MKW2xD Reference Manual

Supports: MKW24D512, MKW22D512, MKW21D512, and MKW21D256

Document Number: MKW2xDRM
Rev. 3, 05/2016

Contents

Section number **Title** **Page**

Chapter 1
About This Document

Chapter 2
SiP: Introduction to the MKW2xD SiP

Section number Title Page

Chapter 3
SiP: Signal Multiplexing and Signal Descriptions

Chapter 4
SiP: System Considerations

Chapter 5
Modem: Modes of Operation

Chapter 6
Modem: Interrupts

Chapter 7
Modem: Timer Information

Chapter 8
MCU-Modem SPI Interface

Chapter 9
Modem: SPI Register Descriptions

Section number	Title	Page

Section number Title Page

Chapter 10
MCU: Chip Configuration

Section number	Title	Page

Section number Title Page

Chapter 11
MCU: Memory Map

Chapter 12
MCU: Clock Distribution

Chapter 13
MCU: Reset and Boot

Chapter 14
MCU: Power Management

Chapter 15
MCU: Security

Chapter 16
MCU: Debug

Section number Title Page

Chapter 17
MCU: Signal Multiplexing and Signal Descriptions

Chapter 18
MCU: Port control and interrupts (PORT)

Chapter 19
MCU: System Integration Module (SIM)

Chapter 20 MCU: Reset Control Module (RCM)

Chapter 21 MCU: System Mode Controller (SMC)

Chapter 22
MCU: Power Management Controller (PMC)

Chapter 23
MCU: Low-Leakage Wakeup Unit (LLWU)

Chapter 24
MCU: Miscellaneous Control Module (MCM)

Chapter 25
MCU: Crossbar Switch Lite (AXBS-Lite)

Chapter 26
MCU: Peripheral Bridge (AIPS-Lite)

Chapter 27
MCU: Direct Memory Access Multiplexer (DMAMUX)

Chapter 28
MCU: Direct Memory Access Controller (eDMA)

Section number	Title	Page

Chapter 29
External Watchdog Monitor (EWM)

Chapter 30
MCU: Watchdog Timer (WDOG)

Chapter 31
MCU: Multipurpose Clock Generator (MCG)

Chapter 32
MCU: Oscillator (OSC)

Chapter 33
MCU: RTC Oscillator

Chapter 34
MCU: Flash Memory Controller (FMC)

Chapter 35
MCU: Flash Memory Module (FTFL)

Chapter 36
MCU: EzPort

Chapter 37

MCU: Cyclic Redundancy Check (CRC)

Chapter 38
MCU: Memory-Mapped Cryptographic Acceleration Unit (MMCAU)

Chapter 39
MCU: Random Number Generator Accelerator (RNGA)

Chapter 40
MCU: Analog-to-Digital Converter (ADC)

Chapter 41
MCU: Comparator (CMP)

Chapter 42
MCU: Programmable Delay Block (PDB)

Chapter 43
MCU: FlexTimer Module (FTM)

Section number Title Page

Chapter 44

Section number Title Page

MCU: Periodic Interrupt Timer (PIT)

Chapter 45
MCU: Low-Power Timer (LPTMR)

Chapter 46
MCU: Carrier Modulator Transmitter (CMT)

Chapter 47
MCU: Real Time Clock (RTC)

Chapter 48 MCU: Universal Serial Bus OTG Controller (USBOTG)

Chapter 49
MCU: USB Device Charger Detection Module (USBDCD)

Chapter 50
MCU: USB Voltage Regulator

Chapter 51
MCU: Serial Peripheral Interface (SPI)

Chapter 52
MCU: Inter-Integrated Circuit (I2C)

Chapter 53
MCU: Universal Asynchronous Receiver/Transmitter (UART)

Chapter 54
MCU: Integrated Interchip Sound (I2S) / Synchronous Audio Interface (SAI)

Section number Title Page

Chapter 55
MCU: General-Purpose Input/Output (GPIO)

Chapter 56
MCU: JTAG Controller (JTAGC)

Chapter 1
About This Document

1.1 Overview

1.1.1 Purpose

This document describes the features, architecture, and programming model of the MKW2xD microcontroller family.

1.1.2 Audience

This document is intended for system architects and software application developers who are using (or considering using) the microcontroller in a system.

1.2 Conventions

1.2.1 Numbering systems

The following suffixes identify different numbering systems:

This suffix	Identifies a
b	Binary number. For example, the binary equivalent of the number 5 is written 101b. In some cases, binary numbers are shown with the prefix *0b*.
d	Decimal number. Decimal numbers are followed by this suffix only when the possibility of confusion exists. In general, decimal numbers are shown without a suffix.
h	Hexadecimal number. For example, the hexadecimal equivalent of the number 60 is written 3Ch. In some cases, hexadecimal numbers are shown with the prefix *0x*.

1.2.2 Typographic notation

The following typographic notation is used throughout this document:

Example	Description
placeholder, x	Items in italics are placeholders for information that you provide. Italicized text is also used for the titles of publications and for emphasis. Plain lowercase letters are also used as placeholders for single letters and numbers.
`code`	Fixed-width type indicates text that must be typed exactly as shown. It is used for instruction mnemonics, directives, symbols, subcommands, parameters, and operators. Fixed-width type is also used for example code. Instruction mnemonics and directives in text and tables are shown in all caps; for example, BSR.
SR[SCM]	A mnemonic in brackets represents a named field in a register. This example refers to the Scaling Mode (SCM) field in the Status Register (SR).
REVNO[6:4], XAD[7:0]	Numbers in brackets and separated by a colon represent either: • A subset of a register's named field For example, REVNO[6:4] refers to bits 6–4 that are part of the COREREV field that occupies bits 6–0 of the REVNO register. • A continuous range of individual signals of a bus For example, XAD[7:0] refers to signals 7–0 of the XAD bus.

1.2.3 Special terms

The following terms have special meanings:

Term	Meaning
asserted	Refers to the state of a signal as follows: • An active-high signal is asserted when high (1). • An active-low signal is asserted when low (0).
deasserted	Refers to the state of a signal as follows: • An active-high signal is deasserted when low (0). • An active-low signal is deasserted when high (1). In some cases, deasserted signals are described as *negated*.
reserved	Refers to a memory space, register, or field that is either reserved for future use or for which, when written to, the module or chip behavior is unpredictable.
w1c	Write 1 to clear: Refers to a register bitfield that must be written as 1 to be "cleared."

Chapter 2
SiP: Introduction to the MKW2xD SiP

2.1 Introduction

The MKW2xD family is NXP's latest generation IEEE 802.15.4 platform which incorporates a complete low power 2.4 GHz radio frequency transceiver and a Kinetis family low power, mixed-signal ARM® Cortex™- M4 MCU into a single package. The MKW2xD solution can be used for wireless applications from simple proprietary point-to-point connectivity to a complete IEEE 802.15.4 based network. The combination of the radio and a microcontroller in a small footprint package allows for a cost-effective solution requiring a limited amount of PCB area.

The MKW2xD portfolio meets the higher performance requirements of Thread or ZigBee Pro based applications, especially Smart Energy and Commercial Building automation. This product is a cost-effective solution that matches or exceeds competitive solutions.

The MKW2xD contains an RF transceiver which is an 802.15.4 Standard-compliant radio that operates in the 2.4 GHz ISM frequency band and supports 2.36 to 2.4 GHz Medical Band (MBAN) frequencies. The transceiver includes antenna diversity, 1mW nominal output power, hardware acceleration for dual PAN modes, integrated transmit/receive switch, on-board power supply regulation, and full spread-spectrum encoding and decoding. Additionally, the transceiver includes a PA with internal voltage controlled oscillator (VCO), integrated transmit/receive switch, on-board power supply regulation.

The MKW2xD also contains a Kinetis family low power, mixed-signal ARM® eCortex™- M4 MCU with a full functional set of MCU peripherals (same die as K21P121 with exceptions noted below) and can provide up to 512KB of flash memory and 64KB of RAM. The onboard MCU allows the communications stack and also the application to reside on the same system-in-package (SiP).

The MKW2xD family is organized as follows:

- The MKW24D512 has 512KB of flash and 64KB of RAM. It is intended for use with the NXP fully compliant 802.15.4 MAC, Thread and ZigBee stacks. Custom networks based on the 802.15.4 Standard MAC can be implemented to fit user needs.

The 802.15.4 Standard supports star, mesh and cluster tree topologies as well as beaconed networks. The MKW24D512 supports full speed USB 2.0 at 12 Mb per second.

- The MKW22D512 contains 512K of flash and 64KB of RAM and is also intended for use with the NXP fully compliant 802.15.4 MAC. The MKW22D512 supports full speed USB 2.0 at 12 Mb per second.
- The MKW21D256 with FlexMemory has fixed 256KB of flash and 32KB of RAM and also contains an additional 64KB FlexNVM and 4KB FlexRAM configurable as EEPROM backup, extra p-flash or a combination of both. It is an ideal solution for low cost, proprietary applications requiring wireless point-to-point or star network connectivity.
- The MKW21D512 contains 512KB of flash and 64KB of RAM. Like its counter-part devices, it gives offers a large amount of memory and additional GPIO for instances where both maybe required. Non-USB device.

NOTE

The MKW2xD has the following exceptions to the documented features of the K21P121 CPU:

- Active tamper is not available.
- There are up to 29 I/O pins (25 I/O for USB devices) available depending on device and dedicated peripheral required.
- SPI1 module on the MCU is dedicated for communication to the transceiver and is not available for peripheral use.

This table lists the MKW2xD family devices, their temperature range, package memory options and a short description.

Table 2-1. Ordering Information

Device	Operating Temp Range (TA)	Package	Memory Options	Description
MKW21D256VHA5(R)	'-40° to 105° C	LGA (R: tape and reel)	32 KB RAM, 256 KB Flash, 64 KB FlexMemory	FlexMemory: 64KB FlexNVM / 4KB FlexRAM configurable as EEPROM backup, extra Flash or a combination of both. No USB
MKW21D512VHA5(R)	-40° to 105° C	LGA (R: tape and reel)	64 KB SRAM, 512 KB flash	Offers a large amount of memory and additional GPIO for instances where both may be required. No USB device.

Table continues on the next page...

Table 2-1. Ordering Information (continued)

Device	Operating Temp Range (TA)	Package	Memory Options	Description
MKW22D512VHA5(R)	'-40° to 105° C	LGA (R: tape and reel)	64 KB SRAM, 512 KB flash	Supports full speed USB 2.0. No FlexNVM or FlexRAM.
MKW24D512VHA5(R)	'-40° to 105° C	LGA (R: tape and reel)	64 KB SRAM, 512 KB flash	Supports full-speed USB 2.0. No FlexNVM or FlexRAM.

Target markets include, but are not limited, to the following:

- Smart Energy
  - Meter
  - ESI (Energy Service Interface)
  - IHD (In Home Display)
  - Gateway
  - Appliance
  - PHEV (Plug-in Hybrid Electric Vehicle)
- Building Control and Home Automation
  - Lighting
  - HVAC
  - Security
- Medical / Personal Health Care
  - Patient Monitoring
  - Institutional Care
- Industrial Control (3rd party stacks - (Low PAN, ISA100, Wireless HART)

2.2 Block Diagrams

This figure shows a simplified block diagram of the MKW2xD system in package (SiP).

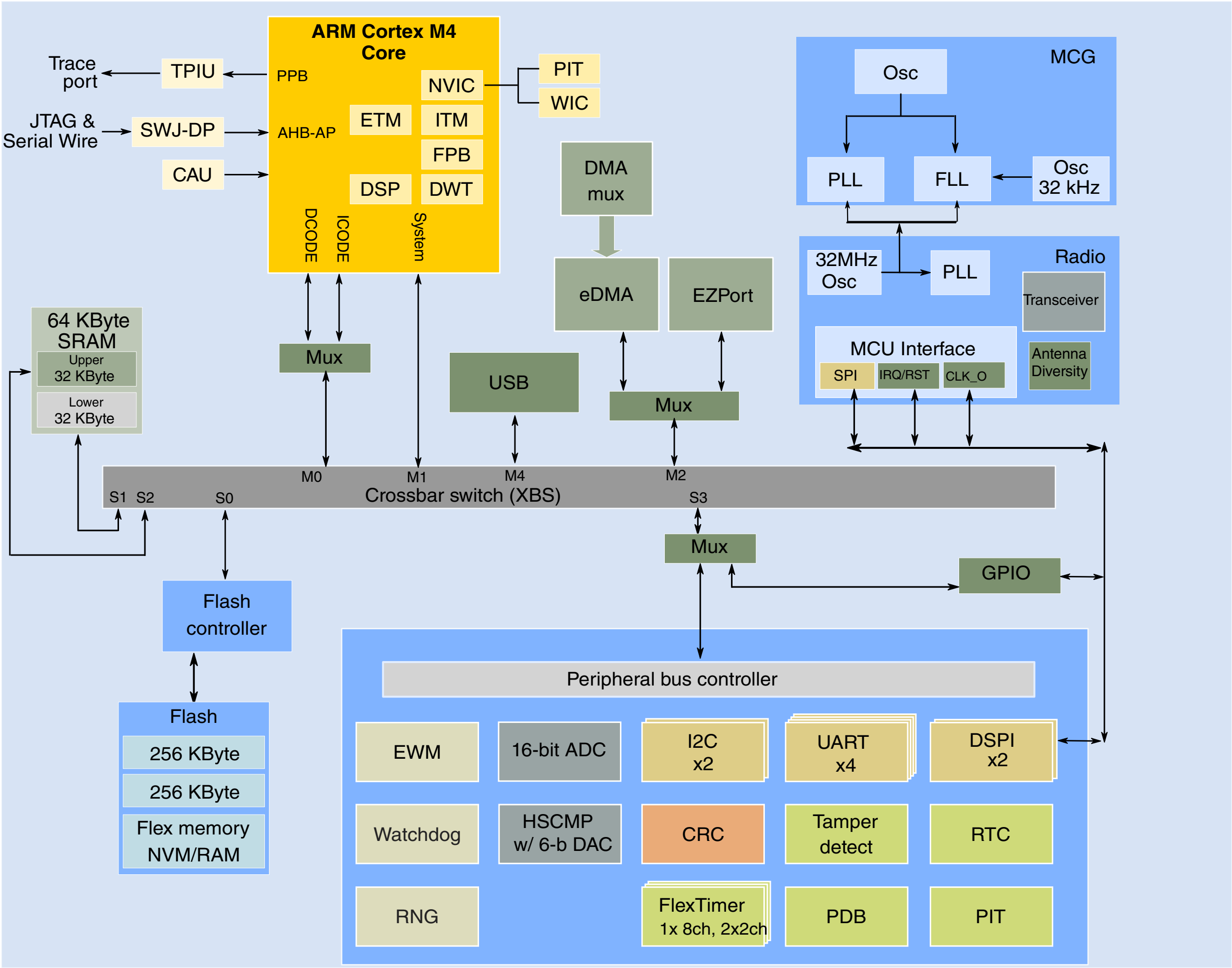


Figure 2-1. SiP Simplified Block Diagram

Figure 2-2 shows a simplified block diagram of the modem in the SiP which is an 802.15.4 Standard compatible transceiver that provides functions required in the physical layer (PHY) and media access control (MAC) specifications.

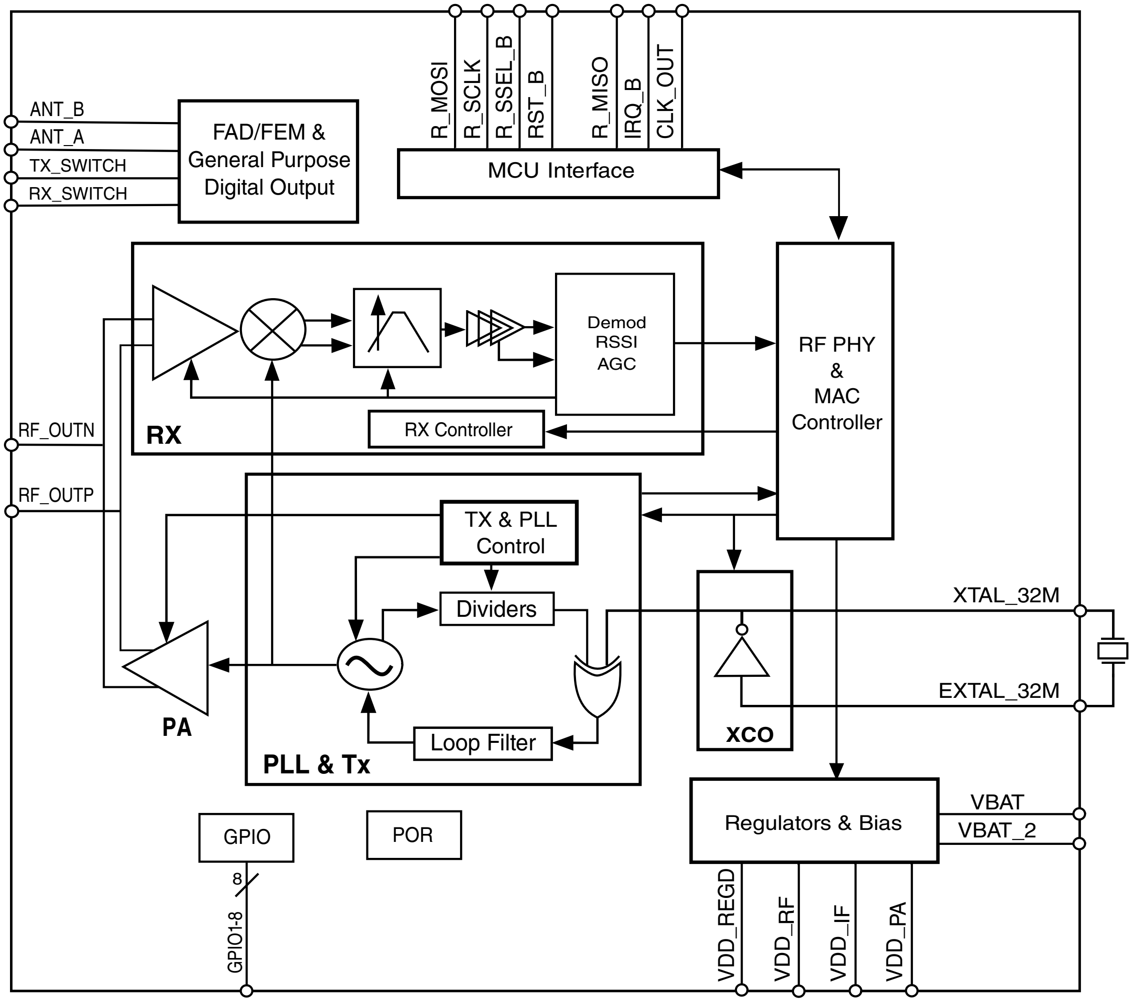


Figure 2-2. Modem Simplified Block Diagram

The modem is used in concert with the MCU. Interface between the devices is accomplished through a 4-wire SPI port and interrupt request line. The media access control (MAC), drivers, and network and application software (as required) reside on the host processor.

2.3 Modem Features Summary

The transceiver has the following features:

- Fully compliant IEEE 802.15.4 Standard 2006 transceiver supports 250 kbps O-QPSK data in 5.0 MHz channels and full spread-spectrum encode and decode, and also extends radio operation to the 2.36 GHz to 2.40 GHz Medical Band (MBAN) frequencies with IEEE 802.15.4j channel, spacing and modulation requirements.

    - 2.4 GHz frequency band of operation (ISM).
    - 250 kbps data rate with O-QPSK modulation in 5.0 MHz channels with direct sequence spread spectrum (DSSS) encode and decode.
    - Operates on one of 16 selectable ISM channels per IEEE 802.15.4 specification.
    - Programmable output power
    - Supports 2.36 GHz to 2.40 GHz Medical Band (MBAN) frequencies with IEEE 802.15.4j channel, spacing and modulation requirements.
- Small RF foot print
    - Differential input/output port used with external balun for single port operation.
    - Supports diversity antenna operation with external front end (FE).
    - Low external component count.
- Hardware acceleration for IEEE® 802.15.4 Standard
    - Complete 802.15.4 onboard modem
    - IEEE 802.15.4 Standard 2006 packet processor/sequencer with receiver frame filtering
    - Random number generator
    - Support for dual PAN ID mode
    - Internal event timer block with four comparators to assist sequencer and provide timer capability
- 32 MHz crystal reference oscillator with onboard trim capability to supplement external load capacitors
- Programmable frequency clock output (CLK_OUT) for use by MCU
- SPI Command Channel interface slave port with burst mode operation
- Interrupt request output (IRQ) - provides interrupt request capability to MCU
- 128-byte RAM data buffer to store 802.15.4 packet contents for transceiver sequences
- Two (2) software programmable GPIOs
- Low power operational modes with single SPI command device wake-up (SPI communication is enabled in LP mode)

- 1.8 V to 3.6 V operating voltage with on chip voltage regulators
- -40C to +105C temperature range

2.4 RF Interface and Usage

The modem RF interface provides a bidirectional, differential port that connects directly to a balun. The balun connects directly to a single-ended antenna and converts that interface to a fully differential, bidirectional, on-chip interface with transmit/receive switch, LNA, and complementary PA outputs. This combination allows for a small footprint and low cost RF solution.

In additional the modem provides dedicated output signals that can be used to control external RF components. These outputs are hardware switched and also support antenna diversity.

2.5 Radio Architecture

The radio structure is built upon the IEEE 802.15.4 Standard packet structure.

2.5.1 Packet Structure

The following figure shows the packet structure

4 bytes	1 byte	1 byte	125 bytes maximum	2 bytes
Preamble	SFD	FLI	Payload Data	FCS

Figure 2-3. MKW2xD Packet Structure

2.5.2 Receive Path Description

The receive path operates in duplex with the transmit mode having an additional feature to operate in a low power run state that can also be considered as a partial power down mode. Architecture is Near Zero IF (NZIF) having front end amplification, one (1) mixed signal down conversion to IF that is filtered, demodulated and digitally processed. The RF Front End (FE) is differential and shares the same off chip matching network with the transmit path.

2.5.3 Transmit Path Description

The modem transmits OQPSK modulation having power and channel selection adjustment per user application. After the channel of operation is determined, coarse and fine tuning is executed within the Frac-N PLL to engage signal lock. After signal lock is established, the modulated buffered signal is then routed to a multi-stage amplifier for transmission. The PA differential outputs share the pins with the front end.

2.6 IEEE 802.15.4 Acceleration Hardware

The 802.15.4 transceiver has several hardware features that reduce the software stack size, off-load functions from the CPU, and improve performance:

- Fully supports 2003 & 2006 versions of the IEEE 802.15 Standard.
- Supports slotted and unslotted modes
- Supports beacon enabled and non-beacon enabled networks
- Onboard 128-byte packet data buffering
- Random number generator
- 802.15.4 Sequence support
    - RX (conditionally followed by TXAck)
    - TX
    - CCA (used for CCA and ED cycles)
    - Tx/Rx (Tx followed by unconditional Rx or RCACK)
    - Continuous CCA
- 802.15.4 Receiver Frame filtering.

2.7 MCU Interface with SPI Overview

The following figure illustrates the microcontroller interface with the modem for the SiP. The typical required signals are:

- 4-wire SPI port - slave mode only

  - Maximum bitrate is 16 MHz for writes and 9 MHz for reads
  - Clock phase and polarity - CPHA=0 and CPOL=0
  - MSB first shifting
  - Supports Register Access and Packet Buffer accesses in bursts of byte transfers
  - Most registers and Packet Buffer accessible with crystal "on" or "off"
- Interrupt request output - active low
- Device asynchronous hardware reset RST_B - active low

The SPI interface provides communication between the MCU and the modem's register set and Packet Buffer. The modem SPI is a slave-only interface; the MCU must drive R_SSEL_B, R_SCLK and R_MOSI. Write and read access to both Direct and Indirect registers is supported, and transfer length can be single-byte or bursts of unlimited length. Write and read access to the Packet buffer can also be single-byte or a burst mode of unlimited length.

The SPI interface is asynchronous to the rest of the device. No relationship between R_SCLK and the modem's internal oscillator is assumed. All synchronization of the SPI interface to modem takes place inside the SPI module and is done for both register writes and reads.

The SPI is capable of operation in all power modes except reset. Operation in Hibernate mode enables most of the registers and includes the complete Packet Buffer. In this state minimal power consumption will be realized especially during the register-initialization phase.

The SPI design features a compact, single-byte control word reducing SPI access latency to a minimum. Most SPI access types require only a single-byte control word with the address embedded in the control word. During control word transfer (the first byte of any SPI access), the contents of the IRQSTS1 register (highest-priority status register) are are always shifted out so that the MCU gets access to IRQSTS1 with the minimum possible latency on every SPI access.

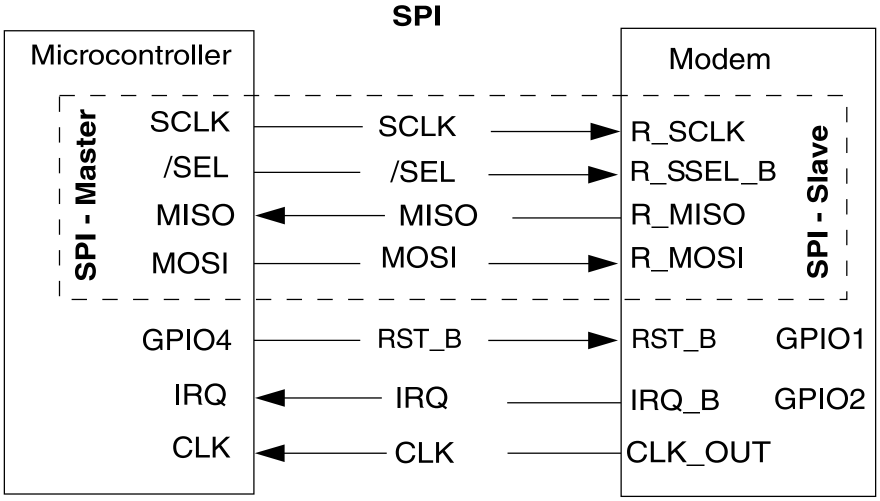


Figure 2-4. Microcontroller to modem interface block diagram

2.7.1 Transceiver Control Overview

The transceiver is controlled by a bank of registers (Direct Register Space) accessed via the SPI interface. The transmit and receive packet data (stored in a 128-byte buffer) is also accessed via the SPI. The onboard registers provide control and status for the entire device.

2.7.1.1 Interrupt Request Overview

The modem has up to 13 individual sources of interrupt request to the MCU. These are all capable of individual control, and are logically OR-combined to drive a single, active low, interrupt request pin (IRQ_B) to the external MCU.

- The IRQ_B pin is configured as actively-driven high by the modem.
- Each interrupt source has its own interrupt status bit in Direct Register space.
- Each interrupt can be individually controlled by an interrupt mask - The IRQ is issued when the mask is cleared to 0.
- There is also a global interrupt mask, TRCV_MSK, which can enable/disable all IRQ_B assertions by programming a single masking bit.
- All status bits use a write-1-to-clear protocol - interrupt status bits are not affected by reads.
- IRQ_B will remain asserted until all active interrupt sources are cleared or masked.

2.7.1.2 Event Timer Overview

The modem features a 24-bit Event Timer that can be used in conjunction with the sequencer to provide protocol control as well as timing interrupts. The Event Timer consists of a continuously running counter and four (4) separate 24-bit comparators:

- The Event Timer counter runs at the 802.15.4 bit rate of 250 kHz (programmable).
- Each comparator has an individual interrupt request capability - the compare status is set when there is a match between the comparator and the timer counter. Each status can be enabled to generate an IRQ.
- In addition, a separate 16-bit T2PRIMECMP comparator is provided, which uses only the *lower 16 bits* of Event Timer, rather than require a full 24-bit compare.

2.8 Clock Output, RF Control, and GPIO Summary

The modem provides a set of I/O pins useful for supplying a system clock to the MCU, controlling external RF LNA/PA or antenna diversity circuitry and GPIO. The following sections discuss these options.

2.8.1 CLK_OUT Reference

The CLK_OUT digital output can be enabled to drive the system clock to the MCU. This provides a highly accurate clock source based on the transceiver reference oscillator. The clock is programmable over a wide range of frequencies divided down from its 32 MHz reference. CLK_OUT can also be used as an external reference of the SiP device at pin 39 (PTA18) but not as a GPIO due to the SiPs internal configuration.

2.8.2 RF Control Signals

The modem provides four dedicated signals for control of external RF components. These signals designated as ANT_A, ANT_B, RX_SWITCH, and TX_SWITCH can be enabled to control external amplifiers, antenna switches, and other modules. When enabled they are switched via an internal hardware state machine. Typical uses include:

- Antenna diversity

- External PA
- External LNA
- T/R switching

2.8.3 Antenna Diversity

To improve the reliability of RF connectivity to long range applications, Antenna Diversity feature is supported without utilizing the MCU through use of four dedicated control pins by direct register antenna selection. The digital regulator supplies bias to analog switches for control of external pin diodes or external PA/LNA. These switches are programmable to sink and source 1mA, 2mA, 4mA and 8mA currents or can operate in a high impedance mode.

2.8.4 General Purpose Input Output (GPIO)

In addition to the MCU supported GPIOs, two modem GPIOs are also provided for general use; GPIO1 and GPIO2. Features for these pins include:

- Programmable output drive strength
- Programmable output slew rate
- Hi-Z mode
- Programmable as outputs or inputs (default)
- No IRQ capability

2.9 Modem Operational Modes

The modem has six operating modes which include:

- Reset / Power-down
- Low Power (LP) / Hibernate
- Doze (low power with reference oscillator active)
- Idle

- Receive
- Transmit

The following table describes these modes:

Table 2-2. Modem Mode Definitions and Transition Times

Mode	Definition	Transition Time To or From Idle
Off	All modem functions Off, Leakage only. $\overline{\text{RST}}$ asserted. Digital outputs are tri-stated including IRQ	500 us
Hibernate	Crystal Reference Oscillator Off. Modem responds to SPI activity.	250 us
Doze	Crystal reference oscillator ON but CLK_OUT output available only if selected. Digital regulator in Low Power mode.	< 1 us[1]
Reset	Crystal reference oscillator ON, enable CLK_OUT output at 4 MHz and 32.787 kHz.	376 us
Receive	Crystal reference oscillator ON. Receiver ON.	144 us
Transmit	Crystal reference oscillator ON. Transmitter ON.	144 us

[1] At 9 MHz, the doze to idle transistion time will be less than 2 SPI writes (1.8 us).

2.10 Modem Features

All devices within the family feature the following:

Table 2-3. Common features among all devices

Operating characteristics	• Voltage range 1.8V - 3.6V • Flash memory programming down to 1.8V • Temperature range (T_A) -40 to 125°C • Flexible modes of operation
Core features	• Next generation 32-bit ARM Cortex-M4 core • Supports DSP instructions • Nested vectored interrupt controller (NVIC) • Asynchronous wake-up interrupt controller (AWIC) • Debug & trace capability • 2-pin serial wire debug (SWD) • IEEE 1149.1 Joint Test Action Group (JTAG) • IEEE 1149.7 compact JTAG (cJTAG) • Trace port interface unit (TPIU) • Flash patch and breakpoint (FPB) • Data watchpoint and trace (DWT) • Instrumentation trace macrocell (ITM) • Enhanced Trace Macrocell (ETM)
Radio features	• 2.4 GHz frequency band of operation • 250 kbps data rate with O-QPSK modulation in 5.0 MHz channels with direct sequence spread-spectrum (DSSS) encode and decode

Table continues on the next page...

Table 2-3. Common features among all devices (continued)

	• Operates on one of 16 selectable channels per IEEE 802.15.4 specification • Programmable output power • Supports 2.36 to 2.4 GHz Medical Band (MBAN) frequencies with same modulation as IEEE 802.15.4 2.4GHz with 1MHz channel spacing • Hardware acceleration for IEEE® 802.15.4 2006 packet processing • 32 MHz crystal reference oscillator with trim capability • GPIO for Antenna Diversity control • Small RF footprint • Differential input/output port used with external balun • Integrated transmit/receive switch • Supports single ended and diversity antenna options • Low external component count • Supports external PA and LNA
System and power management	• Software and hardware watchdog with external monitor pin • DMA controller with 16 channels • Low-leakage wake-up unit (LLWU) • Power management controller with 10 different power modes • Non-maskable interrupt (NMI) • 128-bit unique identification (ID) number per chip
Clocks	• Multi-purpose clock generator • PLL and FLL operation • Internal reference clocks (32kHz or 2MHz) • Three separate crystal oscillators • 3 MHz to 32MHz crystal oscillator for MCU • 32 kHz to 40kHz crystal oscillator for MCU or RTC • 32 MHz crystal oscillator for Radio • Internal 1 kHz low power oscillator • DC to 50 MHz external square wave input clock
Memories and Memory Interfaces	• FlexMemory consisting of FlexNVM (non-volatile flash memory that can execute program code, store data, or backup EEPROM data) or FlexRAM (RAM memory that can be used as traditional RAM or as high-endurance EEPROM storage and also accelerates flash programming) • Flash security and protection features • Serial flash programming interface (EzPort)
Security and integrity	• Cyclic redundancy check (CRC) • Tamper detect • Hardware encryption
Analog	• 16-bit SAR ADC
Timers	• 1x8ch motor control/general purpose/PWM flexible timer (FTM) • x2ch quadrature decoder/general purpose/PWM flexible timer (FTM) • Carrier modulator timer (CMT) • Programmable delay block (PDB)

Table continues on the next page...

Table 2-3. Common features among all devices (continued)

	• 1x4ch programmable interrupt timer (PIT) • Low-power timer (LPT)
Communications	• SPI • I^2C with SMBUS support • UART (w/ ISO7816, IrDA and hardware flow control)
Human-machine interface	• GPIO with pin interrupt support, DMA request capability, digital glitch filter, and other pin control options • Capacitive touch sensing inputs

2.10.1 Memory and package options

The following table summarizes the memory and package options for the MKW2xD family. All devices which share a common package are pin-for-pin compatible.

Table 2-4. MKW2xD family summary

	Memory					Part/Package
Performance (MHz)	Total NVM	Flash (KB)	FlexNVM (KB)	SRAM (KB)	EEPROM/ FlexRAM (KB)	63 LGA (8x8)
50	256	256	64	32	4	MKW21D256
50	512	512	—	64	—	MKW21D512
50	512	512	—	64	—	MKW22D512
50	512	512	—	64	—	MKW24D512

2.10.2 FlexMemory for the MWK21D256

The FlexMemory technology provides an extremely versatile and powerful solution for designers seeking on-chip EEPROM and/or additional program or data flash memory. As easy and as fast as SRAM, it requires no user or system intervention to complete programming and erase functions when used as high endurance byte-write/byte-erase EEPROM. EEPROM array size can also be configured for improved endurance to suit application requirements. FlexMemory can also provide additional flash memory (FlexNVM) for data or program storage in parallel with the main program flash.

The key features of FlexMemory include:

- Configurability for designer:
  - EEPROM array size and number of write/erase cycles
  - Program or data flash size
  - Up to 64 KB FlexNVM, up to 4 KB FlexRAM

- EEPROM endurance of 10M write/erase cycles possible over full voltage and temperature range
- Seamless EEPROM read/write operations: simply write or read a memory address
- High-speed byte, 16-bit, and 32-bit write/erase operations to EEPROM
- Eliminates the costs associated with external EEPROM ICs, and the software headaches and resource (CPU/flash/RAM) impact of EEPROM emulation schemes
- Storage for large data tables or bootloader
- Read-while-write operation with main program flash memory

2.10.2.1 Programmable Trade-Off

FlexMemory lets you fully configure the way FlexNVM and FlexRAM blocks are used to provide the best balance of memory resources for their application.

The user can configure several parameters, including EEPROM size, endurance, write size, and the size of additional program/data flash.

In addition to this flexibility, FlexMemory provides superior EEPROM performance, endurance, and low-voltage operation when compared to traditional EEPROM solutions.

- Enhanced EEPROM — Combines FlexRAM and FlexNVM to create byte-write/erase, high-speed, and high-endurance EEPROM
- FlexNVM — Can be used as:
    - part of the EEPROM configuration,
    - additional program or data flash, or
    - a combination of the above. For example, a portion can be used as flash while the rest is used for enhanced EEPROM backup.
- FlexRAM — Can be used as part of the EEPROM configuration or as additional system RAM

2.10.2.2 Use Case Example

The MCU has 256 KB program flash, 32 KB SRAM, and FlexMemory has 64 KB FlexNVM and 4 KB FlexRAM.

The application requires 8 KB additional program flash for a bootloader and 256 bytes of high-endurance EEPROM. The user allocates 8 KB of FlexNVM for the additional program flash and the remaining 56 KB for EEPROM backup.

The user defines 256 bytes of EEPROM size from the FlexRAM. In this example, the EEPROM endurance results in a minimum of 1.08M write/erase cycles.

2.10.3 External PA and LNA

The modem supports features to add either an external PA, LNA or RF switch which can extend the range or add antenna diversity to the target application. The following hardware features aid in the configuration of an FEM:

- Four dedicated programmable pins to sink and source 1mA, 2mA, 4mA and 8mA currents to control FEM (Front End Module) features such as a PA, LNA and RF switches for antenna diversity, etc.
- Balun used to optimize performance and provide a differential RX/TX output to single ended feature. Modem RX/TX outputs are differential “I” and “Q” and can be utilized in that format if desired.

2.11 MCU Introduction

2.11.1 Module Functional Categories

The modules on this device are grouped into functional categories. The following sections describe the modules assigned to each category in more detail.

Table 2-5. Module functional categories

Module category	Description
ARM Cortex-M4 core	• 32-bit MCU core from ARM’s Cortex-M class adding DSP instructions, 1.25 DMIPS/MHz, based on ARMv7 architecture
System	• System integration module • Power management and mode controllers • Multiple power modes available based on run, wait, stop, and power-down modes • Low-leakage wakeup unit • Miscellaneous control module • Crossbar switch • Peripheral bridge • Direct memory access (DMA) controller with multiplexer to increase available DMA requests • External watchdog monitor • Watchdog
Memories	• Internal memories include: • Program flash memory • On devices with FlexMemory: FlexMemory • FlexNVM • FlexRAM • On devices with program flash only: Programming acceleration RAM • SRAM • Serial programming interface: EzPort

Table continues on the next page...

Table 2-5. Module functional categories (continued)

Module category	Description
Clocks	• Multiple clock generation options available from internally- and externally-generated clocks • System oscillator to provide clock source for the MCU • RTC oscillator to provide clock source for the RTC
Security	• Cyclic Redundancy Check module for error detection • Hardware encryption, along with a random number generator • Tamper detect and secure storage
Analog	• High speed analog-to-digital converter • Comparator • Bandgap voltage reference
Timers	• Programmable delay block • FlexTimers • Periodic interrupt timer • Low power timer • Carrier modulator transmitter • Independent real time clock
Communications	• USB OTG controller with built-in FS/LS transceiver • USB device charger detect • USB voltage regulator • Serial peripheral interface • Inter-integrated circuit (I^2C) • UART • Integrated interchip sound (I^2S)
Human-Machine Interfaces (HMI)	• General purpose input/output controller

2.11.1.1 ARM® Cortex™-M4 Core Modules

The following core modules are available on this device.

Table 2-6. Core modules

Module	Description
ARM Cortex-M4	The ARM Cortex-M4 is the newest member of the Cortex M Series of processors targeting microcontroller cores focused on very cost sensitive, deterministic, interrupt driven environments. The Cortex M4 processor is based on the ARMv7 Architecture and Thumb®-2 ISA and is upward compatible with the Cortex M3, Cortex M1, and Cortex M0 architectures. Cortex M4 improvements include an ARMv7 Thumb-2 DSP (ported from the ARMv7-A/R profile architectures) providing 32-bit instructions with SIMD (single instruction multiple data) DSP style multiply-accumulates and saturating arithmetic.
NVIC	The ARMv7-M exception model and nested-vectored interrupt controller (NVIC) implement a relocatable vector table supporting many external interrupts, a single non-maskable interrupt (NMI), and priority levels.

Table continues on the next page...

Table 2-6. Core modules (continued)

Module	Description
	The NVIC replaces shadow registers with equivalent system and simplified programmability. The NVIC contains the address of the function to execute for a particular handler. The address is fetched via the instruction port allowing parallel register stacking and look-up. The first sixteen entries are allocated to ARM internal sources with the others mapping to MCU-defined interrupts.
AWIC	The primary function of the Asynchronous Wake-up Interrupt Controller (AWIC) is to detect asynchronous wake-up events in stop modes and signal to clock control logic to resume system clocking. After clock restart, the NVIC observes the pending interrupt and performs the normal interrupt or event processing.
Debug interfaces	Most of this device's debug is based on the ARM CoreSight™ architecture. Four debug interfaces are supported: • IEEE 1149.1 JTAG • IEEE 1149.7 JTAG (cJTAG) • Serial Wire Debug (SWD) • ARM Real-Time Trace Interface

2.11.1.2 System Modules

The following system modules are available on this device.

Table 2-7. System modules

Module	Description
System integration module (SIM)	The SIM includes integration logic and several module configuration settings.
System mode controller	The SMC provides control and protection on entry and exit to each power mode, control for the Power management controller (PMC), and reset entry and exit for the complete MCU.
Power management controller (PMC)	The PMC provides the user with multiple power options. Ten different modes are supported that allow the user to optimize power consumption for the level of functionality needed. Includes power-on-reset (POR) and integrated low voltage detect (LVD) with reset (brownout) capability and selectable LVD trip points.
Low-leakage wakeup unit (LLWU)	The LLWU module allows the device to wake from low leakage power modes (LLS and VLLS) through various internal peripheral and external pin sources.
Miscellaneous control module (MCM)	The MCM includes integration logic
Crossbar switch (XBS)	The XBS connects bus masters and bus slaves, allowing all bus masters to access different bus slaves simultaneously and providing arbitration among the bus masters when they access the same slave.
Peripheral bridges	The peripheral bridge converts the crossbar switch interface to an interface to access a majority of peripherals on the device.
DMA multiplexer (DMAMUX)	The DMA multiplexer selects from many DMA requests down to a smaller number for the DMA controller.
Direct memory access (DMA) controller	The DMA controller provides programmable channels with transfer control descriptors for data movement via dual-address transfers for 8-, 16-, 32- and 128-bit data values.

Table continues on the next page...

Table 2-7. System modules (continued)

Module	Description
External watchdog monitor (EWM)	The EWM is a redundant mechanism to the software watchdog module that monitors both internal and external system operation for fail conditions.
Software watchdog (WDOG)	The WDOG monitors internal system operation and forces a reset in case of failure. It can run from an independent 1 KHz low power oscillator with a programmable refresh window to detect deviations in program flow or system frequency.

2.11.1.3 Memories and Memory Interfaces

The following memories and memory interfaces are available on this device.

Table 2-8. Memories and memory interfaces

Module	Description
Flash memory	• Program flash memory — non-volatile flash memory that can execute program code • FlexMemory — encompasses the following memory types: • For devices with FlexNVM: FlexNVM — Non-volatile flash memory that can execute program code, store data, or backup EEPROM data • For devices with FlexNVM: FlexRAM — RAM memory that can be used as traditional RAM or as high-endurance EEPROM storage, and also accelerates flash programming • For devices with only program flash memory: Programming acceleration RAM — RAM memory that accelerates flash programming
Flash memory controller	Manages the interface between the device and the on-chip flash memory.
SRAM	Internal system RAM. Partial SRAM kept powered in VLLS2 low leakage mode.
SRAM controller	Manages simultaneous accesses to system RAM by multiple master peripherals and core.
System register file	32-byte register file that is accessible during all power modes and is powered by VDD.
VBAT register file	32-byte register file that is accessible during all power modes and is powered by VBAT.
Serial programming interface (EzPort)	Same serial interface as, and subset of, the command set used by industry-standard SPI flash memories. Provides the ability to read, erase, and program flash memory and reset command to boot the system after flash programming.

2.11.1.4 Clocks

The following clock modules are available on this device.

Table 2-9. Clock modules

Module	Description
Multi-clock generator (MCG)	The MCG provides several clock sources for the MCU that include: • Phase-locked loop (PLL) — Voltage-controlled oscillator (VCO) • Frequency-locked loop (FLL) — Digitally-controlled oscillator (DCO) • Internal reference clocks — Can be used as a clock source for other on-chip peripherals
System oscillator	The system oscillator, in conjunction with an external crystal or resonator, generates a reference clock for the MCU.
Real-time clock oscillator	The RTC oscillator has an independent power supply and supports a 32 kHz crystal oscillator to feed the RTC clock. Optionally, the RTC oscillator can replace the system oscillator as the main oscillator source.

2.11.1.5 Security and Integrity modules

The following security and integrity modules are available on this device:

Table 2-10. Security and integrity modules

Module	Description
Cryptographic acceleration unit (CAU)	Supports DES, 3DES, AES, MD5, SHA-1, and SHA-256 algorithms via simple C calls to optimized security functions provided by NXP.
Random number generator (RNG)	Supports the key generation algorithm defined in the Digital Signature Standard.
Cyclic Redundancy Check (CRC)	Hardware CRC generator circuit using 16/32-bit shift register. Error detection for all single, double, odd, and most multi-bit errors, programmable initial seed value, and optional feature to transpose input data and CRC result via transpose register.
DryIce (tamper detection and secure storage)	The DryIce module includes a 32-byte secure memory that is asynchronously erased on any tamper detect. In addition, it can optionally force a System Reset and/or invalidate the Real Time Clock.

2.11.1.6 Analog modules

The following analog modules are available on this device:

Table 2-11. Analog modules

Module	Description
16-bit analog-to-digital converters (ADC)	16-bit successive-approximation ADC
Analog comparators	Compares two analog input voltages across the full range of the supply voltage.
6-bit digital-to-analog converters (DAC)	64-tap resistor ladder network which provides a selectable voltage reference for applications where voltage reference is needed.

2.11.1.7 Timer modules

The following timer modules are available on this device:

Table 2-12. Timer modules

Module	Description
Programmable delay block (PDB)	• 16-bit resolution • 3-bit prescaler • Positive transition of trigger event signal initiates the counter • Supports two triggered delay output signals, each with an independently-controlled delay from the trigger event • Outputs can be OR'd together to schedule two conversions from one input trigger event and can schedule precise edge placement for a pulsed output. This feature is used to generate the control signal for the CMP windowing feature and output to a package pin if needed for applications, such as critical conductive mode power factor correction. • Continuous-pulse output or single-shot mode supported, each output is independently enabled, with possible trigger events • Supports bypass mode • Supports DMA
Flexible timer modules (FTM)	• Selectable FTM source clock, programmable prescaler • 16-bit counter supporting free-running or initial/final value, and counting is up or up-down • Input capture, output compare, and edge-aligned and center-aligned PWM modes • Operation of FTM channels as pairs with equal outputs, pairs with complimentary outputs, or independent channels with independent outputs • Deadtime insertion is available for each complementary pair • Generation of hardware triggers • Software control of PWM outputs • Up to 4 fault inputs for global fault control • Configurable channel polarity • Programmable interrupt on input capture, reference compare, overflowed counter, or detected fault condition • Quadrature decoder with input filters, relative position counting, and interrupt on position count or capture of position count on external event • DMA support for FTM events
Periodic interrupt timers (PIT)	• Four general purpose interrupt timers • Interrupt timers for triggering ADC conversions • 32-bit counter resolution • DMA support
Low-power timer (LPTimer)	• Selectable clock for prescaler/glitch filter of 1 kHz (internal LPO), 32.768 kHz (external crystal), or internal reference clock • Configurable Glitch Filter or Prescaler with 16-bit counter • 16-bit time or pulse counter with compare • Interrupt generated on Timer Compare • Hardware trigger generated on Timer Compare
Carrier modulator timer (CMT)	• Four CMT modes of operation: • Time with independent control of high and low times • Baseband • Frequency shift key (FSK) • Direct software control of CMT_IRO pin • Extended space operation in time, baseband, and FSK modes

Table continues on the next page...

Table 2-12. Timer modules (continued)

Module	Description
	• Selectable input clock divider • Interrupt on end of cycle with the ability to disable CMT_IRO pin and use as timer interrupt • DMA support
Real-time clock (RTC)	• Independent power supply, POR, and 32 kHz Crystal Oscillator • 32-bit seconds counter with 32-bit Alarm • 16-bit Prescaler with compensation that can correct errors between 0.12 ppm and 3906 ppm

2.11.1.8 Communication interfaces

The following communication interfaces are available on this device:

Table 2-13. Communication modules

Module	Description
USB OTG (low-/full-speed)	USB 2.0 compliant module with support for host, device, and On-The-Go modes. Includes an on-chip transceiver for full and low speeds.
USB Device Charger Detect (USBDCD)	The USBDCD monitors the USB data lines to detect a smart charger meeting the USB Battery Charging Specification Rev1.1. This information allows the MCU to better manage the battery charging IC in a portable device.
USB voltage regulator	Up to 5 V regulator input typically provided by USB VBUS power with 3.3 V regulated output that powers on-chip USB subsystem, capable of sourcing 120 mA to external board components.
Serial peripheral interface (SPI)	Synchronous serial bus for communication to an external device
Inter-integrated circuit (I2C)	Allows communication between a number of devices. Also supports the System Management Bus (SMBus) Specification, version 2.
Universal asynchronous receiver/ transmitters (UART)	Asynchronous serial bus communication interface with programmable 8- or 9-bit data format and support of ISO 7816 smart card interface
I2S	The I^2S is a full-duplex, serial port that allows the chip to communicate with a variety of serial devices, such as standard codecs, digital signal processors (DSPs), microprocessors, peripherals, and audio codecs that implement the inter-IC sound bus (I^2S) and the Intel® AC97 standards

2.11.1.9 Human-machine interfaces

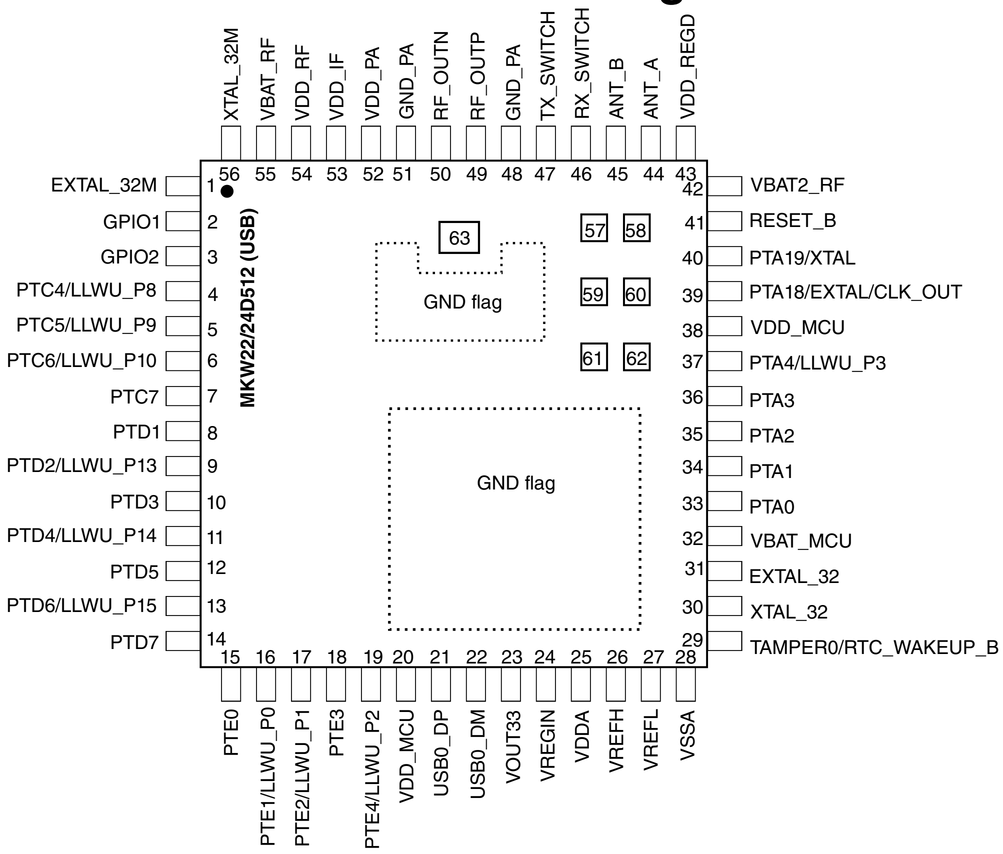
The following human-machine interfaces (HMI) are available on this device:

Table 2-14. HMI modules

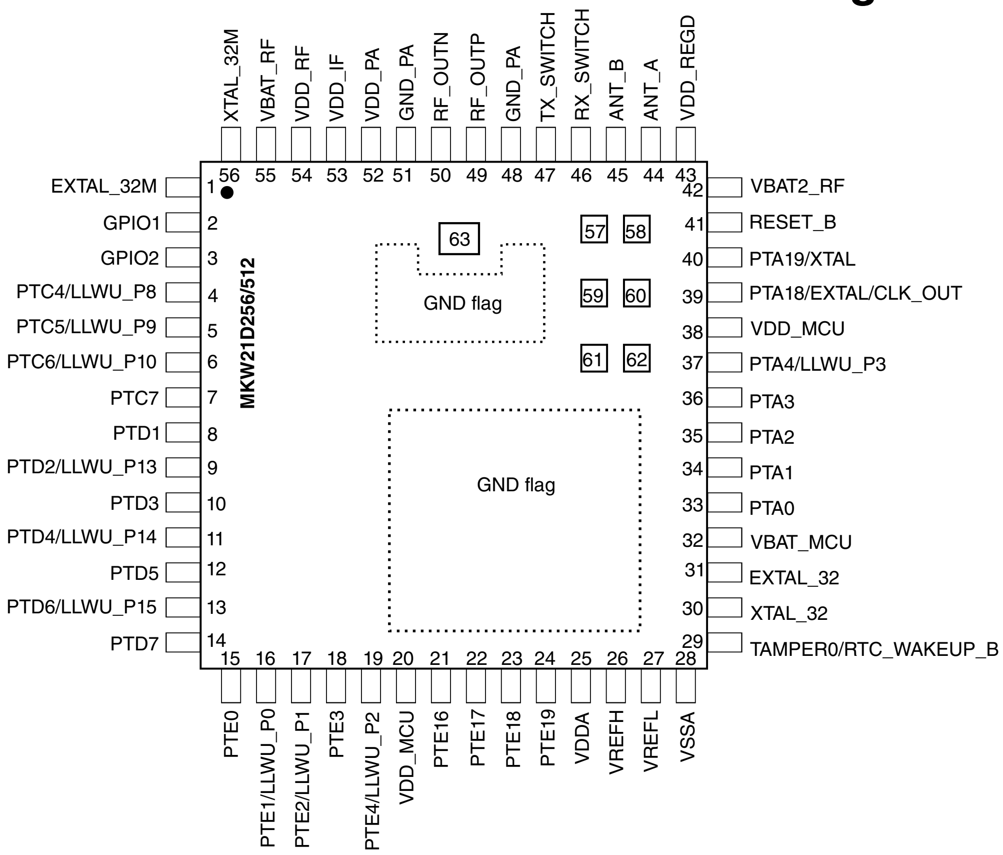
Module	Description
General purpose input/output (GPIO)	All general purpose input or output (GPIO) pins are capable of interrupt and DMA request generation. All GPIO pins have 3.3 V tolerance.

Chapter 3
SiP: Signal Multiplexing and Signal Descriptions

3.1 MKW22/24D512V Pin Assignment



3.2 MKW21D256/MKW21D512 Pin Assignment



3.3 MKW2xD Pins

Table 3-1. MKW2xD Pins

Typical feature	MKW22/24D512V (USB)	MKW21D	Pin Name	Default	ALT0	ALT1	ALT2	ALT3	ALT4	ALT5	ALT6	ALT7
RADIO	1	1	EXTAL_32M	EXTAL_32M	—	—	—	—	—	—	—	—
RADIO	2	2	GPIO1	GPIO1	—	—	—	—	—	—	—	—
RADIO	3	3	GPIO2	GPIO2	—	—	—	—	—	—	—	—
SPI0	4	4	PTC4/LLWU_P8	DISABLED	—	PTC4/LLWU_P8	SPI0_PCS0	UART1_TX	FTM0_CH3	—	CMP1_OUT	—
SPI0	5	5	PTC5/LLWU_P9	DISABLED	—	PTC5/LLWU_P9	SPI0_SCK	LPTMR0_ALT2	I2S0_RXD0	—	CMP0_OUT	—

Table continues on the next page...

Table 3-1. MKW2xD Pins (continued)

Typical feature	MKW22/24D512V (USB)	MKW21D	Pin Name	Default	ALT0	ALT1	ALT2	ALT3	ALT4	ALT5	ALT6	ALT7
SPI0	6	6	PTC6/LLWU_P10	CMP0_IN0	CMP0_IN0	PTC6/LLWU_P10	SPI0_SOUT	PDB0_EXTRG	I2S0_RX_BCLK	—	I2S0_MCLK	—
SPI0	7	7	PTC7	CMP0_IN1	CMP0_IN1	PTC7	SPI0_SIN	USB_SOF_OUT	I2S0_RX_FS	—	—	—
SPI0	8	8	PTD1	ADC0_SE5b	ADC0_SE5b	PTD1	SPI0_SCK	UART2_CTS_b	—	—	—	—
I2C	9	9	PTD2/LLWU_P13	DISABLED	—	PTD2/LLWU_P13	SPI0_SOUT	UART2_RX	I2C0_SCL	—	—	—
I2C	10	10	PTD3	DISABLED	—	PTD3	SPI0_SIN	UART2_TX	I2C0_SDA	—	—	—
UART0	11	11	PTD4/LLWU_P14	ADC0_SE21	ADC0_SE21	PTD4/LLWU_P14	SPI0_PCS1	UART0_RTS_b	FTM0_CH4	—	EWM_IN	—
UART0	12	12	PTD5	ADC0_SE6b	ADC0_SE6b	PTD5	SPI0_PCS2	UART0_CTS_b/UART0_COL_b	FTM0_CH5	—	EWM_OUT_b	—
UART0	13	13	PTD6/LLWU_P15	ADC0_SE7b	ADC0_SE7b	PTD6/LLWU_P15	SPI0_PCS3	UART0_RX	FTM0_CH6	—	FTM0_FLT0	—
UART0	14	14	PTD7	ADC0_SE22	ADC0_SE22	PTD7	CMT_IRO	UART0_TX	FTM0_CH7	—	FTM0_FLT1	—
TRACE, UART1, I2C	15	15	PTE0	ADC0_SE10	ADC0_SE10	PTE0	SPI1_PCS1	UART1_TX	—	TRACE_CLKOUT	I2C1_SDA	RTC_CLKOUT
TRACE,	16	16	PTE1/LLWU_P0	ADC0_SE11	ADC0_SE11	PTE1/LLWU_P0	SPI1_SOUT	UART1_RX	—	TRACE_D3	I2C1_SCL	SPI1_SIN
TRACE,	17	17	PTE2/LLWU_P1	ADC0_DP1	ADC0_DP1	PTE2/LLWU_P1	SPI1_SCK	UART1_CTS_b	—	TRACE_D2	—	—
TRACE,	18	18	PTE3	ADC0_DM1	ADC0_DM1	PTE3	SPI1_SIN	UART1_RTS_b	—	TRACE_D1	—	SPI1_SOUT
TRACE,	19	19	PTE4/LLWU_P2	DISABLED	—	PTE4/LLWU_P2	SPI1_PCS0	—	—	TRACE_D0	—	—
—	20	20	VDD_MCU	VDD_MCU	VDD_MCU	—	—	—	—	—	—	—
—	—	21	PTE16	ADC0_SE4a	ADC0_SE4a	PTE16	SPI0_PCS0	UART2_TX	FTM_CLKIN0	—	FTM0_FLT3	—
—	—	22	PTE17	ADC0_SE5a	ADC0_SE5a	PTE17	SPI0_SCK	UART2_RX	FTM_CLKIN1	—	LPTMR0_ALT3	—

Table continues on the next page...

Table 3-1. MKW2xD Pins (continued)

Typical feature	MKW22/24D512V (USB)	MKW21D	Pin Name	Default	ALT0	ALT1	ALT2	ALT3	ALT4	ALT5	ALT6	ALT7
—	—	23	PTE18	ADC0_SE6a	ADC0_SE6a	PTE18	SPI0_SOUT	UART2_CTS_b	I2C0_SDA	—	—	—
—	—	24	PTE19	ADC0_SE7a	ADC0_SE7a	PTE19	SPI0_SIN	UART2_RTS_b	I2C0_SCL	—	—	—
USB	21	—	USB0_DP	USB0_DP	USB0_DP	—	—	—	—	—	—	—
USB	22	—	USB0_DM	USB0_DM	USB0_DM	—	—	—	—	—	—	—
USB	23	—	VOUT33	VOUT33	VOUT33	—	—	—	—	—	—	—
USB	24	—	VREGIN	VREGIN	VREGIN	—	—	—	—	—	—	—
Analog Power	25	25	VDDA	VDDA	VDDA	—	—	—	—	—	—	—
Analog Power	26	26	VREFH	VREFH	VREFH	—	—	—	—	—	—	—
Analog Power	27	27	VREFL	VREFL	VREFL	—	—	—	—	—	—	—
Analog Power	28	28	VSSA	VSSA	VSSA	—	—	—	—	—	—	—
Tamper	29	29	TAMPER0/RTC_WAKEUP_B	TAMPER0/RTC_WAKEUP_B	TAMPER0/RTC_WAKEUP_B	—	—	—	—	—	—	—
VBAT, 32KHz OSC	30	30	XTAL32	XTAL32	XTAL32	—	—	—	—	—	—	—
VBAT, 32KHz OSC	31	31	EXTAL32	EXTAL32	EXTAL32	—	—	—	—	—	—	—
VBAT, 32KHz OSC	32	32	VBAT_MCU	VBAT_MCU	VBAT_MCU	—	—	—	—	—	—	—
JTAG, Timer	33	33	PTA0	JTAG_TCLK/SWD_CLK/EZP_CLK	—	PTA0	UART0_CTS_b/UART0_COL_b	FTM0_CH5	—	—	—	JTAG_TCLK/SWD_CLK
JTAG, Timer	34	34	PTA1	JTAG_TDI/EZP_DI	—	PTA1	UART0_RX	FTM0_CH6	—	—	—	JTAG_TDI

Table continues on the next page...

