	Feature Set for the module *instance* 0h = Smallest value Fh = Highest possible value
11-8	INSTNUM	R	0h	Instance Number within the device. This will be a parameter to the RTL for modules that can have multiple instances 0h = Smallest value Fh = Highest possible value
7-4	MAJREV	R	1h	Major rev of the IP 0h = Smallest value Fh = Highest possible value
3-0	MINREV	R	0h	Minor rev of the IP 0h = Smallest value Fh = Highest possible value

6 CRCCTRL Register (Offset = 1100h) [Reset = 00000000h]

CRCCTRL is shown in Figure 29-6 and described in Table 29-11.

Return to the Summary Table.

CRC Control Register. Configuration control of the CRC.

Figure 29-6. CRCCTRL Register

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

Table 29-17. CRCCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-5	RESERVED	R/W	0h	
4	OUTPUT_BYTESWAP	R/W	0h	CRC Output Byteswap Enable. This bit controls whether the output is byte-swapped upon a read of the CRCOUT register. If CRCOUT is accessed as a half-word, and the OUTPUT_BYTESWAP is set to to 1, then the two bytes in the 16-bit access are swapped and returned. B1 is returned as B0 B0 is returned as B1 If CRCOUT is accessed as a word, and the OUTPUT_BYTESWAP is set to 1, then the four bytes in the 32-bit read are swapped. B3 is returned as B0 B2 is returned as B1 B1 is returned as B2 B0 is returned as B3 Note that if the CRC POLYSIZE is 16-bit and a 32-bit read of CRCOUT is performed with OUTPUT_BYTESWAP enabled, then the output is: MSB LSB 0x0 0x0 B0 B1 If the CRC POLYSIZE is 16-bit and a 32-bit read of CRCOUT is performed with OUTPUT_BYTESWAP disabled, then the output is: MSB LSB 0x0 0x0 B1 B0 0h = Output byteswapping is disabled 1h = Output byteswapping is enabled.
3	RESERVED	R/W	0h	
2	INPUT_ENDIANNESS	R/W	0h	CRC Endian. This bit indicates the byte order within a word or half word of input data. 0h = LSB is lowest memory address and first to be processed. 1h = LSB is highest memory address and last to be processed.

ADVANCE INFORMATION

Table 29-17. CRCCTRL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
1	BITREVERSE	R/W	0h	CRC Bit Input and output Reverse. This bit indictes that the bit order of each input byte used for the CRC calculation is reversed before it is passed to the generator, and that the bit order of the calculated CRC is be reversed when read from CRC_RESULT. 0h = Bit order is not reversed. 1h = Bit order is reversed.
0	POLYSIZE	R/W	0h	This bit indicates which CRC calculation is performed by the generator. 0h = CRC-32 ISO-3309 calulation is performed 1h = CRC-16 CCITT is performed

7 CRCSEED Register (Offset = 1104h) [Reset = 00000000h]

CRCSEED is shown in Figure 29-7 and described in Table 29-12.

Return to the Summary Table.

CRC Seed Register. The Data written to this register is used to initialize the CRC result with this SEED value. Note that in 16-bit mode only the lower 16-bits of this value are used. After writing this register the CRC Output Result Register will reflect this value.

Figure 29-7. CRCSEED Register

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

Table 29-19. CRCSEED Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	SEED	W	0h	Seed Data 00000000h = Mnimum value FFFFFFFFh = Maximum value

8 CRCIN Register (Offset = 1108h) [Reset = 00000000h]

CRCIN is shown in Figure 29-8 and described in Table 29-13.

Return to the Summary Table.

CRC Input Data Register. The Data writen to this register is used along with the current CRC result to calculate the next CRC result. This is done in a single clock cycle and requires no wait states. This register can by written as a byte, half word or word transfer and the correct number of bits will be used for the next CRC result.
This register is also mapped to a range of registers starting at 0xTDB_X000 and ending at 0xTDB_XFFF to allow memcpy to be used instead of DMA for CRC calculations that do not exceed the bounds of the memory range of this register.

Figure 29-8. CRCIN Register

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

Table 29-21. CRCIN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DATA	W	0h	Input Data 00000000h = Mnimum value FFFFFFFFh = Maximum value

9 CRCOUT Register (Offset = 110Ch) [Reset = 00000000h]

CRCOUT is shown in Figure 29-9 and described in Table 29-14.

Return to the Summary Table.

CRC Output Result Register. This register stores the result of the currect CRC calculation. Note when configured for 16-bit mode the upper bits will read back 0. Note that if output inversion is set in the CRC Control register it will be applied.

Figure 29-9. CRCOUT Register

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

Table 29-23. CRCOUT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RESULT	R	0h	Result 00000000h = Mnimum value FFFFFFFFh = Maximum value

10 CRCPOLY Register (Offset = 1110h) [Reset = 00000000h]

CRCPOLY is shown in Figure 29-10 and described in Table 29-15.

Return to the Summary Table.

CRC Polynomial configuration register

Figure 29-10. CRCPOLY Register

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

Table 29-25. CRCPOLY Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DATA	04C11DB7h	Polynomial definition	

11 CRCIN_IDX_y Register (Offset = 1800h + formula) [Reset = 00000000h]

CRCIN_IDX_y is shown in Figure 29-11 and described in Table 29-16.

Return to the Summary Table.

This register is dual mapped to CRCIN and is intended to allow operation with C memcpy routine.

Offset = 1800h + (y * 4h); where y = 0h to 1FFh

Figure 29-11. CRCIN_IDX_y Register

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

Table 29-27. CRCIN_IDX_y Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DATA	HW	0h	Input Data 00000000h = Minimum value FFFFFFFFh = Maximum value

30.1 AES

The AES accelerator module accelerates encryption and decryption operations in hardware based on the FIPS PUB 197 AES.

30.1.1 AES Overview

The AES accelerator module performs encryption and decryption of 128-bit data blocks with a 128-bit or 256-bit key in hardware. AES is a symmetric-key block cipher algorithm specified in FIPS PUB 197.

The AES accelerator features include:

- AES 128-bit block encryption and decryption
- Key scheduling in hardware
- Enc/decrypt only modes: CBC, CFB-1, CFB-8, CFB-128, OFB-128, CTR/ICM
- Authentication only modes: CBC-MAC, CMAC
- AES-CCM
- AES-GCM
- AES-CCM and AES-GCM modes support continuation with hold/resume of payload data
- 32-bit word access to provide key data, input data, and output data
- AES ready interrupt
- DMA triggers for input/output data
- Supported in RUN and SLEEP (see the *Operating Modes* section of the device technical reference manual)

A high level block diagram of the AES engine is shown in Figure 30-1. The AES engine consists of a processing core that performs both encryption/decryption as well as Galois field multiplication. The core is driven with configuration and data inputs that software will configure via memory mapped registers.

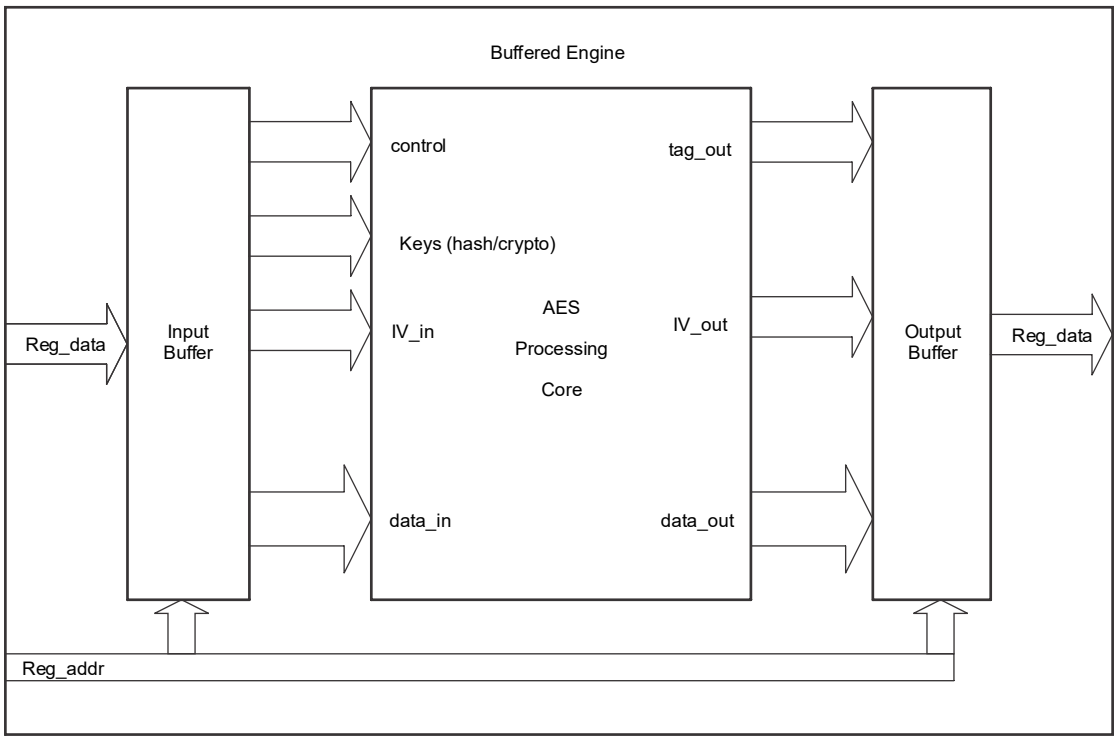


Figure 30-1. AES Block Diagram

30.1.1.1 AESADV Performance

The AESADV accelerator provides fast encryption and decryption of 128-bit blocks. AESADV accelerator performance in both cycles and execution time for block encryption and block decryption (with pregenerated decryption key) is given in Section 30.1.1 . This table assumes that there are no system overheads (delays in supplying next input or reading out available output) that stall the engine.

ADVANCE INFORMATION

Table 30-1. AESADV Hardware Accelerator Key Performance Metrics

AESADV Key Length	Encryption			Decryption		
	Cycles	Time (32MHz)	Time (100MHz)	Cycles	Time (32MHz)	Time (100MHz)
128-bit	76	2.38us	0.76us	76	2.38us	0.76us
256-bit	81	2.53us	0.81us	81	2.53us	0.81us

30.1.2 AESADV Operation

The AESADV engine provides an efficient implementation of the Rijndael cipher (the AES algorithm) and a 128-bit polynomial multiplication (referred here to as 'GHASH', as per the AES-GCM specification). Rijndael is a block cipher with each data block consisting of 128-bits.

AES encryption requires a specific number of rounds depending on the key length. Supported key lengths are 128-bit, and 256 bit, requiring 10, and 14 rounds respectively.

The AESADV engine contains the AES ECB core and a dedicated 32-cycle polynomial multiplication module for performing GHASH operations (when GCM is configured). The polynomial multiplication operates in parallel with the AES core, if there is data available for both modules. This is the case after encryption of the first data block.

30.1.2.1 Loading the Key

Keys can be configured into the engine in one of two ways.

1. Secure key initialization via keystore controller: In this method, the intended AES key is securely transferred from the keystore controller into the engine via a secure private bus connecting the keystore controller and the AES engine. The keystore controller initiates the key transfer and waits for the AES engine to acknowledge transfer completion.
2. Software explicitly configures keys: In this method, software configures the key data into the engine by writing to the KEY0--KEY7 registers, writing 32-bits at a time starting with KEY0. For 128-bit keys, KEY0--KEY3 register will need to be written. For 256-bit keys, KEY0--KEY7 will need to be written.

In order to prevent key stealing attacks by partial modification method, the engine ensures that once a secure key transfer has completed via the keystore, software can no longer explicitly configure/modify the key. This status is provided by the STATUS.KEYWR field. If this field is 0, then software is allowed to write key data. If this field is 1, then software is not allowed to write key data. In order to allow software to write key data, the module has to be reset. The reset operation clears existing key data before new key data can be written by software.

30.1.2.2 Writing Input Data

Input data is written into the engine either through the DATA0, DATA1, DATA2, and DATA3 registers or through the DATA_IN register. If DMA is not being used to automate the input/output transfers (DMA_HS is 0), then CPU software can perform the 128-bit data input by writing 32-bit data to each of the registers DATA0, DATA1, DATA2 and DATA3 in sequence.

If DMA is being used to automate the input/output transfers (DMA_HS is 1), then the DMA channel that is associated with DMA Trigger 0 event must be configured to perform four 32-bit writes to the DATA_IN register.

30.1.2.3 Reading Output Data

Output data is read from the engine either via the DATA0/1/2/3 registers or via the DATA_OUT register. If DMA is not being used to automate the input/output transfers (DMA_HS is 0), then CPU software can read out the 128-bit data output by reading 32-bit data from each of DATA0, DATA1, DATA2 and DATA3 registers in sequence.

If DMA is being used to automate the input/output transfers (DMA_HS is 1), then the DMA channel that is associated with DMA Trigger 1 event will need to be configured to perform 4 32-bit reads from the DATA_OUT register.

30.1.2.4 Operation Descriptions

Both single block and block cipher mode operations are configured by writing the appropriate context to the AESADV registers. Independently, the input and output data can be transferred via CPU software or via DMA.

Block cipher modes are used to implement the ECB, CBC, OFB, and CFB block cipher modes using AES as the underlying block cipher. These modes work together with the DMA using the DMA triggers to support easy and fast encryption or decryption of more than 128 bits.

30.1.2.4.1 Single Block Operation

A single 128-bit block of data can be encrypted or decrypted by first configuring the context registers and then initiating input data transfer. The following pseudo code describes the actions that are typically executed by CPU software for a basic encryption operation with 128-bit key.

```
wait AES_CTRL.CNTXT_RDY == '1' // wait for CNTXT_RDY to become 1
wait AES_CTRL.INPUT_RDY == '10' // wait for INPUT_RDY to become 1
write AES_KEY0 // first 32 bits of key
write AES_KEY1 // next 32 bits of key
write AES_KEY2 // next 32 bits of key
write AES_KEY3 // final 32 bits of key
write AES_CTRL.SAVE_CNTXT == '0' // clear AES save context
write AES_CTRL.KEYSIZE == '01' // Write AES 128 bit mode
write AES_CTRL.DIR == '1'      //Write AES encrypt mode
write AES_DATA0 // write 32-bit data LSW to supply 128-bit block
write AES_DATA1 // write 32-bit data (LSW + 1) to supply 128-bit block
write AES_DATA2 // write 32-bit data (LSW + 2) to supply 128-bit block
write AES_DATA3 // write 32-bit data MSW to supply 128-bit block
wait AES_CTRL.OUTPUT_RDY == '1' // wait for OUTPUT_RDY to become 1
read AES_DATA0 // read 32-bit data LSW to read out 128-bit encrypted data
read AES_DATA1 // read 32-bit data (LSW + 1) to read out 128-bit encrypted data
read AES_DATA2 // read 32-bit data (LSW + 2) to read out 128-bit encrypted data
read AES_DATA3 // read 32-bit data MSW to read out 128-bit encrypted data
```

30.1.2.4.2 Electronic Codebook (ECB) Mode

The electronic codebook (ECB) cipher is the simplest block cipher mode. The plaintext data is divided into 128-bit blocks, and each block is encrypted and decrypted independently from any other block. The ECB cipher is shown in Figure 30-2. Note that each 128-bit data block can be encrypted or decrypted individually without any knowledge of the other blocks of plaintext or ciphertext.

While ECB is simple to understand and implement, it has a key disadvantage: the same 128-bit plaintext block is always encrypted into the same ciphertext block, allowing patterns in the ciphertext to be detected.

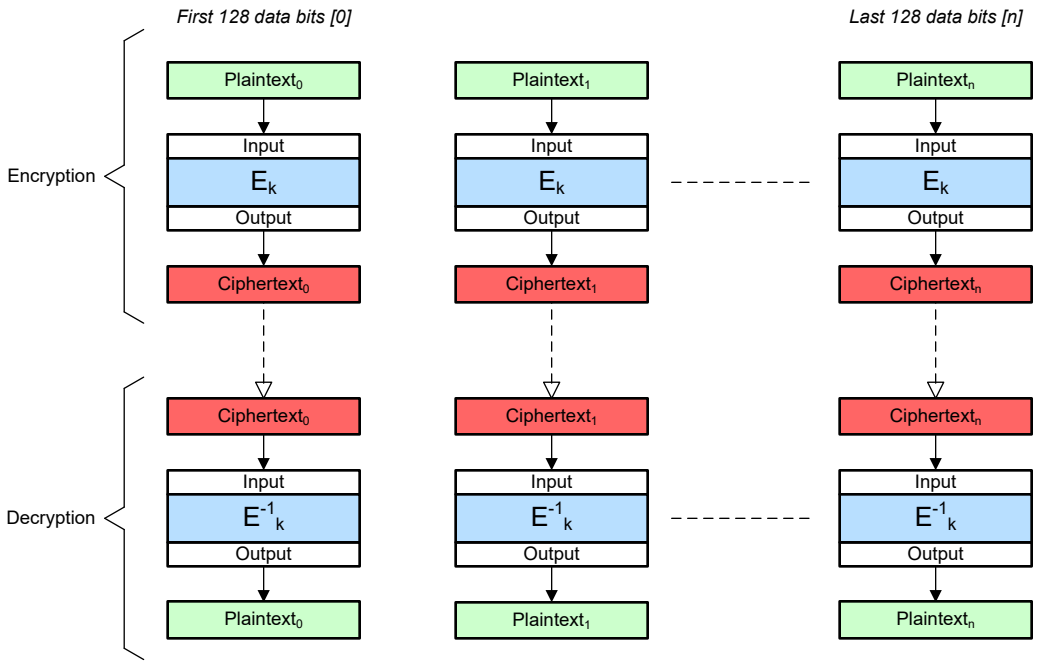


Figure 30-2. ECB Cipher

The AESADV accelerator supports automated encryption and decryption of more than 128 bits of data in ECB block cipher mode either through interrupts or through the use of DMA. Software interrupt-based multi-block handling uses the AESADV interrupt condition reported in the CPU_INT.IIDX.STAT field. If this field reads 0x2 (INPUTRDY), then the next block of input can be written (as 4 32-bit writes to DATA0/1/2/3). If this field reads 0x1 (OUTPUTRDY), then the output block can be read (as 4 32-bit reads from DATA0/1/2/3).

In DMA mode, ECB uses two DMA channels. Channel bound to DMA_TRIG0 is the input channel. Channel bound to DMA_TRIG1 is the output channel. The channels need to be configured to perform one 32-bit read/write per trigger.

30.1.2.4.2.1 ECB Encryption

ECB mode encryption of *N* blocks of plaintext into *N* blocks of ciphertext without CPU interaction is achieved through the use of 2 DMA channels (referred to as DMA_A and DMA_B). To implement ECB encryption follow these steps:

1. Configure Output DMA channel for saving ciphertext:
   a. Set DMA channel trigger selection to AES Trig1
   b. Set DMA channel source address to DATA_OUT
   c. Set DMA channel destination address to location where ciphertext is to be stored (for example, SRAM)
   d. Set DMA channel transfer size to *N*∗4
   e. Set DMA channel mode to single transfer mode
   f. In the AESADV event registers, unmask Trig1 in the IMASK register of DMA_TRIG1
2. Configure Input DMA channel for loading plaintext:
   a. Set DMA channel trigger selection to AES Trig0
   b. Set DMA channel source address to location where plaintext is stored (for example, SRAM)
   c. Set DMA channel destination address to DATA_IN
   d. Set DMA channel transfer size to *N*∗4
   e. Set DMA channel mode to single transfer mode
   f. In the AES event registers, unmask Trig0 in the IMASK register of DMA_TRIG0
3. Configure and enable the DMA interrupt for the Output DMA channel in the DMA controller
4. Configure DMA_HS for DMA based handshake: set DMA_HS[DMA_DATA_ACK] = 1

ADVANCE INFORMATION

5. Configure the CTRL register for block cipher encryption mode for ECB:
   a. Set key size in CTRL[KEY_SIZ]
   b. Select Direction for Encryption by CTRL[DIR] = 1
   c. Select ECB mode by setting rest of the bits in CTRL to 0
6. Load key as described in Section 30.1.2.1 .
7. Start encryption by writing the number of bytes *N×4* to AES C_LENGTH_0 and C_LENGTH_1 registers
8. Wait for the DMA channel interrupt which indicates completion of the entire operation. The ciphertext output will be stored in the location configured in step 1c.

30.1.2.4.2.2 ECB Decryption

ECB Mode Decryption configuration is nearly identical to ECB Mode Encryption. The only difference is that the direction bit (CTRL.DIR) has to be set to 0.

30.1.2.4.3 Cipher Block Chaining (CBC) Mode

The cipher block chaining (CBC) cipher mode builds upon the ECB cipher mode to make the ciphertext output for each block dependent not only on the plaintext and the cipher key *k*, but also upon the ciphertext of the previous block. The CBC cipher is shown in Figure 30-3. Like ECB mode, the plaintext data is divided into 128-bit blocks. Unlike ECB mode, in CBC mode each new plaintext block is bit-wise XORed with the previous ciphertext block to create the input to the AES block cipher.

In CBC mode, an unpredictable initialization vector (IV) must be provided. The initialization vector is XORed with the first plaintext block, as there is no "previous" ciphertext block to XOR the first plaintext block with.

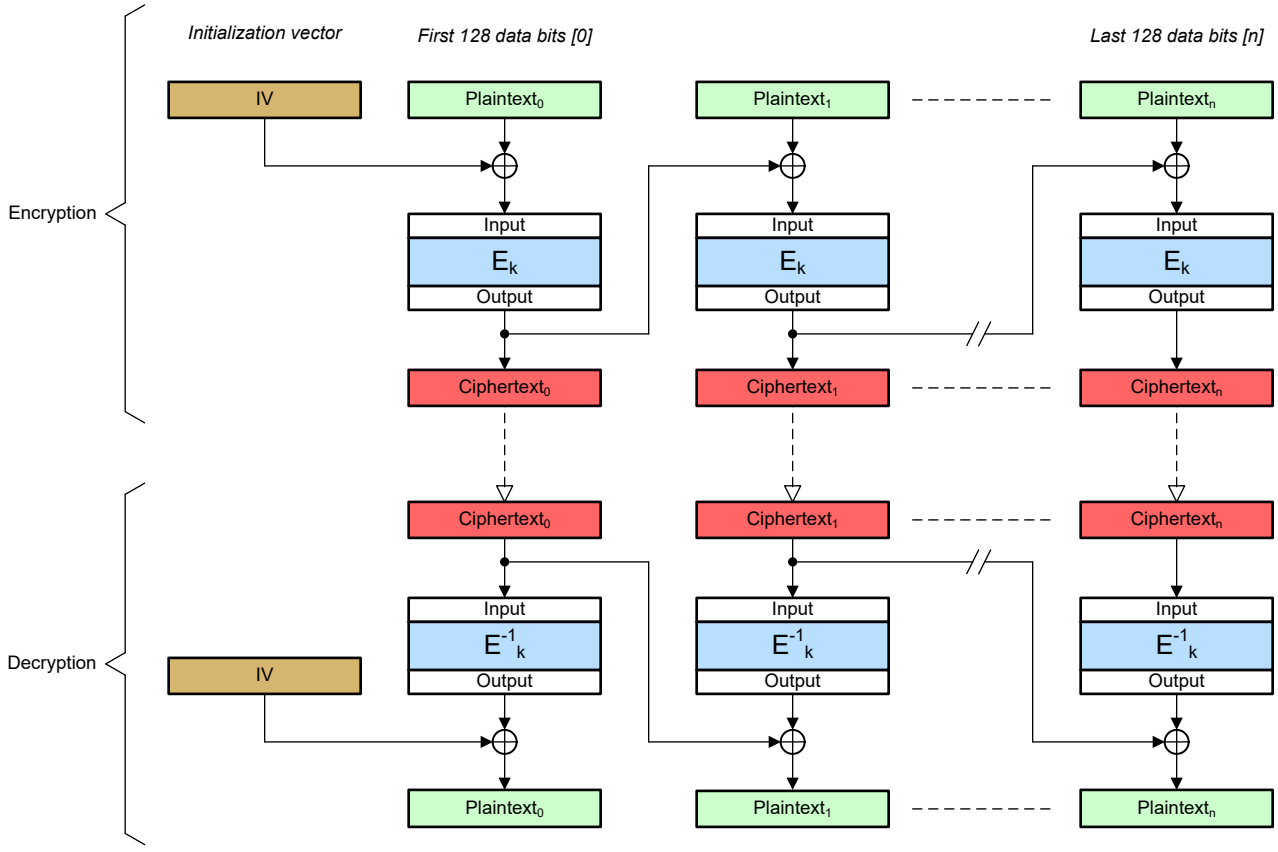


Figure 30-3. CBC Cipher

The AESADV accelerator supports automated CBC mode operation of more than 128 bits of data either through interrupts or through the use of DMA. Software interrupt-based multi-block handling uses the AES interrupt

condition reported in the CPU_INT.IIDX.STAT field. If this field reads 0x2 (INPUTRDY), then the next block of input can be written (as 4 32-bit writes). If this field reads 0x1 (OUTPUTRDY), then the output block can be read (as 4 32-bit reads).

In DMA mode, CBC utilizes two DMA channels. Channel bound to DMA_TRIG0 is the input channel. Channel bound to DMA_TRIG1 is the output channel. The channels need to be configured to perform one 32-bit read/write per trigger.

30.1.2.4.3.1 CBC Encryption

CBC mode encryption of *N* blocks of plaintext into *N* blocks of ciphertext without CPU interaction is achieved through the use of 2 DMA channels. To implement CBC encryption follow these steps:

1. Configure Output DMA channel for saving ciphertext:
   a. Set DMA channel trigger selection to AES Trig1
   b. Set DMA channel source address to DATA_OUT
   c. Set DMA channel destination address to location where ciphertext is to be stored (for example, SRAM)
   d. Set DMA channel transfer size to *N*∗4
   e. Set DMA channel mode to single transfer mode
   f. In the AES event registers, unmask Trig1 in the IMASK register of DMA_TRIG1
2. Configure Input DMA channel for loading plaintext:
   a. Set DMA channel trigger selection to AES Trig0
   b. Set DMA channel source address to location where plaintext is stored (for example, SRAM)
   c. Set DMA channel destination address to DATA_IN
   d. Set DMA channel transfer size to *N*∗4
   e. Set DMA channel mode to single transfer mode
   f. In the AES event registers, unmask Trig0 in the IMASK register of DMA_TRIG0
3. Configure and enable the DMA interrupt for the Output DMA channel in the DMA controller
4. Configure DMA_HS for DMA based handshake: set DMA_HS[DMA_DATA_ACK] = 1
5. Load key as described in Section 30.1.2.1
6. Load Initialization vector (IV) by writing to IV0, IV1, IV2 and IV3 registers
7. Configure the CTRL register for block cipher encryption mode for CBC
   a. Select key size via CTRL[KEY_SIZ]
   b. Select Direction for Encryption by CTRL[DIR] = 1
   c. Select CBC mode by setting CTRL[CBC]=1
8. Start encryption by writing the number of bytes N×4 to AES C_LENGTH_0 and C_LENGTH_1 registers
9. Wait for the DMA channel interrupt that indicates completion of the entire operation. The ciphertext output is stored in the location configured in step 1c.

30.1.2.4.3.2 CBC Decryption

CBC-mode decryption is nearly identical to CBC-mode encryption setup, with the exception of configuring CTRL[DIR] = 0 for decryption.

30.1.2.4.4 Output Feedback (OFB) Mode

The output feedback mode leverages an initialization vector (IV) to generate a keystream by repeatedly encrypting the IV with the cipher key. The output ciphertext is obtained by XORing plaintext with the encrypted and re-encrypted versions of the initialization vector. The OFB cipher is shown in Figure 30-4.

In OFB mode, the initialization vector must be a nonce (number used once). To prevent loss of confidentiality, each IV must only be used one time with a given key, and any value passed into the cipher E_k for a given key k must not be used as an initialization vector with the same key k.

Note

As OFB is a stream cipher, the AES block function is used in the forward (encryption) mode when performing OFB encryption or OFB decryption. When decrypting, the keystream is simply regenerated again to be XORed with the ciphertext, giving back the plaintext.

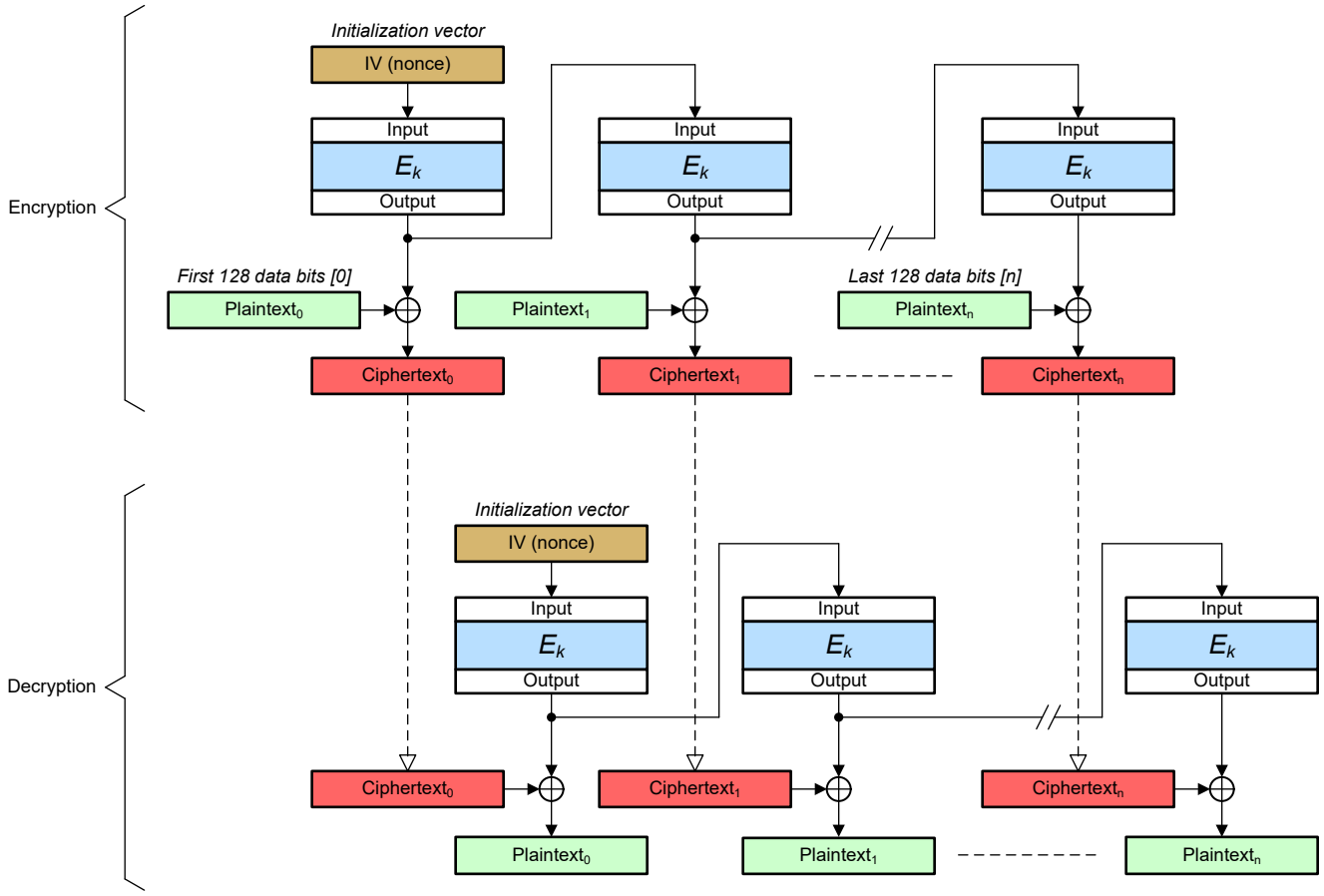


Figure 30-4. OFB Cipher

The AES accelerator supports automated OFB mode operation of more than 128 bits of data either through interrupts or through the use of DMA. Software interrupt-based multi-block handling uses the AES interrupt condition reported in the CPU_INT.IIDX.STAT field. If this field reads 0x2 (INPUTRDY), then the next block of input can be written (as 4 32-bit writes). If this field reads 0x1 (OUTPUTRDY), then the output block can be read (as 4 32-bit reads).

In DMA mode, OFB utilizes two DMA channels. Channel bound to DMA_TRIG0 is the input channel. Channel bound to DMA_TRIG1 is the output channel. The channels need to be configured to perform one 32-bit read/ write per trigger.

30.1.2.4.4.1 OFB Encryption

OFB mode encryption of *N* blocks of plaintext into *N* blocks of ciphertext without CPU interaction is achieved through the use of 2 DMA channels. To implement OFB encryption follow these steps:

1. Configure Output DMA channel for saving ciphertext:
   a. Set DMA channel trigger selection to AES Trig1
   b. Set DMA channel source address to DATA_OUT
   c. Set DMA channel destination address to location where ciphertext is to be stored (for example, SRAM)
   d. Set DMA channel transfer size to *N**4

e. Set DMA channel mode to single transfer mode
f. In the AES event registers, unmask Trig1 in the IMASK register of DMA_TRIG_DATAOUT

2. Configure Input DMA channel for loading plaintext:
   a. Set DMA channel trigger selection to AES Trig0
   b. Set DMA channel source address to location where plaintext is stored (for example, SRAM)
   c. Set DMA channel destination address to DATA_IN
   d. Set DMA channel transfer size to *N**4
   e. Set DMA channel mode to single transfer mode
   f. In the AES event registers, unmask Trig0 in the IMASK register of DMA_TRIG0
3. Configure and enable the DMA interrupt for the Output DMA channel in the DMA controller
4. Configure DMA_HS for DMA based handshake: set DMA_HS[DMA_DATA_ACK] = 1
5. Load key as described in Section 30.1.2.1
6. Load Initialization vector (IV) by writing to IV0, IV1, IV2 and IV3 registers
7. Configure the CTRL register for block cipher encryption mode for OFB
   a. Select key size via CTRL[KEY_SIZ]
   b. Select Direction for Encryption by CTRL[DIR] = 1
   c. Select OFB mode by setting CTRL[OFB_GCM_CCM_CONT]=1
8. Start encryption by writing number of bytes N×4 to AES C_LENGTH_0 and C_LENGTH_1 registers
9. Wait for the DMA channel interrupt which indicates completion of the entire operation. The ciphertext output will be stored in the location configured in step 1c.

Note

OFB_GCM_CCM_CONT bit has dual use. When CCM/GCM modes are not selected, then this bit serves to select OFB mode.

30.1.2.4.4.2 OFB Decryption

OFB-mode decryption is nearly identical to OFB-mode encryption setup, with the exception of configuring CTRL[DIR] = 0 for decryption.

30.1.2.4.5 Cipher Feedback (CFB) Mode

The cipher feedback (CFB) mode is similar to the output feedback (OFB) mode, with the key difference being that the input block to the block cipher *E* used to generate the key stream is taken from the previous ciphertext block (after being XORed with the plaintext), vs. taken before being XORed with the plaintext (as in the case of OFB). As a result, the keystream is dependent upon the plaintext, which is not true of OFB. The CFB cipher is shown in Figure 30-5.

Like OFB, CFB requires an initialization vector (IV). In CFB mode, the initialization vector (IV) must be unpredictable.

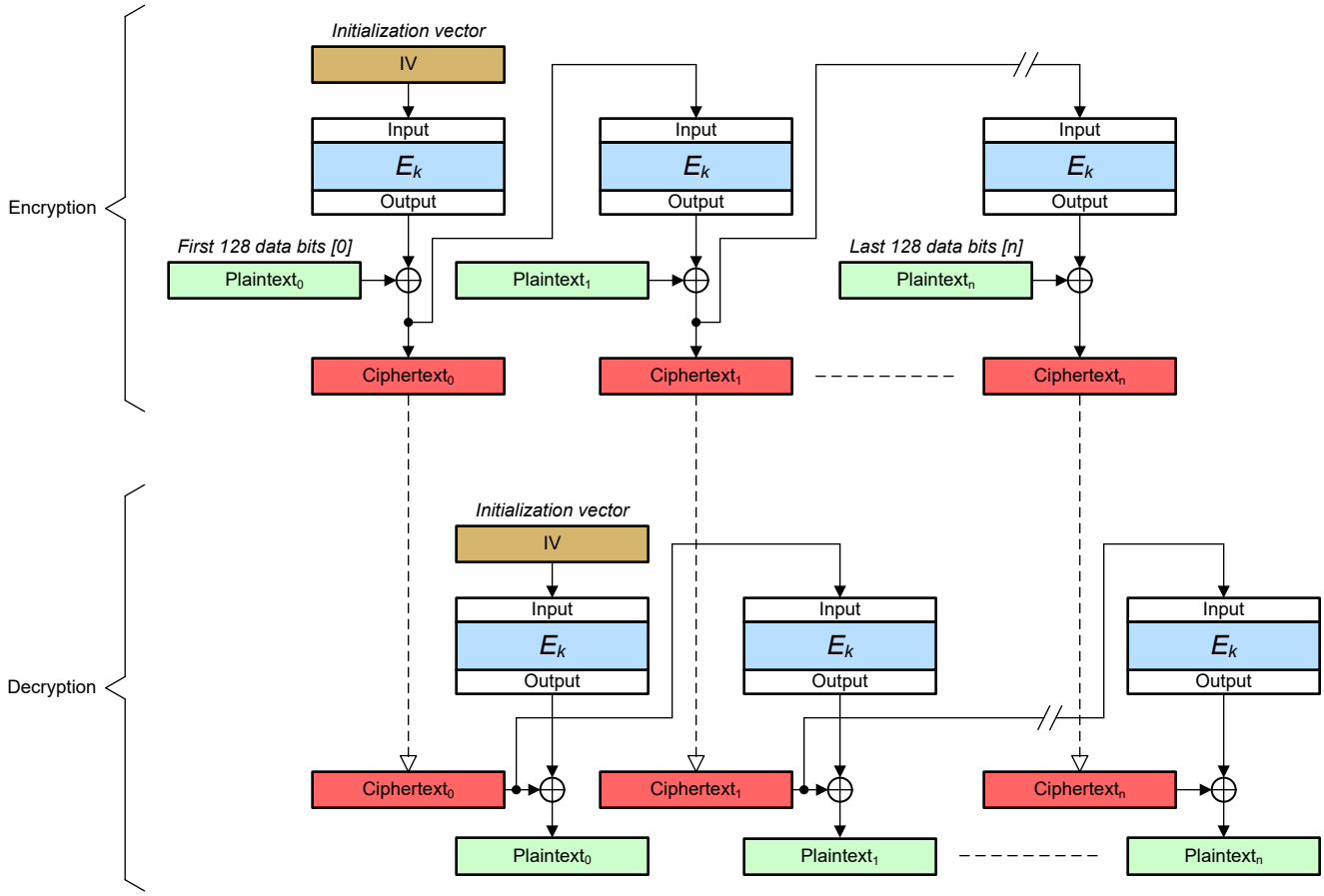


Figure 30-5. CFB Cipher

The AESADV accelerator supports automated CFB mode operation of more than 128 bits of data either through interrupts or through the use of DMA. Software interrupt-based multi-block handling uses the AES interrupt condition reported in the CPU_INT.IIDX.STAT field. If this field reads 0x2 (INPUTRDY), then the next block of input can be written (as 4 32-bit writes). If this field reads 0x1 (OUTPUTRDY), then the output block can be read (as 4 32-bit reads).

In DMA mode, CFB utilizes two DMA channels. Channel bound to DMA_TRIG_DATAIN is the input channel. Channel bound to DMA_TRIG_DATAOUT is the output channel. The channels need to be configured to perform one 32-bit read/write per trigger.

30.1.2.4.5.1 CFB Encryption

CFB mode encryption of *N* blocks of plaintext into *N* blocks of ciphertext without CPU interaction is achieved through the use of 2 DMA channels. To implement CFB encryption follow these steps:

1. Configure Output DMA channel for saving ciphertext:
   a. Set DMA channel trigger selection to AES Trig1
   b. Set DMA channel source address to DATA_OUT
   c. Set DMA channel destination address to location where ciphertext is to be stored (for example, SRAM)
   d. Set DMA channel transfer size to *N*∗4
   e. Set DMA channel mode to single transfer mode
   f. In the AES event registers, unmask Trig1 in the IMASK register of DMA_TRIG_DATAOUT
2. Configure Input DMA channel for loading plaintext:
   a. Set DMA channel trigger selection to AES Trig0
   b. Set DMA channel source address to location where plaintext is stored (for example, SRAM)
   c. Set DMA channel destination address to DATA_IN

   d. Set DMA channel transfer size to *N**4
   e. Set DMA channel mode to single transfer mode
   f. In the AES event registers, unmask Trig0 in the IMASK register of DMA_TRIG0
3. Configure and enable the DMA interrupt for the Output DMA channel in the DMA controller
4. Configure DMA_HS for DMA based handshake: set DMA_HS[DMA_DATA_ACK] = 1
5. Load key as described in Section 30.1.2.1
6. Load Initialization vector (IV) by writing to IV0, IV1, IV2 and IV3 registers
7. Configure the CTRL register for block cipher encryption mode for CFB
   a. Select key size via CTRL[KEY_SIZ]
   b. Select Direction for Encryption by CTRL[DIR] = 1
   c. Select CFB mode by setting CTRL[CFB]=1
   d. Select feedback width by setting CTRL[CTR_WIDTH]
      i. 00b - CFB-128
      ii. 01b - CFB-1
      iii. 10b - CFB-8
8. Start encryption by writing number of bytes N×4 to AES C_LENGTH_0 and C_LENGTH_1 registers
9. Wait for the DMA channel interrupt that indicates completion of the entire operation. The ciphertext output is stored in the location configured in step 1c.

30.1.2.4.5.2 CFB Decryption

CFB-mode decryption is nearly identical to CFB-mode encryption setup, with the exception of configuring CTRL[DIR] = 0 for decryption.

30.1.2.4.6 Counter (CTR) Mode

The counter mode leverages a nonce (number used once) and a counting integer to generate a keystream by appending the counter to the nonce and encrypting the combined nonce || counter value with the cipher key.

The nonce must only be used once with a given key *k*. The counter value can start from any value and is incremented for each 128-bit block of data.

The keystream is derived be encrypting the nonce || counter value for each 128-bit data block with the cipher key *k*. The output ciphertext is then obtained by XORing the plaintext with the encrypted nonce || counter value for each data block. The CTR cipher is shown in Figure 30-6.

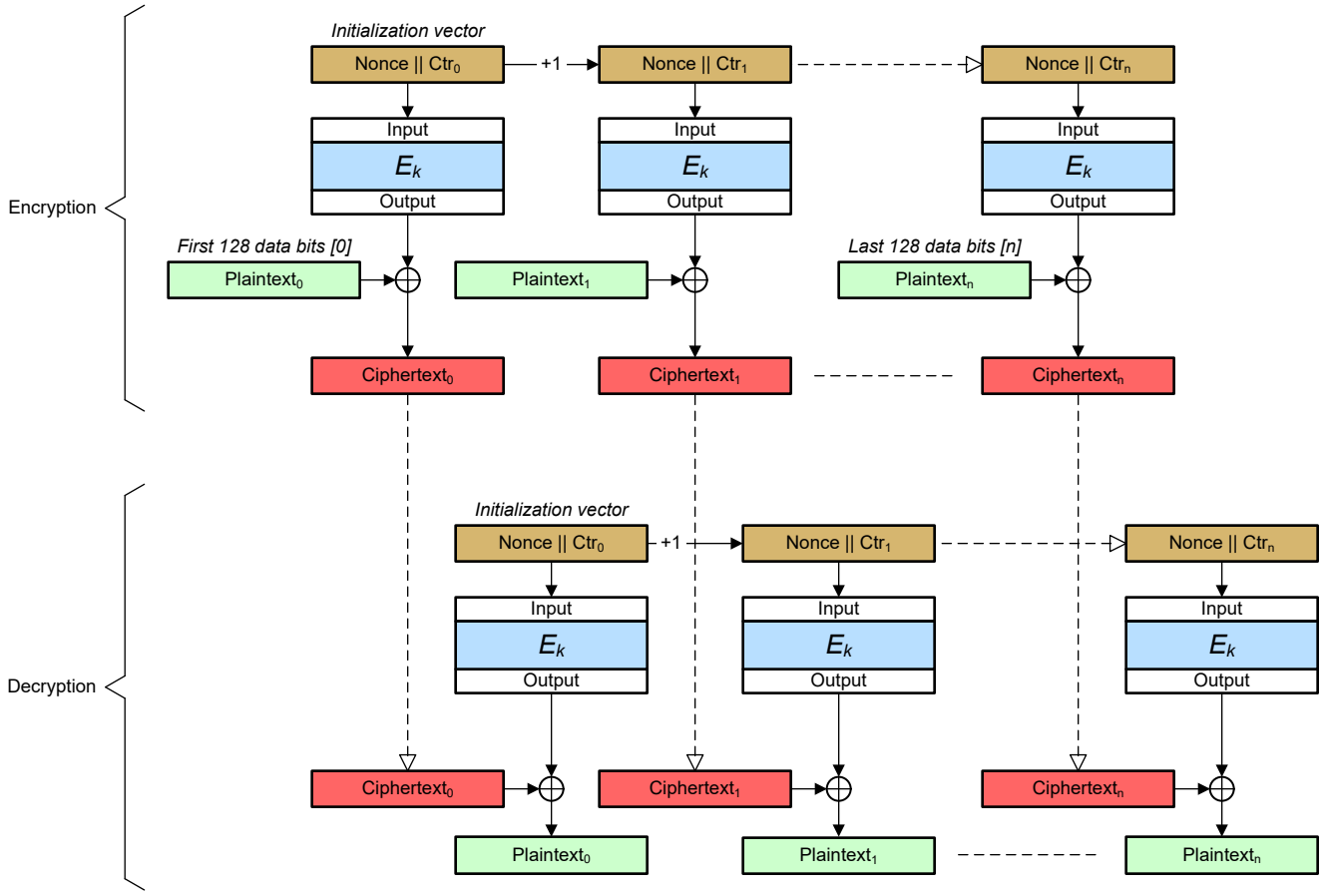


Figure 30-6. CTR Cipher

The AES accelerator implements logic and storage for incrementing and storing nonce || counter from one block to the next.

30.1.2.4.6.1 CTR Encryption

CTR mode encryption of *N* blocks of plaintext into *N* blocks of ciphertext without CPU interaction is achieved through the use of 2 DMA channels. To implement CTR encryption follow these steps:

1. Configure Output DMA channel for saving ciphertext:
   a. Set DMA channel trigger selection to AES Trig1
   b. Set DMA channel source address to DATA_OUT
   c. Set DMA channel destination address to location where ciphertext is to be stored (for example, SRAM)
   d. Set DMA channel transfer size to *N*∗4
   e. Set DMA channel mode to single transfer mode
   f. In the AES event registers, unmask Trig1 in the IMASK register of DMA_TRIG_DATAOUT
2. Configure Input DMA channel for loading plaintext:
   a. Set DMA channel trigger selection to AES Trig0
   b. Set DMA channel source address to location where plaintext is stored (for example, SRAM)
   c. Set DMA channel destination address to DATA_IN
   d. Set DMA channel transfer size to *N*∗4
   e. Set DMA channel mode to single transfer mode
   f. In the AES event registers, unmask Trig0 in the IMASK register of DMA_TRIG0
3. Configure and enable the DMA interrupt for the Output DMA channel in the DMA controller
4. Configure DMA_HS for DMA based handshake: set DMA_HS[DMA_DATA_ACK] = 1
5. Load key as described in Section 30.1.2.1

6. Load Initialization vector (IV) by writing to IV0, IV1, IV2 and IV3 registers
7. Configure the CTRL register for block cipher encryption mode for CTR
   a. Select key size via CTRL[KEY_SIZ]
   b. Select Direction for Encryption by CTRL[DIR] = 1
   c. Select CTR mode by setting CTRL[CTR]=1
   d. Select CTR width by setting CTRL[CTR_WIDTH]
      i. 00b -CTR32
      ii. 01b -CTR64
      iii. 10b -CTR96
      iv. 11b -CTR128

8. Start encryption by writing number of bytes N×4 to AES C_LENGTH_0 and C_LENGTH_1 registers
9. Wait for the DMA channel interrupt which indicates completion of the entire operation. The ciphertext output will be stored in the location configured in step 1c.

30.1.2.4.6.2 CTR Decryption

CTR-mode decryption is nearly identical to CTR-mode encryption setup, with the exception of configuring CTRL[DIR] = 0 for decryption.

30.1.2.4.7 Galois Counter (GCM) Mode

A GCM protocol operation is a combined operation, consisting of encryption/decryption and authentication. Figure 30-7 illustrates an overview of the GCM operation.

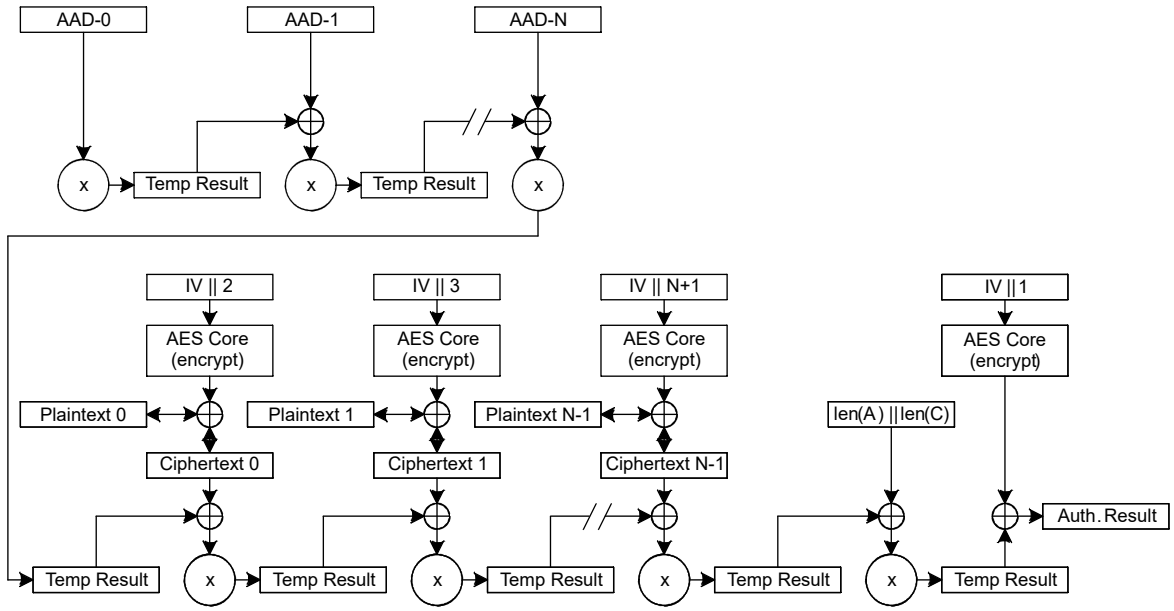


Figure 30-7. GCM Protocol Operation

A part of the input data stream can be authenticated only, while normally most of the input data is encrypted/ decrypted and authenticated. The authentication only data always needs to be in front of the data that requires encryption. Within GCM, the authentication only data is called the AAD (Additional Authentication Data). The AAD is fetched independently of the other data.

The intermediate (temp) result data is used as input for the remaining authentication operation. Since the authentication operation does not require the encryption core but only the polynomial multiplication, both encryption/decryption and authentication are performed in parallel. After encryption of the last data block, an additional polynomial multiplication and encryption are required to respectively authenticate a 128-bit length vector and finally encrypt the authentication result.

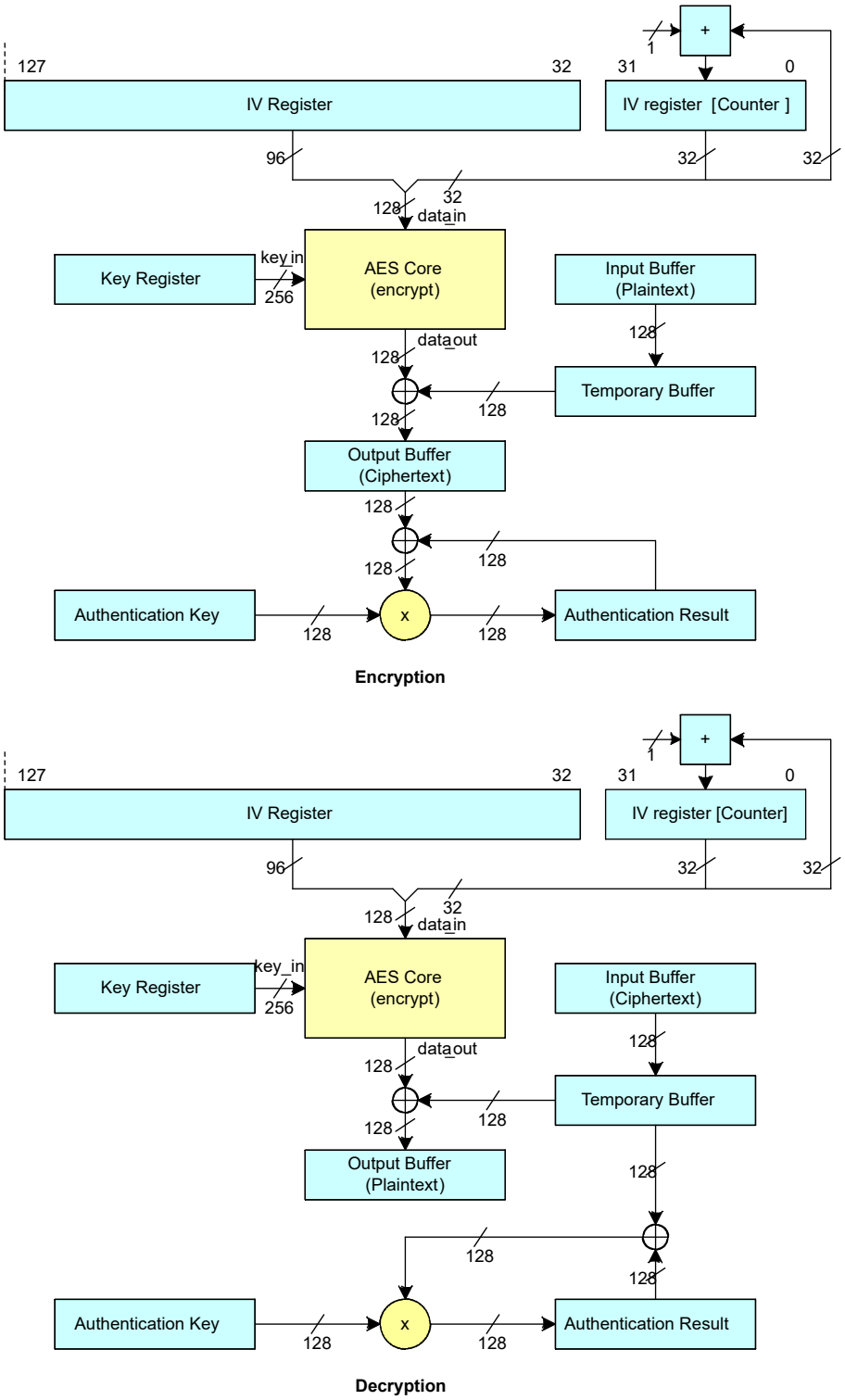


Figure 30-8. GCM Operation on a Block

GMAC operations, as specified in [NIST-SP800-38D], are also supported via the GCM operation. GMAC is a special use of GCM where no crypto data is processed and only AAD data is provided to produce a MAC of the input data. The crypto data length is set to zero for this case.

Figure 30-8 illustrates the operations performed in one round. In one round of a GCM operation for both encryption and decryption, a 32-bit counter is used as IV (as it is for CTR mode). The data is encrypted the same way as CTR mode, by XOR-ing the crypto output with the input. After the encryption/decryption, the ciphertext is XOR-ed with the intermediate authentication result. The XOR-ed result is used as input for the polynomial multiplication to create the next (intermediate) authentication result.

30.1.2.4.7.1 GHASH Operation

For GHASH operations, the engine performs a modular polynomial multiplication in the GF(2128) field. The result is XOR-ed with the encrypted GCM initialization vector (referred to as 'Y0-encrypted'). Y0-encrypted is only relevant when the engine is performing a complete GCM operation, therefore in other modes the value of Y0-encrypted is forced to zero.

Also, for GCM mode only, the GCM 'Hash key' or 'H' input can be pre-calculated and supplied to the engine directly, or it can be calculated by the engine internally, by encrypting the value '0', using the encryption key.

30.1.2.4.7.2 GCM Operating Modes

For GCM three cases need to be distinguished.

- The first one is Autonomous GCM Mode where both H and Y0-encrypted are calculated internally. This mode requires that a 128-bit Y0 be provided to the core via the IV together with the mode.

  Note

  GCM mode bits must be set to 2'b11.

- The second case is the scenario where H is pre-calculated and Y0 still needs to be encrypted by the engine on a per packet basis. This can be useful when multiple packets use the same AES-key. Since H is constant for all packets using the same key, a pre-calculation saves cycles for each packet using that key. H can simply be calculated by performing a basic AES-ECB encryption with the AES-key and a data block containing all zeros or more formally: H=E(K, {}). Once H is calculated, it can be loaded with the control data every time a packet is processed that requires the same AES-key.

  Note

  GCM mode bits must be set to 2'b10.

- In the last case, neither H nor Y0-encrypted are calculated by the core. In this case, Y0-encrypted is forced to zeros, such that the hash result is not encrypted but provided plain via the TAG output registers. This scenario can be selected if a hash (GHASH) only operation needs to be performed. A scenario where this setting can be used is GCM IV-truncation. The GCM specification [GCM] allows an IV that has a length other than 96-bits. In this case, a basic GHASH operation needs to be performed to calculate a 128-bit Y0. For a basic GHASH operation, H needs to be pre-calculated (as explained in the previous paragraph). If H is available, the GHASH operation is similar to that of a general GCM operation with H pre-calculated, only the crypto input data will not be encrypted or decrypted.

  Note

  In the default case a 96-bit IV is combined with a 32-bit counter to create a 128-bit Y0 (meaning: Y0 = {IV||031||1}).

  Note

  GCM mode bits must be set to 2'b01.

30.1.2.4.7.2.1 Autonomous GCM Operation

In this mode, the engine calculates H and Y0-encrypted internally. At the high-level, the programming involves these steps:

- Provide GCM context (key, IV, lengths and mode).
- Provide AAD data (and wait for calculation of H and encryption of Y0).

ADVANCE INFORMATION

- Provide next AAD data.
- Provide last AAD data.
- Provide first crypto data.
- Provide next crypto data.
- Read result data and provide next crypto data.
- ...
- Read result data and provide last crypto data.
- Read result data.
- Read result data.
- Read authentication result (TAG).

To implement GCM encryption over N blocks of plaintext and M blocks of AAD, follow these steps:

1. Configure Output DMA channel for saving ciphertext:
   a. Set DMA channel trigger selection to AES Trig1
   b. Set DMA channel source address to DATA_OUT
   c. Set DMA channel destination address to location where ciphertext is to be stored (for example, SRAM)
   d. Set DMA channel transfer size to *N×4*
   e. Set DMA channel mode to single transfer mode
   f. In the AES event registers, unmask Trig1 in the IMASK register of DMA_TRIG_DATAOUT
2. Configure Input DMA channel for loading AAD and plaintext:
   a. Set DMA channel trigger selection to AES Trig0
   b. Set DMA channel source address to location where plaintext is stored (for example, SRAM)
   c. Set DMA channel destination address to DATA_IN
   d. Set DMA channel transfer size to *(N+M)*4*
   e. Set DMA channel mode to single transfer mode
   f. In the AES event registers, unmask Trig0 in the IMASK register of DMA_TRIG0
3. Configure and enable the DMA interrupt for the Output DMA channel in the DMA controller
4. Configure DMA_HS for DMA based handshake: set DMA_HS[DMA_DATA_ACK] = 1
5. Load Encryption/Decryption key as described in Section 30.1.2.1
6. Load GCMCCM_TAGn (0,1,2,3) registers with 0s.
7. Load Initialization vector (IV) by writing to IV0, IV1, IV2 and IV3 registers
8. Configure the CTRL register for block cipher encryption mode for GCM
   a. Select key size via CTRL[KEY_SIZ]
   b. Select Direction for Encryption by CTRL[DIR] = 1
   c. Select GCM mode by setting CTRL[GCM] = 3
   d. Select CTR mode by setting CTRL[CTR] =

      1
   e. Enabke saving of TAG by setting CTRL[SAVE_CNTXT] = 1
9. Write encryption/decryption byte count N×4 to AES C_LENGTH_0 and C_LENGTH_1 registers
10. Write authentication data (AAD) byte count M*4 to AES AAD_LENGTH register
11. Wait for the DMA output channel interrupt to indicate completion of the entire operation. The output is stored started at the location configured in step 1c.
12. Read out the final TAG from the TAG0, TAG1, TAG2, TAG3 registers

Note

The AAD and cryptographic data can end misaligned. The CPU must pad both to a 128-bit boundary with zeroes. More formally, the AAD and crypto data padding must satisfy the bit string: 0n, with 0 <= n <= 127. This means that the AAD must be provided as separate blocks to the engine, such that the cryptographic data starts 128-bit aligned. If the AAD/cryptographic data stream is 128-bit aligned no padding is required. Because the engine only supports bytes, n must be such that (n MOD 8) = 0. Further, because a single DMA channel supplies both AAD and plaintext, the entire data must be organized contiguously in memory with the first M blocks of AAD followed by N blocks of plaintext. This memory contiguity restriction is not applicable when CPU software directly provides inputs via interrupt handling.

Do not load both length values with zeroes. If a data stream is done and the next data stream uses the same key and control, only the IV and length values need be re-loaded.

30.1.2.4.7.2.1.1 GMAC

GMAC is a special case of GCM wherein there is no payload (N=0). AES C_LENGTH_0 and C_LENGTH_1 registers are set to 0 in this case.

30.1.2.4.7.2.2 GCM With Pre-Calculations

Pre-calculation of H:

1. Provide AES-ECB context (key and mode)
2. Provide zeros as data
3. Read result data (H)

IV truncation/pre-calculation (GHASH only operation):

- Provide GHASH context (H, lengths and mode)

Note

GHASH only: Y0-encrypted forced to zero, H loaded, and no crypto mode selected

1. Provide IV-data-block
2. Provide next IV-data-block
3. Read dummy result and provide next IV-data-block
4. ...
5. Read dummy result data and provide last IV-data-block
6. Read dummy result data
7. Read dummy result data
8. Read authentication result (TAG... this data is Y0)

GCM operation with pre-calculated H:

1. Provide GCM context (key, Y0 as IV, H, lengths and mode)
2. Provide AAD data (and wait for encryption of Y0)
3. Provide next AAD data
4. Provide last AAD data
5. Provide first crypto data
6. Provide next crypto data
7. Read result data and provide next crypto data
8. ...
9. Read result data and provide last crypto data
10. Read result data
11. Read result data
12. Read authentication result (TAG)

For GCM with pre-calculated H, select GCM mode by setting CTRL[GCM] = 2

30.1.2.4.7.2.3 GCM Operation With Precalculated H- and Y0-Encrypted Forced to Zero

The outline of operations is listed below:

1. Provide GCM context (key, H, lengths and mode)
2. Provide AAD data
3. Provide next AAD data
4. Provide last AAD data
5. Provide first crypto data
6. Provide next crypto data
7. Read result data and provide next crypto data
8. ...
9. Read result data and provide last crypto data
10. Read result data
11. Read result data
12. Read plaintext authentication result

30.1.2.4.8 Counter With Cipher Block Chaining Message Authentication Code (CCM)

The CCM (Counter with CBC-MAC) protocol operation is a combined operation, consisting of encryption/decryption and authentication. Both the authentication and encryption/decryption operations use the crypto core; these are executed sequentially on the AES core. A part of the data stream can require authentication only. The authentication only data always needs to be in front of the data that requires encryption.

Figure 30-9 illustrates the CCM protocol. The authentication starts with the encryption of a pre-defined block B0. This block consists of flags, nonce and message length. The next blocks contain the authentication data length concatenated with the authentication only data. After processing the authentication only data, the encryption/decryption operations are performed, each followed by the related authentication of the plaintext data block (which equals the input in the case of encryption and the output in the case of decryption). The final authentication result needs to be encrypted using the output of the encryption of the IV block A0. This block contains the IV (consisting of flags and nonce) concatenated with the counter, which is zero for A0.

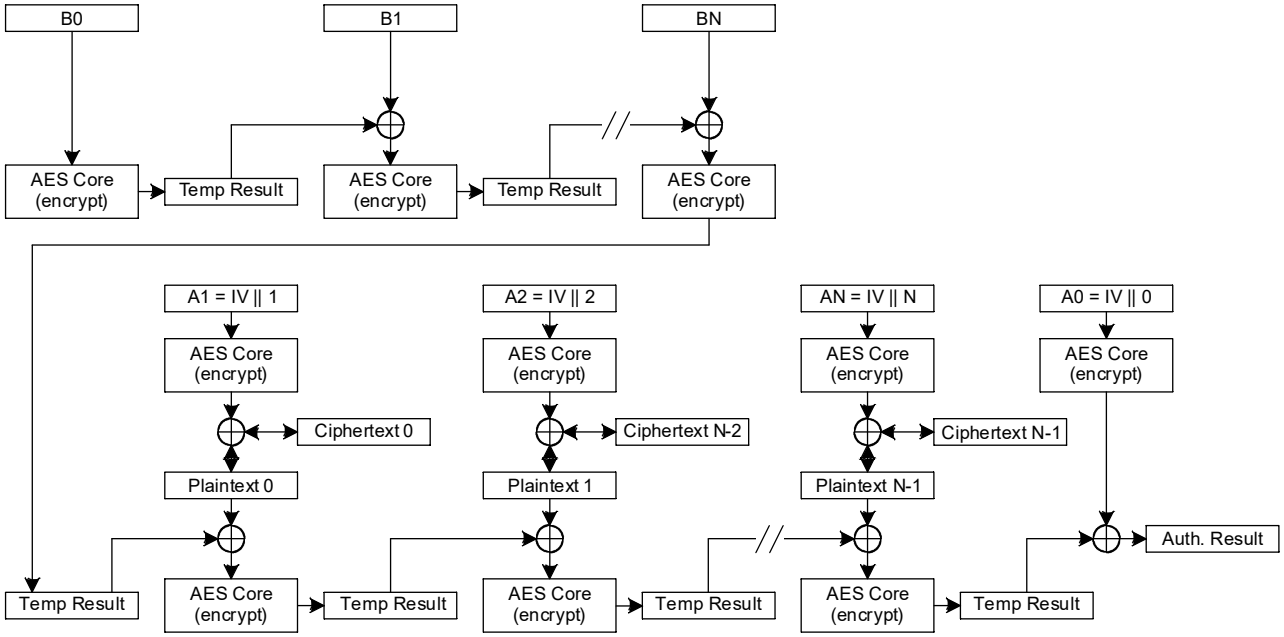


Figure 30-9. CCM Protocol Operation

Figure 30-10 shows one round of a CCM operation for both encryption and decryption. A 32-bit counter is used as IV (as it is for CTR mode). The data is encrypted in the same way as CTR mode, by XOR-ing the crypto core output with the input. Directly after the encrypt-operation, the plaintext is XOR-ed with the intermediate authentication result. The XOR result is used as input for a second encrypt-operation to calculate the next (intermediate) authentication result.

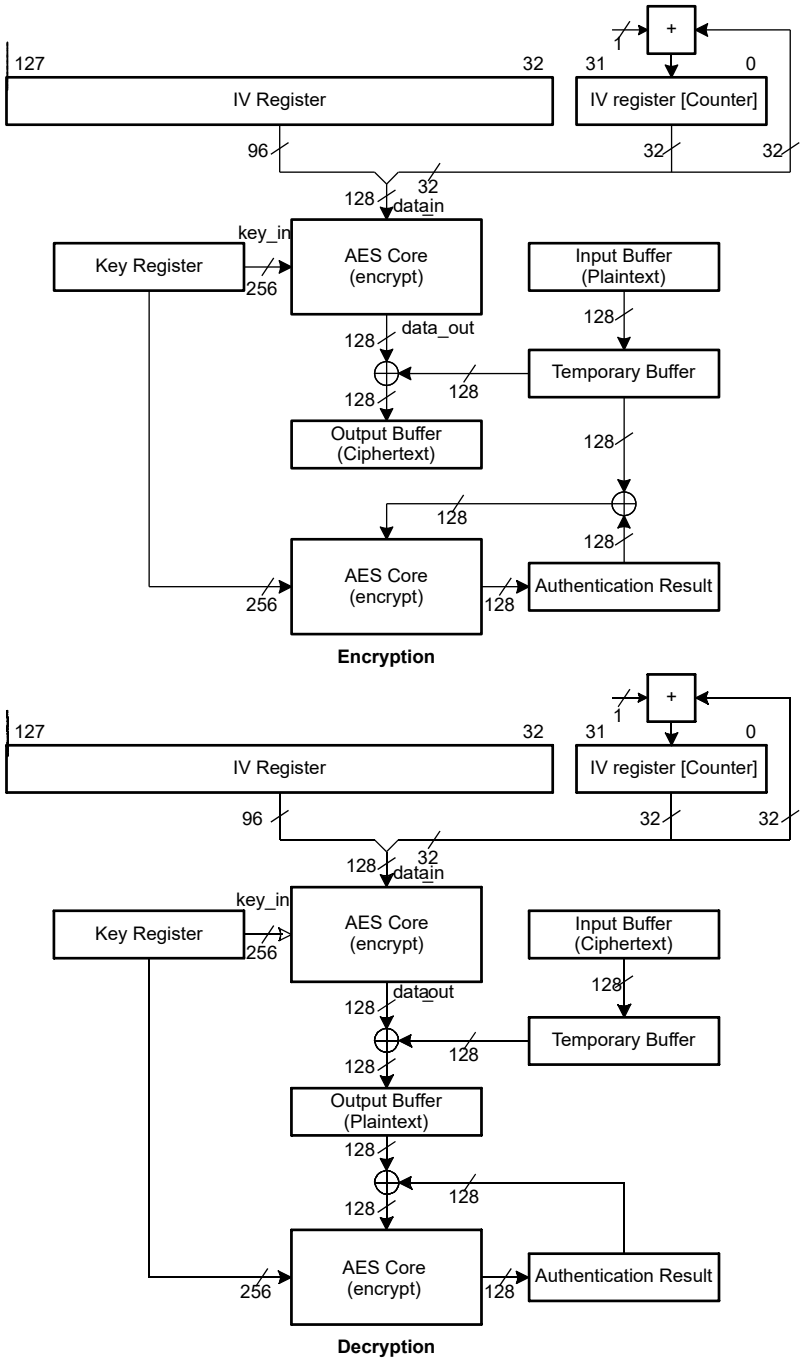


Figure 30-10. CCM Operation

30.1.2.4.8.1 CCM Operation

At the high-level, CCM operation is performed in the following sequence:

- Provide CCM context (key, IV including flags, lengths and mode)
- Provide hash only data
- Provide next hash only data
- Provide last hash only data
- Provide first crypto data
- Provide next crypto data
- Read result data and provide next crypto data
- ...
- Read result data and provide last crypto data
- Read result data
- Read result data
- Read authentication result (TAG)

The read/write of data can be managed via DMA or by CPU software.

To implement CCM encryption over N blocks of plaintext and M blocks of AAD, follow these steps:

1. Configure Output DMA channel for saving ciphertext:
   a. Set DMA channel trigger selection to AES Trig1
   b. Set DMA channel source address to DATA_OUT
   c. Set DMA channel destination address to location where ciphertext is to be stored (for example, SRAM)
   d. Set DMA channel transfer size to *N×4*
   e. Set DMA channel mode to single transfer mode
   f. In the AES event registers, unmask Trig1 in the IMASK register of DMA_TRIG_DATAOUT
2. Configure Input DMA channel for loading AAD and plaintext:
   a. Set DMA channel trigger selection to AES Trig0
   b. Set DMA channel source address to location where plaintext is stored (for example, SRAM)
   c. Set DMA channel destination address to DATA_IN
   d. Set DMA channel transfer size to *(N+M)*4*
   e. Set DMA channel mode to single transfer mode
   f. In the AES event registers, unmask Trig0 in the IMASK register of DMA_TRIG0
3. Configure and enable the DMA interrupt for the Output DMA channel in the DMA controller
4. Configure DMA_HS for DMA based handshake: set DMA_HS[DMA_DATA_ACK] = 1
5. Load Encryption/Decryption key as described in Section 30.1.2.1
6. Load Initialization vector (IV) by writing to IV0, IV1, IV2 and IV3 registers (must contain the flags for the cryptographic operation and the NONCE bytes, for both authentication and encryption)
7. Configure the CTRL register for block cipher encryption mode for CCM
   a. Select key size via CTRL[KEY_SIZ]
   b. Select Direction for Encryption by CTRL[DIR] = 1
   c. Select CCM mode by setting CTRL[CCM] = 1
   d. Select CTR mode by setting CTRL[CTR] =

      1
   e. Enabke saving of TAG by setting CTRL[SAVE_CNTXT] = 1
   f. Configure CTRL[CCML] -- CCM-L can be set to any value, representing a crypto data length field of CCML plus one Bytes.
   g. Configure CTRL[CTR_WIDTH] -- Note: CCM-L sets the actual counter field width in the IV register for CCM operations, in the range from 2 (CCM-L = 1) up to and including 8 (CCM-L = 7) Bytes. The actual counter width chosen with the CTR_WIDTH must be long enough to cover this field. The counter field width must be chosen so that the counter cannot overflow.
   h. Configure CTRL[CCMM] -- CCM-M can be set to any value and has no effect on the actual processing (other than being present in the special ‘B0’ block that is encrypted at the start of the operation). The CPU must select the valid TAG bytes from the 128-bit TAG (which are in the least significant 2 * (CCMM + 1) Bytes).
8. Write encryption/decryption byte count N×4 to AES C_LENGTH_0 and C_LENGTH_1 registers

9. Write authentication data (AAD) byte count M*4 to AES AAD_LENGTH register
10. Wait for the DMA output channel interrupt that indicates completion of the entire operation. The output is stored started at the location configured in step 1c.
11. Read out the final TAG from the TAG0/1/2/3 registers

Note

The AAD and cryptographic data can end misaligned. The CPU must pad both to a 128-bit boundary with zeroes. More formally, the AAD and crypto data padding must satisfy the bit string: 0n, with 0<= n <= 127. This means that the AAD must be provided as separate blocks to the engine, such that the cryptographic data starts 128-bit aligned. If the AAD/cryptographic data stream is 128-bit aligned, no padding is required. Because the engine only supports bytes, n must be such that (n MOD 8) = 0. Further, since a single DMA channel is used to supply both AAD and plaintext, the entire data must be organized contiguously in memory with the first M blocks of AAD followed by N blocks of plaintext. This memory contiguity restriction is not applicable when CPU software directly provides inputs via interrupt handling.

Do not write both length values with zeroes. If a data stream is done and the next data stream uses the same key and control, only the IV and length values need be re-loaded.

30.1.2.5 AES Events

The AES module contains three event publishers and no event subscribers. One event publisher (CPU_INT) manages AES interrupt requests (IRQs) to the CPU subsystem through a static event route. The second, and third event publishers (DMA_TRIG_DATAIN, and DMA_TRIG_DATAOUT) can be used to publish AES events to the DMA through DMA event routes.

The AES events are summarized in Table 30-2.

Table 30-2. AES Events

Event	Type	Source	Destination	Route	Configuration	Functionality
CPU Interrupt Event	Publisher	AES	CPU Subsystem	Static route	CPU_INT registers	Fixed interrupt route from RTC to CPU
DMA Trigger Event 0	Publisher	AES	DMA	DMA route	DMA_TRIG 0 registers	DMA trigger 0: Data Input into engine
DMA Trigger Event 1	Publisher	AES	DMA	DMA route	DMA_TRIG 1 registers	DMA trigger 1: Data Output from engine

In general, the CPU interrupt event is used to communicate completion of an AES operation to the CPU, and the DMA triggers are used together to implement the block cipher modes (ECB, CBC, OFB, CFB, GCM/GMAC, CCM/CMAC) using the DMA together with the AES accelerator.

30.1.2.5.1 CPU Interrupt Event Publisher (CPU_EVENT)

The AESADV module provides four interrupt sources which can be configured to source a CPU interrupt event. The CPU interrupt events from the AES are given inTable 30-3 .

Table 30-3. AES CPU Interrupt Event Conditions (CPU_EVENT)

Index (IIDX)	Name	Description
0	NO_INTR	No interrupt pending.
1	OUTPUTRDY	This indicates that the engine has an output available to be read out. This should not be used if DMA handshake is used (AES_DMA_HS.DMA_DATA_ACK set to 1)

ADVANCE INFORMATION

Table 30-3. AES CPU Interrupt Event Conditions (CPU_EVENT) (continued)

Index (IIDX)	Name	Description
2	INPUTRDY	This indicates that the engine can take new input. This should not be used if DMA handshake is used (AES_DMA_HS.DMA_DATA_ACK set to 1)
3	SAVEDCNTXTRDY	This bit indicates that an AES authentication TAG and/or IV block(s) is/are available for the CPU to retrieve. This bit is only asserted if the 'save_context' bit is set to '1b'. The bit is mutually exclusive with the 'context_ready' bit.
4	CNTXTRDY	This bit indicates that the context data registers can be overwritten, and the CPU is permitted to write new context.

The CPU interrupt event configuration is managed with the event management registers. See *Using Event Registers* for guidance on configuring these registers for CPU interrupts.

30.1.2.5.2 DMA Trigger Event Publisher (DMA_TRIG_DATAIN)

The AESADV module provides a trigger source which can be configured to source DMA trigger 0. The DMA0 trigger events from the AES are given in Table 30-4. When the DMA0 channel is needed by the AES for block cipher operations, the DMA0 trigger should be unmasked in the IMASK register of DMA_TRIG_DATAIN and the DMA should be configured as needed to support the AES operation.

Table 30-4. AES DMA Trigger 0 Event Conditions (DMA_TRIG_DATAIN)

Index (IIDX)	Name	Description
0	NO_INTR	No DMA Trig0 event pending
1	TRIG0	DMA Trigger for Data Input

The DMA trigger 0 event configuration is managed with the DMA_TRIG_DATAIN event management registers. See *Using Event Registers* for guidance on configuring the event registers for DMA triggers.

30.1.2.5.3 DMA Trigger Event Publisher (DMA_TRIG_DATAOUT)

The AES module provides a trigger source which can be configured to source DMA trigger 1. The DMA1 trigger events from the AES are given in Table 30-5. When the DMA1 channel is needed by the AES for block cipher operations, the DMA1 trigger should be unmasked in the IMASK register of DMA_TRIG_DATAOUT and the DMA should be configured as needed to support the AES operation.

Table 30-5. AES DMA Trigger 1 Event Conditions (DMA_TRIG_DATAOUT)

Index (IIDX)	Name	Description
0	NO_INTR	No DMA Trig1 Event Pending
1	TRIG1	DMA Trigger for Data Output

The DMA trigger 1 event configuration is managed with the DMA_TRIG_DATAOUT event management registers. See *Using Event Registers* for guidance on configuring the event registers for DMA triggers.

30.2 AES Registers

This Section describes the AES Registers.

30.2.1 AES Base Address Table

Table 30-6. AES Base Address Table

Bit Field Name		DriverLib Name	Base Address
Instance	Structure		
AesRegs	AES_REGS	AES	0x401B_0000

30.2.2 AES_REGS Registers

Table 30-7 lists the memory-mapped registers for the AES_REGS registers. All register offset addresses not listed in Table 30-7 should be considered as reserved locations and the register contents should not be modified.

Table 30-7. AES_REGS Registers

Offset	Acronym	Register Name	Section
480h	CPU_CONNECT_0	CPU Connect	Go
800h	PWREN	Power enable	Go
804h	RSTCTL	Reset Control	Go
814h	STAT	Status Register	Go
1018h	PDBGCTL	Peripheral Debug Control	Go
1020h	IIDX	Interrupt Index Register	Go
1028h	IMASK	Interrupt mask	Go
1030h	RIS	Raw interrupt status	Go
1038h	MIS	Masked interrupt status	Go
1040h	ISET	Interrupt set	Go
1048h	ICLR	Interrupt clear	Go
1050h	IIDX	Interrupt Index Register	Go
1058h	IMASK	Interrupt mask	Go
1060h	RIS	Raw interrupt status	Go
1068h	MIS	Masked interrupt status	Go
1070h	ISET	Interrupt set	Go
1078h	ICLR	Interrupt clear	Go
1080h	IIDX	Interrupt Index Register	Go
1088h	IMASK	Interrupt mask	Go
1090h	RIS	Raw interrupt status	Go
1098h	MIS	Masked interrupt status	Go
10A0h	ISET	Interrupt set	Go
10A8h	ICLR	Interrupt clear	Go
10E0h	EVT_MODE	Event Mode	Go
1100h	GCMCCM_TAG0	CBC-MAC third key (LSW) / GCM & CCM Intermediate TAG (LSW)	Go
1104h	GCMCCM_TAG1	CBC-MAC third key / GCM & CCM Intermediate TAG	Go
1108h	GCMCCM_TAG2	CBC-MAC third key / GCM & CCM Intermediate TAG	Go
110Ch	GCMCCM_TAG3	CBC-MAC third key (MSW) / GCM & CCM Intermediate TAG (MSW)	Go
1110h	GHASH_H0	CCM & CBC-MAC second key (LSW) / GCM Hash Key input (LSW)	Go
1114h	GHASH_H1	CCM & CBC-MAC second key / GCM Hash Key input	Go
1118h	GHASH_H2	CCM & CBC-MAC second key / GCM Hash Key input	Go
111Ch	GHASH_H3	CCM & CBC-MAC second key (MSW) / GCM Hash Key input (MSW)	Go
1120h	KEY0	KEY (LSW)	Go
1124h	KEY1	KEY	Go
1128h	KEY2	KEY	Go
112Ch	KEY3	KEY	Go
1130h	KEY4	KEY	Go
1134h	KEY5	KEY	Go
1138h	KEY6	KEY	Go

Table 30-7. AES_REGS Registers (continued)

Offset	Acronym	Register Name	Section
113Ch	KEY7	KEY (MSW)	Go
1140h	IV0	IV (LSW)	Go
1144h	IV1	IV	Go
1148h	IV2	IV	Go
114Ch	IV3	IV	Go
1150h	CTRL	Input/Output Buffer Control and Mode selection	Go
1154h	C_LENGTH_0	Crypto data length (LSW)	Go
1158h	C_LENGTH_1	Crypto data length (MSW)	Go
115Ch	AAD_LENGTH	AAD Data Length	Go
1160h	DATA0	Data input (LSW) / Data output (LSW)	Go
1164h	DATA1	Data input / Data output	Go
1168h	DATA2	Data input / Data output	Go
116Ch	DATA3	Data input (LSW) / Data output (MSW)	Go
1170h	TAG0	Hash result (LSW)	Go
1174h	TAG1	Hash result	Go
1178h	TAG2	Hash result	Go
117Ch	TAG3	Hash result (MSW)	Go
1180h	STATUS	Status	Go
1184h	DATA_IN	Data in alias register	Go
1188h	DATA_OUT	Data out alias register	Go
11D0h	FORCE_IN_AV	Data control register for input data	Go
11D4h	CCM_ALN_WRD	AES-CCM AAD alignment data word	Go
11D8h	BLK_CNT0	Internal block counter (LSW)	Go
11DCh	BLK_CNT1	Internal block counter (MSW)	Go
11F4h	DMA_HS	Control register for DMA handshaking	Go

Complex bit access types are encoded to fit into small table cells. Table 30-8 shows the codes that are used for access types in this section.

Table 30-8. AES_REGS Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
Write Type		
W	W	Write
WK	W K	Write Write protected by a key
Reset or Default Value		
-n		Value after reset or the default value

1 CPU_CONNECT_0 Register (Offset = 480h) [Reset = 00000000h]

CPU_CONNECT_0 is shown in Figure 30-11 and described in Table 30-9.

Return to the Summary Table.

Directly connect peripheral publisher port to application processor

Figure 30-11. CPU_CONNECT_0 Register

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

Table 30-10. CPU_CONNECT_0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R/W	0h	
1	RESERVED	R/W	0h	Reserved
0	RESERVED	R/W	0h	

2 PWREN Register (Offset = 800h) [Reset = 00000000h]

PWREN is shown in Figure 30-12 and described in Table 30-10.

Return to the Summary Table.

Register to control the power state

Figure 30-12. PWREN Register

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

Table 30-12. PWREN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	KEY	W	0h	KEY to allow Power State Change 26h = KEY to allow write access to this register
23-1	RESERVED	R/W	0h	
0	ENABLE	R/WK	0h	Enable the power [EXT_GPRCM.GPRCM.PWREN.KEY] must be set to 26h to write to this bit. 0h = Disable Power 1h = Enable Power

ADVANCE INFORMATION

3 RSTCTL Register (Offset = 804h) [Reset = 00000000h]

RSTCTL is shown in Figure 30-13 and described in Table 30-11.

Return to the Summary Table.

Register to control reset assertion and de-assertion

Figure 30-13. RSTCTL Register

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

Table 30-14. RSTCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	KEY	W	0h	Unlock key B1h = KEY to allow write access to this register
23-2	RESERVED	W	0h	
1	RESETSTKYCLR	WK	0h	Clear the RESETSTKY bit in the STAT register [EXT_GPRCM.GPRCM.RSTCTL.KEY] must be set to B1h to write to this bit. 0h = Writing 0 has no effect 1h = Clear reset sticky bit
0	RESETASSERT	WK	0h	Assert reset to the peripheral [EXT_GPRCM.GPRCM.RSTCTL.KEY] must be set to B1h to write to this bit. 0h = Writing 0 has no effect 1h = Assert reset

ADVANCE INFORMATION

4 STAT Register (Offset = 814h) [Reset = 00000000h]

STAT is shown in Figure 30-14 and described in Table 30-12.

Return to the Summary Table.

peripheral enable and reset status

Figure 30-14. STAT Register

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

Table 30-16. STAT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-17	RESERVED	R	0h	
16	RESETSTKY	R	0h	This bit indicates, if the peripheral was reset, since this bit was cleared by RESETSTKYCLR in the RSTCTL register 0h = The peripheral has not been reset since this bit was last cleared by RESETSTKYCLR in the RSTCTL register 1h = The peripheral was reset since the last bit clear
15-0	RESERVED	R	0h	

5 PDBGCTL Register (Offset = 1018h) [Reset = 00000000h]

PDBGCTL is shown in Figure 30-15 and described in Table 30-13.

Return to the Summary Table.

AES can not be halted when the core is halted. In order to halt the AES, the DMA shall be halted.
This achieves the same effect as a halt feature in the AES: when the AES submits the next DMA trigger, if the DMA is halted, then the AES will automatically halt.

Figure 30-15. PDBGCTL Register

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

Table 30-18. PDBGCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	
0	FREE	R	1h	Free run control 1h = The peripheral ignores the state of the Core Halted input

6 IIDX Register (Offset = 1020h) [Reset = 00000000h]

IIDX is shown in Figure 30-16 and described in Table 30-14.

Return to the Summary Table.

This register provides the highest priority enabled interrupt index. Value 0x00 means no event pending. Interrupt 1 is the highest priority, IIDX next highest, 4, 8, ... IIDX^31 is the least priority. That is, the least bit position that is set to 1 denotes the highest priority pending interrupt. The priority order is fixed. However, users can implement their own prioritization schemes using other registers that expose the full set of interrupts that have occurred. On each read, only one interrupt is indicated. On a read, the current interrupt (highest priority) is automatically cleared by the hardware and the corresponding interrupt flag in [RIS] and [MIS] are cleared as well. After a read from the CPU (not from the debug interface), the register is updated with the next highest priority interrupt, if none are pending, then it should display 0x0.

Figure 30-16. IIDX Register

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

Table 30-20. IIDX Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	
7-0	STAT	R	0h	Interrupt index status 00h = No interrupt pending 1h = This indicates that the core has an output available to be read out. This should not be used if DMA handshake is used (AES_DMA_HS.DMA_DATA_ACK set to 1) 2h = This indicates that the engine can take new input. This should not be used if DMA handshake is used (AES_DMA_HS.DMA_DATA_ACK set to 1) 3h = This bit indicates that an AES authentication TAG and/or IV block(s) is/are available for the CPU to retrieve. This bit is only asserted if the 'save_context' bit is set to '1b'. The bit is mutually exclusive with the 'context_ready' bit. 4h = This bit indicates that the context data registers can be overwritten, and the CPU is permitted to write new context.

7 IMASK Register (Offset = 1028h) [Reset = 00000000h]

IMASK is shown in Figure 30-17 and described in Table 30-15.

Return to the Summary Table.

Interrupt Mask. If a bit is set, then corresponding interrupt is un-masked. Un-masking the interrupt causes the raw interrupt to be visible in IIDX, as well as MIS.

Figure 30-17. IMASK Register

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

Table 30-22. IMASK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R/W	0h	
3	CNTXTRDY	R/W	0h	This bit indicates that the context data registers can be overwritten, and the CPU is permitted to write next context. 0h = Clear Interrupt Mask 1h = Set Interrrupt Mask
2	SAVEDCNTXTRDY	R/W	0h	This bit indicates that an AES authentication TAG and/or IV block(s) is/are available for the CPU to retrieve. This bit is only asserted if the 'save_context' bit is set to '1b'. The bit is mutually exclusive with the 'context_ready' bit. 0h = Clear Interrupt Mask 1h = Set Interrrupt Mask
1	INPUTRDY	R/W	0h	This indicates that the engine can take new input. This should not be used if DMA handshake is used (AES_DMA_HS.DMA_DATA_ACK set to 1) 0h = Clear Interrupt Mask 1h = Set Interrrupt Mask
0	OUTPUTRDY	R/W	0h	This indicates that the core has an output available to be read out. This should not be used if DMA handshake is used (AES_DMA_HS.DMA_DATA_ACK set to 1) 0h = Clear Interrupt Mask 1h = Set Interrrupt Mask

8 RIS Register (Offset = 1030h) [Reset = 00000000h]

RIS is shown in Figure 30-18 and described in Table 30-16.

Return to the Summary Table.

Raw interrupt status. Reflects all pending interrupts, regardless of masking. The RIS register allows the user to implement a poll scheme. A flag set in this register can be cleared by writing 1 to the ICLR register bit even if the corresponding IMASK bit is not enabled.

Figure 30-18. RIS Register

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

Table 30-24. RIS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R	0h	
3	CNTXTRDY	R	0h	This bit indicates that the context data registers can be overwritten, and the CPU is permitted to write next context. 0h = Clear Interrupt Mask 1h = Set Interrrupt Mask
2	SAVEDCNTXTRDY	R	0h	This bit indicates that an AES authentication TAG and/or IV block(s) is/are available for the CPU to retrieve. This bit is only asserted if the 'save_context' bit is set to '1b'. The bit is mutually exclusive with the 'context_ready' bit. 0h = Clear Interrupt Mask 1h = Set Interrrupt Mask
1	INPUTRDY	R	0h	This indicates that the engine can take new input. This should not be used if DMA handshake is used (AES_DMA_HS.DMA_DATA_ACK set to 1) 0h = Clear Interrupt Mask 1h = Set Interrrupt Mask
0	OUTPUTRDY	R	0h	This indicates that the core has an output available to be read out. This should not be used if DMA handshake is used (AES_DMA_HS.DMA_DATA_ACK set to 1) 0h = Interrupt did not occur 1h = Interrupt occured

9 MIS Register (Offset = 1038h) [Reset = 00000000h]

MIS is shown in Figure 30-19 and described in Table 30-17.

Return to the Summary Table.

Masked interrupt status. This is an AND of the IMASK and RIS registers.

Figure 30-19. MIS Register

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

Table 30-26. MIS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R	0h	
3	CNTXTRDY	R	0h	This bit indicates that the context data registers can be overwritten, and the CPU is permitted to write next context. 0h = Clear Interrupt Mask 1h = Set Interrrupt Mask
2	SAVEDCNTXTRDY	R	0h	This bit indicates that an AES authentication TAG and/or IV block(s) is/are available for the CPU to retrieve. This bit is only asserted if the 'save_context' bit is set to '1b'. The bit is mutually exclusive with the 'context_ready' bit. 0h = Clear Interrupt Mask 1h = Set Interrrupt Mask
1	INPUTRDY	R	0h	This indicates that the engine can take new input. This should not be used if DMA handshake is used (AES_DMA_HS.DMA_DATA_ACK set to 1) 0h = Clear Interrupt Mask 1h = Set Interrrupt Mask
0	OUTPUTRDY	R	0h	This indicates that the core has an output available to be read out. This should not be used if DMA handshake is used (AES_DMA_HS.DMA_DATA_ACK set to 1) 0h = Interrupt did not occur 1h = Interrupt occured

ADVANCE INFORMATION

10 ISET Register (Offset = 1040h) [Reset = 00000000h]

ISET is shown in Figure 30-20 and described in Table 30-18.

Return to the Summary Table.

Interrupt set. Allows interrupts to be set by software (useful in diagnostics and safety checks). Writing a 1 to a bit in ISET will set the event and therefore the related RIS bit also gets set. If the interrupt is enabled through the mask, then the corresponding MIS bit is also set.

Figure 30-20. ISET Register

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

Table 30-28. ISET Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	W	0h	
3	CNTXTRDY	W	0h	This bit indicates that the context data registers can be overwritten, and the CPU is permitted to write next context. 0h = Clear Interrupt Mask 1h = Set Interrrupt Mask
2	SAVEDCNTXTRDY	W	0h	This bit indicates that an AES authentication TAG and/or IV block(s) is/are available for the CPU to retrieve. This bit is only asserted if the 'save_context' bit is set to '1b'. The bit is mutually exclusive with the 'context_ready' bit. 0h = Clear Interrupt Mask 1h = Set Interrrupt Mask
1	INPUTRDY	W	0h	This indicates that the engine can take new input.This should not be used if DMA handshake is used (AES_DMA_HS.DMA_DATA_ACK set to 1) 0h = Clear Interrupt Mask 1h = Set Interrrupt Mask
0	OUTPUTRDY	W	0h	This indicates that the core has an output available to be read out. This should not be used if DMA handshake is used (AES_DMA_HS.DMA_DATA_ACK set to 1) 0h = Writing 0 has no effect 1h = Set Interrupt

11 ICLR Register (Offset = 1048h) [Reset = 00000000h]

ICLR is shown in Figure 30-21 and described in Table 30-19.

Return to the Summary Table.

Interrupt clear. Write a 1 to clear corresponding Interrupt.

Figure 30-21. ICLR Register

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

Table 30-30. ICLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	W	0h	
3	CNTXTRDY	W	0h	This bit indicates that the context data registers can be overwritten, and the CPU is permitted to write next context. 0h = Clear Interrupt Mask 1h = Set Interrrupt Mask
2	SAVEDCNTXTRDY	W	0h	This bit indicates that an AES authentication TAG and/or IV block(s) is/are available for the CPU to retrieve. This bit is only asserted if the 'save_context' bit is set to '1b'. The bit is mutually exclusive with the 'context_ready' bit. 0h = Clear Interrupt Mask 1h = Set Interrrupt Mask
1	INPUTRDY	W	0h	This indicates that the engine can take new input. This should not be used if DMA handshake is used (AES_DMA_HS.DMA_DATA_ACK set to 1) 0h = Clear Interrupt Mask 1h = Set Interrrupt Mask
0	OUTPUTRDY	W	0h	This indicates that the core has an output available to be read out. This should not be used if DMA handshake is used (AES_DMA_HS.DMA_DATA_ACK set to 1) 0h = Writing 0 has no effect 1h = Clear Interrupt

ADVANCE INFORMATION

12 IIDX Register (Offset = 1050h) [Reset = 00000000h]

IIDX is shown in Figure 30-22 and described in Table 30-20.

Return to the Summary Table.

This register provides the highest priority enabled interrupt index. Value 0x00 means no event pending. Interrupt 1 is the highest priority, IIDX next highest, 4, 8, ... IIDX^31 is the least priority. That is, the least bit position that is set to 1 denotes the highest priority pending interrupt. The priority order is fixed. However, users can implement their own prioritization schemes using other registers that expose the full set of interrupts that have occurred. On each read, only one interrupt is indicated. On a read, the current interrupt (highest priority) is automatically cleared by the hardware and the corresponding interrupt flag in [RIS] and [MIS] are cleared as well. After a read from the CPU (not from the debug interface), the register is updated with the next highest priority interrupt, if none are pending, then it should display 0x0.

Figure 30-22. IIDX Register

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

Table 30-32. IIDX Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	
7-0	STAT	R	0h	Interrupt index status 00h = No interrupt pending 1h = AES trigger 0 DMA (Data Input trigger)

13 IMASK Register (Offset = 1058h) [Reset = 00000000h]

IMASK is shown in Figure 30-23 and described in Table 30-21.

Return to the Summary Table.

Interrupt Mask. If a bit is set, then corresponding interrupt is un-masked. Un-masking the interrupt causes the raw interrupt to be visible in IIDX, as well as MIS.

Figure 30-23. IMASK Register

31	30	29	28	27	26	25	24
RESERVED							
R/W-0h							
23	22	21	20	19	18	17	16
RESERVED							
R/W-0h							
15	14	13	12	11	10	9	8
RESERVED							
R/W-0h							
7	6	5	4	3	2	1	0
RESERVED							TRIG0
R/W-0h							R/W-0h

Table 30-34. IMASK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R/W	0h	
0	TRIG0	R/W	0h	TRIG0 event mask. 0h = Clear Interrupt Mask 1h = Set Interrrupt Mask

ADVANCE INFORMATION

14 RIS Register (Offset = 1060h) [Reset = 00000000h]

RIS is shown in Figure 30-24 and described in Table 30-22.

Return to the Summary Table.

Raw interrupt status. Reflects all pending interrupts, regardless of masking. The RIS register allows the user to implement a poll scheme. A flag set in this register can be cleared by writing 1 to the ICLR register bit even if the corresponding IMASK bit is not enabled.

Figure 30-24. RIS Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED							TRIG0
R-0h							R-0h

Table 30-36. RIS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	
0	TRIG0	R	0h	TRIG0 event 0h = Interrupt did not occur 1h = Interrupt occured

ADVANCE INFORMATION

15 MIS Register (Offset = 1068h) [Reset = 00000000h]

MIS is shown in Figure 30-25 and described in Table 30-23.

Return to the Summary Table.

Masked interrupt status. This is an AND of the IMASK and RIS registers.

Figure 30-25. MIS Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED							TRIG0
R-0h							R-0h

Table 30-38. MIS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	
0	TRIG0	R	0h	TRIG0 event 0h = Interrupt did not occur 1h = Interrupt occured

ADVANCE INFORMATION

16 ISET Register (Offset = 1070h) [Reset = 00000000h]

ISET is shown in Figure 30-26 and described in Table 30-24.

Return to the Summary Table.

Interrupt set. Allows interrupts to be set by software (useful in diagnostics and safety checks). Writing a 1 to a bit in ISET will set the event and therefore the related RIS bit also gets set. If the interrupt is enabled through the mask, then the corresponding MIS bit is also set.

Figure 30-26. ISET Register

31	30	29	28	27	26	25	24
RESERVED							
W-0h							
23	22	21	20	19	18	17	16
RESERVED							
W-0h							
15	14	13	12	11	10	9	8
RESERVED							
W-0h							
7	6	5	4	3	2	1	0
RESERVED							TRIG0
W-0h							W-0h

Table 30-40. ISET Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	W	0h	
0	TRIG0	W	0h	TRIG0 0h = Writing 0 has no effect 1h = Set Interrupt

17 ICLR Register (Offset = 1078h) [Reset = 00000000h]

ICLR is shown in Figure 30-27 and described in Table 30-25.

Return to the Summary Table.

Interrupt clear. Write a 1 to clear corresponding Interrupt.

Figure 30-27. ICLR Register

31	30	29	28	27	26	25	24
RESERVED							
W-0h							
23	22	21	20	19	18	17	16
RESERVED							
W-0h							
15	14	13	12	11	10	9	8
RESERVED							
W-0h							
7	6	5	4	3	2	1	0
RESERVED							TRIG0
W-0h							W-0h

Table 30-42. ICLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	W	0h	
0	TRIG0	W	0h	TRIG0 event 0h = Writing 0 has no effect 1h = Clear Interrupt

18 IIDX Register (Offset = 1080h) [Reset = 00000000h]

IIDX is shown in Figure 30-28 and described in Table 30-26.

Return to the Summary Table.

This register provides the highest priority enabled interrupt index. Value 0x00 means no event pending. Interrupt 1 is the highest priority, IIDX next highest, 4, 8, ... IIDX^31 is the least priority. That is, the least bit position that is set to 1 denotes the highest priority pending interrupt. The priority order is fixed. However, users can implement their own prioritization schemes using other registers that expose the full set of interrupts that have occurred. On each read, only one interrupt is indicated. On a read, the current interrupt (highest priority) is automatically cleared by the hardware and the corresponding interrupt flag in [RIS] and [MIS] are cleared as well. After a read from the CPU (not from the debug interface), the register is updated with the next highest priority interrupt, if none are pending, then it should display 0x0.

Figure 30-28. IIDX Register

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

Table 30-44. IIDX Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	
7-0	STAT	R	0h	Interrupt index status 00h = No interrupt pending 1h = AES DMA Trigger 1 (Data Output trigger)

19 IMASK Register (Offset = 1088h) [Reset = 00000000h]

IMASK is shown in Figure 30-29 and described in Table 30-27.

Return to the Summary Table.

Interrupt Mask. If a bit is set, then corresponding interrupt is un-masked. Un-masking the interrupt causes the raw interrupt to be visible in IIDX, as well as MIS.

Figure 30-29. IMASK Register

31	30	29	28	27	26	25	24
RESERVED							
R/W-0h							
23	22	21	20	19	18	17	16
RESERVED							
R/W-0h							
15	14	13	12	11	10	9	8
RESERVED							
R/W-0h							
7	6	5	4	3	2	1	0
RESERVED							TRIG1
R/W-0h							R/W-0h

Table 30-46. IMASK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R/W	0h	
0	TRIG1	R/W	0h	TRIG1 event mask. 0h = Clear Interrupt Mask 1h = Set Interrrupt Mask

ADVANCE INFORMATION

20 RIS Register (Offset = 1090h) [Reset = 00000000h]

RIS is shown in Figure 30-30 and described in Table 30-28.

Return to the Summary Table.

Raw interrupt status. Reflects all pending interrupts, regardless of masking. The RIS register allows the user to implement a poll scheme. A flag set in this register can be cleared by writing 1 to the ICLR register bit even if the corresponding IMASK bit is not enabled.

Figure 30-30. RIS Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED							TRIG1
R-0h							R-0h

Table 30-48. RIS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	
0	TRIG1	R	0h	TRIG1 event 0h = Interrupt did not occur 1h = Interrupt occured

21 MIS Register (Offset = 1098h) [Reset = 00000000h]

MIS is shown in Figure 30-31 and described in Table 30-29.

Return to the Summary Table.

Masked interrupt status. This is an AND of the IMASK and RIS registers.

Figure 30-31. MIS Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED							TRIG1
R-0h							R-0h

Table 30-50. MIS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	
0	TRIG1	R	0h	TRIG1 event 0h = Interrupt did not occur 1h = Interrupt occured

22 ISET Register (Offset = 10A0h) [Reset = 00000000h]

ISET is shown in Figure 30-32 and described in Table 30-30.

Return to the Summary Table.

Interrupt set. Allows interrupts to be set by software (useful in diagnostics and safety checks). Writing a 1 to a bit in ISET will set the event and therefore the related RIS bit also gets set. If the interrupt is enabled through the mask, then the corresponding MIS bit is also set.

Figure 30-32. ISET Register

31	30	29	28	27	26	25	24
RESERVED							
W-0h							
23	22	21	20	19	18	17	16
RESERVED							
W-0h							
15	14	13	12	11	10	9	8
RESERVED							
W-0h							
7	6	5	4	3	2	1	0
RESERVED							TRIG1
W-0h							W-0h

Table 30-52. ISET Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	W	0h	
0	TRIG1	W	0h	TRIG1 event 0h = Writing 0 has no effect 1h = Set Interrupt

23 ICLR Register (Offset = 10A8h) [Reset = 00000000h]

ICLR is shown in Figure 30-33 and described in Table 30-31.

Return to the Summary Table.

Interrupt clear. Write a 1 to clear corresponding Interrupt.

Figure 30-33. ICLR Register

31	30	29	28	27	26	25	24
RESERVED							
W-0h							
23	22	21	20	19	18	17	16
RESERVED							
W-0h							
15	14	13	12	11	10	9	8
RESERVED							
W-0h							
7	6	5	4	3	2	1	0
RESERVED							TRIG1
W-0h							W-0h

Table 30-54. ICLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	W	0h	
0	TRIG1	W	0h	TRIG1 event 0h = Writing 0 has no effect 1h = Clear Interrupt

24 EVT_MODE Register (Offset = 10E0h) [Reset = 00000000h]

EVT_MODE is shown in Figure 30-34 and described in Table 30-32.

Return to the Summary Table.

Event mode register. It is used to select whether each line is disabled, in software mode (software clears the RIS) or in hardware mode (hardware clears the RIS)

Figure 30-34. EVT_MODE Register

31	30	29	28	27	26	25	24
RESERVED							
R/W-0h							
23	22	21	20	19	18	17	16
RESERVED							
R/W-0h							
15	14	13	12	11	10	9	8
RESERVED							
R/W-0h							
7	6	5	4	3	2	1	0
RESERVED		EVT2_CFG		EVT1_CFG		INT0_CFG	
R/W-0h		R-0h		R-0h		R-0h	

Table 30-56. EVT_MODE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-6	RESERVED	R/W	0h	
5-4	EVT2_CFG	R	2h	Event line mode select for event corresponding to [IPSTANDARD.INT_EVENT2] 0h = The interrupt or event line is disabled. 1h = The interrupt or event line is in software mode. Software must clear the RIS. 2h = The interrupt or event line is in hardware mode. The hardware (another module) clears automatically the associated RIS flag.
3-2	EVT1_CFG	R	2h	Event line mode select for event corresponding to [IPSTANDARD.INT_EVENT1] 0h = The interrupt or event line is disabled. 1h = The interrupt or event line is in software mode. Software must clear the RIS. 2h = The interrupt or event line is in hardware mode. The hardware (another module) clears automatically the associated RIS flag.
1-0	INT0_CFG	R	1h	Event line mode select for event corresponding to [IPSTANDARD.INT_EVENT0] 0h = The interrupt or event line is disabled. 1h = The interrupt or event line is in software mode. Software must clear the RIS. 2h = The interrupt or event line is in hardware mode. The hardware (another module) clears automatically the associated RIS flag.

25 GCMCCM_TAG0 Register (Offset = 1100h) [Reset = 00000000h]

GCMCCM_TAG0 is shown in Figure 30-35 and described in Table 30-33.

Return to the Summary Table.

CBC-MAC third key (LSW) / GCM & CCM Intermediate TAG (LSW)
For CBC-MAC: Pre-calculated CBC-MAC third key used to perform a final XOR operation on the last input data block.
For CCM: This register is internally used to store intermediate values. This register must be initialized with zeroes when writing a new CCM context. To restore an interrupted CCM operation, this register needs to be loaded with the intermediate TAG.
For GCM: This register is internally used to store intermediate values. This register must be initialized with zeroes when writing a new GCM context. To restore an interrupted GCM operation, this register needs to be loaded with the intermediate TAG.

Figure 30-35. GCMCCM_TAG0 Register

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

Table 30-58. GCMCCM_TAG0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DATA	W	0h	Key data

26 GCMCCM_TAG1 Register (Offset = 1104h) [Reset = 00000000h]

GCMCCM_TAG1 is shown in Figure 30-36 and described in Table 30-34.

Return to the Summary Table.

CBC-MAC third key / GCM & CCM Intermediate TAG
For CBC-MAC: Pre-calculated CBC-MAC third key used to perform a final XOR operation on the last input data block.
For CCM: This register is internally used to store intermediate values. This register must be initialized with zeroes when writing a new CCM context. To restore an interrupted CCM operation, this register needs to be loaded with the intermediate TAG.
For GCM: This register is internally used to store intermediate values. This register must be initialized with zeroes when writing a new GCM context. To restore an interrupted GCM operation, this register needs to be loaded with the intermediate TAG.

Figure 30-36. GCMCCM_TAG1 Register

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

Table 30-60. GCMCCM_TAG1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DATA	W	0h	Key data

27 GCMCCM_TAG2 Register (Offset = 1108h) [Reset = 00000000h]

GCMCCM_TAG2 is shown in Figure 30-37 and described in Table 30-35.

Return to the Summary Table.

CBC-MAC third key / GCM & CCM Intermediate TAG
For CBC-MAC: Pre-calculated CBC-MAC third key used to perform a final XOR operation on the last input data block.
For CCM: This register is internally used to store intermediate values. This register must be initialized with zeroes when writing a new CCM context. To restore an interrupted CCM operation, this register needs to be loaded with the intermediate TAG.
For GCM: This register is internally used to store intermediate values. This register must be initialized with zeroes when writing a new GCM context. To restore an interrupted GCM operation, this register needs to be loaded with the intermediate TAG.

Figure 30-37. GCMCCM_TAG2 Register

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

Table 30-62. GCMCCM_TAG2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DATA	W	0h	Key data

28 GCMCCM_TAG3 Register (Offset = 110Ch) [Reset = 00000000h]

GCMCCM_TAG3 is shown in Figure 30-38 and described in Table 30-36.

Return to the Summary Table.

CBC-MAC third key (MSW) / GCM & CCM Intermediate TAG (MSW)
For CBC-MAC: Pre-calculated CBC-MAC third key used to perform a final XOR operation on the last input data block.
For CCM: This register is internally used to store intermediate values. This register must be initialized with zeroes when writing a new CCM context. To restore an interrupted CCM operation, this register needs to be loaded with the intermediate TAG.
For GCM: This register is internally used to store intermediate values. This register must be initialized with zeroes when writing a new GCM context. To restore an interrupted GCM operation, this register needs to be loaded with the intermediate TAG.

Figure 30-38. GCMCCM_TAG3 Register

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

Table 30-64. GCMCCM_TAG3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DATA	W	0h	Key data

29 GHASH_H0 Register (Offset = 1110h) [Reset = 00000000h]

GHASH_H0 is shown in Figure 30-39 and described in Table 30-37.

Return to the Summary Table.

CCM & CBC-MAC second key (LSW) / GCM Hash Key input (LSW)
For CBC-MAC: Pre-calculated CBC-MAC second key used to perform a final XOR operation on the last input data block.
For GCM: Hash key
can be calculated internal or written via these registers. Only used for GHASH (GCM) modes.
For a CPU write operation, these registers must be written with the new values to be subsequently transferred to the engine.

Figure 30-39. GHASH_H0 Register

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

Table 30-66. GHASH_H0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DATA	W	0h	Key data

30 GHASH_H1 Register (Offset = 1114h) [Reset = 00000000h]

GHASH_H1 is shown in Figure 30-40 and described in Table 30-38.

Return to the Summary Table.

CCM & CBC-MAC second key / GCM Hash Key input
For CBC-MAC: Pre-calculated CBC-MAC second key used to perform a final XOR operation on the last input data block.
For GCM: Hash key
can be calculated internal or written via these registers. Only used for GHASH (GCM) modes.
For a CPU write operation, these registers must be written with the new values to be subsequently transferred to the engine.

Figure 30-40. GHASH_H1 Register

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

Table 30-68. GHASH_H1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DATA	W	0h	Key data

ADVANCE INFORMATION

31 GHASH_H2 Register (Offset = 1118h) [Reset = 00000000h]

GHASH_H2 is shown in Figure 30-41 and described in Table 30-39.

Return to the Summary Table.

CCM & CBC-MAC second key / GCM Hash Key input
For CBC-MAC: Pre-calculated CBC-MAC second key used to perform a final XOR operation on the last input data block.
For GCM: Hash key
can be calculated internal or written via these registers. Only used for GHASH (GCM) modes.
For a CPU write operation, these registers must be written with the new values to be subsequently transferred to the engine.

Figure 30-41. GHASH_H2 Register

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

Table 30-70. GHASH_H2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DATA	W	0h	Key data

32 GHASH_H3 Register (Offset = 111Ch) [Reset = 00000000h]

GHASH_H3 is shown in Figure 30-42 and described in Table 30-40.

Return to the Summary Table.

CCM & CBC-MAC second key (MSW) / GCM Hash Key input (MSW)
For CBC-MAC: Pre-calculated CBC-MAC second key used to perform a final XOR operation on the last input data block.
For GCM: Hash key
can be calculated internal or written via these registers. Only used for GHASH (GCM) modes.
For a CPU write operation, these registers must be written with the new values to be subsequently transferred to the engine.

Figure 30-42. GHASH_H3 Register

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

Table 30-72. GHASH_H3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DATA	W	0h	Key data

33 KEY0 Register (Offset = 1120h) [Reset = 00000000h]

KEY0 is shown in Figure 30-43 and described in Table 30-41.

Return to the Summary Table.

KEY (LSW)

Figure 30-43. KEY0 Register

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

Table 30-74. KEY0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DATA	W	0h	Key data

ADVANCE INFORMATION

34 KEY1 Register (Offset = 1124h) [Reset = 00000000h]

KEY1 is shown in Figure 30-44 and described in Table 30-42.

Return to the Summary Table.

KEY

Figure 30-44. KEY1 Register

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

Table 30-76. KEY1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DATA	W	0h	Key data

ADVANCE INFORMATION

35 KEY2 Register (Offset = 1128h) [Reset = 00000000h]

KEY2 is shown in Figure 30-45 and described in Table 30-43.

Return to the Summary Table.

KEY

Figure 30-45. KEY2 Register

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

Table 30-78. KEY2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DATA	W	0h	Key data

ADVANCE INFORMATION

36 KEY3 Register (Offset = 112Ch) [Reset = 00000000h]

KEY3 is shown in Figure 30-46 and described in Table 30-44.

Return to the Summary Table.

KEY

Figure 30-46. KEY3 Register

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

Table 30-80. KEY3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DATA	W	0h	Key data

ADVANCE INFORMATION

37 KEY4 Register (Offset = 1130h) [Reset = 00000000h]

KEY4 is shown in Figure 30-47 and described in Table 30-45.

Return to the Summary Table.

KEY

Figure 30-47. KEY4 Register

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

Table 30-82. KEY4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DATA	W	0h	Key data

ADVANCE INFORMATION

38 KEY5 Register (Offset = 1134h) [Reset = 00000000h]

KEY5 is shown in Figure 30-48 and described in Table 30-46.

Return to the Summary Table.

KEY

Figure 30-48. KEY5 Register

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

Table 30-84. KEY5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DATA	W	0h	Key data

ADVANCE INFORMATION

39 KEY6 Register (Offset = 1138h) [Reset = 00000000h]

KEY6 is shown in Figure 30-49 and described in Table 30-47.

Return to the Summary Table.

KEY

Figure 30-49. KEY6 Register

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

Table 30-86. KEY6 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DATA	W	0h	Key data

ADVANCE INFORMATION

40 KEY7 Register (Offset = 113Ch) [Reset = 00000000h]

KEY7 is shown in Figure 30-50 and described in Table 30-48.

Return to the Summary Table.

KEY (MSW)

Figure 30-50. KEY7 Register

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

Table 30-88. KEY7 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DATA	W	0h	Key data

41 IV0 Register (Offset = 1140h) [Reset = 00000000h]

IV0 is shown in Figure 30-51 and described in Table 30-49.

Return to the Summary Table.

IV (LSW)

Figure 30-51. IV0 Register

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

Table 30-90. IV0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DATA	W	0h	Key data

42 IV1 Register (Offset = 1144h) [Reset = 00000000h]

IV1 is shown in Figure 30-52 and described in Table 30-50.

Return to the Summary Table.

IV

Figure 30-52. IV1 Register

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

Table 30-92. IV1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DATA	W	0h	Key data

43 IV2 Register (Offset = 1148h) [Reset = 00000000h]

IV2 is shown in Figure 30-53 and described in Table 30-51.

Return to the Summary Table.

IV

Figure 30-53. IV2 Register

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

Table 30-94. IV2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DATA	W	0h	Key data

44 IV3 Register (Offset = 114Ch) [Reset = 00000000h]

IV3 is shown in Figure 30-54 and described in Table 30-52.

Return to the Summary Table.

IV

Figure 30-54. IV3 Register

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

Table 30-96. IV3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DATA	W	0h	Key data

45 CTRL Register (Offset = 1150h) [Reset = 80000000h]

CTRL is shown in Figure 30-55 and described in Table 30-53.

Return to the Summary Table.

Input/Output Buffer Control and Mode selection. The content of this register determines the mode of operation of the EIP-39.

Figure 30-55. CTRL Register

31	30	29	28	27	26	25	24
CNTXT_RDY	SAVED_CNTXT_RDY	SAVE_CNTXT	GCM_CONT	GET_DIGEST	OFB_GCM_CCM_CONT	RESERVED	CCMM
R-1h	R-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
CCMM		CCML			CCM	GCM	
R/W-0h		R/W-0h			R/W-0h	R/W-0h	
15	14	13	12	11	10	9	8
CBCMAC	RESERVED				CFB	ICM	CTR_WIDTH
R/W-0h	R/W-0h				R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
CTR_WIDTH	CTR	CBC	KEYSIZE		DIR	INPUT_RDY	OUTPUT_RDY
R/W-0h	R/W-0h	R/W-0h	R/W-0h		R/W-0h	R-0h	R-0h

Table 30-98. CTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31	CNTXT_RDY	R	1h	If '1b', this read-only status bit indicates that the context data registers can be overwritten, and the CPU is permitted to write the next context. 0h = Not ready 1h = Ready
30	SAVED_CNTXT_RDY	R	0h	If '1b', this read-only status bit indicates that an AES authentication TAG and/or IV block(s) is/are available for the Host to retrieve. This bit is only asserted if the 'save_context' bit is set to '1b'. The bit is mutually exclusive with the 'context_ready' bit. 0h = Not ready 1h = Ready
29	SAVE_CNTXT	R/W	0h	This bit is used to indicate that an authentication TAG or result IV needs to be stored as a result context. If this bit is set, context output DMA and/or interrupt will be asserted if the operation is finished, and related signals are enabled. Typically, this value must be set for authentication modes returning a TAG (CBC-MAC, GCM and CCM), or for basic encryption modes that require future continuation with the current result IV. If this bit is set, the engine will hold its full context until the TAG and/or IV registers are read. Only after reading the TAG or IV, a new DMA request for a new (input) context will be asserted. If this bit is not set, the engine will assert the context input DMA request signal directly after starting to process the last block with the current context. 0h = No effect 1h = Enable

Table 30-98. CTRL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
28	GCM_CONT	R/W	0h	Continue processing of an interrupted AES-GCM or AES-CCM operation in the crypto/payload phase. Set this write-only signal to '1b' together with the regular mode bit settings for a GCM or CCM operation, to continue processing from the next full block (128 bits) boundary. Before setting this bit all applicable context to resume processing must have been loaded into the engine: Keys, IV, intermediate digest/TAG and block counter. The mode can be written together with this bit, as it is part of the same register. 0h = No effect 1h = Enable
27	GET_DIGEST	R/W	0h	Interrupt processing and generate an intermediate digest during an AES-GCM or AES-CCM operation. Set this write-only signal to '1b' to interrupt GCM or CCM processing at the next full block (128 bits) boundary. An intermediate digest may be requested during the encryption/decryption data phase or in the AAD phase. Note: Interruption can only be done on full block (128 bits) boundaries. The minimum number of remaining bytes to resume and finalize the operation, must be greater than or equal to 1. 0h = No effect 1h = Enable
26	OFB_GCM_CCM_CONT	R/W	0h	This bit has a dual use, depending on the selection of CCM/GCM, see bits [18:16]. If CCM/GCM is not selected: If this bit is set to '1b', full block AES output feedback mode (OFB-128) is selected. If CCM/GCM is selected: Continue processing of an interrupted AES-GCM or AES-CCM operation in the AAD phase. Set this write-only signal to '1b' together with the regular mode bit settings for a GCM or CCM operation, to continue processing from the next full AAD block (128 bits) boundary. Before setting this bit all applicable context to resume processing must have been loaded into the engine: Keys, IV, intermediate digest/TAG, block counter and the CCM align data word (the latter is for CCM mode only). The mode can be written together with this bit, as it is part of the same register. 1h = Continue GCM/CCM processing in AAD phase
25	RESERVED	R/W	0h	
24-22	CCMM	R/W	0h	Defines "M" that indicates the length of the authentication field for CCM operations the authentication field length equals two times (the value of CCM-M plus one). Note: The EIP-39 always returns a 128-bit authentication field, of which the M least significant bytes are valid. All values are supported. 0h = Length is 1 7h = Length is 8
21-19	CCML	R/W	0h	Defines "L" that indicates the width of the length field for CCM operations the length field in bytes equals the value of CMM-L plus one. All values are supported. 0h = Length is 1 7h = Length is 8
18	CCM	R/W	0h	If set to '1b', AES-CCM is selected, this is a combined mode, using AES for both authentication and encryption. In addition to the CCM bit, the CTR mode bit must be set such that AES-CTR is enabled. Other combinations with CCM are invalid. 0h = Disable CBC mode 1h = Select CBC mode

Table 30-98. CTRL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
17-16	GCM	R/W	0h	If not set to '00b', AES-GCM mode is selected, this is a combined mode, using the Galois field multiplier GF(2128) for authentication and AES-CTR mode for encryption, the bits specify the GCM mode: 01b = GHASH with H loaded and Y0-encrypted forced to zero 10b = GHASH with H loaded and Y0-encrypted calculated internally 11b = Autonomous GHASH (both H and Y0-encrypted calculated internally) Note: Besides GCM, the CTR mode bits must also be set to '1b' to enable GCM with AES-CTR if the CTR bit is not set a GHASH (authentication) only operation is performed. A GHASH only operation is only allowed if the GCM mode is set to '01b' and the direction bit is set to '0b'. Other modes may not be selected in combination with GCM. Table 14 below shows the valid combinations for the GCM and CTR mode bits, all other options are invalid and must not be selected. 1h = GHASH with H loaded and Y0-encrypted forced to 0. 2h = GHASH with H loaded and Y0-encrypted calculated internally 3h = Autonomous GHASH (both H and Y0-encrypted calculated internally)
15	CBCMAC	R/W	0h	If set to '1b', AES-CBC MAC is selected, the Direction bit must be set to '1' for this mode. 0h = Disable CBC mode 1h = Select CBC mode
14-11	RESERVED	R/W	0h	
10	CFB	R/W	0h	If set to '1b', AES cipher feedback mode CFB is selected. Use the ctr_width field to specify the feedback width. 0h = Disable CBC mode 1h = Select CBC mode
9	ICM	R/W	0h	When the CFB bit is set, specifies the CFB mode feedback width: 0h = Disable CBC mode 1h = Select CBC mode
8-7	CTR_WIDTH	R/W	0h	When the CTR bit is set, specifies the counter width for AES-CTR mode. When the CFB bit is set, specifies the CFB mode feedback width: 0h = CFB-128 mode 1h = 64-bit counter 2h = 96-bit counter 3h = 128-bit counter
6	CTR	R/W	0h	If set to '1b', AES counter mode (CTR) is selected. Note: This bit must also be set for GCM and CCM, when encryption/decryption is required. 0h = Disable CBC mode 1h = Select CBC mode
5	CBC	R/W	0h	If set to '1b', cipher-block-chaining (CBC) mode is selected. 0h = Disable CBC mode 1h = Select CBC mode
4-3	KEYSIZE	R/W	0h	Specifies the encryption strength / key width 1h = 128-bit key 3h = 256-bit key
2	DIR	R/W	0h	Direction. If set to '1b' an encrypt operation is performed. If set to '0b' a decrypt operation is performed. Note: This bit must be written with a '1b' when CBC-MAC is selected. 0h = Decryption 1h = Encryption
1	INPUT_RDY	R	0h	Ready for input. If '1b', this read-only status bit indicates that the 16-byte input buffer is empty, and the CPU is permitted to write the next block of data. After reset, this bit is '0'. After writing a context, this bit will become '1b'. 0h = Not Ready 1h = Ready

Table 30-98. CTRL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
0	OUTPUT_RDY	R	0h	Output Ready. If '1b', this read-only status bit indicates that an AES output block is available for the CPU to retrieve. 0h = Not Ready 1h = Ready

46 C_LENGTH_0 Register (Offset = 1154h) [Reset = 00000000h]

C_LENGTH_0 is shown in Figure 30-56 and described in Table 30-54.

Return to the Summary Table.

Crypto data length (LSW). These registers buffer the Length values to the EIP-39. While processing, the length values decrement to zero. If both lengths are zero, the data stream is finished, and a new context is requested. For basic AES modes (ECB/CBC/CTR/ICM/CFB/OFB), a crypto length of '0' can be written if the context DMA is disabled. Writing a zero length results in continued data requests until a new context is written. For the other modes (GCM and CCM) no (new) data requests are done if the length decrements to or equals zero.
It is advised to write a new length per packet. If the length registers decrement to zero, no new data is processed until a new context or length value is written.
When writing a new context without writing the length registers, the length register values from the previous context are reused.

Figure 30-56. C_LENGTH_0 Register

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

Table 30-100. C_LENGTH_0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DATA	W	0h	Bits [60:0] of the crypto length registers (LSW and MSW) store the cryptographic data length in bytes for all modes. Once processing with this context is started, this length decrements to zero. Data lengths up to (261-1) bytes are allowed. For GCM, any value up to 236-32 bytes can be used. This is because a 32-bit counter mode is used the maximum number of 128-bit blocks is 232-2, resulting in a maximum number of bytes of 236-32. A write to this register triggers the engine to start using this context. This is valid for all modes except GCM and CCM. Note that for the combined modes, this length does not include the authentication only data the authentication length is specified in the AES_AAD_LENGTH register below. All modes must have a length > 0. For the combined modes, it is allowed to have one of the lengths equal to zero. For the basic encryption modes (ECB/CBC/CTR/ICM/CFB/OFB) it is allowed to program zero to the length field in that case the length is assumed infinite. All data must be byte (8-bit) aligned for stream cipher modes bit aligned data streams are not supported. For block cipher modes, the data length must be programmed in multiples of the block cipher size, 16 bytes.

47 C_LENGTH_1 Register (Offset = 1158h) [Reset = 00000000h]

C_LENGTH_1 is shown in Figure 30-57 and described in Table 30-55.

Return to the Summary Table.

Crypto data length (MSW). These registers buffer the Length values to the EIP-39. While processing, the length values decrement to zero. If both lengths are zero, the data stream is finished, and a new context is requested. For basic AES modes (ECB/CBC/CTR/ICM/CFB/OFB), a crypto length of '0' can be written if the context DMA is disabled. Writing a zero length results in continued data requests until a new context is written. For the other modes (GCM and CCM) no (new) data requests are done if the length decrements to or equals zero.
It is advised to write a new length per packet. If the length registers decrement to zero, no new data is processed until a new context or length value is written.
When writing a new context without writing the length registers, the length register values from the previous context are reused.

Figure 30-57. C_LENGTH_1 Register

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

Table 30-102. C_LENGTH_1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	W	0h	
28-0	DATA	W	0h	Bits [60:0] of the crypto length registers (LSW and MSW) store the cryptographic data length in bytes for all modes. Once processing with this context is started, this length decrements to zero. Data lengths up to (261-1) bytes are allowed. For GCM, any value up to 236-32 bytes can be used. This is because a 32-bit counter mode is used the maximum number of 128-bit blocks is 232-2, resulting in a maximum number of bytes of 236-32. A write to this register triggers the engine to start using this context. This is valid for all modes except GCM and CCM. Note that for the combined modes, this length does not include the authentication only data the authentication length is specified in the AES_AAD_LENGTH register below. All modes must have a length > 0. For the combined modes, it is allowed to have one of the lengths equal to zero. For the basic encryption modes (ECB/CBC/CTR/ICM/CFB/OFB) it is allowed to program zero to the length field in that case the length is assumed infinite. All data must be byte (8-bit) aligned for stream cipher modes bit aligned data streams are not supported. For block cipher modes, the data length must be programmed in multiples of the block cipher size, 16 bytes.

48 AAD_LENGTH Register (Offset = 115Ch) [Reset = 00000000h]

AAD_LENGTH is shown in Figure 30-58 and described in Table 30-56.

Return to the Summary Table.

AAD Data Length

Figure 30-58. AAD_LENGTH Register

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

Table 30-104. AAD_LENGTH Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DATA	W	0h	Bits [31:0] of the authentication length register store the authentication data length in bytes for combined modes only (GCM or CCM) Supported AAD-lengths for CCM are from 0 to (216-28) bytes. For GCM any value up to (232-1) bytes can be used. Once processing with this context is started, this length decrements to zero. A write to this register triggers the engine to start using this context for GCM and CCM.

49 DATA0 Register (Offset = 1160h) [Reset = 00000000h]

DATA0 is shown in Figure 30-59 and described in Table 30-57.

Return to the Summary Table.

Data input (LSW) / Data output (LSW)\. The Data Input/Output Registers buffer the input/output data blocks to/from the EIP-39. Notice that the data input buffer (AES_DATA_IN_n) and data output buffer (AES_DATA_OUT_n) are mapped to the same address locations. Writes to these addresses load the Input Buffer while reads pull from the Output Buffer. Therefore, for write access, the data input buffer is written for read access, the data output buffer is read. The data input buffer must be written prior to starting an operation. The data output buffer contains valid data on completion of an operation. All writes from, and reads to, these registers are tracked independently per direction. Therefore, any 128-bit data block can be split over multiple 32-bit word transfers, which can be mixed with other transfers over the external interface.

Figure 30-59. DATA0 Register

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

Table 30-106. DATA0 Register Field Descriptions

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

ADVANCE INFORMATION

50 DATA1 Register (Offset = 1164h) [Reset = 00000000h]

DATA1 is shown in Figure 30-60 and described in Table 30-58.

Return to the Summary Table.

Data input / Data output. The Data Input/Output Registers buffer the input/output data blocks to/from the EIP-39. Notice that the data input buffer (AES_DATA_IN_n) and data output buffer (AES_DATA_OUT_n) are mapped to the same address locations. Writes to these addresses load the Input Buffer while reads pull from the Output Buffer. Therefore, for write access, the data input buffer is written
for read access, the data output buffer is read. The data input buffer must be written prior to starting an operation. The data output buffer contains valid data on completion of an operation. All writes from, and reads to, these registers are tracked independently per direction. Therefore, any 128-bit data block can be split over multiple 32-bit word transfers, which can be mixed with other transfers over the external interface.

Figure 30-60. DATA1 Register

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

Table 30-108. DATA1 Register Field Descriptions

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

51 DATA2 Register (Offset = 1168h) [Reset = 00000000h]

DATA2 is shown in Figure 30-61 and described in Table 30-59.

Return to the Summary Table.

Data input / Data output. The Data Input/Output Registers buffer the input/output data blocks to/from the EIP-39. Notice that the data input buffer (AES_DATA_IN_n) and data output buffer (AES_DATA_OUT_n) are mapped to the same address locations. Writes to these addresses load the Input Buffer while reads pull from the Output Buffer. Therefore, for write access, the data input buffer is written
for read access, the data output buffer is read. The data input buffer must be written prior to starting an operation. The data output buffer contains valid data on completion of an operation. All writes from, and reads to, these registers are tracked independently per direction. Therefore, any 128-bit data block can be split over multiple 32-bit word transfers, which can be mixed with other transfers over the external interface.

Figure 30-61. DATA2 Register

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

Table 30-110. DATA2 Register Field Descriptions

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

52 DATA3 Register (Offset = 116Ch) [Reset = 00000000h]

DATA3 is shown in Figure 30-62 and described in Table 30-60.

Return to the Summary Table.

Data input (MSW) / Data output (MSW). The Data Input/Output Registers buffer the input/output data blocks to/from the EIP-39. Notice that the data input buffer (AES_DATA_IN_n) and data output buffer (AES_DATA_OUT_n) are mapped to the same address locations. Writes to these addresses load the Input Buffer while reads pull from the Output Buffer. Therefore, for write access, the data input buffer is written for read access, the data output buffer is read. The data input buffer must be written prior to starting an operation. The data output buffer contains valid data on completion of an operation. All writes from, and reads to, these registers are tracked independently per direction. Therefore, any 128-bit data block can be split over multiple 32-bit word transfers, which can be mixed with other transfers over the external interface.

Figure 30-62. DATA3 Register

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

Table 30-112. DATA3 Register Field Descriptions

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

ADVANCE INFORMATION

53 TAG0 Register (Offset = 1170h) [Reset = 00000000h]

TAG0 is shown in Figure 30-63 and described in Table 30-61.

Return to the Summary Table.

Hash result (LSW). These registers buffer the TAG from the EIP-39. The registers are shared with the intermediate authentication result registers but cannot be read until the processing is finished.

Figure 30-63. TAG0 Register

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

Table 30-114. TAG0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DATA	R	0h	For a CPU read operation, these registers contain the last 128-bit TAG output of the EIP-39 the TAG is available until the next context is written. This register will only contain valid data if the TAG is available, when the 'saved_context_ready' or 'get_digest' bit from AES_CTRL register is set. In case of get_digest, the output will be an intermediate TAG for CCM or GCM operation continuation. During processing or for operations/modes that do not return a TAG, reads from this register returns data from the IV register. For operations that do return a TAG in the IV register, the IV register must be accessed directly.

54 TAG1 Register (Offset = 1174h) [Reset = 00000000h]

TAG1 is shown in Figure 30-64 and described in Table 30-62.

Return to the Summary Table.

Hash result. These registers buffer the TAG from the EIP-39. The registers are shared with the intermediate authentication result registers but cannot be read until the processing is finished.

Figure 30-64. TAG1 Register

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

Table 30-116. TAG1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DATA	R	0h	For a CPU read operation, these registers contain the last 128-bit TAG output of the EIP-39 the TAG is available until the next context is written. This register will only contain valid data if the TAG is available, when the 'saved_context_ready' or 'get_digest' bit from AES_CTRL register is set. In case of get_digest, the output will be an intermediate TAG for CCM or GCM operation continuation. During processing or for operations/modes that do not return a TAG, reads from this register returns data from the IV register. For operations that do return a TAG in the IV register, the IV register must be accessed directly.

55 TAG2 Register (Offset = 1178h) [Reset = 00000000h]

TAG2 is shown in Figure 30-65 and described in Table 30-63.

Return to the Summary Table.

Hash result. These registers buffer the TAG from the EIP-39. The registers are shared with the intermediate authentication result registers but cannot be read until the processing is finished.

Figure 30-65. TAG2 Register

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

Table 30-118. TAG2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DATA	R	0h	For a CPU read operation, these registers contain the last 128-bit TAG output of the EIP-39 the TAG is available until the next context is written. This register will only contain valid data if the TAG is available, when the 'saved_context_ready' or 'get_digest' bit from AES_CTRL register is set. In case of get_digest, the output will be an intermediate TAG for CCM or GCM operation continuation. During processing or for operations/modes that do not return a TAG, reads from this register returns data from the IV register. For operations that do return a TAG in the IV register, the IV register must be accessed directly.

56 TAG3 Register (Offset = 117Ch) [Reset = 00000000h]

TAG3 is shown in Figure 30-66 and described in Table 30-64.

Return to the Summary Table.

Hash result (MSW). These registers buffer the TAG from the EIP-39. The registers are shared with the intermediate authentication result registers but cannot be read until the processing is finished.

Figure 30-66. TAG3 Register

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

Table 30-120. TAG3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DATA	R	0h	For a CPU read operation, these registers contain the last 128-bit TAG output of the EIP-39 the TAG is available until the next context is written. This register will only contain valid data if the TAG is available, when the 'saved_context_ready' or 'get_digest' bit from AES_CTRL register is set. In case of get_digest, the output will be an intermediate TAG for CCM or GCM operation continuation. During processing or for operations/modes that do not return a TAG, reads from this register returns data from the IV register. For operations that do return a TAG in the IV register, the IV register must be accessed directly.

57 STATUS Register (Offset = 1180h) [Reset = 00000000h]

STATUS is shown in Figure 30-67 and described in Table 30-65.

Return to the Summary Table.

Status register

Figure 30-67. STATUS Register

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

Table 30-122. STATUS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	
0	KEYWR	R	0h	Key write status. 0 - user write to KEY register is allowed. 1 - user write to KEY register is ignored. In order to allow user write, perform a module reset. 0h = User write to KEY MMR is allowed 1h = User write to KEY MMR is disabled. Writing has no effect.

58 DATA_IN Register (Offset = 1184h) [Reset = 00000000h]

DATA_IN is shown in Figure 30-68 and described in Table 30-66.

Return to the Summary Table.

Data-in register: alias for DATA0/1/2/3 at a single address for DMA addressing

Figure 30-68. DATA_IN Register

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

Table 30-124. DATA_IN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DATA	W	0h	Data input word

ADVANCE INFORMATION

59 DATA_OUT Register (Offset = 1188h) [Reset = 00000000h]

DATA_OUT is shown in Figure 30-69 and described in Table 30-67.

Return to the Summary Table.

Data-out register: alias for DATA0/1/2/3 at a single address for DMA addressing

Figure 30-69. DATA_OUT Register

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

Table 30-126. DATA_OUT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DATA	R	0h	Data output word

60 FORCE_IN_AV Register (Offset = 11D0h) [Reset = 00000000h]

FORCE_IN_AV is shown in Figure 30-70 and described in Table 30-68.

Return to the Summary Table.

Data control register for input data. This write-only register provides a means to force the availability of the input data buffer of the EIP-39.

Figure 30-70. FORCE_IN_AV Register

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

Table 30-128. FORCE_IN_AV Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DATA	W	0h	Any write to this register forces the input data buffer to valid and will force the engine to start processing this data. The data written here is not used. The core must be configured to have input and output data acknowledge be I/O register based

61 CCM_ALN_WRD Register (Offset = 11D4h) [Reset = 00000000h]

CCM_ALN_WRD is shown in Figure 30-71 and described in Table 30-69.

Return to the Summary Table.

AES-CCM AAD alignment data word. This register provides a means to access an internal EIP-39 register that stores alignment data bytes during the AAD phase of AES-CCM processing. This register needs to be read and stored when an AES-CCM operation is interrupted during the AAD phase. This value needs to be restored by writing this register, when resuming that AES-CCM operation in a later session.

Figure 30-71. CCM_ALN_WRD Register

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

Table 30-130. CCM_ALN_WRD Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DATA	W	0h	This register provides a means to access an internal EIP-39 register that stores alignment data bytes during the AAD phase of AES-CCM processing. This register needs to be read and stored when an AES-CCM operation is interrupted during the AAD phase. This value needs to be restored by writing this register, when resuming that AES-CCM operation in a later session.

62 BLK_CNT0 Register (Offset = 11D8h) [Reset = 00000000h]

BLK_CNT0 is shown in Figure 30-72 and described in Table 30-70.

Return to the Summary Table.

Internal block counter (LSW). This register along with BLK_CNT1 register provides access to the internal data block counter of the EIP-39. This counter keeps track of the number of data blocks during AES-CCM and AES-GCM operations. Reading and writing this counter allows interruption and resuming of long CCM or GCM operations. Note that internally, the block counter is used for AAD data as well as encryption/decryption data. Interruption and resuming is only supported in the encryption/decryption data phase and not during AAD.

Figure 30-72. BLK_CNT0 Register

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

Table 30-132. BLK_CNT0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DATA	R/W	0h	Internal block counter for AES GCM and CCM operations. These bits read the block count value that represents the number of blocks to go. This value is valid with saved_context_ready after a request for an intermediate GCM/CCM digest. Writing these registers will restore the internal block counter to the programmed value. This only needs to be done to prepare the engine to continue processing of an interrupted GCM or CCM operation. Also refer to the get_digest and gcm_ccm_continue bits in AES_CTRL register.

63 BLK_CNT1 Register (Offset = 11DCh) [Reset = 00000000h]

BLK_CNT1 is shown in Figure 30-73 and described in Table 30-71.

Return to the Summary Table.

Internal block counter (MSW). This register along with BLK_CNT0 register provides access to the internal data block counter of the EIP-39. This counter keeps track of the number of data blocks during AES-CCM and AES-GCM operations. Reading and writing this counter allows interruption and resuming of long CCM or GCM operations. Note that internally, the block counter is used for AAD data as well as encryption/decryption data. Interruption and resuming is only supported in the encryption/decryption data phase and not during AAD.

Figure 30-73. BLK_CNT1 Register

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

Table 30-134. BLK_CNT1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	RESERVED	R/W	0h	
23-0	DATA	R/W	0h	Internal block counter for AES GCM and CCM operations. These bits read the block count value that represents the number of blocks to go. This value is valid with saved_context_ready after a request for an intermediate GCM/CCM digest. Writing these registers will restore the internal block counter to the programmed value. This only needs to be done to prepare the engine to continue processing of an interrupted GCM or CCM operation. Also refer to the get_digest and gcm_ccm_continue bits in AES_CTRL register.