Table 3-1. MKW2xD Pins (continued)

Typical feature	MKW22/24D512V (USB)	MKW21D	Pin Name	Default	ALT0	ALT1	ALT2	ALT3	ALT4	ALT5	ALT6	ALT7
JTAG, Timer	35	35	PTA2	JTAG_TDO/ TRACE_SWO/ EZP_DO	—	PTA2	UART0_TX	FTM0_CH7	—	—	—	JTAG_TDO/ TRACE_SWO
JTAG, Timer	36	36	PTA3	JTAG_TMS/ SWD_DIO	—	PTA3	UART0_RTS_b	FTM0_CH0	—	—	—	JTAG_TMS/ SWD_DIO
NMI	37	37	PTA4/ LLWU_P3	NMI_b/ EZP_CS_b	—	PTA4/ LLWU_P3	—	FTM0_CH1	—	—	—	NMI_b
—	38	38	VDD_MCU	VDD_MCU	VDD_MCU	—	—	—	—	—	—	—
MCU XTAL/ Radio	39	39	PTA18/ CLK_OUT	EXTAL0/ CLK_OUT	EXTAL0	PTA18	—	FTM0_FLT2	FTM_CLKIN0	—	—	—
MCU XTAL	40	40	PTA19	XTAL0	XTAL0	PTA19	—	FTM1_FLT0	FTM_CLKIN1	—	LPTMR0_ALT1	—
RESET	41	41	RESET_b	RESET_b	RESET_b	—	—	—	—	—	—	—
RADIO	42	42	VBAT2_RF	VBAT2_RF	—	—	—	—	—	—	—	—
RADIO	43	43	VDD_REGD	VDD_REGD	—	—	—	—	—	—	—	—
RADIO	44	44	ANT_A	ANT_A	—	—	—	—	—	—	—	—
RADIO	45	45	ANT_B	ANT_B	—	—	—	—	—	—	—	—
RADIO	46	46	RX_SWITCH	RX_SWITCH	—	—	—	—	—	—	—	—
RADIO	47	47	TX_SWITCH	TX_SWITCH	—	—	—	—	—	—	—	—
RADIO	48	48	GND_PA	GND_PA	—	—	—	—	—	—	—	—
RADIO	49	49	RF_OUTP	RF_OUTP	—	—	—	—	—	—	—	—
RADIO	50	50	RF_OUTN	RF_OUTN	—	—	—	—	—	—	—	—
RADIO	51	51	GND_PA	GND_PA	—	—	—	—	—	—	—	—
RADIO	52	52	VDD_PA	VDD_PA	—	—	—	—	—	—	—	—
RADIO	53	53	VDD_IF	VDD_IF	—	—	—	—	—	—	—	—

Table continues on the next page...

Table 3-1. MKW2xD Pins (continued)

Typical feature	MKW22/24D512V (USB)	MKW21D	Pin Name	Default	ALT0	ALT1	ALT2	ALT3	ALT4	ALT5	ALT6	ALT7
RADIO	54	54	VDD_RF	VDD_RF	—	—	—	—	—	—	—	—
RADIO	55	55	VBAT_RF	VBAT_RF	—	—	—	—	—	—	—	—
RADIO	56	56	XTAL_32M	XTAL_32M	—	—	—	—	—	—	—	—
RADIO	57	57	Factory Test	Do not connect	—	—	—	—	—	—	—	—
RADIO	58	58	Factory Test	Do not connect	—	—	—	—	—	—	—	—
RADIO	59	59	Factory Test	Do not connect	—	—	—	—	—	—	—	—
RADIO	60	60	Factory Test	Do not connect	—	—	—	—	—	—	—	—
RADIO	61	61	Factory Test	Do not connect	—	—	—	—	—	—	—	—
RADIO	62	62	Factory Test	Do not connect	—	—	—	—	—	—	—	—
RADIO	63	63	GND_PA	Connect to ground	—	—	—	—	—	—	—	—

3.4 Modem: Digital Signal Properties Summary

The following table summarizes modem's digital I/O pin characteristics.

Table 3-2. Modem Digital Signal Properties

Modem Pin Name	Dir	High Current Pin	Output Slew[1]	Pull-Up[2]	Comments
$\overline{\text{IRQ}}$	O	N	SWC	SWC	Open drain
XTAL1	I	—	—	N	
XTAL2	O	N	N	N	
$\overline{\text{RST}}$	I	—	—	N	
CLKO	O	N	SWC	N	
SPICLK	I	—	—	N	
MOSI	I	—	—	N	
MISO	O	N	SWC	N	Off state is SWC
$\overline{\text{CE}}$	I	—	—	N	

Table continues on the next page...

Table 3-2. Modem Digital Signal Properties (continued)

Modem Pin Name	Dir	High Current Pin	Output Slew[1]	Pull-Up[2]	Comments
GPIO1	I/O	N	SWC	N	
GPIO2	I/O	N	SWC	N	
GPIO3	I/O	N	SWC	N	
GPIO4	I/O	N	SWC	N	
GPIO5	I/O	N	SWC	N	
GPIO6	I/O	N	SWC	N	
GPIO7	I/O	N	SWC	N	

[1] SWC is software controlled slew rate, the register is associated with the respective port.

[2] SWC is software controlled pull-up resistor, the register is associated with the respective port.

Chapter 4
SiP: System Considerations

4.1 Introduction

The MKW2xD is the embodiment of a 802.15.4 node in a single SiP package which can provide solutions to proprietary nets, 802.15.4 MAC-compatible. All control of the node is done through the onboard ARM Cortex-M4 processor, and all MCU peripherals, MCU GPIO, modem functionality, and modem GPIO are manipulated by the processor. The MCU GPIO and MCU peripherals are accessed from the MCU internal bus and can be programmed directly.

Communication to the modem function is through the common SPI bus, the MCU interrupt request, and several MCU GPIO lines. Primary interface with the modem is through the SPI command structure that allows reading/writing modem registers and provides initialization of parameters, reading of status, and control of modem operation. The modem can ask for real time response through the interrupt request structure, and four GPIO signals allow control of the modem reset and monitoring of some real time status.

This chapter presents information addressing application and operation of the node from a system level. The areas considered here are also covered in greater detail in the following sections of the book. The book is organized such that the first three chapters present the top-level view of the MKW2xD device and the following chapters present individual functions in detailed descriptions.

4.2 Power Connections

The modem power connections are listed in Table 4-1.

Table 4-1. Power Pin Descriptions

Pin Name	Type	Description	Functionality
VDD_RF	Power Output	Regulator output for RF circuitry	Decouple to ground.
VBAT_RF	Power Input	Main voltage supply	Decouple to ground.
VDD_REGD	Power Output	Regulated output supply for digital circuitry	Decouple to ground
VBAT2_RF	Power Input	Main voltage supply	Decouple to ground.
GND_PA	Power Input	PA supply ground	Connect to ground
GND_PA	Power Input	PA supply ground	Connect to ground
VDD_PA	Power Input	Regulated supply for PA	Decouple to ground
VDD_IF	Power Input	Regulated supply for IF	Decouple to ground
VSS	Power Input	External package flag. Common VSS	Connect to ground

When designing power to the device, the following points need to be considered for the modem:

- The LGA package has two common EP ground flags (VSS) (one for the MCU and one for the modem).
- There are two modem primary power inputs, which include VBAT_RF for modem power and VBAT2_RF for digital circuitry.
- For logic level compatibility between the modem and the system MCU, VBAT_RF, and VBAT2_RF must be connected with the MCU to a common source supply of 1.8–3.6 VDC. It is not recommended to supply the MCU below 1.8V.
- VDD_PA is the supply of the internal power amplifier and is powered by the VDD_RF regulator output via the analog regulator.
- Output VDD_RF and VDD_IF are provided to allow bypass of the RF and IF circuitry regulated supplies. The decoupling capacitor shall be in the range of 220nF and 470nF as shown in the figure.
- The VDD_REGD supply is decoupled externally as shown in the figure. The external decoupling capacitor is in the range of 220 nF and 470 nF.

Power supply connections for the modem are shown in the following figure.

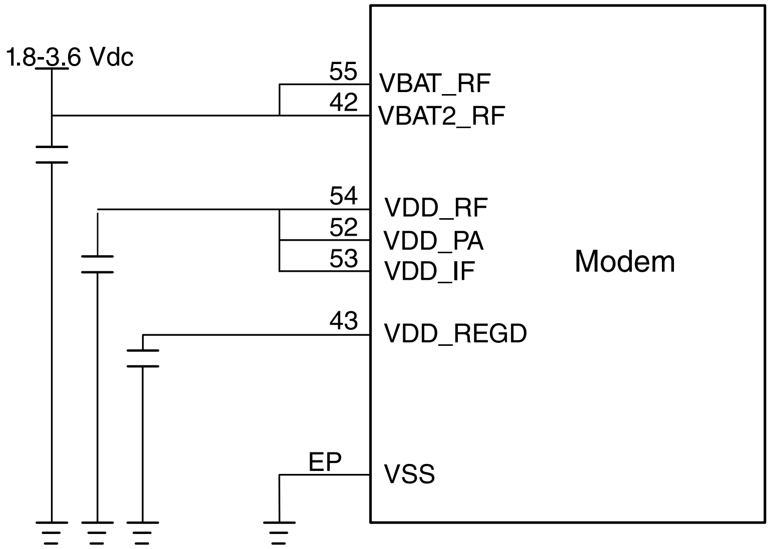


Figure 4-1. Modem Power Supply Connections

Note

There are separate bypass capacitors on VDD_RF, VDD_IF, VDD_PA, and VDD_REGD. Additional inormation for the modem's power supply can be found in Digital Regulator and POR and Analog Regulator (AREG)

4.3 Internal Functional Interconnects and System Reset

The MCU provides control for the 802.15.4 modem. The required interconnects between the devices are routed onboard the SiP. In addition, three modem GPIOs connected to the MCU can optionally be passed through the MCU to SIP pads; this allows the SIP pads to be either used by the MCU or used as GPIOs by the modem. All of the interconnects are listed in the table below.

Table 4-2. Internal Connections

MCU pad	Modem pad	Description
PTB19	RST_B	MCU PTB19 GPIO used to reset the modem under software control
PTB3	IRQ_B	Modem interrupt request to the MCU
PTA18/EXTAL0	CLK_OUT	Modem clock output used as a clock input to the MCU. **NOTE:** This signal is also connected to a SIP package pin
PTB10/SPI1_PCS0	R_SSEL_B	MCU SPI master slave select drives modem SPI slave select input
PTB11/SPI1_SCLK	R_SCLK	MCU SPI master clock output drives modem SPI slave clock input
PTB16/SPI1_SOUT	R_MOSI	MCU SPI master data output drives modem SPI slave data in
PTB17/SPI1_SIN	R_MISO	MCU SPI master data input is driven by modem SPI slave data out

Table continues on the next page...

Table 4-2. Internal Connections (continued)

MCU pad	Modem pad	Description
PTC3	GPIO3	Modem GPIO3. This is connected to MCU PTC3 where it can be optionally passed through the MCU to SIP pin PTD4
PTC1	GPIO4	Modem GPIO4. This is connected to MCU PTC1
PTC0	GPIO5	Modem GPIO5. This is connected to MCU PTC0. At modem reset, MCU drives high or low value for GPIO5 to select default CLK_OUT frequency.

NOTE

To use the MCU and modem signals as described in the table, the MCU and modem need to be programmed appropriately for the stated function.

The internal interconnects are shown graphically and grouped functionally in the figure below. The MCU reset input RESET_b is also shown, as are the modem GPIO1-2 and GPIO6-8. System reset, control and monitoring of the modem, and SPI communication are all concerned with these signals.

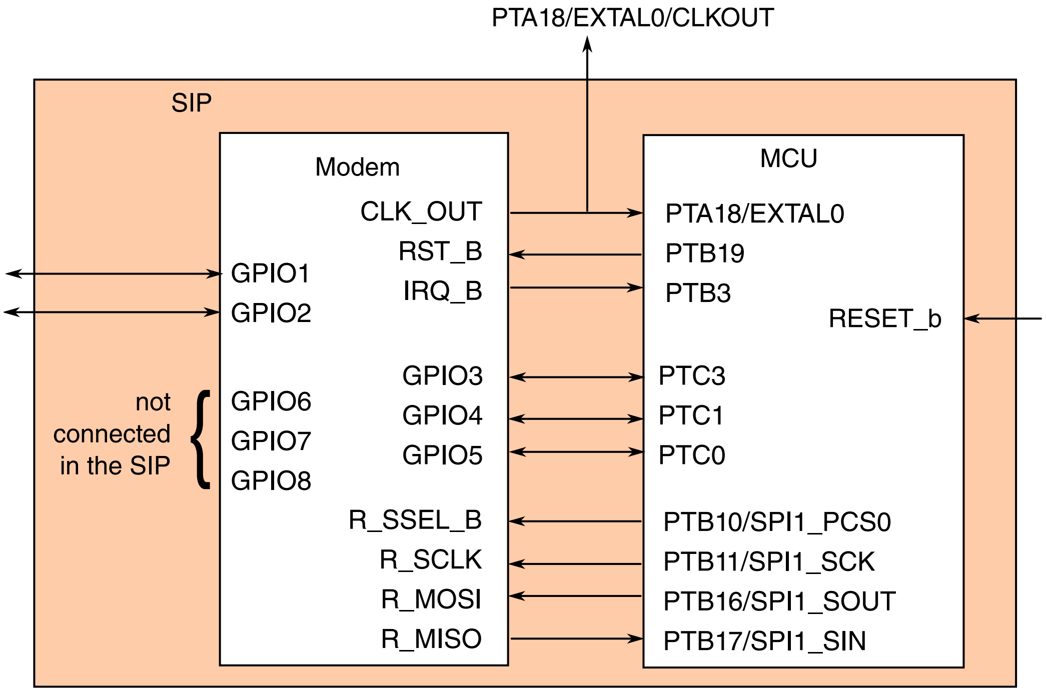


Figure 4-2. Internal Functional Interconnects

4.3.1 System Reset

The MKW2xD does not have a master input that resets the entire SiP. Instead, the MCU reset input, RESET_b, is the overriding reset and the reset input to the modem RST_B is controlled by the MCU (in Run Mode).

4.3.1.1 MCU Reset Input RESET_b

RESET_b is an active low dedicated open-drain pin an internal pull-up device built in. Internal power-on reset and low-voltage reset circuitry typically make external reset circuitry unnecessary. This pin is normally connected to the debug connector so a development system can directly reset the MCU system. If desired, a manual external reset can be added by supplying a simple switch to ground (pull RESET_b pin low to force a reset).

Whenever any reset is initiated (whether from an external signal or from an internal system), the reset pin is driven low for approximately 128 bus clock cycles and then released.The reset circuitry decodes the cause of reset and records it by setting a corresponding bit in the MCU Mode Controller's System Reset Status Registers (MC_SRSH/L).

MCU reset operation and control is detailed in Chapter 13, “MCU Reset and Boot”.

4.3.1.2 Modem Reset

The modem active low reset input RST_B is driven onboard the SiP from MCU GPIO pad PTB19. In the interest of lowest power, there is no pull-up resistor on input RST_B.

From a MCU reset condition, the PTB19 is configured as a high-impedance input with the internal pull-up disabled. As a result, the modem reset input will be floating and the modem will not be held in reset.

As part of the MCU initialization,

- The MCU software must configure PTB19 as an output and then drive it low to reset the modem.

During the modem reset procedure,

- The MCU software should also configure PTC0 as an output and drive it appropriately to select the default modem CLK_OUT frequency (PTC0 high = 32.787 kHz; PTC0 low = 4 MHz).
- The modem RST_B input is asynchronous and needs to be held low for only a short period.

In the reset condition, the modem is totally powered down and no clocks are available. After RST_B is released, the modem will power up, initialize, and go to its idle condition in less than 1 millisecond, and in turn, this causes an IRQ_B interrupt request and allows CLK_OUT to start toggling. Coming out of reset, the WAKE_IRQ interrupt status bit is set and causes the assertion on the IRQ_B because the wake interrupt source is not masked by default.

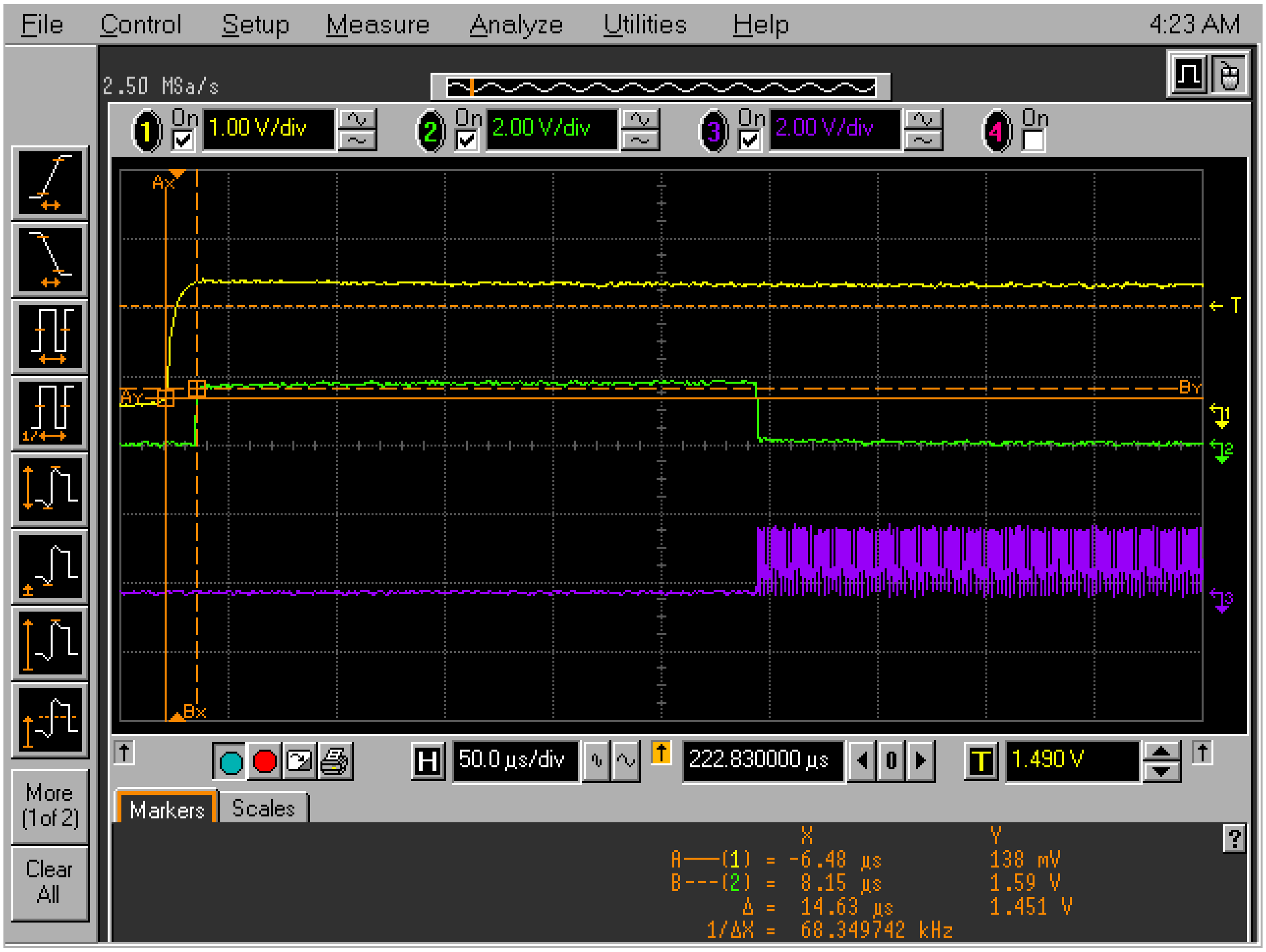
Once the interrupt request is seen by the MCU, the MCU can assume the modem is alive and ready for programming via the SPI bus.

4.3.2 Modem Interrupt Request to MCU

The modem interrupt request IRQ_B is an active low output that is asserted when an interrupt request is pending. The signal is released to high by writing a 1 to all asserted interrupt status bits in the modem status registers via a SPI transaction.

The modem interrupt request is connected inside the SIP to the MCU PTB3 pad. The MCU PTB3 must be programmed as an external interrupt and can be enabled for edge-only or edge-and-level events. The MCU IRQ also has a programmable pull-up. IRQ is covered in detail in Chapter 6, “Modem Interrupts”.

The following figure shows the timing for the radio out of reset to the IRQ signal asserted high. The timing was measured with Vbatt=1.8 V, which is worst case representing the 15 us delta. Any higher Vbatt voltage decreases this time.



```
2.50 MSa/s
1 On ☑ 1.00 V/div    2 On ☑ 2.00 V/div    3 On ☑ 2.00 V/div    4 On □
H 50.0 µs/div    222.830000 µs    0    T 1.490 V

                    X                  Y
A——(1) = -6.48 µs          138 mV
B---(2) =  8.15 µs          1.59 V
      Δ =  14.63 µs         1.451 V
   1/ΔX =  68.349742 kHz
```

Figure 4-3. KW2x Reset to IRQ Assert High Timing at Vbatt = 1.8 V

4.3.3 SPI Command Channel

The MCU SPI Module is dedicated to the modem SPI interface. The modem is a slave only and the MCU SPI must be programmed and used as a master only. The SPI bus connections are:

- MCU PTB6/SPI1_SOUT output drives modem R_MOSI
- Modem R_MISO output drives MCU PTB17/SPI1_SIN
- MCU PTB11/SPI1_SCK output drives modem R_SCLK
- MCU PTB10/SPI1_PCS0 output drives modem R_SSEL_B

More information on the SPI command channel can be found in the MCU-Modem SPI Interface chapter.

4.4 Clock Sources

The MKW2xD device allows for a wide array of system clock configurations. Three separate crystal oscillators are provided:

- A 32 MHz crystal used as the modem clock source. This crystal is required in all clock configurations. The modem crystal oscillator frequency can be trimmed through programming to maintain tight tolerances required by the 802.15.4 Standard. The modem provides a CLK_OUT programmable frequency clock output that is derived from the 32MHz crystal, and which is connected inside the SIP to the MCU EXTAL clock input. As a result, a single crystal system clock solution is possible
- A 3-32 MHz crystal provides an optional external clock source for the MCU as an alternate to using the modem CLK_OUT.
- A 32 kHz crystal can be used to provide a clock to the MCU RTC, and can also be used with the MCU MCG's FLL to provide a higher frequency clock to the MCU.

The MCU also contains internal slow (32 kHz) and fast (4 MHz) clock generators (each of which can be trimmed) that can be used to run the MCU for low power operation. Out of reset, the MCU uses the internally generated slow clock and FLL to provide a 20-25MHz MCU clock for start-up. This allows recovery from stop or reset without a long crystal start-up delay

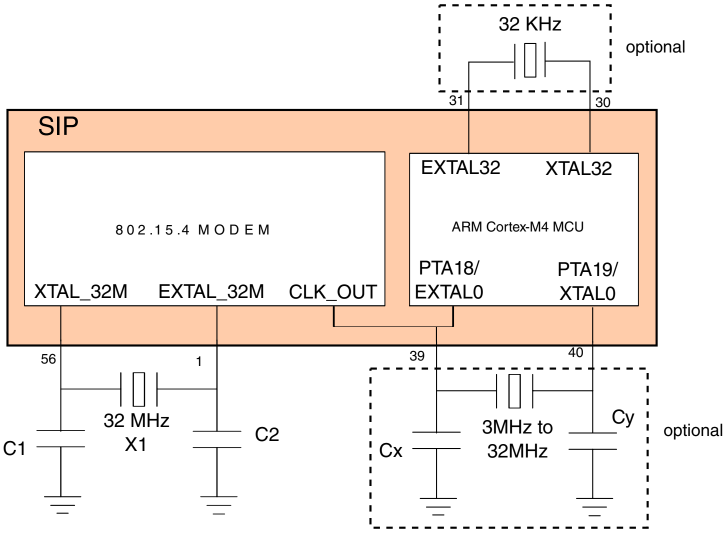


Figure 4-4. Clock pins

4.4.1 Modem Oscillator

The modem oscillator source must always be present and an external crystal is used to implement the oscillator. The source frequency must be 32 MHz with a total accuracy of +/- 40 ppm or greater as required by the 802.15.4 Standard. A detailed discussion of required crystal characteristics is in “Crystal Requirements”.

In the previous figure, crystal X1 and capacitors C1 and C2 form the modem crystal oscillator circuit. An onboard feedback resistor of approximately 1 MOhm (not shown) between input XTAL_32M and output EXTAL_32M provides DC biasing for the oscillator buffer. An important parameter for the 32 MHz crystal X1 is a load capacitance of <9 pF. The oscillator needs to see a balanced load capacitance at each terminal of about 18pF. As a result, the sum of the stray capacitance of the pcb board, device pin (XTAL_32M or EXTAL_32M), and load capacitor (C1 or C2) at each terminal must equal about 18 pF. C1 and C2 are typically values of 6-9 pF. Higher values can load the crystal buffer and cause oscillator start-up problems.

As described in “Crystal Trim Operation”, the modem crystal oscillator frequency can be trimmed by programming modem XTAL_TRIM register. The trimming procedure varies the frequency by a few hertz per step, depending on the type of crystal. As the XTAL_TRIM register value is increased, the frequency is decreased. This feature is useful for factory calibration of the crystal frequency to set the accuracy for the radio as required by the 802.15.4 Standard.

4.4.2 Modem CLK_OUT Clock Source Output

The modem provides a programmable clock source output CLK_OUT. The CLK_OUT output frequency can be varied from 32.787 kHz to 32 MHz. The primary uses of CLK_OUT include:

- MCU clock source - for lowest cost operation, a separate crystal is not required for the MCU. The CLK_OUT signal can drive input pin EXTAL to drive the MCU MCG as an external clock source. The MCG is covered in detail in the module’s dedicated chapter..
- A monitor point for the modem clock frequency - the CLK_OUT is a buffered output that can be used to observe the modem frequency for calibration and design purposes. Modem oscillator pins EXTAL32M and XTAL32M cannot be used for monitoring the oscillator frequency because this will load the circuit and cause either a frequency error or shutdown the oscillator.

The CLK_OUT output frequency is controlled by programming the modem 3-bit field CLK_OUT_DIV [2:0] in the CLK_OUT_CTRL Register. The default frequency is either 32.787 kHz or 4 MHz depending on state of GPIO5 at reset determine by the MCU.

- If the GPIO5/BOPT pin is low upon POR, then the frequency will be 4 MHz (32 MHz/8).
- If this GPIO5/BOPT pin is high upon POR, then the frequency will be 32.78689 kHz (32 MHz/976).

4.4.3 MCU Clock Sources

The MCU has several options for its primary clock source depending on its mode of operation as well as its hardware configuration. For more details on the MCU MCG, system OSC, and RTC OSC, refer to their chapters.

4.4.3.1 MCU 3-32MHz OSC Clock Source

The modem CLK_OUT is connected to the MCU EXTAL input inside the SIP. If an alternate clock is needed, the modem's CLK_OUT can be disabled and a 3-32MHz crystal can be connected to the MCU EXTAL/XTAL pins. Use of the modem CLK_OUT clock is described more fully in Single Crystal with CLK_OUT used by MCU. and use of the optional 3-32MHz OSC clock is described in Dual Crystal Operation with 3-32MHz OSC

4.4.3.2 MCU 32kHz OSC Clock Source

The MCU supports an external 32kHz crystal. This is primarily used to provide a 32kHz clock to the RTC, but can be used by the MCG as a reference for the FLL to generate a higher frequency clock, and by other circuits as well. This is described below in Dual Crystal Operation with 32KHz RTC OSC

4.4.3.3 MCU Internal Clock Sources

The MCU MCG includes two internal reference clock (IRC) sources:

- Fast IRC. This clock is approximately 4 MHz and can be trimmed.
- Slow IRC. This clock is approximately 32 KHz and can also be trimmed. At reset, this clock is used as the source to the MCG's FLL which provides a 20-25MHz clock for the MCU.

This option is discussed more in Single Crystal with MCU Using Internal Clock Only

4.4.4 System Clock Configurations

Because of the multiple clock configurations of the MCU, the availability of external clock source pins of both the modem and the MCU, and the CLK_OUT output from the modem, there are a number of variations for system clock configurations. Key considerations for any system clock configuration are:

- The modem 32 MHz source (typically the crystal oscillator) must always be present. The crystal has special requirements and the reference frequency must meet the 802.15.4 Standard.
- Battery-operated application requirements for low power can impact the choices for MCU clock source.
- The system clock configuration will impact system initialization procedures.
- Software requirements can impact MCU processor and bus speed. The user must be aware of the performance requirements for the MCU. The application software may require specific bus rates. The implication is that the MCU clock source must be capable of generating these bus rates with or without the MCU PLL or FLL.

As far as external connections are concerned, there are three possibilities which are covered in the following sections.

4.4.4.1 Single Crystal with CLK_OUT used by MCU

The single crystal (modem crystal) with CLK_OUT driving the MCU EXTAL input (external clock) is the most common configuration for low cost and excellent frequency accuracy. The CLK_OUT frequency is programmable from 32.787 kHz to 32 Mhz and drives the MCU external source. The MCU PLL or FLL can also be used.

In this configuration, clock start-up from a reset condition involves:

- MCU reset is released and MCU starts on internal 20-25 MHz clock using the internal slow IRC and FLL
- Initialization software must reset and then release reset to the modem (MCU still running on start-up clock)
- Wait for modem start-up interrupt request (approximately 25 msec). CLK_OUT default is active with a frequency of 32.787 kHz or 4MHz depending on how MCU asserts GPIO5 during modem reset.
- MCU uses DSPI1 to program CLK_OUT to a different frequency (if desired)
- Switch the MCU to CLK_OUT, or program and wait for the PLL or FLL to lock, and then switch MCU clock to that source

Additional considerations for this mode of operation include:

- If the modem is forced to an Off condition (such as a sleep mode), there is a 25 msec wait for the modem CLK_OUT to start from the off condition after RST_B is released
- If the MCU puts the modem in a low power mode, keeping the CLK_OUT alive is a higher power option
- The internal reference for the MCU can still be used. Disable the CLK_OUT output on the modem and program the MCU clock for internal operation
- If an accurate period is required for longer time delays (such as a beacon period), keeping CLK_OUT alive for very long periods is not the lowest power option

4.4.4.2 Single Crystal with MCU Using Internal Clock Only

The single crystal (modem crystal) with the MCU using internal clock only has no real advantage over using the CLK_OUT output. The CLK_OUT output can be disabled to save power (EXTAL is grounded), and using the onboard slow IRC with the FLL, a high clock frequency can be generated for the MCU.

In this configuration, clock start-up from a reset condition involves:

- MCU reset is released and MCU starts on internal 20-25 MHz clock using the internal slow IRC and FLL
- Initialization software must reset and then release reset to the modem (MCU still running on start-up clock)
- Wait for modem start-up interrupt request (approximately 25 msec). CLK_OUT default is active with a frequency of 32.787 kHz or 4MHz depending on how MCU asserts GPIO5 during modem reset.
- MCU uses DSPI1 to program modem to disable CLK_OUT

Additional considerations for this mode of operation include:

- The MCU does not have a high accuracy time base when using the internal reference

4.4.4.3 Dual Crystal Operation with 32KHz RTC OSC

The modem crystal can be augmented by the use of a second crystal on the MCU RTC OSC. The main advantage of this is to provide an accurate time base in the MCU for long power down delays. The obvious disadvantage of this configuration is additional cost.

In this configuration, clock start-up from a reset condition involves:

- MCU reset is released and MCU starts on internal 20-25 MHz clock using the internal slow IRC and FLL
- Switch the MCU FLL to use the RTC OSC source, and reprogram the FLL as needed.
- Initialization software must reset and then release reset to the modem (MCU still running on start-up clock)
- MCU uses DSPI1 to program modem to disable CLK_OUT

Alternately, the MCU can use the CLK_OUT as the MCU clock source during normal power modes, and only switch to the RTC clock during low power modes where the modem is turned off.

Additional considerations for this mode of operation include:

- This mode allows CLK_OUT to be disabled for lower power in the modem
- With using the FLL, there are limited clock output frequency options (24MHz, 48MHz, 72MHz, 96MHz)
- The second crystal is justified when accurate power down time periods are required. The external clock with the 32.768 kHz crystal allows an accurate 1 second time tick for RTC at very low power. Although power is not as low as the internal 32kHz oscillator, the time tick now has crystal accuracy.

4.4.4.4 Dual Crystal Operation with 3-32MHz OSC

The modem crystal can be augmented by the use of a second crystal on the MCU OSC. The only advantage of this is to provide a independent clock for the MCU which can be used even when the modem is off. The obvious disadvantage of this configuration is additional cost.

In this configuration, clock start-up from a reset condition involves:

- MCU reset is released and MCU starts on internal 20-25 MHz clock using the internal slow IRC and FLL
- Reset the modem. Out of reset the modem CLK_OUT will be active and driving a clock to the MCU EXTAL and external 3-32MHz crystal.
- MCU software performs a modem SPI write to disable the CLK_OUT.
- MCU configures and enables the OSC.
- MCU switch the MCU to the OSC or switches to the FLL or PLL which use the OSC clock for reference.

Additional considerations for this mode of operation include:

- The second crystal may be justified when a high precision clock is required and the modem needs to be turned off. Otherwise, the MCU could use its internal references when the modem is turned off.

4.4.4.5 Triple Crystal Operation

The modem crystal can be augmented by both a 32kHz crystal used with the MCU RTC OSC and a 3-32Mhz crystal used with the MCU OSC. The obvious disadvantage of this configuration is additional cost.

In this configuration, clock start-up from a reset condition would be the same as that described in Dual Crystal Operation with 3-32MHz OSC.

Additional considerations for this mode of operation include:

- The additional crystals may be justified when a high precision clock is required and the modem needs to be turned off, and the RTC time-of-day also needs to be maintained.

4.5 GPIO for Modem and MCU

The SIP supports a total of 25 (MKW22/24D512) to 29 (MKW21D256) pins which can be configured as GPIO, and which originate from both the modem and the MCU:

- Two package pins (GPIO1-2) are GPIOs associated with the modem and can be used as inputs or outputs
- Due to package limitations, modem GPIO6-8 are not available in the SIP package.
- Up to 25 (MKW22/24D512) to 29 (MKW21D256) additional SIP pins can be configured as MCU GPIOs. The actual MCU hardware supports more, but due to package limitations only a limit number can be used realistically. Also, all of these pins share functionality with MCU peripherals.

4.5.1 MCU GPIO Characteristics

The internal MCU GPIO hardware consists of 5 ports with up to 32 signals per port, though not all are available in the SIP. In the SIP, there are a total of 25 (MKW22/24D512) to 29 (MKW21D256) pins which can be configured as MCU GPIOs. Immediately after MCU reset, none of these pins are configured as GPIOs; some are disabled; and the others are configured as A/D or CMP inputs, JTAG or NMI (with internal pull up or downs enabled), or as EXTAL0/XTAL0 for the oscillator.

The pins internally connected to the modem are dedicated to SPI communication, or modem control. For all these cases the functionality of the GPIO are controlled by the applications program and direct programming of the MCU peripherals. For information about controlling these pins as general-purpose I/O pins, see MCU: Port control and interrupts (PORT). For information about how and when on-chip peripheral systems use these pins, refer to Signal Multiplexing and Signal Descriptions.

4.5.2 Modem GPIO Characteristics

The modem GPIO hardware consists of eight (8) signals total (GPIO1-GPIO8) though only GPIO1-6 are used in the SIP, and only GPIO1-2 are connected directly to SIP package pins. Immediately after reset, each GPIO has either an internal pull-up (GPIO1-5) or pull-down (GPIO6-8) enabled.

NOTE

> To avoid extra current drain from floating input pins, the reset initialization routine in the application program should change the direction of unused pins to outputs (programmed low) so the pins do not float. Outputs programmed low is the preferred option for lowest power.

The functionality of the modem GPIO is controlled by programming of the modem SPI registers via the SPI interface. For information about controlling all these pins as general-purpose I/O pins, the Modem Registers.

4.6 Transceiver RF Configurations and External Connections

The MKW2xD radio has features that allow for a flexible and low cost RF interface:

- Programmable output power from –35 dBm to +8 dBm. This allows the user to set the power based on their specific applications minimizing current consumption while maximizing battery life.
- –102 dBm (typical) receive sensitivity — At 1% PER, 20-byte packet (well above IEEE 802.15.4 specification of –85 dBm).
- Integrated transmit/receive (T/R) switch for low cost operation — With internal PAs and LNA, the internal T/R switch allows a minimal part count radio interface using only a single balun to interface to a single-ended antenna.

4.6.1 RF Interface Pins

The following figure shows the RF interface pins and the associated analog blocks. The MKW2xD radio is a differential transceiver architecture that has two (2) I/Os that support both receive and transmit functionality. Specifically, these I/Os are opposite in phase and will require off-chip matching for optimal performance. Because RF_OUTP and RF_OUTN ports are shared and duplexed, matching networks will need to accommodate both modes of operation.

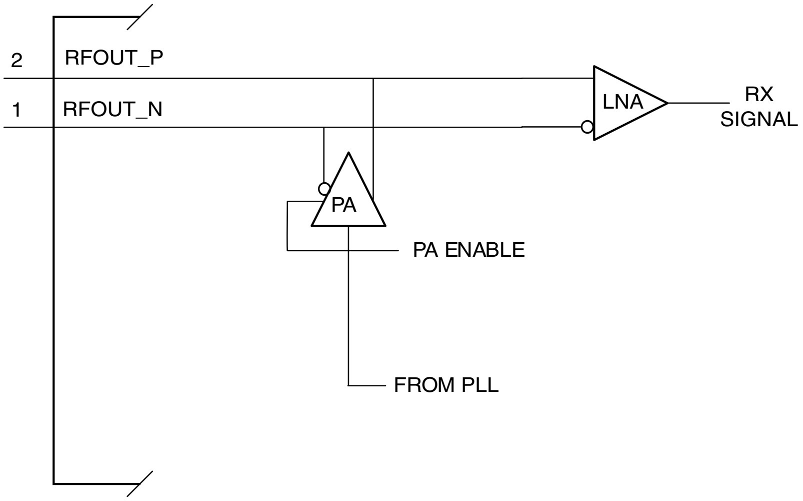


Figure 4-5. RF Interface Pins

4.6.1.1 Antenna Diversity

Fast Antenna diversity (FAD) is supported in hardware through use of the four (4) dedicated control pins by register setup and control selection. They are ANT_A, ANT_B, RX_SWITCH, and TX_SWITCH. The digital regulator supplies bias to analog switches for control of external pin diodes or external PA/LNA. These switches are programmable to sink and source two (2) levels of current (2 - 3 mA and 10 mA) or can operate in a high impedance mode. Each I/O will have a buffered feature with adjustable slew rate capability. Unless overridden by software, the last-good-antenna-selection (based on a successful preamble detect), is locked in place for TX sequences, as well as CCA and Energy Detect sequences. The hardware-selected antenna will not change again until the next RX preamble detection.

4.6.1.2 Fast Antenna Diversity (FAD)

The following system factors can be considered when configuring FAD for end applications:

- With Fast Antenna Diversity, sensitivity performance increases the low end limit of the dynamic range of the system
- There is little cost of FAD which is essentially enabling the functionality and adding the circuitry for situations where the application runs the possibility of operating at the limits of RX sensitivity

Figure 4-5 shows a simplified block diagram of the two port antenna control topology

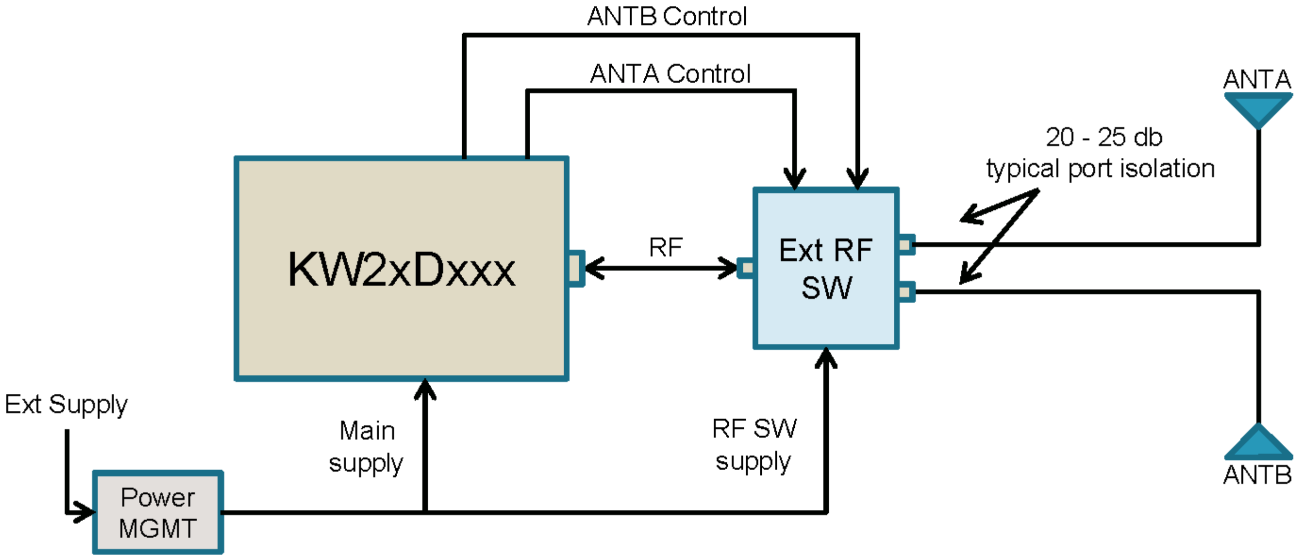


Figure 4-6. FAD control simplified block diagram

MKW2xD FAD Control Register Setup

The following registers allow setup, control and enable of the Fast Antenna Diversity (FAD) feature:

Antenna PAD Control (Indirect Modem_ANT_PAD_CTRL, 0x30): This register sets the action on the pads ANTA, ANTB, RX_SWITCH and TX_SWITCH.

Set bit field ANTX_EN [1:0] to proper selection for end application (default is 0)

- ANTA and ANTB switch signal during the preamble phase of the receive cycle
- RX_SWITCH and TX_SWITCH switch during the receive and transmit cycle respectively

Miscellaneous Pad Control (Indirect Modem_MISC_PAD_CTRL, 0x31): This register sets the pad drive strength for ANTA, ANTB, RX_SWITCH and TX_SWITCH

Bit[0] = 0 (default) for normal drive strength

Antenna AGC and FAD Control (Indirect Modem_ANT_AGC_CTRL , 0x51): This register controls the FAD feature

Enable bit field [0] to 1 - Turns FAD on (default is 0)

Figure 4-6 shows the flow and timing diagram from the idle state where the the receiver take either the FAD RX Flow or the standard RX flow:

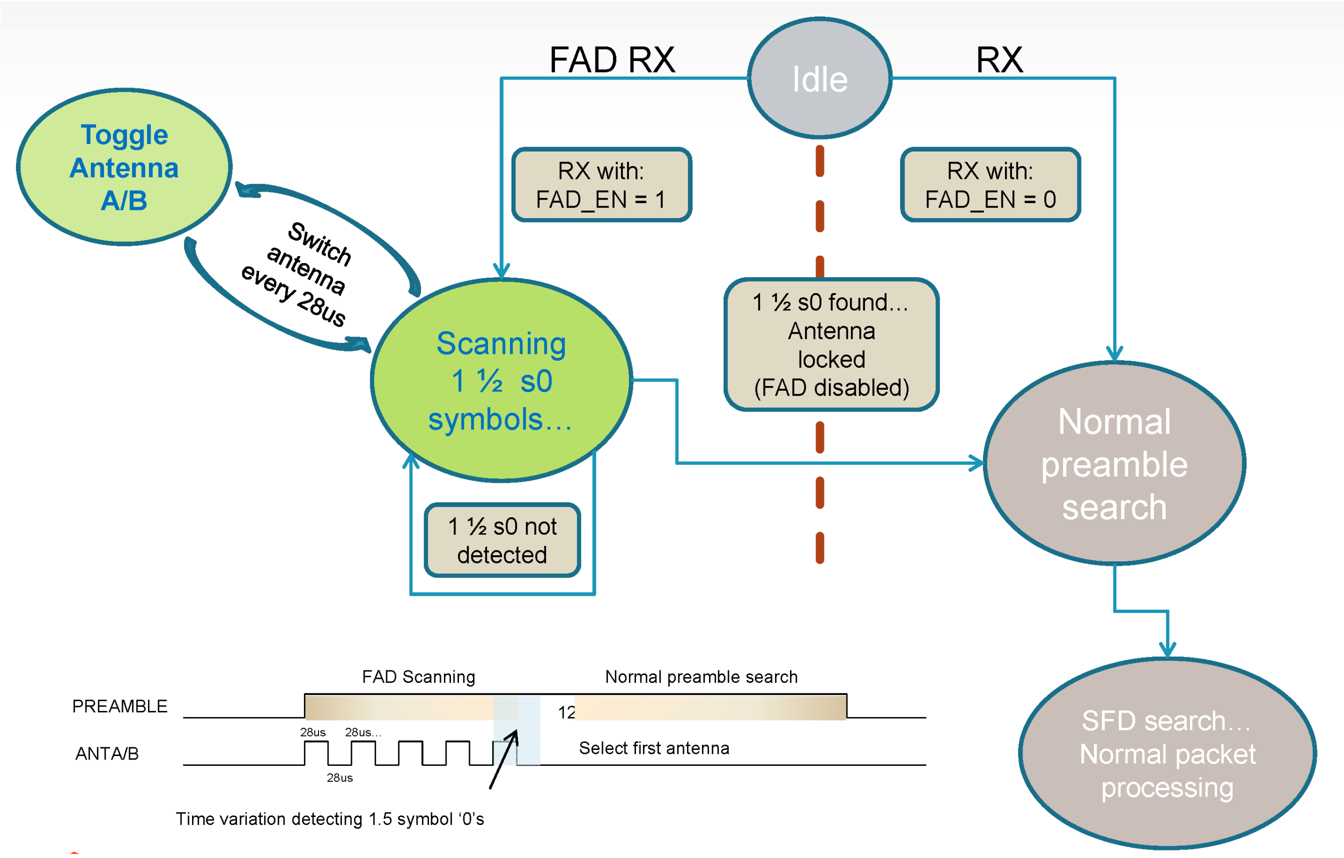


Figure 4-7. FAD flow and timing diagram

MKW2xD FAD and Packet Timing

Packet timing with FAD enabled with two external antenna’s A and B

During the time period t1 – t0, the antenna’s are switching at a rate of 28 us

- The circular region shows the timing variation while detecting 1.5 symbol ‘0’s and is a function of system sensitivity level
- At t1, the ‘first’ antenna meeting the 1.5 symbol ‘0’ criteria is selected … 3 more symbol ‘0’s are then detected to complete the preamble search

Normal packet processing follows a good preamble match

- The preamble period t2 – t0 is fixed at 128 us of the total packet cycle time – if a valid preamble is not found during this period, the receiver will start a new RX cycle

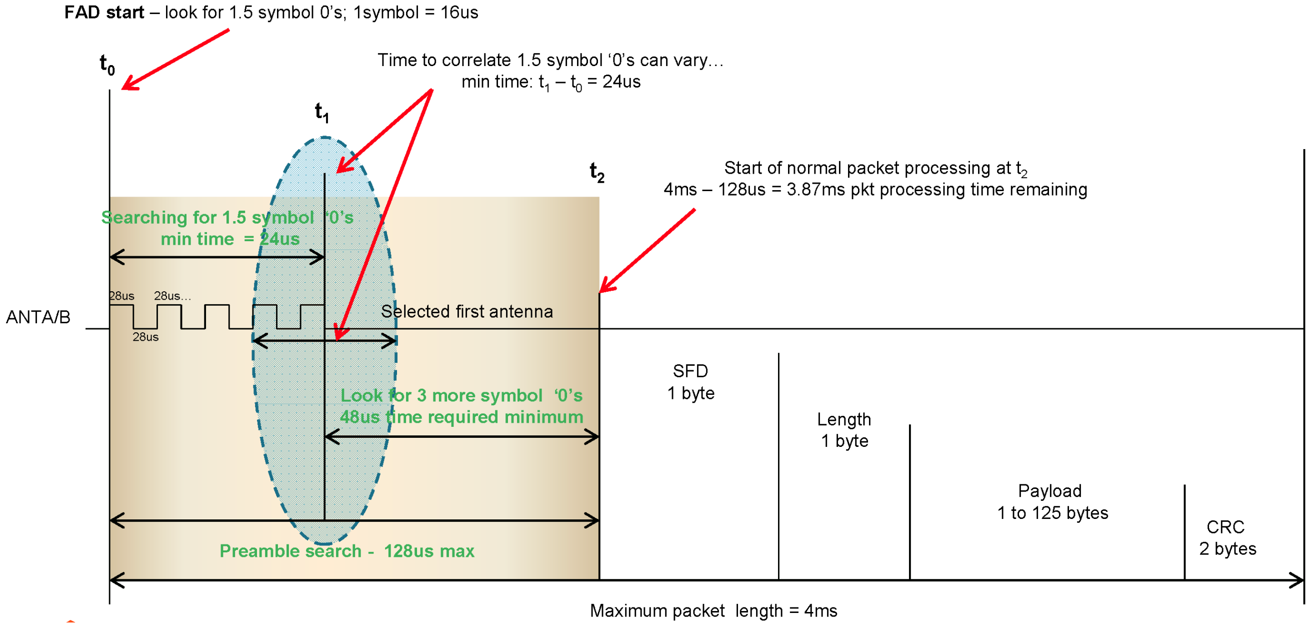

FAD start – look for 1.5 symbol 0's; 1symbol = 16us
t_0
t_1
Time to correlate 1.5 symbol '0's can vary…
min time: $t_1 – t_0$ = 24us
t_2
Start of normal packet processing at t_2
4ms – 128us = 3.87ms pkt processing time remaining
Searching for 1.5 symbol '0's
min time = 24us
ANTA/B
28us
28us…
28us
Selected first antenna
Look for 3 more symbol '0's
48us time required minimum
SFD
1 byte
Length
1 byte
Payload
1 to 125 bytes
CRC
2 bytes
Preamble search - 128us max
Maximum packet length = 4ms

Figure 4-8. FAD preamble and packet timing

Figure 4-8 shows a typical FAD control pin design topology.

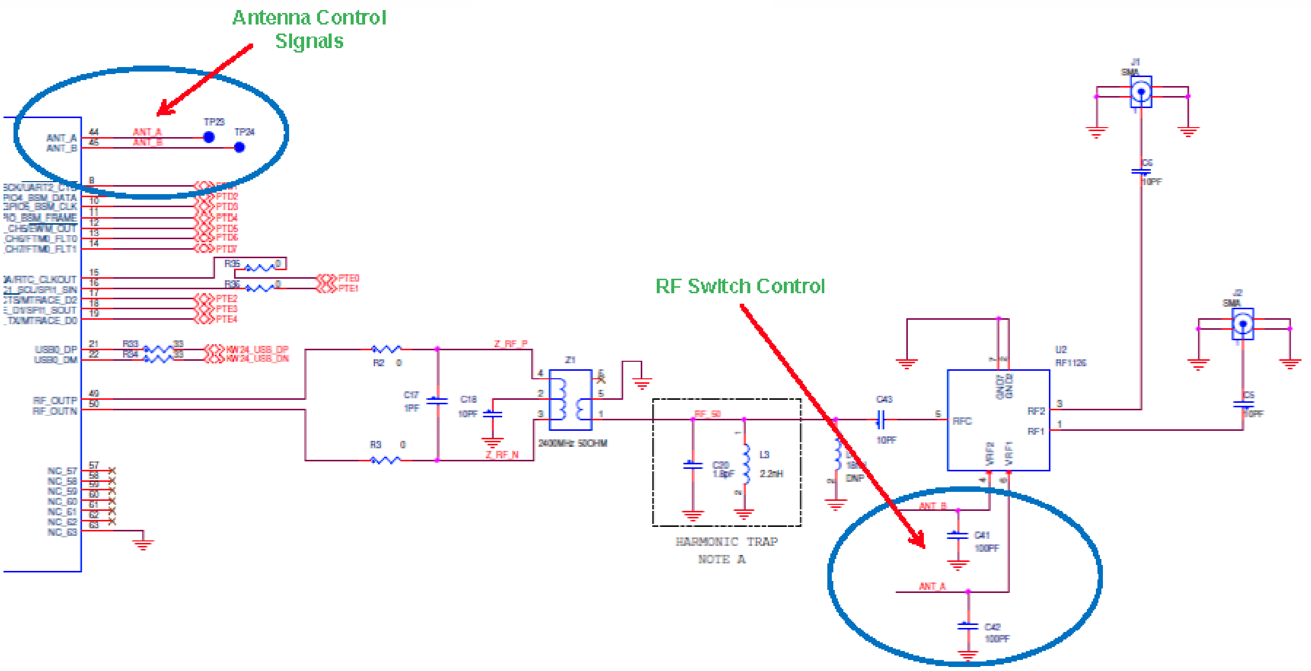


Figure 4-9. Typical circuitry for a two antenna external switch design

Figure 4-9 shows a simulated out of range antenna scenario when using the isolation through an RF switch to offset the RF signal by 21 dBm. The following sensitivity performance summary explains non- FAD and FAD Enabled Sensitivity Improvement.

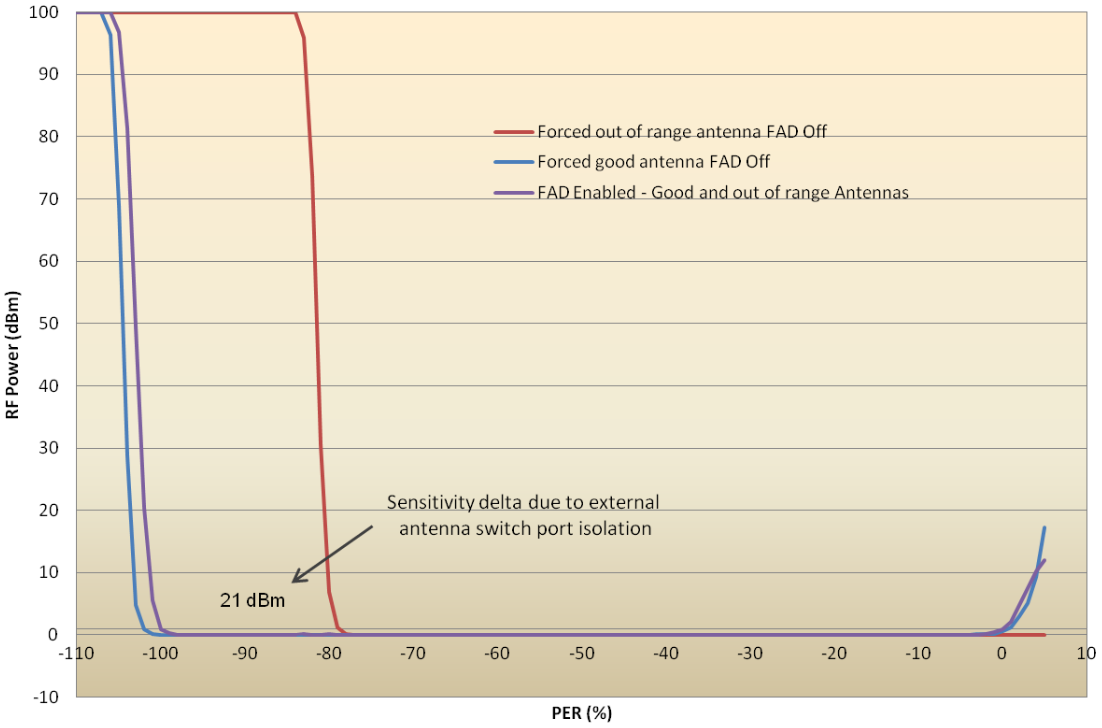


Figure 4-10. Sensitivity FAD mode at 2450 MHz

Sensitivity Performance Summary

Non-FAD mode

- By purposely choosing the bad antenna (out of range RF signal) it can be seen (orange curve) that the sensitivity is shifted to –78 dBm.
- By purposely choosing the good antenna (RF signal in range) it can be seen (blue curve) that the sensitivity is normal.
- The sensitivity delta is due to the isolation characteristics of the external RF switch component. This is the basic limitation of the system. The isolation of the RF switch ports in most cases is between 20-25 dB.

With FAD enabled (purple curve): With one antenna out of range and the other antenna good, the sensitivity is NOW in the normal range (about -99 dBm or better)

- Sensitivity performance is impacted by 1-2 dB in FAD mode vs. an ideally placed antenna but overall PER performance is significantly increased vs. having a single antenna with poor placement.

- Within the dynamic range of the receiver, sensitivity in FAD mode does not impact performance and therefore is not discernible to the user.
- Outside the dynamic range of the receiver is where FAD will help system performance and will have noticeable improvements to the system.
- Unless overridden by software, the last-good-antenna-selection (based on a successful preamble detect), is locked in place for TX sequences as well as CCA and Energy Detect sequences. The hardware-selected antenna will not change again until the next RX sequence.

4.6.2 RF Output Power Distribution

The following figure shows the linear region of the output and the typical power distribution of the radio as a function of PA_PWR [4:0] range. The PA_PWR [4:0] is the lower 5 bits of the PA_PWR 0x23 direct register and has a usable range of 3 to 31 decimal.

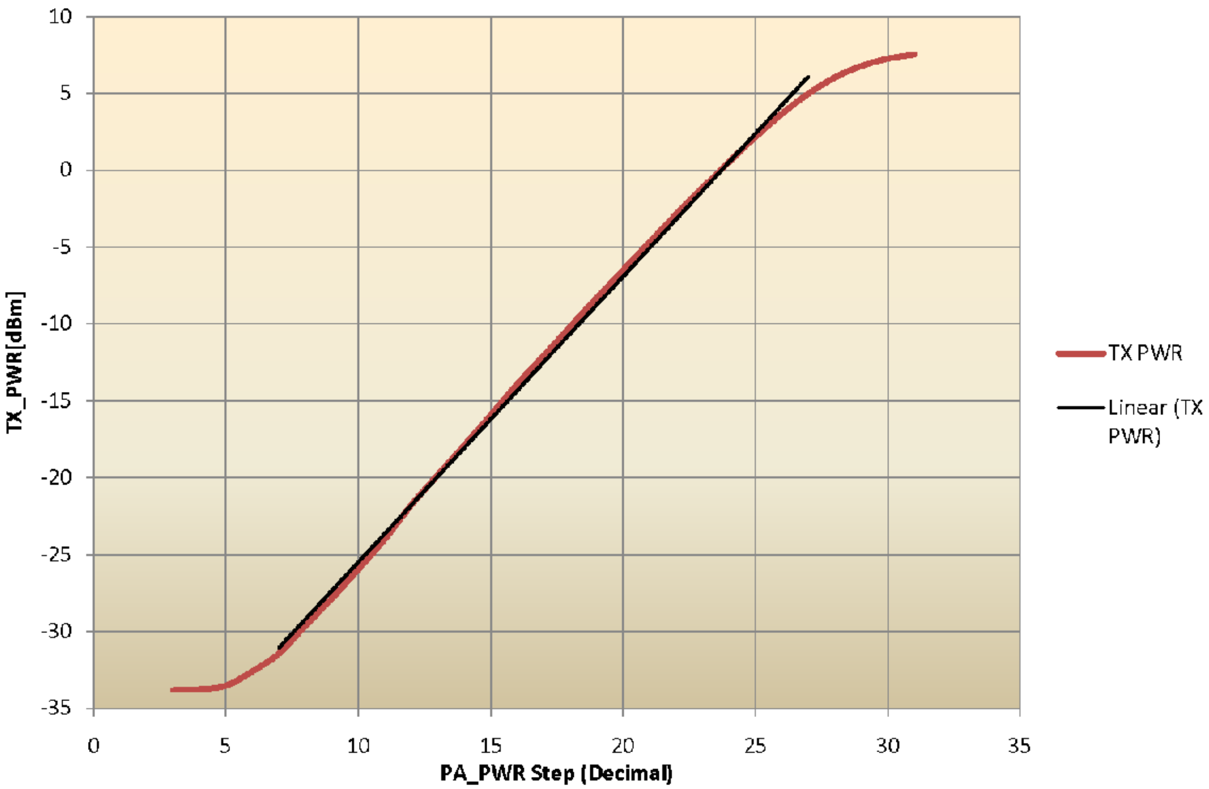


Figure 4-11. MKW2xD TX power vs. PA_PWR step

4.6.3 LQI ED RSSI

The following figure shows the MCR20A transceiver's reported energy detect (ED) as a function of input power. The figure also shows the link quality indication (LQI)/ RSSI_CONT (continuous) and the RSSI as a function of a unitless numeric register reading. LQI is available 64 μs after the preamble is detected. RSSI_CONT can be used to read LQI when a continuous value averaged over the entire received packet is desired. This value may be read as LQI if continuous update of LQI is desired during packet reception (both scaled the same but averaged differently). The curves are measured using the default offset compensation value and can be changed to center the curve if desired.

The following sequences are performed in test software to create the curves that are shown in the figure. Similar sequences are incorporated into NXP software stacks.

- For ED, the device is configured to perform an ED via write to the register 0x7 with 0x00.
    - ED/CCA sequence is started via write to register 0x3 with 0x3
    - Register 0 is polled for bit 3 to be set
    - RSSI_CONT_EN bit is disabled in indirect 0x25 (CCA_CTRL) all other bits are reset/overwrite values
    - Direct register 0x0B is read (CCA final)
    - Direct register 0x25 is read (LQI)
    - Indirect register 0x5B is read (RSSI)
    - Direct register 0x26 is read (RSSI_CONT), however, not enabled in ED case so ignored
- For LQI, the device is configured to perform an LQI via write to the register 0x7 with 0x00.
    - The device is configured to perform an LQI via write to the register 0x7 with 0x00
    - Receive sequence is started via write to register 0x3 with 0x1
    - Register 0 is polled for bit 2 to be set
    - RSSI_CONT_EN bit is enabled in indirect 0x25
    - RSSI_CONT_EN bit is disabled in indirect 0x25 (CCA_CTRL) all other bits are reset or overwrite values

- Direct register 0x0B is read (CCA final)
- Direct register 0x25 is read (LQI)
- Indirect register 0x5B is read (RSSI)
- Direct register 0x26 is read (RSSI_CONT), valid for LQI

For both LQI and ED the input power is swept with a modulated input signal from -100 dBm to -20 dBm in steps of 1 dBm and the ED and LQI sequences are called during each step and the results recorded.

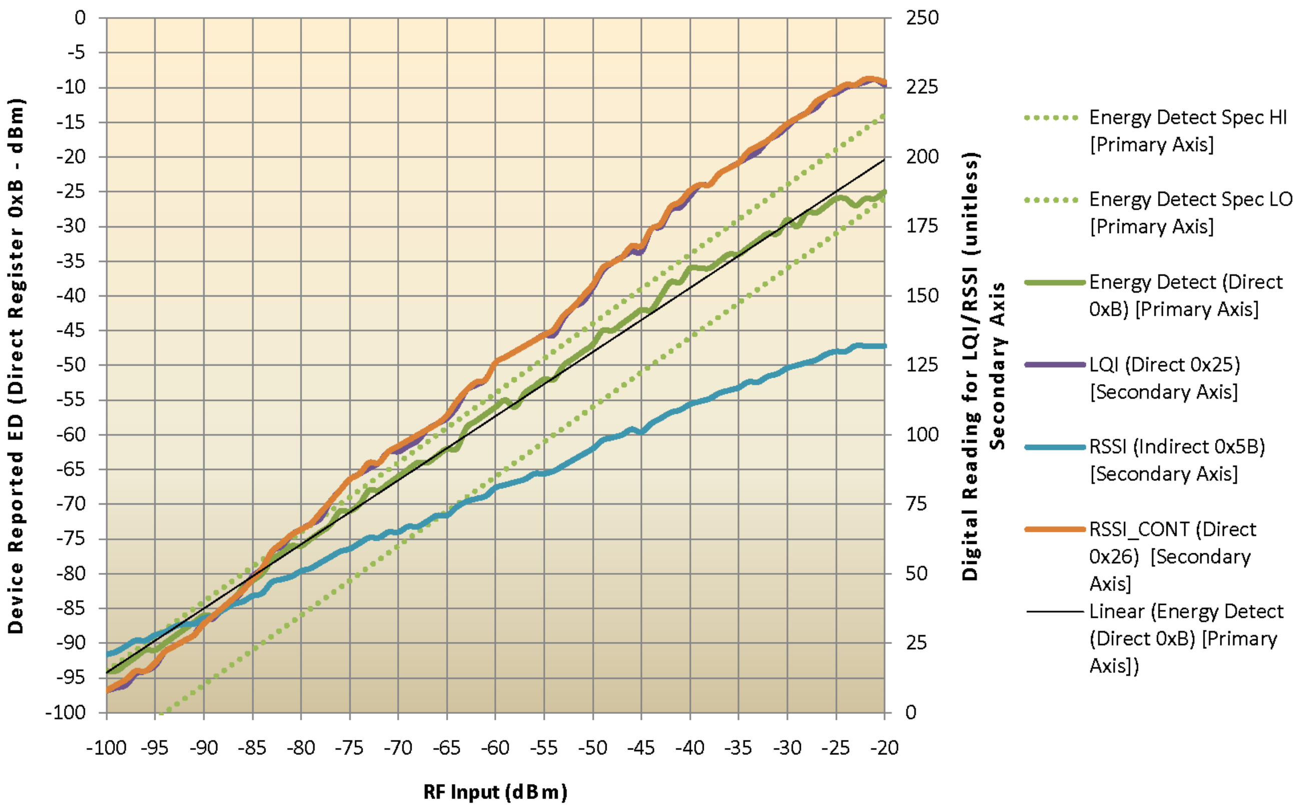


Figure 4-12. ED/LQI relation to RSSI/RSSI_CONT

Based on the figure, an equation can be derived as follows. Using Y(LQI) and X(RF), the minimum and maximum values from the secondary axis are determined as:

Table 4-3. Minimum and Maximum Values

LQI	Y(LQI)	X(RF)
Maximum	225	-25
Minimum	12	-100
Dynamic range mid point	125	-60

Therefore the slope = 2.84 and B = 295.4. So to convert the 8-bit unitless number read from direct register 0x25 to an ideal RF equivalent, use the following equation:

RF = (LQI – 295.4) / 2.84

4.7 Timer Resources

When writing software applications, the user should be aware of the set of timer resources available on the MKW2xD. The timer resources derive from both the MCU and modem functions and are individually best suited to different types of uses.

4.7.1 MCU Timers

The MCU includes the following timers :

- Three FlexTimer Modules (FTMs). FTM0 has 8 channels, while FTM1 and FTM2 have 2 channels each. The FTMs support input-capture, output-capture pins.
- The Periodic Interrupt Timer (PIT) has four channels and is primarily used as a generic timer for software
- The Low Power Timer (LPTMR) operates through all low power modes, so it is can be used to program a wake-up time for low power modes.
- The Real Time Clock (RTC) uses the 32kHz oscillator, and provides time-of-day function
- The Programmable Delay Block (PDB) is primarily used to provide programmable delays or ticks for the on-chip ADC, DAC, and HSCMP.

Each of these timers is described in its own chapter.

4.7.2 Modem Event Timer

The MKW2xD contains an internal Event Timer block whose clock is derived from a programmable prescaler which is driven by the 32 MHz crystal source. The Event Timer consists of the prescaler and a 24-bit counter which increment whenever the modem crystal clock is operating. The Event Timer provides the following functions:

- The 24-bit counter generates “current” system time - The counter runs continuously when the modem is not in a low power mode and provides a local “current” time. The present value can be read from the EVENT_TMR_USB/MSB/LSB registers. A counter frequency of 250kHz is used. The counter value is used to generate a time stamp for received frames and is also used for comparison to four different comparators that allow up to four different time delays

- Interrupt generation at pre-determined system times - There are four comparator functions that can generate an interrupt to the MCU when the value of the comparator matches the system counter. These function as timers to generate time delays or trigger events and can augment the MCU timers for generating time delays required within the 802.15.4 application software
- Exit from Doze Mode at pre-determined system time - Timer comparator Tmr_Cmp2 can be used to wake-up the modem from low power Doze Mode
- Latches “timestamp” value during packet reception - With every received frame, there is a timestamp generated and stored in a modem SPI register. The timestamp is taken from the counter when the frame length indicator (FLI) is received
- Initiates timer-triggered sequences - The modem has three basic active modes of RX, TX, and CCA. Each of these sequences can be timer-triggered to become active after a set delay based on timer comparator Tmr_Cmp2. This feature can be useful for implementing 802.15.4 application sequences

4.8 Low Power Considerations

Many IEEE 802.15.4 applications such as sensor End Devices are required to be battery operated. As expected, long battery life is highly desirable and is very dependent on application parameters. Over-the-air operation uses RX, TX, and CCA modes, where power is highest. As a result, the time between radio operations should be kept at the longest possible period that the application will allow.

When designing for low power operation consider:

- The modem and the MCU each have several low power options
- The modem is entirely controlled by the MCU; the low power options/combinations will be determined by MCU programming
- The power down control of the modem must be maintained by the MCU in the MCU's power down configuration
- All unused port GPIO (including those not pinned-out) must be configured to a known state (preferably output low) to prevent unwanted leakage current.

Lowest power in a system is more than just putting the modem and/or the MCU in a low power mode. The relationship between the functions, the timing between them, and clock management must all be considered. The duty cycle between active operations is also very important as it can impact whether sleep operation or active operation will have the biggest impact over an extended time period.

4.8.1 Low-Power Preamble Search (LPPS)

The Low Power Preamble Search (LPPS) feature of the radio allows the user to effectively save receiver current while maintaining adequate system performance. Attributes of the the LPPS feature are described as follows:

- When enabling the LPPS feature the demodulator sub-section of the receiver is duty-cycled at 50%
- The receiver current consumption is reduced approximately 4 mA
- Sensitivity performance decreases slightly due to time delta in the correlator while searching for an additional s0 before entering normal preamble search
- There is no added cost of using the LPPS feature since no extra components are required. In fact, enabling LPPS mode all the time has little system design impact and the current savings is 20%

MKW2xD LPPS Control Register Setup

There is one register that allows setup, control and enabling of the Low-Power Preamble Search (LPPS) feature:

- LPPS Control (Indirect Modem_LPPS_CTRL, 0x56) This is the control register and has this bit field definition: LPPS_EN [bit 0] (Master Enable for Low-Power Preamble Search (LPPS) mode)
- 1: Enable Low-Power Preamble Search mode
- 0: Normal operation

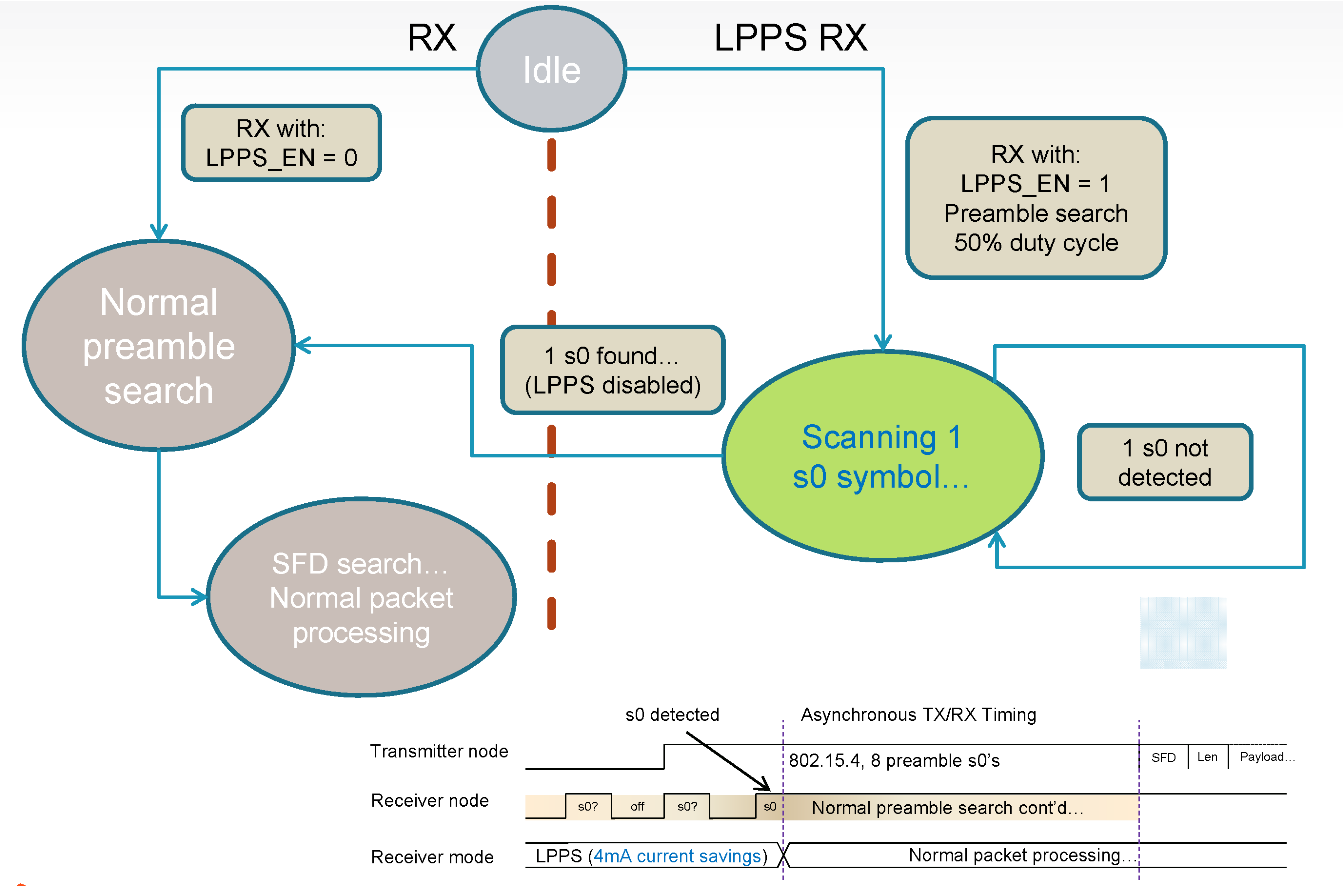


Figure 4-13. LPPS flow diagram

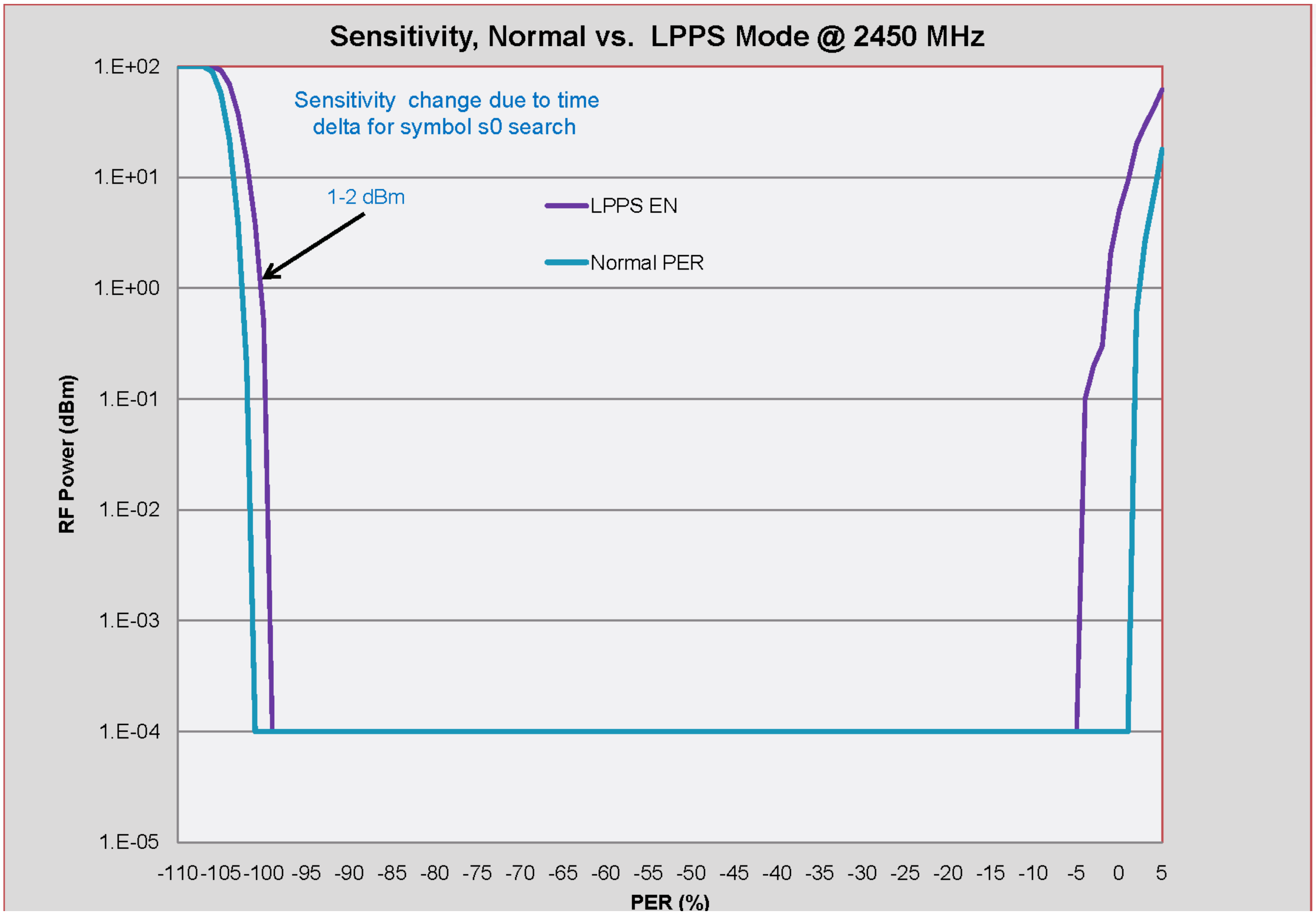


Figure 4-14. LPPS enabled sensitivity performance

Sensitivity Performance Summary

Normal (non-LPPS) PER mode

- Sensitivity in normal PER mode is -102 dBm as shown by the blue curve

LPPS enabled (purple curve)

- Sensitivity is slightly reduced by 1 - 2 dBm due to time delta the receiver requires to search for the s0 symbol defined by the mode
- The current is reduce by 4mA in LPPS mode with no additional system cost
- There is no discernible disadvantage of enabling LPPS all the time and taking advantage of the current savings

4.8.2 Modem Low Power States

The following table lists the modem low power states. There are three low power modes available:

- Off — Requires the modem reset RST_B input to stay asserted low. Lowest possible power and all functions are disabled. Digital GPIO default to inputs
- Hibernate — Has next lowest power, all hardware blocks deactivated, SPI register data are retained. Digital I/O retain their state
- Doze — Allows use of the Event Timer when active. The Event Timer can be used to cause a timed exit from Doze, and the CLK_OUT output can be kept active in Doze to provide a clock to the MCU. Doze uses considerable more current than Off or Hibernate. Register data are retained and digital I/O retain their state. Programming CLK_OUT to >4 MHz is not recommended in the Doze state.

Table 4-4. Modem Low Power States

Mode	Current(typ @ 2.7V)	Advantages	Disadvantages	Comment
Off	50 nA	Lowest power (leakage only)	Digital outputs tristated. All register data lost.	RST_B must remain asserted. All IC functions off.
Hibernate	1.0 uA	Register data retained. Digital outputs retain their states.		SPI used to exit state.
Doze[1]	500 uA (no CLK_OUT)	Crystal reference oscillator is on and CLK_OUT can be enabled. Timed exit is possible. Register data retained. Digital outputs retain their states.	Much higher current than Hibernate or Off.	Use SPI or timed exit to exit state. CLK_OUT can be kept active as a clock source.
Idle	700 uA	Fast transition to RX, TX, CCA	Much higher current than Doze or Hibernate	

[1] CLK_OUT frequency at default value of 37.786 kHz.

Although Idle is not considered a low power state, it is listed in Table 4-4 for comparison. It is also important to remember that all active states of RX, TX and CCA must be initiated from the Idle or Doze condition.

4.8.3 MCU Low Power Modes

The MCU supports numerous power modes. These are described in detail in MCU Power Management

4.8.4 Recovery Times from Low Power Modes

The Modem Mode of operation is controlled by the MCU. The modem may be powered down if it is not in use while the MCU is doing another task or while the whole node is "sleeping", i.e., the MCU is also powered down. Recovery time for both the modem and the MCU are important to system performance and the recovery times are independent of each other.

4.8.5 Run Time Current

As previously stated, the modem is fully under control of the MCU. For lowest power, the modem should be kept in a low power mode as much as possible. However, it is also important to understand the current profile of the modem under active operation. In a similar sense, the MCU can use varying levels of current depending primarily on the clock sources and frequencies used.

4.8.5.1 Modem Active Currents

In normal operational mode the modem's rest state in the Idle Mode. All active sequences originate from the Idle or Doze Mode and return to the Idle or Doze Mode. The three active sequences are Clear Channel Assessment (CCA), RX, and TX, and each has a separate current profile. The following table lists the typical currents while in the listed modes, but does not show the transition profiles when moving between modes.

Table 4-5. Modem Active State Currents

Mode	Current (typ @ 2.7V)
Idle (No CLK_OUT)	700 uA
Hibernate / Doze (No CLK_OUT)	<1uA / 500 uA
CCA/ED	19 mA
RX / RX LPPS Mode	19 mA, 15 mA LPPS
TX (0 dBm nominal output power)	17 mA

A normal sequence of events may include a 802.15.4 node performing first a CCA to see if the channel is clear, second transmitting a frame (assuming the channel is clear), and finally after the TX, going into to Receive Mode to look for an acknowledge. The modem has built-in complex autosequences to manage different operation modes.

4.8.5.2 MCU Active Current

The MCU active (run) current is dominated by the system clock frequency. Table 3-12 shows typical MCU supply current under several conditions. The Wait Mode is not running code but has the clocks active and has a baseline of about 0.56 mA. As the bus clock gets faster the supply current increases accordingly with a value of 1.1 mA at 1 MHz bus clock and 6.5 mA at 8 MHz bus frequency. A simple formula for estimating typical current vs. bus clock is:

Typical current = 0.56 mA + (bus clock frequency (MHz) x 0.77 mA/MHz)

A 16 MHz bus clock example would be:

Typical current @ 16 MHz = 0.56 mA + (16 MHz x 0.77 mA/MHz) = 12.9 mA

Another significant factor can be the ATD. The ATD has a typical enabled current of 0.7 mA. To get lowest power only enable the ATD as required to sample inputs.

Table 4-6. MCU Active Mode Supply Current

Condition	Current (typ @ 3.0V)
Run supply current[1] at CPU clock = 2 MHZ, fBus = 1 MHz	1.1 mA
Run supply current at CPU clock = 16 MHZ, fBus = 8 MHz	6.5 mA
Run supply current at CPU clock = 32 MHZ, fBus = 16 MHz	2.9 mA (calculated)
Wait supply current with fBus = 1 MHz	560 µA
ATD supply current when enabled	700 µA

[1] All MCU modules except ATD active, ICG configured for FBE, and no dc loads on port pins.

4.8.6 Configuration of Interconnected GPIO for Low Power Operation

The device GPIO must be configured properly for low power operation. The GPIO on the SIP are part of either the transceiver or the MCU, and their behavior is somewhat different. Also, some I/O are interconnected between chips onboard the device (see Internal Functional Interconnects and System Reset), and this affects their use during low power. There are generally the following low power scenarios:

- Hardware reset (RESET_b pin to the MCU is asserted low). NOT RECOMMENDED - this holds the MCU in reset, but not the transceiver because the transceiver reset is controlled by the MCU. This mode is not normally recommended as a long-term, “held” state.

- MCU active and transceiver held in reset. - If the transceiver is not used for an extended period, the transceiver can be held in reset (MCU port PTB19 asserted low). For lowest power in the transceiver:
    - SPI port pins SPI1_SCLK, SPI1_SOUT, and SPI1_SIN on the MCU should be driven to low, and SPI1_PCS0 should be driven to high.
    - Other interconnected MCU pins (PTC0, PTC1, and PTC3) should be driven to low
    - Transceiver GPIO1-GPIO2 should be held low externally through hardware.
- MCU active and transceiver programmed to Hibernate or Doze mode. - For lowest power in the transceiver:
    - SPI port pinsSPI1_SCLK, and SPI1_SOUT on the MCU should be driven to low, and SPI1_PCS0 should be driven to high. R_MISO pin on the transceiver should be programmed to drive low when SPI is disabled (R_SSEL_B is high)
    - Other interconnected MCU pins (PTC0, PTC1, and PTC3) should be driven to low
    - Transceiver GPIO1-GPIO2 should be programmed to outputs in the low state, and the unconnected transceiver GPIO6-GPIO8 driven low.
- MCU in low power (Stop2 or Stop3) and transceiver in Hibernate or Doze mode - These variations are most common and are better as GPIO on both chips retain their programmed state.
    - SPI port pinsSPI1_SCLK, and SPI1_SOUT on the MCU should be driven to low, and SPI1_PCS0 should be driven to high. R_MISO pin on the transceiver should be programmed to drive low when SPI is disabled (R_SSEL_B is high)
    - Other interconnected MCU pins (PTC0, PTC1, and PTC3) should be driven to low
    - Transceiver GPIO1-GPIO2 should be programmed to outputs in the low state, and the unconnected transceiver GPIO6-GPIO8 driven low.

4.8.7 General System Considerations for Low Power

Low power is most important for battery operated applications such as IEEE 802.15.4 End Devices. In the modem the highest instantaneous power is dominated by the CCA, RX and TX modes. In the MCU, the current draw is typically dominated by the clock frequency. A number of general recommendations can help the user:

- Clock Management on the MCU
    - Use clock management as required to lower required power
    - Run fast when the CPU performance is the critical path
    - Run slow when waiting on peripherals

  - Be aware of software performance requirements
  - Use the external clock option for lowest power if possible (modem CLK_OUT can supply frequencies as high as 32 MHz to the MCU)
- For long “sleep” periods between node activity and lowest power, use the Off condition on the modem, and use the LPTMR capability to wake up the MCU for its low power mode.
  - If the wake up time period does not need to be crystal accurate, the slow internal reference can be used. This has lowest power and lowest cost
  - If the wake up time period must be crystal accurate, use the RTC OSC with a 32.768 kHz crystal. This has the advantage of crystal accuracy and next lowest power with the disadvantage of higher cost.
  - If lowest cost is important and crystal accuracy is required for the wake up time period, one option is to use the modem Doze Mode with CLK_OUT active. CLK_OUT can be programmed for 32.787 kHz and is crystal accurate. Two disadvantages are a penalty of additional “sleep” current and the reference frequency is not exactly 32.768 kHz
- The MCU can go to a lower power mode such as Wait modem when the modem is recovering from the Off (up to 100 us) or the Hibernate modes
- When the MCU is initializing from a "cold start" POR condition, the modem reset must be driven to a low condition early in the routine. This is so the modem will start from a known condition and low power
- Profile the use scenario of both the modem and the MCU when they are active. This is required to estimate current usage versus time and mode of operation
- If possible put the modem in a low power mode, when the MCU is active and does not require interface with the modem. Doze can be useful to save power, provide CLK_OUT, and be ready with a quick recovery time
- Alternately put the MCU in a lower power mode if waiting on the modem IRQ
- Program all unused GPIO on both the modem and the MCU (including port signals not pinned-out) as outputs for lowest power
- The default condition is for modem R_MISO to go to tristate when R_SSEL_B is de-asserted. Program MISC_PAD_CTRL Register (Indirect Address 0x31), Bit 3, miso_hiz_en = 0 so R_MISO will be driven low when R_SSEL_B is de-asserted. As a result R_MISO will not float when Doze or Hibernate mode is enabled.

Chapter 5
Modem: Modes of Operation

5.1 Power Management Overview

The power management structure provides the biasing and the framework to power manage the MKW2xD device integrating the MCU with the modem.

The power management includes :

- A 1.8V digital regulator with POR
- A 1.8V analog regulator
- Biasing block

Provided in Figure 5-1 is the supply strategy block diagram.

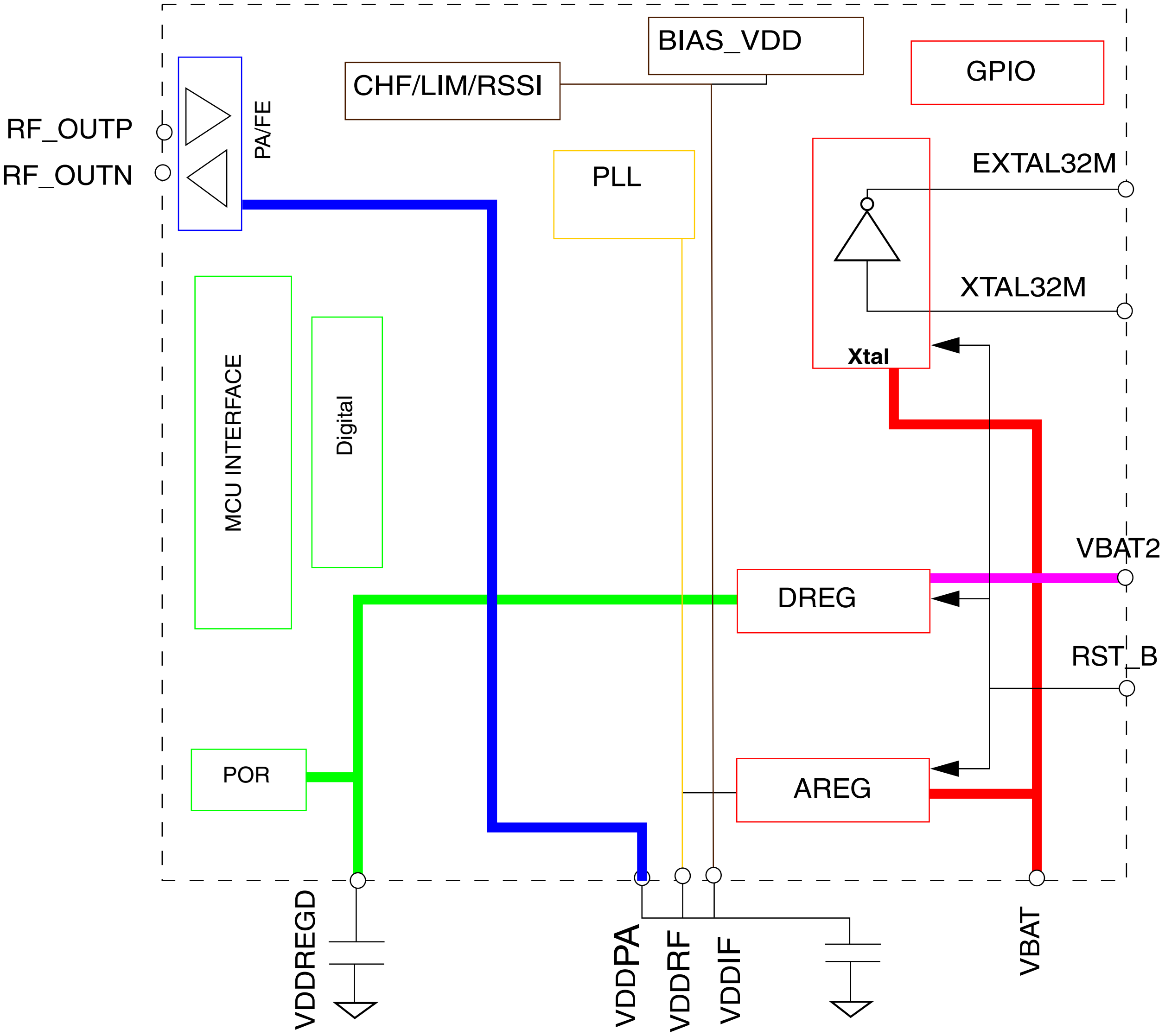


Figure 5-1. Supply strategy block diagram

5.1.1 Features

The modems power management structure has the following features:

- 1.8V voltage regulation for internal analog circuitry.
- Power on Reset (POR) for global digital reset.
- 1.8V voltage regulation for internal digital circuitry
- Centralized biasing

5.1.2 Power and Regulation Topology

The following blocks are sourced by the battery supply:

- XTAL
- Digital regulator 1.8 V
- Analog regulator 1.8 V
- The RF blocks are supplied by dedicated supplies:
- VDDPA for the PA that source high current at the Frontend (PA is not used at the same time)
- VDDRF for the PLL block (mainly VCO), clean supply regulated by the LDO directly
- VDDIF for the other analog blocks and also used potentially by the XOR.

VDDPA and VDDIF are connected externally such that decoupling between supplies is easy

5.1.3 Digital Regulator and POR

The digital regulator provides the 1.8 V supply to the digital logic. It is always turned on when the RST_B pin has been released. For VDDREGD, the external decoupling capacitor is in the range of 220 nF and 470 nF. As the figure below shows, the VDDREGD supply is decoupled externally. The digital regulator includes three modes of operation:

- Off mode - controlled by the active low RST_B pin, and active when reset is active
- On mode with maximum current capability - required in RUN mode and during startup
- Low power mode (in LP/HB and DOZE) - The current capability in this mode is limited but retains register contents and SPI functionality. The current consumption is max 300uA.

POR block provides reset if VBAT is low (</=1.4 V). POR comes out of reset if VBAT is high (>/=1.65 V). The POR is used to hold the digital logic in reset and then release the xtal oscillator when the power is going up.

Note

- The POR is disabled during loose mode. See Table 5-2.
- There is no battery monitoring

5.1.4 Analog Regulator (AREG)

The analog regulator is a low drop linear regulator that regulates to 1.8 V typical from the battery voltage 3.6 V-1.8 V. When the battery is low the regulated supply VDDRF follows the battery voltage (max 150 mV drop). This LDO provides the supply to the RF (including PA) and analog circuitry of the modem. The decoupling capacitor is in the range of 220 nF and 470 nF. As this figure shows, the three VDDA supplies (VDDRF, VDDIF and VDDPA) are decoupled externally.

5.2 Modem Operational Modes Summary

The modem has a number of passive operational modes that allow for low-current operation as well as modes where the transceiver is active. These modes are summarized in the following table.

Table 5-1. Modem Mode Definitions

Mode	Definition
Off	$\overline{\text{RST}}$ asserted. All IC functions Off, Leakage only.
Hibernate	Crystal Reference Oscillator Off.
Doze	Crystal Reference Oscillator On but CLK_OUT is available only if enabled
Idle	Crystal Reference Oscillator On with CLK_OUT output available.
Receive	Crystal Reference Oscillator On. Receiver On.
Transmit	Crystal Reference Oscillator On. Transmitter On.
CCA / Energy Detect	Crystal Reference Oscillator On. Receiver On.

5.2.1 Power modes

The RESET/power down mode is the mode used to turn everything off in the circuit. All the blocks supplied by VBAT are controlled directly by the RST_B pin for the power consumption of the circuit to be minimal.

Note

GPIO's must not be floating.

When RST_B is released, the digital regulator will automatically startup. When the digital supply is sufficiently high the POR signal is released. During startup, the RSTB high must be ensured for proper startup (pullup)

When POR is released, it starts automatically where the XTAL and a wake up interrupt is generated. The circuit is then set in IDLE mode.

All the other modes can be accessed through SPI generation by programming the blocks independently. For example, the RUN mode (TX or RX) is activated through the xcvseq setting. Care must be taken that before going to this mode the XTAL is enabled and the digital regulator is in full mode. The analog regulator will be automatically turned "on" then by the state machine.

The LP/hibernate mode is a specific mode where the XTAL is disabled. Going out of this mode via the SPI means that the XTAL needs to be started up again, the digital regulator forced to high power mode and a new wake up interrupt is generated.

Figure 5-2 provides the power state flow diagram for MKW2xD and Table 5-2 summarizes power mode and start up processes.

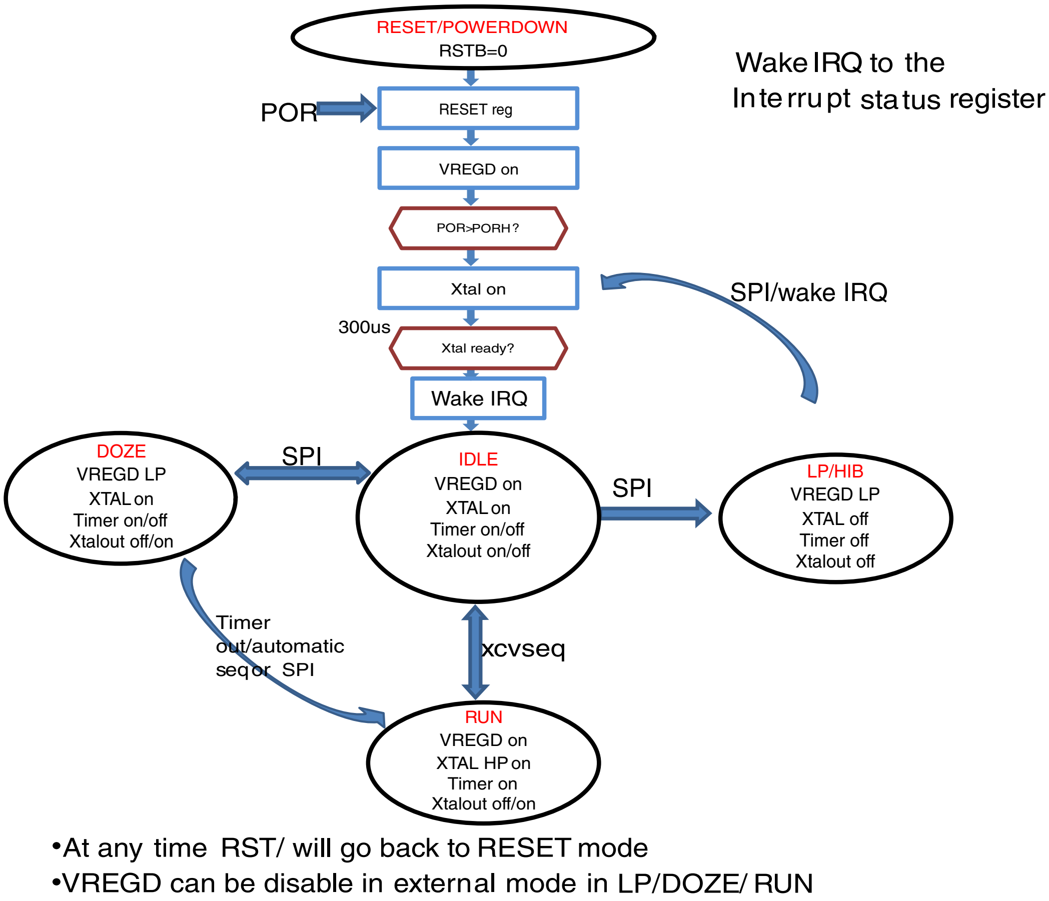


Figure 5-2. Power State Diagram

Table 5-2. Modes and start up process

Mode[1]	VREGD/POR	XTAL	VREGA	XTAL_out	Timer	Registers	GPIO	Current consumption
IDLE	ON	ON	OFF	Programmable	ON	Normal	Known	700uA, typ. (no clockout)
DOZE	Loose mode	ON	OFF	Programmable	ON/OFF	Retained	Known	500 uA, typ. (no clockout)
Low power / Hibernate	Loose mode	OFF	OFF	OFF	OFF	Retained	Known	<1uA[2]
Reset / Powerdown	OFF	OFF	OFF	OFF	OFF	Reseted	Known	<30nA
RUN[1]	ON	ON	ON	Programmable	ON	Normal	Known	15mA

1. Values identified in this table reflect nominal voltage and temperature.
2. Value does not include SPI activity.
1. Current of 15mA refers to radio current.

Note

> Providing a clock output signal on CLK_OUT will add some power consumption depending on the capacitive load on CLK_OUT to include the frequency.

5.3 Sequence Manager

The Sequence Manager (also known as the RF digital manager), is a hardware state machine that controls the timing of all transmit, receive, and CCA operations in the modem. The Sequence Manager interfaces directly with 3 lower level managers (RX digital manager, TX digital manager, and PLL digital manager). For each digital and analog element in the modem's transceiver, these 4 state machines operate collaboratively, to perform some or all of the following functions:

- activation
- deactivation
- initialization
- mode transitions
- calibration routines
- clock enabling and disabling
- interrupt triggering and status generation

Designed into the sequence manager are a variety of complex sequences (called autosequences), which are sequential concatenations of simpler, building-block sequences. The hardware autosequences provide the following advantages:

- smaller S/W memory footprint: H/W automates operations previously done by S/W
- increases MCU sleep times, reducing current consumption
- handles timing-critical sequences (rx-to-tx), e.g., data-polling, not possible in S/W

Sequences can be initiated by software directly, or alternatively, can be initiated automatically at the expiration of a hardware timer. The general parameters of the sequences can be programmed by software prior to initiating the sequences. Such parameters allow software to select:

- whether CCA is required ahead of a transmit sequence
- whether an automatic Ack frame should be transmitted after a receive sequence
- slotted vs. unslotted mode
- type of CCA (e.g., Energy Detect, CCA mode 2)
- whether the device is a PAN Coordinator or not
- whether the device resides on 2 networks simultaneously (Dual PAN mode)

The modem's Sequence Manager also includes support for Dual PAN mode, and Active Promiscuous mode.

5.3.1 Modes of Operation

The Sequence Manager controls, coordinates, and automates all transceiver sequences within the modem. Software selects the desired sequence, sets the general parameters, and (optionally) configures a hardware timer to trigger the sequence at a precise time in the future. Subsequently, the MCU can enter a low-power state, or tend to other activities, while the sequence manager state machine conducts the programmed transceiver operations. When the sequence completes, an interrupt alerts the MCU. At this point the MCU can check the completion status of the sequence, download received data from the packet buffer, and then prepare the next sequence. Sequences can be simple transceiver operations. For example, transmit one frame, or, perform a CCA on a channel. Alternatively, the sequence manager supports complex sequences, which use simple operations as building blocks, strung together in a back-to-back fashion, with precise timing intervals between operations. For example, a “TR” sequence calls for a transmit frame followed by a receive frame. For complex sequences, interrupts can optionally be generated at the completion of each stage of the sequence, providing greater visibility for the MCU into the timeline of the sequence. Software can determine the precise state of the sequence manager state machine at any time. Software can opt to abort a sequence at any time; when this happens, the sequence manager will return to its IDLE state in an orderly fashion.

The sequence manager is responsible for the implementation of the Wireless Physical Layer (PHY) specifications described in the IEEE 802.15.4 standard, in the modem. This is accomplished by activating and deactivating the key digital and analog elements in the transmit or receive chain, at precisely the right times, and for precisely the right durations. Most of the low-level timing parameters are hardwired within the state machine, and thus are not programmable. These low-level timing parameters are based on the specifications of the analog and digital blocks (e.g., PLL lock time), which are presumed to have been verified by the respective block designers using functional simulation, across all supported temperature and process corners. The time resolution of the sequence manager is 2us (1/8 of a transmitted symbol), so this is the granularity of the individual steps. The higher-level timing parameters, such as the interval between operations in a complex sequence, are largely programmable, except for those defined as constants in the IEEE 802.15.4 standard.

5.3.2 Functional Description

This section describes the supported sequences in more detail. All sequences can be initiated by software directly (instantaneously) by writing the desired sequence to the XCVSEQ register, with the TMRTRIGEN bit deasserted. Alternatively, software can schedule the initiation of a sequence at a precise time in the future, by programming the desired start time into the TC2 timer compare register, and setting the TMRTRIGEN bit. The XCVSEQ register field is writable and readable, so software can read this register to determine which sequence it had programmed earlier. After the sequence completes (SEQIRQ interrupt), software must set the XCVSEQ to IDLE (0x0), before initiating a new sequence. This prevents the undesired, inadvertent re-triggering of the sequence manager on the next TC2 match. If the sequence-complete interrupt (SEQIRQ) is masked, software can determine when the sequence is complete by polling the SEQ_STATE register, until it returns to idle (0x0). While a sequence is ongoing, software can abort the sequence at any time by writing IDLE (0x0) to XCVSEQ; the sequence manager state machine will respond with an orderly return to the SEQ_IDLE state, and a SEQIRQ will be generated. While a sequence is underway, software should not attempt to change XCVSEQ (other than aborting the sequence as previously described); the sequence manager will ignore all mid-sequence attempts to change XCVSEQ. As mentioned previously, software must write the IDLE value to XCVSEQ before changing sequences.

The supported sequences are shown in the following table.

XCVSEQ	Sequence	Description
0	I	Idle

Table continues on the next page...

XCVSEQ	Sequence	Description
1	R	Receive Sequence - conditionally followed by a TxAck
2	T	Transmit Sequence
3	C	Standalone CCA
4	TR	Transmit /Receive Sequence - transmit unconditionally followed by either an R or RxAck
5	CCCA	Continuous CCA - sequence completes when channel is idle
6	Reserved	
7	Reserved	

All sequences can be aborted by a PLL unlock detection. In the modem, this is referred to as PLL unlock. The unlock detection is qualified by the sequence manager; in other words, an unlock detection during the early stages of the PLL warmup is expected, and is not allowed to abort the sequence at that point. The primary rationale for the PLL unlock abort, is to prevent transmission on an incorrect, or unstable, frequency during a transmit operation, or to increase battery life by not prolonging a receive or CCA operation that is bound to fail. Software can disable the PLL unlock auto-abort altogether by asserting the PLL_ABORT_OVERRIDE bit.

5.3.2.1 Supported Sequences

5.3.2.1.1 Sequence (Idle)

When the IDLE value is written to XCVSEQ, the sequence manager goes to its SEQ_IDLE state. If not already in SEQ_IDLE, the sequence manager executes an orderly warmdown to return to SEQ_IDLE. A SEQIRQ interrupt is then issued. Writing IDLE value to XCVSEQ is the proper way to abort a sequence.

5.3.2.1.2 Sequence R (Receive)

Sequence R is the basic receive sequence. Sequence R can be used to receive all types of 802.15.4 PHY- and MAC-compliant frames, including reserved frame types. However, reception of Acknowledge frames is not recommended using Sequence R. This is because reception of an Acknowledge frame, usually follows a transmitted frame, with a designated Sequence Number, so that the received Acknowledge frame can be verified for the matching Sequence Number. In a standalone Sequence R, there is no Sequence Number to verify against, so any Acknowledge frame is merely transferred to the Packet

Buffer. (Note: the appropriate way to receive Acknowledge frames is with Sequence TR, with the RXACKRQD bit asserted). Using Sequence R, all frames that pass frame-filtering rules and CRC check, are transferred to the Packet Buffer.

The basic execution of a successful Sequence R is as follows:

- MCU writes TC2 compare value to the desired initiation time, and sets TMRTRIGEN (optional)
- MCU writes XCVSEQ=R
- Wait for TC2 match (if TMRTRIGEN=1)
- SEQ_MGR executes RxWarmup
- Preamble Detected
- SFD Detected
- FrameLength octet received, transferred to Packet Buffer
- N additional octets received (N = FrameLength), transferred to Packet Buffer. Note if (FrameLength > 127), then N = 127.
- If frame-filtering rules and CRC check pass
    - RXIRQ interrupt issued
    - SEQ_MGR executes RxWarmdown
    - If received Frame Control Field indicates AckRequest=1
        - SEQ_MGR executes TxWarmup
        - an Ack frame is transmitted using the received Sequence Number
        - SEQ_MGR executes TxWarmdown
- SEQIRQ interrupt issued

When Sequence R is initiated, the sequence manager warms up the receiver (analog and digital elements). This process takes 144 us. From this point forward, timer TMR3 can be enabled as a “bracketing” timer (software option). If a TMR3 timer match occurs, and software has asserted the TC3TMOUT bit, the sequence manager will warm down the receiver and return to SEQ_IDLE state. Assuming no TMR3 timeout, reception proceeds in preamble-search mode. After a preamble is detected, reception continues in SFD-search mode. At this point, the sequence manager’s slot timer is activated. This is because a slotted “transmit acknowledge” frame (TxAck) may be required later in this sequence. The slot timer is loaded with the following value (in us):

Slot Timer Preload = 160 + TXWARMUPTIME + (2*ACKDELAY).

The 160 term arises because, at the point of SFD detect, we are exactly 10 symbols into the backoff slot, and the symbol period is 16 us. (The slot timer timebase is actually 2 us, half the time-resolution of the equation shown above). The slot timer will count up and eventually rollover at 319, to 0. (There are 320 us in a backoff slot). The receiver receives the next octet (FrameLength). Starting with this octet, and continuing through the remainder of the frame, each octet that is received is transferred to Packet Buffer. After each octet is transferred to Packet Buffer, the Packet Buffer address is incremented by 1.

The next 2 octets (Frame Control Field) are then received, which are parsed to obtain the AckRequest field, and the FrameVersion field. The next octet is then received (SequenceNumber), and stored. The remainder of the frame, determined by FrameLength, is then received. As the octets are received, the packet data is parsed and subjected to packet-filtering rules. If any of the packet-filtering rules fail, or if the CRC check on the FCS field fails, the sequence manager will return the receiver to preamble-search mode, after the last octet has been received and transferred to Packet Buffer. Assuming the packet-filtering rules and CRC pass, an interrupt (RXIRQ) is issued to the MCU. At this point, the sequence manager must determine whether a TxAck is required. A TxAck is required if the following 3 conditions are all met:

- AUTOACK=1 (register bit)
- PROMISCUOUS=0 (register bit)
- The received FrameControlField has AckRequest=1

If these conditions are met, and SLOTTED mode is not in effect, the sequence manager loads its TxAck timer with the following value (in us):

TxAck Timer Preload = 192 - TXWARMUPTIME - 2*ACKDELAY + 16*ACKPOSTPONE. Note: ACKPOSTPONE is not supported in the modem and is hardwired to 0.

The 192 us is the non-slotted RX-to-TX turnaround requirement in the 802.15.4 standard. The TXWARMUPTIME takes into account the fact that the TxAck timer must expire early for the Tx warmup to be complete at exactly the 192 us point. The ACKDELAY is a signed, fine-tune adjustment to the TxAck transmission time (+/- 62 us), and the ACKPOSTPONE allows the TxAck to be delayed an additional 255 symbols, if necessary.

If SLOTTED mode is in effect, the slot timer was previously loaded (at SFD detect), but the sequence manager needs to determine how much time is left in the current backoff slot. If less than 192 us remain, the sequence manager must wait an additional backoff slot before initiating the Ack transmission. So it sets a flag indicating it must allow the slot counter to rollover an additional iteration.

Finally, at TxAck timer expiration (in non-SLOTTED mode), or at the qualified rollover of the slot timer (in SLOTTED mode), a Tx warmup is initiated. The conclusion of the Tx warmup will coincide precisely with a backoff slot boundary if SLOTTED mode is in effect. The sequence manager will then initiate the transmission of the Ack frame. The entire Ack frame will be built up by hardware; the Packet Buffer is not used for the auto-TxAck feature. The hardware will insert the following parameters into the Ack frame’s Frame Control Field:

Table 5-3. Frame Control Field for Hardware-generated Auto-TxAck Frame

FCF Field	Value
FrameType	Acknowledge
SecurityEnabled	0
FramePending	If SRCADDR_EN=1, FramePending gets the value of FramePendingFlag from Source Address Matching accelerator hardware. If SRCADDR_EN=0, FramePending gets the value of ACK_FRM_PND.
AckRequest	0
PanIDCompression	0
Reserved (bits 7-9)	000
DstAddrMode	0
FrameVersion	copy FrameVersion from the previously-received frame
SrcAddrMode	0

The Sequence Number for the Ack packet, will be the Sequence Number obtained from the previously received frame.

After the Ack frame has been transmitted, the sequence manager will issue a TXIRQ interrupt, and then perform a Tx warmdown. Finally, a SEQIRQ interrupt will be issued, and the sequence manager returns to SEQ_IDLE state.

5.3.2.1.3 Sequence T (Transmit)

Sequence T is the basic, standalone transmit sequence. Sequence T can be used to transmit all types of 802.15.4 PHY- and MAC-compliant frames. However, transmission of Acknowledge frames is not recommended using Sequence T. This is because transmission of an Acknowledge frame, usually follows a received frame, with a designated Sequence Number. The sequence manager automates the transmission of an Ack frame that follows a received frame, using Sequence R with the AUTOACK bit asserted. This is the recommended method for transmitting an Ack frame. This is especially beneficial, because the transmitted Ack frame octets are generated by hardware, and so no setup of a transmit packet in Packet Buffer, is required. However, sequence manager hardware does not prohibit Ack frames using Sequence T.

Sequence T allows for the insertion of 1 or 2 CCA (clear channel assessment) measurements prior to transmission, to ensure that the selected channel is idle. The CCA-before-TX feature is governed by two register bits, CCABFRTX and SLOTTED. All CCA measurements must indicate channel-idle, in order for the sequence manager to proceed to transmission; if the channel is determined by CCA to be busy, the sequence manager will terminate the sequence without a transmission. The number of CCA measurements attempted by the sequence manager, and the timing of the measurements, is shown in the following table.

CCABFRTX	SLOTTED	Number of CCA measurements	Timing of Initiation of CCA measurement #1	Timing of Initiation of CCA measurement #2	Timing of First Bit of Transmitted Frame (assumes channel idle)
0	X	0	-	-	Immediately follows TxWarmup
1	0	1	Immediately follows RxWarmup	-	Immediately follows RxPartialWarmdown and TxPartialWarmup
1	1	2	Immediately follows RxWarmup	320 us after Initiation of CCA measurement #2	320 us after Initiation of CCA measurement #2

Any CCA mode can be used for Sequence T and Sequence TR (CCA modes 1, 2, and 3 are available). Software must configure the CCATYPE bits prior to initiating the autosequence, to select the CCA mode.

The basic execution of a successful Sequence T is as follows:

- MCU writes TC2 compare value to the desired initiation time, and sets TMRTRIGEN (optional)
- MCU sets CCABFRTX if one or more CCA's are required prior to transmi
- MCU sets SLOTTED if in slotted mode (2 CCA's will be attempted)
- MCU writes XCVSEQ=T
- Wait for TC2 match (if TMRTRIGEN=1)
- If CCABFRTX=1 :
  - SEQ_MGR executes RxWarmup
  - SEQ_MGR initiates CCA measurement (takes 128us)
  - if CCA indicates channel busy, sequence terminates, CCAIRQ & SEQIRQ issued
  - otherwise, if CCA indicates channel idle:
    - if SLOTTED=0, SEQ_MGR executes a RxPartialWarmdown
    - if SLOTTED=1, SEQ_MGR initiates a 2nd CCA, 320 us after initiating 1st CCA.
    - if SLOTTED=1, and channel busy, sequence terminates, CCAIRQ & SEQIRQ issued
    - if SLOTTED=1, and channel idle, SEQ_MGR executes a RxPartialWarmdown
- if SLOTTED=1, SEQ_MGR waits until 320 ms elapse after 2nd CCA initiation
- SEQ_MGR executes TxPartialWarmup
- Preamble transmitted
- SFD transmitted
- FrameLength octet is obtained from Packet Buffer Address 0, and transmitted

- CRC engine is reset
- N additional octets are obtained from Packet Buffer and transmitted (N = FrameLength - 2)
- Each octet that is transmitted, is shifted through the CRC engine
- After Nth octet transmitted, FCS is available from CRC engine.
- FCS lower octet transmitted
- FCS upper octet transmitted
- SEQ_MGR issues TXIRQ
- SEQ_MGR executes TxWarmdown
- SEQ_MGR issues SEQIRQ

When Sequence T is initiated, the sequence manager first must determine whether one or more CCA measurements are required. If CCABFRTX bit is asserted, the sequence manager warms up the receiver (analog and digital elements). This process takes 144 us. Immediately after the conclusion of the warmup, a CCA measurement is initiated. The CCA measurement takes 128 ms. If CCA indicates the channel is busy, the sequence manager warms down the receiver, issues CCAIRQ and SEQIRQ interrupts, and terminates the sequence. If, however, CCA indicates the channel is idle, the sequence manager inspects the SLOTTED bit to determine what to do next. If the SLOTTED bit is clear, the sequence manager issues a CCAIRQ interrupt, and then executes a partial warmdown of the receiver. If, however, the SLOTTED bit is set, the sequence manager does not issue a CCAIRQ interrupt; instead, the sequence manager initiates a second CCA measurement, precisely 320 us after the initiation of the first CCA. This 320 us interval is timed by the slot timer, which had been preloaded with 320 us at the instant that Rx warmup was complete. At the conclusion of the second CCA, the sequence manager issues a CCAIRQ. If the second CCA indicates the channel is busy, the sequence manager warms down the receiver, issues a SEQIRQ interrupt, and terminates the sequence. If, however, CCA indicates the channel is idle, the sequence manager executes a warmdown of the receiver. Prior to executing the warmdown, the sequence manager reloads the slot timer, in order to precisely schedule the ensuing transmission. This time, the slot timer is loaded with a smaller value, in order to account for the fact that the timer must expire early in order to trigger the Tx warmup, which must complete at precisely the next backoff slot boundary. Thus, this time the slot timer is preloaded with:

Slot Timer Preload = 320 - TXWARMUPTIME + (2*TXDELAY).

The 320 us is the duration of the backoff slot. The TXWARMUPTIME takes into account the fact that the TxAck timer must expire early for the Tx warmup to be complete at exactly the 320 us point. The TXDELAY is a signed, fine-tune adjustment to the TxAck transmission time (+/- 62 us).

If the SLOTTED bit is asserted, after the slot timer expires, the sequence manager executes a Tx warmup. The conclusion of this warmup coincides with the backoff slot boundary. If the SLOTTED bit is deasserted, the sequence manager does not wait for the slot timer, and instead executes the partial Tx warmup immediately after the partial Rx warmdown.

At this point, the sequence manager begins the transmit operation. The preamble and SFD are transmitted first. The preamble and SFD octets are generated by hardware, and are not included in the Packet Buffer. Then, the FrameLength octet is obtained from the Packet Buffer (address 0), and the buffer address pointer is incremented by one. The FrameLength octet is transmitted. The CRC engine is reset at this point. The FrameLength octet is used to determine how many more octets remain in the Packet Buffer. The number of remaining octets is:

Number of octets remaining in Tx buffer = FrameLength - 2

This is because the final 2 octets of the transmit frame constitute the FCS (Frame Check Sequence), and this value is hardware-computed. Thus, FCS is not part of the Tx buffer. Each of the octets in the Tx buffer is transmitted, and simultaneously shifted through CRC, to generate the FCS field. After the Tx buffer is drained and all of its octets transmitted, the 2 FCS octets are transmitted, starting with the least significant octet. At this point, the transmit operation is complete. The sequence manager issues a TXIRQ interrupt. The sequence manager then executes a complete Tx warmdown, and follows up with a SEQIRQ interrupt. The sequence manager returns to SEQ_IDLE state.

5.3.2.1.4 Sequence C (Standalone CCA)

Sequence C is the standalone CCA sequence. During Sequence C, the sequence manager executes a Clear Channel Assessment (CCA). The result of the CCA measurement is reported back to software in the CCA bit of the IRQSTS2 Register. The CCA bit indicates either a busy channel (1), or an idle channel (0).

The basic execution of a Sequence C is as follows:

- MCU writes TC2 compare value to the desired initiation time, and sets TMRTRIGEN (optional)
- MCU writes XCVSEQ=C
- Wait for TC2 match (if TMRTRIGEN=1)
- SEQ_MGR asserts one of {cca1_en, cca2_en, cca3_en, ed_en} to CCA_DIG
- SEQ_MGR executes RxWarmup
- SEQ_MGR waits an additional 26us for RSSI/AGC settling
- SEQ_MGR initiates CCA measurement by asserting rx_cca_en to CCA_DIG
- SEQ_MGR waits for CCA to complete
- CCA bit is set to the CCA status (1=busy 0=idle)

- CCAIRQ interrupt issued
- SEQ_MGR executes RxWarmdown
- SEQIRQ interrupt issued

When Sequence C is initiated, the sequence manager asserts one of 4 control signals to CCA_DIG, based on which CCA or ED mode is programmed into CCATYPE. The mapping of CCATYPE to control signal is shown in the following table:

CCATYPE	CCA Control Signal
00	ed_en
01	cca1_en
10	cca2_en
11	cca3_en

The asserted control signal will remain asserted for the duration of the Sequence C. The sequence manager warms up the receiver (analog and digital elements). This process takes 144 us. The TC3 “bracketing” timer has no effect on the sequence manager during Sequence C, so it cannot abort the sequence. At the completion of the warmup, the sequence manager initiates the CCA by asserting the rx_cca_en signal to CCA_DIG. While CCA is enabled, the CCA_DIG measures and averages the energy or signal on the channel. At the completion of the CCA process, the CCA bit is set to the status of the channel (idle or busy). The CCAIRQ is issued, the receiver is warmed down, the SEQIRQ is issued, and the sequence manager returns to SEQ_IDLE state.

Any CCATYPE setting may be used for Sequence C.

5.3.2.1.5 Sequence TR (Transmit/Receive)

Sequence TR is a combination Transmit/Receive sequence. The sequence is executed as a concatenation of 1 transmit operation followed by 1 receive operation, with a minimum TX-to-RX turnaround time in between. There are 2 permutations of Sequence TR, depending on the RXACKRQD bit. For both permutations, the Sequence T that constitutes the first half of a Sequence TR, is identical to the standalone Sequence T (XCVSEQ=2). This means that the transmit operation can be slotted or unslotted, and can be preceded by 1 or 2 CCA measurements, depending on the state of the CCABFRTX and SLOTTED bits.

If RXACKRQD=0, then Sequence TR is executed as a Sequence T followed by a Sequence R, with a minimum TX-to-RX turnaround time in between. The Sequence R, which constitutes the second half of the Sequence TR, is identical to the standalone Sequence R (XCVSEQ=1). This means that the receive operation can be followed by an automatic, hardware-generated transmit Acknowledge frame, if all the necessary conditions are met. As with basic Sequence R, data for a successful receive operation is always transferred to Packet Buffer.

If RXACKRQD=1, then Sequence TR is executed as a Sequence T followed by a Receive-Acknowledge-Only frame. This type of receive operation is special, and not the same as a standalone Sequence R. The Ack-only receive operation filters all incoming frames, looking only for an Acknowledge frame whose Sequence Number matches the Sequence Number that was transmitted in the Sequence T portion of the sequence. All non-matching frames are discarded, and after each non-matching frame, the sequence manager will return the receiver to preamble-detect mode. This receive operation will continue until the matching Ack frame is received. Because this is a Receive-Acknowledge-Only operation, and not a Sequence R, there will be no receive octets transferred to Packet Buffer (an exception is made if ACTIVE_PROMISCUOUS=1; if so, all receive octets are transferred to the Packet Buffer, even for the Receive-Acknowledge-Only frame). The sequence will end with a RXIRQ interrupt, which indicates that the matching Ack frame was successfully received.

Needless to say, if during the transmit operation of a Sequence TR, an Acknowledge frame is being requested of the receiving end device (Frame Control Field: AckRequest=1), then Sequence TR with RXACKRQD=1 is the appropriate procedure. (Using Sequence TR with RXACKREQ=0 is not recommended for this scenario, because there is no Sequence Number matching.)

Conversely, if during the transmit operation of a Sequence TR, an Acknowledge frame is not being requested of the receiving end device (Frame Control Field: AckRequest=0), then Sequence TR with RXACKRQD=1 should never be used, because a receive acknowledge frame will not be forthcoming. Instead, use Sequence TR with RXACKRQD=0, or simply Sequence T, if no followup receive frame is expected.

5.3.2.1.6 Sequence CCCA (Continuous CCA)

Sequence CCCA is the Continuous CCA sequence. This sequence is designed to accommodate situations where channel availability may be infrequent, or low-duty-cycle, and a device needs to be able to transmit at the earliest available opportunity. During Sequence CCCA, the sequence manager repeats CCA measurements continuously until a channel-idle condition is found. The interval between the end of one CCA measurement, and the start of the next, is 126us. The actual interval between consecutive CCA measurements (i.e., from measurement-start to measurement-start), in the CCCA sequence, is 126 + 128 = 254us. After each CCA iteration, the CCA bit (IRQSTS2 Register) is set to the result of the measurement. As long as the CCA bit is high (busy) at the end of the iteration, the sequence manager will initiate a new measurement. Unlike Sequence C (standalone CCA), there is no CCAIRQ interrupt generated at the end of each CCA measurement, until the channel-idle condition is detected, at which point CCAIRQ is issued. The SLOTTED bit has no effect on this sequence. A Sequence CCCA will be aborted by the Sequence Manager if a TMR3 match occurs, and TC3TMOUT=1. CCA modes 1, 2, or 3, can be used for Continuous CCA

The basic execution of a Sequence CCCA is as follows:

- MCU writes TC2 compare value to the desired initiation time, and sets TMRTRIGEN (optional)
- MCU writes XCVSEQ=CCCA
- Wait for TC2 match (if TMRTRIGEN=1)
- SEQ_MGR asserts one of {cca1_en, cca2_en, cca3_en} to CCA_DIG
- SEQ_MGR executes RxWarmup
- SEQ_MGR waits for 26 us for RSSI/AGC settling
- SEQ_MGR initiates CCA measurement by asserting rx_cca_en to CCA_DIG
- SEQ_MGR waits for CCA to complete
- CCA bit is set to the CCA status (1=busy 0=idle)
- If CCA=1, initiate a new CCA measurement immediately and repeat the previous 4 steps
- If CCA=0, CCAIRQ is issued
- SEQ_MGR executes RxWarmdown
- SEQIRQ interrupt issued

The CCA bit (IRQSTS2 register) may change dynamically during the CCA measurement, and are considered valid (and sampled by the Sequence Manager), only at the end of the CCA measurement; therefore the user should not continuously poll CCA during CCCA and be expecting continuously-valid results

The user can continuously monitor progress of the Sequence CCCA by reading the CCCA_BUSY_CNT register.

5.3.2.2 Sequence Manager Warmup Timing

The Sequence Manager not only controls autosequence timing, but also controls the warmup and warmdown timing of the digital and analog transceiver elements. The Sequence Managers executes warmups, in concert with the 3 lower-level managers:

- RX Digital Manager, or RSM
- TX Digital Manager, or TSM
- PLL Digital Manager, or PSM

For each low-level manager, the Sequence Manager has a single enable. The 3 manager-enables are:

- RXON (activates the RSM)
- TXON (activates the TSM)
- PLLON (activates the PSM)

After activation, the lower level managers take responsibility for carrying out the enabling and calibration of the individual transceiver blocks, both digital and analog. The lower level managers remain in their active states for as long as the Sequence Manager holds the corresponding manager-enable high. Normally, the Sequence Manager will hold the signals high for the duration of an operation (i.e., a packet reception, or a CCA measurement as part of a larger autosequence). After the Sequence Manager deasserts a manager enable, the lower-level manager shall return to its "off" or idle state, within 2us. In the case of an abnormal termination (i.e., a software abort, PLL unlock abort, or a TMR3 timeout), the Sequence Manager will deassert all manager enables, and all 4 state machines will return to idle. For aborts that occur during TX sequences, the Sequence Manager will perform a staged warmdown, to ensure the PLL and other analog elements remain stable while the PA gradually powers off. Aside from enabling and disabling of the lower-level managers by the top-level Sequence Manager, the 3 lower-level managers run autonomously of the top-level manager; there is no feedback, or other information, conveyed from the satellite managers to the Sequence Manager. All 4 managers run synchronously, using the same 2MHz clock. There are no clock domain crossings amongst them.