64 DMA_HS Register (Offset = 11F4h) [Reset = 00000000h]

DMA_HS is shown in Figure 30-74 and described in Table 30-72.

Return to the Summary Table.

Control register for DMA handshaking

Figure 30-74. DMA_HS Register

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

Table 30-136. DMA_HS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R/W	0h	
0	DMA_DATA_ACK	R/W	0h	When this bit is 0b, input and output data acknowledge is I/O register based, as specified in the description of the AES_DATA_IN_n / AES_DATA_OUT_n registers. When this bit is 1b, input and ouput data acknowledge is based on DMA handshake signals. 0h = Disable DMA based data handshake 1h = Enables DMA based handshake

ADVANCE INFORMATION

This chapter describes the Keystore controller.

31.1 Keystore

31.1.1 Overview

The Keystore controller provides secure management of the Advanced Encryption Engine (AES) keys. The use model of the keystore controller is to securely deposit keys into it during the execution of customer secure code and to have the AES engine access them subsequently in a secure manner without leaking any key data to observers. Both 128 and 256-bit keys can be stored in the keystore key slots. The keystore and its interaction with the AES engine are designed for secure operation including thwarting partial key modification attacks.

31.1.2 Detailed Description

The keystore supports storage of AES keys. Up to four 128-bit key slots are provisioned and can be used in different software-managed configurations, such as:

- Four 128-bit keys
- Two 256-bit keys
- One 256-bit key and two 128-bit keys

Note

Refer to the device-specific data sheet to determine the number of AES keys supported.

This configuration selection is the first to be performed by customer secure code before depositing keys into slots. Configuration is done by writing the number of 256-bit keys into the NK256 field of the CFG register.

After configuration, the key bytes can be written to selected slots in the keystore controller register. This is done by programming the KEYSZSEL (key size selection) and KEYSLOTSEL (key slot selection) fields of the KEYWR register. Note that the hardware expects any 256-bit keys to be store in lower-numbered slots. For example, if the store is configured for one 256-bit key and two 128-bit keys, the 256-bit key must be stored in slots 0 and 1. The IP reports an error (invalid configuration) if any other slot is used for the 256-bit key. If the store is configured for two 256-bit keys, then key 0 occupies slots 0 and 1 and key 1 occupies slots 2 and 3.

Depositing the key bytes into slots is done by writing them to the KEYIN register. Once a key write transaction is initiated, until and unless all the key bytes are written (4 words for a 128-bit key and 8 words for a 256-bit key), the key is not considered valid. Validity of slots is presented via a status register for the application. When customer secure code has deposited the required number of bytes, the key becomes valid and can be used by the application. The STATUS register provides status and validity fields to indicate status of the keystore controller as well as the validity of key slots.

Note that key configuration operations are permitted only while customer secure code is executing. Subsequently, the keystore configuration is locked and can not be modified. The signaling of the end of customer secure code is discussed in the Security Architecture chapter.

At run-time (after the end of customer secure code execution), the application is able to use one of the valid keys for an AES encryption/decryption operation. To configure the AES engine to use a specific key, the application references a key slot and initiates a transfer of the key data into the AES engine via a secure internal bus. This is performed by writing to the KEYSZSEL (key size selection), and KEYSLOTSEL (Key slot selection) fields of the KEYRD register. The selected key data is securely transferred to the AES engine over a private bus that is not accessible by software or the debugger.

The keystore holds the state and key data until a subsequent boot reset. At boot reset, customer startup code configures the keystore. If one or more slots are not used at the end of customer startup code execution, those slots remain unusable for the rest of the application execution.

31.2 KEYSTORE Registers

This Section describes the KEYSTORE Registers.

31.2.1 KEYSTORE Base Address Table

Table 31-1. KEYSTORE Base Address Table

Bit Field Name		DriverLib Name	Base Address
Instance	Structure		
KeystoreRegs	KEYSTORE_REGS	KEYSTORE	0x401B_6000

31.2.2 KEYSTORE_REGS Registers

Table 31-2 lists the memory-mapped registers for the KEYSTORE_REGS registers. All register offset addresses not listed in Table 31-2 should be considered as reserved locations and the register contents should not be modified.

Table 31-2. KEYSTORE_REGS Registers

Offset	Acronym	Register Name	Write Protection	Section
1000h	REVISION	IP revision id register		Go
1010h	SECURITY_ERR_FLAG	Security error flag		Go
1014h	SECURITY_ERR_CLR	Security error clear		Go
1018h	SECURITY_ERR_MSTID	Security error master id		Go
101Ch	SECURITY_ERR_ADDR	Security error address		Go
1030h	FRI_SEC_FLAG	FRI SEC flag		Go
1034h	FRI_SEC_CLR	FRI SEC clear		Go
1038h	FRI_SEC_ADDR	FRI SEC address		Go
103Ch	FRI_SEC_MSTID	FRI SEC master id		Go
1050h	FRI_DED_FLAG	FRI DED flag		Go
1054h	FRI_DED_CLR	FRI DED clear		Go
1058h	FRI_DED_ADDR	FRI DED address		Go
105Ch	FRI_DED_MSTID	FRI DED master id		Go
10B0h	SYSMEM_PAR_FLAG	SYSMEM DED flag		Go
10B4h	SYSMEM_PAR_CLR	SYSMEM DED clear		Go
10B8h	SYSMEM_PAR_ADDR	SYSMEM DED address		Go
10BCh	SYSMEM_PAR_MSTID	SYSMEM DED master id		Go
10D0h	TMUROM_PAR_FLAG	SYSMEM DED flag		Go
10D4h	TMUROM_PAR_CLR	SYSMEM DED clear		Go
10D8h	TMUROM_PAR_ADDR	SYSMEM DED address		Go
10DCh	TMUROM_PAR_TYPE	SYSMEM DED master id		Go

Complex bit access types are encoded to fit into small table cells. Table 31-3 shows the codes that are used for access types in this section.

Table 31-3. KEYSTORE_REGS Access Type Codes

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

ADVANCE INFORMATION

Table 31-3. KEYSTORE_REGS Access Type Codes (continued)

Access Type	Code	Description
y		When this variable is used in a register name, an offset, or an address it refers to the value of a register array.

1 REVISION Register (Offset = 1000h) [Reset = 00000000h]

REVISION is shown in Figure 31-1 and described in Table 31-4.

Return to the Summary Table.

IP revision id register

Figure 31-1. REVISION Register

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

Table 31-5. REVISION Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R-0	0h	Reserved
0	REVISION	R	0h	Revision register Reset type: PORESETn

ADVANCE INFORMATION

2 SECURITY_ERR_FLAG Register (Offset = 1010h) [Reset = 00000000h]

SECURITY_ERR_FLAG is shown in Figure 31-2 and described in Table 31-5.

Return to the Summary Table.

Security error flag

Figure 31-2. SECURITY_ERR_FLAG Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED				FLSEM_ACCESS_ERROR	FLC_MMR_ACCESS_ERROR	FPI_ILLSIZE	FPI_ILLCMD
R-0-0h				R-0h	R-0h	R-0h	R-0h
7	6	5	4	3	2	1	0
FPI_ILLMODECH	FPI_ILLRDVER	FPI_ILLERASE	FPI_ILLPROG	FPI_ILLADDR	HDP	PRIV	RESERVED
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

Table 31-7. SECURITY_ERR_FLAG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-12	RESERVED	R-0	0h	Reserved
11	FLSEM_ACCESS_ERROR	R	0h	FLSEM ACCESS ERROR flag Reset type: PORESETn
10	FLC_MMR_ACCESS_ERROR	R	0h	FLC MMR ACCESS ERROR flag Reset type: PORESETn
9	FPI_ILLSIZE	R	0h	FPI ILLSIZE flag Reset type: PORESETn
8	FPI_ILLCMD	R	0h	FPI ILLCMD flag Reset type: PORESETn
7	FPI_ILLMODECH	R	0h	FPI ILLMODECH flag Reset type: PORESETn
6	FPI_ILLRDVER	R	0h	FPI ILLRDVER flag Reset type: PORESETn
5	FPI_ILLERASE	R	0h	FPI ILLERASE flag Reset type: PORESETn
4	FPI_ILLPROG	R	0h	FPI ILLPROG flag Reset type: PORESETn
3	FPI_ILLADDR	R	0h	FPI ILLADDR flag Reset type: PORESETn

Table 31-7. SECURITY_ERR_FLAG Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
2	HDP	R	0h	Hide protection error flag Reset type: PORESETn
1	PRIV	R	0h	Privilege error flag Reset type: PORESETn
0	RESERVED	R	0h	Reserved

3 SECURITY_ERR_CLR Register (Offset = 1014h) [Reset = 00000000h]

SECURITY_ERR_CLR is shown in Figure 31-3 and described in Table 31-6.

Return to the Summary Table.

Security error clear

Figure 31-3. SECURITY_ERR_CLR Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED				FLSEM_ACCESS_ERROR	FLC_MMR_ACCESS_ERROR	FPI_ILLSIZE	FPI_ILLCMD
R-0-0h				R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h
7	6	5	4	3	2	1	0
FPI_ILLMODECH	FPI_ILLRDVER	FPI_ILLERASE	FPI_ILLPROG	FPI_ILLADDR	HDP	PRIV	RESERVED
R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h	R-0/W1C-0h

Table 31-9. SECURITY_ERR_CLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-12	RESERVED	R-0	0h	Reserved
11	FLSEM_ACCESS_ERROR	R-0/W1C	0h	writing '1' will clear SECURITY_ERR_FLAG[FLSEM_ACCESS_ERROR] register. Reset type: PORESETn
10	FLC_MMR_ACCESS_ERROR	R-0/W1C	0h	writing '1' will clear SECURITY_ERR_FLAG[FLC_MMR_ACCESS_ERROR] register. Reset type: PORESETn
9	FPI_ILLSIZE	R-0/W1C	0h	writing '1' will clear SECURITY_ERR_FLAG[FPI_ILLSIZE] register. Reset type: PORESETn
8	FPI_ILLCMD	R-0/W1C	0h	writing '1' will clear SECURITY_ERR_FLAG[FPI_ILLCMD] register. Reset type: PORESETn
7	FPI_ILLMODECH	R-0/W1C	0h	writing '1' will clear SECURITY_ERR_FLAG[FPI_ILLMODECH] register. Reset type: PORESETn
6	FPI_ILLRDVER	R-0/W1C	0h	writing '1' will clear SECURITY_ERR_FLAG[FPI_ILLRDVER] register. Reset type: PORESETn
5	FPI_ILLERASE	R-0/W1C	0h	writing '1' will clear SECURITY_ERR_FLAG[FPI_ILLERASE] register. Reset type: PORESETn
4	FPI_ILLPROG	R-0/W1C	0h	writing '1' will clear SECURITY_ERR_FLAG[FPI_ILLPROG] register. Reset type: PORESETn

Table 31-9. SECURITY_ERR_CLR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3	FPI_ILLADDR	R-0/W1C	0h	writing '1' will clear SECURITY_ERR_FLAG[FPI_ILLADDR] register. Reset type: PORESETn
2	HDP	R-0/W1C	0h	writing '1' will clear SECURITY_ERR_FLAG[HDP] register. Reset type: PORESETn
1	PRIV	R-0/W1C	0h	writing '1' will clear SECURITY_ERR_FLAG[PRIV] register. Reset type: PORESETn
0	RESERVED	R-0/W1C	0h	Reserved

4 SECURITY_ERR_MSTID Register (Offset = 1018h) [Reset = 00000000h]

SECURITY_ERR_MSTID is shown in Figure 31-4 and described in Table 31-7.

Return to the Summary Table.

Security error master id

Figure 31-4. SECURITY_ERR_MSTID Register

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

Table 31-11. SECURITY_ERR_MSTID Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R-0	0h	Reserved
7-0	MSTID	R	0h	Security error master ID. This field will be cleared along with the corresponding flag clear. Reset type: PORESETn

5 SECURITY_ERR_ADDR Register (Offset = 101Ch) [Reset = 00000000h]

SECURITY_ERR_ADDR is shown in Figure 31-5 and described in Table 31-8.

Return to the Summary Table.

Security error address

Figure 31-5. SECURITY_ERR_ADDR Register

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

Table 31-13. SECURITY_ERR_ADDR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ADDRESS	R	0h	Security error address. This field will be cleared along with the corresponding flag clear. Reset type: PORESETn

6 FRI_SEC_FLAG Register (Offset = 1030h) [Reset = 00000000h]

FRI_SEC_FLAG is shown in Figure 31-6 and described in Table 31-9.

Return to the Summary Table.

FRI single error correction flag

Figure 31-6. FRI_SEC_FLAG Register

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

Table 31-15. FRI_SEC_FLAG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R-0	0h	Reserved
0	SEC	R	0h	Single error correction error Reset type: PORESETn

ADVANCE INFORMATION

7 FRI_SEC_CLR Register (Offset = 1034h) [Reset = 00000000h]

FRI_SEC_CLR is shown in Figure 31-7 and described in Table 31-10.

Return to the Summary Table.

FRI single error correction clear

Figure 31-7. FRI_SEC_CLR Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
R-0-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED															SEC
R-0-0h															R-0/W1C-0h

Table 31-17. FRI_SEC_CLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R-0	0h	Reserved
0	SEC	R-0/W1C	0h	writing '1' will clear FRI_SEC_FLAG[SEC] register. Reset type: PORESETn

ADVANCE INFORMATION

8 FRI_SEC_ADDR Register (Offset = 1038h) [Reset = 00000000h]

FRI_SEC_ADDR is shown in Figure 31-8 and described in Table 31-11.

Return to the Summary Table.

FRI single error correction address

Figure 31-8. FRI_SEC_ADDR Register

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

Table 31-19. FRI_SEC_ADDR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ADDRESS	R	0h	Single error correction error address. This field will be cleared along with the corresponding flag clear. Reset type: PORESETn

9 FRI_SEC_MSTID Register (Offset = 103Ch) [Reset = 00000000h]

FRI_SEC_MSTID is shown in Figure 31-9 and described in Table 31-12.

Return to the Summary Table.

FRI single error correction master id

Figure 31-9. FRI_SEC_MSTID Register

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

Table 31-21. FRI_SEC_MSTID Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R-0	0h	Reserved
7-0	MSTID	R	0h	Single error correction error master ID. This field will be cleared along with the corresponding flag clear. Reset type: PORESETn

ADVANCE INFORMATION

10 FRI_DED_FLAG Register (Offset = 1050h) [Reset = 00000000h]

FRI_DED_FLAG is shown in Figure 31-10 and described in Table 31-13.

Return to the Summary Table.

FRI double error detection flag

Figure 31-10. FRI_DED_FLAG Register

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

Table 31-23. FRI_DED_FLAG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R-0	0h	Reserved
1	DIAG_DED	R	0h	diagnostic Double error detect error Reset type: PORESETn
0	DED	R	0h	Double error detect error Reset type: PORESETn

ADVANCE INFORMATION

11 FRI_DED_CLR Register (Offset = 1054h) [Reset = 00000000h]

FRI_DED_CLR is shown in Figure 31-11 and described in Table 31-14.

Return to the Summary Table.

FRI double error detection clear

Figure 31-11. FRI_DED_CLR Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED						DIAG_DED	SEC
R-0-0h						R-0/W1C-0h	R-0/W1C-0h

Table 31-25. FRI_DED_CLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R-0	0h	Reserved
1	DIAG_DED	R-0/W1C	0h	writing '1' will clear FRI_DED_FLAG[DIAG_DED] register. Reset type: PORESETn
0	SEC	R-0/W1C	0h	writing '1' will clear FRI_DED_FLAG[DED] register. Reset type: PORESETn

ADVANCE INFORMATION

12 FRI_DED_ADDR Register (Offset = 1058h) [Reset = 00000000h]

FRI_DED_ADDR is shown in Figure 31-12 and described in Table 31-15.

Return to the Summary Table.

FRI double error detection address

Figure 31-12. FRI_DED_ADDR Register

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

Table 31-27. FRI_DED_ADDR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ADDRESS	R	0h	Single error correction error address. This field will be cleared along with the corresponding flag clear. Reset type: PORESETn

ADVANCE INFORMATION

13 FRI_DED_MSTID Register (Offset = 105Ch) [Reset = 00000000h]

FRI_DED_MSTID is shown in Figure 31-13 and described in Table 31-16.

Return to the Summary Table.

FRI double error detection master id

Figure 31-13. FRI_DED_MSTID Register

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

Table 31-29. FRI_DED_MSTID Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R-0	0h	Reserved
7-0	MSTID	R	0h	Single error correction error master ID. This field will be cleared along with the corresponding flag clear. Reset type: PORESETn

14 SYSMEM_PAR_FLAG Register (Offset = 10B0h) [Reset = 00000000h]

SYSMEM_PAR_FLAG is shown in Figure 31-14 and described in Table 31-17.

Return to the Summary Table.

SYSMEM parity error flag

Figure 31-14. SYSMEM_PAR_FLAG Register

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

Table 31-31. SYSMEM_PAR_FLAG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R-0	0h	Reserved
1	WR_PAR	R	0h	Parity error during write Reset type: PORESETn
0	RD_PAR	R	0h	Parity error during read Reset type: PORESETn

ADVANCE INFORMATION

15 SYSMEM_PAR_CLR Register (Offset = 10B4h) [Reset = 00000000h]

SYSMEM_PAR_CLR is shown in Figure 31-15 and described in Table 31-18.

Return to the Summary Table.

SYSMEM parity error clear

Figure 31-15. SYSMEM_PAR_CLR Register

31	30	29	28	27	26	25	24
RESERVED							
R-0-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0-0h							
7	6	5	4	3	2	1	0
RESERVED						WR_PAR	RD_PAR
R-0-0h						R-0/W1C-0h	R-0/W1C-0h

Table 31-33. SYSMEM_PAR_CLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R-0	0h	Reserved
1	WR_PAR	R-0/W1C	0h	writing '1' will clear SYSMEM_PAR_FLAG[WR_PAR] register. Reset type: PORESETn
0	RD_PAR	R-0/W1C	0h	writing '1' will clear SYSMEM_PAR_FLAG[RD_PAR] register. Reset type: PORESETn

ADVANCE INFORMATION

16 SYSMEM_PAR_ADDR Register (Offset = 10B8h) [Reset = 00000000h]

SYSMEM_PAR_ADDR is shown in Figure 31-16 and described in Table 31-19.

Return to the Summary Table.

SYSMEM parity error address

Figure 31-16. SYSMEM_PAR_ADDR Register

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

Table 31-35. SYSMEM_PAR_ADDR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ADDRESS	R	0h	Parity error address. This field will be cleared along with the corresponding flag clear. Reset type: PORESETn

17 SYSMEM_PAR_MSTID Register (Offset = 10BCh) [Reset = 00000000h]

SYSMEM_PAR_MSTID is shown in Figure 31-17 and described in Table 31-20.

Return to the Summary Table.

SYSMEM parity error master id

Figure 31-17. SYSMEM_PAR_MSTID Register

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

Table 31-37. SYSMEM_PAR_MSTID Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R-0	0h	Reserved
7-0	MSTID	R	0h	Parity error master ID. This field will be cleared along with the corresponding flag clear. Reset type: PORESETn

18 TMUROM_PAR_FLAG Register (Offset = 10D0h) [Reset = 00000000h]

TMUROM_PAR_FLAG is shown in Figure 31-18 and described in Table 31-21.

Return to the Summary Table.

TMUROM parity error flag

Figure 31-18. TMUROM_PAR_FLAG Register

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

Table 31-39. TMUROM_PAR_FLAG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R-0	0h	Reserved
0	PAR	R	0h	Parity error Reset type: PORESETn

19 TMUROM_PAR_CLR Register (Offset = 10D4h) [Reset = 00000000h]

TMUROM_PAR_CLR is shown in Figure 31-19 and described in Table 31-22.

Return to the Summary Table.

TMUROM parity error clear

Figure 31-19. TMUROM_PAR_CLR Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
R-0-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED															PAR
R-0-0h															R-0/W1C-0h

Table 31-41. TMUROM_PAR_CLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R-0	0h	Reserved
0	PAR	R-0/W1C	0h	writing '1' will clear TMUROM_PAR_FLAG[PAR] register. Reset type: PORESETn

20 TMUROM_PAR_ADDR Register (Offset = 10D8h) [Reset = 00000000h]

TMUROM_PAR_ADDR is shown in Figure 31-20 and described in Table 31-23.

Return to the Summary Table.

TMUROM parity error address

Figure 31-20. TMUROM_PAR_ADDR Register

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

Table 31-43. TMUROM_PAR_ADDR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-15	RESERVED	R-0	0h	Reserved
14-0	ADDRESS	R	0h	Parity error address. This field will be cleared along with the corresponding flag clear. Reset type: PORESETn

21 TMUROM_PAR_TYPE Register (Offset = 10DCh) [Reset = 00000000h]

TMUROM_PAR_TYPE is shown in Figure 31-21 and described in Table 31-24.

Return to the Summary Table.

TMUROM parity error type

Figure 31-21. TMUROM_PAR_TYPE Register

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

Table 31-45. TMUROM_PAR_TYPE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R-0	0h	Reserved
2-0	TYPE	R	0h	Parity error type. This field will be cleared along with the corresponding flag clear. [0] - Y0i ROM table read has parity error [1] - S1i ROM table read has parity error [2] - S2i ROM table read has parity error Reset type: PORESETn

32.1 Timers (TIMx)

The timer module (TIMx) is a general purpose timer counting module with multiple compare/capture blocks. The timers include many common features that can be used for a variety of functions such as measuring the input signal edge and period (capture mode) or generating output waveforms like PWMs (compare mode output).

32.1.1 TIMx Overview

The timer module (TIMx) is a timer counting module with multiple compare/capture blocks. There are various types of timer instances available within the module defined by the values G0 to G14 that support different features. All timers use a common timer architecture, which allows for easy migration between timer instances with common functions. This minimizes the need to write extra software for timer-based applications and allows for easy porting and maintenance between TIMx instances.

Note

See Section 32.1.1.1 to determine the common features available between TIMx instances.

In this section:

- “TIMx” indicates a feature common across all timer instances.
- "TIMGx" indicates a feature available only on the specified TIMGx instances.

Note

The 'x' value in the TIMx and TIMGx nomenclature is separate from the 'xy' values used within the register naming within this chapter. The 'xy' value is used to represent the register arrays that exist in the timer module to group registers with same bit fields for different capture/compare ports. For example, TIMx.CC_01[0/1] is a register array that contains the capture/compare registers for both CCP0 and CCP1. Access TIMx.CC_01[0] to read the CC0 capture/compare value, and access TIMx.CC_01[1] to read the CC1 capture/compare value.

The TIMx module consists of 16-bit and 32-bit auto reload counters driven by a programmable prescaler with up to 4 capture/compare (CC) blocks and capture/compare pins (CCP) to enable multiple capture/compares, PWM outputs, and interval timing. TIMx also has extensive event generation capabilities, including counter overflow, reload, and capture/compare actions for a variety of use cases.

32.1.1.1 TIMx Instance Configuration

Table 32-1 shows the TIMx instance configuration for all TIMER instances.

Table 32-1. TIMx Instance Configuration

Instance	Counter Resolution	Prescaler	CCP Channels (External/Internal)	External PWM Channels	Shadow Load	Shadow CCs
TIMG4 (1x)	16-bit	8-bit	2	2	Yes	Yes
TIMG12 (1x)	32-bit	-	2	2	-	Yes

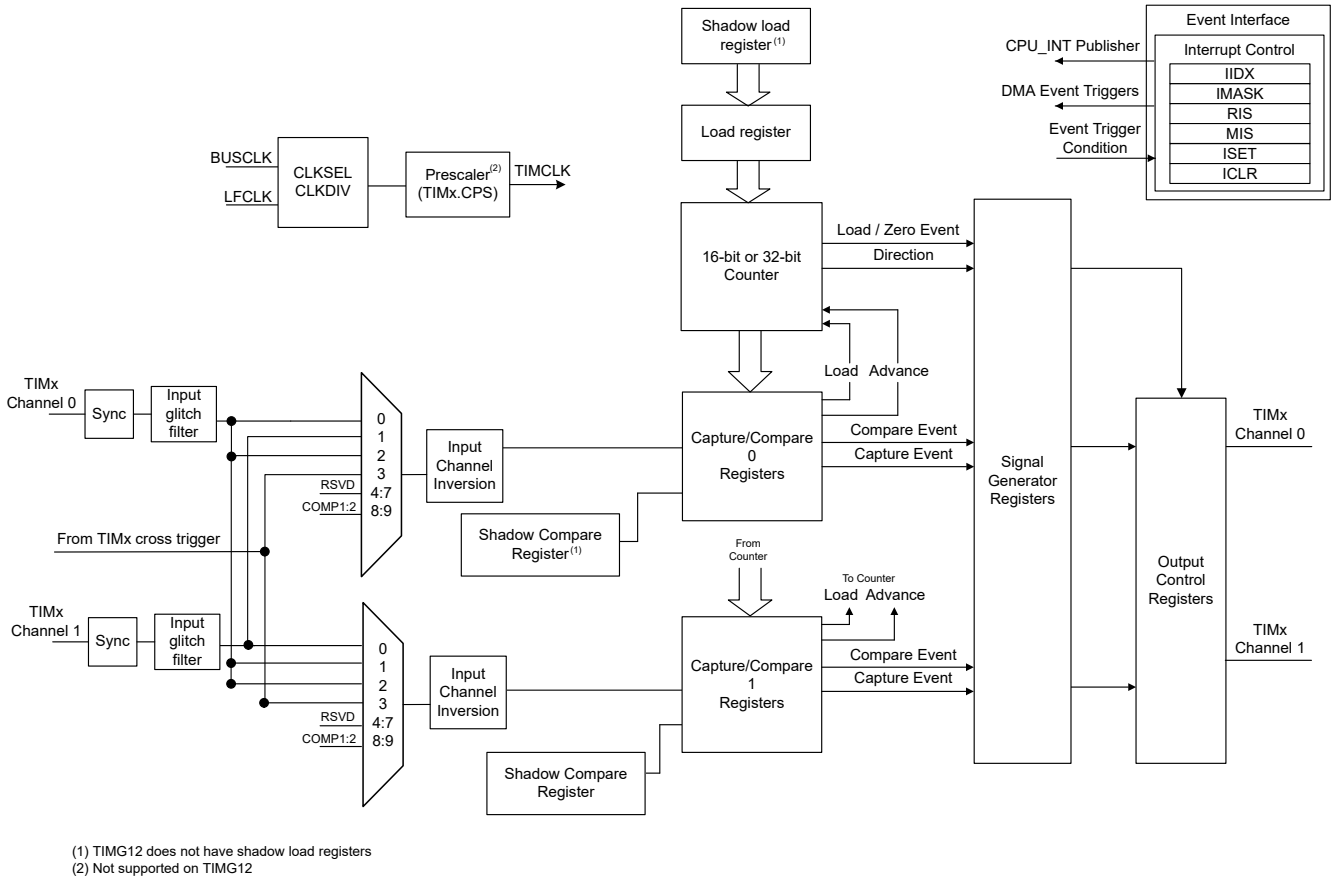
32.1.1.2 TIMG Features

Specific features for TIMG include:

- 16-bit up, down, or up-down counter, with repeat-reload mode
- 8-bit programmable prescaler to divide the counter clock frequency
- Up to four independent channels for
  - Output compare
  - Input capture
  - PWM output (Edge-Aligned and Center-Aligned)
  - One-shot mode
- Selectable and configurable clock source
- Shadow register mode for load and compare values (see Section 32.1.2.4)
- Support synchronization and cross trigger among different TIMx instances in the same power domain (see Section 32.1.2.6)
- Support CPU interrupt generation and cross peripherals (such as ADC, DAC, etc.) using the Event (see Section 32.1.2.8)

32.1.1.3 Functional Block Diagram

TIMGx Functional Block Diagram shows the TIMx block diagram for AM13E23x.



(1) TIMG12 does not have shadow load registers
(2) Not supported on TIMG12

Figure 32-1. TIMGx Functional Block Diagram

32.1.2 TIMx Operation

The TIMx module is configured with user software. The setup and operation of TIMx is discussed in the following sections.

Note

There are register arrays in the timer module to group registers with same bit fields for different capture/compare ports. For example, TIMx.CC_01[0/1] is a register array that contains the capture/compare registers for both CCP0 and CCP1. Access TIMx.CC_01[0] to read the CC0 capture/compare value, and access TIMx.CC_01[1] to read the CC1 capture/compare value. This chapter uses 'xy' in place of the register array numbers to cover all register arrays (TIMx.CC01[0/1] becomes TIMx.CC_xy[0/1]).

32.1.2.1 Timer Counter

All TIMx instances have 16-bit counter blocks except for TIMG12 and TIMG13, which have 32-bit counter blocks. The timer counter register (TIMx.CTR) can count down, up-and-down, or up depending on the operation mode. It can also be read or written with software.

The counter can advance (increment or decrement) using the internal clock TIMCLK, with the rising and/or falling edge of an timer external input, or an internal trigger event from other peripherals (see advance condition). By default, each count occurs with each rising edge of the TIMCLK signal.

Enabling the TIMx Counter

ADVANCE INFORMATION

The counter is clocked by the prescaler output TIMCLK. The counter enable bit TIMx.CTRCTL.EN can be enabled in two ways:

- Set in software manually
- After a load condition (LCOND) or zero condition (ZCOND) is met, and the counter value after enable bit (TIMx.CTRCTL.CVAE) is set to the load value or zero value, respectively.

Note

The ability to write the counter register while TIMx is running is possible but should be avoided because the user write can collide with a load, zero, or advance event. Depending on the prescaler ratio, the application cannot predict the timing of the write, which can affect the correct timer period.

32.1.2.1.1 Clock Source Select and Prescaler

The TIMx clock (TIMCLK) can be sourced from an internal clock or an external signal trigger to advance the clock.

32.1.2.1.1.1 Internal Clock and Prescaler

The TIMx clock (TIMCLK) can be sourced from BUSCLK and LFCLK by setting the TIMx.CLKSEL register. It can also be divided by a ratio by setting the TIMx.CLKDIV register and the prescaler (TIMx.CPS). The selected source clock is always available and the frequency depends on the power mode. For more information, see the Clock Module (CKM) section.

The TIMCLK can come from the following sources:

- BUSCLK: the current bus clock is selected as the source for TIMx. The current bus clock depends on power domain.
  - If the TIMx instance is in Power Domain 1 (PD1), refer to MCLK.
  - If the TIMx instance is in Power Domain 0 (PD0), refer to ULPCLK.
- LFCLK: LFCLK is selected as the source for TIMx, refer to LFCLK.

The selected clock source can be passed directly to TIMx or divided by the 8-bit prescaler. The prescaler setting can be configured with the TIMx.CPS.PCNT bit. The selected TIMCLK source is divided by a value of (PCNT+1). A PCNT value of 0 divides TIMCLK by 1, effectively bypassing the divider. A PCNT value of greater than 0 divides the TIMCLK source to generate a slower clock.

TIMx also has a software mechanism for disabling the timer clock. Set TIMx.CCLKCTL.CLKEN to 0 to put the timer in an IDLE state.

TIMCLK Configuration

To configure the clock source, divider, and prescaler:

1. Select the TIMCLK clock source (BUSCLK or LFCLK) using the CLKSEL register.
2. Optionally divide the TIMCLK using CLKDIV.RATIO.
3. Optionally set a prescaler using CPS.PCNT.
4. Enable the TIMCLK by setting CCLKCTL.CLKEN = 1.

The frequency of TIMCLK is determined using Equation 40.

$$f_{TIMCLK} = \frac{f_{CLK\_SOURCE}}{((CLKDIV.RATIO + 1)*(CPS.PCNT + 1))} \quad (40)$$

32.1.2.1.1.2 External Signal Trigger

The counter can also advance (increment or decrement) by using an external signal on the timer input capture/compare pin (CCP) or by triggering from an event from other peripherals in the system. This can be configured by using the advance condition setting (TIMx.CCCTL_xy[0/1].ACOND) to specify what creates the advance event. To specify what event advances the counter, use the TIMx.CTRCTL.CAC setting.

ADVANCE INFORMATION

32.1.2.2 Counting Mode Control

When the device is out of reset, TIMx is disabled. Writing 1 to the TIMx.CTRCTL.EN bit enables the counter. This bit is automatically cleared if TIMx.CTRCTL.REPEAT=0 (do not automatically reload the counter), and the counter value equals zero in down counting mode or the LOAD value in up counting mode.

TIMx has three counting modes when enabled: down, up/down, and up. The operating mode is selected by TIMx.CTRCTL.CM bit (shown in TIMx Counting Modes (CM)). After the counter is enabled, the timer will start counting from the value specified by the TIMx.CTRCTL.CVAE bitfield.

Table 32-2. TIMx Counting Modes (CM)

TIMx.CTRCTL.CM	Counting Mode
0	Down
1	Up/Down
2	Up

Table 32-3. TIMx Counter Value after Enable (CVAE)

Count Value After Enable (CVAE)	Description	Supported Counting Modes
0	LOAD value	Up, up/down, down
1	Unchanged from current Counter value	Up/down
2	Zero value	Up, up/down, down

32.1.2.2.1 One-shot and Periodic Modes

One-shot and Periodic Mode Behavior shows TIMx working in both one-shot mode (left) and periodic mode (right).

One-shot Mode

When TIMx.CTRCTL.REPEAT bit is set to 0, TIMx does not advance when:

- TIMx.CTR value reaches 0 in either down- or up/down-counting mode
- TIMx.CTR value reaches TIMx.LOAD in up-counting mode

Periodic Mode (Counter Reload)

When TIMx.CTRCTL.REPEAT is set to 1h, TIMx automatically repeats once a zero or load event occurs. This happens when:

- TIMx.CTR reaches 0 in either down- or up/down-counting mode
  - In down-counting mode, a zero event is followed by a load event at the next advance condition
  - In up/down-counting mode, a zero event is followed by an advance event (+1), restarting the counter.
- TIMx.CTR reaches TIMx.LOAD in up-counting mode
  - A load event is followed by a zero at the next advance condition

TIMx.CTRCTL.REPEAT can also be set to 3h for TIMx to repeat only when not in a debug condition. If there is a debug condition, TIMx will count to the zero event and repeat only once the debug condition is removed.

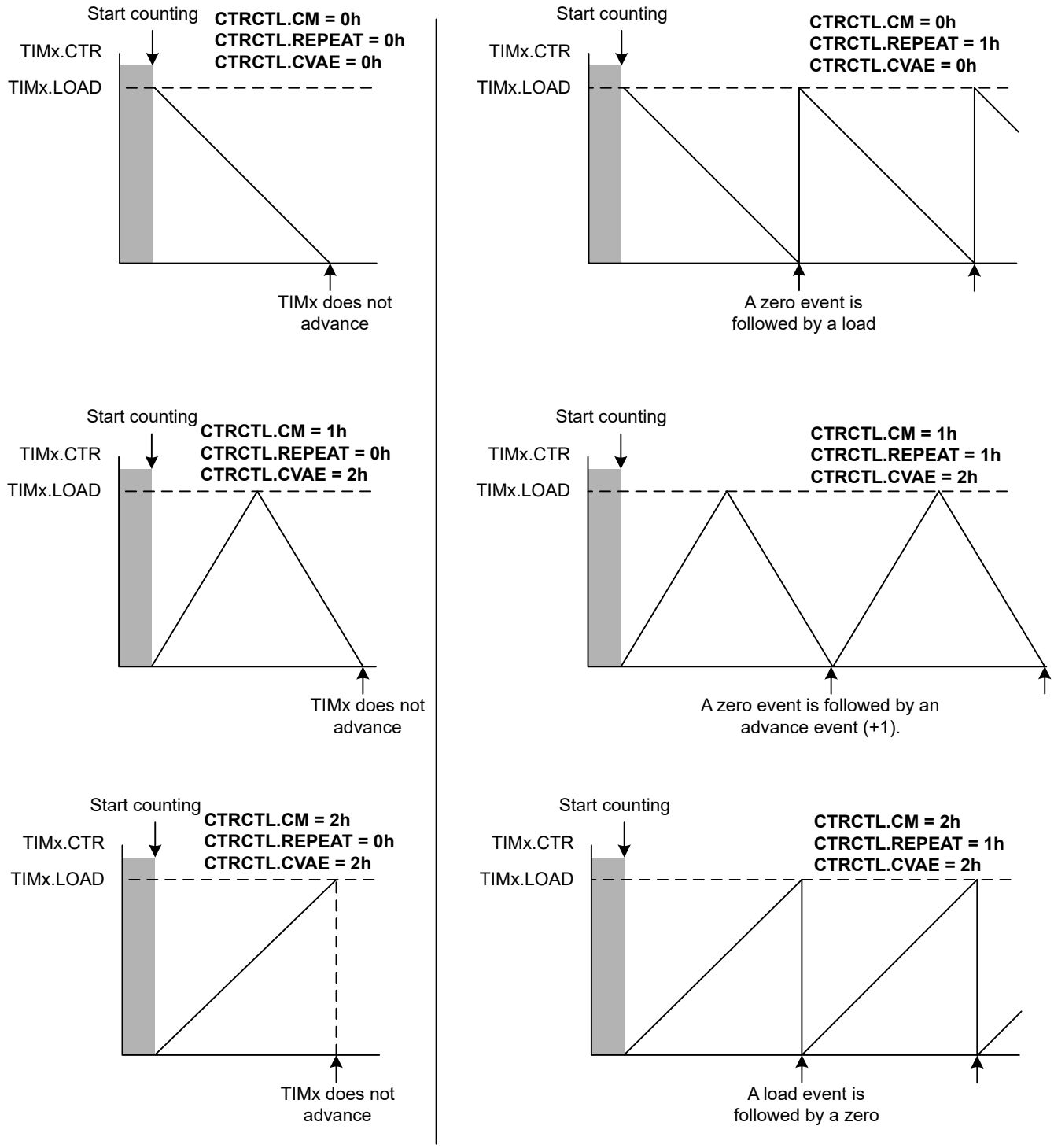


Figure 32-2. One-shot and Periodic Mode Behavior

32.1.2.2.2 Down Counting Mode

In down counting mode (CM = 0) when TIMx.CTRCTL.CVAE = 0, the TIMx.LOAD value is loaded into TIMx.CTR. Then TIMx counts from the LOAD value down to zero. When the TIMx.CTR value equals zero and TIMx.CTRCTL.REPEAT is set to 1 (periodic mode), the TIMx.LOAD value is once again loaded into TIMx.CTR and the timer repeats the down counting pattern as shown in Down Counting Mode, CVAE = 0.

A zero event is generated when TIMx counts to zero. A load event is generated when TIMx counts from zero to the TIMx.LOAD value. Down Counting Mode Event Generation shows the event generating cycle.

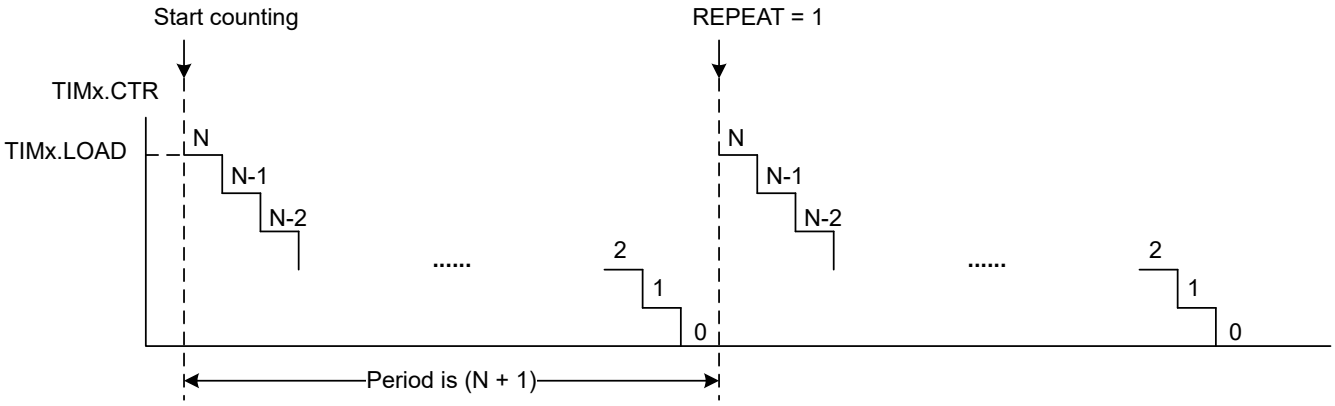


Figure 32-3. Down Counting Mode, CVAE = 0

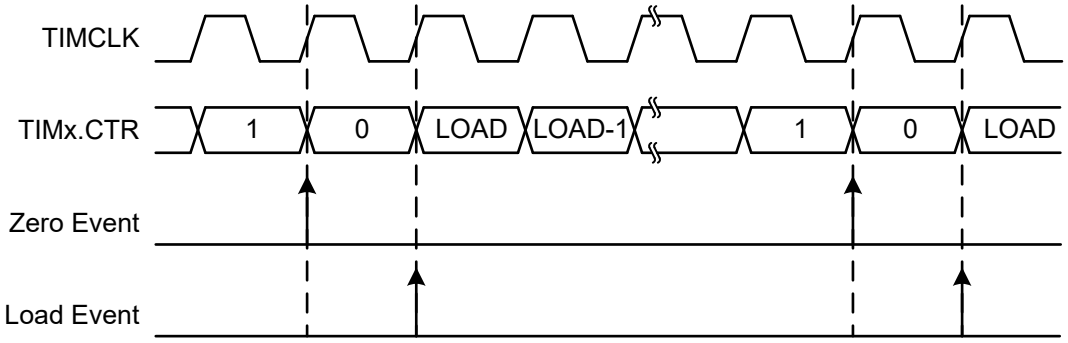


Figure 32-4. Down Counting Mode Event Generation

32.1.2.2.3 Up/Down Counting Mode

The Up/Down counting mode can count in an down-up direction or an up-down direction depending on TIMx.CTRCTL.CVAE value. The TIMx.CTRCTL.CVAE bits specify the initialization condition of the counter.

Table 32-4. Counter Value after Enable Initialization Conditions

TIMx.CTRCTL.CVAE Value	Counter Value After Enable
0x0	Load Value (down-up direction)
0x1	No Change
0x2	Zero (up-down direction)

Counting in down-up direction

When TIMx.CTRCTL.CVAE = 0, TIMx.CTR is set to TIMx.LOAD register value and TIMx counts in the down direction. When it reaches zero, a Zero event is generated and TIMx counts back up to TIMx.LOAD value. A Load event is generated when it reaches TIMx.LOAD value.

Down-up Counting Mode and Event Generation, CVAE = 0 shows TIMx counting in the down-up direction when TIMx.CTRCTL.CVAE = 0.

ADVANCE INFORMATION

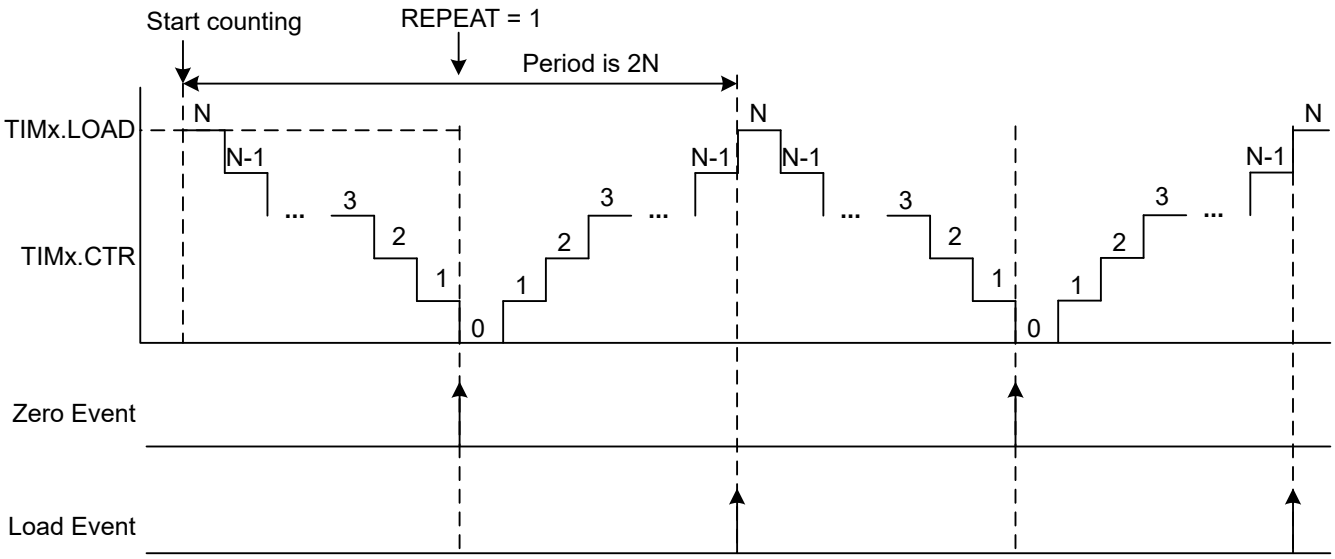


Figure 32-5. Down-up Counting Mode and Event Generation, CVAE = 0

Counting in up-down direction

When TIMx.CTRCTL.CVAE = 2, TIMx.CTR is set to zero and TIMx counts in the up direction. When it reaches TIMx.LOAD, a Load event is generated and TIMx counts back down to zero. A Zero event is generated when it reaches zero.

Up-down Counting Mode and Event Generation, CVAE = 2 shows TIMx counting in up-down direction when TIMx.CTRCTL.CVAE = 2.

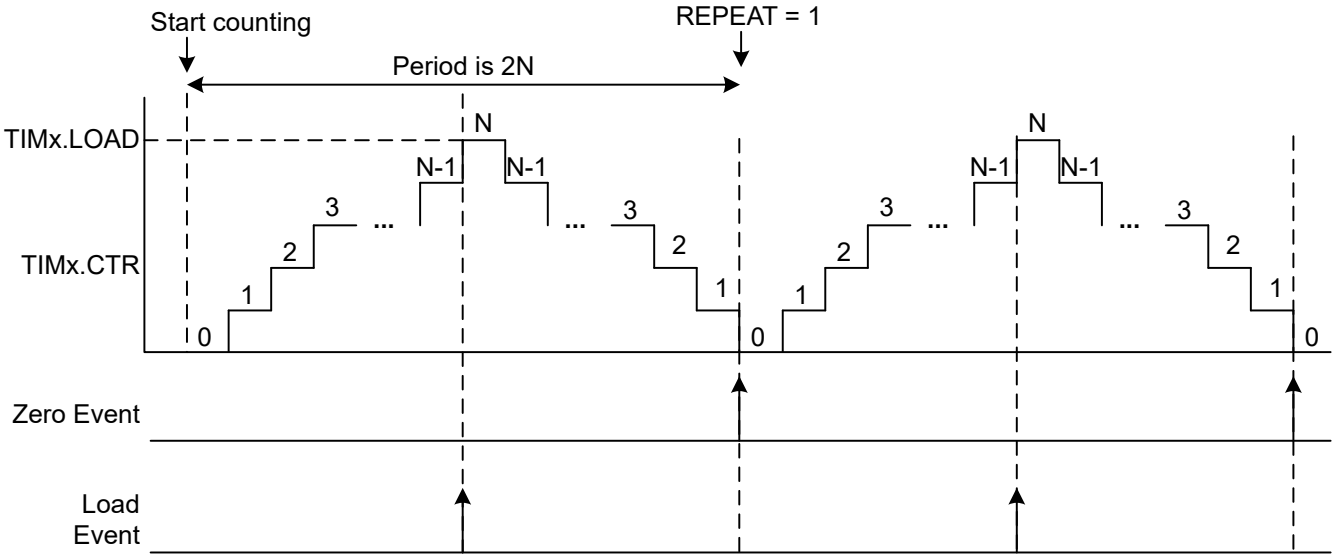


Figure 32-6. Up-down Counting Mode and Event Generation, CVAE = 2

32.1.2.2.4 Up Counting Mode

In up counting mode, TIMx counts from zero up to the value defined in TIMx.LOAD. When the TIMx.CTR value equals TIMx.LOAD and TIMx.CTRCTL.REPEAT is set to 1 (periodic mode), zero is loaded into TIMx.CTR and the timer repeats the up counting pattern as shown in Up Counting Mode, CVAE = 2.

A Load event is generated when TIMx counts to TIMx.LOAD. A Zero event is generated when TIMx counts from TIMx.LOAD to the zero value. Up Counting Mode Event Generation shows the event generating cycle.

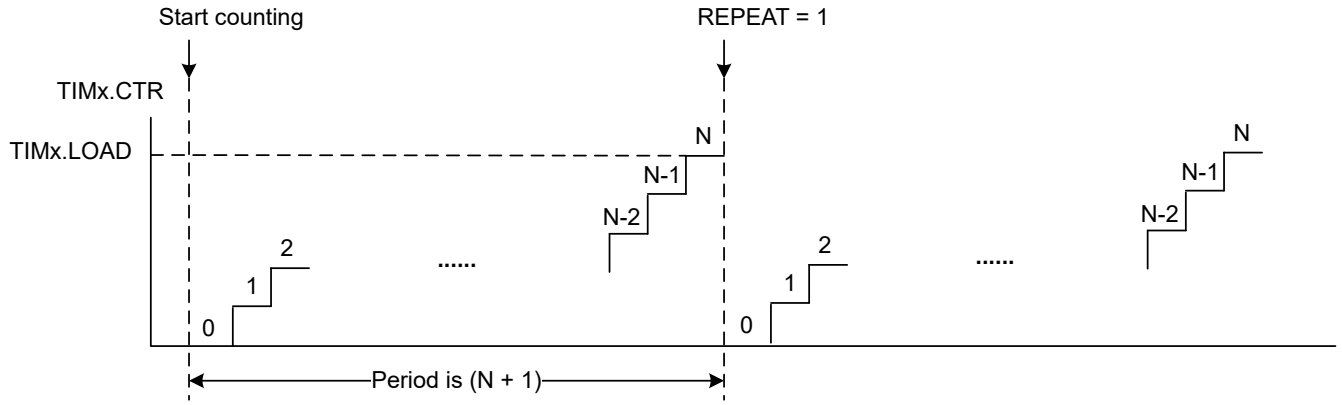


Figure 32-7. Up Counting Mode, CVAE = 2

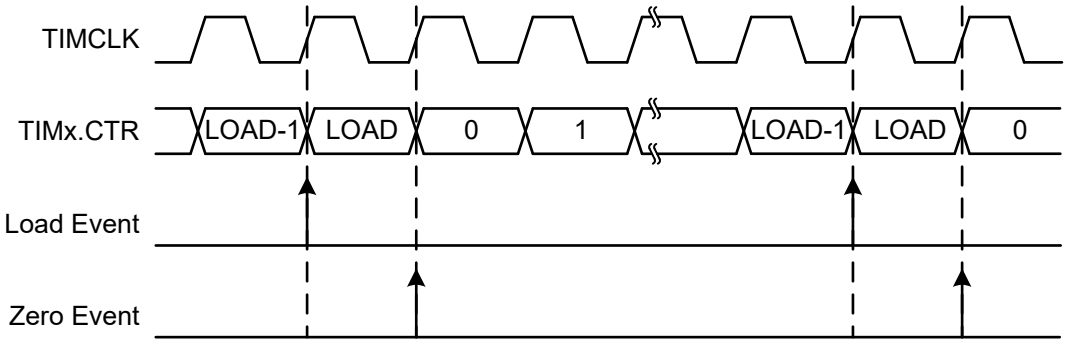


Figure 32-8. Up Counting Mode Event Generation

32.1.2.3 Capture/Compare Module

The capture/compare (CC) block is used for capture events or compare events. TIMG0 through TIMG13 have up to 2 identical capture/compare blocks and TIMG24 has up to 4 identical capture/compare blocks present to support external or internal signals. Any of the TIMx capture/compare blocks may be used to capture timings of an input signal or to generate time intervals.

Key registers for configuring capture/compare mode:

- **TIMx.LOAD**: the contents of this register are copied to counter (TIMx.CTR) on any operation designated to do a "load". This value is also used to compare with the counter value for generating a "Load Event" which can be used for interrupt, trigger, or signal generation actions.
- **TIMx.CC_xy[0/1]**: this is a register that can be used as either a capture register to acquire or record the next counter value on an event, or a compare register to the current counter to create an event.
- **TIMx.CCCTL_xy[0/1]**: this register controls the operations of the respective CC (capture/compare) blocks. In capture mode, it can configure whether a rising edge or falling edge generates a load, zero, advance, or capture condition. In compare mode, it controls which sources generate different types of compare events.
- **TIMx.CTRCTL**: this register provides control over the counter operation in different conditions.
- **TIMx.IFCTL_xy[0/1]**: this register controls the input filtering, input selection, and input inversion for the associated CC block.

Note

Capture and compare modes on a TIMx instance use the same registers. Therefore application cannot perform a capture and compare on the same channel at the same time.

ADVANCE INFORMATION

32.1.2.3.1 Capture Mode

Capture mode is selected when the TIMx.CCCTL_xy[0/1].COC bit is set to 1. Capture mode is used to generate capture events and record time intervals, which is useful for speed computation or time measurements.

32.1.2.3.1.1 Input Selection, Counter Conditions, and Inversion

The TIMx.IFCTL register is used for selecting input source, filtering, and final inversion options for the capture/compare block.

Figure 32-9 shows the block diagram for the TIMx capture block with two CC channels.

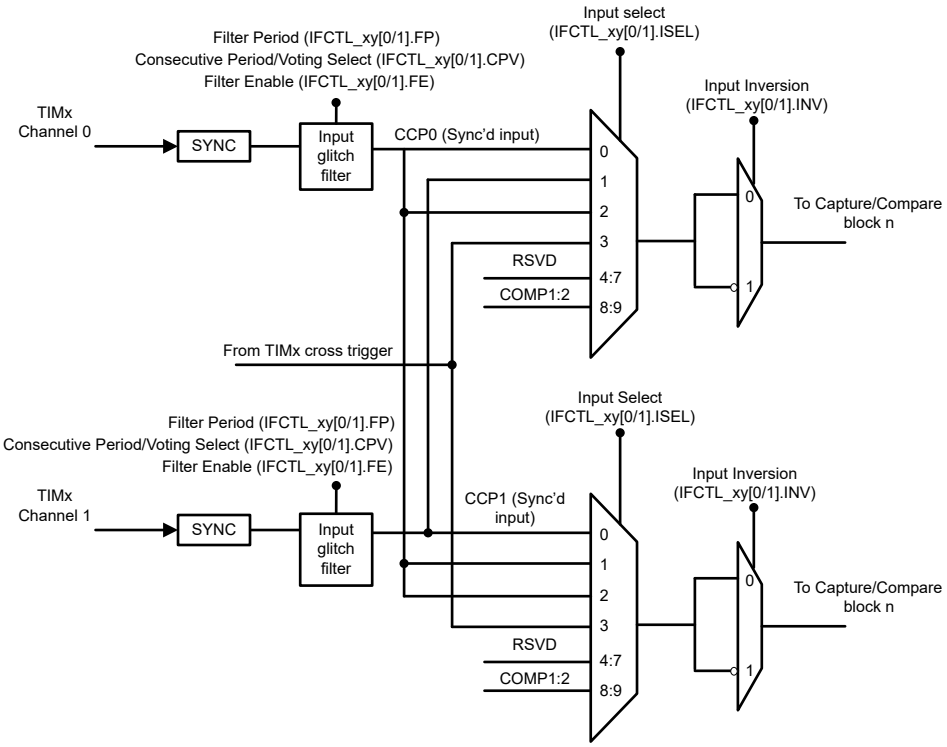


Figure 32-9. TIMx Capture Block Diagram

32.1.2.3.1.1.1 CCP Input Edge Synchronization

When using a capture/compare pin (CCP) as an input, configure the pin control management register (PINCMx) for the TIMx CCP input. See IOMUX for more information.

The CCP input signal is always passed through a synchronizer, and the input state (high or low) must be greater than one TIMCLK clock period for the synchronizer to detect the edge. CCP input edge detection requires at least one TIMCLK cycle (rise and fall) to synchronize the edge input. Timing in the first TIMCLK cycle is uncertain because the edge detection cannot be predicted in the first TIMCLK period.

When the capture condition occurs, an additional TIMCLK cycle is required to generate the capture event.

32.1.2.3.1.1.2 Input Selection

The input select bits TIMx.IFCTL_xy[0/1].ISEL select the input source to the filter input as the corresponding CCP input, the CCP pair of the current CCP input, or an external trigger.

Table 32-5. Input Selection Options for TIMx CC Instances

Input selection (TIMx.IFCTL_xy[0/1].ISEL)	Source
0h	TIMx CCP of the corresponding capture compare unit
1h	Input pair CCPx of the capture compare unit. [CCP0 for CCP1 and CCP1 for CCP0]

Table 32-5. Input Selection Options for TIMx CC Instances (continued)

Input selection (TIMx.IFCTL_xy[0/1].ISEL)	Source
2h	TIMx CCP0

32.1.2.3.1.1.3 CCP Input Filtering

The input glitch filter can be enabled by setting the TIMx. IFCTL_01[0/1].FE bit. The filter period is configured by setting the TIMx. IFCTL_01[0/1].FP bit.

A consecutive period or majority voting format selected by the TIMx.IFCTL_xy[0/1].CPV bit is used to select the criteria for a CCP input signal.

- **Consecutive period** - The CCP input signal must be at the specified level for the defined number of FP timer clocks for the CCP input to be processed.
- **Majority voting** - The filter ignores one clock of opposite logic over the filter period. For example, over the number of FP samples of the input, up to 1 sample can be of an opposite logic value (glitch) without affecting the output.

The example shown in Consecutive Period and Majority Voting for Input Glitch Filtering using FP = 0 (3 TIMCLK cycles) shows the difference between consecutive period and majority voting formats with a digital filter implemented to capture a CCP input signal of 3 TIMCLK periods.

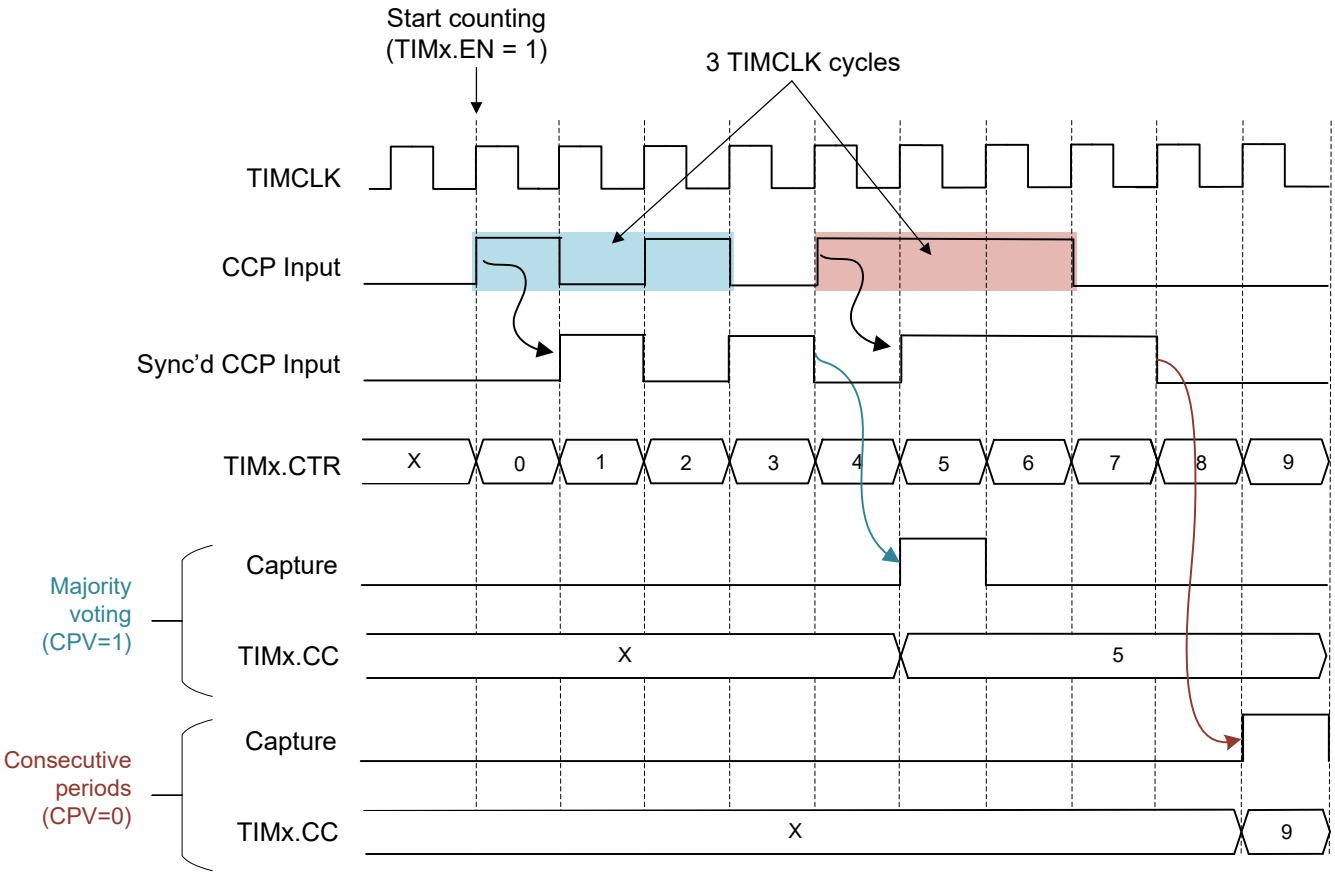


Figure 32-10. Consecutive Period and Majority Voting for Input Glitch Filtering using FP = 0 (3 TIMCLK cycles)

- For Majority Voting, the capture is made with the first CCP Input since the "glitch" period is less than one TIMCLK cycle.
- For Consecutive Period, the capture is made after the CCP Input remains high for 3 full TIMCLK cycles.

32.1.2.3.1.1.4 CCP Input Pulse Conditions

The TIMx.CCCTL_xy[0/1] register can control whether each timer instance generates a condition pulse based on the edge or polarity of the CCP input signal or trigger edge. The conditions that can be generated are:

- Advance condition (ACOND)
- Load condition (LCOND)
- Zero condition (ZCOND)
- Capture condition (CCOND)

Advance conditions

By default, the timer advances based on each rising edge of TIMCLK (TIMx.CCCTL_xy[0/1]ACOND = 0h). However, the timer can also advance based off the specified TIMx.CCCTL_xy[0/1].ACOND settings below.

Table 32-6. Advance pulse condition settings (ACOND)

ACOND	Condition
0h	Each rising edge of TIMCLK
1h	Rising edge of CCP or trigger assertion edge
2h	Falling edge of CCP or trigger de-assertion edge
3h	Either edge of CCP or trigger
5h	CCP high or trigger assertion

Load, zero, and capture conditions

Load, zero, and capture condition pulses can be generated based on the LCOND, ZCOND, and CCOND bitfield settings below within the TIMx.CCCTL_xy[0/1] register.

Table 32-7. Load, zero, and capture condition settings (LCOND, ZCOND, CCOND)

LCOND	ZCOND	CCOND	Condition
N/A	N/A	0h	None
1h	1h	1h	Rising edge of CCP or trigger assertion edge
2h	2h	2h	Falling edge of CCP or trigger de-assertion edge
3h	3h	3h	Either edge of CCP or trigger

32.1.2.3.1.1.5 Counter Control Operation

To specify what CC instance controls the load, zero, or advance event, the CZC, CAC, and CLC bitfields are configured in the TIMx.CTRCTL register.

See Counter Zero Control (CZC) settings for counter zero control settings. For example, if a timer triggers a ZCOND event in Channel 1, then CZC should be set to 1h to register that a ZCOND event in channel 1 triggers the zero event.

Table 32-8. Counter Zero Control (CZC) settings

TIMx.CTRCTL.CZC	Setting
0h	Channel 0 ZCOND event zeroes the TIMx instance
1h	Channel 1 ZCOND event zeroes the TIMx instance

See Counter Load Control (CLC) settings for counter load control settings. For example, if a timer triggers a LCOND event in Channel 1, then CLC should be set to 1h to register that a LCOND event in channel 1 triggers the load event.

Table 32-9. Counter Load Control (CLC) settings

TIMx.CTRCTL.CLC	Setting
0h	Channel 0 LCOND event loads the TIMx instance

Table 32-9. Counter Load Control (CLC) settings (continued)

TIMx.CTRCTL.CLC	Setting
1h	Channel 1 LCOND event loads the TIMx instance

See Counter Advance Control (CAC) settings for counter advance control settings. For example, if a timer triggers a ACOND event in Channel 1, then CAC should be set to 1h to register that a ACOND event in channel 1 triggers the advance event.

Table 32-10. Counter Advance Control (CAC) settings

TIMx.CTRCTL.CAC	Setting
0h	Channel 0 ACOND event advances the TIMx instance
1h	Channel 1 ACOND event advances the TIMx instance

32.1.2.3.1.2 Capture Mode Use Cases

Several different use cases can be achieved in capture mode and are discussed in the following sections.

32.1.2.3.1.2.1 Edge Time Capture

Edge time capture measures the time (in TIMCLK cycles) from the start of the capture operation to the signal edge. The counter is loaded when TIMx is enabled and counts with each TIMCLK period until the CCP edge is detected, which triggers the capture of the timer value and generates a capture event. The capture edge time is equivalent to the difference between the starting value of the counter and the capture value in TIMx.CC_xy[0/1] register.

Edge Time Capture Configuration

1. Set the TIMx.LOAD value.
2. In the CTRCTL register, set the desired counter control settings for:
   a. Counting mode (CM) and counter value after enable (CVAE) (see as described in Section 32.1.2.2)
   b. Zero (CZC), advance (CAC), and load control (CLC) to specify what condition controls zeroing, advancing, or loading the counter
   c. Repeat or one-shot mode (REPEAT)
3. Set TIMx.CCCTL_xy[0/1].COC = 1 for capture mode.
4. Configure CCP as an input for the CC block by setting respective bit in the CCPD registers. For instance, if TIMx Channel 0 is an input, set CCPD.C0CCP0 = 0.
5. For the corresponding CC block control register (CCCTL_01[0/1]), set CCOND to the corresponding setting to capture events based off the input signal condition (rising and/or falling edge). Additionally, set ZCOND or LCOND depending on the counting mode used.
6. Configure input capture settings in the TIMx.IFCTL_xy[0/1] register as described in Section 32.1.2.3.1.1 .
7. Enable the counter by setting EN = 1 or waiting for a capture event to occur from the input edge.

Example using up-counting mode for rising edge capture

In up-counting mode starting from zero (CM = 2, CVAE = 2), TIMx can be configured to generate a zero pulse and start the counter from the configured capture event (CCOND) by setting ZCOND to 1.

The expected internal timing for a rising or positive edge time capture in up-counting mode is shown in Figure 32-11.

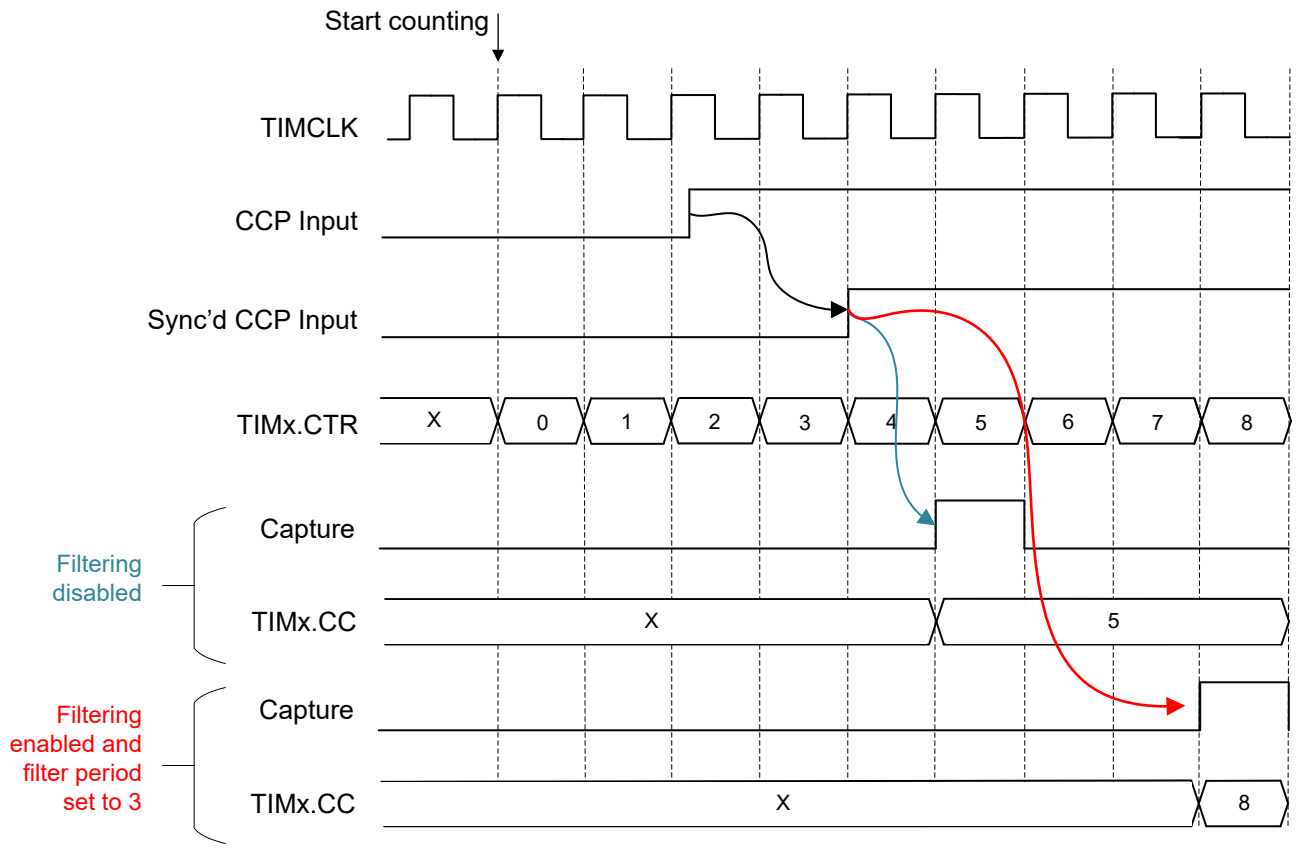


Figure 32-11. Edge Time Capture Mode in Up-Counting Mode, CVAE = 2

32.1.2.3.1.2.2 Period Capture

Period capture measures the period of a signal on an input CCP in TIMCLK cycles. On each positive (or negative) edge of the CCP input, the TIMx.CTR value is both captured into the TIMx.CC register to generate a capture event. The period capture time is equivalent to the difference between the starting value of the counter generated to the capture event, or time between reoccurring capture events for a periodic input signal.