5.3.2.3 Dual PAN Mode Sequencing

The Sequence Manager supports Dual PAN mode, which allows the modem to reside on 2 networks simultaneously, switching back and forth between the 2 networks under software, or hardware control.

In Automatic Dual PAN mode, the modem must switch from one network to another under hardware control, at a software-programmed rate. Because the 2 networks often occupy different channels (frequencies), the sequence manager supports an "on-the-fly channel change", whereby, during a Sequence R, in preamble-search state (RX_PRE), the sequence manager will toggle PLLON and switch to a new set of programmed channel-select registers, wait out a PLL settling time, and then resume preamble search on the new channel. The entire on-the-fly channel change consumes 68us, and 2 sequence manager states. The first state (RX_PAN1) deasserts PLLON and switches the PLL_INT and PLL_FRAC outputs to PHY_DIG to select the new PAN's channel. The RX_PAN1 state also deasserts pll_lock_en so that unlock events that would naturally occur during the channel-change process, will not trigger a PLL_UNLOCK_IRQ or abort a sequence. The duration of RX_PAN1 is 4us. The Sequence Manager then transitions to RX_PAN2, which re-asserts PLLON. This initiates the PLL on the new channel, and PLL calibration and locking begins (controlled by PSM). The pll_lock_en remains deasserted during RX_PAN2 so that unlock events continue to be ignored. The duration of RX_PAN2 is 52us. After RX_PAN2, the Sequence Manager returns to preamble-search (RX_PRE). Note: the RSM takes a few microseconds beyond this, to return to its preamble-search state, bringing the total “on-the-fly channel change” duration to 68us. This 68us represents a short “blind spot” during which preamble detection can’t occur, while the modem switches from one network to another. The following diagram depicts Dual PAN “on-the-fly channel change”.

5.4 Dual PAN ID introduction

The modem represents a high performance platform/radio that includes hardware support for a device to reside in two networks simultaneously. In optional Dual PAN (Personal Area Network) mode, the device will alternate between the two (2) PANs under hardware or software control. Hardware support for Dual PAN operation consists of two (2) sets of PAN and IEEE addresses for the device, two (2) different channels (one for each PAN), a programmable timer to automatically switch PANs (including on the fly channel changing) without software intervention. There are control bits to configure and enable Dual PAN mode and read only bits to monitor status in Dual PAN mode. A device can be configured to be a PAN coordinator on either network, both networks or neither.

For the purpose of defining "PAN" in the content of Dual PAN mode, two (2) sets of network parameters are maintained. In this chapter, "PAN0" and "PAN1" will be used to refer to the two (2) PANs. Each parameter set uniquely identifies a PAN for Dual PAN mode. These parameters are described in the following table.

Table 5-4. PAN0 and PAN1 descriptions

PAN0	PAN1
Channel0 (PHY_INT0, PHY_FRAC0)	Channel1 (PHY_INT1, PHY_FRAC1)
MacPANID0 (16-bit register)	MacPANID1 (16-bit register)
MacShortAddrs0 (16-bit register)	MacShortAddrs1 (16-bit register)
MacLongAddrs0 (64-bit registers)	MacLongAddrs1 (64-bit registers)
PANCORDNTR0 (1-bit register)	PANCORDNTR1 (1-bit register)

During device initialization if Dual PAN mode is used, software will program both parameter sets to configure the hardware for operation on two(2) networks.

5.4.1 Manual and Automatic Modes

Two (2) modes are available for Dual PAN operation:

- Manual
- Automatic

5.4.1.1 Manual Dual PAN Mode

In manual Dual PAN mode software controls which PAN is selected for transmission and reception at all times. Software populates both PAN network parameter-set-registers listed in Table 5-4 where software selects a PAN to begin transmission of reception. A single control bit ACTIVE_NETWORK selects the PAN. To select PAN0, ACTIVE_NETWORK should be set to 0; to select PAN1, ACTIVE_NETWORK should be set to 1. Software then initiates an auto sequence by writing the appropriate value to XCVSEQ. At any point software can elect to switch to the alternate PAN. To switch, software aborts the sequence (if one is active) by writing XCVSEQ = IDLE. Software then toggles ACTIVE_NETWORK and restarts a new sequence with XCVSEQ. Software is responsible for aborting and initiating sequences. Hardware is responsible for warming up the transceiver on the selected channel (CHANNEL0 or CHANNEL1) and performing address-filtering on the selected network's address parameters. (PANCORDNTR, MacPanID, Short Address and/or IEEE address).Software overhead is significantly reduced in this mode because there is no need to re-program all of the PAN-select registers every time the PAN is toggled in Dual PAN mode where only a single bit (ACTIVE_NETWORK) needs to be written. For manual Dual PAN operation, DUAL_PAN_AUTO should be set to 0.

The diagram shown in the next figure depicts Dual PAN operation in manual mode. In this example, software populates both PAN parameter sets during initialization. Also, software elects to initiate packet reception on PAN1, so ACTIVE_NETWORK is set to 1. Software initiates the Sequence R by writing to XCVSEQ. Hardware warms up the receiver on CHANNEL1 and enters preamble-search. In this example no packet is received but a data request from higher level software requires a switch to PAN0. Software responds by aborting the reception by writing Sequence 1 to XCVSEQ. Software then sets ACTIVE_NETWORK to 0 to select PAN0. In the example, software initiates a Sequence TR. The hardware warms up the receiver for CCA on CHANNEL0, performs CCA (channel is idle), warms down the receiver and warms up the transmitter on CHANNEL0. Then, transmits the requested packet, warms down the transmitter and warms up the receiver again (as per the auto sequence) on CHANNEL0 in order to receive the acknowledge packet. The acknowledge packet is received and the auto sequence completes successfully with the RXIRQ and SEQIRQ. Software elects to resume packet reception on PAN1) which was in progress before the data request). Software toggles ACTIVE_NETWORK back to 1 and re-initiates Sequence R.

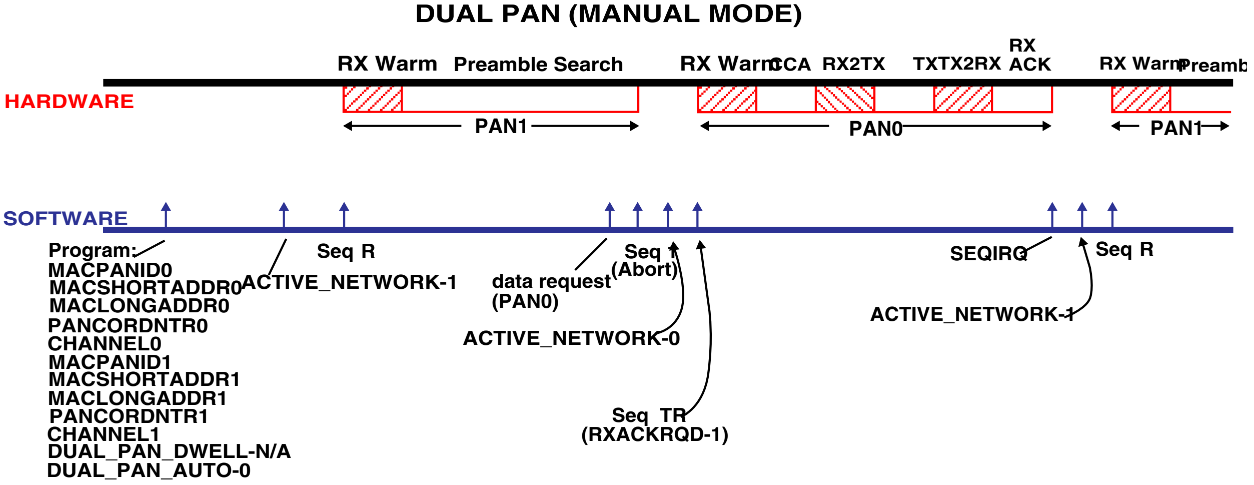


Figure 5-3. MKW2xD Dual PAN (Manual Mode) hardware / software diagram

5.4.1.2 Auto Dual PAN

In automatic Dual PAN mode hardware is responsible for alternating the selected PAN at a pre-programmed interval. Software programs a "PAN dwell time", which determines the amount of time during which the hardware will receive on a given PAN. This is done by programming the DUAL_PAN_DWELL register. Software sets the DUAL_PAN_AUTO bit to "1" for automatic Dual PAN operation and sets ACTIVE_NETWORK to correspond to the PAN on which it desires packet reception to begin. Software then initiates a Sequence R. Hardware warms up the receiver on the selected PAN and begins preamble search. When the dwell time expires, assuming preamble and SFD have not been detected, hardware will switch to the alternate PAN including on-the-fly channel change. After it is switched to the alternate PAN, hardware will use the address filtering parameters (PANCORDNTR, MacPanID, Short Address and/or IEEE) programmed for the alternate PAN. After the dwell time expires on the alternate PAN, assuming preamble and SFD have not been detected, hardware will switch back to the original PAN. This process will repeat indefinitely until one of the following occurs:

1. A packet is successfully received (address match and CRC valid)
2. Software aborts the Sequence R
3. A TMR3 RX timeout which automatically aborts the Sequence R.

4. A PLL unlock automatically aborts the Sequence R.

If the hardware PAN dwell timer expires during a packet reception (i.e., preamble and SFD have been detected during a Sequence R), the PAN-switch will be delayed until the packet is completely received. If the packet was received successfully (address match and CRC valid), the receive sequence will terminate with RXIRQ and SEQIRQ (if an auto-Acknowledge was requested and enabled, the sequence will complete after the Acknowledge packet was transmitted, and TXIRQ will also be set, in addition to RXIRQ and SEQIRQ); the PAN switch will occur at this point, after the Sequence has returned to IDLE. If the received packet fails FCS (CRC check), or, if the packet was not intended for the device (address mismatch), the hardware will toggle the PAN, and return to preamble-search on the new PAN.

If the hardware PAN dwell timer expires during a state *other* than RX preamble search during Sequence R (i.e., if expiration occurs during TX, CCA, ED, Sequence TR, or any warm-up or transitional state), PAN-switch will be delayed until either:

1. Preamble Search state is reached during a Sequence R (*not* Sequence TR)
2. Sequence returns to IDLE.

At any point, software can elect to interrupt the automatic Dual PAN operation, by aborting the sequence with a XCVSEQ=IDLE, and the setting the DUAL_PAN_AUTO bit to 0. In response, hardware will freeze the current state of the dwell timer, and capture the currently-selected PAN, so that hardware-controlled PAN alternation can resume where it left off, later. Software can then tend to that tasks that caused the Dual PAN interruption. When software wants to resume Dual PAN operation, it then sets DUAL_PAN_AUTO=1, and re-initiates Sequence R. Hardware will resume the dwell timer from the point it was frozen earlier, on the same PAN, and will warm-up the receiver to receive on the PAN that was being monitored before the interruption. Hardware will dwell on that PAN until the timer expires, and then switch PAN's (assuming no preamble and SFD detected, subject to the conditions listed above). Software can always determine the selected PAN at any time, in any mode, by reading CURRENT_NETWORK bit. If DUAL_PAN_AUTO=0, then CURRENT_NETWORK = ACTIVE_NETWORK. If DUAL_PAN_AUTO=1, then CURRENT_NETWORK is controlled by hardware.

To initiate auto Dual PAN operation for the first time, or to reset the dwell timer to its programmed value and allow software to select the initial PAN for auto Dual PAN operation, software should program the desired initial PAN to ACTIVE_NETWORK (0 for PAN0, 1 for PAN1), and then program the desired dwell time into the DUAL_PAN_DWELL register. Hardware always responds to a write to DUAL_PAN_DWELL, by resetting the dwell timer to its programmed value, and selecting the initial PAN for auto Dual PAN operation, based on the state of ACTIVE_NETWORK. A write to DUAL_PAN_DWELL, always re-initializes the DWELL TIMER to the programmed value. If a write to DUAL_PAN_DWELL occurs during an auto sequence, the DWELL TIMER will begin counting down immediately. If a write to DUAL_PAN_DWELL occurs when there is no auto sequence underway, the DWELL TIMER will not begin counting until the next auto sequence begins; it will begin counting at the start of the auto sequence warm-up. Under all circumstances, DUAL_PAN_AUTO must be asserted to allow the DWELL TIMER to count. If DUAL_PAN_AUTO=0, the DWELL TIMER will freeze and hold its state, until DUAL_PAN_AUTO=1. However, a write to DUAL_PAN_DWELL will still initialize the DWELL TIMER to its programmed value, regardless of the state of DUAL_PAN_AUTO.

At any point during auto Dual PAN mode, software can ascertain the PAN currently being serviced by hardware by reading the CURRENT_NETWORK bit. Software can also determine the time remaining in the current dwell by reading the DUAL_PAN_REMAIN register.

In the auto Dual PAN mode, software retains responsibility for initiating and aborting sequences. There is no provision for hardware initiated sequences. TMR2 may be used to initiate an auto sequence at a future time. TMR3 may be used as an RX hardware abort mechanism.

The following figure depicts Dual PAN operation in automatic mode. In this example software populates both PAN parameter sets during initialization. Software wants to initiate packet reception in auto Dual PAN mode on PAN1, so ACTIVE_NETWORK is set to "1". For auto Dual PAN operation software programs the desired PAN-switching interval into DUAL_PAN_DWELL register, in this case 4 ms. Software initiates the Sequence R by writing to XCVSEQ. The dwell timer begins counting as soon as the receiver warm-up begins. Hardware warms up the receiver on CHANNEL1 where PAN was selected as the initial PAN shown in Figure 5-4 and begins preamble search. If a packet is received during this interval, hardware applies PAN1 address filtering. No preamble is detected at the point where the dwell timer expires so hardware executes a PAN-switch to PAN0 that includes an on-the-fly-change. On-the-fly changes require approximately 68 us to execute and no preamble detection is possible during this "blind spot". After the PAN switch, hardware assumes preamble search on CHANNEL0 and applies PAN0 address filtering if a packet is received. No preamble is detected at the point which the dwell timer expires, therefore hardware performs a PAN-switch back to PAN1 and so on and so forth.

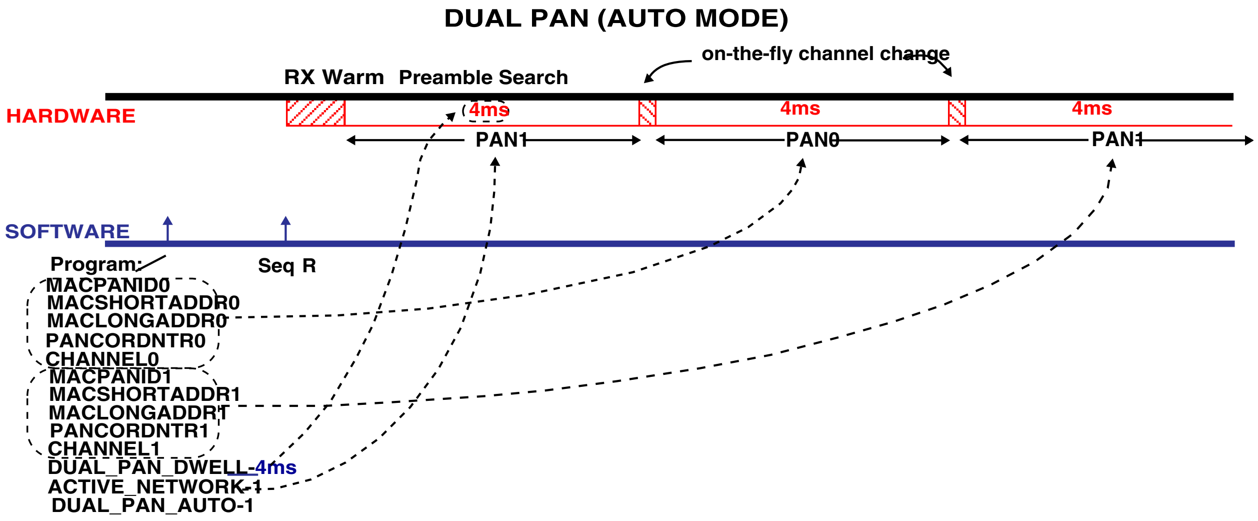


Figure 5-4. MKW2xD Dual PAN Auto mode diagram

5.4.1.3 Two-PAN-one-channel

A special case occurs if a device is operating in Dual PAN mode (either Auto or Manual mode), and both PAN's occupy the same channel; in other words, CHANNEL0=CHANNEL1. In this case, hardware will detect this condition, and apply both sets of address-filtering parameters simultaneously. Hardware will provide Data Indication if the received packet matches either the PAN0 address parameter set {MacPanID0, MacShortAddrs0, MacLongAddrs0, PANCORDNTR0}, or the PAN1 address parameter set {MacPanID1, MacShortAddrs1, MacLongAddrs1, PANCORDNTR1}. At Data Indication, two status bits can be queried by software to quickly determine on which PAN the packet was received, RECD_ON_PAN0 or RECD_ON_PAN1. In the "Two-PAN-one-channel" scenario, it is possible for a packet to satisfy both sets of addressing parameters (PAN0 and PAN1). When this happens, both RECD_ON_PAN0 and RECD_ON_PAN1 will be set. In the "Two-PAN-one-channel" scenario, the CURRENT_NETWORK bit should be ignored, because both networks are simultaneously active.

If Dual PAN mode is not to be used, ACTIVE_NETWORK should be maintained at 0 (default) to select the PAN0 parameters for transceiving. In single PAN operation, the PAN1 parameter registers need not be programmed, and can be used as scratchpad. (PLL_INT1 and PLL_FRAC1 should not be programmed, and should instead be left in their default state, if Dual PAN mode is not in use).

5.4.2 Source Address Matching

This section describes interaction between Dual PAN mode and the modem packet Processors Source Address Matching function. The Source Address Matching Table is two (2) entries deep and 16 bits wide. In Dual PAN mode, entries from both networks will need to be stored in the table. It is highly undesirable to add a 17th bit to each table entry merely to identify the network on which the packet containing the Source Address was received. In order to keep the Source Address table width at 16 bits, the table will be split into two (2) sections:

- Lower section shall hold "PAN0" source address checksums
- Upper section shall hold "PAN1" source address checksums

In Dual PAN mode when software populates the table with a new Source Address Checksum, it will deposit the new checksum in the appropriate section of the table based upon whether the packet was received on PAN0 or PAN1. When the Packet Processor interrogates the table for a matching checksum, it will only search the section of the table corresponding to the PAN on which the current packet is being received. The dividing-line between the PAN0 and PAN1 sections in the table shall be programmability with a 4-bit register DUAL_PAN_SAM_LVL (SAM is an acronym for "Source Address Matching"). The register identifies the first PAN1 entry in the table (indices below this level apply to PAN0). For single-PAN operation, software should set this register to "12" or above which dedicates the entire table to PAN0. The hardware shall reset-default this register to "12". Figure 5-5 shows the effect of setting the PAN dividing line to "8":

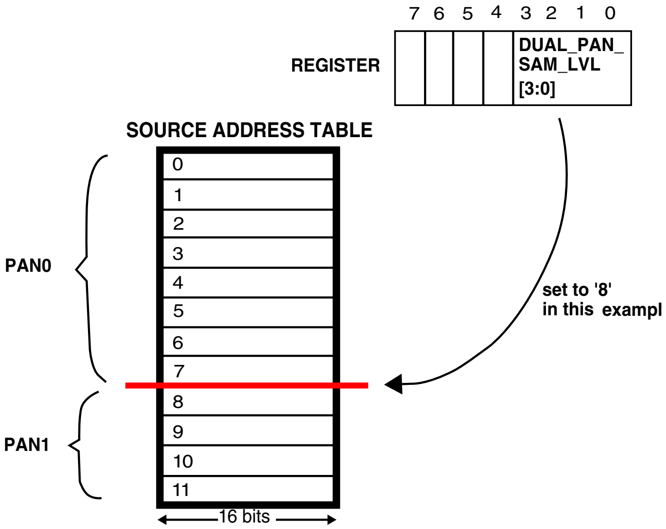


Figure 5-5. Source Address table set to "8"

5.4.3 Programming interface

This section describes registers associated with the Dual PAN ID. All writable registers read back what was previously written by software. Register addresses within the MCR20 memory map and bit assignments are shown below.

Register Name	Access Mode	Width	Description
DUAL_PAN_AUTO	rw	1	Activates automatic Dual PAN operating mode. In this mode, PAN-switching is controlled by hardware at a pre-programmed rate, determined by DUAL_PAN_DWELL. 1: Auto Dual PAN Mode 0: Manual Dual PAN mode (or Single PAN mode). Default. Whenever DUAL_PAN_AUTO=0, CURRENT_NETWORK=ACTIVE_NETWORK at all times. In other words, software directly controls which PAN is selected. Whenever DUAL_PAN_AUTO=1, CURRENT_NETWORK is controlled by HW
ACTIVE_NETWORK	rw	1	Selects the PAN on which to transceiver, by activating a PAN parameter set (PAN0 or PAN1). In Manual Dual PAN mode (or Single PAN mode), this bit selects the active PAN parameter set (channel and addressing parameters) that governs all auto sequences. In Auto Dual PAN mode, this bit selects the PAN on which to begin transceiving, latched at the point at which DUAL_PAN_DWELL register is written. 0: Select PAN0 (Default)
CURRENT_NETWORK	r	1	This read only bit indicates which PAN is currently selected by hardware in automatic Dual PAN mode. 1: PAN1 is selected 0: PAN0 is selected
DUAL_PAN_DWELL	rw	8	Channel Frequency Dwell Time. In Auto Dual PAN mode, hardware will toggle the PAN, after dwelling on the current PAN for the interval described below (assuming Preamble/SFD not detected). A write to DUAL_PAN_DWELL, always re-initializes the DWELL TIMER to the programmed value. If a write to DUAL_PAN_DWELL occurs during an auto sequence, the DWELL TIMER will begin counting down immediately. If a write to DUAL_PAN_DWELL occurs when there is no auto sequence underway, the DWELL TIMER will not begin counting until the next auto sequence begins; it will begin counting at the start of the sequence warm-up. **PRESCALER bits [1:0] (2 bits)** **TIMEBASE** **RANGE (min) - (max) (6 bits)** ------- ---------- ------------- 00 0.5 ms 0.5 - 32 ms 01 2.5 ms 2.5 - 160 ms 10 10 ms 10 - 640 ms 11 50 ms 50 - 3.2 seconds A write to DUAL_PAN_DWELL also causes the value of ACTIVE_NETWORK to get latched into the hardware. This latched value will be the starting point for the automatic dual-pan mode (i.e., start on

Table continues on the next page...

Register Name	Access Mode	Width	Description
			PAN0 or on PAN1). The starting value takes effect immediately (if sequence is underway and DUAL_PAN_AUTO=1), or is otherwise delayed until sequence starts and DUAL_PAN_AUTO=1.
DUAL_PAN_REMAIN	r	6	This read only register indicates time remaining before next PAN switch in auto Dual PAN mode. The units for this register depend on the PRESCALAR setting in the DUAL_PAN_DWELL register according to the following table **DUAL_PAN_DWELL PRESCALAR** — **DUAL_PAN_REMAIN UNITS** 00 — 0.5 ms 01 — 2.5 ms 10 — 10 ms 11 — 50 ms The readback value indicates that between N-1 and N timebase units remain until the next PAN switch. For example, a DUAL_PAN_REMAIN readback value of 3, with a DUAL_PAN_DWELL PRESCALER setting of 2 (10ms), indicates that between 20ms (2*10ms) and 30ms (3*10ms), remain until the next automatic PAN switch.
RECD_ON_PAN0	r	1	Indicates the packet which was just received was received on PAN0
RECD_ON_PAN1	r	1	Indicates the packet which was just received was received on PAN1
DUAL_PAN_SAM_LVL	rw	4	For the Source Address Matching Table, sets the dividing line between PAN0 and PAN1 sections of the table. Table indices at or above this level belong to PAN1. Table entries below this level belong to PAN0. Default = 12 (entire table is PAN0).
CHANNEL0	rw	4	Channel selector for PAN0
MACPANID0	rw	16	MAC PAN ID for PAN0
MACSHORTADDRS0	rw	16	MAC Short Address fro PAN0
MACLONGADDRS0	rw	64	MAC Long Address fro PAN0
PANCORDNTR0	rw	1	1: Device is a PAN Coordinator on PAN0 0: Device is not a PAN Coordinator on PAN0
CHANNEL1	rw	4	Channel selector for PAN1
MACPANID1	rw	16	MAC PAN ID for PAN1
MACSHORTADDRS1	rw	16	MAC Short Address fro PAN1
MACLONGADDRS1	rw	64	MAC Long Address fro PAN1
PANCORDNTR1	rw	1	1: Device is a PAN Coordinator on PAN1 0: Device is not a PAN Coordinator on PAN1

5.5 Active Promiscuous Mode

This section describes the Active Promiscuous mode for a new high performance 2.4 GHz IEEE 802.15.4 platform solution called MKW2xD where primary constraints are radio performance, security and expansion.

5.5.1 Functional Description

The 802.15.4 Wireless Medium Access Control (MAC) and Physical Layer (PHY) specifications for Low-Rate Wireless Personal Area Networks (WPAN's) prescribes the following behavior for Promiscuous mode operation.

In promiscuous mode, the MAC sublayer passes all frames received after the first filter directly to the upper layers without applying and more filtering or processing.

Where the first level is described as:

For the first level of filtering, the MAC sublayer shall disregard all received frames that do not contain a correct value in their FCS field in the MFR (see section 7.2.1.9 in the standard specification).

Promiscuous mode allows a device to receive and process all 802.15.4 frames that pass FCS check (CRC), regardless of whether or not they are addressed to the device. This allows a device to receive and process all valid traffic on the network. In Promiscuous mode, the modem packet processor will provide Data Indication on all packets. This capability allows a simple network sniffer to be constructed, which can display comprehensive packet data for all packets transmitted on a network within receiving range of the device.

Because a device operating in Promiscuous mode is designed to do so discretely, automatic acknowledgement of all received packets is inhibited regardless of the state of the ACK request in the Frame Control Field of the received packet. Otherwise, the sniffing device would be acknowledging received frames for which it was not the address.

In Active Promiscuous mode the packet processor will process packets according to the same rules that it applies to Promiscuous mode, except that it will automatically transmit an acknowledgement packet for frames that were addressed to the device. In other words, in Active Promiscuous mode if the received frame meets all the address and packet filtering requirements that would have been applied in non-promiscuous mode an acknowledgment packet will be transmitted; a Data Indication will still be provided for all packets (those that pass FCS check).

The packet processor can also be programmed to provide Data Indication for packets that fail FCS check. Received packets that fail FCS check are never acknowledged under any circumstances.

5.5.2 Special Handling for Broadcast Packets

Special rules apply to received packets that contain a broadcast PAN identifier (0xffff) and/or a broadcast short address (0xffff) according to the 802.15.4 standard.

Note

> For valid frames that are not broadcast, if the Frame Type subfield indicates a data or MAC command frame and the Acknowledgement Request subfield of the Frame Control Field is set to one, the MAC sublayer shall send an acknowledgement frame.

An exception is made for received packets thatcontain a broadcast PAN identifier and a long address that matches the device's IEEE address. Such packets are acknowledged because the IEEE address uniquely identifies the device regardless of the PAN. This special case will also apply in "Active Promiscuous" mode where such packets will be acknowledged. The following table describes the "acknowledgement decision" for all permutations of broadcast indicators in the received packets addressing fields.

Table 5-5. Acknowledge - on - Broadcast (all permutations)

	PAN ID = !BROADCAST DSTADDR = ! BROADCAST	PAN ID = !BROADCAST DSTADDR = BROADCAST	PAN ID = BROADCAST DSTADDR = BROADCAST	PAN ID = !BROADCAST DSTADDR = ! BROADCAST
SHORT	ACK	NO ACK	NO ACK	NO ACK
LONG	ACK	–	–	ACK
DESTINATION ADDRESS MODE				

5.5.3 Special Handling of Rx Acknowledgement Frames

The modem features an auto sequence to automatically receive and verify Acknowledgment Frames after a Transmit Frame has been sent. This accomplished by programming a "Sequence TR with RXACKRQD=1". During normal operation the receive Acknowledgement Frames are not stored in the packet buffer. They are merely checked for matching Sequence Number and Frame Version with a Data Indication issued for a valid match on both. In Active Promiscuous mode, receive Acknowledgement Frames will be stored into the Packet Buffer just like any other receive frame. This saves software from having to re-construct the contents of the Acknowledgement packet from the contents of the packet that was originally transmitted.

5.5.4 Programming Interface

This section describes registers associated with Active Promiscuous Mode. The definition and functionality of the PROMISCUOUS bit, which will be maintained for the modem is identical to existing FSL 802.15.4 transceivers. The ACTIVE_PROMISCUOUS bit is new for the modem as shown in the following table.

Table 5-6. ACTIVE_PROMISCUOUS register

Register Name	Access Mode	Width	Description
ACTIVE_PROMISCUOUS	rw	1	1: Provide Data Indication on all received packets under the same rules that apply in PROMISCUOUS mode, however acknowledge those packets under rules that apply in non-PROMISCUOUS mode 0: normal operation (Default)

5.6 Clock System

5.6.1 32MHz Crystal Oscillator

The 32 MHz Crystal Oscillator block (XTAL32m) is composed of the following features:

- Amplifier — it associates with an external crystal and tuning capacitors creating the main oscillator loop.
- Hysteresis comparator — it creates a logic-level signal from the sine wave developed across the crystal terminals. It has a high frequency filtering effect and its hysteresis prevents glitches at start-up.
- Clock gate timer — it waits for a delay before passing the clock signal. This delay is set to assure the XTAL has reached the expected final clock +/-10ppm.
- Clock divider that contains a divider by two (2) and divide by eight (8) circuitry. Level shifters that are needed between VBAT and VDD domains.

The XTAL32m frequency is 32 MHz. It can be trimmed with some internal capacitances connected on both pads. The XTAL32m is supplied by the VBAT. An external clock generator can be connected to XTAL32Mout. In this case, the clock is directly connected to the output in bypass mode. A 1.8 V square clock can be connected as required.

A counter will count with the xtal clock generated until TBD ms and at the end of the counting will enable the clock to go outside (gate passing).

The crystal oscillator block provides the clock for:

- The PLL reference
- The RSSI ADC
- The IEEE 802.15.4 digital block

The crystal oscillator block can work in 2 modes:

- Low power mode /RFoff mode where the comparator works with lower current. The clock is used only by the digital manager and provided to the clock output pin. This is the mode by default
- High performance mode/RFon mode where clocks are also sent to the RF part. Jitter is then important. When passing from a low power mode to a high power mode and vice versa, it must be ensured that no glitches are present (preferably should be a smooth transition).

5.6.2 Clock output feature overview

The modem has the capability to output various clock frequencies per this table by register selection. This feature allows the user to make adjustments per their application requirements.

- XTAL domain to minimize dynamic current consumption based on power mode selected.

- XTAL domain can be completely gated off (HIBERNATE mode)

- SPI communication allowed in HIBERNATE

Table 5-7. CLOCK_OUT Frequencies

CLK_OUT_DIV [2:0]	CLK_Out frequency	Comments
0	32Mhz	
1	16MHz	
2	8MHz	
3	4MHz	DEFAULT if GPIO5=0
4	2MHz	
5	1MHz	
6	62.5kHz	
7	32.786kHz	DEFAULT if GPIO5=1

There will be an enable and disable bit for CLK_OUT. When disabling, the clock output will optionally continue to run for 128 clock cycles after disablement. There will also be one(1) bit available to adjust the CLK_OUT I/O pad drive strength. Bits described in the table above will reside in the 8 bit CLK_OUT_CTRL register. Default setting is 32.787kHz with a strap option to select a default of 4MHz.

Chapter 6
Modem: Interrupts

6.1 Introduction

Interrupts provide a way for the radio to inform the host microcontroller (MCU) of onboard events without requiring the MCU to query for status. The following sections describe the interrupt features and sources of the modem.

- 13 interrupt sources (interrupt status bits)
- Each interrupt source individually maskable
- Each interrupt source is individually write-1-to-clear.
- Interrupts remain asserted until cleared.

6.2 Modem Interrupt Sources

The radio has a single, active low, interrupt pin (IRQ_B) provided to the MCU. The interrupt pin is configured as actively-driven by the radio. Internally, the radio has 13 interrupt sources as shown in this table:

Table 6-1. Modem Interrupt Sources

Item	Interrupt Sources Status Bit	Mask Bit	Description	Interrupt Clear Mechanism
1	PLL_UNLOCK_IRQ	PLL_UNLOCK_MSK	PLL Unlock Interrupt. A '1' indicates an unlock event has occurred in the PLL.	write-1-to-clear
2	FILTERFAIL_IRQ	FILTERFAIL_MSK	RX Packet has Failed Filtering. A '1' indicates the most-recently received packet has been rejected due to elements within the packet. In Dual PAN mode, FILTERFAIL_IRQ applies to either or both networks, as follows:	write-1-to-clear

Table continues on the next page...

Table 6-1. Modem Interrupt Sources (continued)

Item	Interrupt Sources Status Bit	Mask Bit	Description	Interrupt Clear Mechanism
			A: If PAN0 and PAN1 occupy different channels and CURRENT_NETWORK=0, FILTERFAIL_IRQ applies to PAN0. B: If PAN0 and PAN1 occupy different channels and CURRENT_NETWORK=1, FILTERFAIL_IRQ applies to PAN1. C: If PAN0 and PAN1 occupy the same channel, FILTERFAIL_IRQ is the logical 'AND' of the individual PANs' Filter Fail status.	
3	RXWTRMRKIRQ	RX_WMRK_MSK	RX Byte Count Watermark Interrupt. A '1' indicates the number of bytes specified in the RX_WTR_MARK register has been reached.	write-1-to-clear
4	CCAIRQ	CCAMSK	CCA Interrupt. A '1' indicates the completion of a CCA operation.	write-1-to-clear
5	RXIRQ	RXMSK	RX Interrupt. A '1' indicates the completion of a receive operation.	write-1-to-clear
6	TXIRQ	TXMSK	TX Interrupt. A '1' indicates the completion of a transmit operation.	write-1-to-clear
7	SEQIRQ	SEQMSK	Sequence Complete Interrupt. A '1' indicates the completion of an autosequence. This interrupt will assert whenever the Sequence Manager transitions from non-idle to idle state, for any reason.	write-1-to-clear
8	PB_ERR_IRQ	PB_ERR_MSK	Packet Buffer Error Interrupt. 1: Packet Buffer Underrun Error has occurred since the last time this bit was cleared by software 0: Packet Buffer Underrun Error has not occurred since the last time this bit was cleared by software. **Note**: PB_ERR_IRQ will occur only when the SPI Packet Buffer access is BURST mode (not BYTE mode)	write-1-to-clear
9	WAKE_IRQ	WAKE_MSK	Wake Interrupt. Indicates either a Wake-from-POR or Wake-from-HIBERNATE event has occurred. The WAKE_FROM_HIB/TMRSTATUS bit indicates which type of wake event occurred, as long as WAKE_IRQ=1.	write-1-to-clear
10	TMR1IRQ	TMR1MSK	TMR1 Interrupt. Indicates T1CMP comparator value matched event timer counter.	write-1-to-clear

Table continues on the next page...

Table 6-1. Modem Interrupt Sources (continued)

Item	Interrupt Sources Status Bit	Mask Bit	Description	Interrupt Clear Mechanism
11	TMR2IRQ	TMR2MSK	TMR2 Interrupt. Indicates comparator value matched event timer counter. This flag is shared between the T2CMP (24-bit) and T2PRIMECMP (16-bit) compare registers.	write-1-to-clear
12	TMR3IRQ	TMR3MSK	TMR3 Interrupt. Indicates T3CMP comparator value matched event timer counter.	write-1-to-clear
13	TMR4IRQ	TMR4MSK	TMR4 Interrupt. Indicates T4CMP comparator value matched event timer counter.	write-1-to-clear

Each interrupt can be controlled by an interrupt mask. All interrupts are OR- combined with the external pin IRQ_B. The IRQ is issued when IRQ_B is set to 0.

Any or all of the interrupt sources, can be enabled to cause an assertion on IRQ_B. Each interrupt source has its own interrupt status bit in modems Direct Register space. Each interrupt source is individually maskable (each has its own MASK bit) so that each interrupt source can be individual enabled to trigger an assertion on IRQ_B.

There is also a global interrupt mask, TRCV_MSK (PHY_CTRL4 register), which can enable/disable all IRQ_B assertions by programming a single masking bit. All 13 interrupt status bits use a write-1-to-clear protocol. Interrupt status bits are not affected by reads. The IRQ_B pin is a level-sensitive interrupt indicator to the MCU; on an interrupt, IRQ_B will remain asserted until all active interrupt sources are cleared or masked.

6.3 Additional Interrupt Mask and Source Descriptions

Numerous register bits are provided to control interrupt behavior within the modem as shown in the table below.

Table 6-2. Interrupt Control Summary

Field	R/W	Description	Default
TRCV_MSK	rw	1: Mask all Interrupts from asserting IRQ_B 0: Enable any Interrupt to assert IRQ_B	0
T1CMP[23:0]	rw	TMR1 compare value. If TMR1CMP_EN=1 and the Event Timer matches this value, TMR1IRQ is set.	0xFFF
T2CMP[23:0]	rw	TMR2 compare value. If TMR2CMP_EN=1 and the Event Timer matches this value, TMR2IRQ is set.	0xFFF

Table continues on the next page...

Table 6-2. Interrupt Control Summary (continued)

Field	R/W	Description	Default
T2PRIMECMP[15:0]	rw	TMR2-prime compare value. If TMR2CMP_EN=1 and TC2PRIME_EN=1 and the lower 16 bits of Event Timer matches this value, TMR2IRQ is set.	0xFF
T3CMP[23:0]	rw	TMR3 compare value. If TMR3CMP_EN=1 and the Event Timer matches this value, TMR3IRQ is set.	0xFFF
T4CMP[23:0]	rw	TMR4 compare value. If TMR4CMP_EN=1 and the Event Timer matches this value, TMR4IRQ is set.	0xFFF
TMR1CMP_EN	rw	1: Allow an Event Timer Match to T1CMP to set TMR1IRQ 0: Don't allow an Event Timer Match to T1CMP to set TMR1IRQ	0
TMR2CMP_EN	rw	1: Allow an Event Timer Match to T2CMP or T2PRIMECMP to set TMR2IRQ 0: Don't allow an Event Timer Match to T2CMP or T2PRIMECMP to set TMR2IRQ	0
TMR3CMP_EN	rw	1: Allow an Event Timer Match to T3CMP to set TMR3IRQ 0: Don't allow an Event Timer Match to T3CMP to set TMR3IRQ	0
TMR4CMP_EN	rw	1: Allow an Event Timer Match to T4CMP to set TMR4IRQ 0: Don't allow an Event Timer Match to T4CMP to set TMR4IRQ	0
TC2PRIME_EN	rw	1: Allow a match of the lower 16 bits of Event Timer to T2PRIMECMP to set TMR2IRQ 0: Don't allow a match of the lower 16 bits of Event Timer to T2PRIMECMP to set TMR2IRQ	0
RX_WTR_MARK[7:0]	rw	Receive byte count (octets) needed to trigger a RXWTRMRKIRQ interrupt. A setting of 0 generates an interrupt at end of the Frame Length field (first byte after SFD). A setting of 1 generates an interrupt after the first byte of Frame Control Field, etc.	0xFF

6.4 Functional Description

6.4.1 Interrupt Status Bit Structure

The modem has 13 interrupt sources, each represented in the register map by an interrupt status bit. All interrupt status bits share a common, generic structure. If a particular interrupt source is enabled, a "TRIGGERING EVENT" for that interrupt source will always set the status bit. A write-1-to-clear input from the SPI module will clear the status bit, but only if there is not a simultaneous triggering event. The triggering event will take priority (occur) over a software clear attempt if both events coincide.

The CLK and CLK_EN are provided by the SPI. This allows interrupt status bits to be read and cleared in the HIBERNATE state, when the modem's crystal oscillator is disabled and the SPI SCLK is the only "clock" available in the IC. The clock gate guarantees that the status bit sees a clock only when either a triggering event occurs or a write-1-to-clear pulse arrives from the SPI. This reduces interrupt status bit power consumption to an absolute minimum. The common interrupt status bit structure is shown in Figure 6-1.

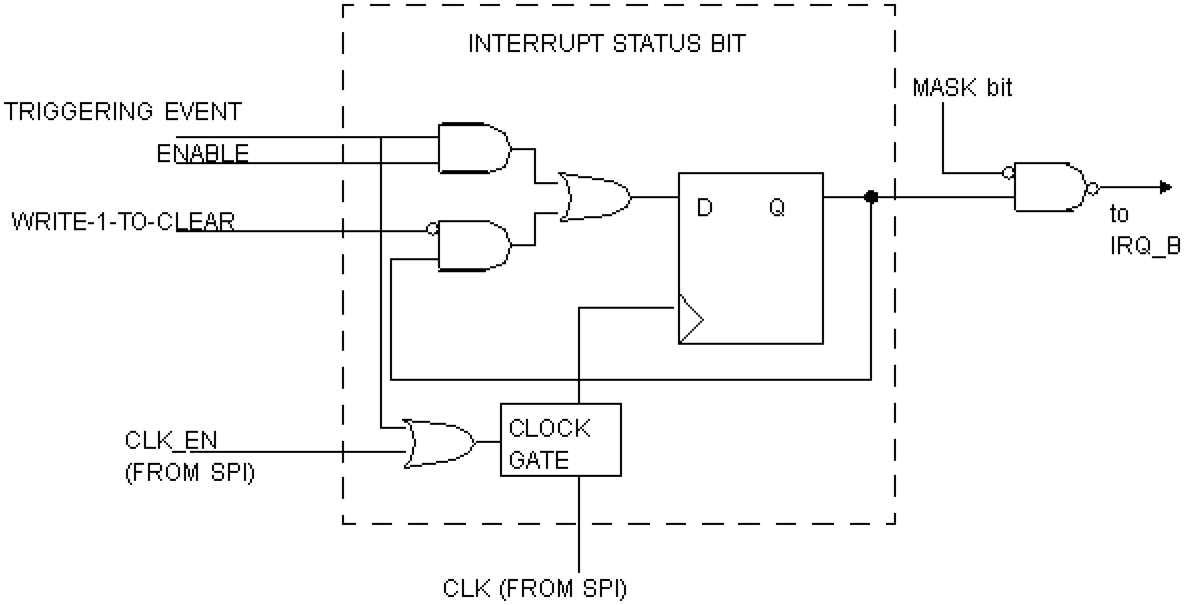


Figure 6-1. Common Interrupt Bit Structure

NOTE

For any modem interrupt source, if the triggering event occurs, the Interrupt Status Bit will be set regardless of the state of the corresponding MASK bit. If any of the 13 interrupts is to be ignored, software should set the corresponding MASK bit and apply the appropriate bit mask when reading the IRQSTS1, IRQSTS2, or IRQSTS3 register to mask out the unwanted status bit.

6.4.2 Clearing Interrupts

All interrupt status bits use a write-1-to-clear protocol. Writing a '1' to the interrupt status bit in any one of the IRQSTS1, IRQSTS2, or IRQSTS3 registers clears the offending interrupt. Writing a '0' to an interrupt status bit has no effect on the bit. Interrupt status bits are not affected by reads.

6.4.3 Timer Interrupts

The modem features a 24-bit Event Timer which runs at the 802.15.4 bit rate of 250 kHz. The modem has four timer interrupts (TMR1IRQ, TMR2IRQ, TMR3IRQ, and TMR4IRQ), each with its own 24-bit compare register (T1CMP, T2CMP, T3CMP, and T4CMP) and each with its own compare enable (TMR1CMP_EN, TMR2CMP_EN, TMR3CMP_EN, and TMR4CMP_EN).

For each timer compare enable:

- If the compare enable bit is set - A match on the respective 24-bit compare value to the Event Timer will cause the corresponding interrupt status bit to become set.
- If the compare enable is low - Event Timer matches will not cause the corresponding interrupt status bit to become set.

In addition, a 16-bit T2PRIMECMP compare value is provided along with a compare-enable bit TC2PRIME_EN.

- When TC2PRIME_EN is set and TMR2CMP_EN is also set - A match on T2PRIMECMP with the *lower 16 bits* of Event Timer will cause TMR2IRQ to become set rather than a full 24-bit compare.

6.4.4 PLL Unlock Interrupt

The PLL_UNLOCK_IRQ status bit indicates the PLL has come out of lock during a TX, RX or CCA transceiver operation. The modems Sequence Manager will begin monitoring for PLL unlocks only after a complete transceiver warm-up has occurred; unlocks that occur during warm-up will not cause a PLL unlock. A PLL unlock that occurs after the warm-up period will cause a sequence abort.

6.4.5 Filterfail Interrupt

The modems Packet Processor performs filtering on all received packets to determine whether the packet is intended for the device based on rules which if violated, will cause a Filterfail interrupt. A Filterfail interrupt will occur during packet reception when a packet fails filtering rules. On a Filterfail interrupt, the FILTERFAIL_CODE register can be read to ascertain more information about the nature of the filter failure.

NOTE

A Filterfail interrupt is not expected to be widely used in mission modes; it has been provided primarily for debug purposes.

NOTE

If a packet is received and First-Stage filtering fails (i.e., one or more of bits filterfail_code[0-3] is set, the Second-Stage filterfail_code bits, 3-9, should be ignored since the modem's packet processor will not know where to search for PAN ID and address fields in the incoming byte stream for illegal addressing modes.

The table below indicates each Filterfail code bit and it respective reason the packet was rejected.

Table 6-3. Filterfail Codes

FILTERFAIL CODE BIT	REASON FOR FILTERFAIL
filterfail_code[0]	Fails Stage 1 Frame Length Checking (FL < 5 or FL > MAXFRAMELENGTH)
filterfail_code[1]	Fails Stage 1 Section 7.2.1.1.6 or Section 7.2.1.1.8 Checking (DST_ADDR_MODE or SRC_ADDR_MODE = 1)
filterfail_code[2]	Fails Stage 1 Section 7.2.1.1.5 Checking (Illegal PAN_ID_COMPRESSION Usage)
filterfail_code[3]	Fails Stage 1 Frame Version Checking
filterfail_code[4]	Fails Stage 2 Auto-RxAck Checking (Illegal Ack Frame Format in Sequence TR)
filterfail_code[5]	Fails Stage 2 Frame Type Checking (Incorrect Frame Filter Bit setting)
filterfail_code[6]	Fails Stage 2 Frame Length Checking (Illegal Beacon, Data, or Cmd FL)
filterfail_code[7]	Fails Stage 2 Addressing Mode Checking (Illegal Addressing Mode for Beacon, Data, or Cmd)
filterfail_code[8]	Fails Stage 2 Sequence Number Matching (Sequence TR Only)
filterfail_code[9]	Fails Stage 2 PAN ID or Address Checking (Beacon, Data, or Cmd)

6.4.6 RX Watermark Interrupt

During packet reception, a RX Watermark interrupt will occur when the number of received bytes matches the contents of the RX_WTR_MARK register minus 1.

NOTE

For the purpose of defining RX_WTR_MARK, the first byte received is the Frame Length field (PHR). The second byte received is the least-significant byte of Frame Control Field, etc.

For example, to cause an RX Watermark Interrupt to occur after the 10th received byte, set RX_WTR_MARK=11.

6.4.7 CCA Interrupt

The modems Clear Channel Assessment (CCA) has been completed; CCA interrupts occur at the end of both CCA and Energy Detect (ED) measurement intervals. For TX and Slotted (beacon-enabled networks) TX modes, the CCAIRQ is asserted indicating the CCA determined the channel is busy. The result of the CCA measurement is reported back to software in the CCA bit of the IRQSTS2 Register. The CCA bit indicates either a busy channel (1) or an idle channel (0).

6.4.8 RX Interrupt

RX interrupts occur at the end of the transceivers RX operation. An RXIRQ in combination with a SEQIRQ is considered a "Data Indication". RX interrupts are not generated on packets which fail FCS (CRC check) or fail packet filtering.

To receive a Data Indication (including RXIRQ) on packets which fail CRC, clear the CRC_MSK bit of the PHY_CTR2 register to 0. To inhibit packet filtering and enable Data Indication on packets which would otherwise fail packet filtering rules, set the PROMISCUOUS bit of the PHY_CTRL4 register to 1.

6.4.9 TX Interrupt

TX interrupts (TX_IRQ) occur at the end of the transceivers TX operation.

6.4.10 Sequencer Interrupt

The Sequencer Interrupt (SEQIRQ), indicates that an autosequence has completed and the Sequence Manager has returned to its idle state. A SEQIRQ will always occur at the end of an autosequence even if the autosequence terminated abnormally, such as a Software Abort, a TC3 Timeout or a PLL Unlock Abort.

NOTE

> The ABORT_STS register can be read to determine which of these abnormal terminations occurred, if any (see the registers section for more details).

The SEQIRQ always occurs whenever the Sequence Manager transitions from non-idle to idle state. When SEQIRQ occurs, software can be sure that the Sequence Manager is in its idle state and a new sequence can be programmed immediately.

6.4.11 Interrupts from Exiting Low Power Modes

The modem has two low power modes which generate an 'Wake' interrupt. A Wake interrupt (WAKE_IRQ) occurs from either of these two conditions:

1. Exit from RESET
2. Exit from HIBERNATE

A transition out of RESET or HIBERNATE will always cause a WAKE_IRQ when unmasked. WAKE_IRQ is the only modem interrupt whose default state is "unmasked" so the MCU can expect this interrupt without any prior programming of the device. The MCU can interpret the WAKE_IRQ interrupt as a "Modem Ready" indication.

6.4.11.1 Exiting from RESET

The modem is put into reset and will stay in reset through the assertion of RST_B (reset bar or $\overline{\text{RESET}}$). In the interest of lowest power, there is no external pull-up resistor on input RST_B. As part of the MCU initialization, an internal GPIO is programmed as an output and then driven low to reset the modem. The RST_B input is asynchronous and needs to be held low for only a short period. In the reset condition, the modem is totally powered down and no clocks are available. Coming out of reset, the WAKE_IRQ interrupt status bit is set and causes an assertion on IRQ_B because WAKE_MSK is 0 by default. After the interrupt request is seen by the MCU, the modem is alive and ready for SPI transactions.

6.4.11.2 Exiting from HIBERNATE

The hibernate or low power mode is a specific mode where the XTAL is disabled. The SPI is capable of operation in modem low power modes, except Reset. Going out of hibernate mode through SPI means that the XTAL needs to be start up again, the digital regulator forced to high power mode and a new wake up interrupt (WAKE_IRQ) is generated as a result.

Operation in Hibernate mode allows most modem registers and the complete Packet Buffer to be accessed in radio's lowest-power operational state thereby, enabling minimal power consumption especially during the register-initialization phase of the IC's software.

In Hibernate, the SPI operates in "asynchronous mode". To write a register, the SPI module generates both clock and clock enable to the Register block since the crystal oscillator is disabled; for reads, no latching of read-back data takes place since the oscillator is disabled therefore, read-back data cannot change.

6.4.12 Packet Buffer Error Interrupt

A Packet Buffer Error Interrupt, PB_ERR_IRQ, indicates that a Packet Buffer Underrun Error has occurred. This condition can occur if a SPI burst read access to the Packet Buffer begins before the incoming packet has been completely received and the SPI Packet Buffer read address overtakes the RX demodulator write address.

Packet Buffer data obtained from a SPI transfer which results in a PB_ERR_IRQ has been corrupted and should be discarded. However, the packet contents have not been lost; the PB_ERR_IRQ should be cleared and the SPI transfer should then be restarted.

Chapter 7
Modem: Timer Information

7.1 Event Timer Block

The modem contains an internal Event Timer block that manages system timing.

The Event Timer consists of a prescaler and a 24-bit counter which increment whenever the crystal clock is operating (i.e. the modem is not in Hibernate mode). Interrupts to the MCU may be generated when the "current time" of the counter matches several predetermined values set in registers via SPI write operations. The current time is accessible at any time via a SPI read operation as well as programmable via a SPI write operation.

The Event Timer provides the following functions:

- Timer to generate current system time
- Interrupt generation at pre-determined system times
- Abort an RX sequence at pre-determined system time
- Latches "timestamp" value during packet reception
- Initiates timer-triggered sequences

7.2 Event Timer Time Base

The Event Timer's base clock (tmr_clk) is derived from a programmable prescaler that is clocked by the 32 MHz crystal source. The prescaler provides counter input frequencies from 500 kHz down to 15.625 kHz, which sets the granularity and resolution of the current time. The prescaler, and thus the Event Timer only increment when the crystal oscillator is active. The field TMR_PRESCALE[2:0] at Indirect Access Register 0x28 establishes the tmr_clk frequency as shown in table below.

Table 7-1. Event Timer Prescaler Settings

Indirect Access Register 0x28 TMR_PRESCALE[2:0]	Event Timer Time Base	Maximum Event Timer Duration
000	Reserved	Reserved
001	Reserved	Reserved
010	500 kHz	33.554 seconds
011 (default)	250 kHz	67.109 seconds
100	125 kHz	134.218 seconds
101	62.5 kHz	268.436 seconds
110	31.25 kHz	536.871 seconds
111	15.625 kHz	1073.742 seconds

The 24-bit counter automatically rolls over upon reaching its maximum value at the corresponding maximum Event Timer durations also provided in the table.

7.3 Setting Current Time

"Current Time" is defined as the value of the Event Timer internal counter. The current time is programmable, but does not have to be programmed. In the reset condition, the modem current time is set to zero. Current time advances from zero at the tmr_clk clock rate and rolls over to zero after reaching its maximum value.

Programming "current time" is accomplished by using four SPI registers:

1. T1CMP_LSB, Direct Register 0x17, T1CMP[7:0]
2. T1CMP_MSB, Direct Register 0x18, T1CMP[15:8]
3. T1CMP_USB, Direct Register 0x19, T1CMP[23:16]
4. PHY_CTRL4, Direct Register 0x07, Bit 2, TMRLOAD

When field TMRLOAD is programmed to high, the value of “current time” is set to the value in T1CMP[23:0]. Thus, T1CMP[23:0] is first programmed to the desired current time value, then TMRLOAD is programmed to 1, which initiates the timer load. The change to the “current time” value occurs within 3 crystal clock cycles, after which normal incrementing resumes on the next rising tmr_clk edge. TMRLOAD is not required to be programmed to zero for the Event Timer to resume normal operation. TMRLOAD is a write-only, self-clearing bit. Writing a 0 to TMRLOAD has no effect. The readback value of TMRLOAD is always 0.

7.4 Reading Current Time

The current value of the Event Timer can be read via the SPI by reading EVENT_TMR_LSB (Direct Register 0x0C) for EVENT_TMR[7:0], EVENT_TMR_MSB (Direct Register 0x0D) for EVENT_TMR[15:8], and EVENT_TMR_USB (Direct Register 0x0E) for EVENT_TMR[23:16].

The “current time” may be obtained using three single-byte SPI reads or a 3-byte SPI burst read (or as part of a longer burst read operation). It is important to realize that the Event Timer may increment during these SPI read operations or between successive SPI reads if single-byte SPI reads are used. To counter this during such an access, the modem latches the “current time” to protect the MCU from obtaining an incorrect value. The “current time” most significant 16 bits (MSB, USB) are latched when the least significant 8 bits (LSB) SPI location is read. This guarantees a stable value until the MCU completes a read of all 3 bytes constituting the “current time” before it is allowed to update. If such latching is undesired, it can be turned off by setting the EVENT_TMR_DO_NOT_LATCH bit to 1. The EVENT_TMR_DO_NOT_LATCH bit resides in the SEQ_MGR_CTRL register (Indirect Access Register 0x3A, bit 3).

NOTE

> The preferred procedure to obtain the "current time" value from the modem is to perform a 3-byte burst read of the "current time" starting at the LSB address.

7.5 Latching the Timestamp

The modem has the ability generate a Timestamp, or to latch a copy of the “current time” while continuing to increment its internal counter. This timestamp value latched within the Event Timer corresponds to the beginning of a receive packet where the actual packet data begins after the SFD (Start-of-Frame-Delimiter) has been received. The timestamp[23:0] can be accessed via SPI by reading TIMESTAMP_LSB (Direct Register 0x0F) for timestamp[7:0], TIMESTAMP_MSB (Direct Register 0x10) for timestamp[15:8], and TIMESTAMP_USB (Direct Register 0x11) for timestamp[23:16]. The timestamp remains latched until another packet is received at which point the timestamp[23:0] value is updated and re-latched.

7.6 Event Timer Comparators

The modem incorporates four full 24-bit programmable fields that compare to the Event Timer's "current time". The intent of these compares are to enable the host to schedule events relative to the "current time". When a match between the "current time" and any one of the four timer compare values occurs, a corresponding flag is sent to internal interrupt logic. This causes the appropriate bit in the IRQSTS3 register (Direct register 0x2) to be set and depending on the interrupt mask control bit, generate an interrupt event on the IRQ_B pin.

7.6.1 Timer Compare Fields

There are four 24-bit timer compare fields and one 16-bit timer compare field:

1. T1CMP[23:0], (T1CMP_LSB, T1CMP_MSB, T1CMP_USB), Direct Address 0x17, 0x18, 0x19 respectively.
2. T2CMP[23:0], (T2CMP_LSB, T2CMP_MSB, T2CMP_USB), Direct Address 0x1A, 0x1B, 0x1C respectively.
3. T3CMP[23:0], (T3CMP_LSB, T3CMP_MSB, T3CMP_USB), Direct Address 0x12, 0x13, 0x14 respectively.
4. T4CMP[23:0], (T4CMP_LSB, T4CMP_MSB, T4CMP_USB), Direct Address 0x1D, 0x1E, 0x1F respectively.

Additionally, there is a special 16-bit timer compare field for situations where the upper byte of EVENT_TMR is a 'don’t care' case.

1. T2PRIMECMP[15:0], (T2PRIMECMP_LSB, T2PRIMECMP_MSB), Direct Address 0x15 and 0x16 respectively.

The TMR2IRQ status bit can be set by a match to T2CMP or T2PRIMECMP:

- If register bit TC2PRIME_EN=1 (PHY_CTRL4 register, Direct Address 0x7, bit 0), then TMR2IRQ becomes set by a match to T2PRIMECMP (16-bit compare).
- If TC2PRIME_EN=0, then TMR2IRQ becomes set by a match to T2CMP (24-bit compare).

7.6.2 Timer Compare-Enable Bits

Each timer comparator has a enable bit that enables or disables the compare function. To enable, write a “1” to the corresponding comparator. The default condition is the timer disabled (reset to “0”):

1. TMR1CMP_EN, PHY_CTRL3 Register, Direct Address 0x05, bit 4
2. TMP2CMP_EN, PHY_CTRL3 Register, Direct Address 0x05, bit 5
3. TMR3CMP_EN, PHY_CTRL3 Register, Direct Address 0x05, bit 6
4. TMR4CMP_EN, PHY_CTRL3 Register, Direct Address 0x05, bit 7

If a timer comparator is enabled using its associated bit, the corresponding interrupt status bit (TMRxIRQ) will be set by a timer compare match to its respective TxCMP register. If a timer comparator is disabled using its associated bit, the corresponding interrupt status bit (TMRxIRQ) will not be set by a timer compare match to its respective TxCMP register.

7.6.3 Timer Interrupt Status Bits

When enabled, all four fields can be continuously compared to the current value of the Event Timer counter. When a match occurs, the following corresponding internal status flags assert:

1. TMR1IRQ, IRQSTS3 Register, Direct Address 0x02, bit 0
2. TMR2IRQ, IRQSTS3 Register, Direct Address 0x02, bit 1
3. TMR3IRQ, IRQSTS3 Register, Direct Address 0x02, bit 2
4. TMR4IRQ, IRQSTS3 Register, Direct Address 0x02, bit 3

The four interrupt status bits are write-1-to-clear. The status bit remains set until a 1 is written to the bit location in the IRQSTS3 register. Writing 0 to the bit or reading the bit will not change its state.

7.6.4 Timer Interrupt Masks

When a comparator match occurs and the internal status flag asserts, the following interrupt masks can enable an interrupt on the IRQ pin:

1. TMR1MSK, IRQSTS3 Register, Direct Address 0x02, bit 4
2. TMR2MSK, IRQSTS3 Register, Direct Address 0x02, bit 5
3. TMR3MSK, IRQSTS3 Register, Direct Address 0x02, bit 6
4. TMR4MSK, IRQSTS3 Register, Direct Address 0x02, bit 7

If the interrupt mask is set to “1” (masked), the respective interrupt status bit, TMRxIRQ, will not cause an interrupt on the IRQ_B pin. If the interrupt mask is set to “0” (enabled), the respective interrupt status bit, TMRxIRQ, will cause an interrupt on the IRQ_B pin. Timer mask bits, TMRxMSK, have no effect on the state of the TMRxIRQ interrupt status bits; they only allow (or prevent) the respective TMRxIRQ bit from generating an interrupt on IRQ_B.

7.6.5 Setting Compare Values

Because the primary timer compare fields are 24-bit values, they are each shared across 3 sequential SPI register addresses. The timer compare value can be changed using 3 single-byte SPI writes, one 3-byte SPI burst write or as part of a longer multi-byte write operation.

Note

Not all bits of the timer compare value are updated simultaneously by the SPI. To prevent the Event Timer from generating a false match to a partially updated timer compare value, software should clear the respective TMRxCMP_EN bit prior to changing the timer compare register TxCMP. Once TxCMP has been updated, software can re-enable the respective TMRxCMP_EN bit.

7.7 Intended Event Timer Usage

It is intended that the system use the "current time" value and the timer compare functions of the Event Timer to schedule system events, including:

- Generating time-based interrupts
- Abort an RX operation
- Triggering transceiver operations

7.7.1 Generating Time-Based Interrupts

Generating time-based interrupts is accomplished by setting timer compare values relative to the "current time" allowing the Event Timer counter to increment until a timer compare match is generated, and using this match to generate an interrupt to the host. The general procedure is as follows:

1. Disable the timer compare.
2. Enable the timer compare interrupt mask.
3. Read the "current time" value from event_tmr[23:0].
4. Add an offset to the "current time" value to equal the desired "future time".
5. Program the appropriate timer_compare value (TxCMP) to "future time".
6. Program the appropriate TMRxCMP_EN bit to enable the compare.
7. Allow a timer compare match to set the status register bit and generate an interrupt (see Section 9.6.3, Timer Interrupt Status Bits). The appropriate internal status register bit is always set upon a TxCMP match. An external interrupt is generated when the corresponding SPI interrupt mask bit, TMRxMSK, is clear.
8. Program the appropriate TMRxCMP_EN bit to disable the compare function. If this is not done, the compare function will continue to run and generate another interrupt every time the counter rolls over and again matches the comparator.