Period Capture Configuration

1. Set the TIMx.LOAD value.
2. In the CTRCTL register, set the desired counter control settings for:
   a. Counting mode (CM) and counter value after enable (CVAE) (see as described in Section 32.1.2.2)
   b. Advance (CAC) to specify what condition controls advancing the counter
   c. Repeat or one-shot mode (REPEAT)
3. Set TIMx.CCCTL_xy[0/1].COC = 1 for capture mode.
4. Configure CCP as an input for the CC block by setting respective bit in the CCPD registers. For instance, if TIMx Channel 0 is an input, set CCPD.C0CCP0 = 0.
5. For the corresponding CC block control register (CCCTL_01[0/1]),
   a. Set CCOND to the corresponding setting to capture events based off the input signal condition (rising and/or falling edge)
   b. Set ZCOND or LCOND depending on the counting mode used
6. Configure input capture settings in the TIMx.IFCTL_xy[0/1] register as described in Section 32.1.2.3.1.1 .
7. Enable the counter by setting EN = 1 or waiting for a capture event to occur from the input edge.

Example using up-counting mode for rising-edge period capture

ADVANCE INFORMATION

In up-counting mode starting from zero (CM = 2, CVAE = 2), TIMx channel 0 can be configured to generate a capture event from a rising edge input by setting CCOND = 1h. After enabling the counter, when a rising edge input is detected, the counter will capture the counter value in TIMx.CC. After the capture event, set the TIMx.LOAD value back to 0 to reset the timer counter for the periodic CCP input signal.

The expected internal timing for a period capture in up-counting mode using two rising edges is shown in Figure 32-11.

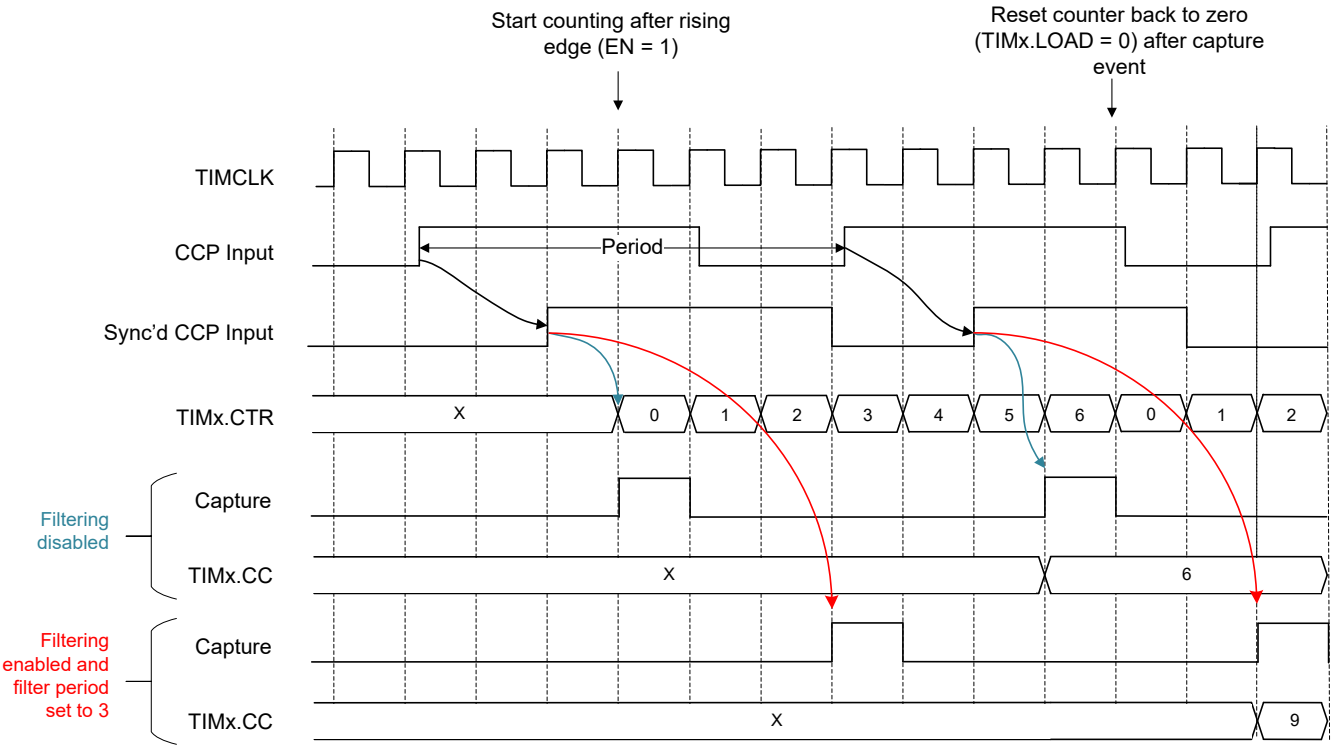


Figure 32-12. Period Capture Mode in Up-Counting Mode, CVAE = 2

32.1.2.3.1.2.3 Pulse Width Capture

Pulse width capture measures the high-time of a signal on CCP. The high time is the number of TIMCLK periods from rising edge to falling edge of the CCP input, and is useful for applications such as measuring the duty cycle of an PWM input signal. The counter is loaded at the positive edge and captured at the negative edge (capture event is generated).

Pulse-Width Capture Configuration

1. Set the TIMx.LOAD value.
2. In the CTRCTL register, set the desired counter control settings for:
   a. Counting mode (CM) and counter value after enable (CVAE) (see as described in Section 32.1.2.2)
   b. Zero (CZC), advance (CAC), and load control (CLC) to specify what condition controls zeroing, advancing, or loading the counter
   c. Repeat or one-shot mode (REPEAT)
3. Set TIMx.CCCTL_xy[0/1].COC = 1 for capture mode.
4. Configure CCP as an input for the CC block by setting respective bit in the CCPD registers. For instance, if TIMx Channel 0 is an input, set CCPD.C0CCP0 = 0.
5. For the corresponding CC block control register (CCCTL_01[0/1]), set CCOND to the corresponding setting to capture events based off the input signal condition (rising and/or falling edge). Additionally, set ZCOND or LCOND depending on the counting mode used.
6. Configure input capture settings in the TIMx.IFCTL_xy[0/1] register as described in Section 32.1.2.3.1.1 .

7. Enable the counter by setting EN = 1 or waiting for a capture event to occur from the input edge.

Example using up-counting mode for pulse width capture

In up-counting mode starting from zero (CM = 2, CVAE = 2), TIMx channel 0 can be configured to generate a zero pulse and start the counter from the configured capture event (CCOND) by setting ZCOND to 1. To start the counter, a load condition can be triggered from the CCP rising edge input by setting LCOND = 1.

The expected internal timing for a pulse width capture in up-counting mode using a rising and falling edge is shown in Figure 32-11.

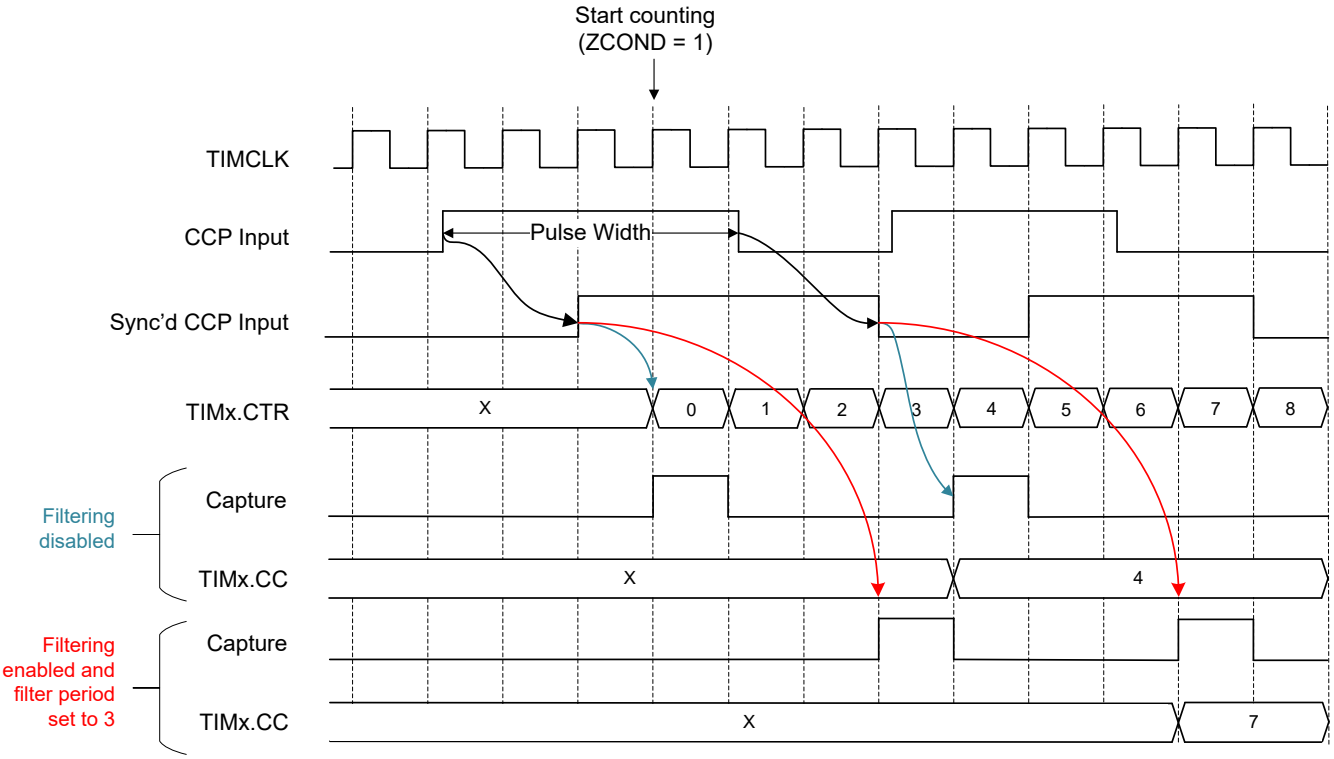


Figure 32-13. Pulse-Width Capture Mode

32.1.2.3.1.2.4 Combined Pulse Width and Period Time

Using two capture registers can combine pulse-width and period capture of a single input waveform. The input signal can be externally connected to CCP channel 0, and the IFCTL_01[1] register can be configured to have the input connected to CCP channel 1 internally so capture register 0 (TIMx.CC0) captures pulse width and capture register 1 (TIMx.CC1) captures period. The expected internal timing for combined pulse-width and period capture is shown in Figure 32-14.

Pulse-Width and Period Capture Configuration

1. Set the TIMx.LOAD value.
2. In the CTRCTL register, set the desired counter control settings for:
   a. Counting mode (CM) and counter value after enable (CVAE) (see as described in Section 32.1.2.2)
   b. Zero (CZC), advance (CAC), and load control (CLC) to specify what condition controls zeroing, advancing, or loading the counter
   c. Repeat or one-shot mode (REPEAT)
3. Set TIMx.CCCTL_xy[0/1].COC = 1 for capture mode for each CC channel.
4. Configure CCP as an input for each CC block by setting respective bits in the CCPD register. For instance, if TIMx Channel 0 is an input, set CCPD.C0CCP0 = 0.
5. For the corresponding CC block control register (CCCTL_01[0/1]),

a. Set CCOND to the corresponding setting to capture events based off the input signal condition (rising and/or falling edge)
b. Set ZCOND or LCOND depending on the counting mode used.

6. Configure input capture settings in the TIMx.IFCTL_xy[0/1] register as described in Section 32.1.2.3.1.1 .
7. Enable the counter by setting EN = 1 or waiting for a capture event to occur from the input edge.

Example using pulse-width and period time capture

In up counting mode, TIMx can be configured to generate a zero pulse and start the counter from the configured capture event (CCOND) by setting ZCOND to 1.

The expected internal timing for a pulse-width and period capture in up-counting mode using two CC blocks is shown in Figure 32-11.

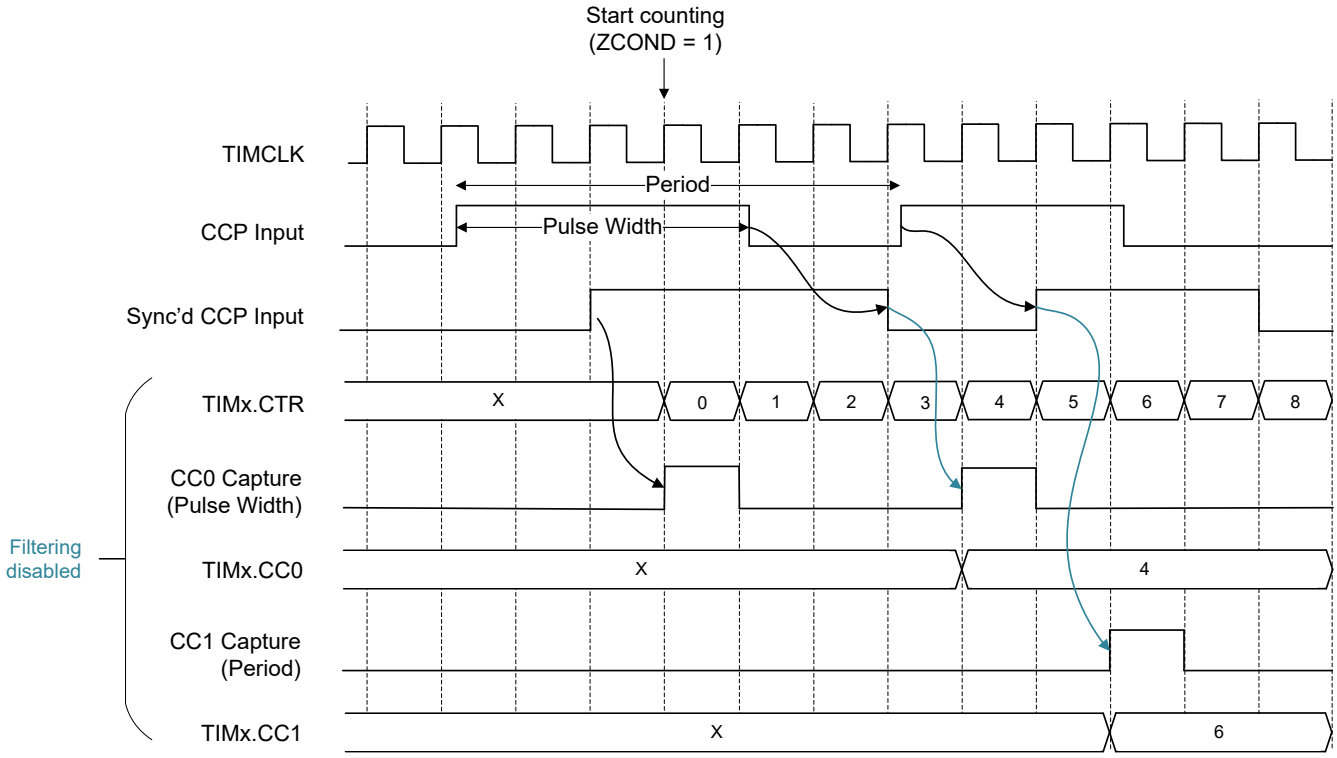


Figure 32-14. Combined Pulse-Width and Period Capture

32.1.2.3.2 Compare Mode

Compare mode is selected when TIMx.CCCTL_xy[0/1].COC = 0. Compare mode is used to generate a compare event to output PWM output signals at specific time intervals. Compare events can be used to generate a timing base internally or generate a PWM output with specific profiles using active, inactive, or toggle action behaviors, and optional deadband insertion.

Many types of compare mode events can be generated based on the configuration of the CC action register CCACT_xy[0/1]:

- Compare events: occurs for a CC channel when TIMx.CTR counts up (CCU) or down (CCD) to the value in TIMx.CC_xy[0/1]
- Secondary compare events: occurs when a secondary CCx block compare up (CC2U) or down (CC2U) is configured for another CCx block's CCU event or CCD event, respectively. This enables more flexible output generation from other external events such as comparator outputs or fault signals. It can be useful for real-time control applications, like digital power or motor control.

Table 32-11 shows the types of compare mode events that can be generated and conditions to generate the events.

Table 32-11. Compare Mode Events

Event	Name	Event Condition
CCDn (n = CC channel)	Capture/compare down event	When timer is counting down, TIMx.CTR = TIMx.CC_xy[0/1]
CCUn (n = CC channel)	Capture/compare up event	When timer is counting up, TIMx.CTR = TIMx.CC_xy[0/1]
CC2Dn (n = CC channel)	Secondary capture/compare down event	Occurs when CC2SELD is configured for the source of the CCDn event
CC2Un (n = CC channel)	Secondary capture/compare up event	Occurs when CC2SELU is configured for the source of the CCUn event

Note

Look at the device specific data sheet to check how many CC channels are available in each TIMx instance on the device.

32.1.2.3.2.1 Edge Count

In addition to event or PWM output generation, compare mode can also be used for input signal edge counting to determine when a number of edges has been detected. In edge count operation, a CCP input edge can advance the counter based on the ACOND condition. The counter register is initialized with the starting value, and the number of detected CCP input edges at any time can increment or decrement depending on the counting mode configuration. The user can count rising edges, falling edges, or both edges by configuring the CCOND value.

Edge Count Configuration

1. Set TIMx.CCCTL_xy[0/1].COC = 0 for compare mode.
2. Optionally set the corresponding TIMx.CC_xy[0/1] to a compare value to generate a compare interrupt when the counter reaches this value.
3. In the CTRCTL register, set the desired counter control settings for:
   a. Counting mode (CM) and counter value after enable (CVAE) (see as described in Section 32.1.2.2)
   b. Zero (CZC), advance (CAC), and load control (CLC) to specify what condition controls zeroing, advancing, or loading the counter
   c. Repeat or one-shot mode (REPEAT)
4. Set ACOND to a setting to advance the counter based on the input edge polarity.
5. Configure input capture settings as described in Section 32.1.2.3.1.1 , if desired.
6. Enable the counter by setting EN = 1.

Example using edge count operation using up-counting mode

In up-counting mode starting from zero (CM = 2, CVAE = 2), the expected internal timing for rising edge count operation to increment the counter is shown in Figure 32-15.

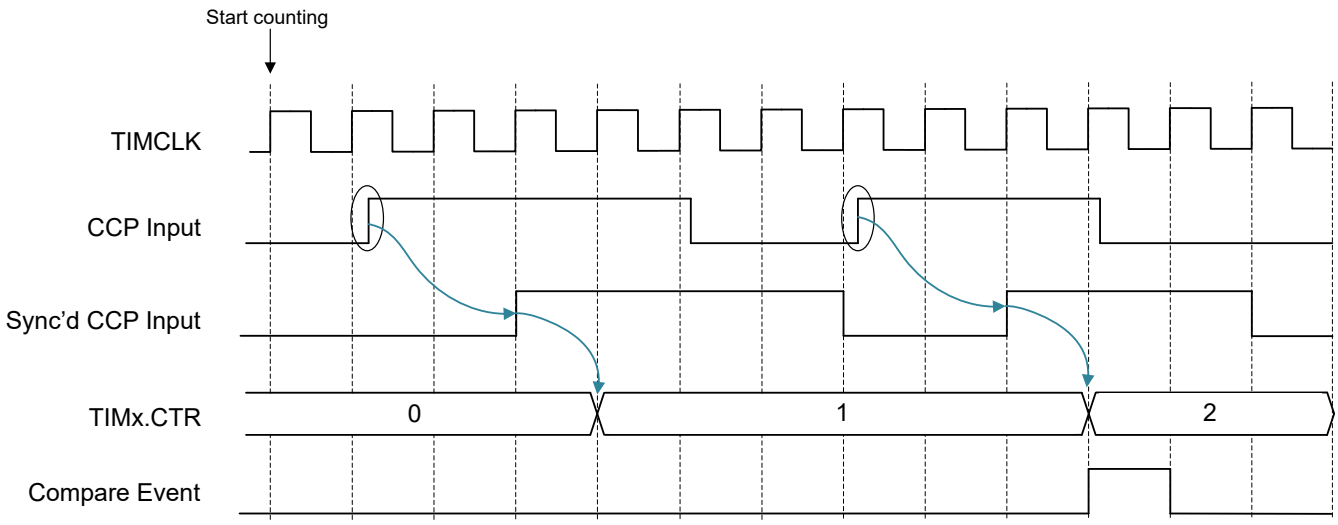


Figure 32-15. Edge Count Operation to Generate Compare Event (TIMx.CC = 2)

32.1.2.4 Shadow Load and Shadow Compare

Some timer modules have a shadow load and shadow compare register feature which gives the user the flexibility of holding the update of load and CC values until a certain event occurs. This is useful in timing-critical applications where PWM control signals need to be updated with correct timings, such as duty cycle updates. Refer to the timer features for specific configurations of the TIMx modules.

Note

See Section 32.1.1.1 and the device-specific data sheet for TIMx instances that support Shadow Load and Shadow Compare.

32.1.2.4.1 Shadow Load (TIMG4-7)

On shadow-load capable timers, the shadow load feature allows holding the update of load values until a zero event occurs. To enable shadow loading, set the TIMx.GCTL.SHDWLDEN bit while the timer is enabled (EN = 1).

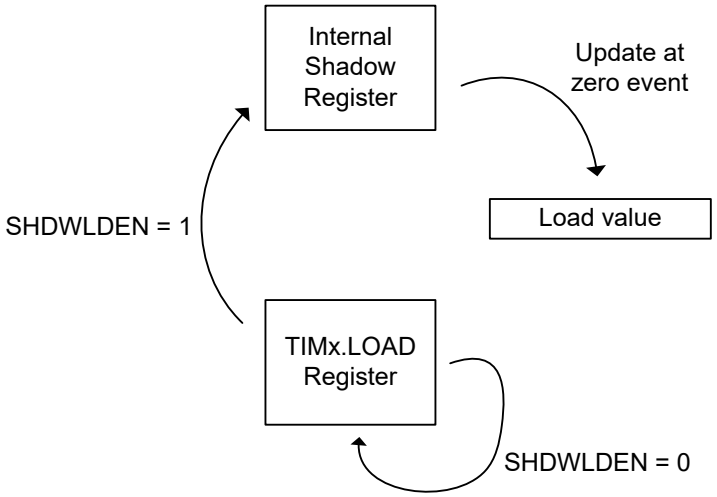
If the TIMx module has a shadow load feature, there is an internal shadow register for the load value (TIMx.LOAD). The shadow register will update the load value at a zero event as shown in Figure 32-16.

Note

On shadow-load capable timers, SHDWLDEN must be set or else the load value will not update.

If the timer instance does not have shadow load capability (standard timer), the load value will update immediately when TIMx.LOAD is written to.

Shadow-load capable timers update the load value from the internal shadow register when:
- TIMx.CTRCTL.EN = 1
- GCTL.SHDWLDEN = 1
- Available for TIMG4-7, TIMAx



Standard timers update the load value from TIMx.LOAD immediately when:
- TIMx.CTRCTL.EN = 1
- Available for TIMG0-3, TIMG8-14

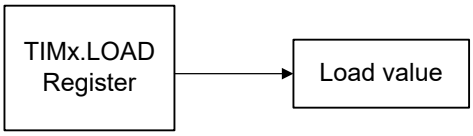


Figure 32-16. Load value updates for TIMx instances when EN=1

Note

To update the load value for shadow load capable timers, the timer must be disabled before changing the TIMx.LOAD value.

When TIMx.GCTL.SHDWLDEN = 1, load values update at zero events for all counting modes. Consult the counting mode operations below to determine if a shadow load is needed:

- In down-counting modes, since TIMx.LOAD value is updated when a zero event occurs, a shadow load is not needed in these modes.
- In up/down counting mode, TIMx.LOAD is compared with the counter value to determine if the peak is reached and when to start to counting down. A shadow load is necessary to ensure that TIMx counts up to the load value before the zero event, or else the load value can update immediately and cause incorrect timings.
- In up-counting mode, the timer counts to TIMx.LOAD. A shadow load is necessary to ensure that TIMx counts up to the load value before the zero event, or else the load value can update immediately and cause incorrect timings.

Figure 32-17 shows an example of how shadow load and shadow compare takes effect at the zero event for both the TIMx.LOAD and TIMx.CC value in up/down counting mode.

32.1.2.4.2 Shadow Compare (TIMG4-7, TIMG12-13)

When shadow compare is enabled for updating the capture/compare register (TIMx.CC), the value written to the respective compare register is first stored into a shadow compare register and then transferred to the compare register at different events configured by setting the TIMx.CCCTL_xy[0/1].CCUPD bits.

Additionally, the capture/compare action register (TIMx.CCACT) has the ability to update the action at different events configured by setting the TIMx.CCCTL_xy[0/1].CCACTUPD bits.

Table 32-12 shows the settings for configuring when shadow compare and actions occur at different events.

Table 32-12. Shadow Compare and Action Update Behavior

Bit Field	Value	Description/Comment
CCUPD / CCACTUPD	0	The value written to TIMx.CC register take effect immediately.
	1	The value written to the TIMx.CC register is stored in a shadow compare register and gets transferred to the TIMx.CC register in the TIMCLK cycle following a zero event (TIMx.CTR value equals 0).
	2	The value written to the TIMx.CC register is stored in a shadow compare register and gets transferred to the TIMx.CC register in the TIMCLK cycle following a compare (down) event (TIMx.CTR value equals TIMx.CC)
	3	The value written to the TIMx.CC register is stored in a shadow compare register and gets transferred to the TIMx.CC register in the TIMCLK cycle following a compare (up) event (TIMx.CTR value equals TIMx.CC)
	4	The value written to the TIMx.CC register is stored in a shadow compare register and gets transferred to the TIMx.CC register in the TIMCLK cycle following a zero or load event (TIMx.CTR value equals 0 or TIMx.CTR equals TIMx.LOAD). **Note: this update mechanism is defined for use only in up/down counting mode.**
	5	The value written to the TIMx.CC register is stored in a shadow compare register and gets transferred to the TIMx.CC register in the TIMCLK cycle following a zero event and the repeat count equaling zero (TIMx.CTR value equals 0)
	6	The value written to the TIMx.CC register is stored in a shadow compare register, and gets transferred to the TIMx.CC register in the TIMCLK cycle following a trigger pulse. See Section 32.1.2.6 .

Figure 32-17 shows an example of how shadow load and shadow compare takes effect at the zero event for both the TIMx.LOAD and TIMx.CC value in up/down counting mode.

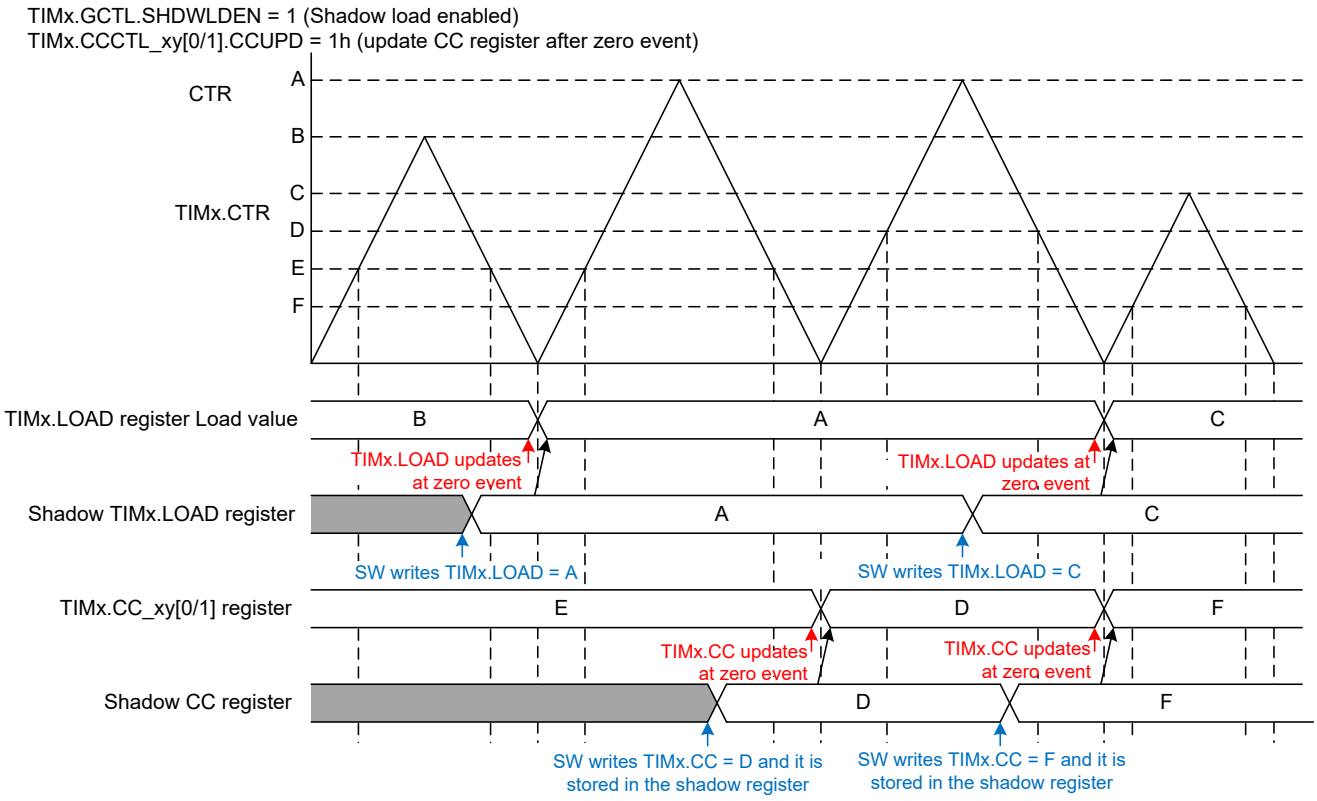


Figure 32-17. Shadow Load and Shadow Compare Taking Effect at Zero Event in Up/Down Mode

ADVANCE INFORMATION

32.1.2.5 Output Generator

The output signal generation unit can be used with the counter and capture/compare modules to generate desired pulse-width modulation (PWM) output waveforms, event signals, synchronized capture inputs, or the counter direction. Many output waveforms are generated from counter events (load, zero, counter direction) and a compare match within the capture/compare block.

The key registers for generation of output signals are:

- **LOAD**: the contents of this register are copied to the counter (TIMx.CTR) on any operation designated to do a "load". This value is also used to compare with the counter value for generating a "Load Event" that can be used for interrupt, trigger, or signal generator actions.
- **CCPD:** this register configures the direction of the CCP pins as inputs or outputs.
- **CC_xy[0/1]**: this is a register used as a compare value to the current counter to create an match event.
- **CTRCTL**: this register provides control over the counter operation in different conditions.
- **CCCTL_xy[0/1]**: this register controls the operations of the respective CC registers and the counter
- **OCTL_xy[0/1]**: this register controls the output of the capture-compare portion of the counter. This includes the ability to select the source of what is driven out along with initial condition values and final inversion options.
- **CCACT_xy[0/1]**: this register controls the actions of the signal generator of the capture-compare portion based on the events created in the counter block, the capture and compare block, and debug events.
- **ODIS:** this register disables the output signal selected by OCTL.CCPO (before conditional inversion) to allow software the ability to hold the CCP output low during configuration or shutdown.

Output Connection for TIMx shows the TIMx output block diagram.

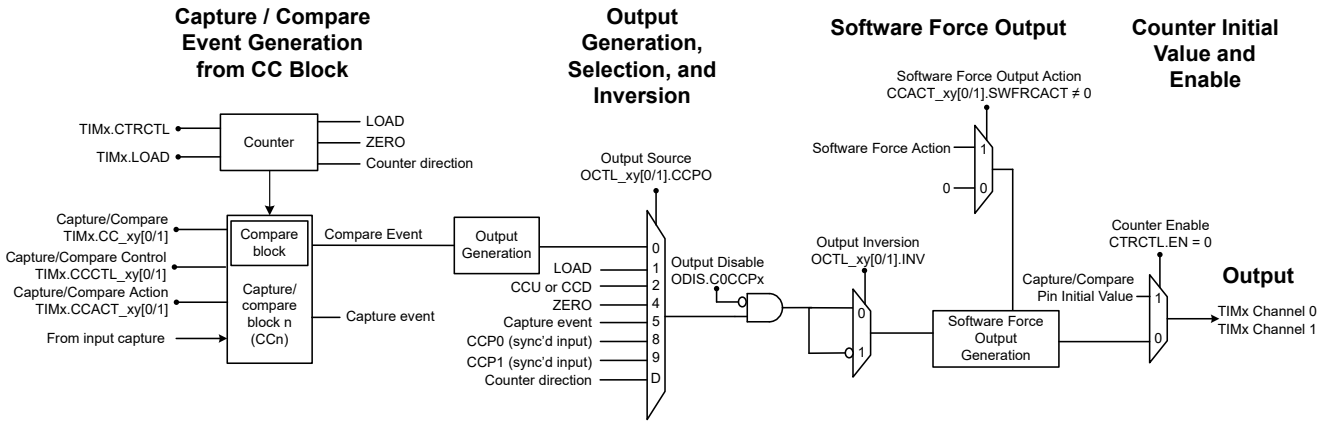


Figure 32-18. Output Connection for TIMx

Signal Generator Actions

Signal Generator Actions from Compare Event shows the types of signal generator actions capable by the output generator. Signal generator actions are configured in the CCACT_xy[0/1] register for zero, load, and compare events. For types of compare events, see Compare Mode Events.

Table 32-13. Signal Generator Actions from Compare Event

Value	Action
0h	Event is disabled and a lower priority event is selected if asserting
1h	CCP output value is set high
2h	CCP output value is set low
3h	CCP output value is toggled

32.1.2.5.1 Configuration

There are five stages to configuring output signal generation in TIMx devices:

- Counter and CC Block Event Generation
- Output Generation, Selection and Inversion
- Software Force Output
- Counter Initial Value and Enable

Counter and CC Block Event Generation

The counter block contains the counter and produces a load event (L), zero event (Z), and direction of counting based on the counting mode used.

The CC blocks contain the CC register and can generate two types of output signals: compare match events and capture events. Please see Table 32-11 for the compare events that can be generated.

Output Generation, Selection and Inversion

The TIMx.CCACT register specifies the waveform generation of a CCP output depending on the counting mode and counter compare actions.

TIMx.OCTL_xy[0/1].CCPO controls the CCP output selection from the output generation unit, counter events, compare events, capture events, fault events, or signal inputs. The output disable register (ODIS) can optionally disable the CCP output to optionally hold the CCP output low during configuration or shutdown. TIMx.OCTL_xy[0/1].INV controls final inversion options.

Software Force Output

The output of the signal generator can be overwritten in software by setting CCCTL_xy[0/1].SWFRCACT to a nonzero setting.

For more information, see Section 32.1.2.5.3 .

Counter Compare Initial Value and Enable

To specify an initial value for the CCP output while the counter is disabled, set OCTL_xy[0/1].CCPIV to 0 for a low value or 1 for a high value. This is useful for applications where CCP outputs need to be in a default state before enabling the counter.

To enable the counter, set TIMx.CTRCTL.EN to 1.

32.1.2.5.2 Use Cases

Several different use cases can be achieved with the output generator and are discussed in the following sections.

32.1.2.5.2.1 Edge-Aligned PWM

To generate edge-aligned PWMs, TIMx can be configured for up- or down-counting mode. The waveform uses load, zero, and compare events to drive the CCPx output high or low depending on the configuration settings of the compare/capture block and counter.

In TIMCLK cycles, the PWM period is (TIMx.LOAD + 1) and the duty cycle is 1 - (TIMx.CC_xy[0/1] / TIMx.LOAD) for down counting mode and (TIMx.CC_xy[0/1] / TIMx.LOAD) for up counting mode.

Edge-Aligned PWM Configuration

To generate edge-aligned PWMs using compare match events from the counter:

1. In the TIMx.CTRCTL register, set the desired counter control settings for:
   a. Up-counting (CM = 2) or down-counting mode (CM = 0) and counter value after enable (CVAE) (see as described in Section 32.1.2.2)

ADVANCE INFORMATION

b. Zero (CZC), advance (CAC), and load control (CLC) to specify what condition controls zeroing, advancing, or loading the counter
c. Repeat or one-shot mode (REPEAT)

2. Set the TIMx.LOAD value to configure the PWM period.
3. Set the TIMx.CC_xy[0/1] value to configure the duty cycle.
4. Set TIMx.CCCTL_xy[0/1].COC = 1 for compare mode.
5. Configure CCP as an output for the CC block by setting respective bit in the CCPD registers. For instance, if TIMx Channel 0 is an output, set CCPD.C0CCP0 = 1.
6. In TIMx.CCACT_xy[0/1], set the CCP output action settings for compare events, zero events, load events, or software force action.
7. In TIMx.OCTL_xy[0/1], set CCPO = 0 to select the signal generator output.
8. Enable the corresponding CCP output by setting ODIS.C0CCPn to 1 for the corresponding counter n.
9. Configure polarity of the signal using the CCPOINV bit, and configure CCPIV to specify the CCP output state while disabled.
10. Enable the counter by setting TIMx.CTRCTL.EN = 1.

Example using edge-aligned PWM in down-counting mode

A typical 2-channel edge-aligned PWM generation for down-counting mode is shown in Figure 32-19 with the following edge-aligned PWM output waveforms:

- CCP0 output generates:
  - High pulse-width from TIMx.LOAD to TIMx.CC0 value (LACT = 1h)
  - Low pulse-width from TIMx.CC0 value to zero (CDACT = 2h)
- CCP1 output generates:
  - High pulse-width from TIMx.LOAD to TIMx.CC1 value (LACT = 1h)
  - Low pulse-width from TIMx.CC1 value to zero (CDACT = 2h)

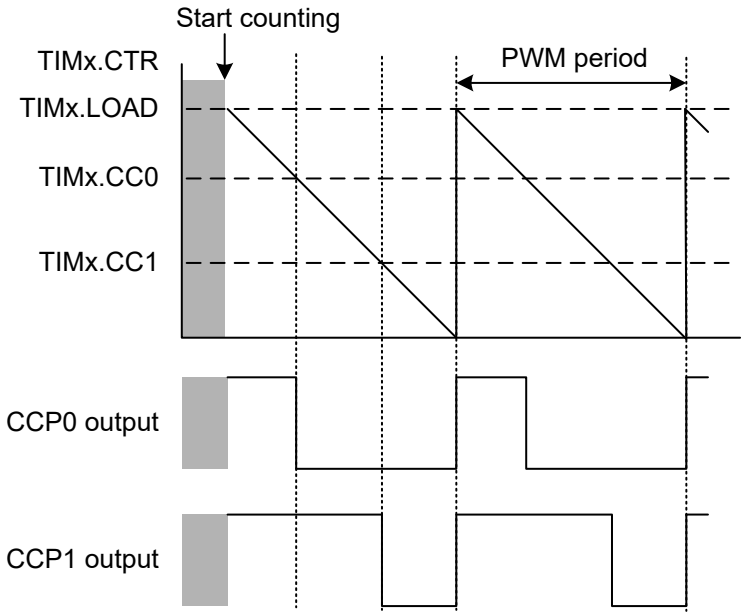


Figure 32-19. Edge-Aligned PWM Signals in Down-Counting Mode

Example using edge-aligned PWM in up-counting mode

A typical 2-channel edge-aligned PWM generation for up-counting mode is shown in Figure 32-19 with the following edge-aligned PWM output waveforms:

ADVANCE INFORMATION

- CCP0 output generates:
  - High pulse-width from zero to TIMx.CC0 value (ZACT = 1h)
  - Low pulse-width from TIMx.CC0 value to TIMx.LOAD (CUACT = 2h)
- CCP1 output generates:
  - High pulse-width from zero to TIMx.CC1 value (ZACT = 1h)
  - Low pulse-width from TIMx.CC1 value to TIMx.LOAD (CUACT = 2h)

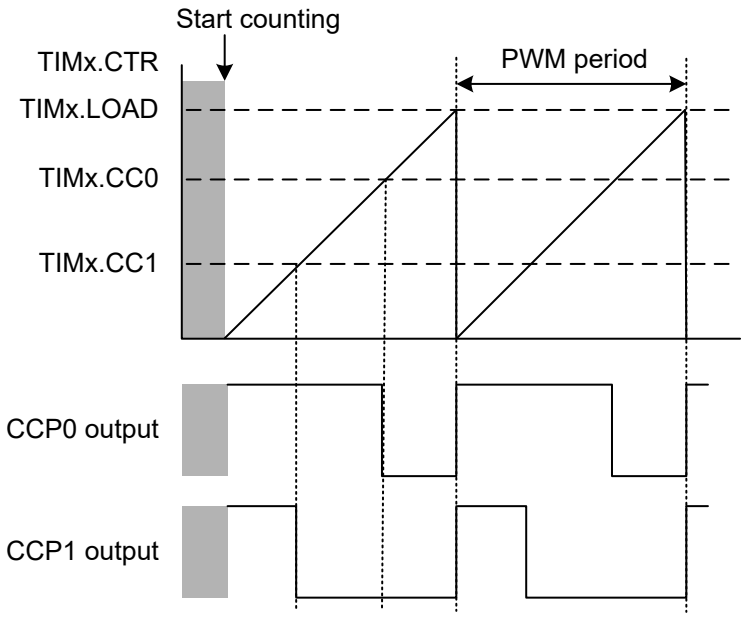


Figure 32-20. Edge-Aligned PWM Signals in Up-Counting Mode

32.1.2.5.2.2 Center-Aligned PWM

To generate center-aligned PWMs, TIMx is configured for up/down counting mode and the TIMx.LOAD value contains the half-period. The waveform uses up compare events and down compare events to drive the CCPx output high or low depending on the configuration settings of the compare/capture block and counter.

In TIMCLK cycles, the PWM period is (2 * TIMx.LOAD) and the duty cycle is 1 - (TIMx.CC_xy[0/1] / TIMx.LOAD).

Center-Aligned PWM Configuration

To generate center-aligned PWMs using compare match events from the counter:

1. In the TIMx.CTRCTL register, set the desired counter control settings for:
   a. Up/down counting mode (CM = 1) and counter value after enable (CVAE) (as shown in Section 32.1.2.2)
   b. Zero (CZC), advance (CAC), and load control (CLC) to specify what condition controls zeroing, advancing, or loading the counter
   c. Repeat or one-shot mode (REPEAT)
2. Set the TIMx.LOAD value to configure the PWM period.
3. Set the TIMx.CC_xy[0/1] value to configure the duty cycle.
4. Set TIMx.CCCTL_xy[0/1].COC = 1 for compare mode.
5. Configure CCP as an output for the CC block by setting respective bit in the CCPD registers. For instance, if TIMx Channel 0 is an output, set CCPD.C0CCP0 = 1.
6. In TIMx.CCACT_xy[0/1], set the CCP output action settings for compare events, zero events, load events, or software force action.
7. In TIMx.OCTL_xy[0/1], set CCPO = 0 to select the signal generator output.
8. Enable the corresponding CCP output by setting ODIS.C0CCPn to 1 for the corresponding counter n.

9. Configure polarity of the signal using the CCPOINV bit, and configure CCPIV to specify the CCP output state while disabled.
10. Enable the counter by setting TIMx.CTRCTL.EN = 1.

Example using center-aligned PWM in up/down counting mode

A typical 2-channel center-aligned PWM generation using up/down counting mode is shown in Figure 32-21 with the following center-aligned PWM output waveforms:

- CCP0 output generates:
  - High pulse-width from TIMx.CC0 compare up event to TIMx.CC0 compare down event (CUACT = 1h)
  - Low pulse-width from TIMx.CC0 compare down event to TIMx.CC0 compare up event (CDACT = 2h)
- CCP1 output generates:
  - High pulse-width from TIMx.CC0 compare up event to TIMx.CC0 compare down event (CUACT = 1h)
  - Low pulse-width from TIMx.CC0 compare down event to TIMx.CC0 compare up event (CDACT = 2h)

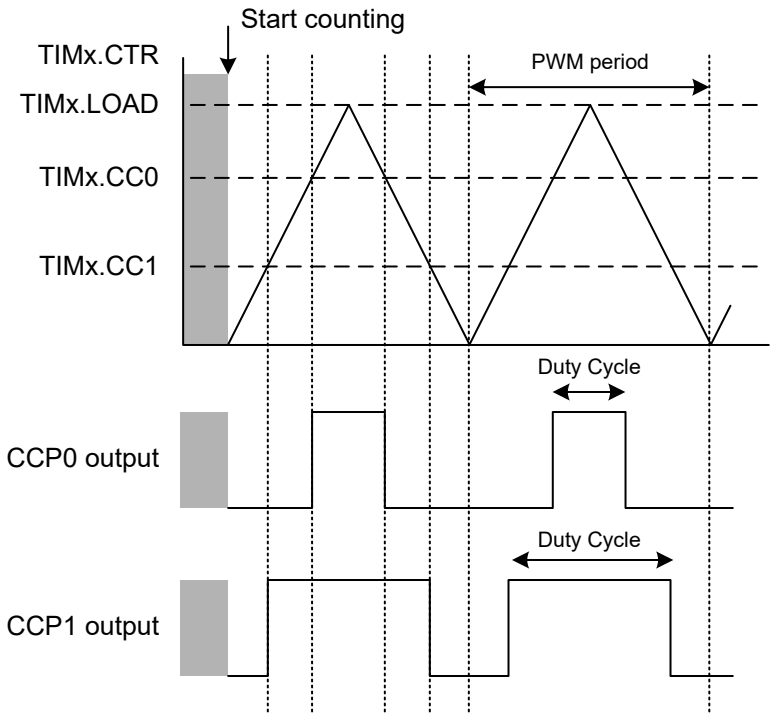


Figure 32-21. Center-Aligned PWM

32.1.2.5.3 Forced Output

Each output channel signal can be forced to a high or low level directly by software, independently of any comparison between the compare register and the counter. A shadow register exists to ensure the forced output action occurs at the end of the timer period.

The output of the CCP channel can be forced to high or low by setting the SWFRCACT bit in the TIMx.CCACT_xy[0/1] register.

Force Output Action Configuration shows the software force output action configuration options.

Table 32-14. Force Output Action Configuration

Bit Field	Value	Description/Comment
SWFRCACT	0	No forced output. Output is directly from the signal generation block.
	1	Force output high

ADVANCE INFORMATION

Table 32-14. Force Output Action Configuration (continued)

Bit Field	Value	Description/Comment
	2	Force output low

32.1.2.6 Synchronization With Cross Trigger

When using a main-secondary timer configuration by connecting multiple timers together, the cross-trigger feature can instruct multiple timer modules in the same power domain or across different power domains using the event fabric to start counting simultaneously.

Cross-triggers can be enabled using software, compare events from other timer instances, zero or load events, or generic subscriber events. Some applications may require more than one counter block that can be simultaneously started across the same power domain (such as TIMA0 and TIMA1) or different power domains (such as TIMA0 and TIMG0).

This configuration uses cross triggers from a main timer module as the input trigger condition for the secondary timers. The timer cross trigger is essentially the combined logic of the hardware and software conditions that control the EN bit in the TIMx.CTRCTL register.

The cross triggers that are outputted from the main timer are connected to the external trigger input of other secondary timer modules. As shown in Figure 32-22, TIMGx is the main timer and TIMAx is the secondary timer that will be cross triggered in the configuration example.

Note

For power domains and cross trigger selection sources enabled for timer instances, refer to the device-specific data sheet.

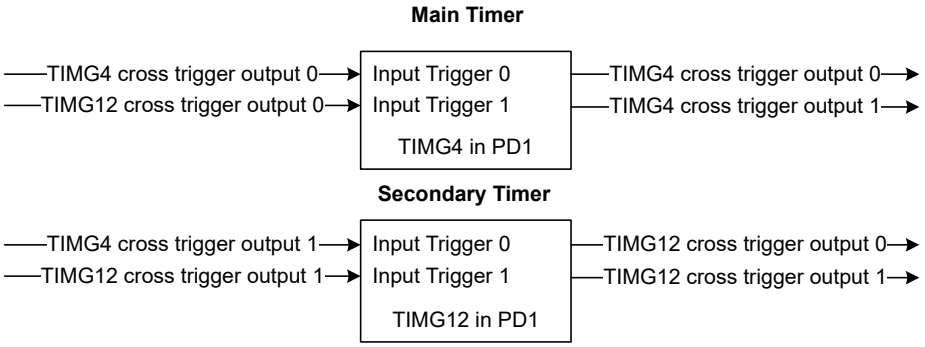


Figure 32-22. Cross Trigger Connections for Main Timer (TIMGx) and Secondary Timer (TIMAx) in Power Domain 1

32.1.2.6.1 Main Timer Cross Trigger Configuration

The following steps are used to configure the main timer cross trigger (which is TIMGx in this example):

1. Configure the main timer (which triggers other secondary timers) for the desired function, such as PWM output generation or using compare mode, to trigger other peripherals. See Section 32.1.2.5 for how to configure for PWM generation.
2. Select which cross trigger output needs to be generated. For example, in Figure 32-22, TIMGx cross trigger 1 can be used to trigger TIMAx and TIMGx cross trigger 0 can be used to trigger itself.
3. Enable the cross trigger output function by setting TIMx.CTTRIGCTL.CTEN bit to 1.
4. Choose how to trigger the start of these connected timers, which can be a software trigger or hardware trigger from a subscriber port, zero, load, or compare event.
   a. For a software event trigger, set the TIMx.CTTRIG.TRIG bit.
   b. For a hardware trigger event, select the source for the trigger using TIMx.CTTRIGCTL.EVTCTTRIGSEL and enable the hardware trigger by setting TIMx.CTTRIGCTL.EVTCTEN.

ADVANCE INFORMATION

Figure 32-23 shows the connection logic and registers.

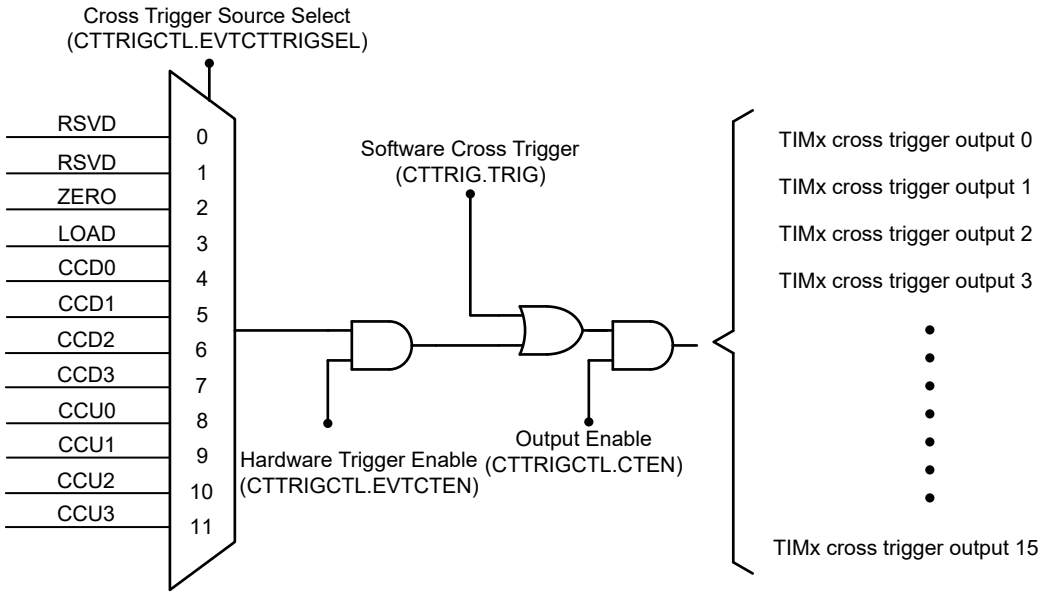


Figure 32-23. Main Timer Cross Trigger Output Configuration

32.1.2.6.2 Secondary Timer Cross Trigger Configuration

The following steps are used to configure the secondary timer cross trigger (which is TIMAx in this example):

1. Configure the secondary timer (triggered by the main timer) for the desired function for this timer, such as PWM output generation or using compare mode, to trigger other peripherals. See Section 32.1.2.5 for how to configure for PWM generation.
2. Select which input trigger to use according to the device-specific data sheet. Using the example connection in Figure 32-22, TIMAx must be triggered by TIMGx and the cross trigger output 1 of TIMGx is connected to input trigger 0 of TIMAx. Therefore, select input trigger 0 of TIMAx by setting TIMA.TSEL.ETSEL bit to 0.
3. Enable the input trigger function by setting the TIMA.TSEL.TE bit to 1.
4. Set TIMAx.IFCTL_01[0].ISEL = 3 and TIMAx.IFCTL_01[1].ISEL = 3 to select the trigger as the input source.
   a. For a center-aligned PWM, set the TIMA.CCCTL_01[0].ZCOND and TIMA.CCCTL_01[1].ZCOND bits to 1 to use a trigger assertion edge for a zero event.
   b. For an edge-aligned PWM, set the TIMA.CCCTL_01[0].LCOND and TIMA.CCCTL_01[1].LCOND bits to 1 to use the trigger assertion edge for a load event.
5. The TIMx.CTRCTL.EN bit is set as the result of an LCOND or ZCOND condition being met, and the counter value changes to the load value or zero value, respectively.

As the main timer TIMGx must also trigger itself, complete the previous configuration steps for TIMAx to trigger TIMGx itself.

Figure 32-24 shows the logic connection.

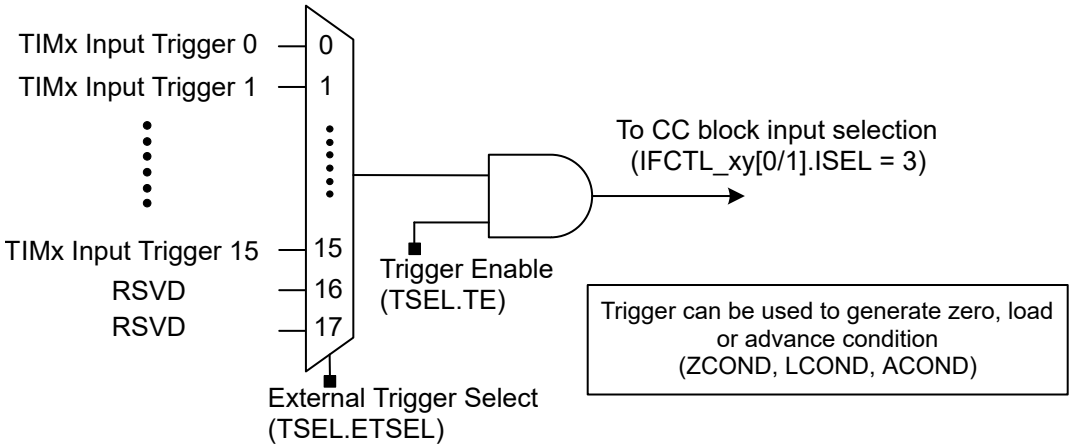


Figure 32-24. Secondary Timer Cross Trigger Input Configuration

Note

Refer to "TIMx Cross Trigger Map" in the device-specific data sheet for enabled cross trigger mapping using the ETSEL bit. For instance, if the timer instances of a device all use trigger input 0 (TRIG0) to cross trigger other timers, then only TRIG0 can be used to cross trigger other instances.

32.1.2.7 Low Power Operation

For detailed information on the low power modes in terms of available clock source and behaviors, refer to *Power Domains*.

Timer modules in power domain PD0 can be active and configured to continue counting in all power modes except SHUTDOWN mode. See *Power Domains* for the available clock sources in each low-power mode. The user needs to configure the proper clock to source the timer in low-power mode.

Timer modules in power domain PD1 can only be active in RUN and SLEEP modes. When the system goes to STOP or STANDBY mode, the timer modules will be forced to a disabled state and resume when the systems moves back to RUN or SLEEP modes.

32.1.2.8 Interrupt and Event Support

TIMx interrupts and events can be configured to any peripheral of the device using the Event Manager. The timer can generate interrupts or events as an **event publisher**. See *Using Event Registers* for guidance on configuring the event registers for CPU interrupts or other events.

The TIMx module contains three event publishers.

- One event publisher (CPU_INT) manages TIMx interrupt requests (IRQs) to the CPU subsystem through a static event route.
- The second event (GEN_EVENT0) generates an interrupt to ADC through a static event route.
- The third event (GEN_EVENT1) generates an interrupt to the DMA through a static event route.

TIMx events are summarized in Table 32-15.

Table 32-15. TIMx Events

Event	Type	Source	Destination	Route	Configuration	Functionality
CPU interrupt	Publisher	TIMx	CPU Subsystem	Static route	CPU_INT registers	Fixed interrupt route from TIMx to CPU
Generic publisher event	Publisher	TIMx	ADC	Static route	GEN_EVENT0 registers	Configurable interrupt route from TIMx to ADC
Generic publisher event	Publisher	TIMx	DMA	Static route	GEN_EVENT 1 registers	Configurable interrupt route from TIMx to DMA

ADVANCE INFORMATION

32.1.2.8.1 CPU Interrupt Event Publisher (CPU_INT)

The TIMx module provides 18 interrupt sources (depending on the specific TIMx module features) which can be configured to source a CPU interrupt event. The CPU interrupt event configuration is managed with the CPU_INT event management registers. Table 32-16 lists the CPU interrupt events from the TIMx in order of decreasing interrupt priority.

Table 32-16. TIMx CPU Interrupt Event Conditions (CPU_INT)

IIDX STAT	Name	Description	Timer Module
0x01	Z	Zero event interrupt. This interrupt is set when there is a zero event.	TIMx
0x02	L	Load event interrupt. This interrupt is set when there is a load event.	TIMx
0x05	CCD0	Capture or compare 0 down event. This interrupt is set when there is a down compare match event at CC0.	TIMx
0x06	CCD1	Capture or compare 1 down event. This interrupt is set when there is a down compare match event at CC1.	TIMx
0x09	CCU0	Capture or compare 0 up event. This interrupt is set when there is a up compare match event at CC0.	TIMx
0x0A	CCU1	Capture or compare 1 up event. This interrupt is set when there is a up compare match event at CC1.	TIMx
0x1A	TOV	Trigger overflow interrupt. This interrupt is set if a trigger event is generated while the associated trigger channel is active.	TIMx

See *Using Event Registers* for guidance on configuring the event registers for CPU interrupts.

32.1.2.8.2 GEN_EVENT0 and GEN_EVENT1

The GEN_EVENT0 and GEN_EVENT1 registers are used to select a peripheral condition (TIMx GEN_EVENT0 and GEN_EVENT1 Conditions) to use for generating an event to ADC (GEN_EVENT0) or DMA (GEN_EVENT1).

Table 32-17. TIMx GEN_EVENT0 and GEN_EVENT1 Conditions

IIDX STAT	Name	Description	Timer Module
0x01	Z	Zero event interrupt. This interrupt is set when there is a zero event.	TIMx
0x02	L	Load event interrupt. This interrupt is set when there is a load event.	TIMx
0x05	CCD0	Capture or compare 0 down event. This interrupt is set when there is a down compare match event at CC0.	TIMx
0x06	CCD1	Capture or compare 1 down event. This interrupt is set when there is a down compare match event at CC1.	TIMx
0x09	CCU0	Capture or compare 0 up event. This interrupt is set when there is a up compare match event at CC0.	TIMx
0x0A	CCU1	Capture or compare 1 up event. This interrupt is set when there is a up compare match event at CC1.	TIMx
0x1A	TOV	Trigger overflow interrupt. This interrupt is set if a trigger event is generated while the associated trigger channel is active.	TIMx

See *Using Event Registers* for guidance on configuring the event registers.

32.1.2.9

Note

MFCLK refers to a 4MHz fixed frequency clock that is not configured on AM13E230x devices. Please disregard any mention of MFCLK within the following registers.

32.2 TIMERS Registers

This Section describes the TIMERS Registers.

ADVANCE INFORMATION

32.2.1 TIMERS Base Address Table

Table 32-18. TIMERS Base Address Table

Bit Field Name		DriverLib Name	Base Address
Instance	Structure		
Timg4Regs	TIMG4_REGS	TIMG4	0x4018_0000
Timg12Regs	TIMG12_REGS	TIMG12	0x4018_8000

32.2.2 TIMG4_REGS Registers

Table 32-19 lists the memory-mapped registers for the TIMG4_REGS registers. All register offset addresses not listed in Table 32-19 should be considered as reserved locations and the register contents should not be modified.

Table 32-19. TIMG4_REGS Registers

Offset	Acronym	Register Name	Section
4h	CCP0	CCP0 PinCM	Go
8h	CCP1	CCP0 PinCM	Go
24h	CCP2_CMPL	CCP2_CMPL PinCM	Go
28h	CCP3_CMPL	CCP3_CMPL PinCM	Go
204h	CCP0	CCP0 FUPDATE	Go
208h	CCP1	CCP1 FUPDATE	Go
224h	CCP2_CMPL	CCP2_CMPL FUPDATE	Go
228h	CCP3_CMPL	CCP3_CMPL FUPDATE	Go
400h	FSUB_0	Subsciber Port 0	Go
404h	FSUB_1	Subscriber Port 1	Go
444h	FPUB_0	Publisher Port 0	Go
448h	FPUB_1	Publisher Port 1	Go
480h	CPU_CONNECT_0	CPU connect 0 configuration byte	Go
800h	PWREN	Power enable	Go
804h	RSTCTL	Reset Control	Go
814h	STAT	Status Register	Go
1000h	CLKDIV	Clock Divider	Go
1008h	CLKSEL	Clock Select for Ultra Low Power peripherals	Go
1018h	PDBGCTL	Peripheral Debug Control	Go
1020h	IIDX	Interrupt index	Go
1028h	IMASK	Interrupt mask	Go
1030h	RIS	Raw interrupt status	Go
1038h	MIS	Masked interrupt status	Go
1040h	ISET	Interrupt set	Go
1048h	ICLR	Interrupt clear	Go
1050h	IIDX	Interrupt index	Go
1058h	IMASK	Interrupt mask	Go
1060h	RIS	Raw interrupt status	Go
1068h	MIS	Masked interrupt status	Go
1070h	ISET	Interrupt set	Go
1078h	ICLR	Interrupt clear	Go
1080h	IIDX	Interrupt index	Go
1088h	IMASK	Interrupt mask	Go
1090h	RIS	Raw interrupt status	Go
1098h	MIS	Masked interrupt status	Go
10A0h	ISET	Interrupt set	Go
10A8h	ICLR	Interrupt clear	Go
10E0h	EVT_MODE	Event Mode	Go
10FCh	DESC	Module Description	Go
1100h	CCPD	CCP Direction	Go

Table 32-19. TIMG4_REGS Registers (continued)

Offset	Acronym	Register Name	Section
1104h	ODIS	Output Disable	Go
1108h	CCLKCTL	Counter Clock Control Register	Go
110Ch	CPS	Clock Prescale Register	Go
1110h	CPSV	Clock prescale count status register	Go
1114h	CTTRIGCTL	Timer Cross Trigger Control Register	Go
111Ch	CTTRIG	Timer Cross Trigger Register	Go
1800h	CTR	Counter Register	Go
1804h	CTRCTL	Counter Control Register	Go
1808h	LOAD	Load Register	Go
1810h + formula	CC_01_y	Capture or Compare Register 0/1	Go
1830h + formula	CCCTL_01_y	Capture or Compare Control Registers 0/1	Go
1850h + formula	OCTL_01_y	CCP Output Control Registers 0/1	Go
1870h + formula	CCACT_01_y	Capture or Compare Action Registers 0/1	Go
1880h + formula	IFCTL_01_y	Input Filter Control Register 0/1	Go
18B0h	TSEL	Trigger Select Register	Go

Complex bit access types are encoded to fit into small table cells. Table 32-20 shows the codes that are used for access types in this section.

Table 32-20. TIMG4_REGS Access Type Codes

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

1 CCP0 Register (Offset = 4h) [Reset = 00000000h]

CCP0 is shown in Figure 32-25 and described in Table 32-21.

Return to the Summary Table.

CCP0 PinCM register in full write region

Figure 32-25. CCP0 Register

31	30	29	28	27	26	25	24
RESERVED	GFLT	SLEW	WCOMP	WUEN	INV	HIGHZ1	HIGHZ0
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
RESERVED	DRV			HYSTEN	INENA	PIPU	PIPD
R/W-0h	R/W-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
GSTATE		RESERVED					
R/W-0h		R/W-0h					
7	6	5	4	3	2	1	0
PSTATE		RESERVED					
R/W-0h		R/W-0h					

Table 32-22. CCP0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	
30	GFLT	R/W	0h	Glitch Filter Enable 0h = No internal glitch filter 1h = Use internal glitch filter
29	SLEW	R/W	0h	Reserved Slew Rate Control 0h = No Slew Rate Control 1h = Use Slew Rate Control
28	WCOMP	R/W	0h	Wake up compare value 0h = Match '0' will wake 1h = Match '1' will wake
27	WUEN	R/W	0h	Wake up enable 0h = Wake up not enabled 1h = Wake up enabled
26	INV	R/W	0h	Invert digital input/output relative to peripheral/GPIO 0h = Input and output are non-inverted 1h = Input and output are inverted
25	HIGHZ1	R/W	0h	High-Z instead of high output 0h = Pin can be driven high 1h = Pin is tri-stated instead of driven high
24	HIGHZ0	R/W	0h	High-Z instead of low output 0h = Pin can be driven low 1h = Pin is tri-stated instead of driven low
23	RESERVED	R/W	0h	
22-20	DRV	R/W	0h	Drive strength options 0h = Lowest drive strength 1h = Drive strength 2/8 2h = Drive strength 3/8 3h = Drive strength 4/8 4h = Drive strength 5/8 5h = Drive strength 6/8 6h = Drive strength 7/8 7h = Highest drive strength

Table 32-22. CCP0 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
19	HYSTEN	R/W	0h	Hysteresis enable 0h = No hysteresis 1h = Hysteresis on
18	INENA	R/W	0h	Input enable 0h = Inputs 0 to connected core 1h = Inputs IO pad value to connected core
17	PIPU	R/W	0h	Pull up enable 0h = No pull up 1h = Pull up
16	PIPD	R/W	0h	Pull down enable 0h = No pull down 1h = Pull down
15-14	GSTATE	R/W	0h	GPIO Channel State 0h = G-Channel is in Unassigned State 1h = G-Channel is in Handover State 2h = G-Channle is in Connected State and not Locked (That is F field is allowed to change without going back through Unassigned state) 3h = G-Channel is in Connected State and Locked (That is F field is not allowed to change to a different non-Zero value until both G and P channels go to Unassigned)
13-8	RESERVED	R/W	0h	
7-6	PSTATE	R/W	0h	Peripheral-Analog Channel State 0h = P-Channel is in Unassigned State 1h = P-Channel is in Handover State 2h = P-Channle is in Connected State and not Locked (That is F field is allowed to change without going back through Unassigned state) 3h = P-Channel is in Connected State and Locked (That is F field is not allowed to change to a different non-Zero value until both G and P channels go to Unassigned)
5-0	RESERVED	R/W	0h	

2 CCP1 Register (Offset = 8h) [Reset = 00000000h]

CCP1 is shown in Figure 32-26 and described in Table 32-22.

Return to the Summary Table.

CCP0 PinCM register in full write region

Figure 32-26. CCP1 Register

31	30	29	28	27	26	25	24
RESERVED	GFLT	SLEW	WCOMP	WUEN	INV	HIGHZ1	HIGHZ0
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
RESERVED	DRV			HYSTEN	INENA	PIPU	PIPD
R/W-0h	R/W-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
GSTATE		RESERVED					
R/W-0h		R/W-0h					
7	6	5	4	3	2	1	0
PSTATE		RESERVED					
R/W-0h		R/W-0h					

Table 32-24. CCP1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	
30	GFLT	R/W	0h	Glitch Filter Enable 0h = No internal glitch filter 1h = Use internal glitch filter
29	SLEW	R/W	0h	Reserved Slew Rate Control 0h = No Slew Rate Control 1h = Use Slew Rate Control
28	WCOMP	R/W	0h	Wake up compare value 0h = Match '0' will wake 1h = Match '1' will wake
27	WUEN	R/W	0h	Wake up enable 0h = Wake up not enabled 1h = Wake up enabled
26	INV	R/W	0h	Invert digital input/output relative to peripheral/GPIO 0h = Input and output are non-inverted 1h = Input and output are inverted
25	HIGHZ1	R/W	0h	High-Z instead of high output 0h = Pin can be driven high 1h = Pin is tri-stated instead of driven high
24	HIGHZ0	R/W	0h	High-Z instead of low output 0h = Pin can be driven low 1h = Pin is tri-stated instead of driven low
23	RESERVED	R/W	0h	
22-20	DRV	R/W	0h	Drive strength options 0h = Lowest drive strength 1h = Drive strength 2/8 2h = Drive strength 3/8 3h = Drive strength 4/8 4h = Drive strength 5/8 5h = Drive strength 6/8 6h = Drive strength 7/8 7h = Highest drive strength

Table 32-24. CCP1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
19	HYSTEN	R/W	0h	Hysteresis enable 0h = No hysteresis 1h = Hysteresis on
18	INENA	R/W	0h	Input enable 0h = Inputs 0 to connected core 1h = Inputs IO pad value to connected core
17	PIPU	R/W	0h	Pull up enable 0h = No pull up 1h = Pull up
16	PIPD	R/W	0h	Pull down enable 0h = No pull down 1h = Pull down
15-14	GSTATE	R/W	0h	GPIO Channel State 0h = G-Channel is in Unassigned State 1h = G-Channel is in Handover State 2h = G-Channle is in Connected State and not Locked (That is F field is allowed to change without going back through Unassigned state) 3h = G-Channel is in Connected State and Locked (That is F field is not allowed to change to a different non-Zero value until both G and P channels go to Unassigned)
13-8	RESERVED	R/W	0h	
7-6	PSTATE	R/W	0h	Peripheral-Analog Channel State 0h = P-Channel is in Unassigned State 1h = P-Channel is in Handover State 2h = P-Channle is in Connected State and not Locked (That is F field is allowed to change without going back through Unassigned state) 3h = P-Channel is in Connected State and Locked (That is F field is not allowed to change to a different non-Zero value until both G and P channels go to Unassigned)
5-0	RESERVED	R/W	0h	

3 CCP2_CMPL Register (Offset = 24h) [Reset = 00000000h]

CCP2_CMPL is shown in Figure 32-27 and described in Table 32-23.

Return to the Summary Table.