7.7.2 Using T3CMP to Abort an RX operation

The Event Timer provides a timer-based mechanism to automatically abort an RX operation. The modem is put into an RX operation when XCVSEQ[2:0] (PHY_CTRL1 register, Direct Address 0x3) is set to 0x01. An RX autosequence will commence and will begin a search for preamble. Software can program, in advance, a hardware timeout to occur such that the RX autosequence can be automatically aborted by hardware after a certain amount of time elapses without a packet being detected.

The general procedure is as follows:

1. Read the “current time” value from EVENT_TMR[23:0].
2. Add an offset to the “current time” value to equal desired “future time” to abort the RX operation.

3. Program register T3CMP[23:0] to value “future time”.
4. Program TC3TMOUT, (PHY_CTRL4 Register, Direct Address 0x07, bit 6) to 1, to enable the hardware aborting of the RX operation.
5. Program XCVSEQ=1, to start the RX autosequence.
6. When “current time” equals T3CMP[23:0], the modem aborts the RX autosequence and the Sequence Manager returns to SEQ_IDLE state. The SEQIRQ interrupt status bit (IRQSTS1 Register, Direct Address 0, bit 0) will be set, and the TMR3IRQ interrupt status bit will also be set. The RXIRQ interrupt status bit (IRQSTS1 Direct Address 0, bit 2) will not be set since no packet was received before the timeout.

7.7.3 Using T2CMP or T2PRIMECMP to Trigger Transceiver Operations

The Event Timer provides a timer-based mechanism to automatically launch a transceiver autosequence. Using this method, a sequence can be “scheduled” for time in the future, and the MCU can be put into a reduced power state, without the need to wake up at the appropriate time to start the autosequence.

The general procedure is as follows:

1. Read the “current time” value from EVENT_TMR[23:0].
2. Add an offset to the “current time” value to equal desired “future time” to launch the autosequence.
3. For either a 24-bit or a 16-bit compare:
   - For a 24-bit compare: program register T2CMP[23:0] to value “future time”, and set TC2PRIME_EN=0.
   - For a 16-bit compare: program register T2PRIMECMP[15:0] to value “future time”, and set TC2PRIME_EN=1.
4. Set TMRTRIGEN=1 and program the desired autosequence into XCVSEQ[2:0]. Both TMRTRIGEN and XCVSEQ fields reside in the PHY_CTRL1 register (Direct Address 0x03).
5. When “current time” equals T2CMP[23:0] (for a 24-bit compare), or T2PRIMECMP[15:0] (for a 16-bit compare), the modem will launch the scheduled autosequence in accordance with the XCVSEQ[2:0] field. The TMR2IRQ interrupt status bit will become set.

6. The MCU can wake up at this point (if TMR2MSK=0), or can elect to continue in low-power mode until the autosequence completes, at which time SEQIRQ will become set. If SEQMSK=0, then SEQIRQ will interrupt the MCU at the completion of the transceiver operation.

Chapter 8
MCU-Modem SPI Interface

8.1 SiP Level SPI Pin Connections

The SiP level SPI pin connections are all internal to the device. The figure below shows the interconnections.

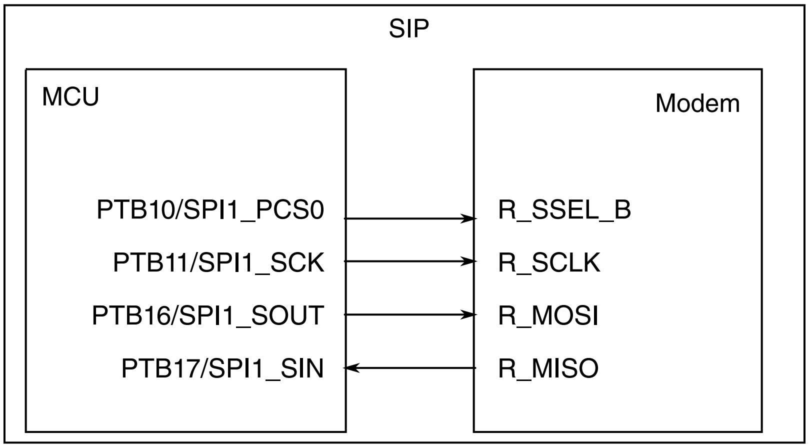


Figure 8-1. SPI Interconnects

Table 8-1. Internal SPI Connections

MCU Signal	Modem Signal	Description
PTB10/SPI1_PCS0	R_SSEL_B	MCU SPI master slave select drives modem SPI slave select input
PTB11/SPI1_SCLK	R_SCLK	MCU SPI master clock output drives modem SPI slave clock input
PTB16/SPI1_SOUT	R_MOSI	MCU SPI master data output drives modem SPI slave data in
PTB17/SPI1_SIN	R_MISO	MCU SPI master data input is driven by modem SPI slave data out

8.2 Features

Features of the SPI bus interface:

- MCU bus master
- Modem bus slave
- Bi-direction data transfer
- Dedicated interface; must meet modem protocol requirements
- Programmable SPI clock rate; maximum rate is 9 MHz for reads from the modem and 16 MHz for writes to the modem
- Transmit and recieve FIFOs at MCU
- Serial clock phase and polarity must meet modem requirements (MCU control bis CPHA=0 and CPOL=0)
- Slave select programmed to meet modem protocol
- MSB-first shifting

8.3 SPI System Operation

The MCU operates as SPI master, initiating all SPI transfers. The MCU selects the modem as the slave device by driving the SPI1_PCS0 signal low and outputting the clock on SPI1_SCK. During a transfer, the master shifts data out (on the SPI1_SOUT pin) to the modem while simultaneously shifting data in (on the SPI1_SIN pin) from the modem. Although the SPI interface supports simultaneous data exchange between the master and slave, the modem SPI protocol only uses data exchange in one direction at a time: either to write data to the modem, or read data from the modem.

8.4 Modem SPI Overview

The modem's SPI interface allows an MCU to communicate with the modem's register set and Packet Buffer. The modem's SPI is a slave-only interface; the MCU must drive R_SSEL_B, R_SCLK and R_MOSI. Write and read access to both Direct and Indirect registers is supported, and transfer length can be single-byte, or bursts of unlimited length. Write and read access to the Packet buffer can also be single-byte, or a burst mode of unlimited length. The modem's SPI interface is asynchronous to the rest of the IC. No relationship between R_SCLK and the modem's internal oscillator is assumed. And no relationship between R_SCLK and the CLK_OUT pin is assumed. All synchronization of the SPI interface to the modem IC takes place inside the SPI module, alleviating the burden on other block designers to provide this synchronization internal to their modules. SPI synchronization takes place in both directions: SPI-to-IC (register writes), and IC-to-SPI (register reads). The SPI is capable of operation in all modem Power Modes, except Reset. Operation in Hibernate mode allows most modem registers, and the complete Packet Buffer, to be accessed in the modem's lowest-power operational state, enabling minimal power consumption, especially during the register-initialization phase of the IC.

The SPI design features a compact, single-byte control word, reducing SPI access latency to a minimum. Most SPI access types require only a single-byte control word, with the address embedded in the control word. During control word transfer (the first byte of any SPI access), the contents of the IRQSTS1 register (the modem's highest-priority status register) are are always shifted out, so that the MCU gets access to IRQSTS1, with the minimum possible latency, on every SPI access.

8.4.1 Features

- 4-wire industry standard interface
- SPI R_SCLK maximum frequency is 9 MHz for reads and 16 MHz for writes
- Write and read access to all modem registers (Direct and Indirect)
- Write and read access to Packet Buffer
- SPI accesses can be single-byte or burst.
- Automatic address auto-incrementing for burst accesses
- The entire Packet Buffer can be uploaded or downloaded in a single SPI burst.
- Entire Packet Buffer, and most registers, can be accessed in Hibernate mode
- Built-in synchronization inside the SPI module to/from the rest of the IC.
- R_MISO can be tri-stated when SPI inactive, enabling multi-slave configurations

8.5 Modem SPI Basic Operation

The modem operates as a SPI slave only. The microcontroller supplies the interface clock and acts as SPI master.

8.5.1 Modem SPI Pin Definition

8.5.1.1 Radio Select SEL (R_SSEL_B)

The MCU Master SPI selects and activates the modem's Slave SPI by asserting R_SSEL_B (low).

8.5.1.2 Radio SPI Clock (R_SCLK)

The MCU Master SPI provides a serial bit clock to the modem's Slave SPI on R_SCLK.

8.5.1.3 Radio MOSI - Master Data Out Slave Data In (R_MOSI)

The MCU Master SPI transmits Control, Address and Write data to the modem's SPI Slave on R_MOSI.

8.5.1.4 Radio MISO - Master Data In, Slave Data Out (R_MISO)

The modem's Slave SPI transmits readback data to the MCU Master SPI on R_MISO.

8.5.1.4.1 Setting R_MISO Off Impedance

MISO_HIZ_EN - Determines the output state of the modem's R_MISO output when R_SSEL_B is deasserted (high).

1: R_MISO output is tristated (default)

0: R_MISO output is driven low by the SPI slave

8.5.1.4.2 R_MISO Pull-up Requirement

Because the modem tri states its R_MISO output when a SPI transfer is not in progress (R_SSEL_B deasserted), a pull-up is required to ensure that the SPI Master does not see a floating MISO input.

In a configuration consisting of only a single Master (MCU) and single Slave (the modem), this pull-up can be an internal pull-up on the MCU's MISO input. Moreover, after initialization, the modem's MISO_HIZ_EN bit can be cleared (enabling the modem to drive R_MISO at all times), and the MCU pull-up can be disabled.

In a configuration consisting of more than just 1 SPI master and 1 SPI slave, the pull-up on R_MISO should be external, and the MISO_HIZ_EN bit should be left in its default state.

8.5.2 Modem SPI Timing

8.5.2.1 SPI Transfer Protocol

The modem's SPI is a slave-only interface, and follows a CPHA=0 and CPOL=0 protocol. Taken together, CPHA and CPOL determine the clock polarity and clock edge used to transfer data between the SPI master and slave, in both communication directions.

CPHA=0 (clock phase) indicates that data is captured on the leading edge of SCK and changed on the following edge. CPOL=0 (clock polarity) indicates the inactive state value of R_SCLK is low. All data is transferred MSB-first. The following diagram depicts the transfer protocol of the modem's SPI.

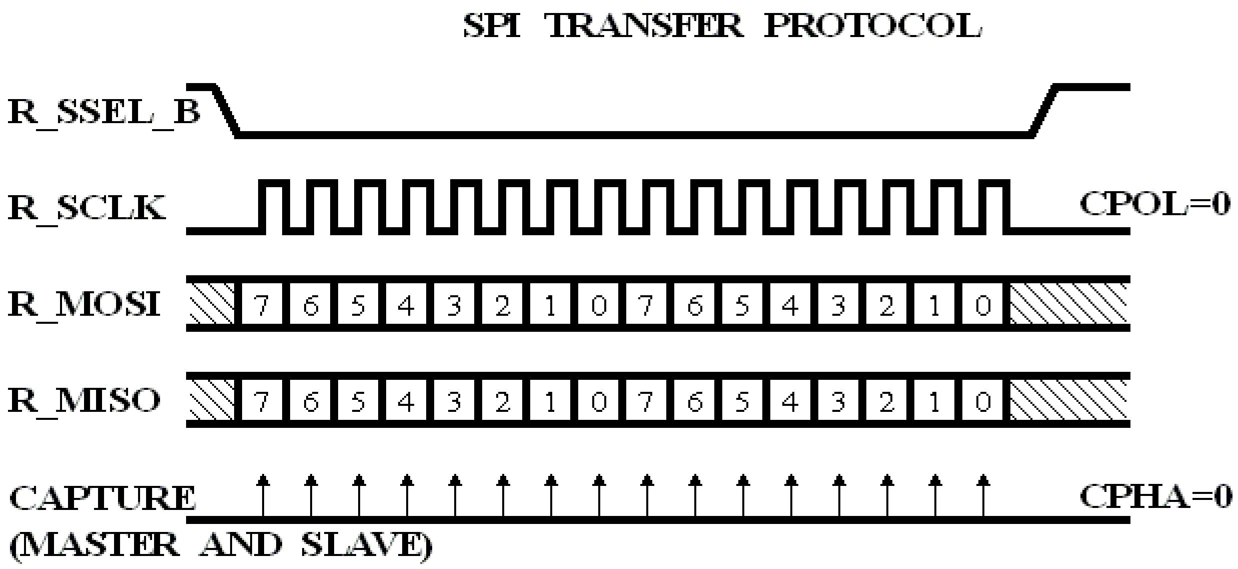


Figure 8-2. Transfer Protocol of the modem's SPI

8.5.2.2 SPI Timing: R_SSEL_B to R_SCLK

The following diagram describes timing constraints that must be guaranteed by the system designer.

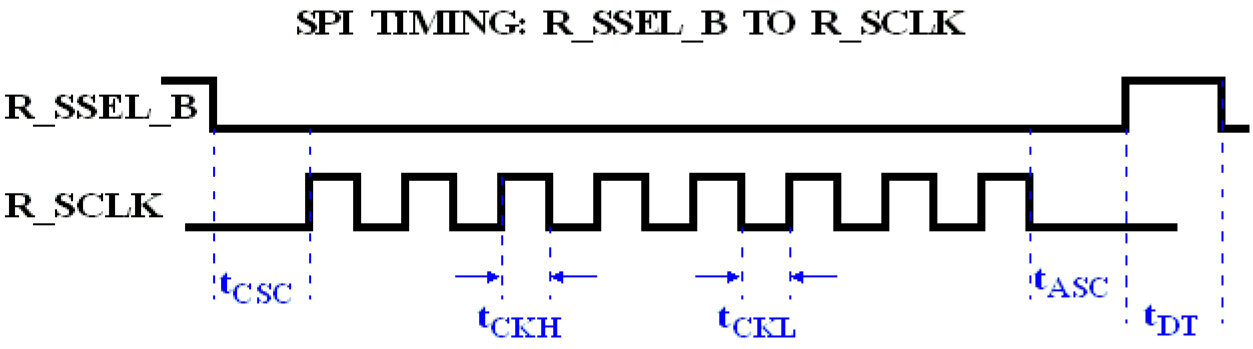


Figure 8-3. SPI Timing of R_SSEL to R_SCLK

tCSC (CS-to-SCK delay): 31.25ns

tASC (After SCK delay): 31.25ns

tDT (Minimum CS idle time): 62.5ns

tCKH (Minimum SCLK high time): 31.25ns (for SPI writes); 55ns (for SPI reads)

tCKL (Minimum SCKH high time): 31.25ns (for SPI writes); 55ns (for SPI reads)

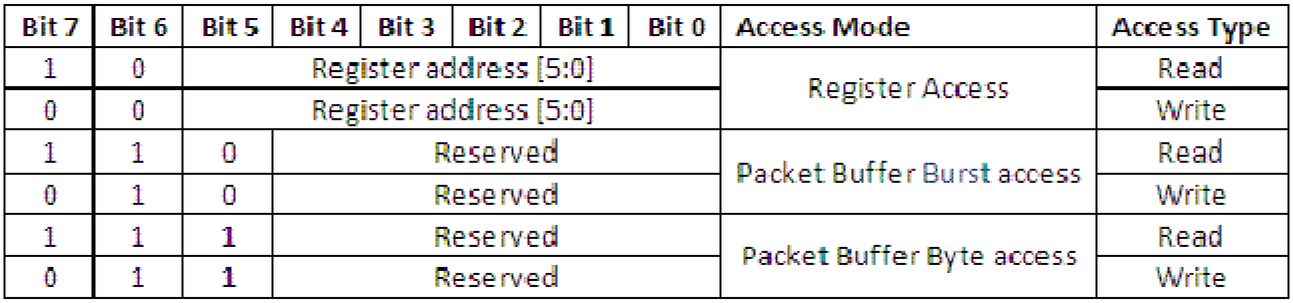
NOTE

The SPI Master device shall only deassert R_SSEL_B on byte boundaries, and only after guaranteeing the tASC constraint shown above.

8.6 Modem SPI Transactions

8.6.1 SPI Control Word

Every SPI transaction begins with one single-byte control word. The control word consists of the address, direction (write or read), transaction target (register or Packet Buffer), and for the Packet Buffer, the access mode (burst or byte). For most transactions, data transfer follows immediately after the control word. However, for Indirect Access Register transactions, and for Packet Buffer byte-mode transactions, an additional address byte follows the control word, before data transfer begins. Bit 7 of the control word selects the transfer direction (1=READ, 0=WRITE). Bit 6 selects the transfer target (1=PACKET BUFFER, 0=REGISTER). For Register accesses, the remaining bits select the register address. For Packet Buffer Access, bit 5 selects the access mode (1=BYTE, 0=BURST). For Packet Buffer Access, the remaining bits in the control word are reserved and ignored by the modem. Details and examples of control word usage appear in the following sections. The following table depicts an overview of the control word.



Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access Mode	Access Type
1	0	Register address [5:0]						Register Access	Read
0	0	Register address [5:0]							Write
1	1	0	Reserved					Packet Buffer Burst access	Read
0	1	0	Reserved						Write
1	1	1	Reserved					Packet Buffer Byte access	Read
0	1	1	Reserved						Write

Figure 8-4. Control Word Overview

8.6.2 Direct Register Write Access (single byte)

The following diagram depicts a single-byte write access to a Direct Register in the modem.

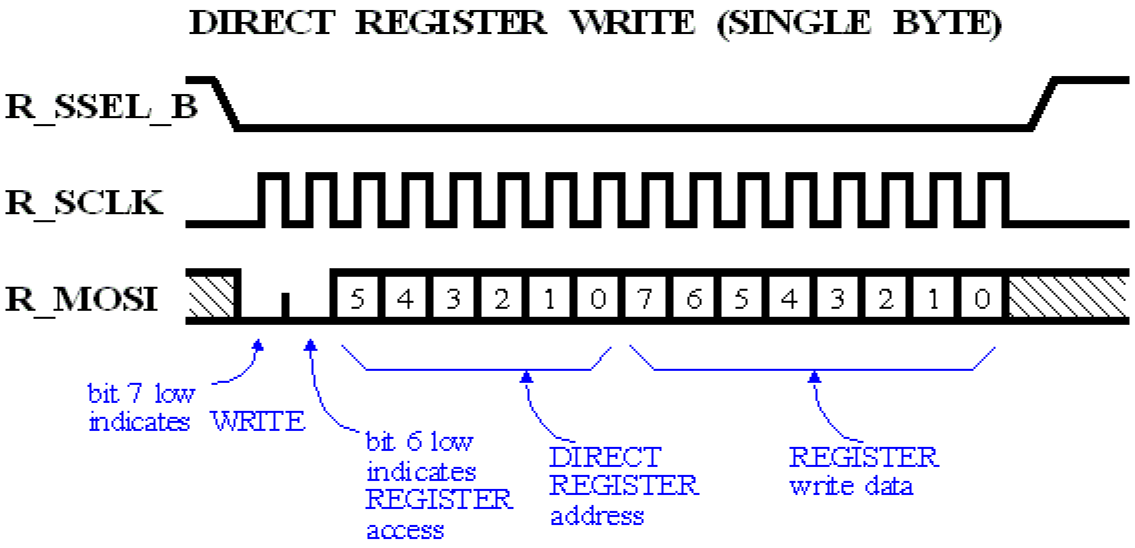

DIRECT REGISTER WRITE (SINGLE BYTE)
R_SSEL_B
R_SCLK
R_MOSI
5 4 3 2 1 0 7 6 5 4 3 2 1 0
bit 7 low indicates WRITE
bit 6 low indicates REGISTER access
DIRECT REGISTER address
REGISTER write data

Figure 8-5. Single-Byte Write Access to a Direct Register

8.6.3 Direct Register Read Access (single byte)

The following diagram depicts a single-byte read access to a Direct Register in the modem.

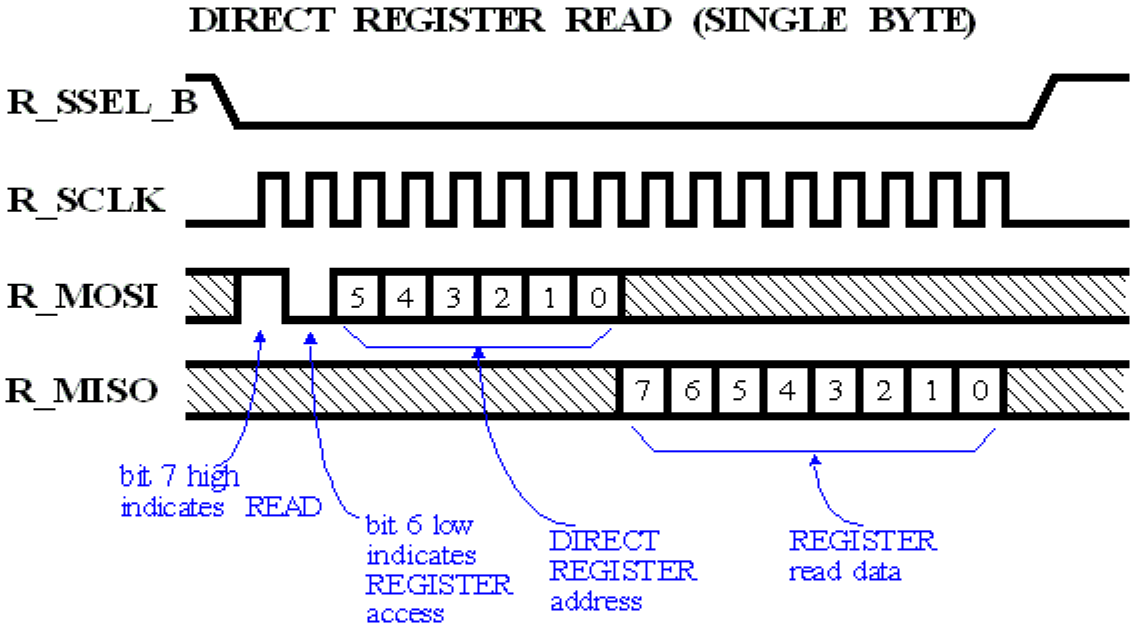

DIRECT REGISTER READ (SINGLE BYTE)
R_SSEL_B
R_SCLK
R_MOSI
5 4 3 2 1 0
R_MISO
7 6 5 4 3 2 1 0
bit 7 high indicates READ
bit 6 low indicates REGISTER access
DIRECT REGISTER address
REGISTER read data

Figure 8-6. Single-Byte read Access to a Direct Register

8.6.4 Direct Register Write Access (multi byte)

The following diagram depicts a multi-byte write access to Direct Register space in the modem.

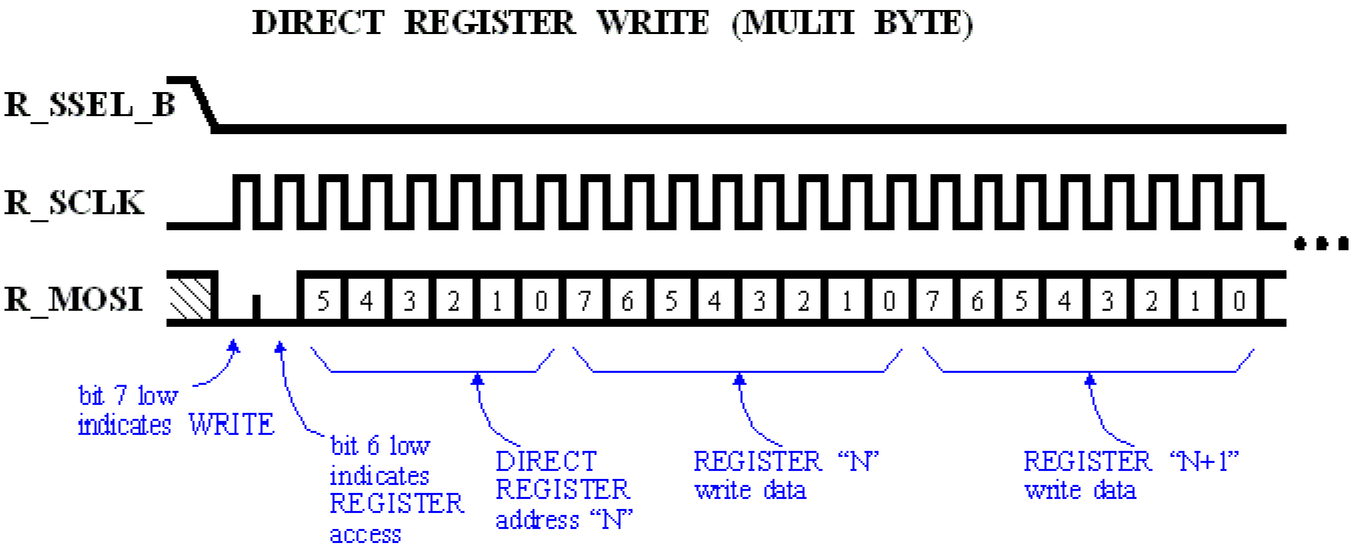

DIRECT REGISTER WRITE (MULTI BYTE)
R_SSEL_B
R_SCLK
R_MOSI
5 4 3 2 1 0 7 6 5 4 3 2 1 0 7 6 5 4 3 2 1 0
bit 7 low indicates WRITE
bit 6 low indicates REGISTER access
DIRECT REGISTER address "N"
REGISTER "N" write data
REGISTER "N+1" write data

Figure 8-7. Multi-Byte Write Access to a Direct Register Space

8.6.5 Direct Register Read Access (multi byte)

The following diagram depicts a multi-byte read access to Direct Register space in the modem.

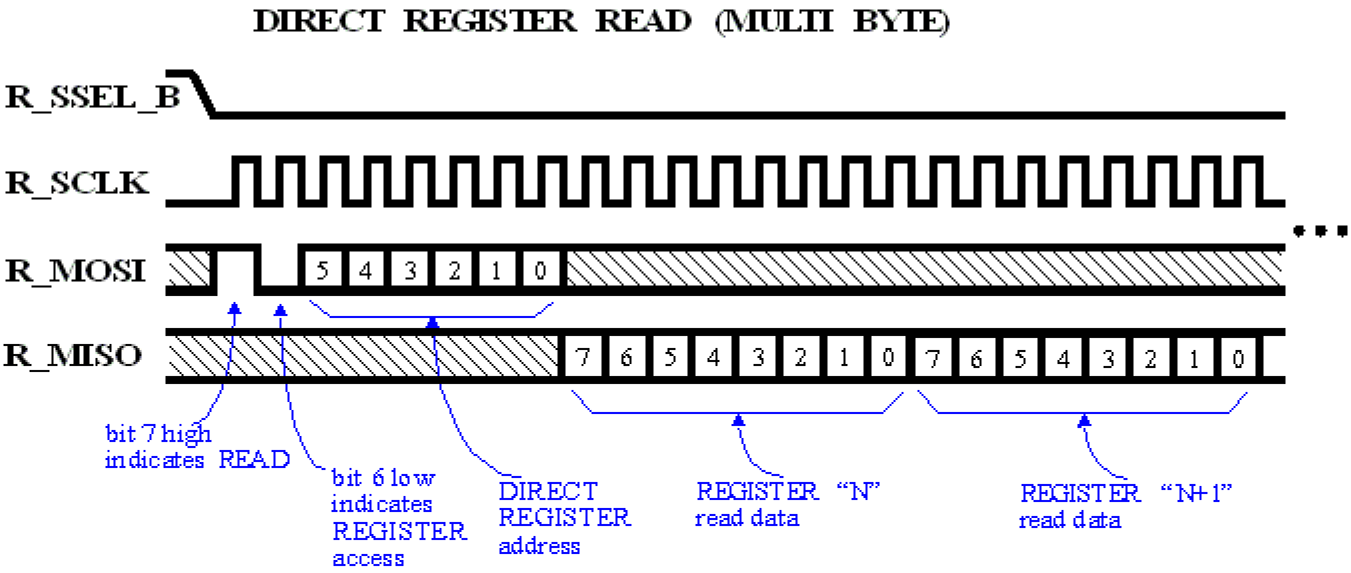


Figure 8-8. Multi-Byte Read Access to a Direct Register Space

8.6.6 Indirect Register Write Access (multi byte)

The following diagram depicts a multi-byte write access to Indirect Register space in the modem.

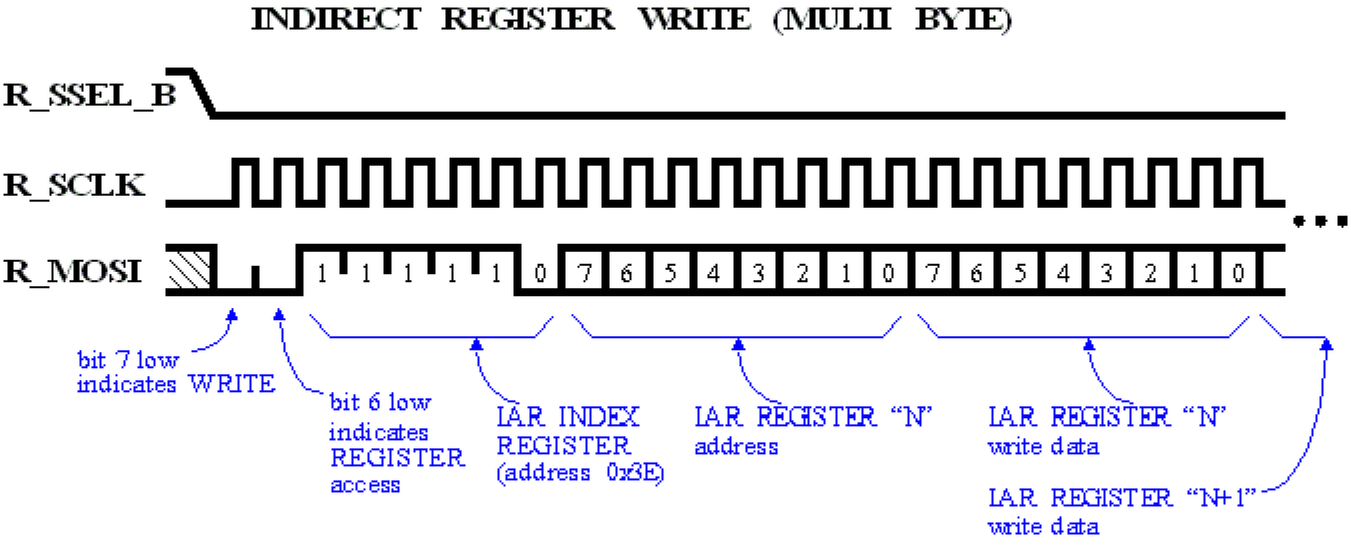

INDIRECT REGISTER WRITE (MULTI BYTE)
R_SSEL_B
R_SCLK
R_MOSI
1 1 1 1 1 0 7 6 5 4 3 2 1 0 7 6 5 4 3 2 1 0
bit 7 low indicates WRITE
bit 6 low indicates REGISTER access
IAR INDEX REGISTER (address 0x3E)
IAR REGISTER "N" address
IAR REGISTER "N" write data
IAR REGISTER "N+1" write data

Figure 8-9. Multi-Byte Write Access to an Indirect Register Space

NOTE

Multi-byte SPI transactions to an Indirect Address space are not permitted in Hibernate state. Use single-byte transactions only.

8.6.7 Indirect Register Read Access (multi byte)

The following diagram depicts a multi-byte read access to Indirect Register space in the modem.

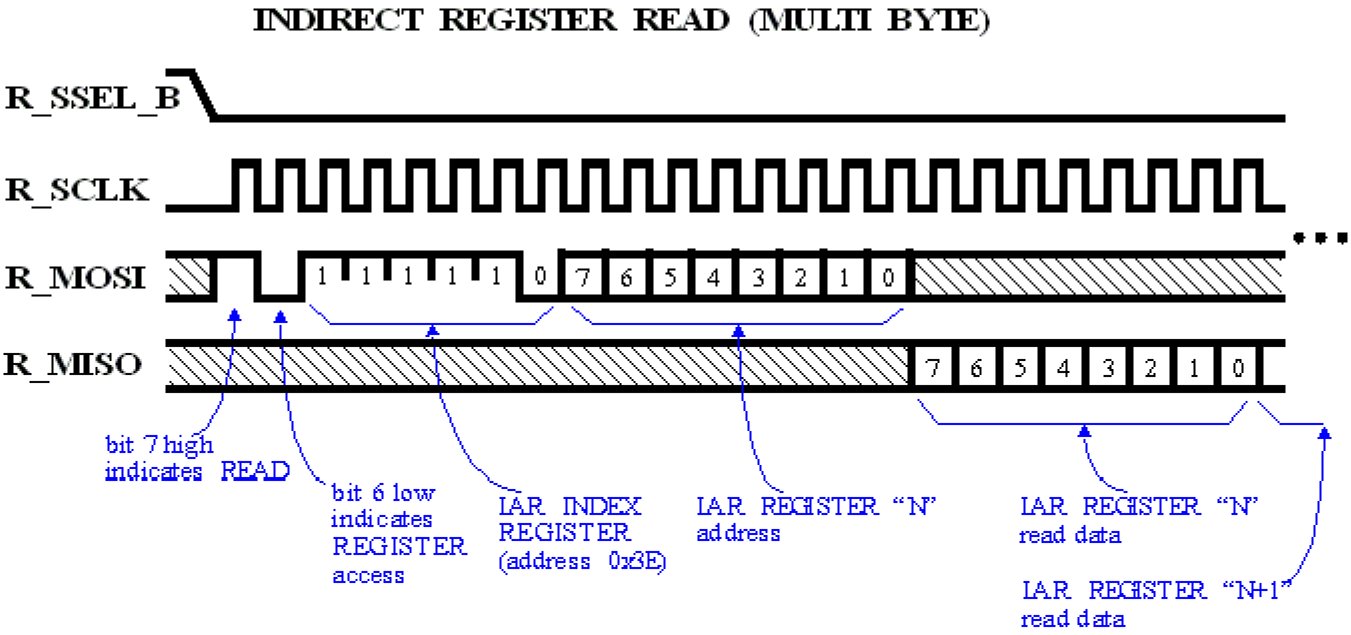

INDIRECT REGISTER READ (MULTI BYTE)
R_SSEL_B
R_SCLK
R_MOSI
1 1 1 1 1 0 7 6 5 4 3 2 1 0
R_MISO
7 6 5 4 3 2 1 0
bit 7 high indicates READ
bit 6 low indicates REGISTER access
IAR INDEX REGISTER (address 0x3E)
IAR REGISTER "N" address
IAR REGISTER "N" read data
IAR REGISTER "N+1" read data

Figure 8-10. Multi-Byte Read Access to an Indirect Register Space

NOTE

Multi-byte SPI transactions to Indirect Address space are not permitted in Hibernate state. Use single-byte transactions only.

8.6.8 Synchronous and Asynchronous Operating Modes

The SPI module is designed to be fully operational in all modem power states, except reset. Operation in the Hibernate state means the modem's crystal oscillator is disabled. Most modem registers, direct and indirect, can be accessed, both read and write, during HibernateIn all non-Hibernate states, the SPI operates in "synchronous mode"; to write a register, the SPI module generates a clock enable to the Register block, where the enable is used to gate the 32MHz oscillator to each individual register; for reads, the readback data is registered using the 32MHz clock at the end of the SPI address phase, so that readback data cannot change during shift-out on R_MISO. In Hibernate, the SPI operates in "asynchronous mode"; to write a register, the SPI module generates both clock and clock enable to the Register block (because the crystal oscillator is disabled); for reads, no latching of readback data takes place, since the oscillator is disabled, readback data cannot change.

The SPI module automatically determines its operating state, based on the xtal_ready signal from the crystal oscillator. This signal is captured by the SPI module during the SPI address phase, during the first R_SCLK period. If it is high, the SPI operates in synchronous mode (as described above) for the duration of the SPI transaction. If xtal_ready is captured low, the SPI operates in asynchronous mode (as described above) for the duration of the SPI transaction. The transaction ends when R_SSEL_B is deasserted.

Transitions on xtal_ready are triggered by a SPI write access to the PWR_MODES register. A SPI write to clear the XTALEN bit of the PWR_MODES register will cause an immediate high-to-low transition of xtal_ready; this transition occurs at the end of the SPI transfer, after the last data bit is shifted in on R_MOSI. No further registers can be written during this transfer, even if the transaction is multi-byte, because the PWR_MODES registers is at the end of Direct Address space. A SPI write to set the XTALEN bit of PWR_MODES register will cause a delayed low-to-high transition on xtal_ready, because there is a finite crystal warm-up time. The SPI module will continue to operate in asynchronous mode, until a new SPI transaction starts with xtal_ready asserted.

8.6.9 Shifting Out IRQSTS1 During Control Word

During each SPI transaction, while the control word is being shifted in on R_MOSI (SPI address phase), the SPI module always shifts out the contents of Direct Register 0 "IRQSTS1" on R_MISO. The IRQSTS1 register is the highest-priority register in the modem; it contains the transceiver interrupt status bits, and must be accessed with low latency on any 802.15.4 interrupt. Because multiple registers must be accessed on transceiver interrupts, Direct Register space has been designed to put the highest-priority registers at the beginning of address space. A single, multi-byte transfer can be performed to access all of these registers. For such interrupt-driven SPI transfers, software can take advantage of the automatic shifting of IRQSTS1 during the SPI address phase, by programming a multi-byte SPI transfer to begin at Address 1 (not Address 0), because Address 0 data is provided "for free" on R_MISO during the control word phase. Thus, a 7-byte SPI transfer (to read Direct Registers 0 through 6) would require only 7 x 8 = 56 SPI clock periods, instead of 64 clock periods, which would be the case if Register 0 had to be addressed explicitly during such a transaction. The following diagram depicts the latency advantage that can be gained using this approach.

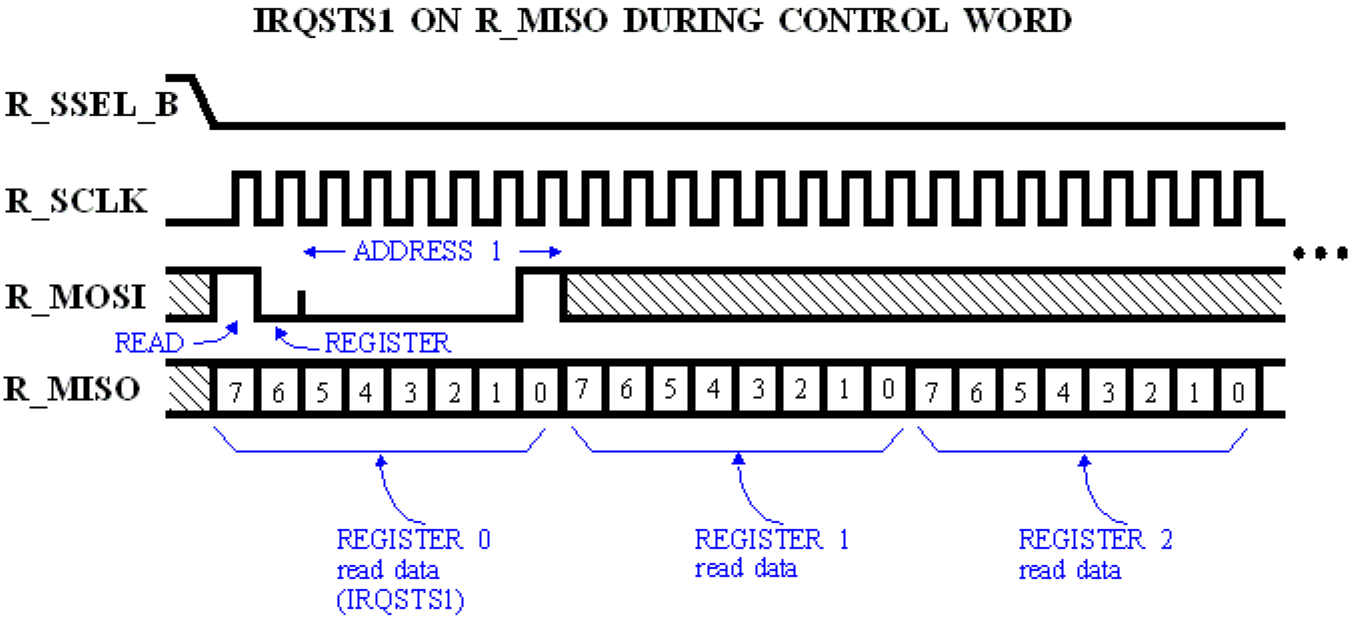


Figure 8-11. IRQSTS1 on R_MISO During Control Word

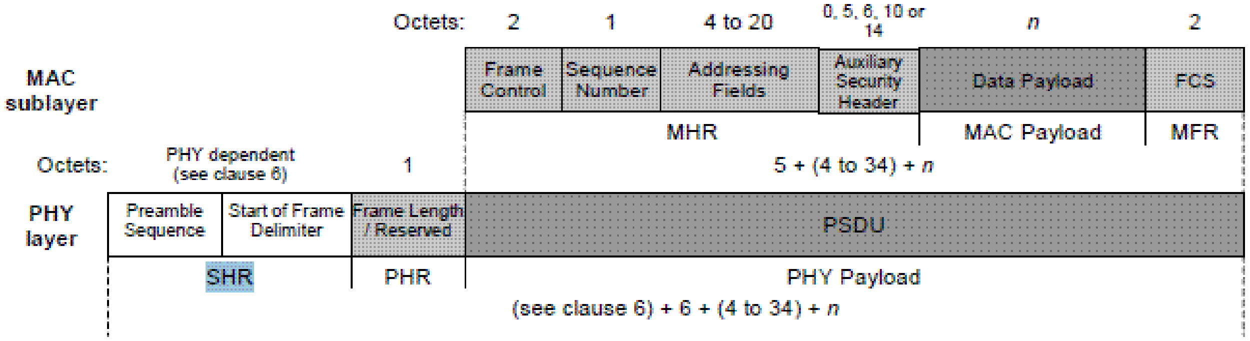
8.6.10 Packet Buffer

The Packet Buffer is a 128-byte Random Access Memory (RAM) dedicated to the storage of 802.15.4 packet contents for both TX and RX sequences. For TX sequences, software stores the contents of the packet buffer starting with the Frame Length byte at Packet Buffer Address 0, followed by the packet contents at the subsequent Packet Buffer Addresses. For RX sequences the incoming packet's Frame Length is stored in a register, external to the Packet Buffer.Software will read this register to determine the number of bytes of Packet Buffer to read. This facilitates DMA transfer through the SPI. For receive packets, an LQI byte is stored at the byte immediately following the last byte of the packet (Frame Length +1).

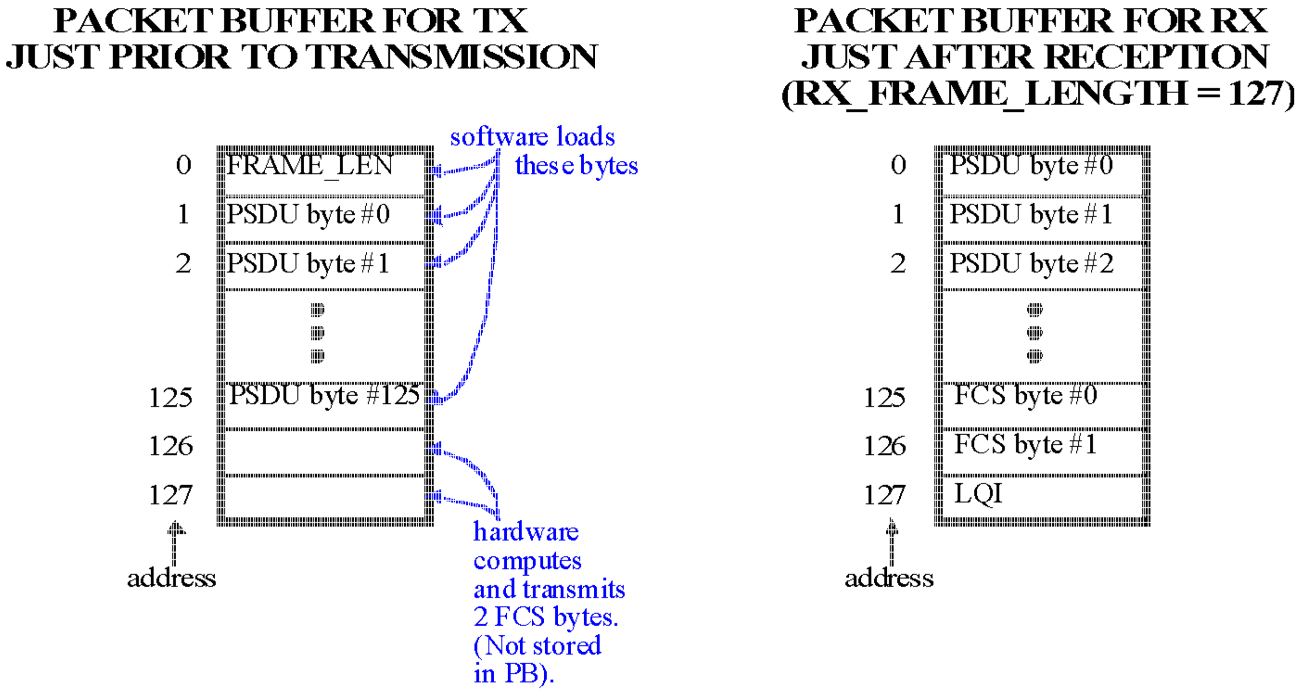
The Packet Buffer is not a FIFO. It is a byte-addressable memory capable of burst-mode transfers. Reading from the Packet Buffer does not destroy its contents. As long as no subsequent received packets are received, the Packet Buffer may be downloaded as many times as desired and the contents will not change.

8.6.10.1 Packet Buffer Architecture

The 802.15.4 standard specifies a maximum packet size of 127 bytes. The Packet Buffer is sized to accommodate a maximum-length packet, plus one additional byte. For TX, this additional byte is the “Frame Length” byte, or PHR, as it is known in 802.15.4 nomenclature. Software loads the Frame Length byte into address 0 of the Packet Buffer (the first data phase of a multi-byte SPI burst). The packet contents (PSDU) follow. The following diagram depicts the 802.15.4 packet structure, detailing the PHR and PSDU.



The last 2 bytes of PSDU are CRC bytes for the 802.15.4 FCS (Frame Check Sequence). The radio hardware computes and transmits these 2 bytes automatically, so software need not load them into the Packet Buffer. Thus, for TX sequences, the number of bytes to load into the buffer, after Frame Length, is “Frame Length - 2”. For RX, the received Frame Length is routed to a SPI-readable register, RX_FRAME_LENGTH, not the Packet Buffer. The remainder of the packet’s PSDU, is stored in the Packet Buffer, starting at address 0. At Data Indication, software should read the RX_FRAME_LENGTH register to determine the number of bytes of Packet Buffer it needs to read, in order to download the entire packet via SPI transfer. A DMA transfer can be set up to perform the transfer, if the MCU is so equipped. The PSDU includes the 2 received FCS bytes. For RX packets, after the last packet FCS byte, an LQI byte is stored. Therefore, to download the entire packet plus LQI from the packet buffer, the number of bytes (data phases) in the SPI burst will be RX_FRAME_LENGTH+1, where RX_FRAME_LENGTH is the contents of the register of the same name. The following diagram depicts the contents of a TX buffer containing a maximum-length packet (just prior to transmission), and an RX buffer containing a maximum length packet (just after a data indication).



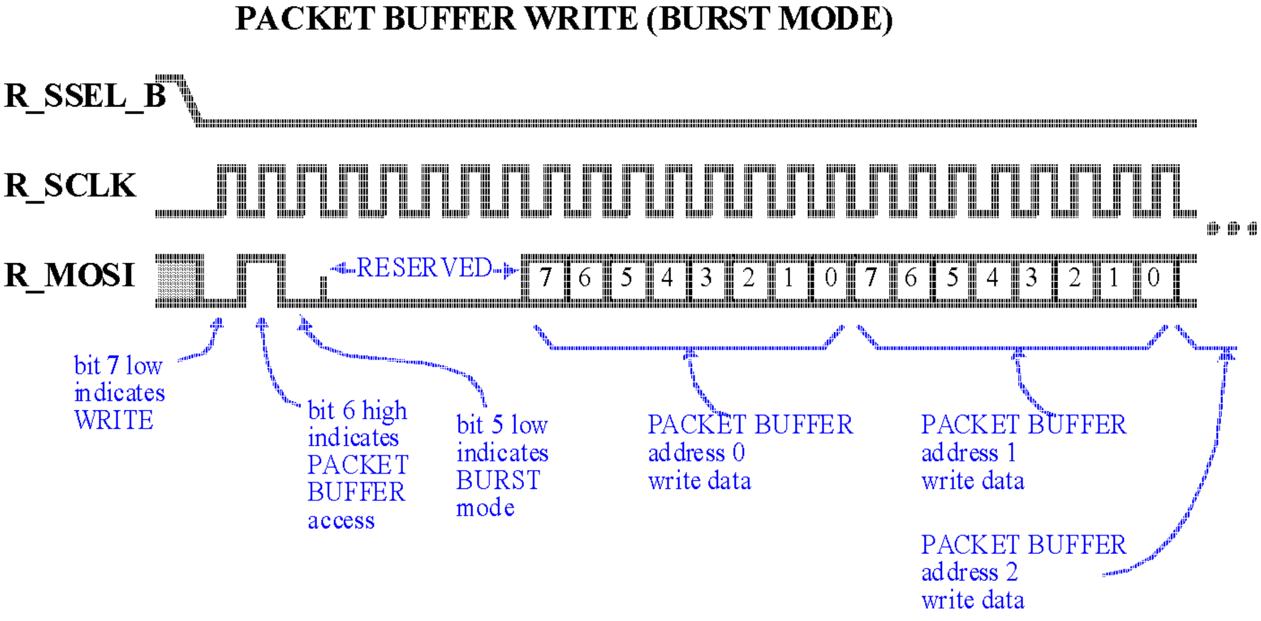
8.6.10.2 Packet Buffer SPI Transactions

8.6.10.2.1 Packet Buffer Access Modes: Burst Mode vs Byte Mode

Burst mode is recommended for Packet Buffer uploading (TX) and downloading (RX). For Packet Buffer Burst mode, no Packet Buffer address need be provided by the SPI. SPI hardware will always start the burst at Packet Buffer address 0. A single SPI transfer of up to 128 bytes can be used for either uploading or downloading in burst mode. This, in combination with DMA, reduces SPI latency to an absolute minimum.

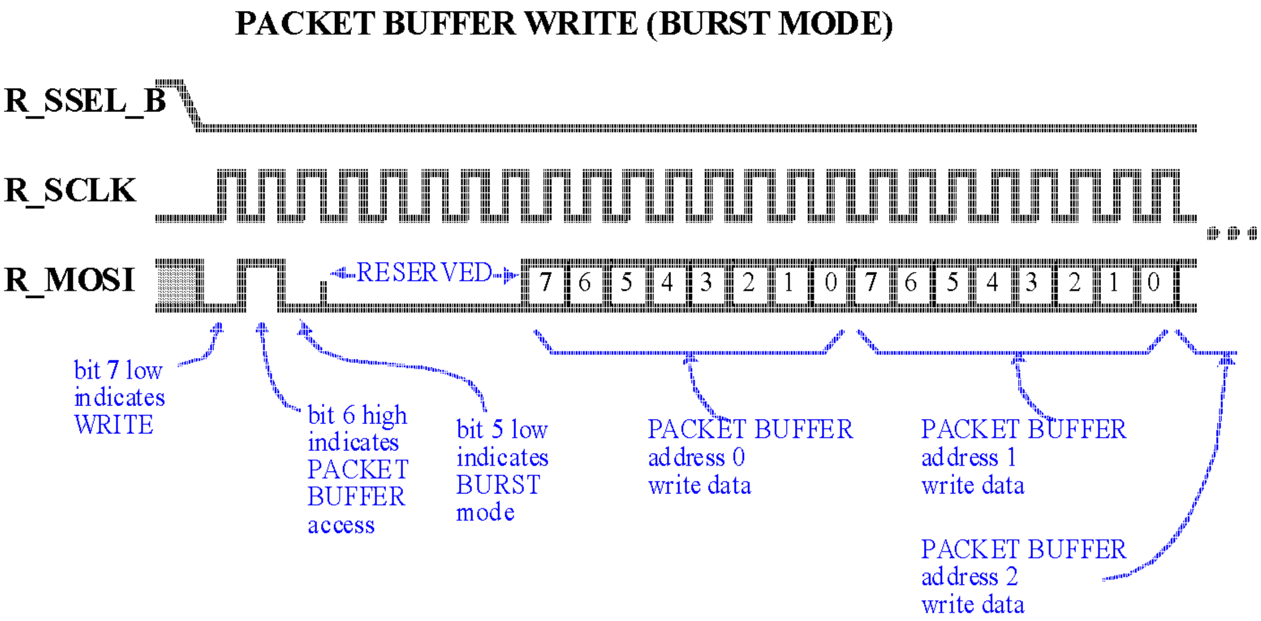
In case the user desires to access an individual byte within the Packet Buffer, and not the entire buffer, a "Byte" access mode has also been provided. In this mode, the SPI transfer's second byte indicates the starting Packet Buffer address for the transfer. The subsequent bytes write (or read) Packet Buffer contents, starting at the indicated address. Byte mode is less efficient than Burst mode, because an address byte must follow the SPI control word.

In both Byte and Burst modes, there is no hardware limit to the number of bytes that can be uploaded or downloaded through the SPI. If a SPI transfer of greater than 128 bytes is attempted, the Packet Buffer address will wrap around to 0 after Packet Buffer address 127 has been accessed.


PACKET BUFFER WRITE (BURST MODE)
R_SSEL_B
R_SCLK
R_MOSI
RESERVED
7 6 5 4 3 2 1 0 7 6 5 4 3 2 1 0
bit 7 low indicates WRITE
bit 6 high indicates PACKET BUFFER access
bit 5 low indicates BURST mode
PACKET BUFFER address 0 write data
PACKET BUFFER address 1 write data
PACKET BUFFER address 2 write data

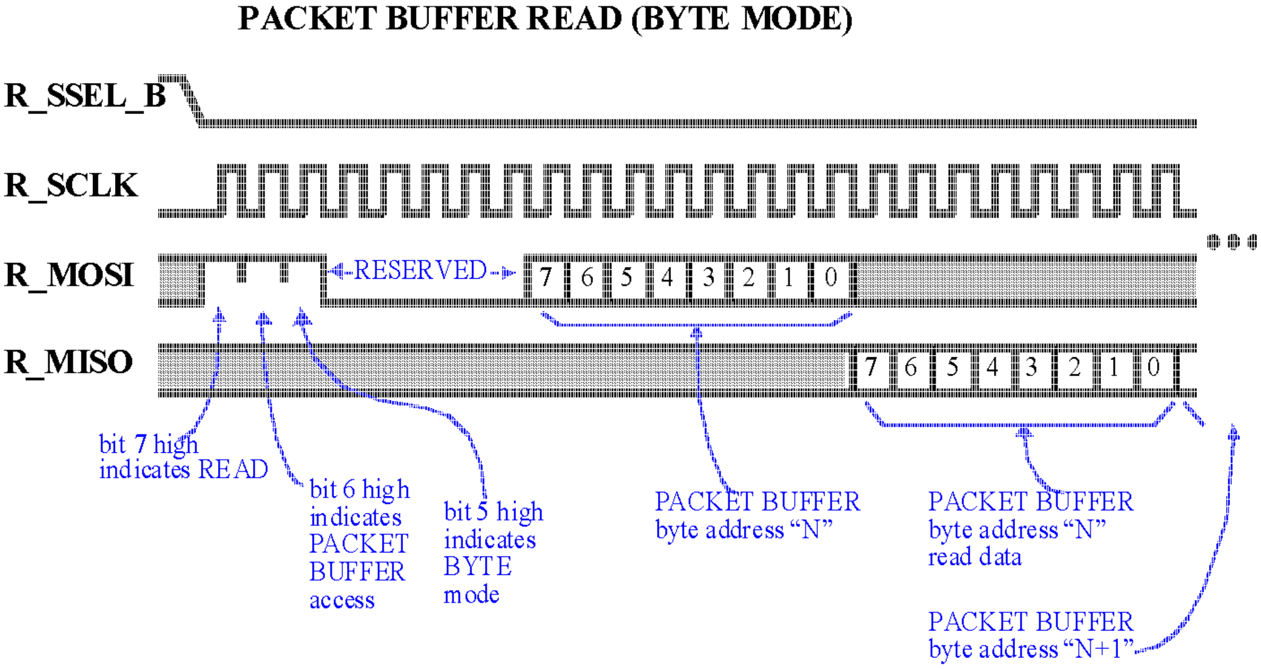
8.6.10.2.2 Packet Buffer Write Access (Burst Mode)

The following diagram depicts a SPI burst-mode write access to the Packet Buffer.


PACKET BUFFER WRITE (BURST MODE)
R_SSEL_B
R_SCLK
R_MOSI
RESERVED
7 6 5 4 3 2 1 0 7 6 5 4 3 2 1 0
bit 7 low indicates WRITE
bit 6 high indicates PACKET BUFFER access
bit 5 low indicates BURST mode
PACKET BUFFER address 0 write data
PACKET BUFFER address 1 write data
PACKET BUFFER address 2 write data

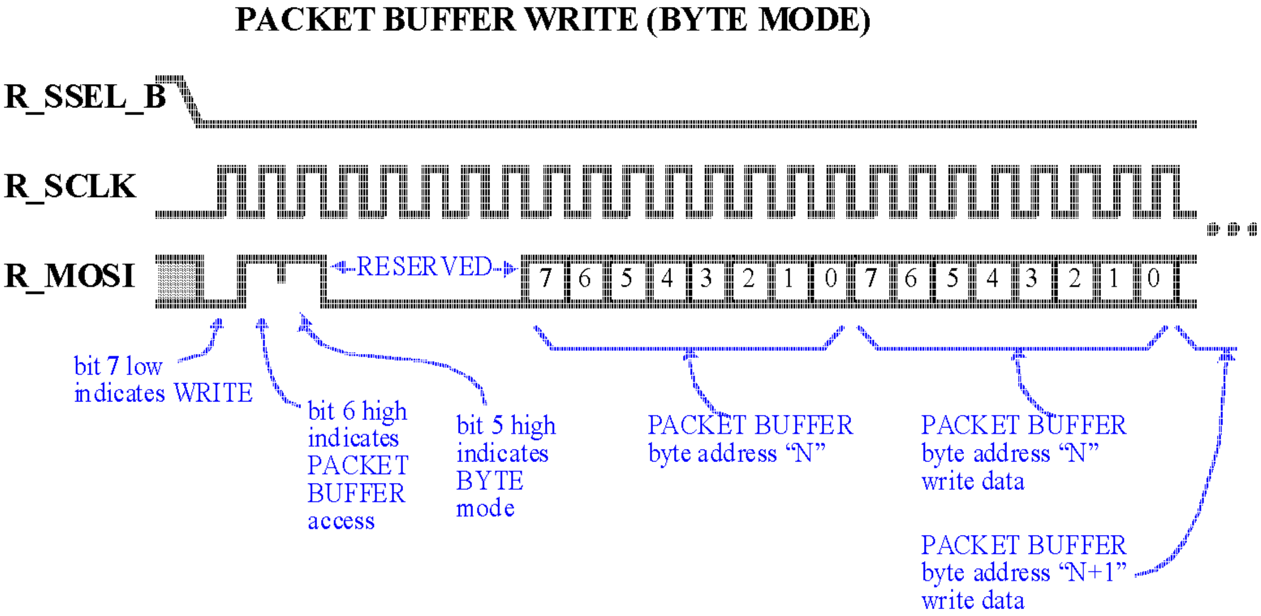
8.6.10.2.3 Packet Buffer Read Access (Burst Mode)

The following diagram depicts a SPI burst-mode read access to the Packet Buffer.


PACKET BUFFER READ (BYTE MODE)
R_SSEL_B
R_SCLK
R_MOSI
R_MISO
RESERVED
7 6 5 4 3 2 1 0
7 6 5 4 3 2 1 0
bit 7 high indicates READ
bit 6 high indicates PACKET BUFFER access
bit 5 high indicates BYTE mode
PACKET BUFFER byte address "N"
PACKET BUFFER byte address "N" read data
PACKET BUFFER byte address "N+1"

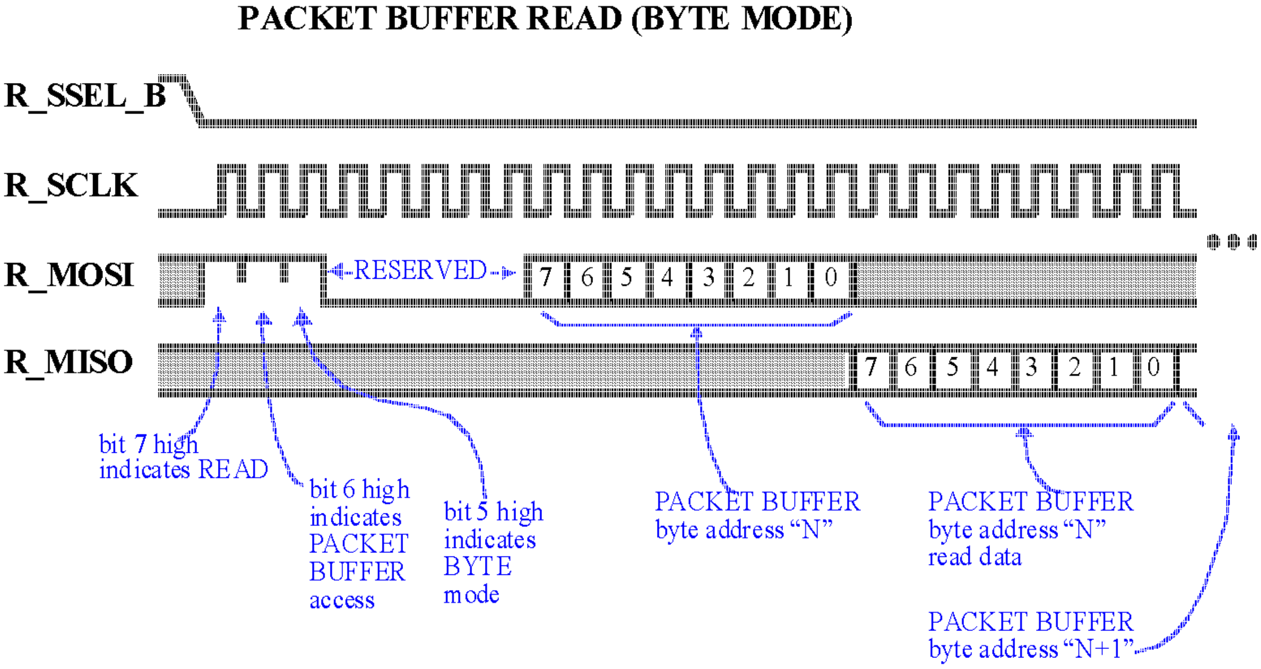
8.6.10.2.4 Packet Buffer Write Access (Byte Mode)

The following diagram depicts a SPI byte-mode write access to the Packet Buffer.