CCP2_CMPL PinCM register in full write region

Figure 32-27. CCP2_CMPL Register

31	30	29	28	27	26	25	24
RESERVED	GFLT	SLEW	WCOMP	WUEN	INV	HIGHZ1	HIGHZ0
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
RESERVED	DRV			HYSTEN	INENA	PIPU	PIPD
R/W-0h	R/W-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
GSTATE		RESERVED					
R/W-0h		R/W-0h					
7	6	5	4	3	2	1	0
PSTATE		RESERVED					
R/W-0h		R/W-0h					

Table 32-26. CCP2_CMPL Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	
30	GFLT	R/W	0h	Glitch Filter Enable 0h = No internal glitch filter 1h = Use internal glitch filter
29	SLEW	R/W	0h	Reserved Slew Rate Control 0h = No Slew Rate Control 1h = Use Slew Rate Control
28	WCOMP	R/W	0h	Wake up compare value 0h = Match '0' will wake 1h = Match '1' will wake
27	WUEN	R/W	0h	Wake up enable 0h = Wake up not enabled 1h = Wake up enabled
26	INV	R/W	0h	Invert digital input/output relative to peripheral/GPIO 0h = Input and output are non-inverted 1h = Input and output are inverted
25	HIGHZ1	R/W	0h	High-Z instead of high output 0h = Pin can be driven high 1h = Pin is tri-stated instead of driven high
24	HIGHZ0	R/W	0h	High-Z instead of low output 0h = Pin can be driven low 1h = Pin is tri-stated instead of driven low
23	RESERVED	R/W	0h	
22-20	DRV	R/W	0h	Drive strength options 0h = Lowest drive strength 1h = Drive strength 2/8 2h = Drive strength 3/8 3h = Drive strength 4/8 4h = Drive strength 5/8 5h = Drive strength 6/8 6h = Drive strength 7/8 7h = Highest drive strength

Table 32-26. CCP2_CMPL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
19	HYSTEN	R/W	0h	Hysteresis enable 0h = No hysteresis 1h = Hysteresis on
18	INENA	R/W	0h	Input enable 0h = Inputs 0 to connected core 1h = Inputs IO pad value to connected core
17	PIPU	R/W	0h	Pull up enable 0h = No pull up 1h = Pull up
16	PIPD	R/W	0h	Pull down enable 0h = No pull down 1h = Pull down
15-14	GSTATE	R/W	0h	GPIO Channel State 0h = G-Channel is in Unassigned State 1h = G-Channel is in Handover State 2h = G-Channle is in Connected State and not Locked (That is F field is allowed to change without going back through Unassigned state) 3h = G-Channel is in Connected State and Locked (That is F field is not allowed to change to a different non-Zero value until both G and P channels go to Unassigned)
13-8	RESERVED	R/W	0h	
7-6	PSTATE	R/W	0h	Peripheral-Analog Channel State 0h = P-Channel is in Unassigned State 1h = P-Channel is in Handover State 2h = P-Channle is in Connected State and not Locked (That is F field is allowed to change without going back through Unassigned state) 3h = P-Channel is in Connected State and Locked (That is F field is not allowed to change to a different non-Zero value until both G and P channels go to Unassigned)
5-0	RESERVED	R/W	0h	

4 CCP3_CMPL Register (Offset = 28h) [Reset = 00000000h]

CCP3_CMPL is shown in Figure 32-28 and described in Table 32-24.

Return to the Summary Table.

CCP3_CMPL PinCM register in full write region

Figure 32-28. CCP3_CMPL Register

31	30	29	28	27	26	25	24
RESERVED	GFLT	SLEW	WCOMP	WUEN	INV	HIGHZ1	HIGHZ0
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
RESERVED	DRV			HYSTEN	INENA	PIPU	PIPD
R/W-0h	R/W-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
GSTATE		RESERVED					
R/W-0h		R/W-0h					
7	6	5	4	3	2	1	0
PSTATE		RESERVED					
R/W-0h		R/W-0h					

Table 32-28. CCP3_CMPL Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	
30	GFLT	R/W	0h	Glitch Filter Enable 0h = No internal glitch filter 1h = Use internal glitch filter
29	SLEW	R/W	0h	Reserved Slew Rate Control 0h = No Slew Rate Control 1h = Use Slew Rate Control
28	WCOMP	R/W	0h	Wake up compare value 0h = Match '0' will wake 1h = Match '1' will wake
27	WUEN	R/W	0h	Wake up enable 0h = Wake up not enabled 1h = Wake up enabled
26	INV	R/W	0h	Invert digital input/output relative to peripheral/GPIO 0h = Input and output are non-inverted 1h = Input and output are inverted
25	HIGHZ1	R/W	0h	High-Z instead of high output 0h = Pin can be driven high 1h = Pin is tri-stated instead of driven high
24	HIGHZ0	R/W	0h	High-Z instead of low output 0h = Pin can be driven low 1h = Pin is tri-stated instead of driven low
23	RESERVED	R/W	0h	
22-20	DRV	R/W	0h	Drive strength options 0h = Lowest drive strength 1h = Drive strength 2/8 2h = Drive strength 3/8 3h = Drive strength 4/8 4h = Drive strength 5/8 5h = Drive strength 6/8 6h = Drive strength 7/8 7h = Highest drive strength

Table 32-28. CCP3_CMPL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
19	HYSTEN	R/W	0h	Hysteresis enable 0h = No hysteresis 1h = Hysteresis on
18	INENA	R/W	0h	Input enable 0h = Inputs 0 to connected core 1h = Inputs IO pad value to connected core
17	PIPU	R/W	0h	Pull up enable 0h = No pull up 1h = Pull up
16	PIPD	R/W	0h	Pull down enable 0h = No pull down 1h = Pull down
15-14	GSTATE	R/W	0h	GPIO Channel State 0h = G-Channel is in Unassigned State 1h = G-Channel is in Handover State 2h = G-Channle is in Connected State and not Locked (That is F field is allowed to change without going back through Unassigned state) 3h = G-Channel is in Connected State and Locked (That is F field is not allowed to change to a different non-Zero value until both G and P channels go to Unassigned)
13-8	RESERVED	R/W	0h	
7-6	PSTATE	R/W	0h	Peripheral-Analog Channel State 0h = P-Channel is in Unassigned State 1h = P-Channel is in Handover State 2h = P-Channle is in Connected State and not Locked (That is F field is allowed to change without going back through Unassigned state) 3h = P-Channel is in Connected State and Locked (That is F field is not allowed to change to a different non-Zero value until both G and P channels go to Unassigned)
5-0	RESERVED	R/W	0h	

5 CCP0 Register (Offset = 204h) [Reset = 00000000h]

CCP0 is shown in Figure 32-29 and described in Table 32-25.

Return to the Summary Table.

FUPDATE version of CCP0

Figure 32-29. CCP0 Register

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

Table 32-30. CCP0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	RESERVED	W	0h	
27-2	IOADDR	W	0h	IO Address This is the address that corresponds to the SOC address[27:2] of the module IP instance specific IO signal in the "Full Write" subregion of the pinmux subregion.
1	LOCK	W	0h	Sets lock bit 0h = Writing this value has no effect 1h = Set channel lock bit
0	GSEL	W	0h	GPIO channel Select 0: Select the P-Channel for the F update 1: Select the G-Channel for the F update 0h = Select the P-Channel for the F update 1h = Select the G-Channel for the F update

6 CCP1 Register (Offset = 208h) [Reset = 00000000h]

CCP1 is shown in Figure 32-30 and described in Table 32-26.

Return to the Summary Table.

FUPDATE version of CCP1

Figure 32-30. CCP1 Register

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

Table 32-32. CCP1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	RESERVED	W	0h	
27-2	IOADDR	W	0h	IO Address This is the address that corresponds to the SOC address[27:2] of the module IP instance specific IO signal in the "Full Write" subregion of the pinmux subregion.
1	LOCK	W	0h	Sets lock bit 0h = Writing this value has no effect 1h = Set channel lock bit
0	GSEL	W	0h	GPIO channel Select 0: Select the P-Channel for the F update 1: Select the G-Channel for the F update 0h = Select the P-Channel for the F update 1h = Select the G-Channel for the F update

7 CCP2_CMPL Register (Offset = 224h) [Reset = 00000000h]

CCP2_CMPL is shown in Figure 32-31 and described in Table 32-27.

Return to the Summary Table.

FUPDATE version of CCP2_CMPL

Figure 32-31. CCP2_CMPL Register

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

Table 32-34. CCP2_CMPL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	RESERVED	W	0h	
27-2	IOADDR	W	0h	IO Address This is the address that corresponds to the SOC address[27:2] of the module IP instance specific IO signal in the "Full Write" subregion of the pinmux subregion.
1	LOCK	W	0h	Sets lock bit 0h = Writing this value has no effect 1h = Set channel lock bit
0	GSEL	W	0h	GPIO channel Select 0: Select the P-Channel for the F update 1: Select the G-Channel for the F update 0h = Select the P-Channel for the F update 1h = Select the G-Channel for the F update

8 CCP3_CMPL Register (Offset = 228h) [Reset = 00000000h]

CCP3_CMPL is shown in Figure 32-32 and described in Table 32-28.

Return to the Summary Table.

FUPDATE version of CCP3_CMPL

Figure 32-32. CCP3_CMPL Register

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

Table 32-36. CCP3_CMPL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	RESERVED	W	0h	
27-2	IOADDR	W	0h	IO Address This is the address that corresponds to the SOC address[27:2] of the module IP instance specific IO signal in the "Full Write" subregion of the pinmux subregion.
1	LOCK	W	0h	Sets lock bit 0h = Writing this value has no effect 1h = Set channel lock bit
0	GSEL	W	0h	GPIO channel Select 0: Select the P-Channel for the F update 1: Select the G-Channel for the F update 0h = Select the P-Channel for the F update 1h = Select the G-Channel for the F update

9 FSUB_0 Register (Offset = 400h) [Reset = 00000000h]

FSUB_0 is shown in Figure 32-33 and described in Table 32-29.

Return to the Summary Table.

Subscriber port

Figure 32-33. FSUB_0 Register

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

Table 32-38. FSUB_0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R/W	0h	
3-0	CHANID	R/W	0h	0 = disconnected. 1-15 = connected to channelID = CHANID. 0h = A value of 0 specifies that the event is not connected Fh = Consult your device datasheet as the actual allowed maximum may be less than 15.

10 FSUB_1 Register (Offset = 404h) [Reset = 00000000h]

FSUB_1 is shown in Figure 32-34 and described in Table 32-30.

Return to the Summary Table.

Subscriber port

Figure 32-34. FSUB_1 Register

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

Table 32-40. FSUB_1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R/W	0h	
3-0	CHANID	R/W	0h	0 = disconnected. 1-15 = connected to channelID = CHANID. 0h = A value of 0 specifies that the event is not connected Fh = Consult your device datasheet as the actual allowed maximum may be less than 15.

11 FPUB_0 Register (Offset = 444h) [Reset = 00000000h]

FPUB_0 is shown in Figure 32-35 and described in Table 32-31.

Return to the Summary Table.

Publisher port

Figure 32-35. FPUB_0 Register

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

Table 32-42. FPUB_0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R/W	0h	
3-0	CHANID	R/W	0h	0 = disconnected. 1-15 = connected to channelID = CHANID. 0h = A value of 0 specifies that the event is not connected Fh = Consult your device datasheet as the actual allowed maximum may be less than 15.

12 FPUB_1 Register (Offset = 448h) [Reset = 00000000h]

FPUB_1 is shown in Figure 32-36 and described in Table 32-32.

Return to the Summary Table.

Publisher port

Figure 32-36. FPUB_1 Register

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

Table 32-44. FPUB_1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R/W	0h	
3-0	CHANID	R/W	0h	0 = disconnected. 1-15 = connected to channelID = CHANID. 0h = A value of 0 specifies that the event is not connected Fh = Consult your device datasheet as the actual allowed maximum may be less than 15.

13 CPU_CONNECT_0 Register (Offset = 480h) [Reset = 00000000h]

CPU_CONNECT_0 is shown in Figure 32-37 and described in Table 32-33.

Return to the Summary Table.

Connect peripheral interrupts / publisher port (FPUB_1) to application processor

Figure 32-37. CPU_CONNECT_0 Register

31	30	29	28	27	26	25	24
RESERVED							
R/W-0h							
23	22	21	20	19	18	17	16
RESERVED							
R/W-0h							
15	14	13	12	11	10	9	8
RESERVED							
R/W-0h							
7	6	5	4	3	2	1	0
RESERVED						CPUSS0_CONN	RESERVED
R/W-0h						R/W-0h	R/W-0h

Table 32-46. CPU_CONNECT_0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R/W	0h	
1	CPUSS0_CONN	R/W	0h	CPUSS0 connect bit. 0h = The CPU is not connected. 1h = The CPU is connected.
0	RESERVED	R/W	0h	

ADVANCE INFORMATION

14 PWREN Register (Offset = 800h) [Reset = 00000000h]

PWREN is shown in Figure 32-38 and described in Table 32-34.

Return to the Summary Table.

Register to control the power state

Figure 32-38. PWREN Register

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

Table 32-48. PWREN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	KEY	W	0h	KEY to allow Power State Change 26h = KEY to allow write access to this register
23-1	RESERVED	R/W	0h	
0	ENABLE	R/WK	0h	Enable the power [EXT_GPRCM.GPRCM.PWREN.KEY] must be set to 26h to write to this bit. 0h = Disable Power 1h = Enable Power

ADVANCE INFORMATION

15 RSTCTL Register (Offset = 804h) [Reset = 00000000h]

RSTCTL is shown in Figure 32-39 and described in Table 32-35.

Return to the Summary Table.

Register to control reset assertion and de-assertion

Figure 32-39. RSTCTL Register

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

Table 32-50. RSTCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	KEY	W	0h	Unlock key B1h = KEY to allow write access to this register
23-2	RESERVED	W	0h	
1	RESETSTKYCLR	WK	0h	Clear the RESETSTKY bit in the STAT register [EXT_GPRCM.GPRCM.RSTCTL.KEY] must be set to B1h to write to this bit. 0h = Writing 0 has no effect 1h = Clear reset sticky bit
0	RESETASSERT	WK	0h	Assert reset to the peripheral [EXT_GPRCM.GPRCM.RSTCTL.KEY] must be set to B1h to write to this bit. 0h = Writing 0 has no effect 1h = Assert reset

ADVANCE INFORMATION

16 STAT Register (Offset = 814h) [Reset = 00000000h]

STAT is shown in Figure 32-40 and described in Table 32-36.

Return to the Summary Table.

peripheral enable and reset status

Figure 32-40. STAT Register

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

Table 32-52. STAT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-17	RESERVED	R	0h	
16	RESETSTKY	R	0h	This bit indicates, if the peripheral was reset, since this bit was cleared by RESETSTKYCLR in the RSTCTL register 0h = The peripheral has not been reset since this bit was last cleared by RESETSTKYCLR in the RSTCTL register 1h = The peripheral was reset since the last bit clear
15-0	RESERVED	R	0h	

17 CLKDIV Register (Offset = 1000h) [Reset = 00000000h]

CLKDIV is shown in Figure 32-41 and described in Table 32-37.

Return to the Summary Table.

This register is used to specify module-specific divide ratio of the functional clock

Figure 32-41. CLKDIV Register

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

Table 32-54. CLKDIV Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R/W	0h	
2-0	RATIO	R/W	0h	Selects divide ratio of module clock 0h = Do not divide clock source 1h = Divide clock source by 2 2h = Divide clock source by 3 3h = Divide clock source by 4 4h = Divide clock source by 5 5h = Divide clock source by 6 6h = Divide clock source by 7 7h = Divide clock source by 8

18 CLKSEL Register (Offset = 1008h) [Reset = 00000000h]

CLKSEL is shown in Figure 32-42 and described in Table 32-38.

Return to the Summary Table.

Clock Source Select Register

Figure 32-42. CLKSEL Register

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

Table 32-56. CLKSEL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-5	RESERVED	R/W	0h	
4	RESERVED	R/W	0h	Reserved
3	BUSCLK_SEL	R/W	0h	Selects BUSCLK as clock source if enabled 0h = Does not select this clock as a source 1h = Select this clock as a source
2	MFCLK_SEL	R/W	0h	Selects MFCLK as clock source if enabled 0h = Does not select this clock as a source 1h = Select this clock as a source
1	LFCLK_SEL	R/W	0h	Selects LFCLK as clock source if enabled 0h = Does not select this clock as a source 1h = Select this clock as a source
0	RESERVED	R/W	0h	

19 PDBGCTL Register (Offset = 1018h) [Reset = 00000000h]

PDBGCTL is shown in Figure 32-43 and described in Table 32-39.

Return to the Summary Table.

This register can be used by the software developer to control the behavior of the peripheral relative to the 'Core Halted' input

Figure 32-43. PDBGCTL Register

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

Table 32-58. PDBGCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R/W	0h	
1	SOFT	R/W	1h	Soft halt boundary control. This function is only available, if [FREE] is set to 'STOP' 0h = The peripheral will halt immediately, even if the resultant state will result in corruption if the system is restarted 1h = The peripheral blocks the debug freeze until it has reached a boundary where it can resume without corruption
0	FREE	R/W	1h	Free run control 0h = The peripheral freezes functionality while the Core Halted input is asserted and resumes when it is deasserted. 1h = The peripheral ignores the state of the Core Halted input

20 IIDX Register (Offset = 1020h) [Reset = 00000000h]

IIDX is shown in Figure 32-44 and described in Table 32-40.

Return to the Summary Table.

This register provides the highest priority enabled interrupt index. Value 0x00 means no event pending. Interrupt 1 is the highest priority, IIDX next highest, 4, 8, ... IIDX^31 is the least priority. That is, the least bit position that is set to 1 denotes the highest priority pending interrupt. The priority order is fixed. However, users can implement their own prioritization schemes using other registers that expose the full set of interrupts that have occurred. On each read, only one interrupt is indicated. On a read, the current interrupt (highest priority) is automatically cleared by the hardware and the corresponding interrupt flag in [RIS] and [MIS] are cleared as well. After a read from the CPU (not from the debug interface), the register is updated with the next highest priority interrupt, if none are pending, then it should display 0x0.

Figure 32-44. IIDX Register

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

Table 32-60. IIDX Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	
7-0	STAT	R	0h	Interrupt index status 00h = No interrupt pending 01h = Interrupt Source: Zero event (Z) 02h = nterrupt Source: Load event (L) 05h = Interrupt Source: Capture or compare down event (CCD0) 06h = Interrupt Source: Capture or compare down event (CCD1) 07h = Interrupt Source: Capture or compare down event (CCD2) 08h = Interrupt Source: Capture or compare down event (CCD3) 09h = Interrupt Source: Capture or compare up event (CCU0) 0Ah = Interrupt Source: Capture or compare up event (CCU1) 0Bh = Interrupt Source: Capture or compare up event (CCU2) 0Ch = Interrupt Source: Capture or compare up event (CCU3) 0Dh = Interrupt Source: Compare down event (CCD4) 0Eh = Interrupt Source: Compare down event (CCD5) 0Fh = Interrupt Source: Compare down event (CCU4) 10h = Interrupt Source: Compare down event (CCU5) 19h = Interrupt Source: Fault Event generated an interrupt. (F) 1Ah = Interrupt Source: Trigger overflow (TOV) 1Bh = Interrupt Source: Repeat Counter Zero (REPC) 1Ch = Interrupt Source: Direction Change (DC) 1Dh = Interrupt Source:QEI Incorrect state transition error (QEIERR)

21 IMASK Register (Offset = 1028h) [Reset = 00000000h]

IMASK is shown in Figure 32-45 and described in Table 32-41.

Return to the Summary Table.

Interrupt Mask. If a bit is set, then corresponding interrupt is un-masked. Un-masking the interrupt causes the raw interrupt to be visible in IIDX, as well as MIS."

Figure 32-45. IMASK Register

31	30	29	28	27	26	25	24
RESERVED			RESERVED	RESERVED	RESERVED	TOV	RESERVED
R/W-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
RESERVED							
R/W-0h							
15	14	13	12	11	10	9	8
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	CCU1	CCU0
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
RESERVED	RESERVED	CCD1	CCD0	RESERVED		L	Z
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h		R/W-0h	R/W-0h

Table 32-62. IMASK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R/W	0h	
28	RESERVED	R/W	0h	Reserved
27	RESERVED	R/W	0h	Reserved
26	RESERVED	R/W	0h	Reserved
25	TOV	R/W	0h	Trigger Overflow Event mask 0h = Disable Event 1h = Enable Event
24	RESERVED	R/W	0h	Reserved
23-16	RESERVED	R/W	0h	
15	RESERVED	R/W	0h	Reserved
14	RESERVED	R/W	0h	Reserved
13	RESERVED	R/W	0h	Reserved
12	RESERVED	R/W	0h	Reserved
11	RESERVED	R/W	0h	Reserved
10	RESERVED	R/W	0h	Reserved
9	CCU1	R/W	0h	Capture or Compare UP event mask CCP1 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
8	CCU0	R/W	0h	Capture or Compare UP event mask CCP0 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
7	RESERVED	R/W	0h	Reserved
6	RESERVED	R/W	0h	Reserved
5	CCD1	R/W	0h	Capture or Compare DN event mask CCP1 0h = Clear Interrupt Mask 1h = Set Interrupt Mask

Table 32-62. IMASK Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
4	CCD0	R/W	0h	Capture or Compare DN event mask CCP0 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
3-2	RESERVED	R/W	0h	
1	L	R/W	0h	Load Event mask 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
0	Z	R/W	0h	Zero Event mask 0h = Disable Event 1h = Enable Event

22 RIS Register (Offset = 1030h) [Reset = 00000000h]

RIS is shown in Figure 32-46 and described in Table 32-42.

Return to the Summary Table.

Raw interrupt status. Reflects all pending interrupts, regardless of masking. The RIS register allows the user to implement a poll scheme. A flag set in this register can be cleared by writing 1 to the ICLR register bit even if the corresponding IMASK bit is not enabled.

Figure 32-46. RIS Register

31	30	29	28	27	26	25	24
RESERVED			RESERVED	RESERVED	RESERVED	TOV	RESERVED
R-0h			R-0h	R-0h	R-0h	R-0h	R-0h
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	CCU1	CCU0
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
7	6	5	4	3	2	1	0
RESERVED	RESERVED	CCD1	CCD0	RESERVED		L	Z
R-0h	R-0h	R-0h	R-0h	R-0h		R-0h	R-0h

Table 32-64. RIS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R	0h	
28	RESERVED	R	0h	Reserved
27	RESERVED	R	0h	Reserved
26	RESERVED	R	0h	Reserved
25	TOV	R	0h	Trigger overflow 0h = Event Cleared 1h = Event Set
24	RESERVED	R	0h	Reserved
23-16	RESERVED	R	0h	
15	RESERVED	R	0h	Reserved
14	RESERVED	R	0h	Reserved
13	RESERVED	R	0h	Reserved
12	RESERVED	R	0h	Reserved
11	RESERVED	R	0h	Reserved
10	RESERVED	R	0h	Reserved
9	CCU1	R	0h	Capture or compare up event generated an interrupt CCP1 0h = Event Cleared 1h = Event Set
8	CCU0	R	0h	Capture or compare up event generated an interrupt CCP0 0h = Event Cleared 1h = Event Set
7	RESERVED	R	0h	Reserved
6	RESERVED	R	0h	Reserved
5	CCD1	R	0h	Capture or compare down event generated an interrupt CCP1 0h = Event Cleared 1h = Event Set

Table 32-64. RIS Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
4	CCD0	R	0h	Capture or compare down event generated an interrupt CCP0 0h = Event Cleared 1h = Event Set
3-2	RESERVED	R	0h	
1	L	R	0h	Load event generated an interrupt. 0h = Event Cleared 1h = Event Set
0	Z	R	0h	Zero event generated an interrupt. 0h = Event Cleared 1h = Event Set

23 MIS Register (Offset = 1038h) [Reset = 00000000h]

MIS is shown in Figure 32-47 and described in Table 32-43.

Return to the Summary Table.

Masked interrupt status. This is an AND of the IMASK and RIS registers.

Figure 32-47. MIS Register

31	30	29	28	27	26	25	24
RESERVED			RESERVED	RESERVED	RESERVED	TOV	RESERVED
R-0h			R-0h	R-0h	R-0h	R-0h	R-0h
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	CCU1	CCU0
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
7	6	5	4	3	2	1	0
RESERVED	RESERVED	CCD1	CCD0	RESERVED		L	Z
R-0h	R-0h	R-0h	R-0h	R-0h		R-0h	R-0h

Table 32-66. MIS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R	0h	
28	RESERVED	R	0h	Reserved
27	RESERVED	R	0h	Reserved
26	RESERVED	R	0h	Reserved
25	TOV	R	0h	Trigger overflow 0h = Event Cleared 1h = Event Set
24	RESERVED	R	0h	Reserved
23-16	RESERVED	R	0h	
15	RESERVED	R	0h	Reserved
14	RESERVED	R	0h	Reserved
13	RESERVED	R	0h	Reserved
12	RESERVED	R	0h	Reserved
11	RESERVED	R	0h	Reserved
10	RESERVED	R	0h	Reserved
9	CCU1	R	0h	Capture or compare up event generated an interrupt CCP1 0h = Event Cleared 1h = Event Set
8	CCU0	R	0h	Capture or compare up event generated an interrupt CCP0 0h = Event Cleared 1h = Event Set
7	RESERVED	R	0h	Reserved
6	RESERVED	R	0h	Reserved
5	CCD1	R	0h	Capture or compare down event generated an interrupt CCP1 0h = Event Cleared 1h = Event Set

Table 32-66. MIS Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
4	CCD0	R	0h	Capture or compare down event generated an interrupt CCP0 0h = Event Cleared 1h = Event Set
3-2	RESERVED	R	0h	
1	L	R	0h	Load event generated an interrupt. 0h = Event Cleared 1h = Event Set
0	Z	R	0h	Zero event generated an interrupt. 0h = Event Cleared 1h = Event Set

24 ISET Register (Offset = 1040h) [Reset = 00000000h]

ISET is shown in Figure 32-48 and described in Table 32-44.

Return to the Summary Table.

Interrupt set. Allows interrupts to be set by software (useful in diagnostics and safety checks). Writing a 1 to a bit in ISET will set the event and therefore the related RIS bit also gets set. If the interrupt is enabled through the mask, then the corresponding MIS bit is also set.

Figure 32-48. ISET Register

31	30	29	28	27	26	25	24
RESERVED			RESERVED	RESERVED	RESERVED	TOV	RESERVED
W-0h			W-0h	W-0h	W-0h	W-0h	W-0h
23	22	21	20	19	18	17	16
RESERVED							
W-0h							
15	14	13	12	11	10	9	8
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	CCU1	CCU0
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h
7	6	5	4	3	2	1	0
RESERVED	RESERVED	CCD1	CCD0	RESERVED		L	Z
W-0h	W-0h	W-0h	W-0h	W-0h		W-0h	W-0h

Table 32-68. ISET Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	W	0h	
28	RESERVED	W	0h	Reserved
27	RESERVED	W	0h	Reserved
26	RESERVED	W	0h	Reserved
25	TOV	W	0h	Trigger Overflow event SET 0h = Writing 0 has no effect. 1h = Event Set
24	RESERVED	W	0h	Reserved
23-16	RESERVED	W	0h	
15	RESERVED	W	0h	Reserved
14	RESERVED	W	0h	Reserved
13	RESERVED	W	0h	Reserved
12	RESERVED	W	0h	Reserved
11	RESERVED	W	0h	Reserved
10	RESERVED	W	0h	Reserved
9	CCU1	W	0h	Capture or compare up event SET 0h = Writing 0 has no effect. 1h = Event Set
8	CCU0	W	0h	Capture or compare up event SET 0h = Writing 0 has no effect. 1h = Event Set
7	RESERVED	W	0h	Reserved
6	RESERVED	W	0h	Reserved
5	CCD1	W	0h	Capture or compare down event SET 0h = Writing 0 has no effect. 1h = Event Set

Table 32-68. ISET Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
4	CCD0	W	0h	Capture or compare down event SET 0h = Writing 0 has no effect. 1h = Event Set
3-2	RESERVED	W	0h	
1	L	W	0h	Load event SET 0h = Writing 0 has no effect. 1h = Event Set
0	Z	W	0h	Zero event SET 0h = Writing 0 has no effect. 1h = Event Set

25 ICLR Register (Offset = 1048h) [Reset = 00000000h]

ICLR is shown in Figure 32-49 and described in Table 32-45.

Return to the Summary Table.

Interrupt clear. Write a 1 to clear corresponding Interrupt.

Figure 32-49. ICLR Register

31	30	29	28	27	26	25	24
RESERVED			RESERVED	RESERVED	RESERVED	TOV	RESERVED
W-0h			W-0h	W-0h	W-0h	W-0h	W-0h
23	22	21	20	19	18	17	16
RESERVED							
W-0h							
15	14	13	12	11	10	9	8
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	CCU1	CCU0
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h
7	6	5	4	3	2	1	0
RESERVED	RESERVED	CCD1	CCD0	RESERVED		L	Z
W-0h	W-0h	W-0h	W-0h	W-0h		W-0h	W-0h

Table 32-70. ICLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	W	0h	
28	RESERVED	W	0h	Reserved
27	RESERVED	W	0h	Reserved
26	RESERVED	W	0h	Reserved
25	TOV	W	0h	Trigger Overflow event CLEAR 0h = Writing 0 has no effect. 1h = Event Clear
24	RESERVED	W	0h	Reserved
23-16	RESERVED	W	0h	
15	RESERVED	W	0h	Reserved
14	RESERVED	W	0h	Reserved
13	RESERVED	W	0h	Reserved
12	RESERVED	W	0h	Reserved
11	RESERVED	W	0h	Reserved
10	RESERVED	W	0h	Reserved
9	CCU1	W	0h	Capture or compare up event CLEAR 0h = Writing 0 has no effect. 1h = Event Clear
8	CCU0	W	0h	Capture or compare up event CLEAR 0h = Writing 0 has no effect. 1h = Event Clear
7	RESERVED	W	0h	Reserved
6	RESERVED	W	0h	Reserved
5	CCD1	W	0h	Capture or compare down event CLEAR 0h = Writing 0 has no effect. 1h = Event Clear

Table 32-70. ICLR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
4	CCD0	W	0h	Capture or compare down event CLEAR 0h = Writing 0 has no effect. 1h = Event Clear
3-2	RESERVED	W	0h	
1	L	W	0h	Load event CLEAR 0h = Writing 0 has no effect. 1h = Event Clear
0	Z	W	0h	Zero event CLEAR 0h = Writing 0 has no effect. 1h = Event Clear

26 IIDX Register (Offset = 1050h) [Reset = 00000000h]

IIDX is shown in Figure 32-50 and described in Table 32-46.

Return to the Summary Table.

This register provides the highest priority enabled interrupt index. Value 0x00 means no event pending. Interrupt 1 is the highest priority, IIDX next highest, 4, 8, ... IIDX^31 is the least priority. That is, the least bit position that is set to 1 denotes the highest priority pending interrupt. The priority order is fixed. However, users can implement their own prioritization schemes using other registers that expose the full set of interrupts that have occurred. On each read, only one interrupt is indicated. On a read, the current interrupt (highest priority) is automatically cleared by the hardware and the corresponding interrupt flag in [RIS] and [MIS] are cleared as well. After a read from the CPU (not from the debug interface), the register is updated with the next highest priority interrupt, if none are pending, then it should display 0x0.

Figure 32-50. IIDX Register

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

Table 32-72. IIDX Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	
7-0	STAT	R	0h	Interrupt index status 00h = No interrupt pending 01h = Interrupt Source: Zero event (Z) 02h = nterrupt Source: Load event (L) 05h = Interrupt Source: Capture or compare down event (CCD0) 06h = Interrupt Source: Capture or compare down event (CCD1) 07h = Interrupt Source: Capture or compare down event (CCD2) 08h = Interrupt Source: Capture or compare down event (CCD3) 09h = Interrupt Source: Capture or compare up event (CCU0) 0Ah = Interrupt Source: Capture or compare up event (CCU1) 0Bh = Interrupt Source: Capture or compare up event (CCU2) 0Ch = Interrupt Source: Capture or compare up event (CCU3) 0Dh = Interrupt Source: Compare down event (CCD4) 0Eh = Interrupt Source: Compare down event (CCD5) 0Fh = Interrupt Source: Compare down event (CCU4) 10h = Interrupt Source: Compare down event (CCU5) 19h = Interrupt Source: Fault Event generated an interrupt. (F) 1Ah = Interrupt Source: Trigger overflow (TOV) 1Bh = Interrupt Source: Repeat Counter Zero (REPC) 1Ch = Interrupt Source: Direction Change (DC) 1Dh = Interrupt Source:QEI Incorrect state transition error (QEIERR)

27 IMASK Register (Offset = 1058h) [Reset = 00000000h]

IMASK is shown in Figure 32-51 and described in Table 32-47.

Return to the Summary Table.

Interrupt Mask. If a bit is set, then corresponding interrupt is un-masked. Un-masking the interrupt causes the raw interrupt to be visible in IIDX, as well as MIS."

Figure 32-51. IMASK Register

31	30	29	28	27	26	25	24
RESERVED			RESERVED	RESERVED	RESERVED	TOV	RESERVED
R/W-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
RESERVED							
R/W-0h							
15	14	13	12	11	10	9	8
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	CCU1	CCU0
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
RESERVED	RESERVED	CCD1	CCD0	RESERVED		L	Z
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h		R/W-0h	R/W-0h

Table 32-74. IMASK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R/W	0h	
28	RESERVED	R/W	0h	Reserved
27	RESERVED	R/W	0h	Reserved
26	RESERVED	R/W	0h	Reserved
25	TOV	R/W	0h	Trigger Overflow Event mask 0h = Disable Event 1h = Enable Event
24	RESERVED	R/W	0h	Reserved
23-16	RESERVED	R/W	0h	
15	RESERVED	R/W	0h	Reserved
14	RESERVED	R/W	0h	Reserved
13	RESERVED	R/W	0h	Reserved
12	RESERVED	R/W	0h	Reserved
11	RESERVED	R/W	0h	Reserved
10	RESERVED	R/W	0h	Reserved
9	CCU1	R/W	0h	Capture or Compare UP event mask CCP1 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
8	CCU0	R/W	0h	Capture or Compare UP event mask CCP0 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
7	RESERVED	R/W	0h	Reserved
6	RESERVED	R/W	0h	Reserved
5	CCD1	R/W	0h	Capture or Compare DN event mask CCP1 0h = Clear Interrupt Mask 1h = Set Interrupt Mask

Table 32-74. IMASK Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
4	CCD0	R/W	0h	Capture or Compare DN event mask CCP0 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
3-2	RESERVED	R/W	0h	
1	L	R/W	0h	Load Event mask 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
0	Z	R/W	0h	Zero Event mask 0h = Disable Event 1h = Enable Event

28 RIS Register (Offset = 1060h) [Reset = 00000000h]

RIS is shown in Figure 32-52 and described in Table 32-48.

Return to the Summary Table.

Raw interrupt status. Reflects all pending interrupts, regardless of masking. The RIS register allows the user to implement a poll scheme. A flag set in this register can be cleared by writing 1 to the ICLR register bit even if the corresponding IMASK bit is not enabled.

Figure 32-52. RIS Register

31	30	29	28	27	26	25	24
RESERVED			RESERVED	RESERVED	RESERVED	TOV	RESERVED
R-0h			R-0h	R-0h	R-0h	R-0h	R-0h
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	CCU1	CCU0
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
7	6	5	4	3	2	1	0
RESERVED	RESERVED	CCD1	CCD0	RESERVED		L	Z
R-0h	R-0h	R-0h	R-0h	R-0h		R-0h	R-0h

Table 32-76. RIS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R	0h	
28	RESERVED	R	0h	Reserved
27	RESERVED	R	0h	Reserved
26	RESERVED	R	0h	Reserved
25	TOV	R	0h	Trigger overflow 0h = Event Cleared 1h = Event Set
24	RESERVED	R	0h	Reserved
23-16	RESERVED	R	0h	
15	RESERVED	R	0h	Reserved
14	RESERVED	R	0h	Reserved
13	RESERVED	R	0h	Reserved
12	RESERVED	R	0h	Reserved
11	RESERVED	R	0h	Reserved
10	RESERVED	R	0h	Reserved
9	CCU1	R	0h	Capture or compare up event generated an interrupt CCP1 0h = Event Cleared 1h = Event Set
8	CCU0	R	0h	Capture or compare up event generated an interrupt CCP0 0h = Event Cleared 1h = Event Set
7	RESERVED	R	0h	Reserved
6	RESERVED	R	0h	Reserved
5	CCD1	R	0h	Capture or compare down event generated an interrupt CCP1 0h = Event Cleared 1h = Event Set

Table 32-76. RIS Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
4	CCD0	R	0h	Capture or compare down event generated an interrupt CCP0 0h = Event Cleared 1h = Event Set
3-2	RESERVED	R	0h	
1	L	R	0h	Load event generated an interrupt. 0h = Event Cleared 1h = Event Set
0	Z	R	0h	Zero event generated an interrupt. 0h = Event Cleared 1h = Event Set

29 MIS Register (Offset = 1068h) [Reset = 00000000h]

MIS is shown in Figure 32-53 and described in Table 32-49.

Return to the Summary Table.

Masked interrupt status. This is an AND of the IMASK and RIS registers.

Figure 32-53. MIS Register

31	30	29	28	27	26	25	24
RESERVED			RESERVED	RESERVED	RESERVED	TOV	RESERVED
R-0h			R-0h	R-0h	R-0h	R-0h	R-0h
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	CCU1	CCU0
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
7	6	5	4	3	2	1	0
RESERVED	RESERVED	CCD1	CCD0	RESERVED		L	Z
R-0h	R-0h	R-0h	R-0h	R-0h		R-0h	R-0h

Table 32-78. MIS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R	0h	
28	RESERVED	R	0h	Reserved
27	RESERVED	R	0h	Reserved
26	RESERVED	R	0h	Reserved
25	TOV	R	0h	Trigger overflow 0h = Event Cleared 1h = Event Set
24	RESERVED	R	0h	Reserved
23-16	RESERVED	R	0h	
15	RESERVED	R	0h	Reserved
14	RESERVED	R	0h	Reserved
13	RESERVED	R	0h	Reserved
12	RESERVED	R	0h	Reserved
11	RESERVED	R	0h	Reserved
10	RESERVED	R	0h	Reserved
9	CCU1	R	0h	Capture or compare up event generated an interrupt CCP1 0h = Event Cleared 1h = Event Set
8	CCU0	R	0h	Capture or compare up event generated an interrupt CCP0 0h = Event Cleared 1h = Event Set
7	RESERVED	R	0h	Reserved
6	RESERVED	R	0h	Reserved
5	CCD1	R	0h	Capture or compare down event generated an interrupt CCP1 0h = Event Cleared 1h = Event Set

Table 32-78. MIS Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
4	CCD0	R	0h	Capture or compare down event generated an interrupt CCP0 0h = Event Cleared 1h = Event Set
3-2	RESERVED	R	0h	
1	L	R	0h	Load event generated an interrupt. 0h = Event Cleared 1h = Event Set
0	Z	R	0h	Zero event generated an interrupt. 0h = Event Cleared 1h = Event Set

30 ISET Register (Offset = 1070h) [Reset = 00000000h]

ISET is shown in Figure 32-54 and described in Table 32-50.

Return to the Summary Table.

Interrupt set. Allows interrupts to be set by software (useful in diagnostics and safety checks). Writing a 1 to a bit in ISET will set the event and therefore the related RIS bit also gets set. If the interrupt is enabled through the mask, then the corresponding MIS bit is also set.

Figure 32-54. ISET Register

31	30	29	28	27	26	25	24
RESERVED			RESERVED	RESERVED	RESERVED	TOV	RESERVED
W-0h			W-0h	W-0h	W-0h	W-0h	W-0h
23	22	21	20	19	18	17	16
RESERVED							
W-0h							
15	14	13	12	11	10	9	8
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	CCU1	CCU0
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h
7	6	5	4	3	2	1	0
RESERVED	RESERVED	CCD1	CCD0	RESERVED		L	Z
W-0h	W-0h	W-0h	W-0h	W-0h		W-0h	W-0h

Table 32-80. ISET Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	W	0h	
28	RESERVED	W	0h	Reserved
27	RESERVED	W	0h	Reserved
26	RESERVED	W	0h	Reserved
25	TOV	W	0h	Trigger Overflow event SET 0h = Writing 0 has no effect. 1h = Event Set
24	RESERVED	W	0h	Reserved
23-16	RESERVED	W	0h	
15	RESERVED	W	0h	Reserved
14	RESERVED	W	0h	Reserved
13	RESERVED	W	0h	Reserved
12	RESERVED	W	0h	Reserved
11	RESERVED	W	0h	Reserved
10	RESERVED	W	0h	Reserved
9	CCU1	W	0h	Capture or compare up event SET 0h = Writing 0 has no effect. 1h = Event Set
8	CCU0	W	0h	Capture or compare up event SET 0h = Writing 0 has no effect. 1h = Event Set
7	RESERVED	W	0h	Reserved
6	RESERVED	W	0h	Reserved
5	CCD1	W	0h	Capture or compare down event SET 0h = Writing 0 has no effect. 1h = Event Set

Table 32-80. ISET Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
4	CCD0	W	0h	Capture or compare down event SET 0h = Writing 0 has no effect. 1h = Event Set
3-2	RESERVED	W	0h	
1	L	W	0h	Load event SET 0h = Writing 0 has no effect. 1h = Event Set
0	Z	W	0h	Zero event SET 0h = Writing 0 has no effect. 1h = Event Set

31 ICLR Register (Offset = 1078h) [Reset = 00000000h]

ICLR is shown in Figure 32-55 and described in Table 32-51.

Return to the Summary Table.

Interrupt clear. Write a 1 to clear corresponding Interrupt.

Figure 32-55. ICLR Register

31	30	29	28	27	26	25	24
RESERVED			RESERVED	RESERVED	RESERVED	TOV	RESERVED
W-0h			W-0h	W-0h	W-0h	W-0h	W-0h
23	22	21	20	19	18	17	16
RESERVED							
W-0h							
15	14	13	12	11	10	9	8
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	CCU1	CCU0
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h
7	6	5	4	3	2	1	0
RESERVED	RESERVED	CCD1	CCD0	RESERVED		L	Z
W-0h	W-0h	W-0h	W-0h	W-0h		W-0h	W-0h

Table 32-82. ICLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	W	0h	
28	RESERVED	W	0h	Reserved
27	RESERVED	W	0h	Reserved
26	RESERVED	W	0h	Reserved
25	TOV	W	0h	Trigger Overflow event CLEAR 0h = Writing 0 has no effect. 1h = Event Clear
24	RESERVED	W	0h	Reserved
23-16	RESERVED	W	0h	
15	RESERVED	W	0h	Reserved
14	RESERVED	W	0h	Reserved
13	RESERVED	W	0h	Reserved
12	RESERVED	W	0h	Reserved
11	RESERVED	W	0h	Reserved
10	RESERVED	W	0h	Reserved
9	CCU1	W	0h	Capture or compare up event CLEAR 0h = Writing 0 has no effect. 1h = Event Clear
8	CCU0	W	0h	Capture or compare up event CLEAR 0h = Writing 0 has no effect. 1h = Event Clear
7	RESERVED	W	0h	Reserved
6	RESERVED	W	0h	Reserved
5	CCD1	W	0h	Capture or compare down event CLEAR 0h = Writing 0 has no effect. 1h = Event Clear

Table 32-82. ICLR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
4	CCD0	W	0h	Capture or compare down event CLEAR 0h = Writing 0 has no effect. 1h = Event Clear
3-2	RESERVED	W	0h	
1	L	W	0h	Load event CLEAR 0h = Writing 0 has no effect. 1h = Event Clear
0	Z	W	0h	Zero event CLEAR 0h = Writing 0 has no effect. 1h = Event Clear

32 IIDX Register (Offset = 1080h) [Reset = 00000000h]

IIDX is shown in Figure 32-56 and described in Table 32-52.

Return to the Summary Table.

This register provides the highest priority enabled interrupt index. Value 0x00 means no event pending. Interrupt 1 is the highest priority, IIDX next highest, 4, 8, ... IIDX^31 is the least priority. That is, the least bit position that is set to 1 denotes the highest priority pending interrupt. The priority order is fixed. However, users can implement their own prioritization schemes using other registers that expose the full set of interrupts that have occurred. On each read, only one interrupt is indicated. On a read, the current interrupt (highest priority) is automatically cleared by the hardware and the corresponding interrupt flag in [RIS] and [MIS] are cleared as well. After a read from the CPU (not from the debug interface), the register is updated with the next highest priority interrupt, if none are pending, then it should display 0x0.

Figure 32-56. IIDX Register

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

Table 32-84. IIDX Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	
7-0	STAT	R	0h	Interrupt index status 00h = No interrupt pending 01h = Interrupt Source: Zero event (Z) 02h = nterrupt Source: Load event (L) 05h = Interrupt Source: Capture or compare down event (CCD0) 06h = Interrupt Source: Capture or compare down event (CCD1) 07h = Interrupt Source: Capture or compare down event (CCD2) 08h = Interrupt Source: Capture or compare down event (CCD3) 09h = Interrupt Source: Capture or compare up event (CCU0) 0Ah = Interrupt Source: Capture or compare up event (CCU1) 0Bh = Interrupt Source: Capture or compare up event (CCU2) 0Ch = Interrupt Source: Capture or compare up event (CCU3) 0Dh = Interrupt Source: Compare down event (CCD4) 0Eh = Interrupt Source: Compare down event (CCD5) 0Fh = Interrupt Source: Compare down event (CCU4) 10h = Interrupt Source: Compare down event (CCU5) 19h = Interrupt Source: Fault Event generated an interrupt. (F) 1Ah = Interrupt Source: Trigger overflow (TOV) 1Bh = Interrupt Source: Repeat Counter Zero (REPC) 1Ch = Interrupt Source: Direction Change (DC) 1Dh = Interrupt Source:QEI Incorrect state transition error (QEIERR)

33 IMASK Register (Offset = 1088h) [Reset = 00000000h]

IMASK is shown in Figure 32-57 and described in Table 32-53.

Return to the Summary Table.

Interrupt Mask. If a bit is set, then corresponding interrupt is un-masked. Un-masking the interrupt causes the raw interrupt to be visible in IIDX, as well as MIS."

Figure 32-57. IMASK Register

31	30	29	28	27	26	25	24
RESERVED			RESERVED	RESERVED	RESERVED	TOV	RESERVED
R/W-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
RESERVED							
R/W-0h							
15	14	13	12	11	10	9	8
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	CCU1	CCU0
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
RESERVED	RESERVED	CCD1	CCD0	RESERVED		L	Z
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h		R/W-0h	R/W-0h

Table 32-86. IMASK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R/W	0h	
28	RESERVED	R/W	0h	Reserved
27	RESERVED	R/W	0h	Reserved
26	RESERVED	R/W	0h	Reserved
25	TOV	R/W	0h	Trigger Overflow Event mask 0h = Disable Event 1h = Enable Event
24	RESERVED	R/W	0h	Reserved
23-16	RESERVED	R/W	0h	
15	RESERVED	R/W	0h	Reserved
14	RESERVED	R/W	0h	Reserved
13	RESERVED	R/W	0h	Reserved
12	RESERVED	R/W	0h	Reserved
11	RESERVED	R/W	0h	Reserved
10	RESERVED	R/W	0h	Reserved
9	CCU1	R/W	0h	Capture or Compare UP event mask CCP1 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
8	CCU0	R/W	0h	Capture or Compare UP event mask CCP0 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
7	RESERVED	R/W	0h	Reserved
6	RESERVED	R/W	0h	Reserved
5	CCD1	R/W	0h	Capture or Compare DN event mask CCP1 0h = Clear Interrupt Mask 1h = Set Interrupt Mask

Table 32-86. IMASK Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
4	CCD0	R/W	0h	Capture or Compare DN event mask CCP0 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
3-2	RESERVED	R/W	0h	
1	L	R/W	0h	Load Event mask 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
0	Z	R/W	0h	Zero Event mask 0h = Disable Event 1h = Enable Event

34 RIS Register (Offset = 1090h) [Reset = 00000000h]

RIS is shown in Figure 32-58 and described in Table 32-54.

Return to the Summary Table.

Raw interrupt status. Reflects all pending interrupts, regardless of masking. The RIS register allows the user to implement a poll scheme. A flag set in this register can be cleared by writing 1 to the ICLR register bit even if the corresponding IMASK bit is not enabled.

Figure 32-58. RIS Register

31	30	29	28	27	26	25	24
RESERVED			RESERVED	RESERVED	RESERVED	TOV	RESERVED
R-0h			R-0h	R-0h	R-0h	R-0h	R-0h
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	CCU1	CCU0
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
7	6	5	4	3	2	1	0
RESERVED	RESERVED	CCD1	CCD0	RESERVED		L	Z
R-0h	R-0h	R-0h	R-0h	R-0h		R-0h	R-0h

Table 32-88. RIS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R	0h	
28	RESERVED	R	0h	Reserved
27	RESERVED	R	0h	Reserved
26	RESERVED	R	0h	Reserved
25	TOV	R	0h	Trigger overflow 0h = Event Cleared 1h = Event Set
24	RESERVED	R	0h	Reserved
23-16	RESERVED	R	0h	
15	RESERVED	R	0h	Reserved
14	RESERVED	R	0h	Reserved
13	RESERVED	R	0h	Reserved
12	RESERVED	R	0h	Reserved
11	RESERVED	R	0h	Reserved
10	RESERVED	R	0h	Reserved
9	CCU1	R	0h	Capture or compare up event generated an interrupt CCP1 0h = Event Cleared 1h = Event Set
8	CCU0	R	0h	Capture or compare up event generated an interrupt CCP0 0h = Event Cleared 1h = Event Set
7	RESERVED	R	0h	Reserved
6	RESERVED	R	0h	Reserved
5	CCD1	R	0h	Capture or compare down event generated an interrupt CCP1 0h = Event Cleared 1h = Event Set

Table 32-88. RIS Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
4	CCD0	R	0h	Capture or compare down event generated an interrupt CCP0 0h = Event Cleared 1h = Event Set
3-2	RESERVED	R	0h	
1	L	R	0h	Load event generated an interrupt. 0h = Event Cleared 1h = Event Set
0	Z	R	0h	Zero event generated an interrupt. 0h = Event Cleared 1h = Event Set

35 MIS Register (Offset = 1098h) [Reset = 00000000h]

MIS is shown in Figure 32-59 and described in Table 32-55.

Return to the Summary Table.

Masked interrupt status. This is an AND of the IMASK and RIS registers.

Figure 32-59. MIS Register

31	30	29	28	27	26	25	24
RESERVED			RESERVED	RESERVED	RESERVED	TOV	RESERVED
R-0h			R-0h	R-0h	R-0h	R-0h	R-0h
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	CCU1	CCU0
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
7	6	5	4	3	2	1	0
RESERVED	RESERVED	CCD1	CCD0	RESERVED		L	Z
R-0h	R-0h	R-0h	R-0h	R-0h		R-0h	R-0h

Table 32-90. MIS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R	0h	
28	RESERVED	R	0h	Reserved
27	RESERVED	R	0h	Reserved
26	RESERVED	R	0h	Reserved
25	TOV	R	0h	Trigger overflow 0h = Event Cleared 1h = Event Set
24	RESERVED	R	0h	Reserved
23-16	RESERVED	R	0h	
15	RESERVED	R	0h	Reserved
14	RESERVED	R	0h	Reserved
13	RESERVED	R	0h	Reserved
12	RESERVED	R	0h	Reserved
11	RESERVED	R	0h	Reserved
10	RESERVED	R	0h	Reserved
9	CCU1	R	0h	Capture or compare up event generated an interrupt CCP1 0h = Event Cleared 1h = Event Set
8	CCU0	R	0h	Capture or compare up event generated an interrupt CCP0 0h = Event Cleared 1h = Event Set
7	RESERVED	R	0h	Reserved
6	RESERVED	R	0h	Reserved
5	CCD1	R	0h	Capture or compare down event generated an interrupt CCP1 0h = Event Cleared 1h = Event Set

Table 32-90. MIS Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
4	CCD0	R	0h	Capture or compare down event generated an interrupt CCP0 0h = Event Cleared 1h = Event Set
3-2	RESERVED	R	0h	
1	L	R	0h	Load event generated an interrupt. 0h = Event Cleared 1h = Event Set
0	Z	R	0h	Zero event generated an interrupt. 0h = Event Cleared 1h = Event Set

36 ISET Register (Offset = 10A0h) [Reset = 00000000h]

ISET is shown in Figure 32-60 and described in Table 32-56.

Return to the Summary Table.

Interrupt set. Allows interrupts to be set by software (useful in diagnostics and safety checks). Writing a 1 to a bit in ISET will set the event and therefore the related RIS bit also gets set. If the interrupt is enabled through the mask, then the corresponding MIS bit is also set.

Figure 32-60. ISET Register

31	30	29	28	27	26	25	24
RESERVED			RESERVED	RESERVED	RESERVED	TOV	RESERVED
W-0h			W-0h	W-0h	W-0h	W-0h	W-0h
23	22	21	20	19	18	17	16
RESERVED							
W-0h							
15	14	13	12	11	10	9	8
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	CCU1	CCU0
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h
7	6	5	4	3	2	1	0
RESERVED	RESERVED	CCD1	CCD0	RESERVED		L	Z
W-0h	W-0h	W-0h	W-0h	W-0h		W-0h	W-0h

Table 32-92. ISET Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	W	0h	
28	RESERVED	W	0h	Reserved
27	RESERVED	W	0h	Reserved
26	RESERVED	W	0h	Reserved
25	TOV	W	0h	Trigger Overflow event SET 0h = Writing 0 has no effect. 1h = Event Set
24	RESERVED	W	0h	Reserved
23-16	RESERVED	W	0h	
15	RESERVED	W	0h	Reserved
14	RESERVED	W	0h	Reserved
13	RESERVED	W	0h	Reserved
12	RESERVED	W	0h	Reserved
11	RESERVED	W	0h	Reserved
10	RESERVED	W	0h	Reserved
9	CCU1	W	0h	Capture or compare up event SET 0h = Writing 0 has no effect. 1h = Event Set
8	CCU0	W	0h	Capture or compare up event SET 0h = Writing 0 has no effect. 1h = Event Set
7	RESERVED	W	0h	Reserved
6	RESERVED	W	0h	Reserved
5	CCD1	W	0h	Capture or compare down event SET 0h = Writing 0 has no effect. 1h = Event Set

Table 32-92. ISET Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
4	CCD0	W	0h	Capture or compare down event SET 0h = Writing 0 has no effect. 1h = Event Set
3-2	RESERVED	W	0h	
1	L	W	0h	Load event SET 0h = Writing 0 has no effect. 1h = Event Set
0	Z	W	0h	Zero event SET 0h = Writing 0 has no effect. 1h = Event Set

37 ICLR Register (Offset = 10A8h) [Reset = 00000000h]

ICLR is shown in Figure 32-61 and described in Table 32-57.

Return to the Summary Table.

Interrupt clear. Write a 1 to clear corresponding Interrupt.

Figure 32-61. ICLR Register

31	30	29	28	27	26	25	24
RESERVED			RESERVED	RESERVED	RESERVED	TOV	RESERVED
W-0h			W-0h	W-0h	W-0h	W-0h	W-0h
23	22	21	20	19	18	17	16
RESERVED							
W-0h							
15	14	13	12	11	10	9	8
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	CCU1	CCU0
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h
7	6	5	4	3	2	1	0
RESERVED	RESERVED	CCD1	CCD0	RESERVED		L	Z
W-0h	W-0h	W-0h	W-0h	W-0h		W-0h	W-0h

Table 32-94. ICLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	W	0h	
28	RESERVED	W	0h	Reserved
27	RESERVED	W	0h	Reserved
26	RESERVED	W	0h	Reserved
25	TOV	W	0h	Trigger Overflow event CLEAR 0h = Writing 0 has no effect. 1h = Event Clear
24	RESERVED	W	0h	Reserved
23-16	RESERVED	W	0h	
15	RESERVED	W	0h	Reserved
14	RESERVED	W	0h	Reserved
13	RESERVED	W	0h	Reserved
12	RESERVED	W	0h	Reserved
11	RESERVED	W	0h	Reserved
10	RESERVED	W	0h	Reserved
9	CCU1	W	0h	Capture or compare up event CLEAR 0h = Writing 0 has no effect. 1h = Event Clear
8	CCU0	W	0h	Capture or compare up event CLEAR 0h = Writing 0 has no effect. 1h = Event Clear
7	RESERVED	W	0h	Reserved
6	RESERVED	W	0h	Reserved
5	CCD1	W	0h	Capture or compare down event CLEAR 0h = Writing 0 has no effect. 1h = Event Clear

Table 32-94. ICLR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
4	CCD0	W	0h	Capture or compare down event CLEAR 0h = Writing 0 has no effect. 1h = Event Clear
3-2	RESERVED	W	0h	
1	L	W	0h	Load event CLEAR 0h = Writing 0 has no effect. 1h = Event Clear
0	Z	W	0h	Zero event CLEAR 0h = Writing 0 has no effect. 1h = Event Clear

38 EVT_MODE Register (Offset = 10E0h) [Reset = 00000029h]

EVT_MODE is shown in Figure 32-62 and described in Table 32-58.

Return to the Summary Table.

Event mode register. It is used to select whether each line is disabled, in software mode (software clears the RIS) or in hardware mode (hardware clears the RIS)

Figure 32-62. EVT_MODE Register

31	30	29	28	27	26	25	24
RESERVED							
R/W-0h							
23	22	21	20	19	18	17	16
RESERVED							
R/W-0h							
15	14	13	12	11	10	9	8
RESERVED							
R/W-0h							
7	6	5	4	3	2	1	0
RESERVED		EVT2_CFG		EVT1_CFG		EVT0_CFG	
R/W-0h		R-2h		R-2h		R-1h	

Table 32-96. EVT_MODE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-6	RESERVED	R/W	0h	
5-4	EVT2_CFG	R	2h	Event line mode select for event corresponding to GEN_EVENT1 0h = The interrupt or event line is disabled. 1h = The interrupt or event line is in software mode. Software must clear the RIS. 2h = The interrupt or event line is in hardware mode. The hardware (another module) clears automatically the associated RIS flag.
3-2	EVT1_CFG	R	2h	Event line mode select for event corresponding to GEN_EVENT0 0h = The interrupt or event line is disabled. 1h = The interrupt or event line is in software mode. Software must clear the RIS. 2h = The interrupt or event line is in hardware mode. The hardware (another module) clears automatically the associated RIS flag.
1-0	EVT0_CFG	R	1h	Event line mode select for event corresponding to CPU_INT 0h = The interrupt or event line is disabled. 1h = The interrupt or event line is in software mode. Software must clear the RIS. 2h = The interrupt or event line is in hardware mode. The hardware (another module) clears automatically the associated RIS flag.

39 DESC Register (Offset = 10FCh) [Reset = 00000000h]

DESC is shown in Figure 32-63 and described in Table 32-59.

Return to the Summary Table.

This register identifies the peripheral and its exact version.

Figure 32-63. DESC Register

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

Table 32-98. DESC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	MODULEID	R	1111h	Module identification contains a unique peripheral identification number. The assignments are maintained in a central database for all of the platform modules to ensure uniqueness. 0h = Smallest value FFFFh = Highest possible value
15-12	FEATUREVER	R	0h	Feature Set for the module *instance* 0h = Smallest value Fh = Highest possible value
11-8	INSTNUM	R	0h	Instance Number within the device. This will be a parameter to the RTL for modules that can have multiple instances 0h = Smallest value Fh = Highest possible value
7-4	MAJREV	R	0h	Major rev of the IP 0h = Smallest value Fh = Highest possible value
3-0	MINREV	R	0h	Minor rev of the IP 0h = Smallest value Fh = Highest possible value

40 CCPD Register (Offset = 1100h) [Reset = 00000000h]

CCPD is shown in Figure 32-64 and described in Table 32-60.

Return to the Summary Table.

CCP Direction. Controls whether CCP is used as an input or an output.

Figure 32-64. CCPD Register

31	30	29	28	27	26	25	24
RESERVED							
R/W-0h							
23	22	21	20	19	18	17	16
RESERVED							
R/W-0h							
15	14	13	12	11	10	9	8
RESERVED							
R/W-0h							
7	6	5	4	3	2	1	0
RESERVED				RESERVED	RESERVED	C0CCP1	C0CCP0
R/W-0h				R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 32-100. CCPD Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R/W	0h	
3	RESERVED	R/W	0h	Reserved
2	RESERVED	R/W	0h	Reserved
1	C0CCP1	R/W	0h	CCP1 direction 0h = Input 1h = Output
0	C0CCP0	R/W	0h	CCP0 direction 0h = Input 1h = Output

41 ODIS Register (Offset = 1104h) [Reset = 00000000h]

ODIS is shown in Figure 32-65 and described in Table 32-61.

Return to the Summary Table.

The ODIS register output is inverted and then ANDed with the output signal selected by the OCTL register CCPO field (before conditional inversion) to allow software the ability to hold the CCP output low during configuration or shutdown.

Figure 32-65. ODIS Register

31	30	29	28	27	26	25	24
RESERVED							
R/W-0h							
23	22	21	20	19	18	17	16
RESERVED							
R/W-0h							
15	14	13	12	11	10	9	8
RESERVED							
R/W-0h							
7	6	5	4	3	2	1	0
RESERVED				RESERVED	RESERVED	C0CCP1	C0CCP0
R/W-0h				R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 32-102. ODIS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R/W	0h	
3	RESERVED	R/W	0h	Reserved
2	RESERVED	R/W	0h	Reserved
1	C0CCP1	R/W	0h	Counter CCP1 Disable Mask Defines whether CCP0 of Counter n is forced low or not 0h = Output function as selected by the OCTL register CCPO field are provided to output inversion block. 1h = CCP output occpout[1] is forced low.
0	C0CCP0	R/W	0h	Counter CCP0 Disable Mask Defines whether CCP0 of Counter n is forced low or not 0h = Output function as selected by the OCTL register CCPO field are provided to output inversion block. 1h = CCP output occpout[0] is forced low.

42 CCLKCTL Register (Offset = 1108h) [Reset = 00000000h]

CCLKCTL is shown in Figure 32-66 and described in Table 32-62.

Return to the Summary Table.

The CCLKCTL register provides a SW mechanism for gating the TIMER clock if the module is expected not to be used but the power domain is alive. This effectively puts the IP in an IDLE state

Figure 32-66. CCLKCTL Register

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

Table 32-104. CCLKCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R/W	0h	
0	CLKEN	R/W	0h	Clock Enable Disables the clock gating to the module. SW has to explicitly program the value to 0 to gate the clock. 0h = Clock is disabled. 1h = Clock is enabled

ADVANCE INFORMATION

43 CPS Register (Offset = 110Ch) [Reset = 00000000h]

CPS is shown in Figure 32-67 and described in Table 32-63.

Return to the Summary Table.

The CPS register provides the value for the clock pre-scaler.

Figure 32-67. CPS Register

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

Table 32-106. CPS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R/W	0h	
7-0	PCNT	R/W	0h	Pre-Scale Count This field specifies the pre-scale count value. The selected TIMCLK source is divided by a value of (PCNT+1). A PCNT value of 0 divides TIMCLK by 1, effectively bypassing the divider. A PCNT value of greater than 0 divides the TIMCLK source generating a slower clock 0h = Minimum value FFh = Maximum Value

44 CPSV Register (Offset = 1110h) [Reset = 00000000h]

CPSV is shown in Figure 32-68 and described in Table 32-64.

Return to the Summary Table.

The CPSV register provides the ability to read the current clock prescale count value.

Figure 32-68. CPSV Register

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

Table 32-108. CPSV Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	
7-0	CPSVAL	R	0h	Current Prescale Count Value 0h = Minimum value FFh = Maximum Value

45 CTTRIGCTL Register (Offset = 1114h) [Reset = 00000000h]

CTTRIGCTL is shown in Figure 32-69 and described in Table 32-65.

Return to the Summary Table.

Cross Timer Trigger Control Register
This register is used to control the cross trigger connections for enables and faults of different timer instances in the same power domain. Please refer to sections Timer Module Cross Trigger (In/Out) and Fault Cross Triggering for details.

Figure 32-69. CTTRIGCTL Register

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

Table 32-110. CTTRIGCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-20	RESERVED	R/W	0h	
19-16	EVTCTTRIGSEL	R/W	0h	Used to Select the subscriber port that should be used for input cross trigger. 0h = Use FSUB0 as cross trigger source. 1h = Use FSUB1 as cross trigger source. 2h = Use Zero event as cross trigger source. 3h = Use Load event as cross trigger source. 4h = Use CCD0 event as cross trigger source. 5h = Use CCD1 event as cross trigger source. 6h = Use CCD2 event as cross trigger source. 7h = Use CCD3 event as cross trigger source. 8h = Use CCU0 event as cross trigger source. 9h = Use CCU1 event as cross trigger source. Ah = Use CCU2 event as cross trigger source. Bh = Use CCU3 event as cross trigger source.
15-2	RESERVED	R/W	0h	
1	EVTCTEN	R/W	0h	Enable the Input Trigger Conditions to the Timer module as a condition for Cross Triggers. 0h = Cross trigger generation disabled. 1h = Cross trigger generation enabled
0	CTEN	R/W	0h	Timer Cross trigger enable. This field is used to enable whether the SW or HW logic can generate a timer cross trigger event in the system. These cross triggers are connected to the respective timer trigger in of the other timer IPs in the SOC power domain. The timer cross trigger is essentially the combined logic of the HW and SW conditions controlling EN bit in the CTRCTL register. 0h = Cross trigger generation disabled. 1h = Cross trigger generation enabled

ADVANCE INFORMATION

46 CTTRIG Register (Offset = 111Ch) [Reset = 00000000h]

CTTRIG is shown in Figure 32-70 and described in Table 32-66.

Return to the Summary Table.

Cross Timer Trigger Register
This register is used to trigger the timer instances connected and enabled using CTTRIGCTL and CTTRIGMSK registers.

Figure 32-70. CTTRIG Register

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

Table 32-112. CTTRIG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	W	0h	
0	TRIG	W	0h	Generate Cross Trigger This bit when programmed will generate a synchronized trigger condition all the cross trigger enabled Timer instances including current timer instance. 0h = Cross trigger generation disabled 1h = Generate Cross trigger pulse

ADVANCE INFORMATION

47 CTR Register (Offset = 1800h) [Reset = 00000000h]

CTR is shown in Figure 32-71 and described in Table 32-67.

Return to the Summary Table.

This is the TIMER counter register.
This can be set by SW. However, the writes will be unpredictable if the software tries to set a value while the counter is running.

Figure 32-71. CTR Register

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

Table 32-114. CTR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R/W	0h	
15-0	CCTR	R/W	0h	Current Counter value 0h = Minimum value FFFFFFFFh = Maximum Value

ADVANCE INFORMATION

48 CTRCTL Register (Offset = 1804h) [Reset = 0000FF80h]

CTRCTL is shown in Figure 32-72 and described in Table 32-68.

Return to the Summary Table.

This register provides control over the counter operation.
The configuration can change as well as setting the EN bit in a single write.
There is no requirement to change the configuration first and then do an additional write to set the EN bit.

Figure 32-72. CTRCTL Register

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

Table 32-116. CTRCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	RESERVED	R/W	0h	
29-28	CVAE	R/W	0h	Counter Value After Enable. This field specifies the initialization condition of the counter when the EN bit is changed from 0 to 1 by a write to the CTRCTL register. Note that an external event can also cause the EN bit to go active. 0h = The counter is set to the LOAD register value 1h = The counter value is unchanged from its current value which could have been initialized by software 2h = The counter is set to zero
27-25	RESERVED	R/W	0h	
24	RESERVED	R/W	0h	Reserved
23	RESERVED	R/W	0h	Reserved
22-20	RESERVED	R/W	0h	
19	RESERVED	R/W	0h	Reserved
18	RESERVED	R/W	0h	Reserved
17	DRB	R/W	0h	Debug Resume Behavior This bit specifies what the device does following the release/exit of debug mode. 0h = Resume counting 1h = Perform the action as specified by the CVAE field.
16	RESERVED	R/W	0h	

ADVANCE INFORMATION

Table 32-116. CTRCTL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
15-13	CZC	R/W	7h	Counter Zero Control This field specifies what controls the counter operation with respect to zeroing the counter value. Encodings 1-3 are present based on the CCPC parameter value. Bits 4-5 are present based on the HQEI parameter value. Any encodings not provided are documented as reserved. 0h = CCCTL_0 ZCOND 1h = CCCTL_1 ZCOND 2h = CCCTL_2 ZCOND This value exists when there are 4 channels. 3h = CCCTL_3 ZCOND This value exists when there are 4 channels. 4h = Controlled by 2-input QEI mode This value exists when TIMER support QEI feature. 5h = Controlled by 3-input QEI mode This value exists when TIMER support QEI feature.
12-10	CAC	R/W	7h	Counter Advance Control. This field specifies what controls the counter operation with respect to advancing (incrementing or decrementing) the counter value. Encodings 1-3 are present based on the CCPC parameter value. Bits 4-5 are present based on the HQEI parameter value. Any encodings not provided are documented as reserved. 0h = CCCTL_0 ACOND 1h = CCCTL_1 ACOND 2h = CCCTL_2 ACOND This value exists when there are 4 channels. 3h = CCCTL_3 ACOND This value exists when there are 4 channels. 4h = Controlled by 2-input QEI mode This value exists when TIMER support QEI feature. 5h = Controlled by 3-input QEI mode This value exists when TIMER support QEI feature.
9-7	CLC	R/W	7h	Counter Load Control. This field specifies what controls the counter operation with respect to setting the counter to the LD register value. Encodings 1-3 are present based on the CCPC parameter value. Bits 4-5 are present based on the HQEI parameter value. Any encodings not provided are documented as reserved. 0h = CCCTL_0 LCOND 1h = CCCTL_1 LCOND 2h = CCCTL_2 LCOND This value exists when there are 4 channels. 3h = CCCTL_3 LCOND This value exists when there are 4 channels. 4h = Controlled by 2 input QEI mode. This value exists when TIMER support QEI feature. 5h = Controlled by 3 input QEI mode. This value exists when TIMER support QEI feature.
6	RESERVED	R/W	0h	
5-4	CM	R/W	0h	Count Mode 0h = Down 1h = Up/Down 2h = Counter counts up.