PACKET BUFFER WRITE (BYTE MODE)
R_SSEL_B
R_SCLK
R_MOSI
RESERVED
7 6 5 4 3 2 1 0 7 6 5 4 3 2 1 0
bit 7 low indicates WRITE
bit 6 high indicates PACKET BUFFER access
bit 5 high indicates BYTE mode
PACKET BUFFER byte address "N"
PACKET BUFFER byte address "N" write data
PACKET BUFFER byte address "N+1" write data

8.6.10.2.5 Packet Buffer Read Access (Byte Mode)

The following diagram depicts a SPI byte-mode read access to the Packet Buffer.


PACKET BUFFER READ (BYTE MODE)
R_SSEL_B
R_SCLK
R_MOSI
R_MISO
RESERVED
7
6
5
4
3
2
1
0
7
6
5
4
3
2
1
0
bit 7 high indicates READ
bit 6 high indicates PACKET BUFFER access
bit 5 high indicates BYTE mode
PACKET BUFFER byte address "N"
PACKET BUFFER byte address "N" read data
PACKET BUFFER byte address "N+1"

8.6.10.3 Handling of Auto-RXACK Packets

According to 802.15.4 protocol, when a packet is transmitted by a device, an acknowledge packet is often expected from the intended recipient. The radio features a hardware autosequence to transmit a packet, and wait for an acknowledge packet. This autosequence is initiated by programming a “Sequence TR” with RXACKRQD=1. During this autosequence, the received acknowledge packet is not stored in the Packet Buffer. The Packet Processor (hardware) merely checks the incoming packet for correct Sequence Number and FCS, and notifies software when the correct Acknowledge packet is received.

However, in Active Promiscuous mode, all received packets must be made available to software. Therefore in Active Promiscuous mode, auto-RXACK packets will be stored in the Packet Buffer.

8.6.10.4 Downloading Packet Buffer Contents During Reception

For packet reception, it is recommended that software wait for Data Indication prior to downloading Packet Buffer contents. This approach takes full advantage of the radio’s Packet Processor hardware to check if the incoming packet was intended for the device, and for the FCS-check logic to verify CRC. However, hardware does not prohibit reading of Packet Buffer content while a packet is being received. Early downloading can mean earlier software access to packet contents for software processing, which can be advantageous during periods of high software workload or short turnaround requirements. To download the Packet Buffer during reception, the following steps should be followed:

1. Set the RX_WTR_MARK register to trigger a RXWTRMRKIRQ interrupt after the Nth byte of the packet has been received. The RX_WTR_MARK threshold corresponds directly to Packet Buffer address, plus 1. So, if an RXWTRMRKIRQ is desired after Packet Buffer address 7 has been loaded with receive data, set RX_WTR_MARK=8.
2. Unmask the RXWTRMRKIRQ by programming RX_WMRK_MSK=0 in the PHY_CTRL2 register.
3. Unmask the PB_ERR_IRQ by programming PB_ERR_MSK=0 in the PHY_CTRL3 register.
4. At the RXWTRMRKIRQ interrupt, read the RX_FRAME_LENGTH register to determine the ultimate length of the packet. At this point, the entire Packet Buffer has not been loaded with receive data, but the first “N” bytes (at least) are available, where N = RX_WTR_MARK-1.
5. Begin the SPI burst transfer of length “RX_FRAME_LENGTH”, (or “RX_FRAME_LENGTH+1” to also get LQI) to download the Packet Buffer contents.

Caution must be used when using early downloading:

1. At Data Indication (which will occur some time after RXWTRMRKIRQ), software must check the validity of the received packet. At SEQIRQ, if RXIRQ is also asserted (IRQSTS1 register), then the packet passed filtering and FCS. Otherwise, it failed either filtering, CRC, or both.
2. It is responsibility of the user to select a combination of SPI Transfer Rate, and RX_WTR_MARK threshold, to ensure that the SPI does not attempt to read from Packet Buffer addresses that have not yet been loaded with receive data (i.e., an underrun condition). The SPI transfer rate is determined by the SCLK frequency of the SPI Master. Receive data is loaded into the Packet Buffer at a constant rate of 1 byte every 32us, so in general the SPI transfer rate will be considerably faster than the buffer loading rate. Software must calculate the number of bytes to be downloaded (based on RX_FRAME_LENGTH), and then set the RX_WTR_MARK threshold based on the ratio of SPI transfer rate to buffer loading rate.

If the steps above have not been followed correctly, and a Packet Buffer underrun occurs, this means the SPI read address of the Packet Buffer has overtaken the RX demodulator write address. A PB_ERR_IRQ interrupt will be raised to indicate this condition. A PB_ERR_IRQ is not catastrophic. The following steps should be taken to recover from a PB_ERR_IRQ:

1. Abort the SPI transfer in progress, and discard the data received so far, because the SPI transfer has been corrupted.
2. Clear the PB_ERR_IRQ by writing a ‘1’ to the PB_ERR_IRQ bit in the IRQSTS2 register.
3. Restart the SPI transfer to download the Packet Buffer again. The design of the Packet Buffer ensures that read data has not been lost.

8.6.10.5 Registers Related to Packet Buffer Operation

PB_ERR_IRQ - Indicates that a Packet Buffer Underrun Error has occurred. This condition can occur if a SPI burst read access to the Packet Buffer begins before the incoming packet is completely received, and the SPI PB read address overtakes the RX demodulator write address. (See Downloading Packet Buffer Contents During Reception for information on how to avoid this condition, and what to do if it does occur). Software must write a ‘1’ to this bit to clear the interrupt. The PB_ERR_IRQ interrupt can occur only during SPI BURST accesses to Packet Buffer; SPI BYTE mode accesses to Packet Buffer will not trigger a PB_ERR_IRQ even if the SPI address is greater than the RX Demodulator address.

1: Packet Buffer Underrun Error has occurred since the last time this bit was cleared by software

0: Packet Buffer Underrun Error has not occurred since the last time this bit was cleared by software

ACTIVE_PROMISCUOUS - Normally, received packets that are of Frame Type “Acknowledge”, received in response to a packet transmitted by the radio, are not stored in the Packet Buffer. Such packets are expected during a Sequence TR with RXACKRQD=1. In Active Promiscuous mode, software must have access to all receive packet data, even “auto-RXACK” packets. This bit allows such packets to be stored in the Packet Buffer.

1: Allow “auto-RXACK” packets to be stored in the Packet Buffer

0: Don’t allow “auto-RXACK” packets to be stored in the Packet Buffer (default)

8.7 Configuring MCU for Proper SPI Operation

The MCU port control is configured as follows to operate with the modem SPI:

- PTB10 is configured as SPI1_PCS0
- PTB11 is configured as SPI1_SCK

- PTB16 is configured as SPI1_SOUT
- PTB17 is configured as SPI1_SIN. Also, the default configuration of the modem R_MISO is that the modem tri-states this output when the R_SSEL_B is not asserted. The PTB17 pad should be configured with a pull-up or pull-down enabled.

The MCU DSPI module must also be configured for proper operation to meet the modem SPI transaction format. The following conditions must be met:

- MCU is master
- Maximum baud rate is 9 MHz for reads, and 16 MHz for writes
- Proper clock format must be selected, i.e., CPHA=0 and CPOL=0
- SPI data must be transferred MSB first
- The following modem SPI timings parameters must not be violated:
    - t_{CSC} (R_SSEL_B assertion to SCK delay) must be no less than 31.25ns
    - t_{ASC} (SCK to R_SSEL_B deassertion delay) must be no less than 31.25ns
    - t_{DT} (R_SSEL_B deasserted idle time) must be no less than 62.5ns

The following sections describe the DSPI configuration in more detail. Refer to the MCU: Serial Peripheral Interface (SPI) chapter.

8.7.1 DSPI Mode Configuration

The following DSPI1 register settings are required to communicate correctly with the modem SPI:

- MSTR (SPI1_MCR, bit 31). MSTR =1 to configure the DSPI for master mode
- PCSIS[0] (SPI1_MCR, bit 16). PCSIS[0]=1 must be set to indicate that the inactive state of PCS0 is high
- CPOL (SPI1_CTARn, bit 26) - CPOL=0 to select the correct clock polarity (active high / inactive low)
- CPHA (SPI1_CTARn, bit 25) - CPHA=0 to select the correct clock phase (data captured on rising edge, changed on falling edge)
- LSBFE (SPI1_CTARn, bit 24) - LSBFE=0 to transfer data MSB first

8.7.2 DSPI Baud Rate

The DSPI baud rate needs to be configured so that it is no more than 9MHz for modem SPI reads and 16MHz for modem SPI writes. The DSPI baud rate is determined by the MCU bus clock frequency, as well as by several programmable register bits in the DSPI.

- PBR (SPI1_CTARn, bits 17:16) - selects the baud rate prescaler value

- BR (SPI1_CTARn, bits 3:0) - select the baud rate scaler
- DBR (SPI1_CTARn, bit 31) - selects a double baud rate mode

As an example, assuming a 50MHz bus clock, PBR could be programmed for a divide-by-3 value, BR could be programmed for a divide-by-2 value, and DBR could be programmed to 0 to achieve an 8.3MHz SPI clock rate. For SPI writes, the PBR could be changed to a divide-by-2 value to realize a 12.5MHz SPI clock rate.

If a 32MHz bus clock is used, PBR could be programmed for a divide-by-2 value, BR could be programmed for a divide-by-2 value, and DBR could be programmed to 0 to achieve an 8MHz SPI clock rate. For SPI writes, the DBR could be changed to 1 to realize a 16MHz SPI clock rate.

Refer to the MCU: Serial Peripheral Interface (SPI) chapter for more information on baud rate.

8.7.3 DSPI Timing Control

The DSPI provides the following programmable control relating to the SPI timing parameters t_{CSC}, t_{ASC}, and t_{DT}:

- PCSSCK (SPI1_CTARn, bits 23:22) and CSSCK (SPI1_CTARn, bits 15:12) - these control the delay from PCS assertion to the first SCK edge, that is, t_{CSC}
- PASC (SPI1_CTARn, bits 21:20) and ASC (SPI1_CTARn, bits 11:8) - these control the delay from the last SCK edge to the negation of PCS, that is, t_{ASC}
- PDT (SPI1_CTARn, bits 19:18) and DT (SPI1_CTARn, bits 7:4) - these control the delay from the negation of PCS to the assertion of PCS for the next SPI frame, that is, t_{DT}

Since the maximum bus clock in the MCU is 50MHz, t_{CSC} can be no more than 40ns regardless of programming. So PCSSCK and CSSCK can be programmed to their default values (device-by-1 and divide-by-2 respectively) to minimize this delay and still meet the requirement that t_{CSC} be no less than 32.5ns.

Likewise, t_{ASC} can be no more than 40ns regardless of programming since the MCU bus clock 50MHz limit. So PASC and ASC can also be programmed to their default values (device-by-1 and divide-by-2 respectively) to minimize this delay and still meet the requirement that t_{ASC} be no less than 32.5ns.

However, depending on the bus frequency, the PDT and DT may need to be programmed to ensure that t_{DT} is no less than 62.5ns. If the bus clock is 32MHz or less, the default programming for PDT and DT can be used, since then t_{DT}will be no more than 62.5ns. However, with a 50MHz bus clock, the default programming for PDT and DT would cause t_{DT} to be 40ns, which does not meet the modem requirement. Programming DT to a divide-by-4 value (instead of default divide-by-2) when the bus clock is 50MHz would increase t_{DT} to 80ns to meet the modem timing requirement.

Refer to the MCU: Serial Peripheral Interface (SPI) chapter for more information on baud rate and clock delay generation

Chapter 9
Modem: SPI Register Descriptions

9.1 Introduction

This section provides a complete listing of all the modem registers, including register address, access modes, descriptions, and references to other documents for more complete information.

The transceiver registers allow the MCU to communicate and control the radio as well as receive feedback and status information. All radio registers are accessible via a SPI interface only. No other means of accessing the transceiver registers is possible. The registers are divided into two categories; direct and indirect. The first 64 registers are “Direct Access”, meaning only a single SPI control word is required to identify the register to be accessed. The remaining registers are “Indirect Access” and use a pointer register followed by a data register as discussed following.

Indirect access registers use a pointer register and a window register for access in the indirect space. The pointer register (IAR_INDEX) and the data register (IAR_DATA) occupy the last two registers slots in Direct Space; addresses 0x3E and 0x3F respectively. Both direct- and indirect-access registers can be accessed in SPI burst mode, in which the SPI control word addresses the first register in the burst. After the first register in the burst is accessed, the hardware will auto-increment the register address so that subsequent sequential registers can be accessed during a single multi-byte SPI transaction. The length of the burst is not limited by hardware.

Direct access burst mode is available in all power states, including HIBERNATE. Indirect access burst mode is restricted to states during which the crystal oscillator is running (e.g., DOZE, IDLE, RUN); this excludes HIBERNATE state. Indirect register access in HIBERNATE state is restricted to single-byte.

A few registers, which control special functions within the IC, should not be accessed in HIBERNATE state; these are identified in the “Register Description” tables later in this document. However, most registers are 100% accessible in all power states, including HIBERNATE.

In the following Register Summary table, registers are listed in ascending address order, starting with direct address space (direct addresses 0x00 - 0x3F), followed by indirect address space (addresses 0x00 - 0xFF). For each register, address (direct or indirect) appears in the left-hand column. The register bits are then listed horizontally from left to right, bit 7 ... bit 0. The right-hand column contains the register mnemonic. For each bit, an access mode is shown (r = read-only; rw = read/write; w1tc=write-1-to-clear; sc=self-clearing; etc.). And for all read/write registers, a default value is provided. Unpopoulated registers, or unpopulated register bits within a partially-populated register, are indicated by empty cells in the table, and should be considered reserved; these bit positions read back 0, and only 0 should be written to them.

9.2 Modem Memory map and register definition

NOTE

The Direct and Indirect registers can be accessed only through the SPI interface. Direct access by a single SPI control word and Indirect using the pointer register (IAR_INDEX) and the data register (IAR_DATA), which occupy the last two registers in direct access; addresses 0x3E and 0x3F respectively.

Modem memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
0	Interrupt Request Status 1 (Modem_IRQSTS1)	8	w1c	00h	9.2.1/202
1	Interrupt Request Status 2 (Modem_IRQSTS2)	8	R	01h	9.2.2/204
2	Interrupt Request Status 3 (Modem_IRQSTS3)	8	R/W	F0h	9.2.3/205
3	PHY Control 1 (Modem_PHY_CTRL1)	8	R/W	00h	9.2.4/207
4	PHY Control 2 (Modem_PHY_CTRL2)	8	R/W	FFh	9.2.5/208
5	PHY Control 3 (Modem_PHY_CTRL3)	8	R/W	06h	9.2.6/209
6	Receive Frame Length (Modem_RX_FRM_LEN)	8	R	00h	9.2.7/210
7	PHY Control 4 (Modem_PHY_CTRL4)	8	R/W	08h	9.2.8/211
8	SRC Control (Modem_SRC_CTRL)	8	R/W	0Ch	9.2.9/212
9	SRC Address SUM LSB (Modem_SRC_ADDRS_SUM_LSB)	8	R/W	00h	9.2.10/213
A	SRC Address SUM MSB (Modem_SRC_ADDRS_SUM_MSB)	8	R/W	00h	9.2.11/213
B	CCA1 ED FNL (Modem_CCA1_ED_FNL)	8	R	00h	9.2.12/214
C	Event Timer LSB (Modem_EVENT_TIMER_LSB)	8	R	00h	9.2.13/214
D	Event Timer MSB (Modem_EVENT_TIMER_MSB)	8	R	00h	9.2.14/215
E	Event Timer USB (Modem_EVENT_TIMER_USB)	8	R	00h	9.2.15/215

Table continues on the next page...

Modem memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
F	Timestamp LSB (Modem_TIMESTAMP_LSB)	8	R	00h	9.2.16/215
10	Timestamp MSB (Modem_TIMESTAMP_MSB)	8	R	00h	9.2.17/216
11	Timestamp USB (Modem_TIMESTAMP_USB)	8	R	00h	9.2.18/216
12	Timer 3 Compare Value LSB (Modem_T3CMP_LSB)	8	R/W	FFh	9.2.19/217
13	Timer 3 Compare Value MSB (Modem_T3CMP_MSB)	8	R/W	FFh	9.2.20/217
14	Timer 3 Compare Value USB (Modem_T3CMP_USB)	8	R/W	FFh	9.2.21/217
15	Timer 2-Prime Compare Value LSB (Modem_T2PRIMECMP_LSB)	8	R/W	FFh	9.2.22/218
16	Timer 2-Prime Compare Value MSB (Modem_T2PRIMECMP_MSB)	8	R/W	FFh	9.2.23/218
17	Timer 1 Compare Value LSB (Modem_T1CMP_LSB)	8	R/W	FFh	9.2.24/218
18	Timer 1 Compare Value MSB (Modem_T1CMP_MSB)	8	R/W	FFh	9.2.25/219
19	Timer 1 Compare Value USB (Modem_T1CMP_USB)	8	R/W	FFh	9.2.26/219
1A	Timer 2 Compare Value LSB (Modem_T2CMP_LSB)	8	R/W	FFh	9.2.27/219
1B	Timer 2 Compare Value MSB (Modem_T2CMP_MSB)	8	R/W	FFh	9.2.28/220
1C	Timer 2 Compare Value USB (Modem_T2CMP_USB)	8	R/W	FFh	9.2.29/220
1D	Timer 4 Compare Value LSB (Modem_T4CMP_LSB)	8	R/W	FFh	9.2.30/220
1E	Timer 4 Compare Value MSB (Modem_T4CMP_MSB)	8	R/W	FFh	9.2.31/221
1F	Timer 4 Compare Value USB (Modem_T4CMP_USB)	8	R/W	FFh	9.2.32/221
20	PLL Integer Value for PAN0 (Modem_PLL_INT0)	8	R/W	0Ch	9.2.33/221
21	PLL Frequency Fractional Value for PAN0 (Modem_PLL_FRAC0_LSB)	8	R/W	00h	9.2.34/222
22	PLL Frequency Fractional Value for PAN0 (Modem_PLL_FRAC0_MSB)	8	R/W	90h	9.2.35/222
23	PA Power Control (Modem_PA_PWR) (Modem_PA_PWR)	8	R/W	18h	9.2.36/223
24	Sequence Manager State (Modem_SEQ_STATE)	8	R	00h	9.2.37/223
25	Link Quality Indicator (Modem_LQI_VALUE)	8	R	00h	9.2.38/223
26	RSSI CCA CNT (Modem_RSSI_CCA_CNT)	8	R	00h	9.2.39/224
28	ASM Control 1 (Modem_ASM_CTRL1)	8	R/W	00h	9.2.40/224
29	ASM Control 2 (Modem_ASM_CTRL2)	8	R/W	00h	9.2.41/225
2A	ASM Data (Modem_ASM_DATA_0)	8	R/W	00h	9.2.42/226
2B	ASM Data (Modem_ASM_DATA_1)	8	R/W	00h	9.2.42/226
2C	ASM Data (Modem_ASM_DATA_2)	8	R/W	00h	9.2.42/226
2D	ASM Data (Modem_ASM_DATA_3)	8	R/W	00h	9.2.42/226
2E	ASM Data (Modem_ASM_DATA_4)	8	R/W	00h	9.2.42/226
2F	ASM Data (Modem_ASM_DATA_5)	8	R/W	00h	9.2.42/226
30	ASM Data (Modem_ASM_DATA_6)	8	R/W	00h	9.2.42/226
31	ASM Data (Modem_ASM_DATA_7)	8	R/W	00h	9.2.42/226
32	ASM Data (Modem_ASM_DATA_8)	8	R/W	00h	9.2.42/226

Table continues on the next page...

Modem memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
33	ASM Data (Modem_ASM_DATA_9)	8	R/W	00h	9.2.42/226
34	ASM Data (Modem_ASM_DATA_A)	8	R/W	00h	9.2.42/226
35	ASM Data (Modem_ASM_DATA_B)	8	R/W	00h	9.2.42/226
36	ASM Data (Modem_ASM_DATA_C)	8	R/W	00h	9.2.42/226
37	ASM Data (Modem_ASM_DATA_D)	8	R/W	00h	9.2.42/226
38	ASM Data (Modem_ASM_DATA_E)	8	R/W	00h	9.2.42/226
39	ASM Data (Modem_ASM_DATA_F)	8	R/W	00h	9.2.42/226
3B	Overwrite Version Number (Modem_OVERWRITE_VER)	8	R/W	00h	9.2.43/226
3C	CLK_OUT Control (Modem_CLK_OUT_CTRL)	8	R/W	88h	9.2.44/227
3D	Power Modes (Modem_PWR_MODES)	8	R/W	11h	9.2.45/228
3E	IAR Index (Modem_IAR_INDEX)	8	W	00h	9.2.46/229
3F	IAR Data (Modem_IAR_DATA)	8	R/W	00h	9.2.47/230

9.2.1 Interrupt Request Status 1 (Modem_IRQSTS1)

This register provides IRQ status information.

Address: 0h base + 0h offset = 0h

Bit	7	6	5	4
Read	RX_FRM_PEND	PLL_UNLOCK_IRQ	FILTERFAIL_IRQ	RXWTRMRKIRQ
Write	▨	w1c	w1c	w1c
Reset	0	0	0	0

Bit	3	2	1	0
Read	CCAIRQ	RXIRQ	TXIRQ	SEQIRQ
Write	w1c	w1c	w1c	w1c
Reset	0	0	0	0

Modem_IRQSTS1 field descriptions

Field	Description
7 RX_FRM_PEND	Frame Pending Interrupt Status Shows the status of the frame pending bit of the frame control field for the most-recently received packet. This is a read-only bit. 0 Indicates that an unlock event has not occurred in the PLL. 1 Indicates that an unlock event has not occurred in the PLL.
6 PLL_UNLOCK_IRQ	PLL Unlock Interrupt Status Shows whether an unlock event has occurred in the PLL. This is write a 1 to clear bit.

Table continues on the next page...

Modem_IRQSTS1 field descriptions (continued)

Field	Description
	0 Indicates that an unlock event has not occurred in the PLL. 1 Indicates that an unlock event has not occurred in the PLL.
5 FILTERFAIL_IRQ	Receiver Packet Filter Fail Interrupt Status Shows whether the most-recently received packet has been rejected due to elements within the packet. This is write a 1 to clear bit. • If PAN0 and PAN1 occupy different channels and CURRENT_NETWORK=0, FILTERFAIL_IRQ applies to PAN0. • If PAN0 and PAN1 occupy different channels and CURRENT_NETWORK=1, FILTERFAIL_IRQ applies to PAN1. • If PAN0 and PAN1 occupy the same channel, FILTERFAIL_IRQ is the logical 'AND' of the individual PANs' Filter Fail status.
4 RXWTRMRKIRQ	Receiver Byte Count Water Mark Interrupt Status Shows whether the number of bytes specified in the RX_WTR_MARK register has been reached. This is write a 1 to clear bit. 0 Indicates that the number of bytes specified in the RX_WTR_MARK register has not been reached 1 Indicates that the number of bytes specified in the RX_WTR_MARK register has been reached
3 CCAIRQ	Clear Channel Assessment Interrupt Status Shows the status of a CCA operation. This is write a 1 to clear bit. See the sequence manager section for a description of the timing of this interrupt. 0 Indicates that the completion of a CCA operation has not occurred. 1 Indicates that the completion of a CCA operation has occurred.
2 RXIRQ	Receiver Interrupt Status Shows the status of a receive operation. This is write a 1 to clear bit. 0 Indicates that the completion of a receive operation has not occurred. 1 Indicates that the completion of a receive operation has occurred.
1 TXIRQ	Transmitter Interrupt Status Shows the status of a transmit operation. This is write a 1 to clear bit. 0 Indicates that the completion of a transmit operation has not occurred. 1 Indicates that the completion of a transmit operation has occurred.
0 SEQIRQ	Sequence-end Interrupt Request Status This interrupt will assert whenever the Sequence Manager transitions from non-idle to idle state, for any reason. This is write a 1 to clear bit. 0 Indicates that the completion of an autosequence has not occurred. 1 Indicates that the completion of an autosequence has occurred.

9.2.2 Interrupt Request Status 2 (Modem_IRQSTS2)

This register provides IRQ status information.

Address: 0h base + 1h offset = 1h

Bit	7	6	5	4
Read	CRCVALID	CCA	SRCADDR	PI
Write				
Reset	0	0	0	0

Bit	3	2	1	0
Read	TMRSTATUS/WAKE_FROM_HIB	ASM_IRQ	PB_ERR_IRQ	WAKE_IRQ
Write		w1c	w1c	w1c
Reset	0	0	0	1

Modem_IRQSTS2 field descriptions

Field	Description
7 CRCVALID	Code Redundancy Check Valid Indicates the compare result between the FCS field, in the most-recently received frame, and the internally calculated CRC value. This flag is cleared at the next receiver warmup. 0 Rx FCS != calculated CRC (incorrect) 1 Rx FCS = calculated CRC (correct)
6 CCA	Channel Idle/Busy This indicator is valid at CCAIRQ and also at SEQIRQ. This flag is cleared at next receiver warmup. 0 Channel is idle. 1 Channel is busy.
5 SRCADDR	Source Address Match Shows whether all three of these conditions are true: The incoming packet was a data request. Source address matching is enabled, SCRADDR_EN=1. Source address checksum computed by packet processor matched at least one entry in the source address table. 0 At least one of these conditions is not true. 1 All three of these conditions are true.
4 PI	Poll Indication Indicates whether the received packet was a data request.

Table continues on the next page...

Modem_IRQSTS2 field descriptions (continued)

Field	Description
	0 Indicates that the received packet was not a data request. 1 Indicates that the received packet was a data request, regardless of whether a source address table match occurred, or whether source address matching is enabled. (See SRCADDR_EN in direct register 0x08, MODEM_SRC_CTRL.)
3 TMRSTATUS/WAKE_ FROM_HIB	Timer Status/Wake from Hibernate This is a dual-purpose, read-only status bit. Its value is dependent on the WAKE_IRQ bit. Used to indicate either wakeup status or timer status. 0 If WAKE_IRQ=1, a 0 indicates that a wakeup from POR has occurred. If WAKE_IRQ=0, a 0 indicates that no TMRxIRQ is asserted. See the IRQST3 register for the individual timer interrupt status bits. 1 If WAKE_IRQ=1, a 1 indicates that a wakeup from Hibernate has occurred. If WAKE_IRQ=0, a 1 indicates that at least one of the TMRxIRQ is asserted. See the IRQST3 register for the individual timer interrupt status bits.
2 ASM_IRQ	AES Interrupt Flag A 1 indicates the completion of a AES operation completed. This is write a 1 to clear bit.
1 PB_ERR_IRQ	Packet Buffer Underrun Error Interrupt Status Shows whether a packet buffer underrun error has occurred since the last time this bit was cleared by software. This is a write 1 to clear bit. PB_ERR_IRQ will occur only when the SPI packet buffer access is Burst mode (not Byte mode). 0 A packet buffer underrun error has not occurred since the last time this bit was cleared by software. 1 A packet buffer underrun error has occurred since the last time this bit was cleared by software.
0 WAKE_IRQ	Wake Interrupt Status Indicates that either a wake-from-POR or a wake-from-Hibernate event has occured. This is write a 1 to clear bit. 0 Indicates that neither wake event has occurred. 1 Indicates that one of these wake events has occurred. See the WAKE_FROM_HIB/ TMRSTATUS bit to see which type of wake has occurred.

9.2.3 Interrupt Request Status 3 (Modem_IRQSTS3)

This register provides IRQ status information.

Address: 0h base + 2h offset = 2h

Bit	7	6	5	4	3	2	1	0
Read	TMR4MSK	TMR3MSK	TMR2MSK	TMR1MSK	TMR4IRQ	TMR3IRQ	TMR2IRQ	TMR1IRQ
Write					w1c	w1c	w1c	w1c
Reset	1	1	1	1	0	0	0	0

Modem_IRQSTS3 field descriptions

Field	Description
7 TMR4MSK	Timer Comparator 4 Interrupt Mask 0 Allows interrupt when comparator matches event timer count. 1 Disables interrupt generation, but allows a TMR4IRQ flag to be set.
6 TMR3MSK	Timer Comparator 3 Interrupt Mask 0 Allows interrupt when comparator matches event timer count. 1 Disables interrupt generation, but allows a TMR3IRQ flag to be set.
5 TMR2MSK	Timer Comparator 2 Interrupt Mask 0 Allows interrupt when comparator matches event timer count. 1 Disables interrupt generation, but allows a TMR2IRQ flag to be set.
4 TMR1MSK	Timer Comparator 1 Interrupt Mask 0 Allows interrupt when comparator matches event timer count. 1 Disables interrupt generation, but allows a TMR1IRQ flag to be set.
3 TMR4IRQ	Timer Comparator 4 Interrupt Status Indicates whether the T4CMP comparator value matched the event timer counter. This is a write 1 to clear bit. 0 Indicates that the T4CMP comparator value does not match the event timer counter. 1 Indicates that the T4CMP comparator value does match the event timer counter.
2 TMR3IRQ	Timer Comparator 3 Interrupt Status Indicates whether the T1CMP comparator value matched the event timer counter. This is a write 1 to clear bit. 0 Indicates that the T3CMP comparator value does not match the event timer counter. 1 Indicates that the T3CMP comparator value does match the event timer counter.
1 TMR2IRQ	Timer Comparator 2 Interrupt Status Indicates whether the T2CMP comparator value matched the event timer counter. This is a write 1 to clear bit. 0 Indicates that the T2CMP comparator value does not match the event timer counter. 1 Indicates that the T2CMP comparator value does match the event timer counter.
0 TMR1IRQ	Timer Comparator 1 Interrupt Status Indicates whether the T1CMP comparator value matched the event timer counter. This is a write 1 to clear bit. 0 Indicates that the T1CMP comparator value does not match the event timer counter. 1 Indicates that the T1CMP comparator value does match the event timer counter.

9.2.4 PHY Control 1 (Modem_PHY_CTRL1)

This register is used to control the PHY.

Address: 0h base + 3h offset = 3h

Bit	7	6	5	4	3	2	1	0
Read Write	TMRTRIGEN	SLOTTED	CCABFRTX	RXACKRQD	AUTOACK	XCVSEQ		
Reset	0	0	0	0	0	0	0	0

Modem_PHY_CTRL1 field descriptions

Field	Description
7 TMRTRIGEN	Timer 2 Trigger Enable This control bit allows the transceivers programmed sequence to be immediate or delayed by T2CMP or T2PRIMECMP count. 0 Programmed sequence initiates immediately upon write to XCVSEQ. 1 Allow timer TC2 (or TC2') to initiate a preprogrammed sequence (see XCVSEQ register).
6 SLOTTED	Slotted Mode For beacon-enabled networks. Applies only to Sequences T, TR, and R, ignored during all other sequences. Used, in concert with CCABFRTX, to determine how many CCA measurements are required prior to a transmit operation. Also used during R sequence to determine whether the ensuing transmit acknowledge frame (if any) needs to be synchronized to a backoff slot boundary. 0 Non-slotted 1 Slotted
5 CCABFRTX	CCA Before TX Applies only to sequences T and TR. Ignored during all other sequences. 0 No CCA required. Transmit operation begins immediately. 1 At least one CCA measurement is required prior to the transmit operation (see also SLOTTED).
4 RXACKRQD	Receive Acknowledge Frame Required Applies only to Sequence TR, ignored during all other sequences. 0 An ordinary receive frame (any type of frame) follows the transmit frame. 1 A receive acknowledge frame is expected to follow the transmit frame (non-acknowledge frames are rejected).
3 AUTOACK	Auto Acklowledge Enable Applies only to sequence R and sequence TR. Ignored during other sequences. 0 The sequence manager will not follow a receive frame with a Tx Ack frame, under any condition. The autosequence will terminate after the receive frame. 1 The sequence manager will follow a receive frame with an automatic hardware-generated Tx Ack frame, assuming other necessary conditions are met.
XCVSEQ	Transceiver Sequence Selector

Table continues on the next page...

Modem_PHY_CTRL1 field descriptions (continued)

Field	Description
	Selects an autosequence for the sequence manager to execute. Sequence initiation can be immediate, or scheduled (see TMRTRIGEN). A write of XCVSEQ=IDLE will abort any ongoing sequence. A write of XCVSEQ=IDLE must always be performed after a sequence is complete, and before a new sequence is programmed. Any write to XCVSEQ other than XCVSEQ=IDLE during an ongoing sequence, shall be ignored. The mapping of XCVSEQ to sequence types is as follows: 000 Idle (sequence I) 001 Receive (sequence R) 010 Transmit (sequence T) 011 CCA (sequence C) 100 Transmit/Receive (sequence TR) 101 Continuous CCA (sequence CCCA) 110 Reserved 111 Reserved

9.2.5 PHY Control 2 (Modem_PHY_CTRL2)

This register is used to control the PHY.

Address: 0h base + 4h offset = 4h

Bit	7	6	5	4	3	2	1	0
Read Write	CRC_MSK	PLL_ UNLOCK_ MSK	FILTERFAIL _MSK	RX_WMRK_ MSK	CCAMSK	RXMSK	TXMSK	SEQMSK
Reset	1	1	1	1	1	1	1	1

Modem_PHY_CTRL2 field descriptions

Field	Description
7 CRC_MSK	CRC Mask 0 The sequence manager ignores CRCVALID and considers the receive operation complete after the last octet of the frame has been received. 1 The sequence manager requires CRCVALID=1 at the end of the received frame in order for the receive operation to complete successfully; if CRCVALID=0, sequence manager will return to preamble-detect mode after the last octet of the frame has been received.
6 PLL_UNLOCK_ MSK	PLL Unlock Mask 0 Allows PLL unlock event to generate an interrupt on IRQ_B. 1 A PLL unlock event will set the PLL_UNLOCK_IRQ status bit, but an interrupt is not generated on IRQ_B.
5 FILTERFAIL_ MSK	Filter Fail Mask 0 Allows packet processor filtering Failure to generate an interrupt on IRQ_B. 1 A packet processor filtering failure will set the FILTERFAIL_IRQ status bit, but an interrupt is not generated on IRQ_B.

Table continues on the next page...

Modem_PHY_CTRL2 field descriptions (continued)

Field	Description
4 RX_WMRK_MSK	Recieve Watermark Mask 0 Allows a Received Byte Count match to the RX_WTR_MARK threshold register to generate an interrupt on IRQ_B. 1 A Received Byte Count match to the RX_WTR_MARK threshold register will set the RXWTRMRKIRQ status bit, but an interrupt is not generated on IRQ_B.
3 CCAMSK	Clear Channel Assessment Mask 0 Allows completion of a CCA operation to generate an interrupt on IRQ_B. 1 Completion of a CCA operation will set the CCAIRQ status bit, but an interrupt is not generated on IRQ_B.
2 RXMSK	Receive Mask 0 Allows completion of a RX operation to generate an interrupt on IRQ_B. 1 Completion of a RX operation will set the RXIRQ status bit, but an interrupt is not generated on IRQ_B.
1 TXMSK	Transmit Mask 0 Allows completion of a TX operation to generate an interrupt on IRQ_B. 1 Completion of a TX operation will set the TXIRQ status bit, but an interrupt is not generated on IRQ_B.
0 SEQMSK	Sequence Mask 0 Allows completion of an autosequence to generate an interrupt on IRQ_B. 1 Completion of an autosequence will set the SEQIRQ status bit, but an interrupt is not generated on IRQ_B.

9.2.6 PHY Control 3 (Modem_PHY_CTRL3)

This register is used to control the PHY.

Address: 0h base + 5h offset = 5h

Bit	7	6	5	4
Read Write	TMR4CMP_EN	TMR3CMP_EN	TMR2CMP_EN	TMR1CMP_EN
Reset	0	0	0	0

Bit	3	2	1	0
Read Write	Reserved	ASM_MSK	PB_ERR_MSK	WAKE_MSK
Reset	0	1	1	0

Modem_PHY_CTRL3 field descriptions

Field	Description
7 TMR4CMP_EN	Timer 4 Comparator Enable 0 Don’t allow an event timer match to T4CMP to set TMR4IRQ. 1 Allow an event timer match to T4CMP to set TMR4IRQ.

Table continues on the next page...

Modem_PHY_CTRL3 field descriptions (continued)

Field	Description
6 TMR3CMP_EN	Timer 3 Comparator Enable 0 Don’t allow an event timer match to T3CMP to set TMR3IRQ. 1 Allow an event timer match to T3CMP to set TMR3IRQ.
5 TMR2CMP_EN	Timer 2 Comparator Enable 0 Don’t allow an event timer match to T2CMP to set TMR2IRQ. 1 Allow an event timer match to T2CMP to set TMR2IRQ.
4 TMR1CMP_EN	Timer 1 Comparator Enable 0 Don’t allow an event timer match to T1CMP to set TMR1IRQ. 1 Allow an event timer match to T1CMP to set TMR1IRQ.
3 Reserved	Reserved. This field is reserved.
2 ASM_MSK	ASM Mask 0 Enable AES Interrupt to assert IRQ_B. 1 Mask AES Interrupt from asserting IRQ_B.
1 PB_ERR_MSK	Packet Buffer Error Mask 0 Enable packet buffer error interrupt to assert IRQ_B. 1 Mask Packet Buffer Error Interrupt from asserting IRQ_B.
0 WAKE_MSK	Wake Mask 0 Enable Wake Interrupt to assert IRQ_B 1 Mask Wake Interrupt from asserting IRQ_B.

9.2.7 Receive Frame Length (Modem_RX_FRM_LEN)

Contents of the PHR (PHY header), or FrameLength field, of the most recently received packet.

Address: 0h base + 6h offset = 6h

Bit	7	6	5	4	3	2	1	0
Read	Reserved	RX_FRAME_LENGTH						
Write								
Reset	0	0	0	0	0	0	0	0

Modem_RX_FRM_LEN field descriptions

Field	Description
7 Reserved	Reserved. This field is reserved.

Table continues on the next page...

Modem_RX_FRM_LEN field descriptions (continued)

Field	Description
RX_FRAME_ LENGTH	Receive Frame Length This read-only field contains the contents of the PHR (PHY header), or FrameLength field, of the most recently received packet.

9.2.8 PHY Control 4 (Modem_PHY_CTRL4)

This register is used to control the PHY.

Address: 0h base + 7h offset = 7h

Bit	7	6	5	4
Read Write	TRCV_MSK	TC3TMOUT	PANCORDNTR0	CCATYPE
Reset	0	0	0	0

Bit	3	2	1	0
Read	CCATYPE		PROMISCUOUS	TC2PRIME_EN
Write	CCATYPE	TMRLOAD	PROMISCUOUS	TC2PRIME_EN
Reset	1	0	0	0

Modem_PHY_CTRL4 field descriptions

Field	Description
7 TRCV_MSK	Transceiver IRQ_B All interrupt Mask 0 Enable any interrupt to assert IRQ_B. 1 Mask all interrupts from asserting IRQ_B.
6 TC3TMOUT	Timer 3 Function Control 0 TMR3 is a software timer only. 1 Enable TMR3 to abort Rx or CCCA operations.
5 PANCORDNTR0	PAN Coordinator on PAN0 Device Enable Device is a PAN Coordinator on PAN0. Allows device to receive packets with no destination address, if source PAN ID matches.
4–3 CCATYPE	Clear Channel Assessment Type Selects one of four possible functions for CCA or ED, listed below. 00 Energy detect 01 CCA mode 1 10 CCA mode 2 11 CCA mode 3
2 TMRLOAD	Timer Load

Table continues on the next page...

Modem_PHY_CTRL4 field descriptions (continued)

Field	Description
	A low to high transition of this bit causes the contents of register ‘T1CMP[23:0]’ to be loaded into the Event Timer.This is a self clearing bit and always reads zero.
1 PROMISCUOUS	Promiscuous Bypasses most packet filtering. 0 Normal mode. 1 All packet filtering except frame length checking (FrameLength =5 or greater and FrameLength =127 or less) is bypassed.
0 TC2PRIME_EN	TMR2-prime Compare Match Enable 0 Don’t allow a match of the lower 16 bits of the event timer to T2PRIMECMP to set TMR2IRQ. 1 Allow a match of the lower 16 bits of Event Timer to T2PRIMECMP to set TMR2IRQ.

9.2.9 SRC Control (Modem_SRC_CTRL)

This register is used to control the SRC.

Address: 0h base + 8h offset = 8h

<table>
<tr><td>Bit</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>Read</td><td colspan="4">INDEX</td><td rowspan="2">ACK_FRM_PND</td><td rowspan="2">SRCADDR_EN</td><td>▨</td><td>▨</td></tr>
<tr><td>Write</td><td colspan="4">▨</td><td>INDEX_EN</td><td>INDEX_DISABLE</td></tr>
<tr><td>Reset</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td><td>0</td></tr>
</table>

Modem_SRC_CTRL field descriptions

Field	Description
7–4 INDEX	Index If SRCADDR_EN=0, the value of ACK_FRM_PND is copied into the FramePending subfield of the Frame Control field of an auto-TxAck frame, if conditions are met for such a frame to be sent following a receive frame (see sequence manager section, sequence R and sequence TR). If SRCADDR_EN=1, ACK_FRM_PND is ignored, and the result of the Source Address Matching table-lookup determines the setting of the FramePending subfield of the Frame Control Field.
3 ACK_FRM_PND	Acknowledge Frame Pending If SRCADDR_EN=0, the value of ACK_FRM_PND is copied into the FramePending subfield of the Frame Control field of an auto-TxAck frame, if conditions are met for such a frame to be sent following a receive frame (see sequence manager section, sequence R and sequence TR). If SRCADDR_EN=1, ACK_FRM_PND is ignored, and the result of the Source Address Matching table-lookup determines the setting of the FramePending subfield of the Frame Control Field.
2 SRCADDR_EN	Source Address Matching Enable Part of the Source Address Matching feature. Enables the source address matching feature.
1 INDEX_EN	Source Address Matching Index Enable

Table continues on the next page...

Modem_SRC_CTRL field descriptions (continued)

Field	Description
	Part of the Source Address Matching feature. Setting INDEX_EN activates the selected INDEX in the Source Address Table. Source Address Matching logic will be allowed to compare against the stored checksum associated with this index when an incoming MAC Command data request packet is received. The INDEX_EN bit is write-only, and self-clearing. Writing a 1 to INDEX_EN will enable the selected INDEX. Writing a 0 to this bit has no effect. Reads of INDEX_EN always return zero.
0 INDEX_DISABLE	Source Address Matching Index Disable Part of the source address matching feature. Setting INDEX_DISABLE deactivates (invalidates) the selected INDEX in the Source Address Table. Source Address Matching logic will no longer compare against the stored checksum associated with this index when an incoming MAC Command data request packet is received. The INDEX_DISABLE bit is write-only, and self-clearing. Writing a 1 to INDEX_DISABLE will disable the selected INDEX. Writing a 0 to this bit has no effect. Reads of INDEX_DISABLE always return zero.

9.2.10 SRC Address SUM LSB (Modem_SRC_ADDRS_SUM_LSB)

Address: 0h base + 9h offset = 9h

Bit	7	6	5	4	3	2	1	0
Read Write	SRC_ADDRS_SUM							
Reset	0	0	0	0	0	0	0	0

Modem_SRC_ADDRS_SUM_LSB field descriptions

Field	Description
SRC_ADDRS_SUM	Source Address Checksum LSB Part of the Source Address Matching feature. Software loads a specific index of the Source Address Table with its software-computed checksum by writing to this register, having previously selected the desired index by writing to the INDEX field of the SRC_CTRL Register. This is a 16 bit register. The 2 bytes can be written in any order. Software then activates the selected table index by asserting the INDEX_EN bit of the SRC_CTRL Register. Reads from this register return the 16-bit value from the source address table selected by the INDEX field of the SRC_CTRL register.

9.2.11 SRC Address SUM MSB (Modem_SRC_ADDRS_SUM_MSB)

Address: 0h base + Ah offset = Ah

Bit	7	6	5	4	3	2	1	0
Read Write	SRC_ADDRS_SUM							
Reset	0	0	0	0	0	0	0	0

Modem_SRC_ADDRS_SUM_MSB field descriptions

Field	Description
SRC_ADDRS_ SUM	Source Address Checksum MSB Part of the Source Address Matching feature. Software loads a specific index of the Source Address Table with its software-computed checksum by writing to this register, having previously selected the desired index by writing to the INDEX field of the SRC_CTRL Register. This is a 16 bit register. The 2 bytes can be written in any order. Software then activates the selected table index by asserting the INDEX_EN bit of the SRC_CTRL Register. Reads from this register return the 16-bit value from the source address table selected by the INDEX field of the SRC_CTRL register.

9.2.12 CCA1 ED FNL (Modem_CCA1_ED_FNL)

Address: 0h base + Bh offset = Bh

Bit	7	6	5	4	3	2	1	0
Read	CCA1_ED_FNL							
Write								
Reset	0	0	0	0	0	0	0	0

Modem_CCA1_ED_FNL field descriptions

Field	Description
CCA1_ED_FNL	CCA1/ED Final Average Value Output register to show final averaged RSSI value or compensated value of the same at the end CCA mode1/ED computations.

9.2.13 Event Timer LSB (Modem_EVENT_TIMER_LSB)

Address: 0h base + Ch offset = Ch

Bit	7	6	5	4	3	2	1	0
Read	EVENT_TIMER							
Write								
Reset	0	0	0	0	0	0	0	0

Modem_EVENT_TIMER_LSB field descriptions

Field	Description
EVENT_TIMER	Event Timer LSB Holds the current value of the 24-bit event timer. The hardware latches the upper 2 bytes of EVENT_TMR on each read of least signficant byte.

9.2.14 Event Timer MSB (Modem_EVENT_TIMER_MSB)

Address: 0h base + Dh offset = Dh

Bit	7	6	5	4	3	2	1	0
Read	EVENT_TIMER							
Write								
Reset	0	0	0	0	0	0	0	0

Modem_EVENT_TIMER_MSB field descriptions

Field	Description
EVENT_TIMER	Event Timer MSB Holds the current value of the 24-bit event timer. The hardware latches the upper 2 bytes of EVENT_TMR on each read of least signficant byte.

9.2.15 Event Timer USB (Modem_EVENT_TIMER_USB)

Address: 0h base + Eh offset = Eh

Bit	7	6	5	4	3	2	1	0
Read	EVENT_TIMER							
Write								
Reset	0	0	0	0	0	0	0	0

Modem_EVENT_TIMER_USB field descriptions

Field	Description
EVENT_TIMER	Event Timer USB Holds the current value of the 24-bit event timer. The hardware latches the upper 2 bytes of EVENT_TMR on each read of least signficant byte.

9.2.16 Timestamp LSB (Modem_TIMESTAMP_LSB)

Address: 0h base + Fh offset = Fh

Bit	7	6	5	4	3	2	1	0
Read	TIMESTAMP							
Write								
Reset	0	0	0	0	0	0	0	0

Modem_TIMESTAMP_LSB field descriptions

Field	Description
TIMESTAMP	Timestamp LSB Holds the latched value of the event timer current time corresponding to the beginning of the just received Rx packet, at SFD detect.

9.2.17 Timestamp MSB (Modem_TIMESTAMP_MSB)

Address: 0h base + 10h offset = 10h

Bit	7	6	5	4	3	2	1	0
Read	TIMESTAMP							
Write								
Reset	0	0	0	0	0	0	0	0

Modem_TIMESTAMP_MSB field descriptions

Field	Description
TIMESTAMP	Timestamp MSB Holds the latched value of the event timer current time corresponding to the beginning of the just received Rx packet, at SFD detect.

9.2.18 Timestamp USB (Modem_TIMESTAMP_USB)

Address: 0h base + 11h offset = 11h

Bit	7	6	5	4	3	2	1	0
Read	TIMESTAMP							
Write								
Reset	0	0	0	0	0	0	0	0

Modem_TIMESTAMP_USB field descriptions

Field	Description
TIMESTAMP	Timestamp USB Holds the latched value of the event timer current time corresponding to the beginning of the just received Rx packet, at SFD detect.

9.2.19 Timer 3 Compare Value LSB (Modem_T3CMP_LSB)

Address: 0h base + 12h offset = 12h

Bit	7	6	5	4	3	2	1	0
Read Write	T3CMP							
Reset	1	1	1	1	1	1	1	1

Modem_T3CMP_LSB field descriptions

Field	Description
T3CMP	TMR3 Compare Value If TMR3CMP_EN=1 and the event timer matches this value, TMR3IRQ is set.

9.2.20 Timer 3 Compare Value MSB (Modem_T3CMP_MSB)

Address: 0h base + 13h offset = 13h

Bit	7	6	5	4	3	2	1	0
Read Write	T3CMP							
Reset	1	1	1	1	1	1	1	1

Modem_T3CMP_MSB field descriptions

Field	Description
T3CMP	TMR3 Compare Value If TMR3CMP_EN=1 and the event timer matches this value, TMR3IRQ is set.

9.2.21 Timer 3 Compare Value USB (Modem_T3CMP_USB)

Address: 0h base + 14h offset = 14h

Bit	7	6	5	4	3	2	1	0
Read Write	T3CMP							
Reset	1	1	1	1	1	1	1	1

Modem_T3CMP_USB field descriptions

Field	Description
T3CMP	TMR3 Compare Value If TMR3CMP_EN=1 and the event timer matches this value, TMR3IRQ is set.

9.2.22 Timer 2-Prime Compare Value LSB (Modem_T2PRIMECMP_LSB)

Address: 0h base + 15h offset = 15h

Bit	7	6	5	4	3	2	1	0
Read Write	T2PRIMECMP							
Reset	1	1	1	1	1	1	1	1

Modem_T2PRIMECMP_LSB field descriptions

Field	Description
T2PRIMECMP	TMR2-Prime Compare Value If TMR2CMP_EN=1 and TC2PRIME_EN=1 and the lower 16 bits of the event timer matches this value, TMR2IRQ is set.

9.2.23 Timer 2-Prime Compare Value MSB (Modem_T2PRIMECMP_MSB)

Address: 0h base + 16h offset = 16h

Bit	7	6	5	4	3	2	1	0
Read Write	T2PRIMECMP							
Reset	1	1	1	1	1	1	1	1

Modem_T2PRIMECMP_MSB field descriptions

Field	Description
T2PRIMECMP	TMR2-Prime Compare Value If TMR2CMP_EN=1 and TC2PRIME_EN=1 and the lower 16 bits of the event timer matches this value, TMR2IRQ is set.

9.2.24 Timer 1 Compare Value LSB (Modem_T1CMP_LSB)

Address: 0h base + 17h offset = 17h

Bit	7	6	5	4	3	2	1	0
Read Write	T1CMP							
Reset	1	1	1	1	1	1	1	1

Modem_T1CMP_LSB field descriptions

Field	Description
T1CMP	TMR1 Compare Value If TMR1CMP_EN=1 and the event timer matches this value, TMR1IRQ is set.

9.2.25 Timer 1 Compare Value MSB (Modem_T1CMP_MSB)

Address: 0h base + 18h offset = 18h

Bit	7	6	5	4	3	2	1	0
Read Write	T1CMP							
Reset	1	1	1	1	1	1	1	1

Modem_T1CMP_MSB field descriptions

Field	Description
T1CMP	TMR1 Compare Value If TMR1CMP_EN=1 and the event timer matches this value, TMR1IRQ is set.

9.2.26 Timer 1 Compare Value USB (Modem_T1CMP_USB)

Address: 0h base + 19h offset = 19h

Bit	7	6	5	4	3	2	1	0
Read Write	T1CMP							
Reset	1	1	1	1	1	1	1	1

Modem_T1CMP_USB field descriptions

Field	Description
T1CMP	TMR1 Compare Value If TMR1CMP_EN=1 and the event timer matches this value, TMR1IRQ is set.

9.2.27 Timer 2 Compare Value LSB (Modem_T2CMP_LSB)

Address: 0h base + 1Ah offset = 1Ah

Bit	7	6	5	4	3	2	1	0
Read Write	T2CMP							
Reset	1	1	1	1	1	1	1	1

Modem_T2CMP_LSB field descriptions

Field	Description
T2CMP	TMR2 Compare Value If TMR2CMP_EN=1 and the event timer matches this value, TMR2IRQ is set.

9.2.28 Timer 2 Compare Value MSB (Modem_T2CMP_MSB)

Address: 0h base + 1Bh offset = 1Bh

Bit	7	6	5	4	3	2	1	0
Read Write	T2CMP							
Reset	1	1	1	1	1	1	1	1

Modem_T2CMP_MSB field descriptions

Field	Description
T2CMP	TMR2 Compare Value If TMR2CMP_EN=1 and the event timer matches this value, TMR2IRQ is set.

9.2.29 Timer 2 Compare Value USB (Modem_T2CMP_USB)

Address: 0h base + 1Ch offset = 1Ch

Bit	7	6	5	4	3	2	1	0
Read Write	T2CMP							
Reset	1	1	1	1	1	1	1	1

Modem_T2CMP_USB field descriptions

Field	Description
T2CMP	TMR2 Compare Value If TMR2CMP_EN=1 and the event timer matches this value, TMR2IRQ is set.

9.2.30 Timer 4 Compare Value LSB (Modem_T4CMP_LSB)

Address: 0h base + 1Dh offset = 1Dh

Bit	7	6	5	4	3	2	1	0
Read Write	T4CMP							
Reset	1	1	1	1	1	1	1	1

Modem_T4CMP_LSB field descriptions

Field	Description
T4CMP	TMR4 Compare Value If TMR4CMP_EN=1 and the event timer matches this value, TMR4IRQ is set.

9.2.31 Timer 4 Compare Value MSB (Modem_T4CMP_MSB)

Address: 0h base + 1Eh offset = 1Eh

Bit	7	6	5	4	3	2	1	0
Read Write	T4CMP							
Reset	1	1	1	1	1	1	1	1

Modem_T4CMP_MSB field descriptions

Field	Description
T4CMP	TMR4 Compare Value If TMR4CMP_EN=1 and the event timer matches this value, TMR4IRQ is set.

9.2.32 Timer 4 Compare Value USB (Modem_T4CMP_USB)

Address: 0h base + 1Fh offset = 1Fh

Bit	7	6	5	4	3	2	1	0
Read Write	T4CMP							
Reset	1	1	1	1	1	1	1	1

Modem_T4CMP_USB field descriptions

Field	Description
T4CMP	TMR4 Compare Value If TMR4CMP_EN=1 and the event timer matches this value, TMR4IRQ is set.

9.2.33 PLL Integer Value for PAN0 (Modem_PLL_INT0)

Address: 0h base + 20h offset = 20h

Bit	7	6	5	4	3	2	1	0
Read Write	Reserved			PLL_INT0				
Reset	0	0	0	0	1	1	0	0

Modem_PLL_INT0 field descriptions

Field	Description
7–5 Reserved	This field is reserved.
PLL_INT0	PLL Frequency Integer Value for PAN0 The equation for PLL freqency is: F = ((PLL_INT0+64) + (PLL_FRAC0/65536)) * 32 MHz.

9.2.34 PLL Frequency Fractional Value for PAN0 (Modem_PLL_FRAC0_LSB)

Address: 0h base + 21h offset = 21h

Bit	7	6	5	4	3	2	1	0
Read Write	PLL_FRAC0_LSB							
Reset	0	0	0	0	0	0	0	0

Modem_PLL_FRAC0_LSB field descriptions

Field	Description
PLL_FRAC0_ LSB	PLL Frequency Fractional Value for PAN0 The equation for PLL frequency is: F = ((PLL_INT0+64) + (PLL_FRAC0/65536)) * 3 2MHz.

9.2.35 PLL Frequency Fractional Value for PAN0 (Modem_PLL_FRAC0_MSB)

Address: 0h base + 22h offset = 22h

Bit	7	6	5	4	3	2	1	0
Read Write	PLL_FRAC0_MSB							
Reset	1	0	0	1	0	0	0	0

Modem_PLL_FRAC0_MSB field descriptions

Field	Description
PLL_FRAC0_ MSB	PLL Frequency Fractional Value for PAN0 The equation for PLL frequency is: F = ((PLL_INT0+64) + (PLL_FRAC0/65536)) * 32 MHz.

9.2.36 PA Power Control (Modem_PA_PWR) (Modem_PA_PWR)

Address: 0h base + 23h offset = 23h

Bit	7	6	5	4	3	2	1	0
Read Write	Reserved			PA_PWR				
Reset	0	0	0	1	1	0	0	0

Modem_PA_PWR field descriptions

Field	Description
7–5 Reserved	This field is reserved.
PA_PWR	PA Power Control Adjusts the PA's output power. Linear range is 3 to 31 decimal.

9.2.37 Sequence Manager State (Modem_SEQ_STATE)

Address: 0h base + 24h offset = 24h

Bit	7	6	5	4	3	2	1	0
Read	SEQ_STATE							
Write								
Reset	0	0	0	0	0	0	0	0

Modem_SEQ_STATE field descriptions

Field	Description
SEQ_STATE	Sequence Manager State This read-only register reflects the status of the sequence manager, RX manager, TX manager, and PLL manager state machines. The SEQ_STATE_CTRL[1:0] field of the SEQ_MGR_CTRL register, determines which sequence manager state is monitored by this register. See the sequence manager section for the mapping.

9.2.38 Link Quality Indicator (Modem_LQI_VALUE)

Address: 0h base + 25h offset = 25h

Bit	7	6	5	4	3	2	1	0
Read	LQI_VALUE							
Write								
Reset	0	0	0	0	0	0	0	0

Modem_LQI_VALUE field descriptions

Field	Description
LQI_VALUE	Link Quality Indicator This is the link quality indicator for the most recently received packet. (LQI is also available in the packet buffer, at the end of the received packet data; see packet buffer section for details). See the CCA section for details on the LQI algorithm.

9.2.39 RSSI CCA CNT (Modem_RSSI_CCA_CNT)

Address: 0h base + 26h offset = 26h

Bit	7	6	5	4	3	2	1	0
Read	RSSI_CCA_CNT							
Write								
Reset	0	0	0	0	0	0	0	0

Modem_RSSI_CCA_CNT field descriptions

Field	Description
RSSI_CCA_CNT	RSSI/CCA Continuous Value Continuous update of average RSSI value, available throughout packet reception.

9.2.40 ASM Control 1 (Modem_ASM_CTRL1)

Address: 0h base + 28h offset = 28h

Bit	7	6	5	4	3	2	1	0
Read	0	0	SELFTST	CTR	CBC	AES	0	Reserved
Write	CLEAR	START					LOAD_MAC	
Reset	0	0	0	0	0	0	0	0

Modem_ASM_CTRL1 field descriptions

Field	Description
7 CLEAR	Clear Write 1 to clear all memory in ASM. Writing a 0 does nothing. See the ASM section for programming details.
6 START	Start Write 1 to start the CBC or CTR or CCM or AES mode encryption. If SELFTST is set to 1, then start will start selftest mode.
5 SELFTST	Self Test

Table continues on the next page...

Modem_ASM_CTRL1 field descriptions (continued)

Field	Description
	Write 1 to start set mode to self test (not self clearing). 0 Module not in selftest mode. 1 Module in selftest mode.
4 CTR	Need Name Write 1 to put ASM into counter encryption mode. 0 Counter mode encryption will not be performed. 1 Counter mode encryption will be performed.
3 CBC	Need Name Write 1 to put ASM into CBC-MAC mode. 0 CBC-MAC generation will not be performed. 1 CBC-MAC generation will be performed.
2 AES	Need Name Write 1 to put ASM into AES mode. 0 AES mode encrption will not be performed. 1 AES mode encrption will be performed.
1 LOAD_MAC	Need Name Write 1 to pre-load MAC with a value. Writing a 0 does nothing.
0 Reserved	Reserved This field is reserved.

9.2.41 ASM Control 2 (Modem_ASM_CTRL2)

Address: 0h base + 29h offset = 29h

Bit	7	6	5	4	3	2	1	0
Read	DATA_REG_TYPE_SELECT			Reserved			LOAD_MAC	Reserved
Write								
Reset	0	0	0	0	0	0	0	0

Modem_ASM_CTRL2 field descriptions

Field	Description
7–5 DATA_REG_ TYPE_SELECT	Data Register Type Select This register selects the type of data to be written to or read from the 16x8 Data register field. 000 Key: The 16 AES Data register holds the key used for encryption. 001 Data: The 16 AES Data register hold the data to be encrypted or decrypted 010 CTR: The 16 AES Data register hold the counter value used to encrypt the data in counter mode.

Table continues on the next page...

Modem_ASM_CTRL2 field descriptions (continued)

Field	Description
	011 CTR Result: The 16 AES Data register hold the counter mode encyption result 100 CBC Result: The 16 AES Data register hold CBC-MAC generated results. 101 MAC: The 16 AES Data register hold the starting value for CBC-MAC generation. 110 AES Result: The 16 AES Data register hold the plain AES mode encryption result. 111 Reserved.
4–2 Reserved	Reserved This field is reserved.
1 LOAD_MAC	Test Pass If this bit is 1 then the hardware passed the self test. If self test fails then the module is not usable and the outputs will be set to all 0. The default state for this bit is 0. The software must initiate a self test before the module can be used.
0 Reserved	Reserved This field is reserved.

9.2.42 ASM Data (Modem_ASM_DATA*n*)

Address: 0h base + 2Ah offset + (1d × i), where i=0d to 15d

Bit	7	6	5	4	3	2	1	0
Read Write	ASM_DATA							
Reset	0	0	0	0	0	0	0	0

Modem_ASM_DATA*n* field descriptions

Field	Description
ASM_DATA	ASM Data The 16x8 AES Data registers (128 bits) block serve as the interface between the ASM's internal memory segments and the external MCU. The destination or source of the data in these registers is controlled through the DATA_REG_TYPE_SELECT field in ASM_CTRL2. See the ASM section for programming detail. Register ASM_DATA_0 contains the fields ASM_DATA[7] through ASM_DATA_[0], ASM_DATA_1 contains the fields ASM_DATA[15] through ASM_DATA[8].

9.2.43 Overwrite Version Number (Modem_OVERWRITE_VER)

Address: 0h base + 3Bh offset = 3Bh

Bit	7	6	5	4	3	2	1	0
Read Write	OVERWRITE_VER							
Reset	0	0	0	0	0	0	0	0

Modem_OVERWRITE_VER field descriptions

Field	Description
OVERWRITE_ VER	Overwrite Version Number Contains the version number of the overwrites.h file, used by stack software to write non-default values into the radio's register map.

9.2.44 CLK_OUT Control (Modem_CLK_OUT_CTRL)

This register provides control for CLK_OUT.

Address: 0h base + 3Ch offset = 3Ch

Bit	7	6	5	4
Read Write	CLK_OUT_EXTEND	CLK_OUT_HIZ	CLK_OUT_SR	CLK_OUT_DS
Reset	1	0	0	0

Bit	3	2	1	0
Read Write	CLK_OUT_EN	CLK_OUT_DIV		
Reset	1	0	0	0

Modem_CLK_OUT_CTRL field descriptions

Field	Description
7 CLK_OUT_ EXTEND	CLK_OUT Extend Briefly extend CLK_OUT after deassertion of CLK_OUT_EN. 0 Extends CLK_OUT for 128 clock cycles after the deassertion of CLK_OUT_EN. 1 Deassertion of CLK_OUT_EN immediately stops CLK_OUT.
6 CLK_OUT_HIZ	CLK_OUT High Impedence Select Selects whether CLK_OUT is placed into high-impedence mode or output mode. 0 CLK_OUT pad is in output mode (Output Buffer Enable=1, and Input Buffer Enable=0). 1 Place the CLK_OUT pad in high-impedence mode (Output Buffer Enable=0, and Input Buffer Enable=0).
5 CLK_OUT_SR	CLK_OUT Slew Rate Enables or disables slew rate limiting for CLK_OUT pad. 0 Disable slew rate limiting for CLK_OUT pad. 1 Enable slew rate limiting for CLK_OUT pad.
4 CLK_OUT_DS	CLK_OUT Drive Strength Selects the drive strength for CLK_OUT. 0 Normal drive strength for CLK_OUT pad. 1 High drive strength for CLK_OUT pad.