Table 32-116. CTRCTL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3-1	REPEAT	R/W	0h	Repeat. The repeat bit controls whether the counter continues to advance following a zero event, or the exiting of a debug or fault condition. If counting down, a zero event is followed by a load at the next advance condition. If counting up-down, a zero event is followed by an advance event (+1). The intent of encoding 3 is that if the debug condition is in effect, the generation of the load pulse is deferred until the debug condition is over. This allows the counter to reach zero before counting is suspended. 0h = Does not automatically advance following a zero event. 1h = Continues to advance following a zero event. 2h = Reserved 3h = Continues to advance following a zero event if the debug mode is not in effect, or following the release of the debug mode. 4h = Reserved
0	EN	R/W	0h	Counter Enable. This bit allows the timer to advance This bit is automatically cleared if REPEAT=0 (do not automatically reload) and the counter value equals zero. CPU Write: A register write that sets the EN bit, the counter value is set per the CVAE value. Hardware: This bit may also be set as the result of an LCOND or ZCOND condition being met and the counter value changed to the load value or zero value, respectively. 0h = Disabled 1h = Enabled

49 LOAD Register (Offset = 1808h) [Reset = 00000000h]

LOAD is shown in Figure 32-73 and described in Table 32-69.

Return to the Summary Table.

The contents of LOAD register are copied to CTR on any operation designated to do a "LOAD". The LOAD is used to compare with the CTR for generating a "Load Event" that can be used for interrupt, trigger, or signal generator actions.

Figure 32-73. LOAD Register

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

Table 32-118. LOAD Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R/W	0h	
15-0	LD	R/W	0h	Load Value 0h = Minimum value FFFFFFFFh = Maximum Value

ADVANCE INFORMATION

50 CC_01_y Register (Offset = 1810h + formula) [Reset = 00000000h]

CC_01_y is shown in Figure 32-74 and described in Table 32-70.

Return to the Summary Table.

The CC_01 register is a register that can be used as either a capture register, to capture the next CTR value on an event, or a compare to the current CTR to create an event. It cannot operate concurrently as both. There are two Capture-Compare slices of hardware for each counter, hence there are two CC_01 registers per timer. On a capture event, the next value of the CTR is loaded so that CTR and CC_01 (which captured) will be equal on the cycle that an interrupt or trigger is created from the capture action.

Offset = 1810h + (y * 4h); where y = 0h to 1h

Figure 32-74. CC_01_y Register

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

Table 32-120. CC_01_y Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R/W	0h	
15-0	CCVAL	R/W	0h	Capture or compare value 0h = Minimum value FFFFFFFFh = Maximum Value

51 CCCTL_01_y Register (Offset = 1830h + formula) [Reset = 00000000h]

CCCTL_01_y is shown in Figure 32-75 and described in Table 32-71.

Return to the Summary Table.

The CCCTL_01 registers control the operations of the respective CC registers and the counter.

Offset = 1830h + (y * 4h); where y = 0h to 1h

Figure 32-75. CCCTL_01_y Register

31	30	29	28	27	26	25	24
CC2SELD			CCACTUPD			RESERVED	CC2SELU
R/W-0h			R/W-0h			R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
CC2SELU		RESERVED	RESERVED			COC	RESERVED
R/W-0h		R/W-0h	R/W-0h			R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED	ZCOND			RESERVED	LCOND		
R/W-0h	R/W-0h			R/W-0h	R/W-0h		
7	6	5	4	3	2	1	0
RESERVED	ACOND			RESERVED	CCOND		
R/W-0h	R/W-0h			R/W-0h	R/W-0h		

Table 32-122. CCCTL_01_y Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	CC2SELD	R/W	0h	Selects the source second CCD event. 0h = Selects CCD from CC0. 1h = Selects CCD from CC1. 2h = Selects CCD from CC2. 3h = Selects CCD from CC3. 4h = Selects CCD from CC4. 5h = Selects CCD from CC5.

Table 32-122. CCCTL_01_y Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
28-26	CCACTUPD	R/W	0h	CCACT shadow register Update Method This field controls how updates to the CCACT shadow register are performed 0h = Value written to the CCACT register has immediate effect. 1h = Following a zero event (CTR=0) Writes to the CCACTx_y register are stored in shadow register and transferred to CCACTx_y in the TIMCLK cycle following CTR equals 0. 2h = Following a CCD event (CTR=CC_xy) Writes to the CCACTx_y register are stored in shadow register and transferred to CCACTx_y in the TIMCLK cycle following CTR equals the CCx_y register value. 3h = Following a CCU event (CTR=CC_xy) Writes to the CCACTx_y register are stored in shadow register and transferred to CCACTx_y in the TIMCLK cycle following CTR equals the CCx_y register value. 4h = Following a zero event (CTR=0) or load event (CTR = LOAD) Writes to the CCACTx_y register are stored in shadow register and transferred to CCACTx_y in the TIMCLK cycle following CTR equals 0 or CTR. Equals LDn. Note this update mechanism is defined for use only in configurations using up/down counting. This mode is not intended for use in down count configurations. 5h = Following a zero event (CTR=0) with repeat count also zero (RC=0). Writes to the CCACTx_y register are stored in shadow register and transferred to CCACTx_y in the TIMCLK cycle following CTR equals 0 and if RC equal 0. 6h = On a TRIG pulse, the value stored in CCACT_xy shadow register is loaded into CCACT_xy register.
25	RESERVED	R/W	0h	Reserved
24-22	CC2SELU	R/W	0h	Selects the source second CCU event. 0h = Selects CCU from CC0. 1h = Selects CCU from CC1. 2h = Selects CCU from CC2. 3h = Selects CCU from CC3. 4h = Selects CCU from CC4. 5h = Selects CCU from CC5.
21	RESERVED	R/W	0h	
20-18	RESERVED	R/W	0h	Reserved

Table 32-122. CCCTL_01_y Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
17	COC	R/W	0h	Capture or Compare. Specifies whether the corresponding CC register is used as a capture register or a compare register (never both). 0h = Compare 1h = Capture
16-15	RESERVED	R/W	0h	
14-12	ZCOND	R/W	0h	Zero Condition. This field specifies the condition that generates a zero pulse. 0h = CCP edges have no effect 1h = Rising edge of CCP or trigger assertion edge 2h = Falling edge of CCP or trigger de-assertion edge 3h = Either edge of CCP or trigger change (assertion/de-assertion edge)
11	RESERVED	R/W	0h	
10-8	LCOND	R/W	0h	Load Condition. Specifies the condition that generates a load pulse. 0h = CCP edges have no effect 1h = Rising edge of CCP or trigger assertion edge 2h = Falling edge of CCP or trigger de-assertion edge 3h = Either edge of CCP or trigger change (assertion/de-assertion edge)
7	RESERVED	R/W	0h	
6-4	ACOND	R/W	0h	Advance Condition. Specifies the condition that generates an advance pulse. 0h = Each TIMCLK 1h = Rising edge of CCP or trigger assertion edge 2h = Falling edge of CCP or trigger de-assertion edge 3h = Either edge of CCP or trigger change (assertion/de-assertion edge) 5h = CCP High or Trigger assertion (level)
3	RESERVED	R/W	0h	
2-0	CCOND	R/W	0h	Capture Condition. Specifies the condition that generates a capture pulse. 0h = None (never captures) 1h = Rising edge of CCP or trigger assertion edge 2h = Falling edge of CCP or trigger de-assertion edge 3h = Either edge of CCP or trigger change (assertion/de-assertion edge)

52 OCTL_01_y Register (Offset = 1850h + formula) [Reset = 00000000h]

OCTL_01_y is shown in Figure 32-76 and described in Table 32-72.

Return to the Summary Table.

The OCTL_01 register controls the CCP output of the Capture-Compare slice of the counter. This includes the ability to select the source of what is driven out along with initial condition values and final inversion options.

Offset = 1850h + (y * 4h); where y = 0h to 1h

Figure 32-76. OCTL_01_y Register

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

Table 32-124. OCTL_01_y Register Field Descriptions

Bit	Field	Type	Reset	Description
31-6	RESERVED	R/W	0h	
5	CCPIV	R/W	0h	CCP Initial Value This bit specifies the logical value put on the signal generator state while the counter is disabled (CTRCTL.EN == 0). 0h = Low 1h = High
4	CCPOINV	R/W	0h	CCP Output Invert The output as selected by CCPO is conditionally inverted. 0h = No inversion 1h = Invert
3-0	CCPO	R/W	0h	CCP Output Source 0h = Signal generator value (for example, PWM, triggered PWM) 1h = Load event 2h = CCU event or CCD event 4h = Zero event 5h = Capture event 6h = Fault condition 8h = Mirror CCP of first capture and compare register to other capture compare blocks 9h = Mirror CCP of second capture and compare register in other capture compare blocks Ch = Signal generator output after deadband insertion Dh = Counter direction

53 CCACT_01_y Register (Offset = 1870h + formula) [Reset = 00000000h]

CCACT_01_y is shown in Figure 32-77 and described in Table 32-73.

Return to the Summary Table.

The CCACT_01 register controls the actions of the signal generator of the capture-compare slice based on the events created in the counter block, the capture and compare block and debug events.

Offset = 1870h + (y * 4h); where y = 0h to 1h

Figure 32-77. CCACT_01_y Register

31	30	29	28	27	26	25	24
RESERVED		SWFRCACT		RESERVED			RESERVED
R/W-0h		R/W-0h		R/W-0h			R/W-0h
23	22	21	20	19	18	17	16
RESERVED		RESERVED					CC2UACT
R/W-0h		R/W-0h					R/W-0h
15	14	13	12	11	10	9	8
CC2UACT	RESERVED	CC2DACT		RESERVED	CUACT		RESERVED
R/W-0h	R/W-0h	R/W-0h		R/W-0h	R/W-0h		R/W-0h
7	6	5	4	3	2	1	0
CDACT		RESERVED	LACT		RESERVED	ZACT	
R/W-0h		R/W-0h	R/W-0h		R/W-0h	R/W-0h	

Table 32-126. CCACT_01_y Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	RESERVED	R/W	0h	Reserved
29-28	SWFRCACT	R/W	0h	CCP Output Action on Software Force Output This field describes the resulting action of software force. This action has a shadow register, which will be updated under specific condition. So that this register cannot take into effect immediately. 0h = This event is disabled and a lower priority event is selected if asserting. The CCP output value is unaffected by the event. 1h = CCP output value is set high 2h = CCP output value is set low
27-25	RESERVED	R/W	0h	Reserved
24-22	RESERVED	R/W	0h	Reserved
21-17	RESERVED	R/W	0h	
16-15	CC2UACT	R/W	0h	CCP Output Action on CC2U event. 0h = This event is disabled and a lower priority event is selected if asserting. The CCP output value is unaffected by the event. 1h = CCP output value is set high 2h = CCP output value is set low 3h = CCP output value is toggled
14	RESERVED	R/W	0h	
13-12	CC2DACT	R/W	0h	CCP Output Action on CC2D event. 0h = This event is disabled and a lower priority event is selected if asserting. The CCP output value is unaffected by the event. 1h = CCP output value is set high 2h = CCP output value is set low 3h = CCP output value is toggled
11	RESERVED	R/W	0h	

Table 32-126. CCACT_01_y Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
10-9	CUACT	R/W	0h	CCP Output Action on Compare (Up) This field describes the resulting action of the signal generator upon detecting a compare event while counting up. 0h = This event is disabled and a lower priority event is selected if asserting. The CCP output value is unaffected by the event. 1h = CCP output value is set high 2h = CCP output value is set low 3h = CCP output value is toggled
8	RESERVED	R/W	0h	
7-6	CDACT	R/W	0h	CCP Output Action on Compare (Down) This field describes the resulting action of the signal generator upon detecting a compare event while counting down. 0h = This event is disabled and a lower priority event is selected if asserting. The CCP output value is unaffected by the event. 1h = CCP output value is set high 2h = CCP output value is set low 3h = CCP output value is toggled
5	RESERVED	R/W	0h	
4-3	LACT	R/W	0h	CCP Output Action on Load Specifies what changes occur to CCP output as the result of a load event. 0h = This event is disabled and a lower priority event is selected if asserting. The CCP output value is unaffected by the event. 1h = CCP output value is set high 2h = CCP output value is set low 3h = CCP output value is toggled
2	RESERVED	R/W	0h	
1-0	ZACT	R/W	0h	CCP Output Action on Zero Specifies what changes occur to CCP output as the result of a zero event. 0h = This event is disabled and a lower priority event is selected if asserting. The CCP output value is unaffected by the event. 1h = CCP output value is set high 2h = CCP output value is set low 3h = CCP output value is toggled

54 IFCTL_01_y Register (Offset = 1880h + formula) [Reset = 00000000h]

IFCTL_01_y is shown in Figure 32-78 and described in Table 32-74.

Return to the Summary Table.

The IFCTL_01 register controls the input selection and inversion for the associated Capture-Compare slice.

Offset = 1880h + (y * 4h); where y = 0h to 1h

Figure 32-78. IFCTL_01_y Register

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

Table 32-128. IFCTL_01_y Register Field Descriptions

Bit	Field	Type	Reset	Description
31-13	RESERVED	R/W	0h	
12	FE	R/W	0h	Filter Enable This bit controls whether the input is filtered by the input filter or bypasses to the edge detect. 0h = Bypass. 1h = Filtered.
11	CPV	R/W	0h	Consecutive Period/Voting Select This bit controls whether the input filter uses a stricter consecutive period count or majority voting. 0h = Consecutive Periods The input must be at a specific logic level for the period defined by FP before it is passed to the filter output. 1h = Voting The filter ignores one clock of opposite logic over the filter period. I.e. Over FP samples of the input, up to 1 sample may be of an opposite logic value (glitch) without affecting the output.
10	RESERVED	R/W	0h	
9-8	FP	R/W	0h	Filter Period. This field specifies the sample period for the input filter. I.e. The input is sampled for FP timer clocks during filtering. 0h = The division factor is 3 1h = The division factor is 5 2h = The division factor is 8

ADVANCE INFORMATION

Table 32-128. IFCTL_01_y Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
7	INV	R/W	0h	Input Inversion This bit controls whether the selected input is inverted. 0h = Noninverted 1h = Inverted
6-4	RESERVED	R/W	0h	
3-0	ISEL	R/W	0h	Input Select (CCP0) This field selects the input source to the filter input. 4h-7h = Reserved 0h = CCP of the corresponding capture compare unit 1h = Input pair CCPX of the capture compare unit. For CCP0 input pair is CCP1 and for CCP1 input pair is CCP0. 2h = CCP0 of the counter 3h = Trigger 4h = XOR of CCP inputs as input source (Used in Hall input mode). 5h = subscriber 0 event as input source. 6h = subscriber 1 event as input source. 7h = Comparator 0 output. 8h = Comparator 1 output. 9h = Comparator 2 output.

55 TSEL Register (Offset = 18B0h) [Reset = 00000000h]

TSEL is shown in Figure 32-79 and described in Table 32-75.

Return to the Summary Table.

The TSEL register controls the input trigger enable and selection of the trigger source. Trigger sources are generated by other SoC elements through their respective publisher ports (subscribed in by the timer's subscriber port).

Figure 32-79. TSEL Register

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

Table 32-130. TSEL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-10	RESERVED	R/W	0h	
9	TE	R/W	0h	Trigger Enable. This selects whether a trigger is enabled or not for this counter 0x0 = Triggers are not used 0x1 = Triggers are used as selected by the ETSEL field 0h = Triggers are not used. 1h = Triggers are used as selected by the IE, ITSEL and ETSEL fields.
8-5	RESERVED	R/W	0h	
4-0	ETSEL	R/W	0h	External Trigger Select. This selects which System Event is used if the input filter selects trigger. Triggers 0-15 are used to connect triggers generated by other timer modules. Refer to the SoC datasheet for details related to timer trigger sources. Triggers 16 and 17 are connected to event manager subscriber ports. Event lines 18-31 are reserved for future use. 0h = TRIGx = External trigger input from TIM x. 1h = TRIGx = External trigger input from TIM x. 2h = TRIGx = External trigger input from TIM x. 3h = TRIGx = External trigger input from TIM x. 4h = TRIGx = External trigger input from TIM x. 5h = TRIGx = External trigger input from TIM x. 6h = TRIGx = External trigger input from TIM x. 7h = TRIGx = External trigger input from TIM x. 8h = TRIGx = External trigger input from TIM x. 9h = TRIGx = External trigger input from TIM x. Ah = TRIGx = External trigger input from TIM x. Bh = TRIGx = External trigger input from TIM x. Ch = TRIGx = External trigger input from TIM x. Dh = TRIGx = External trigger input from TIM x. Eh = TRIGx = External trigger input from TIM x. Fh = TRIGx = External trigger input from TIM x. 10h = TRIG_SUBx = External trigger input from subscriber port x. 11h = TRIG_SUBx = External trigger input from subscriber port x.

32.2.3 TIMG12_REGS Registers

Table 32-76 lists the memory-mapped registers for the TIMG12_REGS registers. All register offset addresses not listed in Table 32-76 should be considered as reserved locations and the register contents should not be modified.

Table 32-131. TIMG12_REGS Registers

Offset	Acronym	Register Name	Section
4h	CCP0	CCP0 PinCM	Go
8h	CCP1	CCP0 PinCM	Go
24h	CCP2_CMPL	CCP2_CMPL PinCM	Go
28h	CCP3_CMPL	CCP3_CMPL PinCM	Go
204h	CCP0	CCP0 FUPDATE	Go
208h	CCP1	CCP1 FUPDATE	Go
224h	CCP2_CMPL	CCP2_CMPL FUPDATE	Go
228h	CCP3_CMPL	CCP3_CMPL FUPDATE	Go
400h	FSUB_0	Subsciber Port 0	Go
404h	FSUB_1	Subscriber Port 1	Go
444h	FPUB_0	Publisher Port 0	Go
448h	FPUB_1	Publisher Port 1	Go
480h	CPU_CONNECT_0	CPU connect 0 configuration byte	Go
800h	PWREN	Power enable	Go
804h	RSTCTL	Reset Control	Go
814h	STAT	Status Register	Go
1000h	CLKDIV	Clock Divider	Go
1008h	CLKSEL	Clock Select for Ultra Low Power peripherals	Go
1018h	PDBGCTL	Peripheral Debug Control	Go
1020h	IIDX	Interrupt index	Go
1028h	IMASK	Interrupt mask	Go
1030h	RIS	Raw interrupt status	Go
1038h	MIS	Masked interrupt status	Go
1040h	ISET	Interrupt set	Go
1048h	ICLR	Interrupt clear	Go
1050h	IIDX	Interrupt index	Go
1058h	IMASK	Interrupt mask	Go
1060h	RIS	Raw interrupt status	Go
1068h	MIS	Masked interrupt status	Go
1070h	ISET	Interrupt set	Go
1078h	ICLR	Interrupt clear	Go
1080h	IIDX	Interrupt index	Go
1088h	IMASK	Interrupt mask	Go
1090h	RIS	Raw interrupt status	Go
1098h	MIS	Masked interrupt status	Go
10A0h	ISET	Interrupt set	Go
10A8h	ICLR	Interrupt clear	Go
10E0h	EVT_MODE	Event Mode	Go
10FCh	DESC	Module Description	Go
1100h	CCPD	CCP Direction	Go

Table 32-131. TIMG12_REGS Registers (continued)

Offset	Acronym	Register Name	Section
1104h	ODIS	Output Disable	Go
1108h	CCLKCTL	Counter Clock Control Register	Go
110Ch	CPS	Clock Prescale Register	Go
1110h	CPSV	Clock prescale count status register	Go
1114h	CTTRIGCTL	Timer Cross Trigger Control Register	Go
111Ch	CTTRIG	Timer Cross Trigger Register	Go
1800h	CTR	Counter Register	Go
1804h	CTRCTL	Counter Control Register	Go
1808h	LOAD	Load Register	Go
1810h + formula	CC_01_y	Capture or Compare Register 0/1	Go
1830h + formula	CCCTL_01_y	Capture or Compare Control Registers 0/1	Go
1850h + formula	OCTL_01_y	CCP Output Control Registers 0/1	Go
1870h + formula	CCACT_01_y	Capture or Compare Action Registers 0/1	Go
1880h + formula	IFCTL_01_y	Input Filter Control Register 0/1	Go
18B0h	TSEL	Trigger Select Register	Go

Complex bit access types are encoded to fit into small table cells. Table 32-77 shows the codes that are used for access types in this section.

Table 32-132. TIMG12_REGS Access Type Codes

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

ADVANCE INFORMATION

1 CCP0 Register (Offset = 4h) [Reset = 00000000h]

CCP0 is shown in Figure 32-80 and described in Table 32-78.

Return to the Summary Table.

CCP0 PinCM register in full write region

Figure 32-80. CCP0 Register

31	30	29	28	27	26	25	24
RESERVED	GFLT	SLEW	WCOMP	WUEN	INV	HIGHZ1	HIGHZ0
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
RESERVED	DRV			HYSTEN	INENA	PIPU	PIPD
R/W-0h	R/W-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
GSTATE		RESERVED					
R/W-0h		R/W-0h					
7	6	5	4	3	2	1	0
PSTATE		RESERVED					
R/W-0h		R/W-0h					

Table 32-134. CCP0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	
30	GFLT	R/W	0h	Glitch Filter Enable 0h = No internal glitch filter 1h = Use internal glitch filter
29	SLEW	R/W	0h	Reserved Slew Rate Control 0h = No Slew Rate Control 1h = Use Slew Rate Control
28	WCOMP	R/W	0h	Wake up compare value 0h = Match '0' will wake 1h = Match '1' will wake
27	WUEN	R/W	0h	Wake up enable 0h = Wake up not enabled 1h = Wake up enabled
26	INV	R/W	0h	Invert digital input/output relative to peripheral/GPIO 0h = Input and output are non-inverted 1h = Input and output are inverted
25	HIGHZ1	R/W	0h	High-Z instead of high output 0h = Pin can be driven high 1h = Pin is tri-stated instead of driven high
24	HIGHZ0	R/W	0h	High-Z instead of low output 0h = Pin can be driven low 1h = Pin is tri-stated instead of driven low
23	RESERVED	R/W	0h	
22-20	DRV	R/W	0h	Drive strength options 0h = Lowest drive strength 1h = Drive strength 2/8 2h = Drive strength 3/8 3h = Drive strength 4/8 4h = Drive strength 5/8 5h = Drive strength 6/8 6h = Drive strength 7/8 7h = Highest drive strength

Table 32-134. CCP0 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
19	HYSTEN	R/W	0h	Hysteresis enable 0h = No hysteresis 1h = Hysteresis on
18	INENA	R/W	0h	Input enable 0h = Inputs 0 to connected core 1h = Inputs IO pad value to connected core
17	PIPU	R/W	0h	Pull up enable 0h = No pull up 1h = Pull up
16	PIPD	R/W	0h	Pull down enable 0h = No pull down 1h = Pull down
15-14	GSTATE	R/W	0h	GPIO Channel State 0h = G-Channel is in Unassigned State 1h = G-Channel is in Handover State 2h = G-Channle is in Connected State and not Locked (That is F field is allowed to change without going back through Unassigned state) 3h = G-Channel is in Connected State and Locked (That is F field is not allowed to change to a different non-Zero value until both G and P channels go to Unassigned)
13-8	RESERVED	R/W	0h	
7-6	PSTATE	R/W	0h	Peripheral-Analog Channel State 0h = P-Channel is in Unassigned State 1h = P-Channel is in Handover State 2h = P-Channle is in Connected State and not Locked (That is F field is allowed to change without going back through Unassigned state) 3h = P-Channel is in Connected State and Locked (That is F field is not allowed to change to a different non-Zero value until both G and P channels go to Unassigned)
5-0	RESERVED	R/W	0h	

2 CCP1 Register (Offset = 8h) [Reset = 00000000h]

CCP1 is shown in Figure 32-81 and described in Table 32-79.

Return to the Summary Table.

CCP0 PinCM register in full write region

Figure 32-81. CCP1 Register

31	30	29	28	27	26	25	24
RESERVED	GFLT	SLEW	WCOMP	WUEN	INV	HIGHZ1	HIGHZ0
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
RESERVED	DRV			HYSTEN	INENA	PIPU	PIPD
R/W-0h	R/W-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
GSTATE		RESERVED					
R/W-0h		R/W-0h					
7	6	5	4	3	2	1	0
PSTATE		RESERVED					
R/W-0h		R/W-0h					

Table 32-136. CCP1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	
30	GFLT	R/W	0h	Glitch Filter Enable 0h = No internal glitch filter 1h = Use internal glitch filter
29	SLEW	R/W	0h	Reserved Slew Rate Control 0h = No Slew Rate Control 1h = Use Slew Rate Control
28	WCOMP	R/W	0h	Wake up compare value 0h = Match '0' will wake 1h = Match '1' will wake
27	WUEN	R/W	0h	Wake up enable 0h = Wake up not enabled 1h = Wake up enabled
26	INV	R/W	0h	Invert digital input/output relative to peripheral/GPIO 0h = Input and output are non-inverted 1h = Input and output are inverted
25	HIGHZ1	R/W	0h	High-Z instead of high output 0h = Pin can be driven high 1h = Pin is tri-stated instead of driven high
24	HIGHZ0	R/W	0h	High-Z instead of low output 0h = Pin can be driven low 1h = Pin is tri-stated instead of driven low
23	RESERVED	R/W	0h	
22-20	DRV	R/W	0h	Drive strength options 0h = Lowest drive strength 1h = Drive strength 2/8 2h = Drive strength 3/8 3h = Drive strength 4/8 4h = Drive strength 5/8 5h = Drive strength 6/8 6h = Drive strength 7/8 7h = Highest drive strength

Table 32-136. CCP1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
19	HYSTEN	R/W	0h	Hysteresis enable 0h = No hysteresis 1h = Hysteresis on
18	INENA	R/W	0h	Input enable 0h = Inputs 0 to connected core 1h = Inputs IO pad value to connected core
17	PIPU	R/W	0h	Pull up enable 0h = No pull up 1h = Pull up
16	PIPD	R/W	0h	Pull down enable 0h = No pull down 1h = Pull down
15-14	GSTATE	R/W	0h	GPIO Channel State 0h = G-Channel is in Unassigned State 1h = G-Channel is in Handover State 2h = G-Channle is in Connected State and not Locked (That is F field is allowed to change without going back through Unassigned state) 3h = G-Channel is in Connected State and Locked (That is F field is not allowed to change to a different non-Zero value until both G and P channels go to Unassigned)
13-8	RESERVED	R/W	0h	
7-6	PSTATE	R/W	0h	Peripheral-Analog Channel State 0h = P-Channel is in Unassigned State 1h = P-Channel is in Handover State 2h = P-Channle is in Connected State and not Locked (That is F field is allowed to change without going back through Unassigned state) 3h = P-Channel is in Connected State and Locked (That is F field is not allowed to change to a different non-Zero value until both G and P channels go to Unassigned)
5-0	RESERVED	R/W	0h	

3 CCP2_CMPL Register (Offset = 24h) [Reset = 00000000h]

CCP2_CMPL is shown in Figure 32-82 and described in Table 32-80.

Return to the Summary Table.

CCP2_CMPL PinCM register in full write region

Figure 32-82. CCP2_CMPL Register

31	30	29	28	27	26	25	24
RESERVED	GFLT	SLEW	WCOMP	WUEN	INV	HIGHZ1	HIGHZ0
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
RESERVED	DRV			HYSTEN	INENA	PIPU	PIPD
R/W-0h	R/W-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
GSTATE		RESERVED					
R/W-0h		R/W-0h					
7	6	5	4	3	2	1	0
PSTATE		RESERVED					
R/W-0h		R/W-0h					

Table 32-138. CCP2_CMPL Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	
30	GFLT	R/W	0h	Glitch Filter Enable 0h = No internal glitch filter 1h = Use internal glitch filter
29	SLEW	R/W	0h	Reserved Slew Rate Control 0h = No Slew Rate Control 1h = Use Slew Rate Control
28	WCOMP	R/W	0h	Wake up compare value 0h = Match '0' will wake 1h = Match '1' will wake
27	WUEN	R/W	0h	Wake up enable 0h = Wake up not enabled 1h = Wake up enabled
26	INV	R/W	0h	Invert digital input/output relative to peripheral/GPIO 0h = Input and output are non-inverted 1h = Input and output are inverted
25	HIGHZ1	R/W	0h	High-Z instead of high output 0h = Pin can be driven high 1h = Pin is tri-stated instead of driven high
24	HIGHZ0	R/W	0h	High-Z instead of low output 0h = Pin can be driven low 1h = Pin is tri-stated instead of driven low
23	RESERVED	R/W	0h	
22-20	DRV	R/W	0h	Drive strength options 0h = Lowest drive strength 1h = Drive strength 2/8 2h = Drive strength 3/8 3h = Drive strength 4/8 4h = Drive strength 5/8 5h = Drive strength 6/8 6h = Drive strength 7/8 7h = Highest drive strength

Table 32-138. CCP2_CMPL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
19	HYSTEN	R/W	0h	Hysteresis enable 0h = No hysteresis 1h = Hysteresis on
18	INENA	R/W	0h	Input enable 0h = Inputs 0 to connected core 1h = Inputs IO pad value to connected core
17	PIPU	R/W	0h	Pull up enable 0h = No pull up 1h = Pull up
16	PIPD	R/W	0h	Pull down enable 0h = No pull down 1h = Pull down
15-14	GSTATE	R/W	0h	GPIO Channel State 0h = G-Channel is in Unassigned State 1h = G-Channel is in Handover State 2h = G-Channle is in Connected State and not Locked (That is F field is allowed to change without going back through Unassigned state) 3h = G-Channel is in Connected State and Locked (That is F field is not allowed to change to a different non-Zero value until both G and P channels go to Unassigned)
13-8	RESERVED	R/W	0h	
7-6	PSTATE	R/W	0h	Peripheral-Analog Channel State 0h = P-Channel is in Unassigned State 1h = P-Channel is in Handover State 2h = P-Channle is in Connected State and not Locked (That is F field is allowed to change without going back through Unassigned state) 3h = P-Channel is in Connected State and Locked (That is F field is not allowed to change to a different non-Zero value until both G and P channels go to Unassigned)
5-0	RESERVED	R/W	0h	

4 CCP3_CMPL Register (Offset = 28h) [Reset = 00000000h]

CCP3_CMPL is shown in Figure 32-83 and described in Table 32-81.

Return to the Summary Table.

CCP3_CMPL PinCM register in full write region

Figure 32-83. CCP3_CMPL Register

31	30	29	28	27	26	25	24
RESERVED	GFLT	SLEW	WCOMP	WUEN	INV	HIGHZ1	HIGHZ0
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
RESERVED	DRV			HYSTEN	INENA	PIPU	PIPD
R/W-0h	R/W-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
GSTATE		RESERVED					
R/W-0h		R/W-0h					
7	6	5	4	3	2	1	0
PSTATE		RESERVED					
R/W-0h		R/W-0h					

Table 32-140. CCP3_CMPL Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	
30	GFLT	R/W	0h	Glitch Filter Enable 0h = No internal glitch filter 1h = Use internal glitch filter
29	SLEW	R/W	0h	Reserved Slew Rate Control 0h = No Slew Rate Control 1h = Use Slew Rate Control
28	WCOMP	R/W	0h	Wake up compare value 0h = Match '0' will wake 1h = Match '1' will wake
27	WUEN	R/W	0h	Wake up enable 0h = Wake up not enabled 1h = Wake up enabled
26	INV	R/W	0h	Invert digital input/output relative to peripheral/GPIO 0h = Input and output are non-inverted 1h = Input and output are inverted
25	HIGHZ1	R/W	0h	High-Z instead of high output 0h = Pin can be driven high 1h = Pin is tri-stated instead of driven high
24	HIGHZ0	R/W	0h	High-Z instead of low output 0h = Pin can be driven low 1h = Pin is tri-stated instead of driven low
23	RESERVED	R/W	0h	
22-20	DRV	R/W	0h	Drive strength options 0h = Lowest drive strength 1h = Drive strength 2/8 2h = Drive strength 3/8 3h = Drive strength 4/8 4h = Drive strength 5/8 5h = Drive strength 6/8 6h = Drive strength 7/8 7h = Highest drive strength

Table 32-140. CCP3_CMPL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
19	HYSTEN	R/W	0h	Hysteresis enable 0h = No hysteresis 1h = Hysteresis on
18	INENA	R/W	0h	Input enable 0h = Inputs 0 to connected core 1h = Inputs IO pad value to connected core
17	PIPU	R/W	0h	Pull up enable 0h = No pull up 1h = Pull up
16	PIPD	R/W	0h	Pull down enable 0h = No pull down 1h = Pull down
15-14	GSTATE	R/W	0h	GPIO Channel State 0h = G-Channel is in Unassigned State 1h = G-Channel is in Handover State 2h = G-Channle is in Connected State and not Locked (That is F field is allowed to change without going back through Unassigned state) 3h = G-Channel is in Connected State and Locked (That is F field is not allowed to change to a different non-Zero value until both G and P channels go to Unassigned)
13-8	RESERVED	R/W	0h	
7-6	PSTATE	R/W	0h	Peripheral-Analog Channel State 0h = P-Channel is in Unassigned State 1h = P-Channel is in Handover State 2h = P-Channle is in Connected State and not Locked (That is F field is allowed to change without going back through Unassigned state) 3h = P-Channel is in Connected State and Locked (That is F field is not allowed to change to a different non-Zero value until both G and P channels go to Unassigned)
5-0	RESERVED	R/W	0h	

5 CCP0 Register (Offset = 204h) [Reset = 00000000h]

CCP0 is shown in Figure 32-84 and described in Table 32-82.

Return to the Summary Table.

FUPDATE version of CCP0

Figure 32-84. CCP0 Register

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

Table 32-142. CCP0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	RESERVED	W	0h	
27-2	IOADDR	W	0h	IO Address This is the address that corresponds to the SOC address[27:2] of the module IP instance specific IO signal in the "Full Write" subregion of the pinmux subregion.
1	LOCK	W	0h	Sets lock bit 0h = Writing this value has no effect 1h = Set channel lock bit
0	GSEL	W	0h	GPIO channel Select 0: Select the P-Channel for the F update 1: Select the G-Channel for the F update 0h = Select the P-Channel for the F update 1h = Select the G-Channel for the F update

6 CCP1 Register (Offset = 208h) [Reset = 00000000h]

CCP1 is shown in Figure 32-85 and described in Table 32-83.

Return to the Summary Table.

FUPDATE version of CCP1

Figure 32-85. CCP1 Register

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

Table 32-144. CCP1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	RESERVED	W	0h	
27-2	IOADDR	W	0h	IO Address This is the address that corresponds to the SOC address[27:2] of the module IP instance specific IO signal in the "Full Write" subregion of the pinmux subregion.
1	LOCK	W	0h	Sets lock bit 0h = Writing this value has no effect 1h = Set channel lock bit
0	GSEL	W	0h	GPIO channel Select 0: Select the P-Channel for the F update 1: Select the G-Channel for the F update 0h = Select the P-Channel for the F update 1h = Select the G-Channel for the F update

7 CCP2_CMPL Register (Offset = 224h) [Reset = 00000000h]

CCP2_CMPL is shown in Figure 32-86 and described in Table 32-84.

Return to the Summary Table.

FUPDATE version of CCP2_CMPL

Figure 32-86. CCP2_CMPL Register

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

Table 32-146. CCP2_CMPL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	RESERVED	W	0h	
27-2	IOADDR	W	0h	IO Address This is the address that corresponds to the SOC address[27:2] of the module IP instance specific IO signal in the "Full Write" subregion of the pinmux subregion.
1	LOCK	W	0h	Sets lock bit 0h = Writing this value has no effect 1h = Set channel lock bit
0	GSEL	W	0h	GPIO channel Select 0: Select the P-Channel for the F update 1: Select the G-Channel for the F update 0h = Select the P-Channel for the F update 1h = Select the G-Channel for the F update

8 CCP3_CMPL Register (Offset = 228h) [Reset = 00000000h]

CCP3_CMPL is shown in Figure 32-87 and described in Table 32-85.

Return to the Summary Table.

FUPDATE version of CCP3_CMPL

Figure 32-87. CCP3_CMPL Register

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

Table 32-148. CCP3_CMPL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	RESERVED	W	0h	
27-2	IOADDR	W	0h	IO Address This is the address that corresponds to the SOC address[27:2] of the module IP instance specific IO signal in the "Full Write" subregion of the pinmux subregion.
1	LOCK	W	0h	Sets lock bit 0h = Writing this value has no effect 1h = Set channel lock bit
0	GSEL	W	0h	GPIO channel Select 0: Select the P-Channel for the F update 1: Select the G-Channel for the F update 0h = Select the P-Channel for the F update 1h = Select the G-Channel for the F update

9 FSUB_0 Register (Offset = 400h) [Reset = 00000000h]

FSUB_0 is shown in Figure 32-88 and described in Table 32-86.

Return to the Summary Table.

Subscriber port

Figure 32-88. FSUB_0 Register

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

Table 32-150. FSUB_0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R/W	0h	
3-0	CHANID	R/W	0h	0 = disconnected. 1-15 = connected to channelID = CHANID. 0h = A value of 0 specifies that the event is not connected Fh = Consult your device datasheet as the actual allowed maximum may be less than 15.

10 FSUB_1 Register (Offset = 404h) [Reset = 00000000h]

FSUB_1 is shown in Figure 32-89 and described in Table 32-87.

Return to the Summary Table.

Subscriber port

Figure 32-89. FSUB_1 Register

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

Table 32-152. FSUB_1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R/W	0h	
3-0	CHANID	R/W	0h	0 = disconnected. 1-15 = connected to channelID = CHANID. 0h = A value of 0 specifies that the event is not connected Fh = Consult your device datasheet as the actual allowed maximum may be less than 15.

11 FPUB_0 Register (Offset = 444h) [Reset = 00000000h]

FPUB_0 is shown in Figure 32-90 and described in Table 32-88.

Return to the Summary Table.

Publisher port

Figure 32-90. FPUB_0 Register

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

Table 32-154. FPUB_0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R/W	0h	
3-0	CHANID	R/W	0h	0 = disconnected. 1-15 = connected to channelID = CHANID. 0h = A value of 0 specifies that the event is not connected Fh = Consult your device datasheet as the actual allowed maximum may be less than 15.

12 FPUB_1 Register (Offset = 448h) [Reset = 00000000h]

FPUB_1 is shown in Figure 32-91 and described in Table 32-89.

Return to the Summary Table.

Publisher port

Figure 32-91. FPUB_1 Register

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

Table 32-156. FPUB_1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R/W	0h	
3-0	CHANID	R/W	0h	0 = disconnected. 1-15 = connected to channelID = CHANID. 0h = A value of 0 specifies that the event is not connected Fh = Consult your device datasheet as the actual allowed maximum may be less than 15.

13 CPU_CONNECT_0 Register (Offset = 480h) [Reset = 00000000h]

CPU_CONNECT_0 is shown in Figure 32-92 and described in Table 32-90.

Return to the Summary Table.

Connect peripheral interrupts / publisher port (FPUB_1) to application processor

Figure 32-92. CPU_CONNECT_0 Register

31	30	29	28	27	26	25	24
RESERVED							
R/W-0h							
23	22	21	20	19	18	17	16
RESERVED							
R/W-0h							
15	14	13	12	11	10	9	8
RESERVED							
R/W-0h							
7	6	5	4	3	2	1	0
RESERVED						CPUSS0_CONN	RESERVED
R/W-0h						R/W-0h	R/W-0h

Table 32-158. CPU_CONNECT_0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R/W	0h	
1	CPUSS0_CONN	R/W	0h	CPUSS0 connect bit. 0h = The CPU is not connected. 1h = The CPU is connected.
0	RESERVED	R/W	0h	

ADVANCE INFORMATION

14 PWREN Register (Offset = 800h) [Reset = 00000000h]

PWREN is shown in Figure 32-93 and described in Table 32-91.

Return to the Summary Table.

Register to control the power state

Figure 32-93. PWREN Register

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

Table 32-160. PWREN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	KEY	W	0h	KEY to allow Power State Change 26h = KEY to allow write access to this register
23-1	RESERVED	R/W	0h	
0	ENABLE	R/WK	0h	Enable the power [EXT_GPRCM.GPRCM.PWREN.KEY] must be set to 26h to write to this bit. 0h = Disable Power 1h = Enable Power

ADVANCE INFORMATION

15 RSTCTL Register (Offset = 804h) [Reset = 00000000h]

RSTCTL is shown in Figure 32-94 and described in Table 32-92.

Return to the Summary Table.

Register to control reset assertion and de-assertion

Figure 32-94. RSTCTL Register

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

Table 32-162. RSTCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	KEY	W	0h	Unlock key B1h = KEY to allow write access to this register
23-2	RESERVED	W	0h	
1	RESETSTKYCLR	WK	0h	Clear the RESETSTKY bit in the STAT register [EXT_GPRCM.GPRCM.RSTCTL.KEY] must be set to B1h to write to this bit. 0h = Writing 0 has no effect 1h = Clear reset sticky bit
0	RESETASSERT	WK	0h	Assert reset to the peripheral [EXT_GPRCM.GPRCM.RSTCTL.KEY] must be set to B1h to write to this bit. 0h = Writing 0 has no effect 1h = Assert reset

ADVANCE INFORMATION

16 STAT Register (Offset = 814h) [Reset = 00000000h]

STAT is shown in Figure 32-95 and described in Table 32-93.

Return to the Summary Table.

peripheral enable and reset status

Figure 32-95. STAT Register

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

Table 32-164. STAT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-17	RESERVED	R	0h	
16	RESETSTKY	R	0h	This bit indicates, if the peripheral was reset, since this bit was cleared by RESETSTKYCLR in the RSTCTL register 0h = The peripheral has not been reset since this bit was last cleared by RESETSTKYCLR in the RSTCTL register 1h = The peripheral was reset since the last bit clear
15-0	RESERVED	R	0h	

17 CLKDIV Register (Offset = 1000h) [Reset = 00000000h]

CLKDIV is shown in Figure 32-96 and described in Table 32-94.

Return to the Summary Table.

This register is used to specify module-specific divide ratio of the functional clock

Figure 32-96. CLKDIV Register

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

Table 32-166. CLKDIV Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	RESERVED	R/W	0h	
2-0	RATIO	R/W	0h	Selects divide ratio of module clock 0h = Do not divide clock source 1h = Divide clock source by 2 2h = Divide clock source by 3 3h = Divide clock source by 4 4h = Divide clock source by 5 5h = Divide clock source by 6 6h = Divide clock source by 7 7h = Divide clock source by 8

18 CLKSEL Register (Offset = 1008h) [Reset = 00000000h]

CLKSEL is shown in Figure 32-97 and described in Table 32-95.

Return to the Summary Table.

Clock Source Select Register

Figure 32-97. CLKSEL Register

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

Table 32-168. CLKSEL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-5	RESERVED	R/W	0h	
4	RESERVED	R/W	0h	Reserved
3	BUSCLK_SEL	R/W	0h	Selects BUSCLK as clock source if enabled 0h = Does not select this clock as a source 1h = Select this clock as a source
2	MFCLK_SEL	R/W	0h	Selects MFCLK as clock source if enabled 0h = Does not select this clock as a source 1h = Select this clock as a source
1	LFCLK_SEL	R/W	0h	Selects LFCLK as clock source if enabled 0h = Does not select this clock as a source 1h = Select this clock as a source
0	RESERVED	R/W	0h	

19 PDBGCTL Register (Offset = 1018h) [Reset = 00000000h]

PDBGCTL is shown in Figure 32-98 and described in Table 32-96.

Return to the Summary Table.

This register can be used by the software developer to control the behavior of the peripheral relative to the 'Core Halted' input

Figure 32-98. PDBGCTL Register

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

Table 32-170. PDBGCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R/W	0h	
1	SOFT	R/W	1h	Soft halt boundary control. This function is only available, if [FREE] is set to 'STOP' 0h = The peripheral will halt immediately, even if the resultant state will result in corruption if the system is restarted 1h = The peripheral blocks the debug freeze until it has reached a boundary where it can resume without corruption
0	FREE	R/W	1h	Free run control 0h = The peripheral freezes functionality while the Core Halted input is asserted and resumes when it is deasserted. 1h = The peripheral ignores the state of the Core Halted input

20 IIDX Register (Offset = 1020h) [Reset = 00000000h]

IIDX is shown in Figure 32-99 and described in Table 32-97.

Return to the Summary Table.

This register provides the highest priority enabled interrupt index. Value 0x00 means no event pending. Interrupt 1 is the highest priority, IIDX next highest, 4, 8, ... IIDX^31 is the least priority. That is, the least bit position that is set to 1 denotes the highest priority pending interrupt. The priority order is fixed. However, users can implement their own prioritization schemes using other registers that expose the full set of interrupts that have occurred. On each read, only one interrupt is indicated. On a read, the current interrupt (highest priority) is automatically cleared by the hardware and the corresponding interrupt flag in [RIS] and [MIS] are cleared as well. After a read from the CPU (not from the debug interface), the register is updated with the next highest priority interrupt, if none are pending, then it should display 0x0.

Figure 32-99. IIDX Register

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

Table 32-172. IIDX Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	
7-0	STAT	R	0h	Interrupt index status 00h = No interrupt pending 01h = Interrupt Source: Zero event (Z) 02h = nterrupt Source: Load event (L) 05h = Interrupt Source: Capture or compare down event (CCD0) 06h = Interrupt Source: Capture or compare down event (CCD1) 07h = Interrupt Source: Capture or compare down event (CCD2) 08h = Interrupt Source: Capture or compare down event (CCD3) 09h = Interrupt Source: Capture or compare up event (CCU0) 0Ah = Interrupt Source: Capture or compare up event (CCU1) 0Bh = Interrupt Source: Capture or compare up event (CCU2) 0Ch = Interrupt Source: Capture or compare up event (CCU3) 0Dh = Interrupt Source: Compare down event (CCD4) 0Eh = Interrupt Source: Compare down event (CCD5) 0Fh = Interrupt Source: Compare down event (CCU4) 10h = Interrupt Source: Compare down event (CCU5) 19h = Interrupt Source: Fault Event generated an interrupt. (F) 1Ah = Interrupt Source: Trigger overflow (TOV) 1Bh = Interrupt Source: Repeat Counter Zero (REPC) 1Ch = Interrupt Source: Direction Change (DC) 1Dh = Interrupt Source:QEI Incorrect state transition error (QEIERR)

21 IMASK Register (Offset = 1028h) [Reset = 00000000h]

IMASK is shown in Figure 32-100 and described in Table 32-98.

Return to the Summary Table.

Interrupt Mask. If a bit is set, then corresponding interrupt is un-masked. Un-masking the interrupt causes the raw interrupt to be visible in IIDX, as well as MIS."

Figure 32-100. IMASK Register

31	30	29	28	27	26	25	24
RESERVED			RESERVED	RESERVED	RESERVED	TOV	RESERVED
R/W-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
RESERVED							
R/W-0h							
15	14	13	12	11	10	9	8
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	CCU1	CCU0
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
RESERVED	RESERVED	CCD1	CCD0	RESERVED		L	Z
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h		R/W-0h	R/W-0h

Table 32-174. IMASK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R/W	0h	
28	RESERVED	R/W	0h	Reserved
27	RESERVED	R/W	0h	Reserved
26	RESERVED	R/W	0h	Reserved
25	TOV	R/W	0h	Trigger Overflow Event mask 0h = Disable Event 1h = Enable Event
24	RESERVED	R/W	0h	Reserved
23-16	RESERVED	R/W	0h	
15	RESERVED	R/W	0h	Reserved
14	RESERVED	R/W	0h	Reserved
13	RESERVED	R/W	0h	Reserved
12	RESERVED	R/W	0h	Reserved
11	RESERVED	R/W	0h	Reserved
10	RESERVED	R/W	0h	Reserved
9	CCU1	R/W	0h	Capture or Compare UP event mask CCP1 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
8	CCU0	R/W	0h	Capture or Compare UP event mask CCP0 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
7	RESERVED	R/W	0h	Reserved
6	RESERVED	R/W	0h	Reserved
5	CCD1	R/W	0h	Capture or Compare DN event mask CCP1 0h = Clear Interrupt Mask 1h = Set Interrupt Mask

Table 32-174. IMASK Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
4	CCD0	R/W	0h	Capture or Compare DN event mask CCP0 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
3-2	RESERVED	R/W	0h	
1	L	R/W	0h	Load Event mask 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
0	Z	R/W	0h	Zero Event mask 0h = Disable Event 1h = Enable Event

22 RIS Register (Offset = 1030h) [Reset = 00000000h]

RIS is shown in Figure 32-101 and described in Table 32-99.

Return to the Summary Table.

Raw interrupt status. Reflects all pending interrupts, regardless of masking. The RIS register allows the user to implement a poll scheme. A flag set in this register can be cleared by writing 1 to the ICLR register bit even if the corresponding IMASK bit is not enabled.

Figure 32-101. RIS Register

31	30	29	28	27	26	25	24
RESERVED			RESERVED	RESERVED	RESERVED	TOV	RESERVED
R-0h			R-0h	R-0h	R-0h	R-0h	R-0h
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	CCU1	CCU0
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
7	6	5	4	3	2	1	0
RESERVED	RESERVED	CCD1	CCD0	RESERVED		L	Z
R-0h	R-0h	R-0h	R-0h	R-0h		R-0h	R-0h

Table 32-176. RIS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R	0h	
28	RESERVED	R	0h	Reserved
27	RESERVED	R	0h	Reserved
26	RESERVED	R	0h	Reserved
25	TOV	R	0h	Trigger overflow 0h = Event Cleared 1h = Event Set
24	RESERVED	R	0h	Reserved
23-16	RESERVED	R	0h	
15	RESERVED	R	0h	Reserved
14	RESERVED	R	0h	Reserved
13	RESERVED	R	0h	Reserved
12	RESERVED	R	0h	Reserved
11	RESERVED	R	0h	Reserved
10	RESERVED	R	0h	Reserved
9	CCU1	R	0h	Capture or compare up event generated an interrupt CCP1 0h = Event Cleared 1h = Event Set
8	CCU0	R	0h	Capture or compare up event generated an interrupt CCP0 0h = Event Cleared 1h = Event Set
7	RESERVED	R	0h	Reserved
6	RESERVED	R	0h	Reserved
5	CCD1	R	0h	Capture or compare down event generated an interrupt CCP1 0h = Event Cleared 1h = Event Set

Table 32-176. RIS Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
4	CCD0	R	0h	Capture or compare down event generated an interrupt CCP0 0h = Event Cleared 1h = Event Set
3-2	RESERVED	R	0h	
1	L	R	0h	Load event generated an interrupt. 0h = Event Cleared 1h = Event Set
0	Z	R	0h	Zero event generated an interrupt. 0h = Event Cleared 1h = Event Set

23 MIS Register (Offset = 1038h) [Reset = 00000000h]

MIS is shown in Figure 32-102 and described in Table 32-100.

Return to the Summary Table.

Masked interrupt status. This is an AND of the IMASK and RIS registers.

Figure 32-102. MIS Register

31	30	29	28	27	26	25	24
RESERVED			RESERVED	RESERVED	RESERVED	TOV	RESERVED
R-0h			R-0h	R-0h	R-0h	R-0h	R-0h
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	CCU1	CCU0
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
7	6	5	4	3	2	1	0
RESERVED	RESERVED	CCD1	CCD0	RESERVED		L	Z
R-0h	R-0h	R-0h	R-0h	R-0h		R-0h	R-0h

Table 32-178. MIS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R	0h	
28	RESERVED	R	0h	Reserved
27	RESERVED	R	0h	Reserved
26	RESERVED	R	0h	Reserved
25	TOV	R	0h	Trigger overflow 0h = Event Cleared 1h = Event Set
24	RESERVED	R	0h	Reserved
23-16	RESERVED	R	0h	
15	RESERVED	R	0h	Reserved
14	RESERVED	R	0h	Reserved
13	RESERVED	R	0h	Reserved
12	RESERVED	R	0h	Reserved
11	RESERVED	R	0h	Reserved
10	RESERVED	R	0h	Reserved
9	CCU1	R	0h	Capture or compare up event generated an interrupt CCP1 0h = Event Cleared 1h = Event Set
8	CCU0	R	0h	Capture or compare up event generated an interrupt CCP0 0h = Event Cleared 1h = Event Set
7	RESERVED	R	0h	Reserved
6	RESERVED	R	0h	Reserved
5	CCD1	R	0h	Capture or compare down event generated an interrupt CCP1 0h = Event Cleared 1h = Event Set

Table 32-178. MIS Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
4	CCD0	R	0h	Capture or compare down event generated an interrupt CCP0 0h = Event Cleared 1h = Event Set
3-2	RESERVED	R	0h	
1	L	R	0h	Load event generated an interrupt. 0h = Event Cleared 1h = Event Set
0	Z	R	0h	Zero event generated an interrupt. 0h = Event Cleared 1h = Event Set

24 ISET Register (Offset = 1040h) [Reset = 00000000h]

ISET is shown in Figure 32-103 and described in Table 32-101.

Return to the Summary Table.

Interrupt set. Allows interrupts to be set by software (useful in diagnostics and safety checks). Writing a 1 to a bit in ISET will set the event and therefore the related RIS bit also gets set. If the interrupt is enabled through the mask, then the corresponding MIS bit is also set.

Figure 32-103. ISET Register

31	30	29	28	27	26	25	24
RESERVED			RESERVED	RESERVED	RESERVED	TOV	RESERVED
W-0h			W-0h	W-0h	W-0h	W-0h	W-0h
23	22	21	20	19	18	17	16
RESERVED							
W-0h							
15	14	13	12	11	10	9	8
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	CCU1	CCU0
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h
7	6	5	4	3	2	1	0
RESERVED	RESERVED	CCD1	CCD0	RESERVED		L	Z
W-0h	W-0h	W-0h	W-0h	W-0h		W-0h	W-0h

Table 32-180. ISET Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	W	0h	
28	RESERVED	W	0h	Reserved
27	RESERVED	W	0h	Reserved
26	RESERVED	W	0h	Reserved
25	TOV	W	0h	Trigger Overflow event SET 0h = Writing 0 has no effect. 1h = Event Set
24	RESERVED	W	0h	Reserved
23-16	RESERVED	W	0h	
15	RESERVED	W	0h	Reserved
14	RESERVED	W	0h	Reserved
13	RESERVED	W	0h	Reserved
12	RESERVED	W	0h	Reserved
11	RESERVED	W	0h	Reserved
10	RESERVED	W	0h	Reserved
9	CCU1	W	0h	Capture or compare up event SET 0h = Writing 0 has no effect. 1h = Event Set
8	CCU0	W	0h	Capture or compare up event SET 0h = Writing 0 has no effect. 1h = Event Set
7	RESERVED	W	0h	Reserved
6	RESERVED	W	0h	Reserved
5	CCD1	W	0h	Capture or compare down event SET 0h = Writing 0 has no effect. 1h = Event Set

Table 32-180. ISET Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
4	CCD0	W	0h	Capture or compare down event SET 0h = Writing 0 has no effect. 1h = Event Set
3-2	RESERVED	W	0h	
1	L	W	0h	Load event SET 0h = Writing 0 has no effect. 1h = Event Set
0	Z	W	0h	Zero event SET 0h = Writing 0 has no effect. 1h = Event Set

25 ICLR Register (Offset = 1048h) [Reset = 00000000h]

ICLR is shown in Figure 32-104 and described in Table 32-102.

Return to the Summary Table.

Interrupt clear. Write a 1 to clear corresponding Interrupt.

Figure 32-104. ICLR Register

31	30	29	28	27	26	25	24
RESERVED			RESERVED	RESERVED	RESERVED	TOV	RESERVED
W-0h			W-0h	W-0h	W-0h	W-0h	W-0h
23	22	21	20	19	18	17	16
RESERVED							
W-0h							
15	14	13	12	11	10	9	8
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	CCU1	CCU0
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h
7	6	5	4	3	2	1	0
RESERVED	RESERVED	CCD1	CCD0	RESERVED		L	Z
W-0h	W-0h	W-0h	W-0h	W-0h		W-0h	W-0h

Table 32-182. ICLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	W	0h	
28	RESERVED	W	0h	Reserved
27	RESERVED	W	0h	Reserved
26	RESERVED	W	0h	Reserved
25	TOV	W	0h	Trigger Overflow event CLEAR 0h = Writing 0 has no effect. 1h = Event Clear
24	RESERVED	W	0h	Reserved
23-16	RESERVED	W	0h	
15	RESERVED	W	0h	Reserved
14	RESERVED	W	0h	Reserved
13	RESERVED	W	0h	Reserved
12	RESERVED	W	0h	Reserved
11	RESERVED	W	0h	Reserved
10	RESERVED	W	0h	Reserved
9	CCU1	W	0h	Capture or compare up event CLEAR 0h = Writing 0 has no effect. 1h = Event Clear
8	CCU0	W	0h	Capture or compare up event CLEAR 0h = Writing 0 has no effect. 1h = Event Clear
7	RESERVED	W	0h	Reserved
6	RESERVED	W	0h	Reserved
5	CCD1	W	0h	Capture or compare down event CLEAR 0h = Writing 0 has no effect. 1h = Event Clear

Table 32-182. ICLR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
4	CCD0	W	0h	Capture or compare down event CLEAR 0h = Writing 0 has no effect. 1h = Event Clear
3-2	RESERVED	W	0h	
1	L	W	0h	Load event CLEAR 0h = Writing 0 has no effect. 1h = Event Clear
0	Z	W	0h	Zero event CLEAR 0h = Writing 0 has no effect. 1h = Event Clear

26 IIDX Register (Offset = 1050h) [Reset = 00000000h]

IIDX is shown in Figure 32-105 and described in Table 32-103.

Return to the Summary Table.

This register provides the highest priority enabled interrupt index. Value 0x00 means no event pending. Interrupt 1 is the highest priority, IIDX next highest, 4, 8, ... IIDX^31 is the least priority. That is, the least bit position that is set to 1 denotes the highest priority pending interrupt. The priority order is fixed. However, users can implement their own prioritization schemes using other registers that expose the full set of interrupts that have occurred. On each read, only one interrupt is indicated. On a read, the current interrupt (highest priority) is automatically cleared by the hardware and the corresponding interrupt flag in [RIS] and [MIS] are cleared as well. After a read from the CPU (not from the debug interface), the register is updated with the next highest priority interrupt, if none are pending, then it should display 0x0.

Figure 32-105. IIDX Register

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

Table 32-184. IIDX Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	
7-0	STAT	R	0h	Interrupt index status 00h = No interrupt pending 01h = Interrupt Source: Zero event (Z) 02h = nterrupt Source: Load event (L) 05h = Interrupt Source: Capture or compare down event (CCD0) 06h = Interrupt Source: Capture or compare down event (CCD1) 07h = Interrupt Source: Capture or compare down event (CCD2) 08h = Interrupt Source: Capture or compare down event (CCD3) 09h = Interrupt Source: Capture or compare up event (CCU0) 0Ah = Interrupt Source: Capture or compare up event (CCU1) 0Bh = Interrupt Source: Capture or compare up event (CCU2) 0Ch = Interrupt Source: Capture or compare up event (CCU3) 0Dh = Interrupt Source: Compare down event (CCD4) 0Eh = Interrupt Source: Compare down event (CCD5) 0Fh = Interrupt Source: Compare down event (CCU4) 10h = Interrupt Source: Compare down event (CCU5) 19h = Interrupt Source: Fault Event generated an interrupt. (F) 1Ah = Interrupt Source: Trigger overflow (TOV) 1Bh = Interrupt Source: Repeat Counter Zero (REPC) 1Ch = Interrupt Source: Direction Change (DC) 1Dh = Interrupt Source:QEI Incorrect state transition error (QEIERR)

27 IMASK Register (Offset = 1058h) [Reset = 00000000h]

IMASK is shown in Figure 32-106 and described in Table 32-104.

Return to the Summary Table.

Interrupt Mask. If a bit is set, then corresponding interrupt is un-masked. Un-masking the interrupt causes the raw interrupt to be visible in IIDX, as well as MIS."

Figure 32-106. IMASK Register

31	30	29	28	27	26	25	24
RESERVED			RESERVED	RESERVED	RESERVED	TOV	RESERVED
R/W-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
RESERVED							
R/W-0h							
15	14	13	12	11	10	9	8
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	CCU1	CCU0
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
RESERVED	RESERVED	CCD1	CCD0	RESERVED		L	Z
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h		R/W-0h	R/W-0h

Table 32-186. IMASK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R/W	0h	
28	RESERVED	R/W	0h	Reserved
27	RESERVED	R/W	0h	Reserved
26	RESERVED	R/W	0h	Reserved
25	TOV	R/W	0h	Trigger Overflow Event mask 0h = Disable Event 1h = Enable Event
24	RESERVED	R/W	0h	Reserved
23-16	RESERVED	R/W	0h	
15	RESERVED	R/W	0h	Reserved
14	RESERVED	R/W	0h	Reserved
13	RESERVED	R/W	0h	Reserved
12	RESERVED	R/W	0h	Reserved
11	RESERVED	R/W	0h	Reserved
10	RESERVED	R/W	0h	Reserved
9	CCU1	R/W	0h	Capture or Compare UP event mask CCP1 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
8	CCU0	R/W	0h	Capture or Compare UP event mask CCP0 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
7	RESERVED	R/W	0h	Reserved
6	RESERVED	R/W	0h	Reserved
5	CCD1	R/W	0h	Capture or Compare DN event mask CCP1 0h = Clear Interrupt Mask 1h = Set Interrupt Mask

Table 32-186. IMASK Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
4	CCD0	R/W	0h	Capture or Compare DN event mask CCP0 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
3-2	RESERVED	R/W	0h	
1	L	R/W	0h	Load Event mask 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
0	Z	R/W	0h	Zero Event mask 0h = Disable Event 1h = Enable Event

28 RIS Register (Offset = 1060h) [Reset = 00000000h]

RIS is shown in Figure 32-107 and described in Table 32-105.

Return to the Summary Table.

Raw interrupt status. Reflects all pending interrupts, regardless of masking. The RIS register allows the user to implement a poll scheme. A flag set in this register can be cleared by writing 1 to the ICLR register bit even if the corresponding IMASK bit is not enabled.

Figure 32-107. RIS Register

31	30	29	28	27	26	25	24
RESERVED			RESERVED	RESERVED	RESERVED	TOV	RESERVED
R-0h			R-0h	R-0h	R-0h	R-0h	R-0h
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	CCU1	CCU0
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
7	6	5	4	3	2	1	0
RESERVED	RESERVED	CCD1	CCD0	RESERVED		L	Z
R-0h	R-0h	R-0h	R-0h	R-0h		R-0h	R-0h

Table 32-188. RIS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R	0h	
28	RESERVED	R	0h	Reserved
27	RESERVED	R	0h	Reserved
26	RESERVED	R	0h	Reserved
25	TOV	R	0h	Trigger overflow 0h = Event Cleared 1h = Event Set
24	RESERVED	R	0h	Reserved
23-16	RESERVED	R	0h	
15	RESERVED	R	0h	Reserved
14	RESERVED	R	0h	Reserved
13	RESERVED	R	0h	Reserved
12	RESERVED	R	0h	Reserved
11	RESERVED	R	0h	Reserved
10	RESERVED	R	0h	Reserved
9	CCU1	R	0h	Capture or compare up event generated an interrupt CCP1 0h = Event Cleared 1h = Event Set
8	CCU0	R	0h	Capture or compare up event generated an interrupt CCP0 0h = Event Cleared 1h = Event Set
7	RESERVED	R	0h	Reserved
6	RESERVED	R	0h	Reserved
5	CCD1	R	0h	Capture or compare down event generated an interrupt CCP1 0h = Event Cleared 1h = Event Set

Table 32-188. RIS Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
4	CCD0	R	0h	Capture or compare down event generated an interrupt CCP0 0h = Event Cleared 1h = Event Set
3-2	RESERVED	R	0h	
1	L	R	0h	Load event generated an interrupt. 0h = Event Cleared 1h = Event Set
0	Z	R	0h	Zero event generated an interrupt. 0h = Event Cleared 1h = Event Set

29 MIS Register (Offset = 1068h) [Reset = 00000000h]

MIS is shown in Figure 32-108 and described in Table 32-106.

Return to the Summary Table.

Masked interrupt status. This is an AND of the IMASK and RIS registers.

Figure 32-108. MIS Register

31	30	29	28	27	26	25	24
RESERVED			RESERVED	RESERVED	RESERVED	TOV	RESERVED
R-0h			R-0h	R-0h	R-0h	R-0h	R-0h
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	CCU1	CCU0
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
7	6	5	4	3	2	1	0
RESERVED	RESERVED	CCD1	CCD0	RESERVED		L	Z
R-0h	R-0h	R-0h	R-0h	R-0h		R-0h	R-0h

Table 32-190. MIS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R	0h	
28	RESERVED	R	0h	Reserved
27	RESERVED	R	0h	Reserved
26	RESERVED	R	0h	Reserved
25	TOV	R	0h	Trigger overflow 0h = Event Cleared 1h = Event Set
24	RESERVED	R	0h	Reserved
23-16	RESERVED	R	0h	
15	RESERVED	R	0h	Reserved
14	RESERVED	R	0h	Reserved
13	RESERVED	R	0h	Reserved
12	RESERVED	R	0h	Reserved
11	RESERVED	R	0h	Reserved
10	RESERVED	R	0h	Reserved
9	CCU1	R	0h	Capture or compare up event generated an interrupt CCP1 0h = Event Cleared 1h = Event Set
8	CCU0	R	0h	Capture or compare up event generated an interrupt CCP0 0h = Event Cleared 1h = Event Set
7	RESERVED	R	0h	Reserved
6	RESERVED	R	0h	Reserved
5	CCD1	R	0h	Capture or compare down event generated an interrupt CCP1 0h = Event Cleared 1h = Event Set

Table 32-190. MIS Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
4	CCD0	R	0h	Capture or compare down event generated an interrupt CCP0 0h = Event Cleared 1h = Event Set
3-2	RESERVED	R	0h	
1	L	R	0h	Load event generated an interrupt. 0h = Event Cleared 1h = Event Set
0	Z	R	0h	Zero event generated an interrupt. 0h = Event Cleared 1h = Event Set

30 ISET Register (Offset = 1070h) [Reset = 00000000h]

ISET is shown in Figure 32-109 and described in Table 32-107.

Return to the Summary Table.

Interrupt set. Allows interrupts to be set by software (useful in diagnostics and safety checks). Writing a 1 to a bit in ISET will set the event and therefore the related RIS bit also gets set. If the interrupt is enabled through the mask, then the corresponding MIS bit is also set.

Figure 32-109. ISET Register

31	30	29	28	27	26	25	24
RESERVED			RESERVED	RESERVED	RESERVED	TOV	RESERVED
W-0h			W-0h	W-0h	W-0h	W-0h	W-0h
23	22	21	20	19	18	17	16
RESERVED							
W-0h							
15	14	13	12	11	10	9	8
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	CCU1	CCU0
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h
7	6	5	4	3	2	1	0
RESERVED	RESERVED	CCD1	CCD0	RESERVED		L	Z
W-0h	W-0h	W-0h	W-0h	W-0h		W-0h	W-0h

Table 32-192. ISET Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	W	0h	
28	RESERVED	W	0h	Reserved
27	RESERVED	W	0h	Reserved
26	RESERVED	W	0h	Reserved
25	TOV	W	0h	Trigger Overflow event SET 0h = Writing 0 has no effect. 1h = Event Set
24	RESERVED	W	0h	Reserved
23-16	RESERVED	W	0h	
15	RESERVED	W	0h	Reserved
14	RESERVED	W	0h	Reserved
13	RESERVED	W	0h	Reserved
12	RESERVED	W	0h	Reserved
11	RESERVED	W	0h	Reserved
10	RESERVED	W	0h	Reserved
9	CCU1	W	0h	Capture or compare up event SET 0h = Writing 0 has no effect. 1h = Event Set
8	CCU0	W	0h	Capture or compare up event SET 0h = Writing 0 has no effect. 1h = Event Set
7	RESERVED	W	0h	Reserved
6	RESERVED	W	0h	Reserved
5	CCD1	W	0h	Capture or compare down event SET 0h = Writing 0 has no effect. 1h = Event Set

Table 32-192. ISET Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
4	CCD0	W	0h	Capture or compare down event SET 0h = Writing 0 has no effect. 1h = Event Set
3-2	RESERVED	W	0h	
1	L	W	0h	Load event SET 0h = Writing 0 has no effect. 1h = Event Set
0	Z	W	0h	Zero event SET 0h = Writing 0 has no effect. 1h = Event Set

31 ICLR Register (Offset = 1078h) [Reset = 00000000h]

ICLR is shown in Figure 32-110 and described in Table 32-108.

Return to the Summary Table.

Interrupt clear. Write a 1 to clear corresponding Interrupt.

Figure 32-110. ICLR Register

31	30	29	28	27	26	25	24
RESERVED			RESERVED	RESERVED	RESERVED	TOV	RESERVED
W-0h			W-0h	W-0h	W-0h	W-0h	W-0h
23	22	21	20	19	18	17	16
RESERVED							
W-0h							
15	14	13	12	11	10	9	8
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	CCU1	CCU0
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h
7	6	5	4	3	2	1	0
RESERVED	RESERVED	CCD1	CCD0	RESERVED		L	Z
W-0h	W-0h	W-0h	W-0h	W-0h		W-0h	W-0h

Table 32-194. ICLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	W	0h	
28	RESERVED	W	0h	Reserved
27	RESERVED	W	0h	Reserved
26	RESERVED	W	0h	Reserved
25	TOV	W	0h	Trigger Overflow event CLEAR 0h = Writing 0 has no effect. 1h = Event Clear
24	RESERVED	W	0h	Reserved
23-16	RESERVED	W	0h	
15	RESERVED	W	0h	Reserved
14	RESERVED	W	0h	Reserved
13	RESERVED	W	0h	Reserved
12	RESERVED	W	0h	Reserved
11	RESERVED	W	0h	Reserved
10	RESERVED	W	0h	Reserved
9	CCU1	W	0h	Capture or compare up event CLEAR 0h = Writing 0 has no effect. 1h = Event Clear
8	CCU0	W	0h	Capture or compare up event CLEAR 0h = Writing 0 has no effect. 1h = Event Clear
7	RESERVED	W	0h	Reserved
6	RESERVED	W	0h	Reserved
5	CCD1	W	0h	Capture or compare down event CLEAR 0h = Writing 0 has no effect. 1h = Event Clear

Table 32-194. ICLR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
4	CCD0	W	0h	Capture or compare down event CLEAR 0h = Writing 0 has no effect. 1h = Event Clear
3-2	RESERVED	W	0h	
1	L	W	0h	Load event CLEAR 0h = Writing 0 has no effect. 1h = Event Clear
0	Z	W	0h	Zero event CLEAR 0h = Writing 0 has no effect. 1h = Event Clear

32 IIDX Register (Offset = 1080h) [Reset = 00000000h]

IIDX is shown in Figure 32-111 and described in Table 32-109.

Return to the Summary Table.