Table continues on the next page...

Modem_CLK_OUT_CTRL field descriptions (continued)

Field	Description
3 CLK_OUT_EN	Transmitter Interrupt Status Shows the status of a transmit operation. This is write a 1 to clear bit. 0 CLK_OUT pad is held at 0 (assuming CLK_OUT_HIZ=0). 1 CLK_OUT pad is driven with a clock whose frequency is determined by the CLK_OUT_DIV setting.
CLK_OUT_DIV	CLK_OUT Divide If programming CLK_OUT to >8 MHz, it is advisable to set CLK_OUT_DS=1. (High drive strength CLK_OUT pad.) The CLK_OUT frequency (if CLK_OUT_EN=1), is as follows: 000 32 MHz 001 16 MHz 010 8 MHz 011 4 MHz 100 1 MHz 101 250 kHz 110 62.5 kHz 111 32.787 kHz (32M Hz / 976)

9.2.45 Power Modes (Modem_PWR_MODES)

This register determines the power modes.

Address: 0h base + 3Dh offset = 3Dh

Bit	7	6	5	4
Read	Reserved		XTAL_READY	XTALEN
Write				
Reset	0	0	0	1

Bit	3	2	1	0
Read	ASM_CLK_EN	Reserved	AUTODOZE	PMC_MODE
Write				
Reset	0	0	0	1

Modem_PWR_MODES field descriptions

Field	Description
7–6 Reserved	Reserved This field is reserved.
5 XTAL_READY	XTAL Ready Indicates whether the 32 MHz crystal oscillator has completed its warmup and is supplying clocks to the digital core. Read-only.

Table continues on the next page...

Modem_PWR_MODES field descriptions (continued)

Field	Description
	0 The 32 MHz crystal oscillator has not completed its warmup and is supplying clocks to the digital core.. 1 The 32 MHz crystal oscillator has completed its warmup and is supplying clocks to the digital core.
4 XTALEN	CLK_OUT Slew Rate HIBERNATE mode can be entered by setting XTALEN=0 and PMC_MODE=0. DOZE mode can be entered by setting XTALEN=1 and PMC_MODE=0. IDLE mode can be entered by setting XTALEN=1 and PMC_MODE=1. A programmed low-to-high transition on XTALEN will cause the WAKE_IRQ and WAKE_FROM_HIB status bits to become set, after the oscillator warmup has completed (see IRQSTS2 register). XTALEN=1 is mandatory for all transceiver operations, including transceiver timers. 0 Disable the 32 MHz crystal oscillator. 1 Enable the 32 MHz crystal oscillator.
3 ASM_CLK_EN	ASM Clock Enable.
2 Reserved	Reserved This field is reserved.
1 AUTODOZE	Automatic Doze Mode Automatic Doze mode allows the PMC operating mode to be controlled automatically by the sequence manager. In this mode, the PMC will be in high-power mode during all transceiver sequences (i.e., Run state), and in low power mode at all other times (i.e., Doze state). If AUTODOZE=0, then PMC operating mode is under direct software control (PMC_MODE bit) 0 Disable Automatic Doze mode. 1 Enable Automatic Doze mode.
0 PMC_MODE	PMC Mode Selects whether the PMC is in high-power mode or low-power mode. 0 PMC is in low-power mode. 1 PMC is in high-power mode.

9.2.46 IAR Index (Modem_IAR_INDEX)

Address: 0h base + 3Eh offset = 3Eh

Bit	7	6	5	4	3	2	1	0
Read								
Write	IAR_INDEX							
Reset	0	0	0	0	0	0	0	0

Modem_IAR_INDEX field descriptions

Field	Description
IAR_INDEX	IAR Index

Modem_IAR_INDEX field descriptions (continued)

Field	Description
	A pointer into Indirect Access address space. During an SPI register access, when IAR_INDEX appears as the address in the control word (first SPI byte), the following byte will be loaded into IAR_INDEX, and the byte after that will write (or read) a byte from the indirect access register pointed to by IAR_INDEX. Hardware will automatically increment the pointer for subsequent byte accesses within the SPI burst transfer. See the SPI chapter for more information. Indirect Access auto-incrementing is not available in Hibernate state and should not be attempted; only single-access indirect register transfers are available in Hibernate. Direct address auto-incrementing is available in all power states. IAR_INDEX is not affected by any soft reset bit (see SOFT_RESET register).

9.2.47 IAR Data (Modem_IAR_DATA)

Address: 0h base + 3Fh offset = 3Fh

Bit	7	6	5	4	3	2	1	0
Read Write	IAR_DATA							
Reset	0	0	0	0	0	0	0	0

Modem_IAR_DATA field descriptions

Field	Description
IAR_DATA	IAR DATA Window into Indirect Access Space. Data byte to be written (or read) to (from) this address during an Indirect Register Access will be written to (or read from) the Indirect Register pointed to by IAR_INDEX for the first data phase of the transfer, and the Indirect Address pointer will be automatically incremented for subsequent data phases during the SPI burst transfer. Indirect Access auto-incrementing is not available in Hibernate state and should not be attempted. Only single-access indirect register transfers are available in Hibernate. Direct address auto-incrementing is available in all power states. IAR_INDEX is not affected by any soft reset bit (see SOFT_RESET register). The read-write access of this register is determined by the read-write access of the register in Indirect Access space to which the IAR_INDEX is pointing.

9.3 Indirect registers memory map and register definition

The Indirect radio registers contains many bitfields for communicating with the radio.

NOTE

> The Direct and Indirect registers can be accessed only through the SPI interface. Direct access by a single SPI control word and Indirect using the pointer register (IAR_INDEX) and the data register (IAR_DATA), which occupy the last two registers in direct access; addresses 0x3E and 0x3F, respectively.

Indirect_Modem memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
0	Part Identification (Indirect_Modem_PART_ID)	8	R		9.3.1/233
1	XTAL 32 MHz Trim (Indirect_Modem_XTAL_TRIM)	8	R/W	77h	9.3.2/233
3	MAC PAN ID for PAN0 (Indirect_Modem_MACPANID0_LSB)	8	R/W	FFh	9.3.3/234
4	MAC PAN ID for PAN0 (Indirect_Modem_MACPANID0_MSB)	8	R/W	FFh	9.3.3/234
5	MAC Short Address for PAN0 (Indirect_Modem_MACSHORTADDRS0_LSB)	8	R/W	FFh	9.3.4/234
6	MAC Short Address for PAN0 (Indirect_Modem_MACSHORTADDRS0_MSB)	8	R/W	FFh	9.3.4/234
7	MAC Long Address for PAN0 (Indirect_Modem_MACLONGADDRS0_0)	8	R/W	FFh	9.3.5/235
8	MAC Long Address for PAN0 (Indirect_Modem_MACLONGADDRS0_1)	8	R/W	FFh	9.3.5/235
9	MAC Long Address for PAN0 (Indirect_Modem_MACLONGADDRS0_2)	8	R/W	FFh	9.3.5/235
A	MAC Long Address for PAN0 (Indirect_Modem_MACLONGADDRS0_3)	8	R/W	FFh	9.3.5/235
B	MAC Long Address for PAN0 (Indirect_Modem_MACLONGADDRS0_4)	8	R/W	FFh	9.3.5/235
C	MAC Long Address for PAN0 (Indirect_Modem_MACLONGADDRS0_5)	8	R/W	FFh	9.3.5/235
D	MAC Long Address for PAN0 (Indirect_Modem_MACLONGADDRS0_6)	8	R/W	FFh	9.3.5/235
E	MAC Long Address for PAN0 (Indirect_Modem_MACLONGADDRS0_7)	8	R/W	FFh	9.3.5/235
F	Receive Frame Filter (Indirect_Modem_RX_FRAME_FILTER)	8	R/W	0Fh	9.3.6/235
10	Frequency Integer for PAN1 (Indirect_Modem_PLL_INT1)	8	R/W	1Fh	9.3.7/236
11	Frequency Fractional Value for PAN1 (Indirect_Modem_PLL_FRAC1_LSB)	8	R/W	FFh	9.3.8/237
12	Frequency Fractional Value for PAN1 (Indirect_Modem_PLL_FRAC1_MSB)	8	R/W	FFh	9.3.8/237
13	Frequency Fractional Value for PAN1 (Indirect_Modem_MACPANID1_LSB)	8	R/W	FFh	9.3.9/237
14	Frequency Fractional Value for PAN1 (Indirect_Modem_MACPANID1_MSB)	8	R/W	FFh	9.3.9/237
15	MAC Short Address for PAN1 (Indirect_Modem_MACSHORTADDRS1_LSB)	8	R/W	FFh	9.3.10/238
16	MAC Short Address for PAN1 (Indirect_Modem_MACSHORTADDRS1_MSB)	8	R/W	FFh	9.3.10/238
17	MAC Long Address for PAN1 (Indirect_Modem_MACLONGADDRS1_0)	8	R/W	FFh	9.3.11/238

Table continues on the next page...

Indirect_Modem memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
18	MAC Long Address for PAN1 (Indirect_Modem_MACLONGADDRS1_1)	8	R/W	FFh	9.3.11/238
19	MAC Long Address for PAN1 (Indirect_Modem_MACLONGADDRS1_2)	8	R/W	FFh	9.3.11/238
1A	MAC Long Address for PAN1 (Indirect_Modem_MACLONGADDRS1_3)	8	R/W	FFh	9.3.11/238
1B	MAC Long Address for PAN1 (Indirect_Modem_MACLONGADDRS1_4)	8	R/W	FFh	9.3.11/238
1C	MAC Long Address for PAN1 (Indirect_Modem_MACLONGADDRS1_5)	8	R/W	FFh	9.3.11/238
1D	MAC Long Address for PAN1 (Indirect_Modem_MACLONGADDRS1_6)	8	R/W	FFh	9.3.11/238
1E	MAC Long Address for PAN1 (Indirect_Modem_MACLONGADDRS1_7)	8	R/W	FFh	9.3.11/238
1F	Dual PAN Control (Indirect_Modem_DUAL_PAN_CTRL)	8	R/W	FFh	9.3.12/239
20	Channel Frequency Dwell Time (Indirect_Modem_DUAL_PAN_DWELL)	8	R/W	00h	9.3.13/240
21	Dual PAN Status (Indirect_Modem_DUAL_PAN_STS)	8	R	00h	9.3.14/240
22	Clear Channel Assessment 1 Threshold (Indirect_Modem_CCA1_THRESH)	8	R/W	4Bh	9.3.15/241
23	Clear Channel Assessment / ED Offset Computation (Indirect_Modem_CCA1_ED_OFFSET_COMP)	8	R/W	6Dh	9.3.16/242
24	LQI Offset Computation (Indirect_Modem_LQI_OFFSET_COMP)	8	R/W	24h	9.3.17/242
25	CCA Control (Indirect_Modem_CCA_CTRL)	8	R/W	5Fh	9.3.18/242
26	Clear Channel Assessment 2 Threshold Peak Compare (Indirect_Modem_CCA2_CORR_PEAKS)	8	R/W	60h	9.3.19/243
27	Clear Channel Assessment 2 Threshold (Indirect_Modem_CCA2_THRESH)	8	R/W	82h	9.3.20/244
28	TMR PRESCALE (Indirect_Modem_TMR_PRESCALE)	8	R/W	03h	9.3.21/244
2A	GPIO Data (Indirect_Modem_GPIO_DATA)	8	R/W	00h	9.3.22/245
2B	GPIO Direction Control (Indirect_Modem_GPIO_DIR)	8	R/W	00h	9.3.23/246
2C	GPIO Pullup Enable (Indirect_Modem_GPIO_PUL_EN)	8	R/W	FFh	9.3.24/247
2D	GPIO Pullup Select (Indirect_Modem_GPIO_SEL)	8	R/W	FFh	9.3.25/249
2E	GPIO Drive Strength (Indirect_Modem_GPIO_DS)	8	R/W	FFh	9.3.26/250
30	Antenna Control (Indirect_Modem_ANT_PAD_CTRL)	8	R/W	08h	9.3.27/251
31	Miscellaneous Pad Control (Indirect_Modem_MISC_PAD_CTRL)	8	R/W	1Ah	9.3.28/253
35	RX_BYTE_COUNT (Indirect_Modem_RX_BYTE_COUNT)	8	R	00h	9.3.29/253
36	RX_WTR_MARK (Indirect_Modem_RX_WTR_MARK)	8	R/W	FFh	9.3.30/254
38	TXDELAY (Indirect_Modem_TXDELAY)	8	R/W	00h	9.3.31/254
39	ACKDELAY (Indirect_Modem_ACKDELAY)	8	R/W	3Dh	9.3.32/255

Table continues on the next page...

Indirect_Modem memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
51	Antenna AGC and FAD Control (Indirect_Modem_ANT_AGC_CTRL)	8	R/W	40h	9.3.33/255
56	LPPS_CTRL (Indirect_Modem_LPPS_CTRL)	8	R/W	1Eh	9.3.34/256
5B	RSSI (Indirect_Modem_RSSI)	8	R	00h	9.3.35/257
6E	XTAL Control (Indirect_Modem_XTAL_CTRL)	8	R/W	03h	9.3.36/258

9.3.1 Part Identification (Indirect_Modem_PART_ID)

The Part Identification register contains information on the mask set, the version, and the manufacturer of the transceiver.

Address: 0h base + 0h offset = 0h

Bit	7	6	5	4	3	2	1	0
Read	MANUF_ID		VERSION			MASK_SET		
Write								
Reset								

Indirect_Modem_PART_ID field descriptions

Field	Description
7–6 MANUF_ID	Manufacturer ID
5–3 VERSION	Version This field contains information about the quadrature version of the LNA mixer and VCO. 001 Single quadrature version of the LNA mixer and VCO. 010 Double quadrature version of the LNA mixer and VCO.
MASK_SET	Mask Set This field contains information about the mask set of the transceiver.

9.3.2 XTAL 32 MHz Trim (Indirect_Modem_XTAL_TRIM)

Address: 0h base + 1h offset = 1h

Bit	7	6	5	4	3	2	1	0
Read Write	XTAL_TRIM							
Reset	0	1	1	1	0	1	1	1

Indirect_Modem_XTAL_TRIM field descriptions

Field	Description
XTAL_TRIM	XTAL 32 MHz Trim XTAL 32 MHz trimming. See the XTAL32M section for more information.

9.3.3 MAC PAN ID for PAN0 (Indirect_Modem_MACPANID0*n*)

Address: 0h base + 3h offset + (1d × i), where i=0d to 1d

Bit	7	6	5	4	3	2	1	0
Read Write	MACPANID0							
Reset	1	1	1	1	1	1	1	1

Indirect_Modem_MACPANID0*n* field descriptions

Field	Description
MACPANID0	MAC PAN ID for PAN0 The packet processor compares the incoming packet's destination PAN ID against the contents of this register to determine whether the packet is addressed to this device. Or, if the incoming packet is a Beacon frame, the packet processor compares the incoming packet source PAN ID against this register. Also, if PANCORDNTR0=1, and the incoming packet has no Destination Address field, and if the incoming packet is a Data or MAC Command frame, the packet processor compares the incoming packet source PAN ID against this register.

9.3.4 MAC Short Address for PAN0 (Indirect_Modem_MACSHORTADDRS0*n*)

Address: 0h base + 5h offset + (1d × i), where i=0d to 1d

Bit	7	6	5	4	3	2	1	0
Read Write	MACSHORT_ADDRS0							
Reset	1	1	1	1	1	1	1	1

Indirect_Modem_MACSHORTADDRS0*n* field descriptions

Field	Description
MACSHORT_ ADDRS0	MAC Short Address for PAN0 MAC Short Address for PAN0, for 16-bit destination addressing mode. The packet processor compares the incoming packet's Destination Address against the contents of this register to determine whether the packet is addressed to this device.

9.3.5 MAC Long Address for PAN0 (Indirect_Modem_MACLONGADDRS0*n*)

Address: 0h base + 7h offset + (1d × i), where i=0d to 7d

Bit	7	6	5	4	3	2	1	0
Read Write	MACLONGADDRS0							
Reset	1	1	1	1	1	1	1	1

Indirect_Modem_MACLONGADDRS0*n* field descriptions

Field	Description
MACLONGADDRS0	MAC Long Address for PAN0 MAC Long Address for PAN0, for 64-bit destination addressing mode. The packet processor compares the incoming packet’s Destination Address against the contents of this register to determine whether the packet is addressed to this device.

9.3.6 Receive Frame Filter (Indirect_Modem_RX_FRAME_FILTER)

Address: 0h base + Fh offset = Fh

Bit	7	6	5	4
Read Write	FRM_VER		ACTIVE_ PROMISCUOUS	NS_FT
Reset	0	0	0	0

Bit	3	2	1	0
Read Write	CMD_FT	ACK_FT	DATA_FT	BEACON_FT
Reset	1	1	1	1

Indirect_Modem_RX_FRAME_FILTER field descriptions

Field	Description
7–6 FRM_VER	Frame Version Selector The incoming packet's Frame Control Field is parsed to obtain the FrameVersion subfield, and that value is compared against this register, in accordance with the following. Frames received with FrameVersion 2 or 3 will be treated identically to FrameVersion 1, with respect to parsing of the Auxiliary Security Header. Other than this Header, all 4 frame versions will be treated identically 00 Any FrameVersion accepted (0, 1, 2 and 3). 01 Accept only FrameVersion 0 packets (2003 compliant). 10 Accept only FrameVersion 1 packets (2006 compliant). 11 Accept FrameVersion 0 and 1 packets; reject all others.
5 ACTIVE_ PROMISCUOUS	Active Promiscuous

Table continues on the next page...

Indirect_Modem_RX_FRAME_FILTER field descriptions (continued)

Field	Description
	0 Normal operation (default). 1 Provide data indication on all received packets under the same rules that apply in Promiscuous mode, however acknowledge those packets under rules that apply in non-Promiscuous mode
4 NS_FT	Not-Specified (Reserved) Frame Type Field determines whether Not-Specified (Reserved) frame types are enabled. 0 Reject all reserved frames. 1 Not-Specified (reserved) frame type enabled. No packet filtering is performed, except for frame length checking (FrameLength is greater than or equal to 5 and FrameLength is less than or equal to 127).
3 CMD_FT	MAC Command Frame Type Field determines whether MAC Command frame types are enabled. 0 Reject all MAC Command frames. 1 MAC Command frame type enabled.
2 ACK_FT	Acknowledge Frame Type Field determines whether Acknowledge frame types are enabled. 0 Reject all Acknowledge frames. 1 Acknowledge frame type enabled.
1 DATA_FT	Data Frame Type Field determines whether Data frame types are enabled. 0 Reject all Data frames. 1 Data frame type enabled.
0 BEACON_FT	Beacon Frame Type Field determines whether Beacon frame types are enabled. 0 Reject all Beacon frames. 1 Beacon frame type enabled.

9.3.7 Frequency Integer for PAN1 (Indirect_Modem_PLL_INT1)

Address: 0h base + 10h offset = 10h

Bit	7	6	5	4	3	2	1	0
Read Write	Reserved			PLL_INT1				
Reset	0	0	0	1	1	1	1	1

Indirect_Modem_PLL_INT1 field descriptions

Field	Description
7–5 RESERVED	This field is reserved.

Table continues on the next page...

Indirect_Modem_PLL_INT1 field descriptions (continued)

Field	Description
PLL_INT1	Frequency Integer for PAN1 F = ((PLL_INT1+64) + (PLL_FRAC1/65536)) * 32 MHz. See PLL section for more details. This register should not be programmed (should be left in its default state) if Dual PAN mode is not in use.

9.3.8 Frequency Fractional Value for PAN1 (Indirect_Modem_PLL_FRAC1*n*)

Address: 0h base + 11h offset + (1d × i), where i=0d to 1d

Bit	7	6	5	4	3	2	1	0
Read Write	PLL_FRAC1							
Reset	1	1	1	1	1	1	1	1

Indirect_Modem_PLL_FRAC1*n* field descriptions

Field	Description
PLL_FRAC1	Frequency Fractional Value for PAN1 F = ((PLL_INT1+64) + (PLL_FRAC1/65536)) * 32 MHz. This register should not be programmed (should be left in its default state) if Dual PAN mode is not in use.

9.3.9 Frequency Fractional Value for PAN1 (Indirect_Modem_MACPANID1*n*)

Address: 0h base + 13h offset + (1d × i), where i=0d to 1d

Bit	7	6	5	4	3	2	1	0
Read Write	MACPANID1							
Reset	1	1	1	1	1	1	1	1

Indirect_Modem_MACPANID1*n* field descriptions

Field	Description
MACPANID1	MAC PAN ID for PAN1 The packet processor compares the incoming packet's Destination PAN ID against the contents of this register to determine if the packet is addressed to this device; or if the incoming packet is a Beacon frame, the packet processor compares the incoming packet source PAN ID against this register. Also, if PANCORDNTR0=1, and the incoming packet has no Destination Address field, and if the incoming packet is a Data or MAC Command frame, the packet processor compares the incoming packet source PAN ID against this register.

9.3.10 MAC Short Address for PAN1 (Indirect_Modem_MACSHORTADDRS1*n*)

Address: 0h base + 15h offset + (1d × i), where i=0d to 1d

Bit	7	6	5	4	3	2	1	0
Read Write	MACSHORTADDRS1							
Reset	1	1	1	1	1	1	1	1

Indirect_Modem_MACSHORTADDRS1*n* field descriptions

Field	Description
MACSHORTADDRS1	MAC Short Address for PAN1 For 16-bit destination addressing mode. The packet processor compares the incoming packet's Destination Address against the contents of this register to determine if the packet is addressed to this device.

9.3.11 MAC Long Address for PAN1 (Indirect_Modem_MACLONGADDRS1*n*)

Address: 0h base + 17h offset + (1d × i), where i=0d to 7d

Bit	7	6	5	4	3	2	1	0
Read Write	MACLONGADDRS1							
Reset	1	1	1	1	1	1	1	1

Indirect_Modem_MACLONGADDRS1*n* field descriptions

Field	Description
MACLONGADDRS1	MAC Long Address for PAN1 For 64-bit destination addressing mode. The packet processor compares the incoming packet's Destination Address against the contents of this register to determine whether the packet is addressed to this device.

9.3.12 Dual PAN Control (Indirect_Modem_DUAL_PAN_CTRL)

Address: 0h base + 1Fh offset = 1Fh

Bit	7	6	5	4
Read Write	DUAL_PAN_SAM_LVL			
Reset	1	1	1	1

Bit	3	2	1	0
Read	CURRENT_NETWORK	PANCORDNTR1	DUAL_PAN_AUTO	ACTIVE_NETWORK
Write	▨			
Reset	1	1	1	1

Indirect_Modem_DUAL_PAN_CTRL field descriptions

Field	Description
7–4 DUAL_PAN_ SAM_LVL	Source Address Matching Level For the Source Address Matching Table, sets the dividing line between the PAN0 and PAN1 sections of the table. Table indeces at or above this level belong to PAN1. Table entries below this level belong to PAN0. Default = 12 (entire table is PAN0). In Dual PAN mode, if both PANs occupy the same channel, hardware detects this and makes the entire Source Address Matching Table available to both PANs simultaneously; in this case, DUAL_PAN_SAM_LVL has no effect.
3 CURRENT_ NETWORK	Current Network This read-only bit indicates which PAN is currently selected by hardware in automatic Dual PAN mode. 0 PAN0 is selected. 1 PAN1 is selected.
2 PANCORDNTR1	PAN Coordinator on PAN1 Device is a PAN Coordinator on PAN1. Allows device to receive packets with no destination address, if Source PAN ID matches.
1 DUAL_PAN_ AUTO	Dual Pan Automatic Operating Mode Activates automatic Dual PAN operating mode. In this mode, PAN-switching is controlled by hardware at a pre-programmed rate, determined by DUAL_PAN_DWELL. Whenever DUAL_PAN_AUTO=0, CURRENT_NETWORK=ACTIVE_NETWORK at all times. In other words, software directly controls which PAN is selected. Whenever DUAL_PAN_AUTO=1, CURRENT_NETWORK is controlled by hardware. See the packet processor section for more detail. 0 Manual Dual PAN mode (or Single PAN mode). 1 Auto Dual PAN Mode.
0 ACTIVE_ NETWORK	Active Network Selects the PAN on which to transceive, by activating a PAN parameter set (PAN0 or PAN1). In Manual Dual PAN mode (or Single PAN mode), this bit selects the active PAN parameter set (channel and addressing parameters) which governs all autosequences. In Auto Dual PAN mode, this bit selects the PAN on which to begin transceiving, latched at the point at which DUAL_PAN_DWELL register is written. See packet processor section for more detail.

Table continues on the next page...

Indirect_Modem_DUAL_PAN_CTRL field descriptions (continued)

Field	Description
	0 Select PAN0. 1 Select PAN1.

9.3.13 Channel Frequency Dwell Time (Indirect_Modem_DUAL_PAN_DWELL)

Address: 0h base + 20h offset = 20h

Bit	7	6	5	4	3	2	1	0
Read Write	DUAL_PAN_DWELL							
Reset	0	0	0	0	0	0	0	0

Indirect_Modem_DUAL_PAN_DWELL field descriptions

Field	Description
DUAL_PAN_ DWELL	Channel Frequency Dwell Time In Auto Dual PAN mode, hardware will toggle the PAN, after dwelling on the current PAN for the interval described below (assuming Preamble/SFD not detected). A write to DUAL_PAN_DWELL, always re-initilizes the DWELL TIMER to the programmed value. If a write to DUAL_PAN_DWELL occurs during an autosequence, the DWELL TIMER will begin counting down immediately. If a write to DUAL_PAN_DWELL occurs when there is no autosequence underway, the DWELL TIMER will not begin counting until the next autosequence begins; it will begin counting at the start of the sequence warmup. In Auto Dual PAN mode, hardware will toggle the PAN, after dwelling on the current PAN for the interval described below (assuming Preamble/SFD not detected). A write to DUAL_PAN_DWELL, always re-initilizes the DWELL TIMER to the programmed value. If a write to DUAL_PAN_DWELL occurs during an autosequence, the DWELL TIMER will begin counting down immediately. If a write to DUAL_PAN_DWELL occurs when there is no autosequence underway, the DWELL TIMER will not begin counting until the next autosequence begins; it will begin counting at the start of the sequence warmup. A write to DUAL_PAN_DWELL also causes the value of ACTIVE_NETWORK to get latched into the hardware. This latched value will be the starting point for the automatic dual-pan mode (i.e., start on PAN0 or on PAN1). The starting value takes effect immediately (if sequence is underway and DUAL_PAN_AUTO=1), or is otherwise delayed until sequence starts and DUAL_PAN_AUTO=1. DUAL_PAN_DWELL should not be accessed in the Hibernate state.

9.3.14 Dual PAN Status (Indirect_Modem_DUAL_PAN_STS)

Address: 0h base + 21h offset = 21h

Bit	7	6	5	4	3	2	1	0
Read	RECD_ON_ PAN1	RECD_ON_ PAN0	DUAL_PAN_REMAIN					
Write								
Reset	0	0	0	0	0	0	0	0

Indirect_Modem_DUAL_PAN_STS field descriptions

Field	Description
7 RECD_ON_ PAN1	Received on PAN1 Indicates the packet that was just received, was received on PAN1. In Dual PAN mode operating on 2 different channels, RECD_ON_PAN0 will be set if CURRENT_NETWORK=0 when the packet was received, regardless of FILTERFAIL status. In Dual PAN mode operating with same channel on both networks, CURRENT_NETWORK will be ignored and RECD_ON_PAN0 will be set only if a valid packet was received on PAN0 (PAN0's FILTERFAIL_FLAG is deasserted). RECD_ON_PAN0 remains valid until the start of the next autoseqeuence. See the packet processor section for more detail.
6 RECD_ON_ PAN0	Received on PAN0 Indicates the packet that was just received, was received on PAN0. In Dual PAN mode operating on 2 different channels, RECD_ON_PAN0 will be set if CURRENT_NETWORK=0 when the packet was received, regardless of FILTERFAIL status. In Dual PAN mode operating with same channel on both networks, CURRENT_NETWORK will be ignored and RECD_ON_PAN0 will be set only if a valid packet was received on PAN0 (PAN0's FILTERFAIL_FLAG is deasserted). RECD_ON_PAN0 remains valid until the start of the next autoseqeuence. See the packet processor section for more detail.
DUAL_PAN_ REMAIN	Dual PAN Remaining Time This read-only register indicates time remaining before next PAN switch in auto Dual PAN mode. The units for this register, depend on the PRESCALER setting (bits [1:0]) in the DUAL_PAN_DWELL register. The readback value indicates that between N-1 and N timebase units remain until the next PAN switch. For example, a DUAL_PAN_REMAIN readback value of 3, with a DUAL_PAN_DWELL PRESCALER setting of 2 (10 ms), indicates that between 20 ms (2 * 10 ms) and 30 ms (3 * 10 ms), remain until the next automatic PAN switch. 00 0.5 ms 01 2.5 ms 10 10 ms 11 50 ms

9.3.15 Clear Channel Assessment 1 Threshold (Indirect_Modem_CCA1_THRESH)

Address: 0h base + 22h offset = 22h

Bit	7	6	5	4	3	2	1	0
Read Write	CCA1_THRESH							
Reset	0	1	0	0	1	0	1	1

Indirect_Modem_CCA1_THRESH field descriptions

Field	Description
CCA1_THRESH	Clear Channel Assessment 1 Threshold Programmable energy threshold register to detect CCA mode 1.

9.3.16 Clear Channel Assessment / ED Offset Computation (Indirect_Modem_CCA1_ED_OFFSET_COMP)

Address: 0h base + 23h offset = 23h

Bit	7	6	5	4	3	2	1	0
Read Write	CCA1_ED_OFFSET_COMP							
Reset	0	1	1	0	1	1	0	1

Indirect_Modem_CCA1_ED_OFFSET_COMP field descriptions

Field	Description
CCA1_ED_ OFFSET_COMP	Clear Channel Assessment / ED Offset Computation Programmable amount to offset CCA/ED computations.

9.3.17 LQI Offset Computation (Indirect_Modem_LQI_OFFSET_COMP)

Address: 0h base + 24h offset = 24h

Bit	7	6	5	4	3	2	1	0
Read Write	LQI_OFFSET_COMP							
Reset	0	0	1	0	0	1	0	0

Indirect_Modem_LQI_OFFSET_COMP field descriptions

Field	Description
LQI_OFFSET_ COMP	LQI Offset Computation Programmable amount to offset RSSI based LQI value.

9.3.18 CCA Control (Indirect_Modem_CCA_CTRL)

Address: 0h base + 25h offset = 25h

Bit	7	6	5	4
Read Write	Reserved	AGC_FRZ_EN	CONT_RSSI_EN	LQI_RSSI_NOT_CORR
Reset	0	1	0	1

Bit	3	2	1	0
Read Write	CCA3_AND_NOT_OR	OWER_COMP_EN_ LQI	OWER_COMP_EN_ED	OWER_COMP_EN_ CCA1
Reset	1	1	1	1

Indirect_Modem_CCA_CTRL field descriptions

Field	Description
7 Reserved	This field is reserved.
6 AGC_FRZ_EN	AGC Freeze Enable 0 AGC operates continuously during reception. 1 Enable the AGC Freeze mode after SFD detection.
5 CONT_RSSI_EN	Continuous RSSI Averaging Enable 0 RSSI averaging stops 64 μs after preamble is detected (for LQI), or after the CCA or Energy Detect operation is completed, for Sequence C. 1 Enable continuous RSSI averaging.
4 LQI_RSSI_NOT_ CORR	LQI Mode 0 LQI based on Correlation peaks. 1 LQI based on RSSI.
3 CCA3_AND_ NOT_OR	Determines the way CCA3 is required to be detected. 0 CCA1 or CCA2. 1 CCA1 and CCA2.
2 OWER_COMP_ EN_LQI	Enable offset compensation and slope correction to averaged RSSI value during LQI computation, when LQI_RSSI_NOT_CORR=1.
1 OWER_COMP_ EN_ED	Enable offset compensation and slope correction to averaged RSSI value during Energy Detect mode.
0 OWER_COMP_ EN_CCA1	Enable offset compensation and slope correction to averaged RSSI value during CCA1 mode.

9.3.19 Clear Channel Assessment 2 Threshold Peak Compare (Indirect_Modem_CCA2_CORR_PEAKS)

Address: 0h base + 26h offset = 26h

Bit	7	6	5	4	3	2	1	0
Read	Reserved	CCA2_MIN_NUM_CORR_TH			CCA2_NUM_CORR_PEAKS			
Write								
Reset	0	1	1	0	0	0	0	0

Indirect_Modem_CCA2_CORR_PEAKS field descriptions

Field	Description
7 Reserved	This field is reserved.

Table continues on the next page...

Indirect_Modem_CCA2_CORR_PEAKS field descriptions (continued)

Field	Description
6–4 CCA2_MIN_ NUM_CORR_TH	Programmable threshold to be compared against number of correlation peaks that exceeded cca2_corr_thresh for detecting CCA mode 2. Number of peaks detected = cca2_min_num_corr_th + 1; Example: If it is programmed to 3, CCA2 logic looks for at least 4 correlation peaks that crossed the threshold, to indicate channel is idle or busy.
CCA2_NUM_ CORR_PEAKS	Counts of number of peaks that crossed cca2_corr_thresh. Read-only.

9.3.20 Clear Channel Assessment 2 Threshold (Indirect_Modem_CCA2_THRESH)

Address: 0h base + 27h offset = 27h

Bit	7	6	5	4	3	2	1	0
Read Write	CCA2_THRESH							
Reset	1	0	0	0	0	0	1	0

Indirect_Modem_CCA2_THRESH field descriptions

Field	Description
CCA2_THRESH	Clear Channel Assessment 2 Threshold Programmable energy threshold register to detect CCA mode 2.

9.3.21 TMR PRESCALE (Indirect_Modem_TMR_PRESCALE)

Address: 0h base + 28h offset = 28h

Bit	7	6	5	4	3	2	1	0
Read Write	Reserved					TMR_PRESCALE		
Reset	0	0	0	0	0	0	1	1

Indirect_Modem_TMR_PRESCALE field descriptions

Field	Description
7–3 Reserved	Reserved This field is reserved.
TMR_ PRESCALE	Timer Prescaler Establishes the Event Timer lock rate(Maximum timer duration). 000 Reserved 001 Reserved 010 500 kHz(33.55 S)

Table continues on the next page...

Indirect_Modem_TMR_PRESCALE field descriptions (continued)

Field	Description
	011 250 kHZ(67.11 S)-default 100 125 kHZ(134.22 S) 101 62.5 kHZ(268.44 S) 110 31.25 kHZ(536.87 S) 111 15.625 kHZ(1073.74 S)

9.3.22 GPIO Data (Indirect_Modem_GPIO_DATA)

Address: 0h base + 2Ah offset = 2Ah

Bit	7	6	5	4	3	2	1	0
Read Write	GPIO_ DATA8	GPIO_ DATA7	GPIO_ DATA6	GPIO_ DATA5	GPIO_ DATA4	GPIO_ DATA3	GPIO_ DATA2	GPIO_ DATA1
Reset	0	0	0	0	0	0	0	0

Indirect_Modem_GPIO_DATA field descriptions

Field	Description
7 GPIO_DATA8	GPIO Data 8 Write data to this register determines the state of the GPIO pins {GPIO8, GPIO7, ... , GPIO1}, when the corresponding GPIO_DIR bit for that pin is set to 1. Data read from this register returns the state of the GPIO pins, when the corresponding GPIO_DIR bit for that pin is set to 0
6 GPIO_DATA7	GPIO Data 7 Write data to this register determines the state of the GPIO pins {GPIO8, GPIO7, ... , GPIO1}, when the corresponding GPIO_DIR bit for that pin is set to 1. Data read from this register returns the state of the GPIO pins, when the corresponding GPIO_DIR bit for that pin is set to 0
5 GPIO_DATA6	GPIO Data 6 Write data to this register determines the state of the GPIO pins {GPIO8, GPIO7, ... , GPIO1}, when the corresponding GPIO_DIR bit for that pin is set to 1. Data read from this register returns the state of the GPIO pins, when the corresponding GPIO_DIR bit for that pin is set to 0
4 GPIO_DATA5	GPIO Data 5 Write data to this register determines the state of the GPIO pins {GPIO8, GPIO7, ... , GPIO1}, when the corresponding GPIO_DIR bit for that pin is set to 1. Data read from this register returns the state of the GPIO pins, when the corresponding GPIO_DIR bit for that pin is set to 0
3 GPIO_DATA4	GPIO Data 4 Write data to this register determines the state of the GPIO pins {GPIO8, GPIO7, ... , GPIO1}, when the corresponding GPIO_DIR bit for that pin is set to 1. Data read from this register returns the state of the GPIO pins, when the corresponding GPIO_DIR bit for that pin is set to 0
2 GPIO_DATA3	GPIO Data 3 Write data to this register determines the state of the GPIO pins {GPIO8, GPIO7, ... , GPIO1}, when the corresponding GPIO_DIR bit for that pin is set to 1. Data read from this register returns the state of the GPIO pins, when the corresponding GPIO_DIR bit for that pin is set to 0

Table continues on the next page...

Indirect_Modem_GPIO_DATA field descriptions (continued)

Field	Description
1 GPIO_DATA2	GPIO Data 2 Write data to this register determines the state of the GPIO pins {GPIO8, GPIO7, ... , GPIO1}, when the corresponding GPIO_DIR bit for that pin is set to 1. Data read from this register returns the state of the GPIO pins, when the corresponding GPIO_DIR bit for that pin is set to 0
0 GPIO_DATA1	GPIO Data 1 Write data to this register determines the state of the GPIO pins {GPIO8, GPIO7, ... , GPIO1}, when the corresponding GPIO_DIR bit for that pin is set to 1. Data read from this register returns the state of the GPIO pins, when the corresponding GPIO_DIR bit for that pin is set to 0

9.3.23 GPIO Direction Control (Indirect_Modem_GPIO_DIR)

Address: 0h base + 2Bh offset = 2Bh

Bit	7	6	5	4	3	2	1	0
Read Write	GPIO_DIR8	GPIO_DIR7	GPIO_DIR6	GPIO_DIR5	GPIO_DIR4	GPIO_DIR3	GPIO_DIR2	GPIO_DIR1
Reset	0	0	0	0	0	0	0	0

Indirect_Modem_GPIO_DIR field descriptions

Field	Description
7 GPIO_DIR8	GPIO Direction Control 8 Determines the direction for each of the GPIO pins {GPIO8, GPIO7, ... , GPIO1}. 0 Pin is in input mode. 1 Pin is in output mode.
6 GPIO_DIR7	GPIO Direction Control 7 Determines the direction for each of the GPIO pins {GPIO8, GPIO7, ... , GPIO1}. 0 Pin is in input mode. 1 Pin is in output mode.
5 GPIO_DIR6	GPIO Direction Control 6 Determines the direction for each of the GPIO pins {GPIO8, GPIO7, ... , GPIO1}. 0 Pin is in input mode. 1 Pin is in output mode.
4 GPIO_DIR5	GPIO Direction Control 5 Determines the direction for each of the GPIO pins {GPIO8, GPIO7, ... , GPIO1}. 0 Pin is in input mode. 1 Pin is in output mode.
3 GPIO_DIR4	GPIO Direction Control 4

Table continues on the next page...

Indirect_Modem_GPIO_DIR field descriptions (continued)

Field	Description
	Determines the direction for each of the GPIO pins {GPIO8, GPIO7, ... , GPIO1}. 0 Pin is in input mode. 1 Pin is in output mode.
2 GPIO_DIR3	GPIO Direction Control 3 Determines the direction for each of the GPIO pins {GPIO8, GPIO7, ... , GPIO1}. 0 Pin is in input mode. 1 Pin is in output mode.
1 GPIO_DIR2	GPIO Direction Control 2 Determines the direction for each of the GPIO pins {GPIO8, GPIO7, ... , GPIO1}. 0 Pin is in input mode. 1 Pin is in output mode.
0 GPIO_DIR1	GPIO Direction Control 1 Write data to this register determines the state of the GPIO pins {GPIO8, GPIO7, ... , GPIO1}, when the corresponding GPIO_DIR bit for that pin is set to 1. Data read from this register returns the state of the GPIO pins, when the corresponding GPIO_DIR bit for that pin is set to 0 0 Pin is in input mode. 1 Pin is in output mode.

9.3.24 GPIO Pullup Enable (Indirect_Modem_GPIO_PUL_EN)

Address: 0h base + 2Ch offset = 2Ch

Bit	7	6	5	4
Read Write	GPIO_PUL_EN8	GPIO_PUL_EN7	GPIO_PUL_EN6	GPIO_PUL_EN5
Reset	1	1	1	1

Bit	3	2	1	0
Read Write	GPIO_PUL_EN14	GPIO_PUL_EN3	GPIO_PUL_EN2	GPIO_PUL_EN1
Reset	1	1	1	1

Indirect_Modem_GPIO_PUL_EN field descriptions

Field	Description
7 GPIO_PUL_EN8	GPIO Pullup Enable 8 Determines whether a pull device (pullup or pulldown) is engaged for each of the GPIO pins {GPIO8, GPIO7, ... , GPIO1}. 0 No pullup or pulldown is engaged on the pin. 1 Pullup or pulldown is engaged (see GPIO_PUL_SEL register) on the pin

Table continues on the next page...

Indirect_Modem_GPIO_PUL_EN field descriptions (continued)

Field	Description
6 GPIO_PUL_EN7	GPIO Pullup Enable 7 Determines whether a pull device (pullup or pulldown) is engaged for each of the GPIO pins {GPIO8, GPIO7, ... , GPIO1}. 0 No pullup or pulldown is engaged on the pin. 1 Pullup or pulldown is engaged (see GPIO_PUL_SEL register) on the pin
5 GPIO_PUL_EN6	GPIO Pullup Enable 6 Determines whether a pull device (pullup or pulldown) is engaged for each of the GPIO pins {GPIO8, GPIO7, ... , GPIO1}. 0 No pullup or pulldown is engaged on the pin. 1 Pullup or pulldown is engaged (see GPIO_PUL_SEL register) on the pin
4 GPIO_PUL_EN5	GPIO Pullup Enable 5 Determines whether a pull device (pullup or pulldown) is engaged for each of the GPIO pins {GPIO8, GPIO7, ... , GPIO1}. 0 No pullup or pulldown is engaged on the pin. 1 Pullup or pulldown is engaged (see GPIO_PUL_SEL register) on the pin
3 GPIO_PUL_EN14	GPIO Pullup Enable 4 Determines whether a pull device (pullup or pulldown) is engaged for each of the GPIO pins {GPIO8, GPIO7, ... , GPIO1}. 0 No pullup or pulldown is engaged on the pin. 1 Pullup or pulldown is engaged (see GPIO_PUL_SEL register) on the pin
2 GPIO_PUL_EN3	GPIO Pullup Enable 3 Determines whether a pull device (pullup or pulldown) is engaged for each of the GPIO pins {GPIO8, GPIO7, ... , GPIO1}. 0 No pullup or pulldown is engaged on the pin. 1 Pullup or pulldown is engaged (see GPIO_PUL_SEL register) on the pin
1 GPIO_PUL_EN2	GPIO Pullup Enable 2 Determines whether a pull device (pullup or pulldown) is engaged for each of the GPIO pins {GPIO8, GPIO7, ... , GPIO1}. 0 No pullup or pulldown is engaged on the pin. 1 Pullup or pulldown is engaged (see GPIO_PUL_SEL register) on the pin
0 GPIO_PUL_EN1	GPIO Pullup Enable 1 Determines whether a pull device (pullup or pulldown) is engaged for each of the GPIO pins {GPIO8, GPIO7, ... , GPIO1}. 0 No pullup or pulldown is engaged on the pin. 1 Pullup or pulldown is engaged (see GPIO_PUL_SEL register) on the pin

9.3.25 GPIO Pullup Select (Indirect_Modem_GPIO_SEL)

Address: 0h base + 2Dh offset = 2Dh

Bit	7	6	5	4	3	2	1	0
Read Write	GPIO_PUL_ SEL8	GPIO_PUL_ SEL7	GPIO_PUL_ SEL6	GPIO_PUL_ SEL5	GPIO_PUL_ SEL4	GPIO_PUL_ SEL3	GPIO_PUL_ SEL2	GPIO_PUL_ SEL1
Reset	1	1	1	1	1	1	1	1

Indirect_Modem_GPIO_SEL field descriptions

Field	Description
7 GPIO_PUL_ SEL8	GPIO Pullup Select 8 For each pin {GPIO8, GPIO7, ... , GPIO1}, if GPIO_PUL[x]=1, this bit determines whether a pullup or pulldown is engaged. 0 Pulldown is engaged on the pin if GPIO_PUL_EN[x]=1. 1 Pullup is engaged on the pin if GPIO_PUL_EN[x]=1.
6 GPIO_PUL_ SEL7	GPIO Pullup Select 7 For each pin {GPIO8, GPIO7, ... , GPIO1}, if GPIO_PUL[x]=1, this bit determines whether a pullup or pulldown is engaged. 0 Pulldown is engaged on the pin if GPIO_PUL_EN[x]=1. 1 Pullup is engaged on the pin if GPIO_PUL_EN[x]=1.
5 GPIO_PUL_ SEL6	GPIO Pullup Select 6 For each pin {GPIO8, GPIO7, ... , GPIO1}, if GPIO_PUL[x]=1, this bit determines whether a pullup or pulldown is engaged. 0 Pulldown is engaged on the pin if GPIO_PUL_EN[x]=1. 1 Pullup is engaged on the pin if GPIO_PUL_EN[x]=1.
4 GPIO_PUL_ SEL5	GPIO Pullup Select 5 For each pin {GPIO8, GPIO7, ... , GPIO1}, if GPIO_PUL[x]=1, this bit determines whether a pullup or pulldown is engaged. 0 Pulldown is engaged on the pin if GPIO_PUL_EN[x]=1. 1 Pullup is engaged on the pin if GPIO_PUL_EN[x]=1.
3 GPIO_PUL_ SEL4	GPIO Pullup Select 4 For each pin {GPIO8, GPIO7, ... , GPIO1}, if GPIO_PUL[x]=1, this bit determines whether a pullup or pulldown is engaged. 0 Pulldown is engaged on the pin if GPIO_PUL_EN[x]=1. 1 Pullup is engaged on the pin if GPIO_PUL_EN[x]=1.
2 GPIO_PUL_ SEL3	GPIO Pullup Select 3 For each pin {GPIO8, GPIO7, ... , GPIO1}, if GPIO_PUL[x]=1, this bit determines whether a pullup or pulldown is engaged.

Table continues on the next page...

Indirect_Modem_GPIO_SEL field descriptions (continued)

Field	Description
	0 Pulldown is engaged on the pin if GPIO_PUL_EN[x]=1. 1 Pullup is engaged on the pin if GPIO_PUL_EN[x]=1.
1 GPIO_PUL_ SEL2	GPIO Pullup Select 2 For each pin {GPIO8, GPIO7, ... , GPIO1}, if GPIO_PUL[x]=1, this bit determines whether a pullup or pulldown is engaged. 0 Pulldown is engaged on the pin if GPIO_PUL_EN[x]=1. 1 Pullup is engaged on the pin if GPIO_PUL_EN[x]=1.
0 GPIO_PUL_ SEL1	GPIO Pullup Select 1 For each pin {GPIO8, GPIO7, ... , GPIO1}, if GPIO_PUL[x]=1, this bit determines whether a pullup or pulldown is engaged. 0 Pulldown is engaged on the pin if GPIO_PUL_EN[x]=1. 1 Pullup is engaged on the pin if GPIO_PUL_EN[x]=1.

9.3.26 GPIO Drive Strength (Indirect_Modem_GPIO_DS)

Address: 0h base + 2Eh offset = 2Eh

Bit	7	6	5	4	3	2	1	0
Read Write	GPIO_DS8	GPIO_DS7	GPIO_DS6	GPIO_DS5	GPIO_DS4	GPIO_DS3	GPIO_DS2	GPIO_DS1
Reset	1	1	1	1	1	1	1	1

Indirect_Modem_GPIO_DS field descriptions

Field	Description
7 GPIO_DS8	GPIO Drive Strength 8 For each pin {GPIO8, GPIO7, ... , GPIO1}: 0 Pin uses normal drive strength. 1 Pin uses high drive strength.
6 GPIO_DS7	GPIO Drive Strength 7 For each pin {GPIO8, GPIO7, ... , GPIO1}: 0 Pin uses normal drive strength. 1 Pin uses high drive strength.
5 GPIO_DS6	GPIO Drive Strength 6 For each pin {GPIO8, GPIO7, ... , GPIO1}: 0 Pin uses normal drive strength. 1 Pin uses high drive strength.
4 GPIO_DS5	GPIO Drive Strength 5

Table continues on the next page...

Indirect_Modem_GPIO_DS field descriptions (continued)

Field	Description
	For each pin {GPIO8, GPIO7, ... , GPIO1}: 0 Pin uses normal drive strength. 1 Pin uses high drive strength.
3 GPIO_DS4	GPIO Drive Strength 4 For each pin {GPIO8, GPIO7, ... , GPIO1}: 0 Pin uses normal drive strength. 1 Pin uses high drive strength.
2 GPIO_DS3	GPIO Drive Strength 3 For each pin {GPIO8, GPIO7, ... , GPIO1}: 0 Pin uses normal drive strength. 1 Pin uses high drive strength.
1 GPIO_DS2	GPIO Drive Strength 2 For each pin {GPIO8, GPIO7, ... , GPIO1}: 0 Pin uses normal drive strength. 1 Pin uses high drive strength.
0 GPIO_DS1	GPIO Drive Strength 1 For each pin {GPIO8, GPIO7, ... , GPIO1}: 0 Pin uses normal drive strength. 1 Pin uses high drive strength.

9.3.27 Antenna Control (Indirect_Modem_ANT_PAD_CTRL)

Address: 0h base + 30h offset = 30h

Bit	7	6	5	4	3	2	1	0
Read Write	ANTX_ POL3	ANTX_ POL2	ANTX_ POL1	ANTX_ POL0	ANTX_ CTRLMODE	ANTX_HZ	ANTX_EN	
Reset	0	0	0	0	1	0	0	0

Indirect_Modem_ANT_PAD_CTRL field descriptions

Field	Description
7 ANTX_POL3	Antenna Polarity Control 3 Control the polarity of the Antenna or Switch control pins. 0 Do not invert the RXSWITCH output. 1 Invert the RXSWITCH output.

Table continues on the next page...

Indirect_Modem_ANT_PAD_CTRL field descriptions (continued)

Field	Description
6 ANTX_POL2	Antenna Polarity Control 2 Control the polarity of the Antenna or Switch control pins. 0 Do not invert the TXSWITCH output. 1 Invert the TXSWITCH output.
5 ANTX_POL1	Antenna Polarity Control 1 Control the polarity of the Antenna or Switch control pins. 0 Do not invert the ANT_B output. 1 Invert the ANT_B output.
4 ANTX_POL0	Antenna Polarity Control 0 Control the polarity of the Antenna or Switch control pins. 0 Do not invert the ANT_A output. 1 Invert the ANT_A output.
3 ANTX_ CTRLMODE	Antenna Control Mode Sets the single/dual mode. 0 Single mode: ANT_A=ANTX TX_SWITCH=TXON and RX_SWITCH=(RXON OR TXON). 1 Dual mode: ANT_A=ANTX AND (RXON OR TXON) ANT_B=NOT (ANTX) AND (RXON OR TXON) TX_SWITCH=TXON and RX_SWITCH=RXON.
2 ANTX_HZ	Antenna High Impedance Antenna controls high impedance. 0 ANT_A, ANT_B, RX_SWITCH, and TX_SWITCH are actively driven outputs. 1 Set ANT_A, ANT_B, RX_SWITCH and TX_SWITCH in high impedance.
ANTX_EN	Antenna Controls Enable Enable ANT_A, ANT_B, RX_SWITCH, and TX_SWITCH. 00 All disabled. 01 Only RX_SWITCH/TX_SWITCH enabled. 10 Only ANT_A/ANT_B enabled. 11 All enabled.

9.3.28 Miscellaneous Pad Control (Indirect_Modem_MISC_PAD_CTRL)

Address: 0h base + 31h offset = 31h

Bit	7	6	5	4
Read Write	Reserved			
Reset	0	0	0	1

Bit	3	2	1	0
Read Write	MISO_HIZ_EN	IRQ_B_OD	NON_GPIO_DS	ANTX_CURR
Reset	1	0	1	0

Indirect_Modem_MISC_PAD_CTRL field descriptions

Field	Description
7–4 Reserved	This field is reserved.
3 MISO_HIZ_EN	Determines the output state of the radio's R_MISO output when R_SSEL_B is deasserted (high). 0 R_MISO output is driven low by the SPI slave. 1 R_MISO output is tristated (default). An external pullup on R_MISO must be employed if MISO_HIZ_EN=1.
2 IRQ_B_OD	IRQ_B Open Drain 0 IRQ_B is an actively-driven output. 1 IRQ_B is open-drain (external pullup required).
1 NON_GPIO_DS	Determines the drive strength for the all of the following pins: {IRQ_B, R_MISO, DTM0 ... DTM6}. 0 All the listed pins use normal drive strength. 1 All the listed pins use high drive strength.
0 ANTX_CURR	Determines the drive strength for the all of the following pins: {ANT_A, ANT_B, RX_SWITCH, TX_SWITCH}. 0 All the listed pins use normal drive strength. 1 All the listed pins use high drive strength.

9.3.29 RX_BYTE_COUNT (Indirect_Modem_RX_BYTE_COUNT)

Address: 0h base + 35h offset = 35h

Bit	7	6	5	4	3	2	1	0
Read	RX_BYTE_COUNT							
Write								
Reset	0	0	0	0	0	0	0	0

Indirect_Modem_RX_BYTE_COUNT field descriptions

Field	Description
RX_BYTE_ COUNT	During packet reception, this read-only register is a real-time indicator of the number of bytes that have been received. This register will read 0 until SFD and PHR have been received. It will read 1 after the first byte of Frame Control Field has been received, etc.

9.3.30 RX_WTR_MARK (Indirect_Modem_RX_WTR_MARK)

Address: 0h base + 36h offset = 36h

Bit	7	6	5	4	3	2	1	0
Read Write	RX_WTR_MARK							
Reset	1	1	1	1	1	1	1	1

Indirect_Modem_RX_WTR_MARK field descriptions

Field	Description
RX_WTR_MARK	Receive byte count (octets) needed to trigger a RXWTRMRKIRQ interrupt . A setting of 0 generates an interrupt at end of the Frame Length field (first byte after SFD). A setting of 1 generates an interrupt after the first byte of Frame Control Field, etc.

9.3.31 TXDELAY (Indirect_Modem_TXDELAY)

Address: 0h base + 38h offset = 38h

Bit	7	6	5	4	3	2	1	0
Read Write	Reserved		TXDELAY					
Reset	0	0	0	0	0	0	0	0

Indirect_Modem_TXDELAY field descriptions

Field	Description
7–6 Reserved	This field is reserved. Reserved
TXDELAY	Provides a fine-tune adjustment of the time delay between post-CCA Rx warm-down and the beginning of Tx warm-up for an Tx(non-Ack) packet. TXDELAY register will apply in both SLOTTED and UNSLOTTED modes, but only to T sequences (e.g., T, TR, and T(R)), not TxAck operations. This is a two's complement value. The minimum permissible value is -19(0x2D). Values less than -19 will lead to unexpected results. Resolution = 2µs. Range = +/- 62µs. Max TXDELAY = 0x1F Min TXDELAY = 0x2D

9.3.32 ACKDELAY (Indirect_Modem_ACKDELAY)

Address: 0h base + 39h offset = 39h

Bit	7	6	5	4	3	2	1	0
Read Write	Reserved		ACKDELAY					
Reset	0	0	1	1	1	1	0	1

Indirect_Modem_ACKDELAY field descriptions

Field	Description
7–6 Reserved	This field is reserved. Reserved
ACKDELAY	Provides a fine-tune adjustment of the time delay between Rx warmdown and the beginning of Tx warmup for an Tx Acknowledge packet. ACKDELAY register will apply to both SLOTTED and UNSLOTTED TxAck, but only to TxAck (not T sequences). This is a two's complement value. The minimum permissible value is -19 (0x2D). Values less than -19 will lead to unexpected results. Values less than -19 will lead to unexpected results. Resolution = 2μs. Range = +/- 62μs. Max ACKDELAY = 0x1F Min ACKDELAY = 0x2D

9.3.33 Antenna AGC and FAD Control (Indirect_Modem_ANT_AGC_CTRL)

Address: 0h base + 51h offset = 51h

Bit	7	6	5	4	3	2	1	0
Read Write	SNF_EN	AGC_EN	AGC_LEVEL		Reserved		ANTX	FAD_EN
Reset	0	1	0	0	0	0	0	0

Indirect_Modem_ANT_AGC_CTRL field descriptions

Field	Description
7 SNF_EN	Sniff Enable Enable the sniff function that allows the symbol demodulator to be started when the RSSI level has crossed the defined SNF threshold.
6 AGC_EN	AGC Enable Enable the AGC function. 0 The AGC function is disabled. 1 The AGC function is enabled.
5–4 AGC_LEVEL	AGC Level If AGC_EN=0, forces the attenuation mode AGC_LEVEL_IN[1:0]. For reads:

Table continues on the next page...

Indirect_Modem_ANT_AGC_CTRL field descriptions (continued)

Field	Description
	AGC_EN=1: gives the AGC level chosen. AGC_EN=0: reads back what was last written to the register. 00 FE full gain, CHF full gain. 01 FE full gain, CHF low gain. 10 Reserved 11 FE low gain, CHF low gain.
3–2 Reserved	This field is reserved.
1 ANTX	ANTX bit selects the initial antenna state when FAD_EN=1, or allows software direct control over the antenna selection at all times when FAD_EN=0. 0 Select ANT_B 1 Select ANT_A
0 FAD_EN	FAD Enable Enable the FAD function. 0 FAD function is disabled. 1 FAD function is enabled.

9.3.34 LPPS_CTRL (Indirect_Modem_LPPS_CTRL)

Address: 0h base + 56h offset = 56h

Bit	7	6	5	4
Read Write	Reserved			LPPS_BUFMIX_EN
Reset	0	0	0	1

Bit	3	2	1	0
Read Write	LPPS_LIM_EN	LPPS_RSSI_EN	LPPS_LNA_EN	LPPS_EN
Reset	1	1	1	0

Indirect_Modem_LPPS_CTRL field descriptions

Field	Description
7–5 Reserved	This field is reserved.
4 LPPS_BUFMIX_ EN	VCO buffer enable during LPPS 0 Normal operation. 1 During preamble search in LPPS mode, the VCO buffer is turned on/off with a 50% DC (Duty Cycle).
3 LPPS_LIM_EN	LIM enable during LPPS 0 Normal operation. 1 During preamble search in LPPS mode, the limiter is turned on/off with a 50% DC (Duty Cycle).

Table continues on the next page...

Indirect_Modem_LPPS_CTRL field descriptions (continued)

Field	Description
2 LPPS_RSSI_EN	RSSI enable during LPPS 0 Normal operation. 1 During preamble search in LPPS mode, the RSSI is turned on/off with a 50% DC (Duty Cycle).
1 LPPS_LNA_EN	LNA enable during LPPS 0 Normal operation. 1 During preamble search in LPPS mode, the LNA is turned on/off with a 50% DC (Duty Cycle).
0 LPPS_EN	Master Enable for Low-Power Preamble Search (LPPS) mode 0 Normal operation. 1 Enable Low-Power Preamble Search mode. Note: LPPS mode should not be engaged if Fast Antenna Diversity mode is in use (FAN_EN=1).

9.3.35 RSSI (Indirect_Modem_RSSI)

Address: 0h base + 5Bh offset = 5Bh

Bit	7	6	5	4	3	2	1	0
Read	RSSI_ Level_7	RSSI_ Level_6	RSSI_ Level_5	RSSI_ Level_4	RSSI_ Level_3	RSSI_ Level_2	RSSI_ Level_1	RSSI_ Level_0
Write								
Reset	0	0	0	0	0	0	0	0

Indirect_Modem_RSSI field descriptions

Field	Description
7 RSSI_Level_7	RSSI Level, refreshed every 125 µs.
6 RSSI_Level_6	RSSI Level, refreshed every 125 µs.
5 RSSI_Level_5	RSSI Level, refreshed every 125 µs.
4 RSSI_Level_4	RSSI Level, refreshed every 125 µs.
3 RSSI_Level_3	RSSI Level, refreshed every 125 µs.
2 RSSI_Level_2	RSSI Level, refreshed every 125 µs.
1 RSSI_Level_1	RSSI Level, refreshed every 125 µs.
0 RSSI_Level_0	RSSI Level, refreshed every 125 µs.

9.3.36 XTAL Control (Indirect_Modem_XTAL_CTRL)

Address: 0h base + 6Eh offset = 6Eh

Bit	7	6	5	4	3	2	1	0
Read	Reserved		XTAL_BYPASS			Reserved		
Write	▨					▨		
Reset	0	0	0	0	0	0	1	1

Indirect_Modem_XTAL_CTRL field descriptions

Field	Description
7–6 Reserved	This field is reserved.
5–3 XTAL_BYPASS	XTAL Bypass Reset value = 000 Bypass value = 111 XTAL32m bypass for external injection. Note: XTALEN must be 0 when XTAL_BYPASS=1. See PWR_MODES register for XTALEN.
Reserved	This field is reserved.

Chapter 10
MCU: Chip Configuration

10.1 Introduction

This chapter provides details on the individual modules of the microcontroller. It includes:

- module block diagrams showing immediate connections within the device,
- specific module-to-module interactions not necessarily discussed in the individual module chapters, and
- links for more information.

10.2 Core modules

10.2.1 ARM Cortex-M4 Core Configuration

This section summarizes how the module has been configured in the chip. Full documentation for this module is provided by ARM and can be found at **arm.com**.