This register provides the highest priority enabled interrupt index. Value 0x00 means no event pending. Interrupt 1 is the highest priority, IIDX next highest, 4, 8, ... IIDX^31 is the least priority. That is, the least bit position that is set to 1 denotes the highest priority pending interrupt. The priority order is fixed. However, users can implement their own prioritization schemes using other registers that expose the full set of interrupts that have occurred. On each read, only one interrupt is indicated. On a read, the current interrupt (highest priority) is automatically cleared by the hardware and the corresponding interrupt flag in [RIS] and [MIS] are cleared as well. After a read from the CPU (not from the debug interface), the register is updated with the next highest priority interrupt, if none are pending, then it should display 0x0.

Figure 32-111. IIDX Register

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

Table 32-196. IIDX Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	
7-0	STAT	R	0h	Interrupt index status 00h = No interrupt pending 01h = Interrupt Source: Zero event (Z) 02h = nterrupt Source: Load event (L) 05h = Interrupt Source: Capture or compare down event (CCD0) 06h = Interrupt Source: Capture or compare down event (CCD1) 07h = Interrupt Source: Capture or compare down event (CCD2) 08h = Interrupt Source: Capture or compare down event (CCD3) 09h = Interrupt Source: Capture or compare up event (CCU0) 0Ah = Interrupt Source: Capture or compare up event (CCU1) 0Bh = Interrupt Source: Capture or compare up event (CCU2) 0Ch = Interrupt Source: Capture or compare up event (CCU3) 0Dh = Interrupt Source: Compare down event (CCD4) 0Eh = Interrupt Source: Compare down event (CCD5) 0Fh = Interrupt Source: Compare down event (CCU4) 10h = Interrupt Source: Compare down event (CCU5) 19h = Interrupt Source: Fault Event generated an interrupt. (F) 1Ah = Interrupt Source: Trigger overflow (TOV) 1Bh = Interrupt Source: Repeat Counter Zero (REPC) 1Ch = Interrupt Source: Direction Change (DC) 1Dh = Interrupt Source:QEI Incorrect state transition error (QEIERR)

33 IMASK Register (Offset = 1088h) [Reset = 00000000h]

IMASK is shown in Figure 32-112 and described in Table 32-110.

Return to the Summary Table.

Interrupt Mask. If a bit is set, then corresponding interrupt is un-masked. Un-masking the interrupt causes the raw interrupt to be visible in IIDX, as well as MIS."

Figure 32-112. IMASK Register

31	30	29	28	27	26	25	24
RESERVED			RESERVED	RESERVED	RESERVED	TOV	RESERVED
R/W-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
RESERVED							
R/W-0h							
15	14	13	12	11	10	9	8
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	CCU1	CCU0
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
RESERVED	RESERVED	CCD1	CCD0	RESERVED		L	Z
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h		R/W-0h	R/W-0h

Table 32-198. IMASK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R/W	0h	
28	RESERVED	R/W	0h	Reserved
27	RESERVED	R/W	0h	Reserved
26	RESERVED	R/W	0h	Reserved
25	TOV	R/W	0h	Trigger Overflow Event mask 0h = Disable Event 1h = Enable Event
24	RESERVED	R/W	0h	Reserved
23-16	RESERVED	R/W	0h	
15	RESERVED	R/W	0h	Reserved
14	RESERVED	R/W	0h	Reserved
13	RESERVED	R/W	0h	Reserved
12	RESERVED	R/W	0h	Reserved
11	RESERVED	R/W	0h	Reserved
10	RESERVED	R/W	0h	Reserved
9	CCU1	R/W	0h	Capture or Compare UP event mask CCP1 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
8	CCU0	R/W	0h	Capture or Compare UP event mask CCP0 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
7	RESERVED	R/W	0h	Reserved
6	RESERVED	R/W	0h	Reserved
5	CCD1	R/W	0h	Capture or Compare DN event mask CCP1 0h = Clear Interrupt Mask 1h = Set Interrupt Mask

Table 32-198. IMASK Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
4	CCD0	R/W	0h	Capture or Compare DN event mask CCP0 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
3-2	RESERVED	R/W	0h	
1	L	R/W	0h	Load Event mask 0h = Clear Interrupt Mask 1h = Set Interrupt Mask
0	Z	R/W	0h	Zero Event mask 0h = Disable Event 1h = Enable Event

34 RIS Register (Offset = 1090h) [Reset = 00000000h]

RIS is shown in Figure 32-113 and described in Table 32-111.

Return to the Summary Table.

Raw interrupt status. Reflects all pending interrupts, regardless of masking. The RIS register allows the user to implement a poll scheme. A flag set in this register can be cleared by writing 1 to the ICLR register bit even if the corresponding IMASK bit is not enabled.

Figure 32-113. RIS Register

31	30	29	28	27	26	25	24
RESERVED			RESERVED	RESERVED	RESERVED	TOV	RESERVED
R-0h			R-0h	R-0h	R-0h	R-0h	R-0h
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	CCU1	CCU0
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
7	6	5	4	3	2	1	0
RESERVED	RESERVED	CCD1	CCD0	RESERVED		L	Z
R-0h	R-0h	R-0h	R-0h	R-0h		R-0h	R-0h

Table 32-200. RIS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R	0h	
28	RESERVED	R	0h	Reserved
27	RESERVED	R	0h	Reserved
26	RESERVED	R	0h	Reserved
25	TOV	R	0h	Trigger overflow 0h = Event Cleared 1h = Event Set
24	RESERVED	R	0h	Reserved
23-16	RESERVED	R	0h	
15	RESERVED	R	0h	Reserved
14	RESERVED	R	0h	Reserved
13	RESERVED	R	0h	Reserved
12	RESERVED	R	0h	Reserved
11	RESERVED	R	0h	Reserved
10	RESERVED	R	0h	Reserved
9	CCU1	R	0h	Capture or compare up event generated an interrupt CCP1 0h = Event Cleared 1h = Event Set
8	CCU0	R	0h	Capture or compare up event generated an interrupt CCP0 0h = Event Cleared 1h = Event Set
7	RESERVED	R	0h	Reserved
6	RESERVED	R	0h	Reserved
5	CCD1	R	0h	Capture or compare down event generated an interrupt CCP1 0h = Event Cleared 1h = Event Set

Table 32-200. RIS Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
4	CCD0	R	0h	Capture or compare down event generated an interrupt CCP0 0h = Event Cleared 1h = Event Set
3-2	RESERVED	R	0h	
1	L	R	0h	Load event generated an interrupt. 0h = Event Cleared 1h = Event Set
0	Z	R	0h	Zero event generated an interrupt. 0h = Event Cleared 1h = Event Set

35 MIS Register (Offset = 1098h) [Reset = 00000000h]

MIS is shown in Figure 32-114 and described in Table 32-112.

Return to the Summary Table.

Masked interrupt status. This is an AND of the IMASK and RIS registers.

Figure 32-114. MIS Register

31	30	29	28	27	26	25	24
RESERVED			RESERVED	RESERVED	RESERVED	TOV	RESERVED
R-0h			R-0h	R-0h	R-0h	R-0h	R-0h
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	CCU1	CCU0
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
7	6	5	4	3	2	1	0
RESERVED	RESERVED	CCD1	CCD0	RESERVED		L	Z
R-0h	R-0h	R-0h	R-0h	R-0h		R-0h	R-0h

Table 32-202. MIS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R	0h	
28	RESERVED	R	0h	Reserved
27	RESERVED	R	0h	Reserved
26	RESERVED	R	0h	Reserved
25	TOV	R	0h	Trigger overflow 0h = Event Cleared 1h = Event Set
24	RESERVED	R	0h	Reserved
23-16	RESERVED	R	0h	
15	RESERVED	R	0h	Reserved
14	RESERVED	R	0h	Reserved
13	RESERVED	R	0h	Reserved
12	RESERVED	R	0h	Reserved
11	RESERVED	R	0h	Reserved
10	RESERVED	R	0h	Reserved
9	CCU1	R	0h	Capture or compare up event generated an interrupt CCP1 0h = Event Cleared 1h = Event Set
8	CCU0	R	0h	Capture or compare up event generated an interrupt CCP0 0h = Event Cleared 1h = Event Set
7	RESERVED	R	0h	Reserved
6	RESERVED	R	0h	Reserved
5	CCD1	R	0h	Capture or compare down event generated an interrupt CCP1 0h = Event Cleared 1h = Event Set

Table 32-202. MIS Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
4	CCD0	R	0h	Capture or compare down event generated an interrupt CCP0 0h = Event Cleared 1h = Event Set
3-2	RESERVED	R	0h	
1	L	R	0h	Load event generated an interrupt. 0h = Event Cleared 1h = Event Set
0	Z	R	0h	Zero event generated an interrupt. 0h = Event Cleared 1h = Event Set

36 ISET Register (Offset = 10A0h) [Reset = 00000000h]

ISET is shown in Figure 32-115 and described in Table 32-113.

Return to the Summary Table.

Interrupt set. Allows interrupts to be set by software (useful in diagnostics and safety checks). Writing a 1 to a bit in ISET will set the event and therefore the related RIS bit also gets set. If the interrupt is enabled through the mask, then the corresponding MIS bit is also set.

Figure 32-115. ISET Register

31	30	29	28	27	26	25	24
RESERVED			RESERVED	RESERVED	RESERVED	TOV	RESERVED
W-0h			W-0h	W-0h	W-0h	W-0h	W-0h
23	22	21	20	19	18	17	16
RESERVED							
W-0h							
15	14	13	12	11	10	9	8
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	CCU1	CCU0
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h
7	6	5	4	3	2	1	0
RESERVED	RESERVED	CCD1	CCD0	RESERVED		L	Z
W-0h	W-0h	W-0h	W-0h	W-0h		W-0h	W-0h

Table 32-204. ISET Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	W	0h	
28	RESERVED	W	0h	Reserved
27	RESERVED	W	0h	Reserved
26	RESERVED	W	0h	Reserved
25	TOV	W	0h	Trigger Overflow event SET 0h = Writing 0 has no effect. 1h = Event Set
24	RESERVED	W	0h	Reserved
23-16	RESERVED	W	0h	
15	RESERVED	W	0h	Reserved
14	RESERVED	W	0h	Reserved
13	RESERVED	W	0h	Reserved
12	RESERVED	W	0h	Reserved
11	RESERVED	W	0h	Reserved
10	RESERVED	W	0h	Reserved
9	CCU1	W	0h	Capture or compare up event SET 0h = Writing 0 has no effect. 1h = Event Set
8	CCU0	W	0h	Capture or compare up event SET 0h = Writing 0 has no effect. 1h = Event Set
7	RESERVED	W	0h	Reserved
6	RESERVED	W	0h	Reserved
5	CCD1	W	0h	Capture or compare down event SET 0h = Writing 0 has no effect. 1h = Event Set

Table 32-204. ISET Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
4	CCD0	W	0h	Capture or compare down event SET 0h = Writing 0 has no effect. 1h = Event Set
3-2	RESERVED	W	0h	
1	L	W	0h	Load event SET 0h = Writing 0 has no effect. 1h = Event Set
0	Z	W	0h	Zero event SET 0h = Writing 0 has no effect. 1h = Event Set

37 ICLR Register (Offset = 10A8h) [Reset = 00000000h]

ICLR is shown in Figure 32-116 and described in Table 32-114.

Return to the Summary Table.

Interrupt clear. Write a 1 to clear corresponding Interrupt.

Figure 32-116. ICLR Register

31	30	29	28	27	26	25	24
RESERVED			RESERVED	RESERVED	RESERVED	TOV	RESERVED
W-0h			W-0h	W-0h	W-0h	W-0h	W-0h
23	22	21	20	19	18	17	16
RESERVED							
W-0h							
15	14	13	12	11	10	9	8
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	CCU1	CCU0
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h
7	6	5	4	3	2	1	0
RESERVED	RESERVED	CCD1	CCD0	RESERVED		L	Z
W-0h	W-0h	W-0h	W-0h	W-0h		W-0h	W-0h

Table 32-206. ICLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	W	0h	
28	RESERVED	W	0h	Reserved
27	RESERVED	W	0h	Reserved
26	RESERVED	W	0h	Reserved
25	TOV	W	0h	Trigger Overflow event CLEAR 0h = Writing 0 has no effect. 1h = Event Clear
24	RESERVED	W	0h	Reserved
23-16	RESERVED	W	0h	
15	RESERVED	W	0h	Reserved
14	RESERVED	W	0h	Reserved
13	RESERVED	W	0h	Reserved
12	RESERVED	W	0h	Reserved
11	RESERVED	W	0h	Reserved
10	RESERVED	W	0h	Reserved
9	CCU1	W	0h	Capture or compare up event CLEAR 0h = Writing 0 has no effect. 1h = Event Clear
8	CCU0	W	0h	Capture or compare up event CLEAR 0h = Writing 0 has no effect. 1h = Event Clear
7	RESERVED	W	0h	Reserved
6	RESERVED	W	0h	Reserved
5	CCD1	W	0h	Capture or compare down event CLEAR 0h = Writing 0 has no effect. 1h = Event Clear

Table 32-206. ICLR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
4	CCD0	W	0h	Capture or compare down event CLEAR 0h = Writing 0 has no effect. 1h = Event Clear
3-2	RESERVED	W	0h	
1	L	W	0h	Load event CLEAR 0h = Writing 0 has no effect. 1h = Event Clear
0	Z	W	0h	Zero event CLEAR 0h = Writing 0 has no effect. 1h = Event Clear

38 EVT_MODE Register (Offset = 10E0h) [Reset = 00000029h]

EVT_MODE is shown in Figure 32-117 and described in Table 32-115.

Return to the Summary Table.

Event mode register. It is used to select whether each line is disabled, in software mode (software clears the RIS) or in hardware mode (hardware clears the RIS)

Figure 32-117. EVT_MODE Register

31	30	29	28	27	26	25	24
RESERVED							
R/W-0h							
23	22	21	20	19	18	17	16
RESERVED							
R/W-0h							
15	14	13	12	11	10	9	8
RESERVED							
R/W-0h							
7	6	5	4	3	2	1	0
RESERVED		EVT2_CFG		EVT1_CFG		EVT0_CFG	
R/W-0h		R-2h		R-2h		R-1h	

Table 32-208. EVT_MODE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-6	RESERVED	R/W	0h	
5-4	EVT2_CFG	R	2h	Event line mode select for event corresponding to GEN_EVENT1 0h = The interrupt or event line is disabled. 1h = The interrupt or event line is in software mode. Software must clear the RIS. 2h = The interrupt or event line is in hardware mode. The hardware (another module) clears automatically the associated RIS flag.
3-2	EVT1_CFG	R	2h	Event line mode select for event corresponding to GEN_EVENT0 0h = The interrupt or event line is disabled. 1h = The interrupt or event line is in software mode. Software must clear the RIS. 2h = The interrupt or event line is in hardware mode. The hardware (another module) clears automatically the associated RIS flag.
1-0	EVT0_CFG	R	1h	Event line mode select for event corresponding to CPU_INT 0h = The interrupt or event line is disabled. 1h = The interrupt or event line is in software mode. Software must clear the RIS. 2h = The interrupt or event line is in hardware mode. The hardware (another module) clears automatically the associated RIS flag.

39 DESC Register (Offset = 10FCh) [Reset = 00000000h]

DESC is shown in Figure 32-118 and described in Table 32-116.

Return to the Summary Table.

This register identifies the peripheral and its exact version.

Figure 32-118. DESC Register

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

Table 32-210. DESC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	MODULEID	R	1111h	Module identification contains a unique peripheral identification number. The assignments are maintained in a central database for all of the platform modules to ensure uniqueness. 0h = Smallest value FFFFh = Highest possible value
15-12	FEATUREVER	R	0h	Feature Set for the module *instance* 0h = Smallest value Fh = Highest possible value
11-8	INSTNUM	R	0h	Instance Number within the device. This will be a parameter to the RTL for modules that can have multiple instances 0h = Smallest value Fh = Highest possible value
7-4	MAJREV	R	0h	Major rev of the IP 0h = Smallest value Fh = Highest possible value
3-0	MINREV	R	0h	Minor rev of the IP 0h = Smallest value Fh = Highest possible value

40 CCPD Register (Offset = 1100h) [Reset = 00000000h]

CCPD is shown in Figure 32-119 and described in Table 32-117.

Return to the Summary Table.

CCP Direction. Controls whether CCP is used as an input or an output.

Figure 32-119. CCPD Register

31	30	29	28	27	26	25	24
RESERVED							
R/W-0h							
23	22	21	20	19	18	17	16
RESERVED							
R/W-0h							
15	14	13	12	11	10	9	8
RESERVED							
R/W-0h							
7	6	5	4	3	2	1	0
RESERVED				RESERVED	RESERVED	C0CCP1	C0CCP0
R/W-0h				R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 32-212. CCPD Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R/W	0h	
3	RESERVED	R/W	0h	Reserved
2	RESERVED	R/W	0h	Reserved
1	C0CCP1	R/W	0h	CCP1 direction 0h = Input 1h = Output
0	C0CCP0	R/W	0h	CCP0 direction 0h = Input 1h = Output

41 ODIS Register (Offset = 1104h) [Reset = 00000000h]

ODIS is shown in Figure 32-120 and described in Table 32-118.

Return to the Summary Table.

The ODIS register output is inverted and then ANDed with the output signal selected by the OCTL register CCPO field (before conditional inversion) to allow software the ability to hold the CCP output low during configuration or shutdown.

Figure 32-120. ODIS Register

31	30	29	28	27	26	25	24
RESERVED							
R/W-0h							
23	22	21	20	19	18	17	16
RESERVED							
R/W-0h							
15	14	13	12	11	10	9	8
RESERVED							
R/W-0h							
7	6	5	4	3	2	1	0
RESERVED				RESERVED	RESERVED	C0CCP1	C0CCP0
R/W-0h				R/W-0h	R/W-0h	R/W-0h	R/W-0h

Table 32-214. ODIS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R/W	0h	
3	RESERVED	R/W	0h	Reserved
2	RESERVED	R/W	0h	Reserved
1	C0CCP1	R/W	0h	Counter CCP1 Disable Mask Defines whether CCP0 of Counter n is forced low or not 0h = Output function as selected by the OCTL register CCPO field are provided to output inversion block. 1h = CCP output occpout[1] is forced low.
0	C0CCP0	R/W	0h	Counter CCP0 Disable Mask Defines whether CCP0 of Counter n is forced low or not 0h = Output function as selected by the OCTL register CCPO field are provided to output inversion block. 1h = CCP output occpout[0] is forced low.

42 CCLKCTL Register (Offset = 1108h) [Reset = 00000000h]

CCLKCTL is shown in Figure 32-121 and described in Table 32-119.

Return to the Summary Table.

The CCLKCTL register provides a SW mechanism for gating the TIMER clock if the module is expected not to be used but the power domain is alive. This effectively puts the IP in an IDLE state

Figure 32-121. CCLKCTL Register

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

Table 32-216. CCLKCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R/W	0h	
0	CLKEN	R/W	0h	Clock Enable Disables the clock gating to the module. SW has to explicitly program the value to 0 to gate the clock. 0h = Clock is disabled. 1h = Clock is enabled

ADVANCE INFORMATION

43 CPS Register (Offset = 110Ch) [Reset = 00000000h]

CPS is shown in Figure 32-122 and described in Table 32-120.

Return to the Summary Table.

The CPS register provides the value for the clock pre-scaler.

Figure 32-122. CPS Register

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

Table 32-218. CPS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R/W	0h	
7-0	PCNT	R/W	0h	Pre-Scale Count This field specifies the pre-scale count value. The selected TIMCLK source is divided by a value of (PCNT+1). A PCNT value of 0 divides TIMCLK by 1, effectively bypassing the divider. A PCNT value of greater than 0 divides the TIMCLK source generating a slower clock 0h = Minimum value FFh = Maximum Value

44 CPSV Register (Offset = 1110h) [Reset = 00000000h]

CPSV is shown in Figure 32-123 and described in Table 32-121.

Return to the Summary Table.

The CPSV register provides the ability to read the current clock prescale count value.

Figure 32-123. CPSV Register

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

Table 32-220. CPSV Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	
7-0	CPSVAL	R	0h	Current Prescale Count Value 0h = Minimum value FFh = Maximum Value

45 CTTRIGCTL Register (Offset = 1114h) [Reset = 00000000h]

CTTRIGCTL is shown in Figure 32-124 and described in Table 32-122.

Return to the Summary Table.

Cross Timer Trigger Control Register
This register is used to control the cross trigger connections for enables and faults of different timer instances in the same power domain. Please refer to sections Timer Module Cross Trigger (In/Out) and Fault Cross Triggering for details.

Figure 32-124. CTTRIGCTL Register

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

Table 32-222. CTTRIGCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-20	RESERVED	R/W	0h	
19-16	EVTCTTRIGSEL	R/W	0h	Used to Select the subscriber port that should be used for input cross trigger. 0h = Use FSUB0 as cross trigger source. 1h = Use FSUB1 as cross trigger source. 2h = Use Zero event as cross trigger source. 3h = Use Load event as cross trigger source. 4h = Use CCD0 event as cross trigger source. 5h = Use CCD1 event as cross trigger source. 6h = Use CCD2 event as cross trigger source. 7h = Use CCD3 event as cross trigger source. 8h = Use CCU0 event as cross trigger source. 9h = Use CCU1 event as cross trigger source. Ah = Use CCU2 event as cross trigger source. Bh = Use CCU3 event as cross trigger source.
15-2	RESERVED	R/W	0h	
1	EVTCTEN	R/W	0h	Enable the Input Trigger Conditions to the Timer module as a condition for Cross Triggers. 0h = Cross trigger generation disabled. 1h = Cross trigger generation enabled
0	CTEN	R/W	0h	Timer Cross trigger enable. This field is used to enable whether the SW or HW logic can generate a timer cross trigger event in the system. These cross triggers are connected to the respective timer trigger in of the other timer IPs in the SOC power domain. The timer cross trigger is essentially the combined logic of the HW and SW conditions controlling EN bit in the CTRCTL register. 0h = Cross trigger generation disabled. 1h = Cross trigger generation enabled

46 CTTRIG Register (Offset = 111Ch) [Reset = 00000000h]

CTTRIG is shown in Figure 32-125 and described in Table 32-123.

Return to the Summary Table.

Cross Timer Trigger Register
This register is used to trigger the timer instances connected and enabled using CTTRIGCTL and CTTRIGMSK registers.

Figure 32-125. CTTRIG Register

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

Table 32-224. CTTRIG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	W	0h	
0	TRIG	W	0h	Generate Cross Trigger This bit when programmed will generate a synchronized trigger condition all the cross trigger enabled Timer instances including current timer instance. 0h = Cross trigger generation disabled 1h = Generate Cross trigger pulse

ADVANCE INFORMATION

47 CTR Register (Offset = 1800h) [Reset = 00000000h]

CTR is shown in Figure 32-126 and described in Table 32-124.

Return to the Summary Table.

This is the TIMER counter register.
This can be set by SW. However, the writes will be unpredictable if the software tries to set a value while the counter is running.

Figure 32-126. CTR Register

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

Table 32-226. CTR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R/W	0h	
15-0	CCTR	R/W	0h	Current Counter value 0h = Minimum value FFFFFFFFh = Maximum Value

ADVANCE INFORMATION

48 CTRCTL Register (Offset = 1804h) [Reset = 0000FF80h]

CTRCTL is shown in Figure 32-127 and described in Table 32-125.

Return to the Summary Table.

This register provides control over the counter operation.
The configuration can change as well as setting the EN bit in a single write.
There is no requirement to change the configuration first and then do an additional write to set the EN bit.

Figure 32-127. CTRCTL Register

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

Table 32-228. CTRCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	RESERVED	R/W	0h	
29-28	CVAE	R/W	0h	Counter Value After Enable. This field specifies the initialization condition of the counter when the EN bit is changed from 0 to 1 by a write to the CTRCTL register. Note that an external event can also cause the EN bit to go active. 0h = The counter is set to the LOAD register value 1h = The counter value is unchanged from its current value which could have been initialized by software 2h = The counter is set to zero
27-25	RESERVED	R/W	0h	
24	RESERVED	R/W	0h	Reserved
23	RESERVED	R/W	0h	Reserved
22-20	RESERVED	R/W	0h	
19	RESERVED	R/W	0h	Reserved
18	RESERVED	R/W	0h	Reserved
17	DRB	R/W	0h	Debug Resume Behavior This bit specifies what the device does following the release/exit of debug mode. 0h = Resume counting 1h = Perform the action as specified by the CVAE field.
16	RESERVED	R/W	0h	

Table 32-228. CTRCTL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
15-13	CZC	R/W	7h	Counter Zero Control This field specifies what controls the counter operation with respect to zeroing the counter value. Encodings 1-3 are present based on the CCPC parameter value. Bits 4-5 are present based on the HQEI parameter value. Any encodings not provided are documented as reserved. 0h = CCCTL_0 ZCOND 1h = CCCTL_1 ZCOND 2h = CCCTL_2 ZCOND This value exists when there are 4 channels. 3h = CCCTL_3 ZCOND This value exists when there are 4 channels. 4h = Controlled by 2-input QEI mode This value exists when TIMER support QEI feature. 5h = Controlled by 3-input QEI mode This value exists when TIMER support QEI feature.
12-10	CAC	R/W	7h	Counter Advance Control. This field specifies what controls the counter operation with respect to advancing (incrementing or decrementing) the counter value. Encodings 1-3 are present based on the CCPC parameter value. Bits 4-5 are present based on the HQEI parameter value. Any encodings not provided are documented as reserved. 0h = CCCTL_0 ACOND 1h = CCCTL_1 ACOND 2h = CCCTL_2 ACOND This value exists when there are 4 channels. 3h = CCCTL_3 ACOND This value exists when there are 4 channels. 4h = Controlled by 2-input QEI mode This value exists when TIMER support QEI feature. 5h = Controlled by 3-input QEI mode This value exists when TIMER support QEI feature.
9-7	CLC	R/W	7h	Counter Load Control. This field specifies what controls the counter operation with respect to setting the counter to the LD register value. Encodings 1-3 are present based on the CCPC parameter value. Bits 4-5 are present based on the HQEI parameter value. Any encodings not provided are documented as reserved. 0h = CCCTL_0 LCOND 1h = CCCTL_1 LCOND 2h = CCCTL_2 LCOND This value exists when there are 4 channels. 3h = CCCTL_3 LCOND This value exists when there are 4 channels. 4h = Controlled by 2 input QEI mode. This value exists when TIMER support QEI feature. 5h = Controlled by 3 input QEI mode. This value exists when TIMER support QEI feature.
6	RESERVED	R/W	0h	
5-4	CM	R/W	0h	Count Mode 0h = Down 1h = Up/Down 2h = Counter counts up.

Table 32-228. CTRCTL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3-1	REPEAT	R/W	0h	Repeat. The repeat bit controls whether the counter continues to advance following a zero event, or the exiting of a debug or fault condition. If counting down, a zero event is followed by a load at the next advance condition. If counting up-down, a zero event is followed by an advance event (+1). The intent of encoding 3 is that if the debug condition is in effect, the generation of the load pulse is deferred until the debug condition is over. This allows the counter to reach zero before counting is suspended. 0h = Does not automatically advance following a zero event. 1h = Continues to advance following a zero event. 2h = Reserved 3h = Continues to advance following a zero event if the debug mode is not in effect, or following the release of the debug mode. 4h = Reserved
0	EN	R/W	0h	Counter Enable. This bit allows the timer to advance This bit is automatically cleared if REPEAT=0 (do not automatically reload) and the counter value equals zero. CPU Write: A register write that sets the EN bit, the counter value is set per the CVAE value. Hardware: This bit may also be set as the result of an LCOND or ZCOND condition being met and the counter value changed to the load value or zero value, respectively. 0h = Disabled 1h = Enabled

49 LOAD Register (Offset = 1808h) [Reset = 00000000h]

LOAD is shown in Figure 32-128 and described in Table 32-126.

Return to the Summary Table.

The contents of LOAD register are copied to CTR on any operation designated to do a "LOAD". The LOAD is used to compare with the CTR for generating a "Load Event" that can be used for interrupt, trigger, or signal generator actions.

Figure 32-128. LOAD Register

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

Table 32-230. LOAD Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R/W	0h	
15-0	LD	R/W	0h	Load Value 0h = Minimum value FFFFFFFFh = Maximum Value

ADVANCE INFORMATION

50 CC_01_y Register (Offset = 1810h + formula) [Reset = 00000000h]

CC_01_y is shown in Figure 32-129 and described in Table 32-127.

Return to the Summary Table.

The CC_01 register is a register that can be used as either a capture register, to capture the next CTR value on an event, or a compare to the current CTR to create an event. It cannot operate concurrently as both. There are two Capture-Compare slices of hardware for each counter, hence there are two CC_01 registers per timer. On a capture event, the next value of the CTR is loaded so that CTR and CC_01 (which captured) will be equal on the cycle that an interrupt or trigger is created from the capture action.

Offset = 1810h + (y * 4h); where y = 0h to 1h

Figure 32-129. CC_01_y Register

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

Table 32-232. CC_01_y Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RESERVED	R/W	0h	
15-0	CCVAL	R/W	0h	Capture or compare value 0h = Minimum value FFFFFFFFh = Maximum Value

51 CCCTL_01_y Register (Offset = 1830h + formula) [Reset = 00000000h]

CCCTL_01_y is shown in Figure 32-130 and described in Table 32-128.

Return to the Summary Table.

The CCCTL_01 registers control the operations of the respective CC registers and the counter.

Offset = 1830h + (y * 4h); where y = 0h to 1h

Figure 32-130. CCCTL_01_y Register

31	30	29	28	27	26	25	24
CC2SELD			CCACTUPD			RESERVED	CC2SELU
R/W-0h			R/W-0h			R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
CC2SELU		RESERVED	RESERVED			COC	RESERVED
R/W-0h		R/W-0h	R/W-0h			R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED	ZCOND			RESERVED	LCOND		
R/W-0h	R/W-0h			R/W-0h	R/W-0h		
7	6	5	4	3	2	1	0
RESERVED	ACOND			RESERVED	CCOND		
R/W-0h	R/W-0h			R/W-0h	R/W-0h		

Table 32-234. CCCTL_01_y Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	CC2SELD	R/W	0h	Selects the source second CCD event. 0h = Selects CCD from CC0. 1h = Selects CCD from CC1. 2h = Selects CCD from CC2. 3h = Selects CCD from CC3. 4h = Selects CCD from CC4. 5h = Selects CCD from CC5.

Table 32-234. CCCTL_01_y Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
28-26	CCACTUPD	R/W	0h	CCACT shadow register Update Method This field controls how updates to the CCACT shadow register are performed 0h = Value written to the CCACT register has immediate effect. 1h = Following a zero event (CTR=0) Writes to the CCACTx_y register are stored in shadow register and transferred to CCACTx_y in the TIMCLK cycle following CTR equals 0. 2h = Following a CCD event (CTR=CC_xy) Writes to the CCACTx_y register are stored in shadow register and transferred to CCACTx_y in the TIMCLK cycle following CTR equals the CCx_y register value. 3h = Following a CCU event (CTR=CC_xy) Writes to the CCACTx_y register are stored in shadow register and transferred to CCACTx_y in the TIMCLK cycle following CTR equals the CCx_y register value. 4h = Following a zero event (CTR=0) or load event (CTR = LOAD) Writes to the CCACTx_y register are stored in shadow register and transferred to CCACTx_y in the TIMCLK cycle following CTR equals 0 or CTR. Equals LDn. Note this update mechanism is defined for use only in configurations using up/down counting. This mode is not intended for use in down count configurations. 5h = Following a zero event (CTR=0) with repeat count also zero (RC=0). Writes to the CCACTx_y register are stored in shadow register and transferred to CCACTx_y in the TIMCLK cycle following CTR equals 0 and if RC equal 0. 6h = On a TRIG pulse, the value stored in CCACT_xy shadow register is loaded into CCACT_xy register.
25	RESERVED	R/W	0h	Reserved
24-22	CC2SELU	R/W	0h	Selects the source second CCU event. 0h = Selects CCU from CC0. 1h = Selects CCU from CC1. 2h = Selects CCU from CC2. 3h = Selects CCU from CC3. 4h = Selects CCU from CC4. 5h = Selects CCU from CC5.
21	RESERVED	R/W	0h	
20-18	RESERVED	R/W	0h	Reserved

Table 32-234. CCCTL_01_y Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
17	COC	R/W	0h	Capture or Compare. Specifies whether the corresponding CC register is used as a capture register or a compare register (never both). 0h = Compare 1h = Capture
16-15	RESERVED	R/W	0h	
14-12	ZCOND	R/W	0h	Zero Condition. This field specifies the condition that generates a zero pulse. 0h = CCP edges have no effect 1h = Rising edge of CCP or trigger assertion edge 2h = Falling edge of CCP or trigger de-assertion edge 3h = Either edge of CCP or trigger change (assertion/de-assertion edge)
11	RESERVED	R/W	0h	
10-8	LCOND	R/W	0h	Load Condition. Specifies the condition that generates a load pulse. 0h = CCP edges have no effect 1h = Rising edge of CCP or trigger assertion edge 2h = Falling edge of CCP or trigger de-assertion edge 3h = Either edge of CCP or trigger change (assertion/de-assertion edge)
7	RESERVED	R/W	0h	
6-4	ACOND	R/W	0h	Advance Condition. Specifies the condition that generates an advance pulse. 0h = Each TIMCLK 1h = Rising edge of CCP or trigger assertion edge 2h = Falling edge of CCP or trigger de-assertion edge 3h = Either edge of CCP or trigger change (assertion/de-assertion edge) 5h = CCP High or Trigger assertion (level)
3	RESERVED	R/W	0h	
2-0	CCOND	R/W	0h	Capture Condition. Specifies the condition that generates a capture pulse. 0h = None (never captures) 1h = Rising edge of CCP or trigger assertion edge 2h = Falling edge of CCP or trigger de-assertion edge 3h = Either edge of CCP or trigger change (assertion/de-assertion edge)

52 OCTL_01_y Register (Offset = 1850h + formula) [Reset = 00000000h]

OCTL_01_y is shown in Figure 32-131 and described in Table 32-129.

Return to the Summary Table.

The OCTL_01 register controls the CCP output of the Capture-Compare slice of the counter. This includes the ability to select the source of what is driven out along with initial condition values and final inversion options.

Offset = 1850h + (y * 4h); where y = 0h to 1h

Figure 32-131. OCTL_01_y Register

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

Table 32-236. OCTL_01_y Register Field Descriptions

Bit	Field	Type	Reset	Description
31-6	RESERVED	R/W	0h	
5	CCPIV	R/W	0h	CCP Initial Value This bit specifies the logical value put on the signal generator state while the counter is disabled (CTRCTL.EN == 0). 0h = Low 1h = High
4	CCPOINV	R/W	0h	CCP Output Invert The output as selected by CCPO is conditionally inverted. 0h = No inversion 1h = Invert
3-0	CCPO	R/W	0h	CCP Output Source 0h = Signal generator value (for example, PWM, triggered PWM) 1h = Load event 2h = CCU event or CCD event 4h = Zero event 5h = Capture event 6h = Fault condition 8h = Mirror CCP of first capture and compare register to other capture compare blocks 9h = Mirror CCP of second capture and compare register in other capture compare blocks Ch = Signal generator output after deadband insertion Dh = Counter direction

53 CCACT_01_y Register (Offset = 1870h + formula) [Reset = 00000000h]

CCACT_01_y is shown in Figure 32-132 and described in Table 32-130.

Return to the Summary Table.

The CCACT_01 register controls the actions of the signal generator of the capture-compare slice based on the events created in the counter block, the capture and compare block and debug events.

Offset = 1870h + (y * 4h); where y = 0h to 1h

Figure 32-132. CCACT_01_y Register

31	30	29	28	27	26	25	24
RESERVED		SWFRCACT		RESERVED			RESERVED
R/W-0h		R/W-0h		R/W-0h			R/W-0h
23	22	21	20	19	18	17	16
RESERVED		RESERVED					CC2UACT
R/W-0h		R/W-0h					R/W-0h
15	14	13	12	11	10	9	8
CC2UACT	RESERVED	CC2DACT		RESERVED	CUACT		RESERVED
R/W-0h	R/W-0h	R/W-0h		R/W-0h	R/W-0h		R/W-0h
7	6	5	4	3	2	1	0
CDACT		RESERVED	LACT		RESERVED	ZACT	
R/W-0h		R/W-0h	R/W-0h		R/W-0h	R/W-0h	

Table 32-238. CCACT_01_y Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	RESERVED	R/W	0h	Reserved
29-28	SWFRCACT	R/W	0h	CCP Output Action on Software Force Output This field describes the resulting action of software force. This action has a shadow register, which will be updated under specific condition. So that this register cannot take into effect immediately. 0h = This event is disabled and a lower priority event is selected if asserting. The CCP output value is unaffected by the event. 1h = CCP output value is set high 2h = CCP output value is set low
27-25	RESERVED	R/W	0h	Reserved
24-22	RESERVED	R/W	0h	Reserved
21-17	RESERVED	R/W	0h	
16-15	CC2UACT	R/W	0h	CCP Output Action on CC2U event. 0h = This event is disabled and a lower priority event is selected if asserting. The CCP output value is unaffected by the event. 1h = CCP output value is set high 2h = CCP output value is set low 3h = CCP output value is toggled
14	RESERVED	R/W	0h	
13-12	CC2DACT	R/W	0h	CCP Output Action on CC2D event. 0h = This event is disabled and a lower priority event is selected if asserting. The CCP output value is unaffected by the event. 1h = CCP output value is set high 2h = CCP output value is set low 3h = CCP output value is toggled
11	RESERVED	R/W	0h	

Table 32-238. CCACT_01_y Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
10-9	CUACT	R/W	0h	CCP Output Action on Compare (Up) This field describes the resulting action of the signal generator upon detecting a compare event while counting up. 0h = This event is disabled and a lower priority event is selected if asserting. The CCP output value is unaffected by the event. 1h = CCP output value is set high 2h = CCP output value is set low 3h = CCP output value is toggled
8	RESERVED	R/W	0h	
7-6	CDACT	R/W	0h	CCP Output Action on Compare (Down) This field describes the resulting action of the signal generator upon detecting a compare event while counting down. 0h = This event is disabled and a lower priority event is selected if asserting. The CCP output value is unaffected by the event. 1h = CCP output value is set high 2h = CCP output value is set low 3h = CCP output value is toggled
5	RESERVED	R/W	0h	
4-3	LACT	R/W	0h	CCP Output Action on Load Specifies what changes occur to CCP output as the result of a load event. 0h = This event is disabled and a lower priority event is selected if asserting. The CCP output value is unaffected by the event. 1h = CCP output value is set high 2h = CCP output value is set low 3h = CCP output value is toggled
2	RESERVED	R/W	0h	
1-0	ZACT	R/W	0h	CCP Output Action on Zero Specifies what changes occur to CCP output as the result of a zero event. 0h = This event is disabled and a lower priority event is selected if asserting. The CCP output value is unaffected by the event. 1h = CCP output value is set high 2h = CCP output value is set low 3h = CCP output value is toggled

54 IFCTL_01_y Register (Offset = 1880h + formula) [Reset = 00000000h]

IFCTL_01_y is shown in Figure 32-133 and described in Table 32-131.

Return to the Summary Table.

The IFCTL_01 register controls the input selection and inversion for the associated Capture-Compare slice.

Offset = 1880h + (y * 4h); where y = 0h to 1h

Figure 32-133. IFCTL_01_y Register

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

Table 32-240. IFCTL_01_y Register Field Descriptions

Bit	Field	Type	Reset	Description
31-13	RESERVED	R/W	0h	
12	FE	R/W	0h	Filter Enable This bit controls whether the input is filtered by the input filter or bypasses to the edge detect. 0h = Bypass. 1h = Filtered.
11	CPV	R/W	0h	Consecutive Period/Voting Select This bit controls whether the input filter uses a stricter consecutive period count or majority voting. 0h = Consecutive Periods The input must be at a specific logic level for the period defined by FP before it is passed to the filter output. 1h = Voting The filter ignores one clock of opposite logic over the filter period. I.e. Over FP samples of the input, up to 1 sample may be of an opposite logic value (glitch) without affecting the output.
10	RESERVED	R/W	0h	
9-8	FP	R/W	0h	Filter Period. This field specifies the sample period for the input filter. I.e. The input is sampled for FP timer clocks during filtering. 0h = The division factor is 3 1h = The division factor is 5 2h = The division factor is 8

Table 32-240. IFCTL_01_y Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
7	INV	R/W	0h	Input Inversion This bit controls whether the selected input is inverted. 0h = Noninverted 1h = Inverted
6-4	RESERVED	R/W	0h	
3-0	ISEL	R/W	0h	Input Select (CCP0) This field selects the input source to the filter input. 4h-7h = Reserved 0h = CCP of the corresponding capture compare unit 1h = Input pair CCPX of the capture compare unit. For CCP0 input pair is CCP1 and for CCP1 input pair is CCP0. 2h = CCP0 of the counter 3h = Trigger 4h = XOR of CCP inputs as input source (Used in Hall input mode). 5h = subscriber 0 event as input source. 6h = subscriber 1 event as input source. 7h = Comparator 0 output. 8h = Comparator 1 output. 9h = Comparator 2 output.

55 TSEL Register (Offset = 18B0h) [Reset = 00000000h]

TSEL is shown in Figure 32-134 and described in Table 32-132.

Return to the Summary Table.

The TSEL register controls the input trigger enable and selection of the trigger source. Trigger sources are generated by other SoC elements through their respective publisher ports (subscribed in by the timer's subscriber port).

Figure 32-134. TSEL Register

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

Table 32-242. TSEL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-10	RESERVED	R/W	0h	
9	TE	R/W	0h	Trigger Enable. This selects whether a trigger is enabled or not for this counter 0x0 = Triggers are not used 0x1 = Triggers are used as selected by the ETSEL field 0h = Triggers are not used. 1h = Triggers are used as selected by the IE, ITSEL and ETSEL fields.
8-5	RESERVED	R/W	0h	
4-0	ETSEL	R/W	0h	External Trigger Select. This selects which System Event is used if the input filter selects trigger. Triggers 0-15 are used to connect triggers generated by other timer modules. Refer to the SoC datasheet for details related to timer trigger sources. Triggers 16 and 17 are connected to event manager subscriber ports. Event lines 18-31 are reserved for future use. 0h = TRIGx = External trigger input from TIM x. 1h = TRIGx = External trigger input from TIM x. 2h = TRIGx = External trigger input from TIM x. 3h = TRIGx = External trigger input from TIM x. 4h = TRIGx = External trigger input from TIM x. 5h = TRIGx = External trigger input from TIM x. 6h = TRIGx = External trigger input from TIM x. 7h = TRIGx = External trigger input from TIM x. 8h = TRIGx = External trigger input from TIM x. 9h = TRIGx = External trigger input from TIM x. Ah = TRIGx = External trigger input from TIM x. Bh = TRIGx = External trigger input from TIM x. Ch = TRIGx = External trigger input from TIM x. Dh = TRIGx = External trigger input from TIM x. Eh = TRIGx = External trigger input from TIM x. Fh = TRIGx = External trigger input from TIM x. 10h = TRIG_SUBx = External trigger input from subscriber port x. 11h = TRIG_SUBx = External trigger input from subscriber port x.

33.1 Window Watchdog Timer (WWDT)

The window watchdog timer (WWDT) supervises code execution. If the application software does not successfully reset the window watchdog within the programmed open time window, the window watchdog generates a reset.

33.1.1 WWDT Overview

The primary function of the window watchdog timer (WWDT) is to initiate a reset when correct operation of the device has failed due to an unexpected software or system delay. The WWDT can be programmed with a predefined time window within which the application software must restart the timer, indicating that application execution is proceeding normally. If application software fails to restart the timer within the specified window, the WWDT will issue a WWDT violation signal to SYSCTL to generate a BOOTRST, which resets the peripheral and CPU state and also causes the boot configuration routine (BCR) to run.

If watchdog functionality is not required in an application, the WWDT can also be configured as a basic system interval timer which is capable of generating periodic maskable interrupts to the CPU through the WWDT CPU_INT event.

Key features of the WWDT include:

- A 25-bit counter with closed and open window spans
- Counter driven from LFCLK (fixed 32kHz clock path) with a programmable clock divider
- Eight selectable watchdog timer periods
  - Eight selectable closed window percentages to further refine the open window span
  - Two configurable closed window spans to alter open window span after WWDT has been enabled
- Optional automatic suspension of counter when operating in low power modes
- Support for standard window watchdog mode or interval timer (nonwatchdog) mode

33.1.1.1 Watchdog Mode

In watchdog mode, the WWDT is configured to count up to the specified WWDT period determined by the CLKDIV and PER bits of the WWDTCTL0 register (see WWDT Period Timing Options). The WWDT period consists of a closed window span and an open window span. The WWDT counter must be restarted with the configured open window of the WWDT period, or the WWDT will assert a WWDT violation to SYSCTL and a reset will be generated.

The window watchdog timer supports detecting both a "too late" response as well as a "too early" response through the use of the closed window, as shown in Watchdog Functionality. The closed window span begins first, followed by the open window span. The WWDT can only be restarted during the open window span. Restarting the WWDT during the open window span restarts the WWDT counter and the closed window span begins again. Any attempt to restart the WWDT during the closed window span results in a violation. Following the closed window, if the WWDT is not restarted before the end of the open window, the WWDT period expires and a WWDT violation is generated.

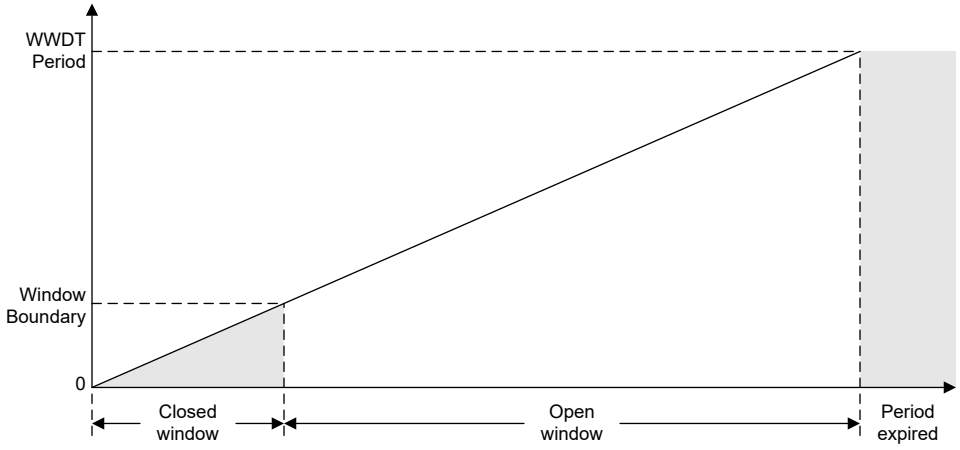


Figure 33-1. WWDT Functionality

If the closed window functionality is not desired, it can be disabled by setting the closed window span to 0%, giving traditional watchdog timer functionality where the WWDT can be reset any time before the WWDT period expires.

33.1.1.2 Interval Timer Mode

The WWDT can be used in interval timer mode to generate periodic interrupts to the CPU when not using the watchdog functionality. When used in interval timer mode, a WWDT interrupt is generated when the WWDT period expires, or when an incorrect password is applied to the WWDT control registers (WWDTCTL0 and WWDTCTL1). The WWDT counter restarts itself once the WWDT period expires. A write of the RESTART value to the WWDTCNTRST register will restart the WWDT interval timer mode, keeping the period from expiring and a WWDT interrupt from generating.

33.1.2 WWDT Operation

The WWDT must be enabled before being configured for use through the PWREN register by writing the ENABLE bit with the correct KEY value to the PWREN register (see peripheral power enable).

The WWDT is configured through the WWDTCTL0 and WWDTCTL1 registers. These registers are password protected. Any register access (read or write) must be a 32-bit access. Write access must also include the corresponding password in the most significant byte (0xC9 for WWDTCTL0 and 0xBE for WWDTCTL1). Attempting a register write without the correct password or attempting a write with an access other than a 32-bit access generates a WWDT violation to SYSCTL, generating a BOOTRST. The password byte always reads as 0x00.

The WWDT is disabled and cleared after a SYSRST, which resets the peripherals and CPU state but does not trigger execution of the BCR, or a BOOTRST. A WWDT violation does not generate a SYSRST. Please see Resets and Device Initialization for more information on resets.

The WWDTCTL0 register sets the static configuration of the WWDT, including: the clock divider (CLKDIV), the timer period (PER), the two closed window percentages (WINDOW0 and WINDOW1), the timer mode (MODE), and the stop-in-sleep status (STISM). The WWDTCTL1 register controls which closed window percentage is being used by the WWDT through the WINSEL bit.

The first write (with a key match) to the WWDTCTL0 register enables the WWDT. Once the WWDT is enabled, the WWDTCTL0 register becomes write protected. Any attempt to write to the WWDTCTL0 register after the WWDT is enabled generates a WWDT violation to SYSCTL. In order to reconfigure the WWDT after it is enabled, the device must undergo either SYSRST or BOOTRST. The RUN bit in the WWDTSTAT register indicates that the WWDT is running.

WWDT Diagram shows the WWDT functional block diagram.

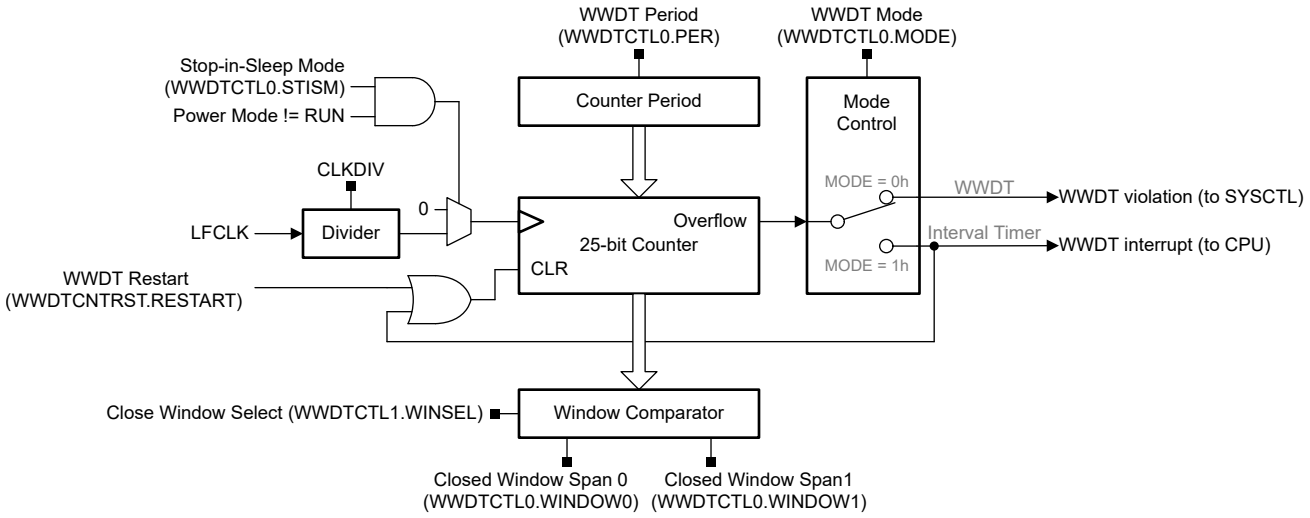


Figure 33-2. WWDT Diagram

33.1.2.1 Mode Selection

The WWDT operating mode (watchdog mode or interval timer mode) is selected by the MODE bit in the WWDTCTL0 register. Watchdog mode is the default mode (MODE cleared). Setting the MODE bit configures the WWDT for interval mode.

When the WWDT is in watchdog mode, the WWDT counter must be restarted within the open window period by writing the RESTART value (0x000000A7) to the WWDTCNTRST register. After a reset or restart, the WWDT counter will restart from zero. A failure to restart the WWDT within the open window or an attempt to restart the WWDT counter during the closed window will generate a WWDT violation to SYSCTL. Writing any value other than the RESTART value to the WWDTCNTRST register also generates a WWDT violation.

When the WWDT is in interval mode, the timer acts as an interval timer, generating WWDT interrupts to the CPU as specified by the WWDT period. As soon as the WWDT is enabled in interval mode, the WWDT interval timer interrupt will be asserted after the expiration of the timer. It is not necessary to restart the WWDT in interval timer mode.

33.1.2.2 Clock Configuration

The WWDT runs from the 32kHz low-frequency clock (LFCLK). A clock divider supports dividing the input clock from /1 (no divide) to /8 (divide-by-8) using the CLKDIV field in the WWDTCTL0 register. The default CLKDIV setting is 0x03 (32kHz divided by 4, or 8kHz).

By running from the LFCLK, the WWDT time base is independent of the main clock (MCLK) and CPU clock (CPUCLK) time base, provided that these clocks are not also configured to be running from the LFCLK. While the time base may be considered as independent and derived from a separate oscillator source, LFCLK edges are synchronized to the MCLK before sourcing the WWDT to enable simple access to the memory-mapped registers from application software. A simplified view of the clock scheme is given in WWDT Simplified Clock Source Tree. In WWDT Simplified Clock Source Tree, the internal LFOSC is configured to set the LFCLK rate, and the internal SYSOSC sets the MCLK/CPUCLK rate. Clock selection muxes and dividers are excluded from the figure to simplify the view; the complete clock tree is provided in Clock Tree.

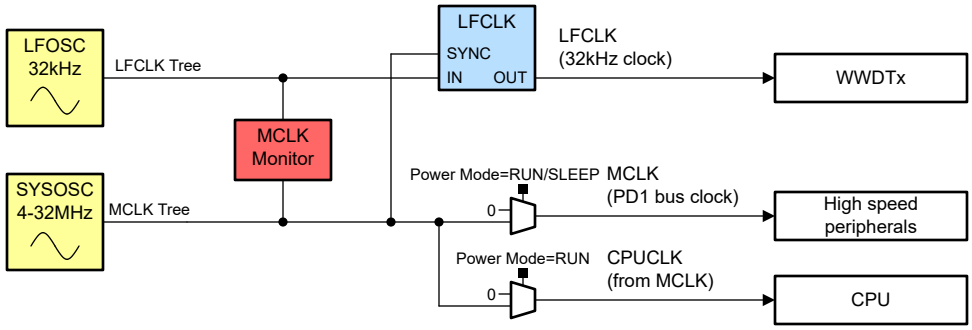


Figure 33-3. WWDT Simplified Clock Source Tree

In the event that the MCLK fails and synchronization of the LFCLK to the MCLK is lost, this failure may be detected by enabling the continuous MCLK monitor. When the MCLK monitor is enabled, a loss of MCLK is always a catastrophic failure which generates a BOOTRST within 12 LFCLK cycles. As a result, a loss of the MCLK tree, and corresponding loss of synchronization, does not prevent a BOOTRST from being generated.

Period Selection

The WWDT has a 25-bit counter which is initially stopped after a SYSRST. The WWDT period (total time interval) is set by the PER field in the WWDTCTL0 register. The total WWDT period is calculated as follows:

$$T_{WWDT} = (CLKDIV + 1) * PER_{COUNT} / 32768Hz \quad (41)$$

The total timer count PER_{COUNT} is selected to be one of 8 possible period count values, with the encoding given in WWDT Period Total Timer Count.

Table 33-1. WWDT Period Total Timer Count

PER	0x0	0x1	0x2	0x3	0x4	0x5	0x6	0x7
PER_{COUNT}	2^{25}	2^{21}	2^{18}	2^{15}	2^{12}	2^{10}	2^{8}	2^{6}

The combination of the period selection (PER) and clock divider (CLKDIV) enable a wide range of WWDT periods, from as short as 1.95ms to as long as 136.53 minutes. WWDT Period Timing Options gives all possible WWDT periods for a given combination of PER and CLKDIV.

Table 33-2. WWDT Period Timing Options

CLKDIV	PER							
	0x0	0x1	0x2	0x3	0x4	0x5	0x6	0x7
	min	*min*	*s*	*s*	*s*	*ms*	*ms*	*ms*
0x0 (/1)	17.07	1.07	8.00	1.00	0.13	31.25	7.81	1.95
0x1 (/2)	34.13	2.13	16.00	2.00	0.25	62.50	15.63	3.91
0x2 (/3)	51.20	3.20	24.00	3.00	0.38	93.75	23.44	5.86
0x3 (/4)	68.27	4.27	32.00	4.00	0.50	125.00	32.25	7.81
0x4 (/5)	85.33	5.33	40.00	5.00	0.63	156.25	39.06	9.77
0x5 (/6)	102.40	6.40	48.00	6.00	0.75	187.50	46.88	11.72
0x6 (/7)	119.47	7.47	56.00	7.00	0.88	218.75	54.69	13.67
0x7 (/8)	136.53	8.53	64.00	8.00	1.00	250.00	62.50	15.63

The default setting for the WWDT upon reset is 0.5s (CLKDIV = 0x3 and PER = 0x4).

ADVANCE INFORMATION

Synchronization Delay

When starting or re-starting the WWDT counter, a maximum synchronization delay of one 32kHz clock cycle (30.5µs) can occur before the WWDT counter begins counting from zero. The periods given in WWDT Period Timing Options do not include this synchronization delay.

Closed Window Selection

Once the WWDT is enabled, the configuration of the WWDT cannot be changed except for the capability to switch the active closed window selection between two closed window spans. Configuration of two closed window spans is supported by setting the WINDOW0 and WINDOW1 fields in the WWDTCTL0 register. The WINSEL bit in the WWDTCTL1 register determines the active window (either WINDOW0 or WINDOW1). Either window can be set to one of 8 possible window settings.

Table 33-3. WWDT Window Options

WINDOW	0x0	0x1	0x2	0x3	0x4	0x5	0x6	0x7
Closed window	0%	12.5%	18.75%	25%	50%	75%	81.25%	87.5%

Setting a WINDOWx value to 0x0 (0% closed, 100% open) is equivalent to disabling the window function of the WWDT. In this configuration, the WWDT can be restarted at any point during the WWDT period.

By configuring both WINDOW0 and WINDOW1 within the WWDTCTL0 register, the active window selection can be changed after the WWDT has been enabled by changing the WINSEL bit. When the WWDT is restarted by writing the RESTART value to the WWDTCNTRST register, the closed window selection (WINSEL) must not be changed for at least four 32kHz clock cycles (≈122µs).

33.1.2.3 Low-Power Mode Behavior

The WWDT counter can be configured to continue counting when the device is in a low-power mode (CPU is disabled) or to continue to run when the device is in a low-power mode.

The STISM bit in the WWDTCTL0 register controls if the WWDT counter stops counting in sleep mode. By default, the STISM bit is cleared, indicating that the WWDT continues to count in low-power modes. To stop the WWDT from counting in low-power modes, set the STISM bit when loading the WWDTCTL0 configuration to start the WWDT. In this case, when the low-power mode is exited and the CPU returns to operation, the WWDT counter resumes counting from the same value it held before entering the low-power mode.

33.1.2.4 Debug Behavior

The WWDT can be configured to stop counting or continue counting when the CPU is halted for debug by the debug subsystem. By default, the WWDT stops counting when the CPU is halted for debug and the device is in a debug state. To allow the WWDT to continue to free run when the CPU is stopped for debug, set the FREE bit in the PDBGCTL register.

33.1.2.5 WWDT Events

The WWDT module uses the CPU_INT event registers to manage WWDT interrupt requests (IRQs) to the CPU subsystem through a static event route.

33.1.2.5.1 CPU Interrupt Event (CPU_INT)

The WWDT module provides one interrupt source which can be configured to source a CPU interrupt event. The WWDT interrupt conditions are given in WWDT CPU Interrupt Conditions (CPU_INT).

Table 33-4. WWDT CPU Interrupt Conditions (CPU_INT)

Index (IIDX)	Name	Description
0	INTTIM	Indicates that the WWDT interval timer period has expired

The CPU interrupt event configuration is managed with the CPU_INT event management registers. See *Using Event Registers* for guidance on configuring the Event registers for CPU interrupts.

33.2 WWDT Registers

This Section describes the WWDT Registers.

33.2.1 WWDT Base Address Table

Table 33-5. WWDT Base Address Table

Bit Field Name		DriverLib Name	Base Address
Instance	Structure		
WwdtRegs	WWDT_REGS	WWDT	0x400D_0000

33.2.2 WWDT_REGS Registers

Table 33-6 lists the memory-mapped registers for the WWDT_REGS registers. All register offset addresses not listed in Table 33-6 should be considered as reserved locations and the register contents should not be modified.

Table 33-6. WWDT_REGS Registers

Offset	Acronym	Register Name	Section
480h	CPU_CONNECT_0	CPU Connect	Go
800h	PWREN	Power enable	Go
804h	RSTCTL	Reset Control	Go
814h	STAT	Status Register	Go
1018h	PDBGCTL	Peripheral Debug Control	Go
1020h	IIDX	Interrupt index	Go
1028h	IMASK	Interrupt mask	Go
1030h	RIS	Raw interrupt status	Go
1038h	MIS	Masked interrupt status	Go
1040h	ISET	Interrupt set	Go
1048h	ICLR	Interrupt clear	Go
10E0h	EVT_MODE	Event Mode	Go
10FCh	DESC	Module Description	Go
1100h	WWDTCTL0	Window Watchdog Timer Control Register 0	Go
1104h	WWDTCTL1	Window Watchdog Timer Control Register 0	Go
1108h	WWDTCNTRST	Window Watchdog Timer Counter Reset Register	Go
110Ch	WWDTSTAT	Window Watchdog Timer Status Register	Go

Complex bit access types are encoded to fit into small table cells. Table 33-7 shows the codes that are used for access types in this section.

Table 33-7. WWDT_REGS Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
Write Type		
W	W	Write
WK	W K	Write Write protected by a key
Reset or Default Value		
-n		Value after reset or the default value

1 CPU_CONNECT_0 Register (Offset = 480h) [Reset = 00000000h]

CPU_CONNECT_0 is shown in Figure 33-4 and described in Table 33-8.

Return to the Summary Table.

Connect peripheral interrupts / publisher port (FPUB_1) to application processor

Figure 33-4. CPU_CONNECT_0 Register

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

Table 33-9. CPU_CONNECT_0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R/W	0h	
1	RESERVED	R/W	0h	Reserved
0	RESERVED	R/W	0h	

2 PWREN Register (Offset = 800h) [Reset = 00000000h]

PWREN is shown in Figure 33-5 and described in Table 33-9.

Return to the Summary Table.

Register to control the power state

Figure 33-5. PWREN Register

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

Table 33-11. PWREN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	KEY	W	0h	KEY to allow Power State Change 26h = KEY to allow write access to this register
23-1	RESERVED	R/W	0h	
0	ENABLE	R/WK	0h	Enable the power Note: For safety devices the power cannot be disabled once enabled. [EXT_GPRCM.GPRCM.PWREN.KEY] must be set to 26h to write to this bit. 0h = Disable Power 1h = Enable Power

ADVANCE INFORMATION

3 RSTCTL Register (Offset = 804h) [Reset = 00000000h]

RSTCTL is shown in Figure 33-6 and described in Table 33-10.

Return to the Summary Table.

Register to control reset assertion and de-assertion

Figure 33-6. RSTCTL Register

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

Table 33-13. RSTCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	KEY	W	0h	Unlock key B1h = KEY to allow write access to this register
23-2	RESERVED	W	0h	
1	RESETSTKYCLR	WK	0h	Clear [GPRCM.STAT.RESETSTKY] [EXT_GPRCM.GPRCM.RSTCTL.KEY] must be set to B1h to write to this bit. 0h = Writing 0 has no effect 1h = Clear reset sticky bit
0	RESETASSERT	WK	0h	Assert reset to the peripheral Note: For safety devices a watchdog reset by software is not possible. [EXT_GPRCM.GPRCM.RSTCTL.KEY] must be set to B1h to write to this bit. 0h = Writing 0 has no effect 1h = Assert reset

ADVANCE INFORMATION

4 STAT Register (Offset = 814h) [Reset = 00000000h]

STAT is shown in Figure 33-7 and described in Table 33-11.

Return to the Summary Table.

peripheral enable and reset status

Figure 33-7. STAT Register

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

Table 33-15. STAT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-17	RESERVED	R	0h	
16	RESETSTKY	R	0h	This bit indicates, if the peripheral was reset, since this bit was cleared by RESETSTKYCLR in the RSTCTL register 0h = The peripheral has not been reset since this bit was last cleared by RESETSTKYCLR in the RSTCTL register 1h = The peripheral was reset since the last bit clear
15-0	RESERVED	R	0h	

5 PDBGCTL Register (Offset = 1018h) [Reset = 00000000h]

PDBGCTL is shown in Figure 33-8 and described in Table 33-12.

Return to the Summary Table.

This register can be used by the software developer to control the behavior of the peripheral relative to the 'Core Halted' input

Figure 33-8. PDBGCTL Register

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

Table 33-17. PDBGCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R/W	0h	
0	FREE	R/W	0h	Free run control 0h = The peripheral freezes functionality while the Core Halted input is asserted and resumes when it is deasserted. 1h = The peripheral ignores the state of the Core Halted input

ADVANCE INFORMATION

6 IIDX Register (Offset = 1020h) [Reset = 00000000h]

IIDX is shown in Figure 33-9 and described in Table 33-13.

Return to the Summary Table.

This register provides the highest priority enabled interrupt index.

Figure 33-9. IIDX Register

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

Table 33-19. IIDX Register Field Descriptions

Bit	Field	Type	Reset	Description
31-5	RESERVED	R	0h	
4-0	STAT	R	0h	Module Interrupt Vector Value. This register provides the highest priority interrupt index. A read clears the corresponding interrupt flag in RIS and MISC. 0h = No interrupt pending 1h = Interval Timer Interrupt Interrupt Flag: INTTIM Interrupt Priority: Highest

7 IMASK Register (Offset = 1028h) [Reset = 00000000h]

IMASK is shown in Figure 33-10 and described in Table 33-14.

Return to the Summary Table.

Interrupt Mask. If a bit is set, then corresponding interrupt is un-masked. Un-masking the interrupt causes the raw interrupt to be visible in IIDX, as well as MIS."

Figure 33-10. IMASK Register

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

Table 33-21. IMASK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R/W	0h	
0	INTTIM	R/W	0h	Interval Timer Interrupt. 0h = Clear Interrupt Mask 1h = Set Interrupt Mask

8 RIS Register (Offset = 1030h) [Reset = 00000000h]

RIS is shown in Figure 33-11 and described in Table 33-15.

Return to the Summary Table.

Raw interrupt status. Reflects all pending interrupts, regardless of masking. The RIS register allows the user to implement a poll scheme. A flag set in this register can be cleared by writing 1 to the ICLR register bit even if the corresponding IMASK bit is not enabled.

Figure 33-11. RIS Register

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

Table 33-23. RIS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	
0	INTTIM	R	0h	Interval Timer Interrupt. 0h = Interrupt did not occur 1h = Interrupt occurred

9 MIS Register (Offset = 1038h) [Reset = 00000000h]

MIS is shown in Figure 33-12 and described in Table 33-16.

Return to the Summary Table.

Masked interrupt status. This is an AND of the IMASK and RIS registers.

Figure 33-12. MIS Register

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

Table 33-25. MIS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	
0	INTTIM	R	0h	Interval Timer Interrupt. 0h = Interrupt did not occur 1h = Interrupt occurred

10 ISET Register (Offset = 1040h) [Reset = 00000000h]

ISET is shown in Figure 33-13 and described in Table 33-17.

Return to the Summary Table.

Interrupt set. Allows interrupts to be set by software (useful in diagnostics and safety checks). Writing a 1 to a bit in ISET will set the event and therefore the related RIS bit also gets set. If the interrupt is enabled through the mask, then the corresponding MIS bit is also set.

Figure 33-13. ISET Register

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

Table 33-27. ISET Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	W	0h	
0	INTTIM	W	0h	Interval Timer Interrupt. 0h = Writing 0 has no effect 1h = Set Interrupt

11 ICLR Register (Offset = 1048h) [Reset = 00000000h]

ICLR is shown in Figure 33-14 and described in Table 33-18.

Return to the Summary Table.

Interrupt clear. Write a 1 to clear corresponding Interrupt.

Figure 33-14. ICLR Register

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

Table 33-29. ICLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	W	0h	
0	INTTIM	W	0h	Interval Timer Interrupt. 0h = Writing 0 has no effect 1h = Clear Interrupt

ADVANCE INFORMATION

12 EVT_MODE Register (Offset = 10E0h) [Reset = 00000000h]

EVT_MODE is shown in Figure 33-15 and described in Table 33-19.

Return to the Summary Table.

Event mode register. It is used to select whether each line is disabled, in software mode (software clears the RIS) or in hardware mode (hardware clears the RIS)

Figure 33-15. EVT_MODE Register

31	30	29	28	27	26	25	24
RESERVED							
R/W-0h							
23	22	21	20	19	18	17	16
RESERVED							
R/W-0h							
15	14	13	12	11	10	9	8
RESERVED							
R/W-0h							
7	6	5	4	3	2	1	0
RESERVED						INT0_CFG	
R/W-0h						R-0h	

Table 33-31. EVT_MODE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R/W	0h	
1-0	INT0_CFG	R	1h	Event line mode select for event corresponding to [IPSTANDARD.CPU_INT] 0h = The interrupt or event line is disabled. 1h = The interrupt or event line is in software mode. Software must clear the RIS. 2h = The interrupt or event line is in hardware mode. The hardware (another module) clears automatically the associated RIS flag.

13 DESC Register (Offset = 10FCh) [Reset = 00000000h]

DESC is shown in Figure 33-16 and described in Table 33-20.

Return to the Summary Table.

This register identifies the peripheral and its exact version.

Figure 33-16. DESC Register

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

Table 33-33. DESC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	MODULEID	R	1F11h	Module identification contains a unique peripheral identification number. The assignments are maintained in a central database for all of the platform modules to ensure uniqueness. 0h = Smallest value FFFFh = Highest possible value
15-12	FEATUREVER	R	7h	Feature Set for the module *instance* 0h = Smallest value Fh = Highest possible value
11-8	INSTNUM	R	0h	Instance Number within the device. This will be a parameter to the RTL for modules that can have multiple instances 0h = Smallest value Fh = Highest possible value
7-4	MAJREV	R	1h	Major rev of the IP 0h = Smallest value Fh = Highest possible value
3-0	MINREV	R	0h	Minor rev of the IP 0h = Smallest value Fh = Highest possible value

14 WWDTCTL0 Register (Offset = 1100h) [Reset = 00000043h]

WWDTCTL0 is shown in Figure 33-17 and described in Table 33-21.

Return to the Summary Table.

Window Watchdog Timer Control 0 Register
NOTE: Write to this register is enabled after System Reset. The first successful write (key match) enables the Watchdog. When the watchdog is enabled all subsequent writes to this register activate the WWDT error signal to the ESM.

Figure 33-17. WWDTCTL0 Register

31	30	29	28	27	26	25	24
KEY							
W-0h							
23	22	21	20	19	18	17	16
RESERVED						STISM	MODE
R/W-0h						R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
RESERVED	WINDOW1			RESERVED	WINDOW0		
R/W-0h	R/W-0h			R/W-0h	R/W-0h		
7	6	5	4	3	2	1	0
RESERVED	PER			RESERVED	CLKDIV		
R/W-0h	R/W-4h			R/W-0h	R/W-3h		

Table 33-35. WWDTCTL0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	KEY	W	0h	KEY to allow write access to this register. Writing to this register with an incorrect key activates the WWDT error signal to the ESM. Read as 0. C9h (W) = KEY to allow write access to this register
23-18	RESERVED	R/W	0h	
17	STISM	R/W	0h	Stop In Sleep Mode. The functionality of this bit requires that POLICY.HWCEN = 0. If POLICY.HWCEN = 1 the WWDT resets during sleep and needs re-configuration. Note: This bit has no effect for the global Window Watchdog as Sleep Mode is not supported. 0h = The WWDT continues to function in Sleep mode. 1h = The WWDT stops in Sleep mode and resumes where it was stopped after wakeup.
16	MODE	R/W	0h	Window Watchdog Timer Mode 0h = Window Watchdog Timer Mode. The WWDT will generate a error signal to the ESM when following conditions occur: - Timer Expiration (Timeout) - Reset WWDT during the active window closed period - Keyword violation 1h = Interval Timer Mode. The WWDT acts as an interval timer. It generates an interrupt on timeout.
15	RESERVED	R/W	0h	

Table 33-35. WWDTCTL0 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
14-12	WINDOW1	R/W	0h	Closed window period in percentage of the timer interval. WWDTCTL1.WINSEL determines the active window setting (WWDTCTL0.WINDOW0 or WWDTCTL0.WINDOW1). 0h = 0% (No closed Window) 1h = 12.50% of the total timer period is closed window 2h = 18.75% of the total timer period is closed window 3h = 25% of the total timer period is closed window 4h = 50% of the total timer period is closed window 5h = 75% of the total timer period is closed window 6h = 81.25% of the total timer period is closed window 7h = 87.50% of the total timer period is closed window
11	RESERVED	R/W	0h	
10-8	WINDOW0	R/W	0h	Closed window period in percentage of the timer interval. WWDTCTL1.WINSEL determines the active window setting (WWDTCTL0.WINDOW0 or WWDTCTL0.WINDOW1). 0h = 0% (No closed Window) 1h = 12.50% of the total timer period is closed window 2h = 18.75% of the total timer period is closed window 3h = 25% of the total timer period is closed window 4h = 50% of the total timer period is closed window 5h = 75% of the total timer period is closed window 6h = 81.25% of the total timer period is closed window 7h = 87.50% of the total timer period is closed window
7	RESERVED	R/W	0h	
6-4	PER	R/W	4h	Timer Period of the WWDT. These bits select the total watchdog timer count. 0h = Total timer count is 2^{25} 1h = Total timer count is 2^{21} 2h = Total timer count is 2^{18} 3h = Total timer count is 2^{15} 4h = Total timer count is 2^{12} (default) 5h = Total timer count is 2^{10} 6h = Total timer count is 2^{8} 7h = Total timer count is 2^{6}
3	RESERVED	R/W	0h	
2-0	CLKDIV	R/W	3h	Module Clock Divider, Divide the clock source by CLKDIV+1. Divider values from /1 to /8 are possible. The clock divider is currently 4 bits. Bit 4 has no effect and should always be written with 0. 0h = Minimum value 7h = Maximum value

15 WWDTCTL1 Register (Offset = 1104h) [Reset = 00000000h]

WWDTCTL1 is shown in Figure 33-18 and described in Table 33-22.

Return to the Summary Table.

Window Watchdog Timer Control 1 Register

Figure 33-18. WWDTCTL1 Register

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

Table 33-37. WWDTCTL1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	KEY	W	0h	KEY to allow write access to this register. Writing to this register with an incorrect key activates the WWDT error signal to the ESM. Read as 0. BEh (W) = KEY to allow write access to this register
23-1	RESERVED	R/W	0h	
0	WINSEL	R/W	0h	Close Window Select 0h = In window mode field WINDOW0 of WDDTCTL0 defines the closed window size. 1h = In window mode field WINDOW1 of WDDTCTL0 defines the closed window size.

ADVANCE INFORMATION

16 WWDTCNTRST Register (Offset = 1108h) [Reset = 00000000h]

WWDTCNTRST is shown in Figure 33-19 and described in Table 33-23.

Return to the Summary Table.

Window Watchdog Timer Counter Restart Register

Figure 33-19. WWDTCNTRST Register

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

Table 33-39. WWDTCNTRST Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RESTART	R/W	0h	Window Watchdog Timer Counter Restart Writing 00A7h to this register restarts the WWDT Counter. Writing any other value causes an error generation to the ESM. Read as 0. 0h = Minimum value FFFFFFFFh = Maximum value

17 WWDTSTAT Register (Offset = 110Ch) [Reset = 00000000h]

WWDTSTAT is shown in Figure 33-20 and described in Table 33-24.

Return to the Summary Table.

Window Watchdog Timer Status Register
A write to this register has no effect.

Figure 33-20. WWDTSTAT Register

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

Table 33-41. WWDTSTAT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	
0	RUN	R	0h	Watchdog running status flag. 0h = Watchdog counter stopped. 1h = Watchdog running.

Chapter 34

Debug Subsystem (DEBUGSS)

34.1 Debug Subsystem

The Debug Subsystem (DEBUGSS) is implemented in all AM13E230x devices. The DEBUGSS enables comprehensive debug of application software running on the processor during development by interfacing an external debug probe to the device systems through a Serial Wire Debug (SWD) and JTAG interface.

ADVANCE INFORMATION

34.1.1 DEBUGSS Overview

The Debug Subsystem (DEBUGSS) interfaces the ARM Serial Wire Debug (SWD) two-wire or JTAG four-wire physical interface to multiple debug functions within the device. AM13E230x devices support debugging of processor execution, the device state, and the power state (through EnergyTrace technology). The DEBUGSS also provides a mailbox system for communicating with software through SWD/JTAG.

Key features provided by the debug subsystem include:

- The ARM Serial Wire Debug (SWD) two-wire (SWDIO, SWCLK) debug interface, compatible with both TI and 3rd party debug probes
  - On-chip pullup and pulldown resistors for SWDIO and SWCLK, respectively, enabled by default
  - Support for disabling SWD functions to use SWD pins as GPIOs
  - Support for debug on all low power modes
- The JTAG (IEEE Standard 1149.1-1990 Standard Test Access Port and Boundary Scan Architecture) four-wire (TMS, TCK, TDO, TDI) debug interface, compatible with both TI and 3rd party debug probes
  - On-chip pullup and pulldown resistors for TMS, TCK, TDI, TDO, enabled by default
  - Support for disabling JTAG functions to use JTAG pins as GPIOs
  - Support for debug on all low power modes
- Debug of the processor
  - Run, halt, and step debug support
  - Up to 8 hardware breakpoints (BPU)
  - Up to 4 hardware watchpoints (DWT)
  - Instruction trace of up to 4 branches through the ARM Micro Trace Buffer (MTB)
  - Non-intrusive program-flow trace through the ARM Embedded Trace Macrocell (ETM)
  - Software breakpoints supported
- Software-configurable peripheral behavior during processor debug
  - Ability to free run select peripherals through debug halt
  - Ability to halt select peripherals on a debug halt
  - Ability to request reset and mode changes to the PMCU
- Monitoring of CPU state through EnergyTrace technology
- Mailbox (DSSM) for passing data and control signals between the SWD interface and Boot ROM (as well as application software)
- Support for various security features, including SWD/JTAG lockout and password authenticated debugging

34.1.1.1 Debug Interconnect

The DEBUGSS architecture is given in Figure 34-1.

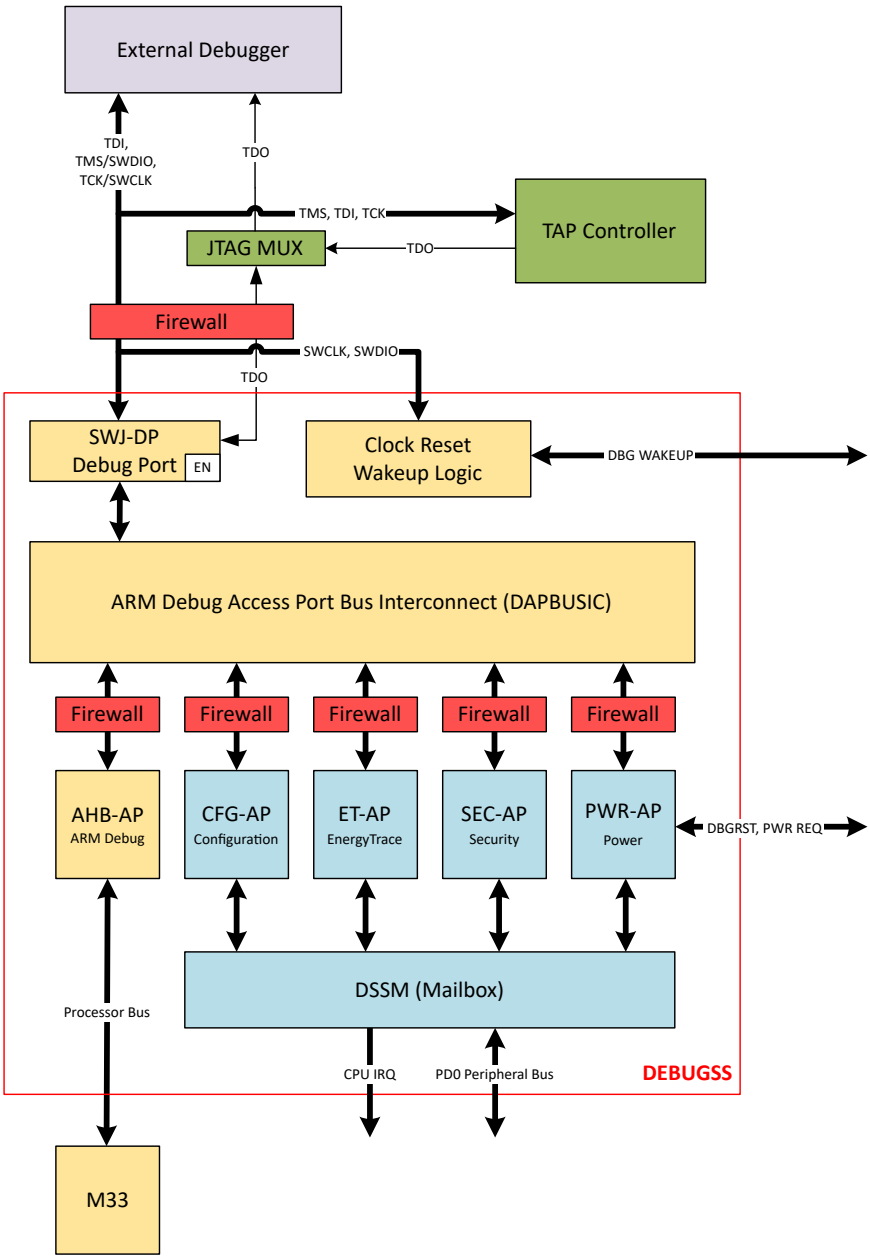


Figure 34-1. Debug Subsystem Block Diagram

The debug probe's physical interface interacts with the Arm Serial Wire Debug and JTAG Debug Port (SWJ-DP) to gain access to the debug access port bus interconnect (DAPBUSIC) when the SWJ-DP is enabled. The

SWJ-DP is a combined JTAG-DP and SW-DP, enabling access to either a SWD or JTAG probe to a target. From TI, devices ship with the SWJ-DP enabled to allow debug access to the device for development and production programming, but the SWJ-DP can be configured to be permanently disabled through the boot configuration policy (see Section 34.1.2.3).

The DAPBUSIC enables a debug probe to access one or more debug Access Ports (APs). For a debug probe to be able to communicate with an access port, the SWJ-DP debug port must not be disabled by the device boot configuration policy, and the target access port must also not be disabled by the boot configuration policy. The available access ports are given in Section 34.1.1.3 .

The SWD/JTAG and SWJ-DP also contain signaling to the PMCU module to support debug-generated resets and operating mode changes (see Section 34.1.2.2).

34.1.1.2 Physical Interfaces

The debug port is the physical interface to access the DEBUGSS. This interface controls any access ports provided within the DEBUGSS. The DEBUGSS supports a combined debug port (SWJ-DP) which includes both JTAG and Serial Wire Debug (SWD), with a mechanism that supports switching between them:

1. The JTAG-DP is based on the IEEE 1149.1 Test Access Port (TAP) and Boundary Scan Architecture, widely referred to as JTAG, and provides a JTAG interface to the DEBUGSS. For more information, see JTAG-DP
2. The SW-DP provides a two-pin (clock + data) interface to the DEBUGSS. For more information, see SW-DP

The SWJ-DP provides the auto-detect logic that selects between JTAG and SWD. This enables the JTAG-DP and SW-DP to share the same pins. For more information, see SWJ-DP.

Note

Only one debug port can be used at once. Switching between the two debug ports must only be performed when neither debug port is in use.

34.1.1.2.1 JTAG Debug Port (JTAG-DP)

By default on POR, debug connections are routed to a JTAG compliant interface known as JTAG-DP. The JTAG-DP interface requires four connections:

- Test Data In (TDI) input to target device which provides a stream of serial data from debug probe
  - TDI must be pulled high on the target to keep the signal inactive when no debug probe is connected
- Test Mode Select (TMS) input to target device which controls the JTAG state-machine
  - TMS must be pulled high on the target to keep the signal inactive when no debug probe is connected
- Test Clock (TCK) input to target device which synchronizes the JTAG state-machine on each rising edge of the TCK signal, sampling the TDI and TMS signals
  - TCK can be pulled high on the target, however, to maintain full compatibility with other JTAG equipment, Arm recommends TCK is pulled low
- Test Data Out (TDO) output from target device which returns a stream of serial data to the debug unit
  - TDO can be left floating on the target, however, to maintain full compatibility with other JTAG equipment, Arm recommends TDO is pulled high

The JTAG interface uses the standard logic levels of the device for JTAG communication. See the device-specific data sheet for input and output logic levels for a given supply voltage (VDD). A TCK frequency of up to 40MHz is supported by the DEBUGSS.

During JTAG operation, the TMS/TDI/TCK line can be driven high or low by the debug probe while the TDO line can be driven high or low by the target device. The primary purpose of the pullup resistors on the TMS/TDI/TDO pins and the pulldown on TCK is to put the signals into a known state when no debug probe is attached.
A minimum resistance of 100kΩ is recommended by Arm. The internal pullup/pulldown resistors fulfill this requirement and external resistors are not required for correct operation of the JTAG interface.

After a power-on reset (POR), AM13E230x devices configure the JTAG pins in JTAG mode with an internal pullup resistor enabled on the TMS/TDI/TDO lines and internal pullup enabled on the TCK line. If the device

configuration has not permanently disabled all JTAG access, then the JTAG interface is enabled during the boot process and a debug probe can be connected to the DEBUGSS.

34.1.1.2.2 Serial Wire Debug (SWD) Debug Port (SW-DP)

Debug connections to the device are also supported through an Arm Serial Wire Debug (SWD) compliant interface known as the SW-DP. The SW-DP interface requires two connections:

- A bidirectional data line (SWDIO) used to send data to, and receive data from, the device
- A unidirectional clock line (SWCLK) driven by the debug probe connecting to the device

The SWD interface uses the standard logic levels of the device for SWD communication. See the device-specific data sheet for input and output logic levels for a given supply voltage (VDD). A SWCLK frequency of up to 40MHz is supported by the DEBUGSS.

During SWD operation, the SWDIO line can be driven high or driven low by either the target device or the debug probe. As either device can drive the line, when ownership of the shared SWDIO line is switched between the device and the debug probe, undriven time slots are inserted as a part of the SWD protocol. The primary purpose of the pullup resistor on the SWDIO line, and the pulldown resistor on the SWCLK line, is to place the SWD pins into a known state when no debug probe is attached. A minimum resistance of 100kΩ is recommended by Arm. The internal pullup/pulldown resistors fulfill this requirement and external resistors are not required for correct operation of the SWD interface.

After a power-on reset (POR), AM13E230x devices configure the SWD pins in SWD mode with an internal pullup resistor enabled on the SWDIO line and an internal pulldown resistor enabled on the SWCLK line. If the device configuration has not permanently disabled all SWD access, then the SWD interface is enabled during the boot process and a debug probe can be connected to the DEBUGSS.

Upon physical connection of a debug probe, a valid JTAG-to-SWD sequence must be sent from the debug probe to the SWJ-DP on SWDIOTMS to initiate a SWD connection with the SW-DP. Once the sequence is applied and the SWD connection is established, communication with enabled debug access points is possible.

34.1.1.2.3 Serial Wire Debug and JTAG Debug Port (SWJ-DP)

The SWJ-DP is a combined JTAG-DP and SW-DP that interfaces directly with a SWD or JTAG probe to connect to a target by enabling access to the SW-DP or JTAG-DP blocks. The SWJ-DP interface requires four connections:

1. TDO: used for JTAG mode only
2. TDI: used for JTAG mode only
3. SWDIOTMS: used as SWDIO for SWD mode and TMS for JTAG mode
4. SWCLKTCK: used as clock signal for both SWD and JTAG modes

To make efficient use of package pins, serial wire shares, and overlays, the JTAG pins use an autodetect mechanism that switches between JTAG-DP and SW-DP depending on which probe is connected. By default on POR, the JTAG-DP is used to enable JTAG debug access. A special sequence on the SWDIOTMS pin is used to switch between JTAG-DP and SW-DP. When the switching sequence has been transmitted to the SWJ-DP, the interface behaves as a dedicated JTAG-DP or SW-DP depending upon which sequence had been performed.

34.1.1.2.4 Debug Wake Up and Interrupts

In the event that a device was configured by software to enter SHUTDOWN mode, and a debug probe is then connected to the SWJ-DP pins with SWCLKTCK active, wakeup logic triggers an exit from SHUTDOWN mode and cause a BOR. A debug connection can then be established to the DEBUGSS after the BOR completes. Please refer to Section 34.1.2.2 for additional details.

For secure applications, the software can be made aware of a debug connection by utilizing the DEBUGSS *PWRUPIFG* interrupt. When the debug probe is disconnected and the connection is lost, the *PWRDWNIFG* interrupt is asserted.

Application software can disable the SWJ-DP interface in SYSCTL, freeing the IO to be used for GPIO functionality. Review Section 3.5.1.4 in SYSCTL for using the debug pins for additional functionality other than JTAG/SWD. Once software disables debug functionality, it is not possible to re-enable it other than by triggering a POR. A POR will automatically re-enable the debug functionality and put the debug pins into JTAG/SWD mode with internal pullup/pulldown resistors enabled. To re-gain debug access to a device which contains software that disables the JTAG/SWD pins at startup, the device needs to be held in a reset state with the NRST pin during a POR. This will prevent the application software from starting and will allow the debug probe to gain access to the device, at which point a mass erase DSSM command can be sent from the integrated development environment to the device by the debug probe to remove the application software which is disabling the JTAG/SWD pins.

Note

BOR, BOOTRST, and SYSRST levels do reset the IOMUX logic, which will re-enable the pullup/pulldown resistors on the JTAG/SWD pins. However, the JTAG/SWD functionality remains disabled until the next POR. Because the device always powers up with the pullup and pulldown resistors enabled, the hardware design must accommodate this when using the JTAG/SWD pins for functions other than JTAG/SWD after startup. After reset, application software may disable the pullup/pulldown resistors in the IOMUX to free the JTAG/SWD pins for other purposes.

34.1.1.3 Debug Access Ports

The debug access ports in the DEBUGSS are given in Table 34-1.

Table 34-1. DEBUGSS Access Port Listing

APSEL	AP	Port Description	Purpose
0x0	AHB-AP	CPUSS debug access port	Debug of the processor and peripherals
0x1	CFG-AP	Configuration access port	Access device type information
0x2	SEC-AP	Security access port	Communicate with boot code and access device security information
0x3	ET-AP	EnergyTrace™ technology access port	Read the power state data from EnergyTrace technology for power aware debug
0x4	PWR-AP	Power access port	Configure the device power states (interfaces with PMCU/SYSCTL)

The AHB-AP, PWR-AP, and ET-AP provide the complete device debug functionality (processor debug, peripheral and memory bus access, power state control, and processor state). See Section 34.1.2.1 for more information.

The CFG-AP provides device information to the debug probe so that the debug probe can identify device characteristics, including the device part number and the device revision.

The SEC-AP provides access to the mailbox for communicating with software running on the device through JTAG/SWD. See Section 34.1.2.4 for more information.

34.1.2 DEBUGSS Operation

The features and behaviors of the DEBUGSS module is discussed in the following sections.

34.1.2.1 Debug Features

The DEBUGSS supports processor debug, processor trace, peripheral debug, and energy state debug.

34.1.2.1.1 Processor Debug

The Arm Cortex-M33 processor supports a wide range of features to simplify debugging of application software during development. Key features supported by AM13E230x MCUs include:

- Ability to halt the processor through a assertion of a halt signal, a configured debug event (such as a hard fault entry or reset), or a BKPT instruction (for software breakpoints)
- Ability to step through instructions (with or without peripheral interrupts enabled)
- Ability to run through instructions (with or without peripheral interrupts enabled)
- Ability to read and write CPU registers when halted
- Ability to read exception information through the Cortex-M33 System Control Space (SCS)

ADVANCE INFORMATION

- Support for up to 8 hardware breakpoints
- Support for up to 4 hardware watchpoints
- Support for accessing the device memory map
- Support for Micro Trace Buffer (MTB) up to 4 jump locations

34.1.2.1.1.1 Breakpoint Unit (BPU)

The breakpoint unit (BPU) provides up to 8 comparators which can be used to generate a debug event when the address of an instruction fetch matches the address programmed into the respective BPU comparator.

The BPU does not generate a debug event upon an address match for a data read or data write access.

Address matching is possible for half-word (16-bit) instructions and word (32-bit) instructions fetched from the CODE region (0x0000_0000 to 0x1FFF_FFFF).

If a debug scenario requires more than the supported hardware breakpoints, software breakpoints can be used together with hardware breakpoints using the BKPT instruction. If debugging of code in the SRAM region is desired, hardware breakpoints are not available and software breakpoints must be inserted by the debug probe instead.

```
// Example of a breakpoint function in C (TI Arm CLANG compiler)
__BKPT(0);
```

Note

For more information, please refer to the ARM Cortex-M33 Processor Technical Reference Manual.

34.1.2.1.1.2 Data Watchpoint and Trace Unit (DWT)

The data watchpoint and trace unit (DWT) provides up to 4 comparators which both support generating an event upon a data address match (watchpoint event) or an instruction address match (PC watchpoint event).

The DWT comparators support masking of the address, enabling an event to be generated when the processor attempts to access an address within a specified address range.

Note

For more information, please refer to the ARM Cortex-M33 Processor Technical Reference Manual.

34.1.2.1.1.3 Processor Trace (MTB)

AM13E230x devices support basic instruction execution trace to obtain context for the sequence of execution which led to a certain state of the processor. The processor trace engine is based on the Arm CoreSight MTB-M33 Micro Trace Buffer (MTB).

The MTB captures the processor's program counter (PC) state when the PC changes in a nonsequential way due to a branch instruction or an exception. Load and store activity is not captured by the MTB. When nonsequential execution is detected by the MTB, the change is captured and stored into a small buffer memory (described in Table 34-2) which can be read out later by application software or the debug probe.

Note

For more information, please refer to the ARM CoreSight MTB-M33 Technical Reference Manual.

Table 34-2. MTB Buffer Memory

Start Address	End Address	Length
0x4040_3000	0x4040_301F	32B (4 trace packets)

34.1.2.1.1.4 External Trace (ETM)

AM13E230x devices integrate a single Embedded Trace Macrocell (ETM) that streams a full-instruction program-counter trace together with a 64-bit cycle-accurate timestamp counter. The ETM can be enabled by

ADVANCE INFORMATION

unlocking the debug registers, setting the ETM control register to enable instruction tracing (and optionally the TSCR bit for timestamps), optionally applying address-range filters to limit bandwidth, and then routing the trace pins or the on-chip Trace Port Interface Unit (TPIU) output to a CoreSight-compatible external debugger, which captures the PC-plus-timestamp stream for precise execution-time reconstruction.

Note

For more information, please refer to the ARM CoreSight ETM-M33 Technical Reference Manual.

Table 34-3. ETM Memory

Base Address	Aperture Size	Notes
0x4004_1000	0x1000	Contains control, status, filter, and the 64-bit timestamp registers; accessed via the standard programmer's model for configuration and run-time

34.1.2.1.2 Peripheral Debug

In addition to processor debug, the DEBUGSS can be used to access the device memory map from the perspective of the processor. Thus, a connected debug probe can be used to read and write memory-mapped peripheral registers, the system SRAM, and the flash memory.

Certain peripherals support advanced debug configuration options. These options are configured by application software (or optionally, the debug probe) by setting/clearing various debug control bits in the memory map of a given peripheral. In general, the debug behavior of a particular peripheral is specified in the PDBGCTL register of each peripheral. Many peripherals offer the option of halting the functional clock to the peripheral when the processor is halted for debug, thus pausing the peripheral together with the processor (default configuration), or letting the peripheral run even when the processor is halted for debug.

For example, the WWDT peripheral supports the FREE bit in the PDBGCTL register. Setting the FREE bit in PDBGCTL for a WWDT causes the WWDT counter to run even if the processor is halted for debug. This results in the watchdog not being serviced (causing an interrupt or reset).

34.1.2.1.3 EnergyTrace Technology

The DEBUGSS in AM13E230x devices support EnergyTrace technology. EnergyTrace technology enables power profiling of MCU devices running application code. This is very useful when developing an application which must be optimized for low-power operation.

Development tools from Texas Instruments, including the AM13E230x LaunchPad development tools, support hardware energy measurement of the target AM13E230x over time through EnergyTrace charge counting. This mechanism enables a developer to obtain an energy usage profile for an application, based on real current measurements with a wide dynamic range.

To give context to the energy measurements made by the hardware development tools supporting EnergyTrace technology, AM13E230x MCUs also enable EnergyTrace+. EnergyTrace+ is a component of the DEBUGSS that lets the debug probe log the state of the processor (RUN, SLEEP) and the current program counter value while the device is running. This state information can be then overlaid with energy measurements to determine if the cause of high current is the processor running or some other activity on the device.

TI's Code Composer Studio integrated development environment provides out-of-the-box support for EnergyTrace energy measurement and EnergyTrace+ processor state logging with AM13E230x devices.

Note

EnergyTrace+ state logging is not supported in SHUTDOWN mode, however the energy profiling function of EnergyTrace is still functional

ADVANCE INFORMATION

34.1.2.2 Behavior in Low Power Modes

The DEBUGSS supports maintaining a debug connection through SWD/JTAG in all operating modes except SHUTDOWN. In SHUTDOWN mode, the DEBUGSS supports wakeup of the device to regain full access only via SWD connection.

By default access to device memory and peripherals is possible in RUN mode and SLEEP mode, in which a debug probe can be actively connected to the AHB-AP access port to interface with the processor. In STOP and STANDBY modes, a debug connection can be established and/or maintained with the DEBUGSS, but not with the CPU debug access port. The PWR-AP of the DEBUGSS allows an overwrite of the default behavior to keep access to the processor interface established while being in STOP or STANDBY mode.

In SHUTDOWN mode, any active debug connection is terminated as the debug logic is powered down with the device VCORE. While a debug connection to the DEBUGSS is not possible while the device is in SHUTDOWN mode, a debug probe can cause the device to exit SHUTDOWN mode by attempting to communicate with the SWD pins. The device detects any attempted SWD debug requests even when the device is in SHUTDOWN. If activity is detected, a SHUTDOWN exit is initiated and the device transitions through a BOR state, after which a debug connection can be made to the DEBUGSS through SWD/JTAG.

The DEBUGSS functionality by operating mode is given in Table 34-4.

Table 34-4. DEBUGSS Functionality by Operating Mode

Capability	RUN	SLEEP	STOP	STANDBY	SHUTDOWN	NRST HOLD
Processor debug	Y	Y	N	N	N	N
Memory map access	Y	Y	N	N	N	N
Debug status through SWJ-DP	Y	Y	Y	Y	N	Y
Debug state maintained	Y	Y	Y	Y	N	N
Wake from SWD	-	-	-	-	Y	-

34.1.2.3 Restricting Debug Access

The DEBUGSS supports several methods for restricting access to the device through the SWD/JTAG interface. The debug access policy is determined by the user configuration specified in the NONMAIN flash region.

There are 4 levels of access control, given in Table 34-5. By default, products shipped from TI arrive in a "Debug Enabled" state where the device is fully open. This state is not recommended for production. For production, TI recommends changing the debug configuration to password protected, debug disabled, or SWD/JTAG disabled.

Table 34-5. Debug Access Control

DEBUGSS Function	Debug Configuration			
	Debug Enabled (default)	Debug Enabled with Password	Debug Disabled	SWD/JTAG Disabled
SWJ-DP (debug port)	EN	EN	EN	DIS
CFG-AP	EN	EN	EN	DIS
SEC-AP	EN	EN	EN	DIS
ET-AP	EN	EN w/ Password	DIS	DIS
AHB-AP (CPU Debug)	EN	EN w/ Password	DIS	DIS

The debug access control can be configured by writing predefined values into the NONMAIN BOOTCFG0.

Table 34-6. Debug Access Control Settings for BOOTCFG0

Debug Configuration	SWJDP_MODE	DEBUGACCESS
Debug Enabled (default)	AABBh	AABBh
Debug Enabled with Password	AABBh	CCDDh
Debug Disabled	AABBh	5566h

Table 34-6. Debug Access Control Settings for BOOTCFG0 (continued)

Debug Configuration	SWJDP_MODE	DEBUGACCESS
SWD Disabled	5566h	5566h

When debug is set to enabled with password, the debug access command together with the user-specified debug access password must be provided to the DEBUGSS mailbox, and a BOOTRST must be issued. The password for access control is stored in NONMAIN utilizing the PWDDEBUGLOCK registers. On AM13E230x devices, the password can either be:

1. 128-bit plain text
   a. Simple password match flow, common for all devices
2. 128-bit AES encryption of the DEVICE_UID
   a. Unique password for each device
3. 128-bit CMAC signature of the DEVICE_UID
   a. Unique password for each device

When debug is disabled, the SWJ-DP is disabled during the boot process and any commands previously sent to the mailbox are ignored during boot. Following boot, any attempt to connect to the SWJ-DP is ignored.

Permanently lock debug access to the device by configuring the NONMAIN flash region to disable debug access while also configuring the NONMAIN flash region as statically write protected (locked). Locking the NONMAIN configuration has the added security of preventing the bootstrap loader (BSL) and application code from changing the debug security policy.

Table 34-7. Debug related password registers

Debug Function	Debug Access Control	Factory Reset	Mass Erase
Register Name	PWDDEBUGLOCK	PWDFACTORYRESET	PWDMASSERASE

Example: 128-bit plain text password

To enable a password on a debug function write a 128-bit hex value split into four 32-bit words into the respective password register.

Example: Factory Reset Password Configuration

1. Create 128-bit hex value
   a. 0xCAFECAFE12345678A5A5C3C30000FFFF
2. Split the 128-bit value into four 32-bit words and write the value into the respective registers
   a. PWDFACTORYRESET[0]←0x0000FFFF
   b. PWDFACTORYRESET[1]←0xA5A5C3C3
   c. PWDFACTORYRESET[2]←0x12345678
   d. PWDFACTORYRESET[3]←0xCAFECAFE
3. Repeat the same steps for any function that needs password protection

34.1.2.4 Mailbox (DSSM)

The Debug Subsystem Mailbox (DSSM) enables a debug probe to pass messages to the target device through the SWJ-DP interface, as well as making it possible for the target device to return data to the debug probe.

The DSSM supports the following functions:

- Transmission of commands to the device during boot, including authenticating the debug probe for password-protected debug, mass erase, and factory reset operations

ADVANCE INFORMATION

- Communicating with application software running on the target device when no other communication interface is present

Two 32-bit word data buffers are provided for TX data (debug probe to target device) and RX data (target device to debug probe). These data buffers are implemented as 32-bit memory-mapped registers in the DEBUGSS. In addition, TXCTL and RXCTL registers are provided for enabling flow control and indicating status of the mailbox.

Table 34-8. DSSM Register Functions

DSSM Register	Description	Debug Probe	Target Device	Actions
TX_DATA	Data buffer	RW	R	TXCTL.TRANSMIT is set on write by the debug probe, and cleared on a read by the target device; TXIFG is also set on a write by the debug probe
TXCTL	Flow control and status	RW	R	None
RX_DATA	Data buffer	R	RW	RXCTL.RECEIVE is set on write by the target device, and cleared on a read by the debug probe; RXIFG is also set on a write by the target device
RXCTL	Flow control and status	R	RW	None

The TXCTL and RXCTL registers provide generic transmit and receive flags. BIT0 of the TXCTL and RXCTL registers provide a specific TRANSMIT and RECEIVE flag which indicate the status of the TX_DATA and RX_DATA. The TRANSMIT bit is set in the TXCTL register when a debug probe writes data to the TX_DATA buffer register. The TRANSMIT flag will then remain set until the CPU in the target device reads TX_DATA or a POR occurs. The RECEIVE flag is set in the RXCTL register when the CPU in the target device writes data to the RX_DATA buffer register. The RECEIVE flag will then remain set until the debug probe reads the data from RX_DATA.

It is not possible for software running on the target device to write to TX_DATA, and it is also not possible for target software to clear the TRANSMIT flag other than by reading TX_DATA.

The upper 31 bits of the TXCTL register, TRANSMIT_FLAGS, contain generic flag bits which can be set or cleared by the debug probe to implement a protocol if desired. Only the debug probe can write to the TRANSMIT_FLAGS field in TXCTL. The Table 34-12 is an example of a protocol to communicate with the target device. Also refer to Boot Configuration (TBD: M33 chapter) for security level control related to the DSSM commands.

In a similar way, only the target device software can write to RX_DATA and RXCTL. The debug probe cannot write to RX_DATA and it can only clear the RECEIVE flag in RXCTL by reading RX_DATA. BIT1 through BIT7 (0xFE) of RXCTL contains the RECEIVE_FLAGS field. Software on the target device can set or clear bits in the RECEIVE_FLAGS field to implement a protocol if desired. These flags can be read by the debug probe but can not be modified by the debug probe.

Table 34-9. DEBUGSS SEC-AP Register Definition

AP SEL	ADDR	BANK	INDEX	REGISTER NAME	BIT 31 - 8	BIT 7 - 1	BIT 0
2	0x00	0	0	TXDATA	TX DATA		
	0x04	0	1	TXCTL	TRANSMIT_FLAGS		TRANSMIT
	0x08	0	2	RXDATA	RX Data		
	0x0C	0	3	RXCTL	Reserved	RECEIVE_FLAGS	RECEIVE
	0x0FC	15	3	IDR	Access point ID		

ADVANCE INFORMATION

34.1.2.4.1 DSSM Events

The DSSM contains one event publisher and no event subscribers. One event publisher (CPU_INT) manages DSSM interrupt requests (IRQs) to the CPU subsystem through a static event route.

The DSSM events are summarized in Table 34-10.

Table 34-10. DSSM Events

Event	Type	Source	Destination	Route	Configuration	Functionality
CPU Interrupt Event	Publisher	DEBUGSS	CPU Subsystem	Static route	CPU_INT registers	Fixed interrupt route from DEBUGSS to CPU

34.1.2.4.1.1 CPU Interrupt Event (CPU_INT)

The DSSM provides 4 interrupt sources which can be configured to source a CPU interrupt event. In order of decreasing interrupt priority, the CPU interrupt events from the DSSM are given in Table 34-11.

Table 34-11. DSSM CPU Interrupt Event Conditions (CPU_INT)

Index (IIDX)	Name	Description
0	TXIFG	Indicates that the TX_DATA buffer in the DSSM has received data.
1	RXIFG	Indicates that the data in RX_DATA buffer in the DSSM was read.
2	PWRUPIFG	Indicates that the DEBUGSS was started due to a debug probe attaching to the device.
3	PWRDWNIFG	Indicates that the DEBUGSS was stopped due to a debug probe disconnecting from the device.

The CPU interrupt event configuration is managed with the CPU_INT event management registers. See *Using Event Registers* for guidance on configuring the Event registers for CPU interrupts.

34.1.2.4.2 DSSM Commands

DSSM commands are serviced by the bootcode upon a BOOTRST. If passwords are enabled for DSSM commands, the commands do not execute until the correct password sequence is entered. Refer to Table 34-12for possible DSSM commands and values.

Table 34-12. DSSM Commands

DSSM Command	Command ID
Wait For Debug	0x03
Invoke BSL	0x04
Factory Reset	0x05
Mass Erase	0x06
Command Debug	0x07
Get Lifecycle	0x0A
Get Device ID	0x0B
Key Provisioning	0x0E
Convert Bad Device	0x15
Update Active Key Revision	0x16
Update Application Revision	0x17
Data Exchange	0x77

Invoke BSL

The "Invoke BSL" command invokes the BSL instead of using GPIO.

Factory Reset

The "Factory Reset" command erases all contents within MAIN and NONMAIN flash memory. Then NONMAIN memory is re-populated with default contents. This command is useful in the cases of:

- NONMAIN misconfiguration
- Disabled debug access
- Peripheral/device misconfiguration

Factory reset has requires a configuration and password, please refer to Section 7.3.1 for security level details.

Note

The factory reset restores the NONMAIN configuration memory and therefore all previous security level settings are lost.

Mass Erase

The "Mass Erase" command erases all contents within the MAIN memory but NONMAIN remains untouched. This command is useful in the case of:

- Peripheral/device misconfiguration

Command Debug

The "Command Debug" command allows debugger access for a session when debug is locked.

Data Exchange

Data exchange is the only DSSM command that does not require a BOOTRST to process the command. This command is used in combination with factory reset, mass erase, or password authentication which do require a password. After sending the initial command to the mailbox and performing a reset, the user must begin sending the password to the TXDATA register. After each word 0x77 must be written to the TXCTL register.

Wait for Debug

The "Wait for Debug" command resets the peripherals defined by the reset level and then forces the device into the reset handler.

Get Lifecycle

The "Get Lifecycle" command returns the current lifecycle of the device.

Get Device ID

The "Get Device ID" command returns the unique ID of the device.

Key Provisioning

The "Key Provisioning" command provisions AES keys into NONMAIN memory.

Convert Bad Device

The "Convert Bad Device" command designates a device as decommissioned.

Update Active Key Revision

The "Update Active Key Revision" command updates the key revision and changes the active key.

Update Application Revision

The "Update Application Revision" command updates the application revision.

Custom DSSM Command

The user can also create a custom DSSM command and perform actions defined by the user. This is done by having the debugger communicate to the M33 core through the TXDATA and TXCTL registers. The core can then receive messages from the debugger and send responses back by using the RXDATA and RXCTL registers for the debugger to read. CPU interrupt events can be configured for activity seen in the TX_DATA buffer, RX_DATA buffer, and DAP connection.

34.2 DEBUGSS Registers

This Section describes the DEBUGSS Registers.

34.2.1 DEBUGSS Base Address Table

Table 34-13. DEBUGSS Base Address Table

Bit Field Name		DriverLib Name	Base Address
Instance	Structure		
DebugssRegs	DEBUGSS_REGS	DEBUGSS	0x400C_7000

34.2.2 DEBUGSS_REGS Registers

Table 34-14 lists the memory-mapped registers for the DEBUGSS_REGS registers. All register offset addresses not listed in Table 34-14 should be considered as reserved locations and the register contents should not be modified.

Table 34-14. DEBUGSS_REGS Registers

Offset	Acronym	Register Name	Section
4h	SWCLK	SWCLK	Go
8h	SWDIO	SWDIO	Go
Ch	SWCLK_ALT	SWCLK_ALT	Go
10h	SWDIO_ALT	SWDIO_ALT	Go
480h	CPU_CONNECT_0	CPU Connect	Go
1020h	IIDX	Interrupt index	Go
1028h	IMASK	Interrupt mask	Go
1030h	RIS	Raw interrupt status	Go
1038h	MIS	Masked interrupt status	Go
1040h	ISET	Interrupt set	Go
1048h	ICLR	Interrupt clear	Go
10E0h	EVT_MODE	Event Mode	Go
10FCh	DESC	Module Description	Go
1100h	TXD	Transmit data register	Go
1104h	TXCTL	Transmit control register	Go
1108h	RXD	Receive data register	Go
110Ch	RXCTL	Receive control register	Go
1200h	SPECIAL_AUTH	Special enable authorization register	Go
1210h	APP_AUTH	Application CPU0 authorization register	Go

Complex bit access types are encoded to fit into small table cells. Table 34-15 shows the codes that are used for access types in this section.

Table 34-15. DEBUGSS_REGS Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
R-0	R -0	Read Returns 0s
Write Type		
W	W	Write
WK	W K	Write Write protected by a key
Reset or Default Value		
-n		Value after reset or the default value

1 SWCLK Register (Offset = 4h) [Reset = 00000000h]

SWCLK is shown in Figure 34-2 and described in Table 34-16.

Return to the Summary Table.

Serial Wire Debug Clock (SWCLK)

Figure 34-2. SWCLK Register

31	30	29	28	27	26	25	24
RESERVED			WCOMP	WUEN	INV	HIGHZ1	HIGHZ0
R/W-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
RESERVED	DRV			HYSTEN	INENA	PIPU	PIPD
R/W-0h	R/W-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
GSTATE		RESERVED					
R/W-0h		R/W-0h					
7	6	5	4	3	2	1	0
PSTATE		RESERVED					
R/W-0h		R/W-0h					

Table 34-17. SWCLK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R/W	0h	
28	WCOMP	R/W	0h	Wake up compare value 0h = Match 0 will wake 1h = Match 1 will wake
27	WUEN	R/W	0h	Wake up enable 0h = Wake up not enabled 1h = Wake up enabled
26	INV	R/W	0h	Invert digital input/output relative to peripheral/GPIO 0h = Input and output are non-inverted 1h = Input and output are inverted
25	HIGHZ1	R/W	0h	High-Z instead of high output 0h = Pin can be driven high 1h = Pin is tri-stated instead of driven high
24	HIGHZ0	R/W	0h	High-Z instead of low output 0h = Pin can be driven low 1h = Pin is tri-stated instead of driven low
23	RESERVED	R/W	0h	
22-20	DRV	R/W	0h	Drive strength options 0h = Lowest drive strength 1h = Drive strength 2/8 2h = Drive strength 3/8 3h = Drive strength 4/8 4h = Drive strength 5/8 5h = Drive strength 6/8 6h = Drive strength 7/8 7h = Highest drive strength
19	HYSTEN	R/W	0h	Hysteresis enable 0h = No hysteresis 1h = Hysteresis on
18	INENA	R/W	0h	Input enable 0h = Inputs 0 to connected core 1h = Inputs IO pad value to connected core

Table 34-17. SWCLK Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
17	PIPU	R/W	0h	Pull up enable 0h = No pull up 1h = Pull up
16	PIPD	R/W	0h	Pull down enable 0h = No pull down 1h = Pull down
15-14	GSTATE	R/W	0h	GPIO Channel State 0h = G-Channel is in Unassigned State 1h = G-Channel is in Handover State 2h = G-Channel is in Connected State and not Locked (That is F field is allowed to change without going back through Unassigned state) 3h = G-Channel is in Connected State and Locked (That is F field is not allowed to change to a different non-Zero value until both G and P channels go to Unassigned)
13-8	RESERVED	R/W	0h	
7-6	PSTATE	R/W	0h	Peripheral-Analog Channel State 0h = P-Channel is in Unassigned State 1h = P-Channel is in Handover State 2h = P-Channel is in Connected State and not Locked (That is F field is allowed to change without going back through Unassigned state) 3h = P-Channel is in Connected State and Locked (That is F field is not allowed to change to a different non-Zero value until both G and P channels go to Unassigned)
5-0	RESERVED	R/W	0h	

2 SWDIO Register (Offset = 8h) [Reset = 00000000h]

SWDIO is shown in Figure 34-3 and described in Table 34-17.

Return to the Summary Table.

Serial Wire Debug Data Input / Output (SWDIO)

Figure 34-3. SWDIO Register

31	30	29	28	27	26	25	24
RESERVED			WCOMP	WUEN	INV	HIGHZ1	HIGHZ0
R/W-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
RESERVED	DRV			HYSTEN	INENA	PIPU	PIPD
R/W-0h	R/W-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
GSTATE		RESERVED					
R/W-0h		R/W-0h					
7	6	5	4	3	2	1	0
PSTATE		RESERVED					
R/W-0h		R/W-0h					

Table 34-19. SWDIO Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R/W	0h	
28	WCOMP	R/W	0h	Wake up compare value 0h = Match 0 will wake 1h = Match 1 will wake
27	WUEN	R/W	0h	Wake up enable 0h = Wake up not enabled 1h = Wake up enabled
26	INV	R/W	0h	Invert digital input/output relative to peripheral/GPIO 0h = Input and output are non-inverted 1h = Input and output are inverted
25	HIGHZ1	R/W	0h	High-Z instead of high output 0h = Pin can be driven high 1h = Pin is tri-stated instead of driven high
24	HIGHZ0	R/W	0h	High-Z instead of low output 0h = Pin can be driven low 1h = Pin is tri-stated instead of driven low
23	RESERVED	R/W	0h	
22-20	DRV	R/W	0h	Drive strength options 0h = Lowest drive strength 1h = Drive strength 2/8 2h = Drive strength 3/8 3h = Drive strength 4/8 4h = Drive strength 5/8 5h = Drive strength 6/8 6h = Drive strength 7/8 7h = Highest drive strength
19	HYSTEN	R/W	0h	Hysteresis enable 0h = No hysteresis 1h = Hysteresis on
18	INENA	R/W	0h	Input enable 0h = Inputs 0 to connected core 1h = Inputs IO pad value to connected core

Table 34-19. SWDIO Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
17	PIPU	R/W	0h	Pull up enable 0h = No pull up 1h = Pull up
16	PIPD	R/W	0h	Pull down enable 0h = No pull down 1h = Pull down
15-14	GSTATE	R/W	0h	GPIO Channel State 0h = G-Channel is in Unassigned State 1h = G-Channel is in Handover State 2h = G-Channel is in Connected State and not Locked (That is F field is allowed to change without going back through Unassigned state) 3h = G-Channel is in Connected State and Locked (That is F field is not allowed to change to a different non-Zero value until both G and P channels go to Unassigned)
13-8	RESERVED	R/W	0h	
7-6	PSTATE	R/W	0h	Peripheral-Analog Channel State 0h = P-Channel is in Unassigned State 1h = P-Channel is in Handover State 2h = P-Channel is in Connected State and not Locked (That is F field is allowed to change without going back through Unassigned state) 3h = P-Channel is in Connected State and Locked (That is F field is not allowed to change to a different non-Zero value until both G and P channels go to Unassigned)
5-0	RESERVED	R/W	0h	

3 SWCLK_ALT Register (Offset = Ch) [Reset = 00000000h]

SWCLK_ALT is shown in Figure 34-4 and described in Table 34-18.

Return to the Summary Table.

Alternate Serial Wire Debug Clock (SWCLK)

Figure 34-4. SWCLK_ALT Register

31	30	29	28	27	26	25	24
RESERVED			WCOMP	WUEN	INV	HIGHZ1	HIGHZ0
R/W-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
RESERVED	DRV			HYSTEN	INENA	PIPU	PIPD
R/W-0h	R/W-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
GSTATE		RESERVED					
R/W-0h		R/W-0h					
7	6	5	4	3	2	1	0
PSTATE		RESERVED					
R/W-0h		R/W-0h					

Table 34-21. SWCLK_ALT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R/W	0h	
28	WCOMP	R/W	0h	Wake up compare value 0h = Match 0 will wake 1h = Match 1 will wake
27	WUEN	R/W	0h	Wake up enable 0h = Wake up not enabled 1h = Wake up enabled
26	INV	R/W	0h	Invert digital input/output relative to peripheral/GPIO 0h = Input and output are non-inverted 1h = Input and output are inverted
25	HIGHZ1	R/W	0h	High-Z instead of high output 0h = Pin can be driven high 1h = Pin is tri-stated instead of driven high
24	HIGHZ0	R/W	0h	High-Z instead of low output 0h = Pin can be driven low 1h = Pin is tri-stated instead of driven low
23	RESERVED	R/W	0h	
22-20	DRV	R/W	0h	Drive strength options 0h = Lowest drive strength 1h = Drive strength 2/8 2h = Drive strength 3/8 3h = Drive strength 4/8 4h = Drive strength 5/8 5h = Drive strength 6/8 6h = Drive strength 7/8 7h = Highest drive strength
19	HYSTEN	R/W	0h	Hysteresis enable 0h = No hysteresis 1h = Hysteresis on
18	INENA	R/W	0h	Input enable 0h = Inputs 0 to connected core 1h = Inputs IO pad value to connected core

Table 34-21. SWCLK_ALT Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
17	PIPU	R/W	0h	Pull up enable 0h = No pull up 1h = Pull up
16	PIPD	R/W	0h	Pull down enable 0h = No pull down 1h = Pull down
15-14	GSTATE	R/W	0h	GPIO Channel State 0h = G-Channel is in Unassigned State 1h = G-Channel is in Handover State 2h = G-Channel is in Connected State and not Locked (That is F field is allowed to change without going back through Unassigned state) 3h = G-Channel is in Connected State and Locked (That is F field is not allowed to change to a different non-Zero value until both G and P channels go to Unassigned)
13-8	RESERVED	R/W	0h	
7-6	PSTATE	R/W	0h	Peripheral-Analog Channel State 0h = P-Channel is in Unassigned State 1h = P-Channel is in Handover State 2h = P-Channel is in Connected State and not Locked (That is F field is allowed to change without going back through Unassigned state) 3h = P-Channel is in Connected State and Locked (That is F field is not allowed to change to a different non-Zero value until both G and P channels go to Unassigned)
5-0	RESERVED	R/W	0h	

4 SWDIO_ALT Register (Offset = 10h) [Reset = 00000000h]

SWDIO_ALT is shown in Figure 34-5 and described in Table 34-19.

Return to the Summary Table.

Alternate Serial Wire Debug Data Input / Output (SWDIO)

Figure 34-5. SWDIO_ALT Register

31	30	29	28	27	26	25	24
RESERVED			WCOMP	WUEN	INV	HIGHZ1	HIGHZ0
R/W-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
RESERVED	DRV			HYSTEN	INENA	PIPU	PIPD
R/W-0h	R/W-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
GSTATE		RESERVED					
R/W-0h		R/W-0h					
7	6	5	4	3	2	1	0
PSTATE		RESERVED					
R/W-0h		R/W-0h					

Table 34-23. SWDIO_ALT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	RESERVED	R/W	0h	
28	WCOMP	R/W	0h	Wake up compare value 0h = Match 0 will wake 1h = Match 1 will wake
27	WUEN	R/W	0h	Wake up enable 0h = Wake up not enabled 1h = Wake up enabled
26	INV	R/W	0h	Invert digital input/output relative to peripheral/GPIO 0h = Input and output are non-inverted 1h = Input and output are inverted
25	HIGHZ1	R/W	0h	High-Z instead of high output 0h = Pin can be driven high 1h = Pin is tri-stated instead of driven high
24	HIGHZ0	R/W	0h	High-Z instead of low output 0h = Pin can be driven low 1h = Pin is tri-stated instead of driven low
23	RESERVED	R/W	0h	
22-20	DRV	R/W	0h	Drive strength options 0h = Lowest drive strength 1h = Drive strength 2/8 2h = Drive strength 3/8 3h = Drive strength 4/8 4h = Drive strength 5/8 5h = Drive strength 6/8 6h = Drive strength 7/8 7h = Highest drive strength
19	HYSTEN	R/W	0h	Hysteresis enable 0h = No hysteresis 1h = Hysteresis on
18	INENA	R/W	0h	Input enable 0h = Inputs 0 to connected core 1h = Inputs IO pad value to connected core

Table 34-23. SWDIO_ALT Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
17	PIPU	R/W	0h	Pull up enable 0h = No pull up 1h = Pull up
16	PIPD	R/W	0h	Pull down enable 0h = No pull down 1h = Pull down
15-14	GSTATE	R/W	0h	GPIO Channel State 0h = G-Channel is in Unassigned State 1h = G-Channel is in Handover State 2h = G-Channel is in Connected State and not Locked (That is F field is allowed to change without going back through Unassigned state) 3h = G-Channel is in Connected State and Locked (That is F field is not allowed to change to a different non-Zero value until both G and P channels go to Unassigned)
13-8	RESERVED	R/W	0h	
7-6	PSTATE	R/W	0h	Peripheral-Analog Channel State 0h = P-Channel is in Unassigned State 1h = P-Channel is in Handover State 2h = P-Channel is in Connected State and not Locked (That is F field is allowed to change without going back through Unassigned state) 3h = P-Channel is in Connected State and Locked (That is F field is not allowed to change to a different non-Zero value until both G and P channels go to Unassigned)
5-0	RESERVED	R/W	0h	

5 CPU_CONNECT_0 Register (Offset = 480h) [Reset = 00000000h]

CPU_CONNECT_0 is shown in Figure 34-6 and described in Table 34-20.

Return to the Summary Table.

Directly connect peripheral publisher port to application processor

Figure 34-6. CPU_CONNECT_0 Register

31	30	29	28	27	26	25	24
RESERVED							
R/W-0h							
23	22	21	20	19	18	17	16
RESERVED							
R/W-0h							
15	14	13	12	11	10	9	8
RESERVED							
R/W-0h							
7	6	5	4	3	2	1	0
RESERVED						CPUSS0_CONN	RESERVED
R/W-0h						R/W-0h	R/W-0h

Table 34-25. CPU_CONNECT_0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R/W	0h	
1	CPUSS0_CONN	R/W	0h	CPUSS0 connect bit. 0h = The CPU is not connected. 1h = The CPU is connected.
0	RESERVED	R/W	0h	

ADVANCE INFORMATION

6 IIDX Register (Offset = 1020h) [Reset = 00000000h]

IIDX is shown in Figure 34-7 and described in Table 34-21.

Return to the Summary Table.

This register provides the highest priority enabled interrupt index. 0xFF means no event pending. Interrupt 0x0 is the highest priority, 0x1 next highest, and 0xFE is the least priority. The priority order is fixed. However, users can implement their own prioritization schemes using other registers that expose the full set of interrupts that have occurred.
On each read, only one interrupt is indicated. On a read, the current interrupt (highest priority) is automatically cleared by the hardware and the corresponding interrupt flag in the RIS and MIS are cleared as well. After a read from the CPU (not from the debug interface), the register must be updated with the next highest priority interrupt, if none are pending, then it displays 0xFF.

Figure 34-7. IIDX Register

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

Table 34-27. IIDX Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	
7-0	STAT	R	0h	Interrupt index status 0h = No pending interrupt request 1h = TX interrupt 2h = RX interrupt 3h = Power-up interrupt. A debug session has started. 4h = Power-up interrupt. A debug session has started.

7 IMASK Register (Offset = 1028h) [Reset = 00000000h]

IMASK is shown in Figure 34-8 and described in Table 34-22.

Return to the Summary Table.

Interrupt Mask. If a bit is set, then corresponding interrupt is un-masked. Un-masking the interrupt causes the raw interrupt to be visible in IIDX, as well as MIS.

Figure 34-8. IMASK Register

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

Table 34-29. IMASK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R/W	0h	
3	PWRDWNIFG	R/W	0h	Masks PWRDWNIFG in MIS register 0h = Interrupt is masked out 1h = Interrupt will request an interrupt service routine and corresponding bit in MIS will be set
2	PWRUPIFG	R/W	0h	Masks PWRUPIFG in MIS register 0h = Interrupt is masked out 1h = Interrupt will request an interrupt service routine and corresponding bit in MIS will be set
1	RXIFG	R/W	0h	Masks RXIFG in MIS register 0h = Interrupt is masked out 1h = Interrupt will request an interrupt service routine and corresponding bit in MIS will be set
0	TXIFG	R/W	0h	Masks TXIFG in MIS register 0h = Interrupt is masked out 1h = Interrupt will request an interrupt service routine and corresponding bit in MIS will be set

8 RIS Register (Offset = 1030h) [Reset = 00000000h]

RIS is shown in Figure 34-9 and described in Table 34-23.

Return to the Summary Table.

Raw interrupt status. Reflects all pending interrupts, regardless of masking. The RIS register allows the user to implement a poll scheme. A flag set in this register can be cleared by writing 1 to the ICLR register bit even if the corresponding IMASK bit is not enabled.

Figure 34-9. RIS Register

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

Table 34-31. RIS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R	0h	
3	PWRDWNIFG	R	0h	Raw interrupt status for PWRDWNIFG 0h = PWRUPIFG did not occur 1h = PWRUPIFG occurred
2	PWRUPIFG	R	0h	Raw interrupt status for PWRUPIFG 0h = PWRUPIFG did not occur 1h = PWRUPIFG occurred
1	RXIFG	R	0h	Raw interrupt status for RXIFG 0h = RXIFG did not occur 1h = RXIFG occurred
0	TXIFG	R	0h	Raw interrupt status for TXIFG 0h = TXIFG did not occur 1h = TXIFG occurred

9 MIS Register (Offset = 1038h) [Reset = 00000000h]

MIS is shown in Figure 34-10 and described in Table 34-24.

Return to the Summary Table.

Masked interrupt status. This is an AND of the IMASK and RIS registers.

Figure 34-10. MIS Register

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

Table 34-33. MIS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R	0h	
3	PWRDWNIFG	R	0h	Masked interrupt status for PWRDWNIFG 0h = PWRUPIFG did not request an interrupt service routine 1h = PWRUPIFG requests an interrupt service routine
2	PWRUPIFG	R	0h	Masked interrupt status for PWRUPIFG 0h = PWRUPIFG did not request an interrupt service routine 1h = PWRUPIFG requests an interrupt service routine
1	RXIFG	R	0h	Masked interrupt status for RXIFG 0h = RXIFG did not request an interrupt service routine 1h = RXIFG requests an interrupt service routine
0	TXIFG	R	0h	Masked interrupt status for TXIFG 0h = TXIFG did not request an interrupt service routine 1h = TXIFG requests an interrupt service routine

10 ISET Register (Offset = 1040h) [Reset = 00000000h]

ISET is shown in Figure 34-11 and described in Table 34-25.

Return to the Summary Table.

Interrupt set. Allows interrupts to be set by software (useful in diagnostics and safety checks). Writing a 1 to a bit in ISET will set the event and therefore the related RIS bit also gets set. If the interrupt is enabled through the mask, then the corresponding MIS bit is also set.

Figure 34-11. ISET Register

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

Table 34-35. ISET Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	W	0h	
3	PWRDWNIFG	W	0h	Sets PWRDWNIFG in RIS register 0h = Writing a 0 has no effect 1h = RIS bit corresponding to PWRUPIFG is set
2	PWRUPIFG	W	0h	Sets PWRUPIFG in RIS register 0h = Writing a 0 has no effect 1h = RIS bit corresponding to PWRUPIFG is set
1	RXIFG	W	0h	Sets RXIFG in RIS register 0h = Writing a 0 has no effect 1h = RIS bit corresponding to RXIFG is set
0	TXIFG	W	0h	Sets TXIFG in RIS register 0h = Writing a 0 has no effect 1h = RIS bit corresponding to TXIFG is set

11 ICLR Register (Offset = 1048h) [Reset = 00000000h]

ICLR is shown in Figure 34-12 and described in Table 34-26.

Return to the Summary Table.

Interrupt clear. Write a 1 to clear corresponding Interrupt.

Figure 34-12. ICLR Register

31	30	29	28	27	26	25	24
RESERVED							
W-0h							
23	22	21	20	19	18	17	16
RESERVED							
W-0h							
15	14	13	12	11	10	9	8
RESERVED							
W-0h							
7	6	5	4	3	2	1	0
RESERVED				PWRDWNIFG	PWRUPIFG	RXIFG	TXIFG
W-0h				W-0h	W-0h	W-0h	W-0h

Table 34-37. ICLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	W	0h	
3	PWRDWNIFG	W	0h	Clears PWRDWNIFG in RIS register 0h = Writing a 0 has no effect 1h = RIS bit corresponding to PWRUPIFG is cleared
2	PWRUPIFG	W	0h	Clears PWRUPIFG in RIS register 0h = Writing a 0 has no effect 1h = RIS bit corresponding to PWRUPIFG is cleared
1	RXIFG	W	0h	Clears RXIFG in RIS register 0h = Writing a 0 has no effect 1h = RIS bit corresponding to RXIFG is cleared
0	TXIFG	W	0h	Clears TXIFG in RIS register 0h = Writing a 0 has no effect 1h = RIS bit corresponding to TXIFG is cleared

12 EVT_MODE Register (Offset = 10E0h) [Reset = 00000000h]

EVT_MODE is shown in Figure 34-13 and described in Table 34-27.

Return to the Summary Table.

Event mode register. It is used to select whether each line is disabled, in software mode (software clears the RIS) or in hardware mode (hardware clears the RIS)

Figure 34-13. EVT_MODE Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED						INT0_CFG	
R-0h						R-0h	

Table 34-39. EVT_MODE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	RESERVED	R	0h	
1-0	INT0_CFG	R	1h	Event line mode select for peripheral events 0h = The interrupt or event line is disabled. 1h = The interrupt or event line is in software mode. Software must clear the RIS. 2h = The interrupt or event line is in hardware mode. The hardware (another module) clears automatically the associated RIS flag.

13 DESC Register (Offset = 10FCh) [Reset = 00000000h]

DESC is shown in Figure 34-14 and described in Table 34-28.

Return to the Summary Table.

This register identifies the peripheral and its exact version.

Figure 34-14. DESC Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MODULEID															
R-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FEATUREVER				INSTNUM				MAJREV				MINREV			
R-0h				R-0h				R-0h				R-0h			

Table 34-41. DESC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	MODULEID	R	340h	Module identification contains a unique peripheral identification number. The assignments are maintained in a central database for all of the platform modules to ensure uniqueness.
15-12	FEATUREVER	R	0h	Feature Set for the module *instance*
11-8	INSTNUM	R	0h	Instance Number within the device. This will be a parameter to the RTL for modules that can have multiple instances
7-4	MAJREV	R	0h	Major rev of the IP
3-0	MINREV	R	0h	Minor rev of the IP

14 TXD Register (Offset = 1100h) [Reset = 00000000h]

TXD is shown in Figure 34-15 and described in Table 34-29.

Return to the Summary Table.

This register is used for data transfers from external debug tools to the DSSM module. The register is written by the debug tool and read by the CPU.

Figure 34-15. TXD Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TX_DATA																															
R-0h																															

Table 34-43. TXD Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TX_DATA	R	0h	Contains data written by an external debug tool to the SEC-AP TXDATA register

15 TXCTL Register (Offset = 1104h) [Reset = 00000000h]

TXCTL is shown in Figure 34-16 and described in Table 34-30.

Return to the Summary Table.

Transmit control register

Figure 34-16. TXCTL Register

31	30	29	28	27	26	25	24
TRANSMIT_FLAGS							
R-0h							
23	22	21	20	19	18	17	16
TRANSMIT_FLAGS							
R-0h							
15	14	13	12	11	10	9	8
TRANSMIT_FLAGS							
R-0h							
7	6	5	4	3	2	1	0
TRANSMIT_FLAGS							TRANSMIT
R-0h							R-0h

Table 34-45. TXCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	TRANSMIT_FLAGS	R	0h	Generic TX flags that can be set by external debug tool. Functionality is defined by SW.
0	TRANSMIT	R	0h	Indicates data request in DSSM.TXD, set on write via Debug AP to DSSM.TXD. A read of the DSSM.TXD register by SW will clear the TX field. The tool can check that TXD is empty by reading this field. 0h = TXD is empty 1h = TXD is full

16 RXD Register (Offset = 1108h) [Reset = 00000000h]

RXD is shown in Figure 34-17 and described in Table 34-31.

Return to the Summary Table.

Receive data register. This register contains the data written by the CPU.
This data is read by external debug tool.

Figure 34-17. RXD Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RX_DATA																															
R/W-0h																															

Table 34-47. RXD Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RX_DATA	R/W	0h	Contains data written by SM/OW.

ADVANCE INFORMATION

17 RXCTL Register (Offset = 110Ch) [Reset = 00000000h]

RXCTL is shown in Figure 34-18 and described in Table 34-32.

Return to the Summary Table.

Receive control register

Figure 34-18. RXCTL Register

31	30	29	28	27	26	25	24
RESERVED							
R/W-0h							
23	22	21	20	19	18	17	16
RESERVED							
R/W-0h							
15	14	13	12	11	10	9	8
RESERVED							
R/W-0h							
7	6	5	4	3	2	1	0
RECEIVE_FLAGS							RECEIVE
R/W-0h							R-0h

Table 34-49. RXCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	RESERVED	R/W	0h	
7-1	RECEIVE_FLAGS	R/W	0h	Generic RX flags that can be set by SW and read by external debug tool. Functionality is defined by SW.
0	RECEIVE	R	0h	Indicates SW write to the DSSM.RXD register. A read of the DSSM.RXD register by SWD Access Port will clear the RX field. 0h = RXD empty 1h = RXD full

18 SPECIAL_AUTH Register (Offset = 1200h) [Reset = 00000013h]

SPECIAL_AUTH is shown in Figure 34-19 and described in Table 34-33.

Return to the Summary Table.

This register is used to control ET-AP, DFT-TAP, SWD, CFG-AP and SEC-AP.

Figure 34-19. SPECIAL_AUTH Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED	PWRAPEN	AHBAPEN	CFGAPEN	ETAPEN	DFTAPEN	SWDPORTEN	SECAPEN
R-0h	R-0h	R-0h	R-1h	R-0h	R-0h	R-1h	R-1h

Table 34-51. SPECIAL_AUTH Register Field Descriptions

Bit	Field	Type	Reset	Description
31-7	RESERVED	R	0h	
6	PWRAPEN	R	0h	An active high input. When asserted (and SWD access is also permitted), the debug tools can then access the PWR-AP to power and reset state of the CPU. When deasserted, a DAPBUS firewall will isolate the AP and prevent access. 0h = Disable PWR-AP 1h = Enable PWR-AP
5	AHBAPEN	R	0h	Disabling / enabling debug access to the M0+ Core via the AHB-AP DAP bus isolation. 0h = Disable AHB-AP 1h = Enable AHB-AP
4	CFGAPEN	R	1h	An active high input. When asserted (and SWD access is also permitted), the debug tools can use the Config-AP to read device configuration information. When deasserted, a DAPBUS firewall will isolate the AP and prevent access to the Config-AP. 0h = Disable CFG-AP 1h = Enable CFG-AP
3	ETAPEN	R	0h	An active high input. When asserted (and SWD access is also permitted), the debug tools can then access an ET-AP external to the DebugSS lite. When deasserted, a DAPBUS firewall will isolate the AP and prevent access. 0h = Disable ET+ -AP 1h = Enable ET+ -AP
2	DFTAPEN	R	0h	An active high input. When asserted (and SWD access is also permitted), the debug tools can then access the DFT-AP external to the DebugSS lite. When deasserted, a DAPBUS firewall will isolate the AP and prevent access. 0h = Disable DFT-TAP 1h = Enable DFT-TAP
1	SWDPORTEN	R	1h	When asserted, the SW-DP functions normally. When deasserted, the SW-DP effectively disables all external debug access. 0h = Disable SWD port 1h = Enable SWD port

Table 34-51. SPECIAL_AUTH Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
0	SECAPEN	R	1h	An active high input. When asserted (and SWD access is also permitted), the debug tools can use the Security-AP to communicate with security control logic. When deasserted, a DAPBUS firewall will isolate the AP and prevent access to the Security-AP. 0h = Disable SEC-AP 1h = Enable SEC-AP

19 APP_AUTH Register (Offset = 1210h) [Reset = 00000000h]

APP_AUTH is shown in Figure 34-20 and described in Table 34-34.

Return to the Summary Table.

This register is used to control DBGEN, NIDEN, SPIDEN, and SPNIDEN of Application CPU0. DBGEN, NIDEN are further processed by DSW based on Active and Debug IPF ID.

Figure 34-20. APP_AUTH Register

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED				SPNIDEN	SPIDEN	NIDEN	DBGEN
R-0h				R-0h	R-0h	R-0h	R-0h

Table 34-53. APP_AUTH Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	RESERVED	R	0h	
3	SPNIDEN	R	0h	Secure non-invasive debug enable. 0h = Invasive debug disabled 1h = Invasive debug enabled
2	SPIDEN	R	0h	Secure invasive debug enable. 0h = Invasive debug disabled 1h = Invasive debug enabled
1	NIDEN	R	0h	Controls non-invasive debug enable. 0h = Non-invasive debug disabled 1h = Non-invasive debug enabled
0	DBGEN	R	0h	Controls invasive debug enable. 0h = Invasive debug disabled 1h = Invasive debug enabled

Revision History

NOTE: Page numbers for previous revisions may differ from page numbers in the current version.

DATE	REVISION	NOTES
March 2026	*	Initial Release

ADVANCE INFORMATION

IMPORTANT NOTICE AND DISCLAIMER

TI PROVIDES TECHNICAL AND RELIABILITY DATA (INCLUDING DATASHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES “AS IS” AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS AND IMPLIED, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT OF THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for skilled developers designing with TI products. You are solely responsible for (1) selecting the appropriate TI products for your application, (2) designing, validating and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, regulatory or other requirements.

These resources are subject to change without notice. TI grants you permission to use these resources only for development of an application that uses the TI products described in the resource. Other reproduction and display of these resources is prohibited. No license is granted to any other TI intellectual property right or to any third party intellectual property right. TI disclaims responsibility for, and you fully indemnify TI and its representatives against any claims, damages, costs, losses, and liabilities arising out of your use of these resources.

TI’s products are provided subject to TI’s Terms of Sale, TI’s General Quality Guidelines, or other applicable terms available either on ti.com or provided in conjunction with such TI products. TI’s provision of these resources does not expand or otherwise alter TI’s applicable warranties or warranty disclaimers for TI products. Unless TI explicitly designates a product as custom or customer-specified, TI products are standard, catalog, general purpose devices.

TI objects to and rejects any additional or different terms you may propose.

Last updated 10/2025