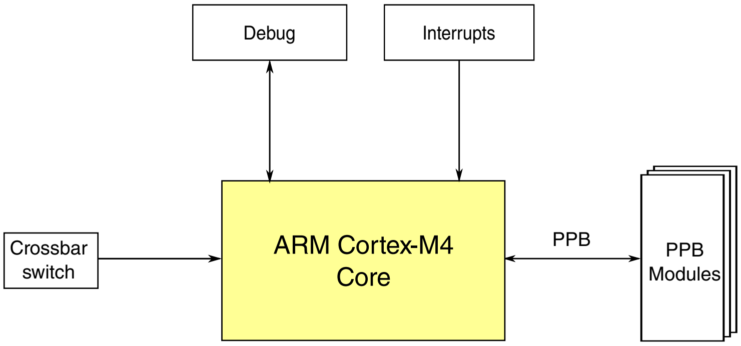


Figure 10-1. Core configuration

Table 10-1. Reference links to related information

Topic	Related module	Reference
Full description	ARM Cortex-M4 core	ARM Cortex-M4 Technical Reference Manual
System memory map		System memory map
Clocking		Clock distribution
Power management		Power management
System/instruction/data bus module	Crossbar switch	Crossbar switch
Debug	IEEE 1149.1 JTAG Serial Wire Debug (SWD) ARM Real-Time Trace Interface	Debug
Interrupts	Nested Vectored Interrupt Controller (NVIC)	NVIC
Private Peripheral Bus (PPB) module	Miscellaneous Control Module (MCM)	MCM
Private Peripheral Bus (PPB) module	Memory-Mapped Cryptographic Acceleration Unit (MMCAU)	MMCAU

10.2.1.1 Buses, interconnects, and interfaces

The ARM Cortex-M4 core has four buses as described in the following table.

Table 10-2. Buses of ARM Cortex-M4 core and their description

Bus name	Description
Instruction code (ICODE) bus	The ICODE and DCODE buses are muxed. This muxed bus is called the CODE bus and is connected to the crossbar switch via a single master port.
Data code (DCODE) bus	

Table continues on the next page...

Table 10-2. Buses of ARM Cortex-M4 core and their description (continued)

Bus name	Description
System bus	The system bus is connected to a separate master port on the crossbar.
Private peripheral (PPB) bus	The PPB provides access to these modules: • ARM modules such as the NVIC, ETM, ITM, DWT, FBP, and ROM table • Miscellaneous Control Module (MCM) • Memory-Mapped Cryptographic Acceleration Unit (MMCAU)

10.2.1.2 System Tick Timer

The System Tick Timer's clock source is always the core clock, FCLK. This results in the following:

- The CLKSOURCE bit in SysTick Control and Status register is always set to select the core clock.
- Because the timing reference (FCLK) is a variable frequency, the TENMS bit in the SysTick Calibration Value Register is always zero.
- The NOREF bit in SysTick Calibration Value Register is always set, implying that FCLK is the only available source of reference timing.

10.2.1.3 Debug facilities

This device has extensive debug capabilities including run control and tracing capabilities. The standard ARM debug port that supports JTAG and SWD interfaces. Also the cJTAG interface is supported on this device.

10.2.1.4 Core privilege levels

The ARM documentation uses different terms than this document to distinguish between privilege levels.

If you see this term...	it also means this term...
Privileged	Supervisor
Unprivileged or user	User

10.2.2 Nested Vectored Interrupt Controller (NVIC) Configuration

This section summarizes how the module has been configured in the chip. Full documentation for this module is provided by ARM and can be found at **arm.com**.

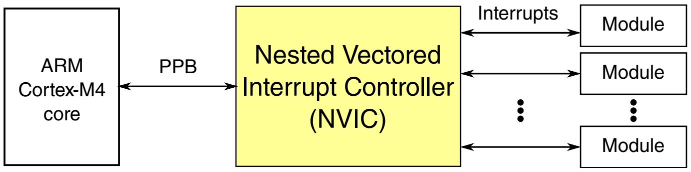


Figure 10-2. NVIC configuration

Table 10-3. Reference links to related information

Topic	Related module	Reference
Full description	Nested Vectored Interrupt Controller (NVIC)	ARM Cortex-M7 Technical Reference Manual
System memory map		System memory map
Clocking		Clock distribution
Power management		Power management
Private Peripheral Bus (PPB)	ARM Cortex-M4 core	ARM Cortex-M4 core

10.2.2.1 Interrupt priority levels

This device supports 16 priority levels for interrupts. Therefore, in the NVIC each source in the IPR registers contains 4 bits. For example, IPR0 is shown below:

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	IRQ3				0	0	0	0	IRQ2				0	0	0	0	IRQ1				0	0	0	0	IRQ0				0	0	0	0
W																																

10.2.2.2 Non-maskable interrupt

The non-maskable interrupt request to the NVIC is controlled by the external $\overline{\text{NMI}}$ signal. The pin the $\overline{\text{NMI}}$ signal is multiplexed on, must be configured for the $\overline{\text{NMI}}$ function to generate the non-maskable interrupt request.

10.2.2.3 Interrupt channel assignments

The interrupt source assignments are defined in the following table.

- Vector number — the value stored on the stack when an interrupt is serviced.
- IRQ number — non-core interrupt source count, which is the vector number minus 16.

The IRQ number is used within ARM's NVIC documentation.

Table 10-5. Interrupt vector assignments

Address	Vector	IRQ[1]	NVIC non-IPR register number [2]	NVIC IPR register number [3]	Source module	Source description
ARM Core System Handler Vectors						
0x0000_0000	0	–	–	–	ARM core	Initial Stack Pointer
0x0000_0004	1	–	–	–	ARM core	Initial Program Counter
0x0000_0008	2	–	–	–	ARM core	Non-maskable Interrupt (NMI)
0x0000_000C	3	–	–	–	ARM core	Hard Fault
0x0000_0010	4	–	–	–	ARM core	MemManage Fault
0x0000_0014	5	–	–	–	ARM core	Bus Fault
0x0000_0018	6	–	–	–	ARM core	Usage Fault
0x0000_001C	7	–	–	–	—	—
0x0000_0020	8	–	–	–	—	—
0x0000_0024	9	–	–	–	—	—
0x0000_0028	10	–	–	–	—	—
0x0000_002C	11	–	–	–	ARM core	Supervisor call (SVCall)
0x0000_0030	12	–	–	–	ARM core	Debug Monitor
0x0000_0034	13	–	–	–	—	—
0x0000_0038	14	–	–	–	ARM core	Pendable request for system service (PendableSrvReq)
0x0000_003C	15	–	–	–	ARM core	System tick timer (SysTick)
Non-Core Vectors						
0x0000_0040	16	0	0	0	DMA	DMA channel 0 transfer complete
0x0000_0044	17	1	0	0	DMA	DMA channel 1 transfer complete
0x0000_0048	18	2	0	0	DMA	DMA channel 2 transfer complete
0x0000_004C	19	3	0	0	DMA	DMA channel 3 transfer complete
0x0000_0050	20	4	0	1	DMA	DMA channel 4 transfer complete
0x0000_0054	21	5	0	1	DMA	DMA channel 5 transfer complete
0x0000_0058	22	6	0	1	DMA	DMA channel 6 transfer complete
0x0000_005C	23	7	0	1	DMA	DMA channel 7 transfer complete
0x0000_0060	24	8	0	2	DMA	DMA channel 8 transfer complete

Table continues on the next page...

Table 10-5. Interrupt vector assignments (continued)

Address	Vector	IRQ[1]	NVIC non-IPR register number [2]	NVIC IPR register number [3]	Source module	Source description
0x0000_0064	25	9	0	2	DMA	DMA channel 9 transfer complete
0x0000_0068	26	10	0	2	DMA	DMA channel 10 transfer complete
0x0000_006C	27	11	0	2	DMA	DMA channel 11 transfer complete
0x0000_0070	28	12	0	3	DMA	DMA channel 12 transfer complete
0x0000_0074	29	13	0	3	DMA	DMA channel 13 transfer complete
0x0000_0078	30	14	0	3	DMA	DMA channel 14 transfer complete
0x0000_007C	31	15	0	3	DMA	DMA channel 15 transfer complete
0x0000_0080	32	16	0	4	DMA	DMA error interrupt channels 0-15
0x0000_0084	33	17	0	4	MCM	—
0x0000_0088	34	18	0	4	Flash memory	Command complete
0x0000_008C	35	19	0	4	Flash memory	Read collision
0x0000_0090	36	20	0	5	Mode Controller	Low-voltage detect, low-voltage warning
0x0000_0094	37	21	0	5	LLWU	Low Leakage Wakeup **NOTE:** The LLWU interrupt must not be masked by the interrupt controller to avoid a scenario where the system does not fully exit stop mode on an LLS recovery.
0x0000_0098	38	22	0	5	WDOG or EWM	Both watchdog modules share this interrupt
0x0000_009C	39	23	0	5	RNG	Random Number Generator
0x0000_00A0	40	24	0	6	I^{2}C0	—
0x0000_00A4	41	25	0	6	I^{2}C1	—
0x0000_00A8	42	26	0	6	SPI0	Single interrupt vector for all sources
0x0000_00AC	43	27	0	6	SPI1	Single interrupt vector for all sources
0x0000_00B0	44	28	0	7	I^{2}S0	Transmit
0x0000_00B4	45	29	0	7	I^{2}S0	Receive
0x0000_00B8	46	30	0	7	—	—
0x0000_00BC	47	31	0	7	UART0	Single interrupt vector for UART status sources
0x0000_00C0	48	32	1	8	UART0	Single interrupt vector for UART error sources
0x0000_00C4	49	33	1	8	UART1	Single interrupt vector for UART status sources
0x0000_00C8	50	34	1	8	UART1	Single interrupt vector for UART error sources
0x0000_00CC	51	35	1	8	UART2	Single interrupt vector for UART status sources

Table continues on the next page...

Table 10-5. Interrupt vector assignments (continued)

Address	Vector	IRQ[1]	NVIC non-IPR register number [2]	NVIC IPR register number [3]	Source module	Source description
0x0000_00D0	52	36	1	9	UART2	Single interrupt vector for UART error sources
0x0000_00D4	53	37	1	9	—	—
0x0000_00D8	54	38	1	9	—	—
0x0000_00DC	55	39	1	9	ADC0	—
0x0000_00E0	56	40	1	10	CMP0	—
0x0000_00E4	57	41	1	10	CMP1	—
0x0000_00E8	58	42	1	10	FTM0	Single interrupt vector for all sources
0x0000_00EC	59	43	1	10	FTM1	Single interrupt vector for all sources
0x0000_00F0	60	44	1	11	FTM2	Single interrupt vector for all sources
0x0000_00F4	61	45	1	11	CMT	—
0x0000_00F8	62	46	1	11	RTC	Alarm interrupt
0x0000_00FC	63	47	1	11	RTC	Seconds interrupt
0x0000_0100	64	48	1	12	PIT	Channel 0
0x0000_0104	65	49	1	12	PIT	Channel 1
0x0000_0108	66	50	1	12	PIT	Channel 2
0x0000_010C	67	51	1	12	PIT	Channel 3
0x0000_0110	68	52	1	13	PDB	—
0x0000_0114	69	53	1	13	USB OTG	—
0x0000_0118	70	54	1	13	USB Charger Detect	—
0x0000_011C	71	55	1	13	—	—
0x0000_0120	72	56	1	14	—	—
0x0000_0124	73	57	1	14	MCG	—
0x0000_0128	74	58	1	14	Low Power Timer	—
0x0000_012C	75	59	1	14	Port control module	Pin detect (Port A)
0x0000_0130	76	60	1	15	Port control module	Pin detect (Port B)
0x0000_0134	77	61	1	15	Port control module	Pin detect (Port C)
0x0000_0138	78	62	1	15	Port control module	Pin detect (Port D)
0x0000_013C	79	63	1	15	Port control module	Pin detect (Port E)
0x0000_0140	80	64	2	16	Software	Software interrupt[4]

1. Indicates the NVIC's interrupt source number.
2. Indicates the NVIC's ISER, ICER, ISPR, ICPR, and IABR register number used for this IRQ. The equation to calculate this value is: IRQ div 32
3. Indicates the NVIC's IPR register number used for this IRQ. The equation to calculate this value is: IRQ div 4
4. This interrupt can only be pended or cleared via the NVIC registers.

10.2.2.3.1 Determining the bitfield and register location for configuring a particular interrupt

Suppose you need to configure the low-power timer (LPTMR) interrupt. The following table is an excerpt of the LPTMR row from Interrupt channel assignments.

Table 10-6. LPTMR interrupt vector assignment

Address	Vector	IRQ[1]	NVIC non-IPR register number [2]	NVIC IPR register number [3]	Source module	Source description
0x0000_0128	74	58	1	14	Low Power Timer	—

[1] Indicates the NVIC's interrupt source number.

[2] Indicates the NVIC's ISER, ICER, ISPR, ICPR, and IABR register number used for this IRQ. The equation to calculate this value is: IRQ div 32

[3] Indicates the NVIC's IPR register number used for this IRQ. The equation to calculate this value is: IRQ div 4

- The NVIC registers you would use to configure the interrupt are:
  - NVICISER1
  - NVICICER1
  - NVICISPR1
  - NVICICPR1
  - NVICIABR1
  - NVICIPR14
- To determine the particular IRQ's bitfield location within these particular registers:
  - NVICISER1, NVICICER1, NVICISPR1, NVICICPR1, NVICIABR1 bit location = IRQ mod 32 = 26
  - NVICIPR14 bitfield starting location = 8 * (IRQ mod 4) + 4 = 20

    Since the NVICIPR bitfields are 4-bit wide (16 priority levels), the NVICIPR14 bitfield range is 20-23

Therefore, the following bitfield locations are used to configure the LPTMR interrupts:

- NVICISER1[26]
- NVICICER1[26]
- NVICISPR1[26]
- NVICICPR1[26]
- NVICIABR1[26]
- NVICIPR14[23:20]

10.2.3 Asynchronous Wake-up Interrupt Controller (AWIC) Configuration

This section summarizes how the module has been configured in the chip. Full documentation for this module is provided by ARM and can be found at **arm.com**.

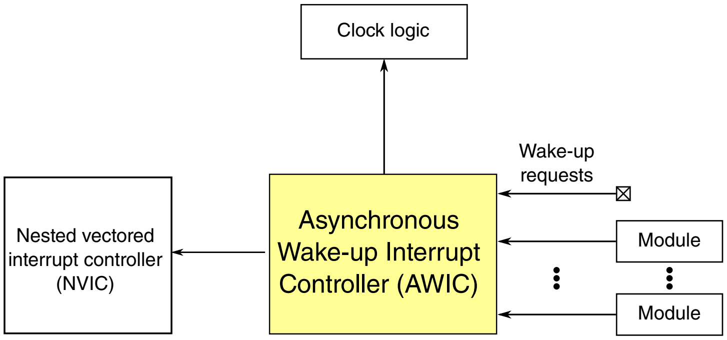

Clock logic
Wake-up requests
Nested vectored interrupt controller (NVIC)
Asynchronous Wake-up Interrupt Controller (AWIC)
Module
Module

Figure 10-3. Asynchronous Wake-up Interrupt Controller configuration

Table 10-7. Reference links to related information

Topic	Related module	Reference
System memory map		System memory map
Clocking		Clock distribution
Power management		Power management
	Nested Vectored Interrupt Controller (NVIC)	NVIC
Wake-up requests		AWIC wake-up sources

10.2.3.1 Wake-up sources

The device uses the following internal and external inputs to the AWIC module.

Table 10-8. AWIC Stop and VLPS Wake-up Sources

Wake-up source	Description
Available system resets	$\overline{\text{RESET}}$ pin and WDOG when LPO is its clock source, and JTAG
Low-voltage detect	Mode Controller
Low-voltage warning	Mode Controller
Pin interrupts	Port Control Module - Any enabled pin interrupt is capable of waking the system
ADCx	The ADC is functional when using internal clock source
CMPx	Since no system clocks are available, functionality is limited
I^2C	Address match wakeup
UART	Active edge on RXD

Table continues on the next page...

Table 10-8. AWIC Stop and VLPS Wake-up Sources (continued)

Wake-up source	Description
USB FS/LS Controller	Wakeup
LPTMR	Functional when using clock source which is active in Stop and VLPS modes
RTC	Functional in Stop/VLPS modes
I2S (SAI)	Functional when using an external bit clock or external master clock
Tamper detect	Interrupt or a reset
NMI	Non-maskable interrupt

10.2.4 JTAG Controller Configuration

This section summarizes how the module has been configured in the chip. For a comprehensive description of the module itself, see the module's dedicated chapter.

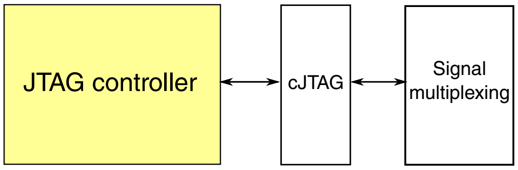


Figure 10-4. JTAGC Controller configuration

Table 10-9. Reference links to related information

Topic	Related module	Reference
Full description	JTAGC	JTAGC
Signal multiplexing	Port control	Signal multiplexing

10.3 System modules

10.3.1 SIM Configuration

This section summarizes how the module has been configured in the chip. For a comprehensive description of the module itself, see the module's dedicated chapter.

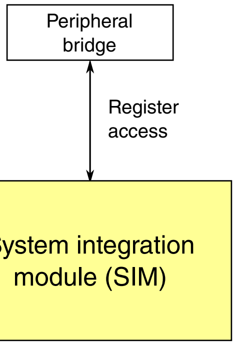


Figure 10-5. SIM configuration

Table 10-10. Reference links to related information

Topic	Related module	Reference
Full description	SIM	SIM
System memory map		System memory map
Clocking		Clock distribution
Power management		Power management

10.3.2 System Mode Controller (SMC) Configuration

This section summarizes how the module has been configured in the chip. For a comprehensive description of the module itself, see the module's dedicated chapter.

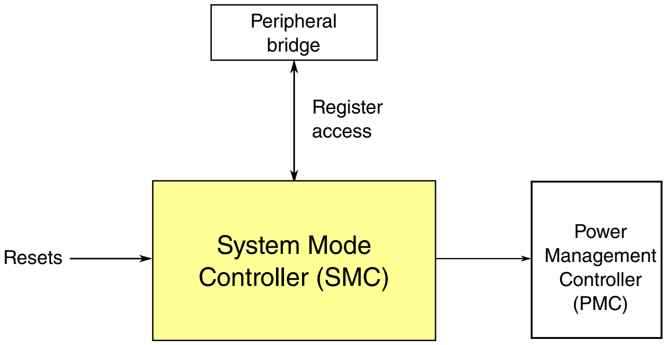

Peripheral bridge
Register access
Resets
System Mode Controller (SMC)
Power Management Controller (PMC)

Figure 10-6. System Mode Controller configuration

Table 10-11. Reference links to related information

Topic	Related module	Reference
Full description	System Mode Controller (SMC)	SMC
System memory map		System memory map
Power management		Power management

Table continues on the next page...

Table 10-11. Reference links to related information (continued)

Topic	Related module	Reference
	Power management controller (PMC)	PMC
	Low-Leakage Wakeup Unit (LLWU)	LLWU
	Reset Control Module (RCM)	Reset

10.3.3 PMC Configuration

This section summarizes how the module has been configured in the chip. For a comprehensive description of the module itself, see the module's dedicated chapter.

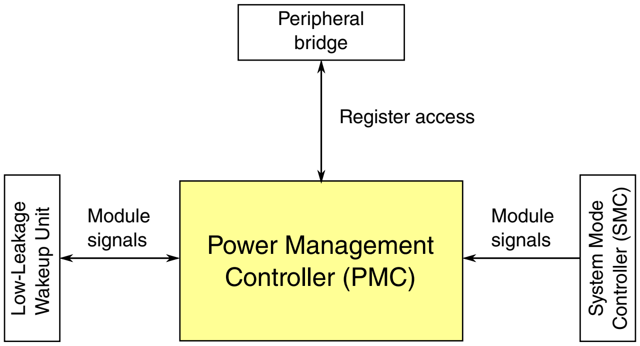


Figure 10-7. PMC configuration

Table 10-12. Reference links to related information

Topic	Related module	Reference
Full description	PMC	PMC
System memory map		System memory map
Power management		Power management
Full description	System Mode Controller (SMC)	System Mode Controller
	Low-Leakage Wakeup Unit (LLWU)	LLWU
	Reset Control Module (RCM)	Reset

10.3.4 Low-Leakage Wake-up Unit (LLWU) Configuration

This section summarizes how the module has been configured in the chip. For a comprehensive description of the module itself, see the module’s dedicated chapter.

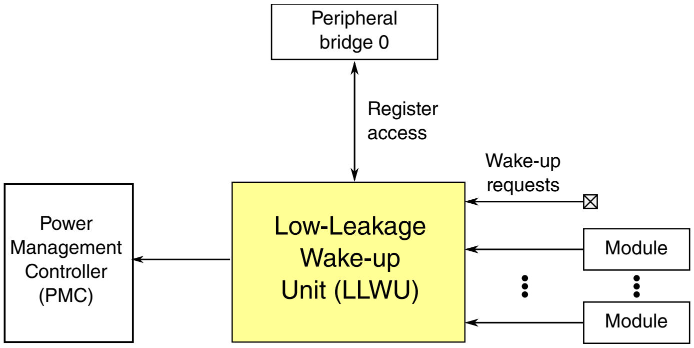

Peripheral bridge 0
Register access
Wake-up requests
Power Management Controller (PMC)
Low-Leakage Wake-up Unit (LLWU)
Module
Module

Figure 10-8. Low-Leakage Wake-up Unit configuration

Table 10-13. Reference links to related information

Topic	Related module	Reference
Full description	LLWU	LLWU
System memory map		System memory map
Clocking		Clock distribution
Power management		Power management chapter
	Power Management Controller (PMC)	Power Management Controller (PMC)
	Mode Controller	Mode Controller
Wake-up requests		LLWU wake-up sources

10.3.4.1 Wake-up Sources

This chip uses the following internal peripheral and external pin inputs as wakeup sources to the LLWU module:

- LLWU_P0-15 are external pin inputs. Any digital function multiplexed on the pin can be selected as the wakeup source. See the chip's signal multiplexing table for the digital signal options.
- LLWU_M0IF-M7IF are connections to the internal peripheral interrupt flags.

NOTE

$\overline{\text{RESET}}$ is also a wakeup source, depending on the bit setting in the LLWU_RST register. On devices where $\overline{\text{RESET}}$ is not a dedicated pin, it must also be enabled in the explicit port mux control.

Table 10-14. Wakeup sources for LLWU inputs

Input	Wakeup source		Input	Wakeup source
LLWU_P0	PTE1/LLWU_P0 pin		LLWU_P12	PTD0/LLWU_P12 pin
LLWU_P1	PTE2/LLWU_P1 pin		LLWU_P13	PTD2/LLWU_P13 pin
LLWU_P2	PTE4/LLWU_P2 pin		LLWU_P14	PTD4/LLWU_P14 pin
LLWU_P3	PTA4/LLWU_P3 pin [1]		LLWU_P15	PTD6/LLWU_P15 pin
LLWU_P4	PTA13/LLWU_P4 pin		LLWU_M0IF	LPTMR
LLWU_P5	PTB0/LLWU_P5 pin		LLWU_M1IF	CMP0[2]
LLWU_P6	PTC1/LLWU_P6 pin		LLWU_M2IF	CMP1[2]
LLWU_P7	PTC3/LLWU_P7 pin		LLWU_M3IF	Reserved
LLWU_P8	PTC4/LLWU_P8 pin		LLWU_M4IF	Reserved
LLWU_P9	PTC5/LLWU_P9 pin		LLWU_M5IF	RTC Alarm[2]
LLWU_P10	PTC6/LLWU_P10 pin		LLWU_M6IF	Reserved
LLWU_P11	PTC11/LLWU_P11 pin		LLWU_M7IF	RTC Seconds[2]

[1] The $\overline{\text{EZP\_CS}}$ signal is checked only on *Chip Reset not VLLS*, so a VLLS wakeup via a non-reset source does not cause EzPort mode entry. If NMI was enabled on entry to LLS/VLLS, asserting the NMI pin generates an NMI interrupt on exit from the low power mode.

[2] Requires the peripheral and the peripheral interrupt to be enabled. The LLWU's WUME bit enables the internal module flag as a wakeup input. After wakeup, the flags are cleared based on the peripheral clearing mechanism.

10.3.5 MCM Configuration

This section summarizes how the module has been configured in the chip. For a comprehensive description of the module itself, see the module's dedicated chapter.

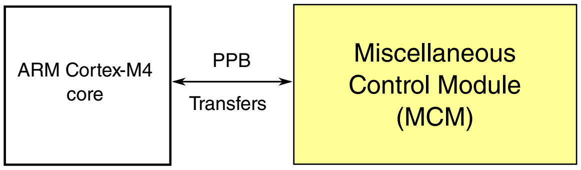


Figure 10-9. MCM configuration

Table 10-15. Reference links to related information

Topic	Related module	Reference
Full description	Miscellaneous control module (MCM)	MCM
System memory map		System memory map
Clocking		Clock distribution
Power management		Power management
Transfers Private Peripheral Bus (PPB)	ARM Cortex-M4 core	ARM Cortex-M4 core

10.3.6 Crossbar-Light Switch Configuration

This section summarizes how the module has been configured in the chip. For a comprehensive description of the module itself, see the module's dedicated chapter.

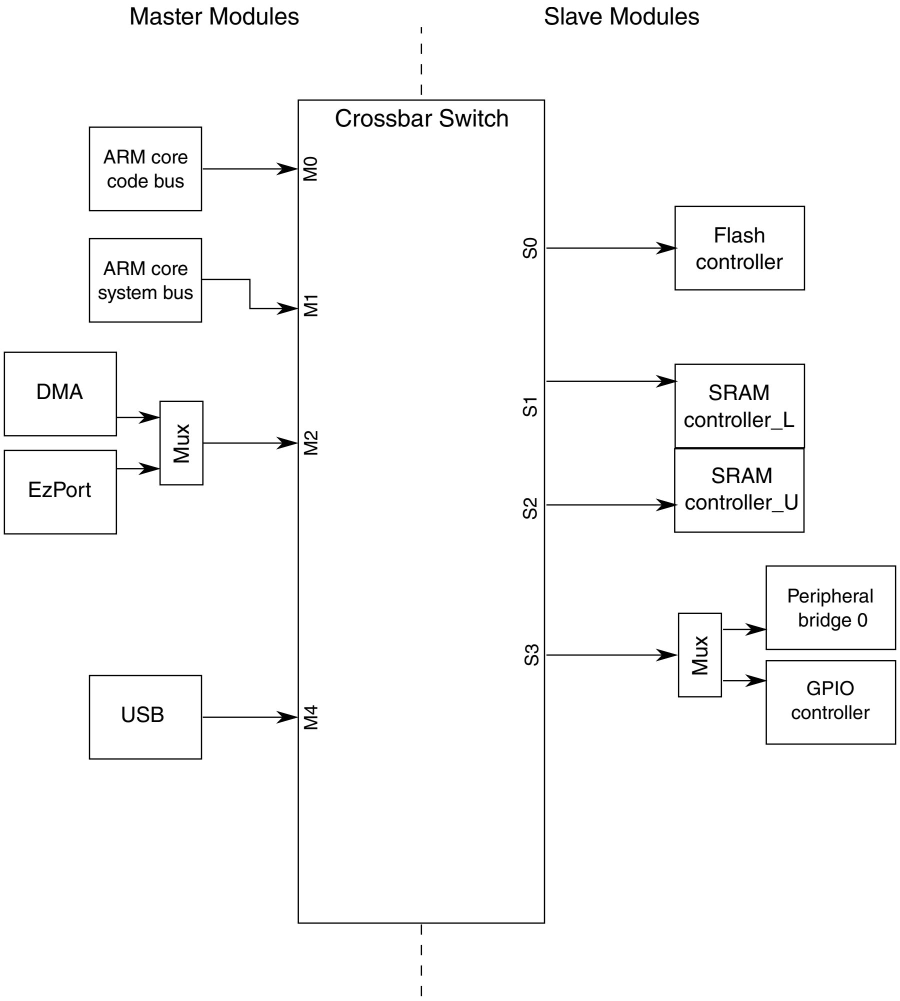


Figure 10-10. Crossbar-Light switch integration

Table 10-16. Reference links to related information

Topic	Related module	Reference
Full description	Crossbar switch	Crossbar Switch
System memory map		System memory map
Clocking		Clock Distribution
Crossbar switch master	ARM Cortex-M4 core	ARM Cortex-M4 core
Crossbar switch master	DMA controller	DMA controller
Crossbar switch master	EzPort	EzPort

Table continues on the next page...

Table 10-16. Reference links to related information (continued)

Topic	Related module	Reference
Crossbar switch master	USB FS/LS	USB FS/LS
Crossbar switch slave	Flash	Flash
Crossbar switch slaves	SRAM controllers	SRAM configuration
Crossbar switch slave	Peripheral bridges	Peripheral bridge
Crossbar switch slave	GPIO controller	GPIO controller

10.3.6.1 Crossbar-Light Switch Master Assignments

The masters connected to the crossbar switch are assigned as follows:

Master module	Master port number
ARM core code bus	0
ARM core system bus	1
DMA/EzPort	2
USB FS/LS OTG	4

NOTE

The DMA and EzPort share a master port. Since these modules never operate at the same time, no configuration or arbitration explanations are necessary.

The crossbar switch comes out of reset with a fixed priority. The priority from highest to lowest is masters M0 -> M1 -> M2 -> M3. M4-M7 are unused.

10.3.6.2 Crossbar-Light Switch Slave Assignments

The slaves connected to the crossbar switch are assigned as follows:

Slave module	Slave port number
Flash memory controller	0
SRAM controllers	1,2
Peripheral bridge 0/GPIO	3

10.3.7 Peripheral Bridge Configuration

This section summarizes how the module has been configured in the chip. For a comprehensive description of the module itself, see the module's dedicated chapter.

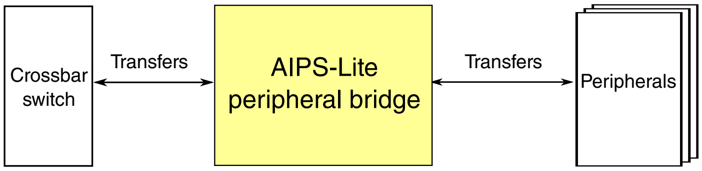


Figure 10-11. Peripheral bridge configuration

Table 10-17. Reference links to related information

Topic	Related module	Reference
Full description	Peripheral bridge (AIPS-Lite)	Peripheral bridge (AIPS-Lite)
System memory map		System memory map
Clocking		Clock Distribution
Crossbar switch	Crossbar switch	Crossbar switch

10.3.7.1 Number of peripheral bridges

This device contains one peripheral bridge.

10.3.7.2 Memory maps

The peripheral bridges are used to access the registers of most of the modules on this device. See AIPS0 Memory Map for the memory slot assignment for each module.

10.3.8 DMA request multiplexer configuration

This section summarizes how the module has been configured in the chip. For a comprehensive description of the module itself, see the module's dedicated chapter.

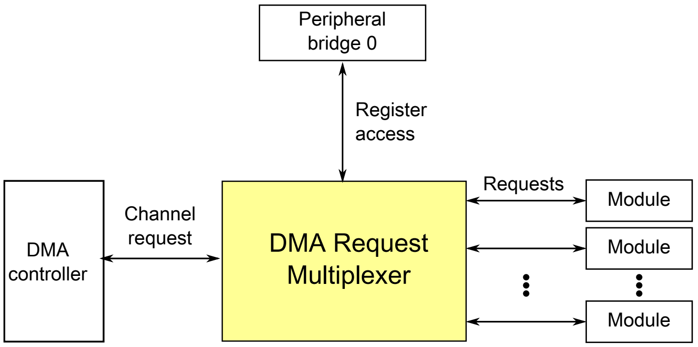


Figure 10-12. DMA request multiplexer configuration

Table 10-18. Reference links to related information

Topic	Related module	Reference
Full description	DMA request multiplexer	DMA Mux
System memory map		System memory map
Clocking		Clock distribution
Power management		Power management
Channel request	DMA controller	DMA Controller
Requests		DMA request sources

10.3.8.1 DMA MUX request sources

Table 10-19. DMA request sources - MUX 0

Source number	Source module	Source description
0	—	Channel disabled[1]
1	Reserved	Not used
2	UART0	Receive
3	UART0	Transmit
4	UART1	Receive
5	UART1	Transmit
6	UART2	Receive
7	UART2	Transmit
8	Reserved	Receive
9	Reserved	Transmit
10	Reserved	Receive
11	Reserved	Transmit
12	Reserved	Receive
13	Reserved	Transmit

Table continues on the next page...

Table 10-19. DMA request sources - MUX 0 (continued)

Source number	Source module	Source description
14	I^2S0	Receive
15	I^2S0	Transmit
16	SPI0	Receive
17	SPI0	Transmit
18	SPI1	Receive
19	SPI1	Transmit
20	—	—
21	—	—
22	I^2C0	—
23	I^2C1	—
24	FTM0	Channel 0
25	FTM0	Channel 1
26	FTM0	Channel 2
27	FTM0	Channel 3
28	FTM0	Channel 4
29	FTM0	Channel 5
30	FTM0	Channel 6
31	FTM0	Channel 7
32	FTM1	Channel 0
33	FTM1	Channel 1
34	FTM2	Channel 0
35	FTM2	Channel 1
36	Reserved	—
37	Reserved	—
38	Reserved	—
39	Reserved	—
40	ADC0	—
41	Reserved	—
42	CMP0	—
43	CMP1	—
44	Reserved	—
45	Reserved	—
46	Reserved	—
47	CMT	—
48	PDB	—
49	Port control module	Port A
50	Port control module	Port B
51	Port control module	Port C
52	Port control module	Port D

Table continues on the next page...

Table 10-19. DMA request sources - MUX 0 (continued)

Source number	Source module	Source description
53	Port control module	Port E
54	DMA MUX	Always enabled
55	DMA MUX	Always enabled
56	DMA MUX	Always enabled
57	DMA MUX	Always enabled
58	DMA MUX	Always enabled
59	DMA MUX	Always enabled
60	DMA MUX	Always enabled
61	DMA MUX	Always enabled
62	DMA MUX	Always enabled
63	DMA MUX	Always enabled

1. Configuring a DMA channel to select source 0 or any of the reserved sources disables that DMA channel.

10.3.8.2 DMA transfers via PIT trigger

The PIT module can trigger a DMA transfer on the first four DMA channels. The assignments are detailed at PIT/DMA Periodic Trigger Assignments .

10.3.9 DMA Controller Configuration

This section summarizes how the module has been configured in the chip. For a comprehensive description of the module itself, see the module’s dedicated chapter.

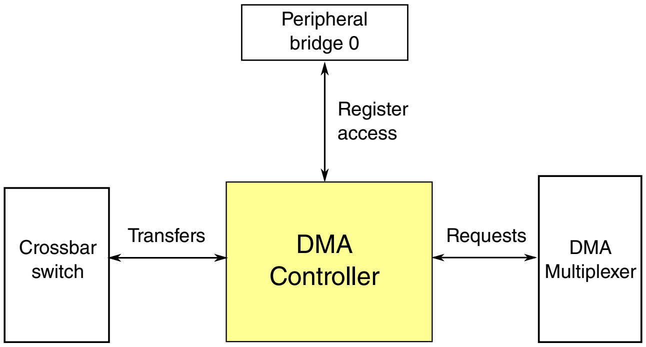


Figure 10-13. DMA Controller configuration

Table 10-20. Reference links to related information

Topic	Related module	Reference
Full description	DMA Controller	DMA Controller
System memory map		System memory map
Register access	Peripheral bridge (AIPS-Lite 0)	AIPS-Lite 0
Clocking		Clock distribution
Power management		Power management
Transfers	Crossbar switch	Crossbar switch

10.3.10 External Watchdog Monitor (EWM) Configuration

This section summarizes how the module has been configured in the chip. For a comprehensive description of the module itself, see the module’s dedicated chapter.

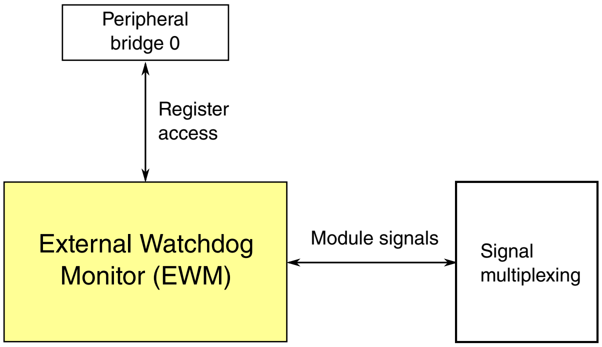


Figure 10-14. External Watchdog Monitor configuration

Table 10-21. Reference links to related information

Topic	Related module	Reference
Full description	External Watchdog Monitor (EWM)	EWM
System memory map		System memory map
Clocking		Clock distribution
Power management		Power management
Signal multiplexing	Port Control Module	Signal multiplexing

10.3.10.1 EWM clocks

This table shows the EWM clocks and the corresponding chip clocks.

Table 10-22. EWM clock connections

Module clock	Chip clock
Low Power Clock	1 kHz LPO Clock

10.3.10.2 EWM low-power modes

This table shows the EWM low-power modes and the corresponding chip low-power modes.

Table 10-23. EWM low-power modes

Module mode	Chip mode
Stop	Wait, VLPW
Stop	Stop, VLPS, LLS

10.3.10.3 $\overline{\text{EWM\_OUT}}$ pin state in low power modes

During Wait, Stop, and Power Down modes the $\overline{\text{EWM\_OUT}}$ pin preserve its state before entering Wait or Stop mode. When the CPU enters a Run mode from Wait or Stop recovery, the pin resumes its previous state before entering Wait or Stop mode. When the CPU enters Run mode from Power Down, the pin returns to its reset state.

10.3.11 Watchdog Configuration

This section summarizes how the module has been configured in the chip. For a comprehensive description of the module itself, see the module's dedicated chapter.

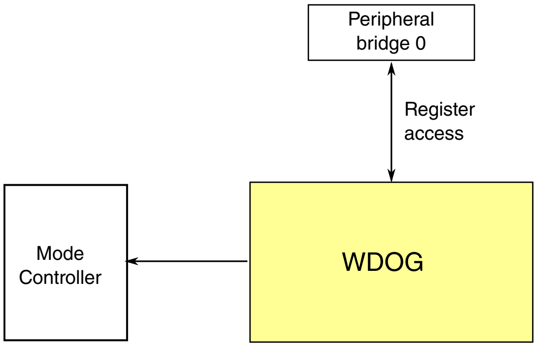


Figure 10-15. Watchdog configuration

Table 10-24. Reference links to related information

Topic	Related module	Reference
Full description	Watchdog	Watchdog
System memory map		System memory map
Clocking		Clock distribution
Power management		Power management
	Mode Controller (MC)	System Mode Controller

10.3.11.1 WDOG clocks

This table shows the WDOG module clocks and the corresponding chip clocks.

Table 10-25. WDOG clock connections

Module clock	Chip clock
LPO Oscillator	1 kHz LPO Clock
Alt Clock	Bus Clock
Fast Test Clock	Bus Clock
System Bus Clock	Bus Clock

10.3.11.2 WDOG low-power modes

This table shows the WDOG low-power modes and the corresponding chip low-power modes.

Table 10-26. WDOG low-power modes

Module mode	Chip mode
Wait	Wait, VLPW
Stop	Stop, VLPS
Power Down	LLS(static retained), VLLSx(Power Off)

10.4 Clock modules

10.4.1 MCG Configuration

This section summarizes how the module has been configured in the chip. For a comprehensive description of the module itself, see the module's dedicated chapter.

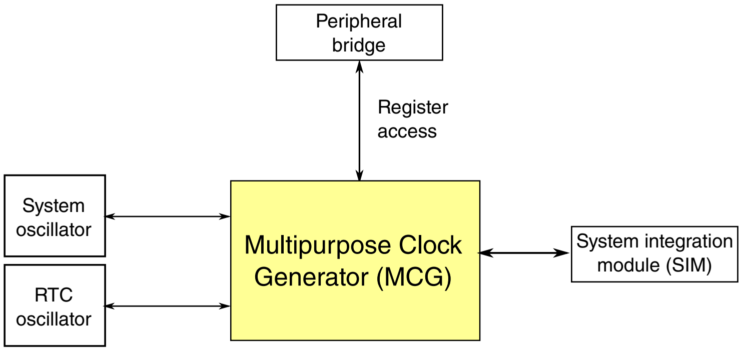

Peripheral bridge
Register access
System oscillator
RTC oscillator
Multipurpose Clock Generator (MCG)
System integration module (SIM)

Figure 10-16. MCG configuration

Table 10-27. Reference links to related information

Topic	Related module	Reference
Full description	MCG	MCG
System memory map		System memory map
Clocking		Clock distribution
Power management		Power management
Signal multiplexing	Port control	Signal multiplexing

10.4.2 OSC Configuration

This section summarizes how the module has been configured in the chip. For a comprehensive description of the module itself, see the module's dedicated chapter.

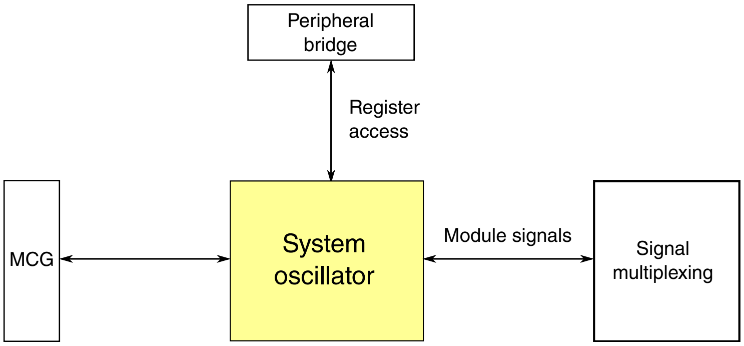


Figure 10-17. OSC configuration

Table 10-28. Reference links to related information

Topic	Related module	Reference
Full description	OSC	OSC
System memory map		System memory map
Clocking		Clock distribution

Table continues on the next page...

Table 10-28. Reference links to related information (continued)

Topic	Related module	Reference
Power management		Power management
Signal multiplexing	Port control	Signal multiplexing
Full description	MCG	MCG

10.4.2.1 OSC modes of operation with MCG

The MCG's C2 register bits configure the oscillator frequency range. See the OSC and MCG chapters for more details.

10.4.3 RTC OSC configuration

This section summarizes how the module has been configured in the chip. For a comprehensive description of the module itself, see the module’s dedicated chapter.

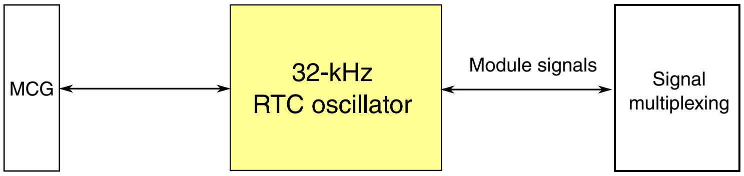


Figure 10-18. RTC OSC configuration

Table 10-29. Reference links to related information

Topic	Related module	Reference
Full description	RTC OSC	RTC OSC
Signal multiplexing	Port control	Signal multiplexing
Full description	MCG	MCG

10.5 Memories and memory interfaces

10.5.1 Flash Memory Configuration

This section summarizes how the module has been configured in the chip. For a comprehensive description of the module itself, see the module’s dedicated chapter.

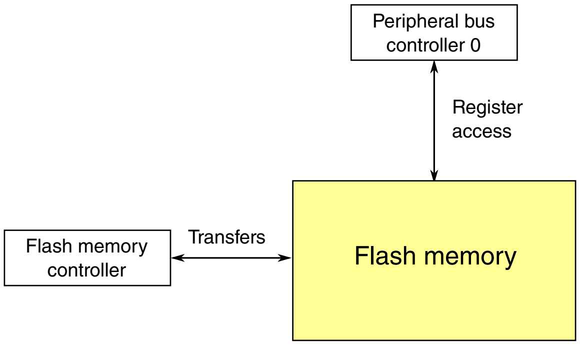


Figure 10-19. Flash memory configuration

Table 10-30. Reference links to related information

Topic	Related module	Reference
Full description	Flash memory	Flash memory
System memory map		System memory map
Clocking		Clock Distribution
Transfers	Flash memory controller	Flash memory controller
Register access	Peripheral bridge	Peripheral bridge

10.5.1.1 Flash memory types

This device contains the following types of flash memory:

- Program flash memory — non-volatile flash memory that can execute program code
- FlexMemory — encompasses the following memory types:
    - For devices with FlexNVM: FlexNVM — Non-volatile flash memory that can execute program code, store data, or backup EEPROM data
    - For devices with FlexNVM: FlexRAM — RAM memory that can be used as traditional RAM or as high-endurance EEPROM storage, and also accelerates flash programming
    - For devices with only program flash memory: Programming acceleration RAM — RAM memory that accelerates flash programming

10.5.1.2 Flash Memory Size Considerations

Since this document covers devices that contain program flash only and devices that contain program flash and FlexNVM, there are some items to consider when reading the flash memory chapter.

- The flash memory chapter shows a mixture of information depending on the device you are using.
- For the program flash only devices:
  - The programming acceleration RAM is used for the Program Section command.
- For the devices containing program flash and FlexNVM:
  - Since there is only one program flash block, the program flash swap feature is not available.

10.5.1.3 Flash Memory Map

The various flash memories and the flash registers are located at different base addresses as shown in the following figure. The base address for each is specified in System memory map.

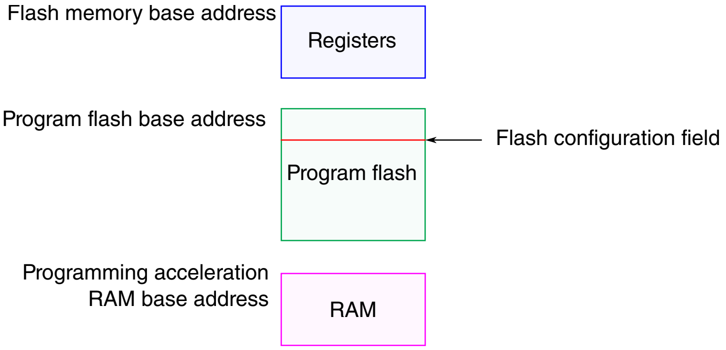

Flash memory base address
Registers
Program flash base address
Flash configuration field
Program flash
Programming acceleration RAM base address
RAM

Figure 10-20. Flash memory map for devices containing only program flash

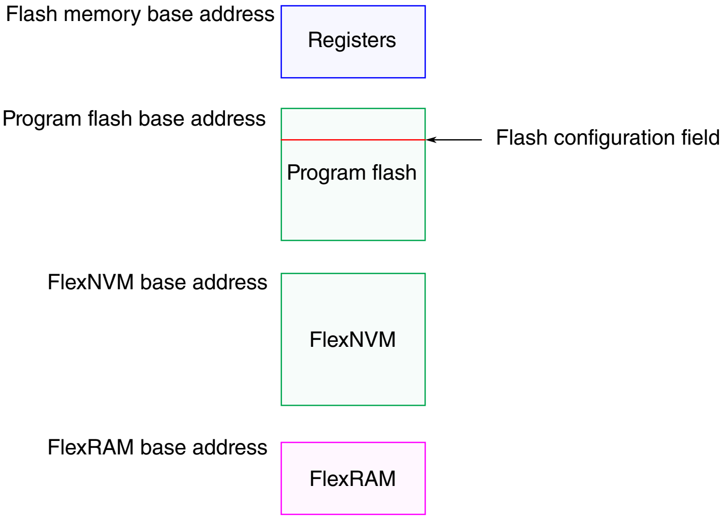

Flash memory base address
Registers
Program flash base address
Flash configuration field
Program flash
FlexNVM base address
FlexNVM
FlexRAM base address
FlexRAM

Figure 10-21. Flash memory map for devices containing FlexNVM

10.5.1.4 Flash Security

How flash security is implemented on this device is described in Chip Security.

10.5.1.5 Flash Modes

The flash memory operates in NVM normal and NVM special modes. The flash memory enters NVM special mode when the EzPort is enabled ($\overline{\text{EZP\_CS}}$ asserted during reset). Otherwise, flash memory operates in NVM normal mode.

10.5.1.6 Erase All Flash Contents

An Erase All Flash Blocks operation can be launched by software through a series of peripheral bus writes to flash registers. In addition the entire flash memory may be erased external to the flash memory from the SWJ-DP debug port by setting DAP_CONTROL[0]. DAP_STATUS[0] is set to indicate the mass erase command has been accepted. DAP_STATUS[0] is cleared when the mass erase completes.

The EzPort can also initiate an erase of flash contents by issuing a bulk erase (BE) command. See the EzPort chapter for more details.

10.5.1.7 FTF_FOPT Register

The flash memory's FTF_FOPT register allows the user to customize the operation of the MCU at boot time. See FOPT boot options for details of its definition.

10.5.2 Flash Memory Controller Configuration

This section summarizes how the module has been configured in the chip. For a comprehensive description of the module itself, see the module’s dedicated chapter.

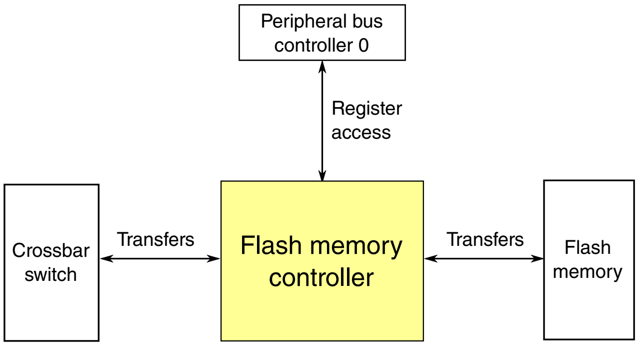


Figure 10-22. Flash memory controller configuration

Table 10-31. Reference links to related information

Topic	Related module	Reference
Full description	Flash memory controller	Flash memory controller
System memory map		System memory map
Clocking		Clock Distribution
Transfers	Flash memory	Flash memory
Transfers	Crossbar switch	Crossbar Switch
Register access	Peripheral bridge	Peripheral bridge

10.5.2.1 Number of masters

The Flash Memory Controller supports up to eight crossbar switch masters. However, this device has a different number of crossbar switch masters. See Crossbar-Light Switch Configuration for details on the master port assignments.

10.5.2.2 Program Flash Swap

On devices that contain program flash memory only, the program flash memory blocks may swap their base addresses.

While not using swap:

- FMC_PFB0CR controls the lower code addresses (block 0)
- FMC_PFB1CR controls the upper code addresses (block 1)

If swap is used, the opposite is true:

- FMC_PFB0CR controls the upper code addresses (now in block 0)
- FMC_PFB1CR controls the lower code addresses (now in block 1)

10.5.3 SRAM Configuration

This section summarizes how the module has been configured in the chip.

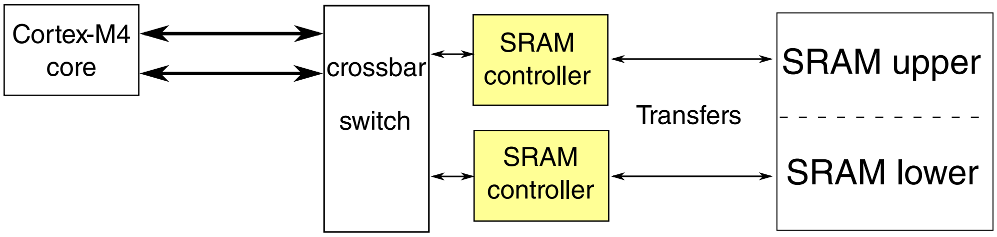


Figure 10-23. SRAM configuration

Table 10-32. Reference links to related information

Topic	Related module	Reference
Full description	SRAM	SRAM
System memory map		System memory map
Clocking		Clock Distribution
Transfers	SRAM controller	SRAM controller
	ARM Cortex-M4 core	ARM Cortex-M4 core

10.5.3.1 SRAM sizes

This device contains SRAM accessed by bus masters through the cross-bar switch. The amount of SRAM for the devices covered in this document is shown in the following table.

Device	SRAM (KB)
MKW21D256VHA5	32
MKW21/22/24D512VHA5	64

10.5.3.2 Flash Memory Sizes

The devices covered in this document contain:

- For devices with program flash only: 2 blocks of program flash consisting of 2 KB sectors

- For devices that contain FlexNVM: 1 block of program flash consisting of 2 KB sectors
- For devices that contain FlexNVM: 1 block of FlexNVM consisting of 2 KB sectors
- For devices that contain FlexNVM: 1 block of FlexRAM

The amounts of flash memory for the devices covered in this document are:

Device	Program flash (KB)	Block 0 (P-Flash) address range	FlexNVM (KB)	Block 1 (FlexNVM/ P-Flash) address range[1]	FlexRAM/ Programming Acceleration RAM (KB)	FlexRAM/ Programming Acceleration RAM address range
MKW21D256VHA5	256	0x0000_0000–0x0003_FFFF	64	0x1000_0000–0x1000_FFFF	4	0x1400_0000–0x1400_0FFF
MK22DX256VLF5	256	0x0000_0000–0x0003_FFFF	–	0x0004_0000–0x0007_FFFF	4	0x1400_0000–0x1400_0FFF

[1] For program flash only devices: The addresses shown assume program flash swap is disabled (default configuration).

10.5.3.3 SRAM Arrays

The on-chip SRAM is split into two equally-sized logical arrays, SRAM_L and SRAM_U.

The on-chip RAM is implemented such that the SRAM_L and SRAM_U ranges form a contiguous block in the memory map. As such:

- SRAM_L is anchored to 0x1FFF_FFFF and occupies the space before this ending address.
- SRAM_U is anchored to 0x2000_0000 and occupies the space after this beginning address.

Valid address ranges for SRAM_L and SRAM_U are then defined as:

- SRAM_L = [0x2000_0000–(SRAM_size/2)] to 0x1FFF_FFFF
- SRAM_U = 0x2000_0000 to [0x2000_0000+(SRAM_size/2)-1]

This is illustrated in the following figure.

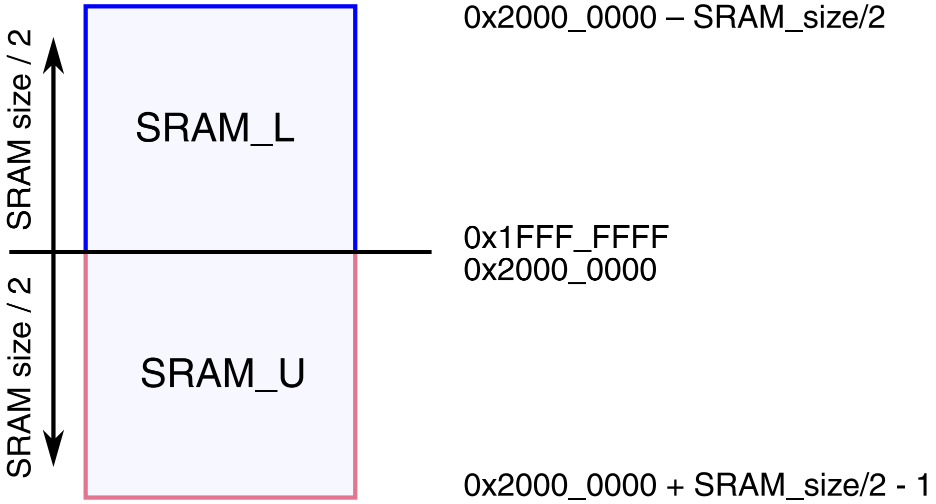


Figure 10-24. SRAM blocks memory map

For example, for a device containing 64 KB of SRAM the ranges are:

- SRAM_L: 0x1FFF_8000 – 0x1FFF_FFFF
- SRAM_U: 0x2000_0000 – 0x2000_7FFF

10.5.3.4 SRAM retention in low power modes

The SRAM is retained down to VLLS3 mode.

In VLLS2 the 4 or 16 KB (user option) region of SRAM_U from 0x2000_0000 is powered. These different regions (or partitions) of SRAM are labeled as follows:

- RAM1: the 4 KB region always powered in VLLS2
- RAM2: the additional 12 KB region optionally powered in VLLS2
- RAM3: the rest of system RAM

In VLLS1 and VLLS0 no SRAM is retained; however, the 32-byte register file is available.

10.5.4 System Register File Configuration

This section summarizes how the module has been configured in the chip.

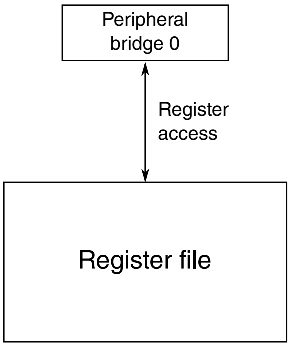


Figure 10-25. System Register file configuration

Table 10-33. Reference links to related information

Topic	Related module	Reference
Full description	Register file	Register file
System memory map		System memory map
Clocking		Clock distribution
Power management		Power management

10.5.4.1 System Register file

This device includes a 32-byte register file that is powered in all power modes.

Also, it retains contents during low-voltage detect (LVD) events and is only reset during a power-on reset.

10.5.5 VBAT Register File Configuration

This section summarizes how the module has been configured in the chip.

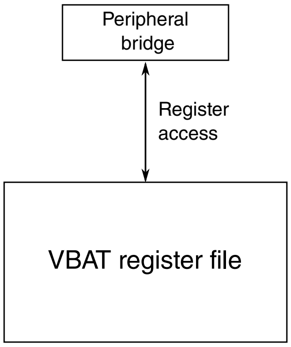


Figure 10-26. VBAT Register file configuration

Table 10-34. Reference links to related information

Topic	Related module	Reference
Full description	VBAT register file	VBAT register file
System memory map		System memory map
Clocking		Clock distribution
Power management		Power management

10.5.5.1 VBAT register file

This device includes a 32-byte register file that is powered in all power modes and is powered by VBAT.

It is only reset during VBAT power-on reset.

10.5.6 EzPort Configuration

This section summarizes how the module has been configured in the chip. For a comprehensive description of the module itself, see the module's dedicated chapter.

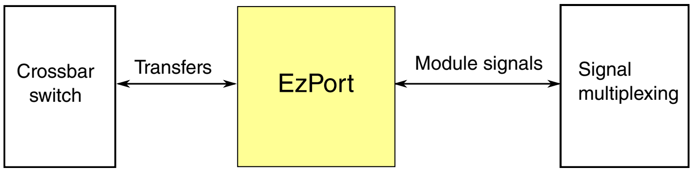


Figure 10-27. EzPort configuration

Table 10-35. Reference links to related information

Topic	Related module	Reference
Full description	EzPort	EzPort

Table continues on the next page...

Table 10-35. Reference links to related information (continued)

Topic	Related module	Reference
System memory map		System memory map
Clocking		Clock Distribution
Transfers	Crossbar switch	Crossbar switch
Signal Multiplexing	Port control	Signal Multiplexing

10.5.6.1 JTAG instruction

The system JTAG controller implements an EZPORT instruction. When executing this instruction, the JTAG controller resets the core logic and asserts the EzPort chip select signal to force the processor into EzPort mode.

10.5.6.2 Flash Option Register (FOPT)

The FOPT[EZPORT_DIS] bit can be used to prevent entry into EzPort mode during reset. If the FOPT[EZPORT_DIS] bit is cleared, then the state of the chip select signal ($\overline{\text{EZP\_CS}}$) is ignored and the MCU always boots in normal mode.

This option is useful for systems that use the $\overline{\text{EZP\_CS}}$/NMI signal configured for its NMI function. Disabling EzPort mode prevents possible unwanted entry into EzPort mode if the external circuit that drives the NMI signal asserts it during reset.

The FOPT register is loaded from the flash option byte. If the flash option byte is modified the new value takes effect for any subsequent resets, until the value is changed again.

10.6 Security

10.6.1 CRC Configuration

This section summarizes how the module has been configured in the chip. For a comprehensive description of the module itself, see the module’s dedicated chapter.

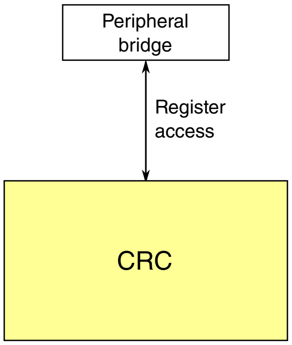


Figure 10-28. CRC configuration

Table 10-36. Reference links to related information

Topic	Related module	Reference
Full description	CRC	CRC
System memory map		System memory map
Power management		Power management

10.6.2 MMCAU Configuration

This section summarizes how the module has been configured in the chip. For a comprehensive description of the module itself, see the module's dedicated chapter.

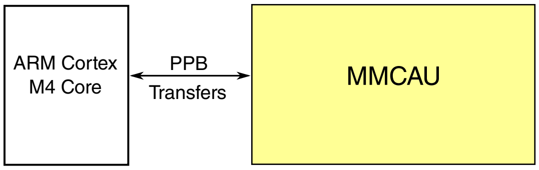


Figure 10-29. MMCAU configuration

Table 10-37. Reference links to related information

Topic	Related module	Reference
Full description	MMCAU	MMCAU
System memory map		System memory map
Clocking		Clock Distribution
Power Management		Power Management
Transfers Private Peripheral Bus (PPB)	ARM Cortex M4 Core	ARM Cortex-M4 Technical Reference Manual

10.6.3 RNG Configuration

This section summarizes how the module has been configured in the chip. For a comprehensive description of the module itself, see the module's dedicated chapter.

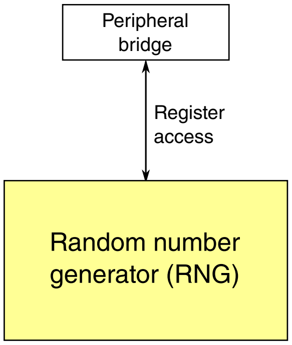


Figure 10-30. RNG configuration

Table 10-38. Reference links to related information

Topic	Related module	Reference
Full description	RNG	RNG
System memory map		System memory map
Clocking		Clock distribution
Power management		Power management

10.6.4 DryIce (tamper detect and secure storage) configuration

This section summarizes how the module has been configured in the chip. For a comprehensive description of the module itself, see the module's dedicated chapter.

NOTE

The information on this module is available only under NDA. Please contact your local sales office for details.

10.7 Analog

10.7.1 16-bit SAR ADC Configuration

This section summarizes how the module has been configured in the chip. For a comprehensive description of the module itself, see the module’s dedicated chapter.

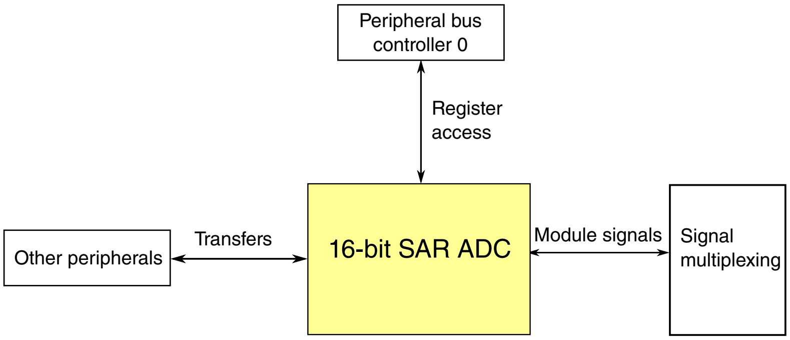


Figure 10-31. 16-bit SAR ADC configuration

Table 10-39. Reference links to related information

Topic	Related module	Reference
Full description	16-bit SAR ADC	16-bit SAR ADC
System memory map		System memory map
Clocking		Clock distribution
Power management		Power management
Signal multiplexing	Port control	Signal multiplexing

10.7.1.1 ADC instantiation information

This device contains one ADC.

10.7.1.1.1 Number of ADC channels

The number of ADC channels present on the device is determined by the pinout of the specific device package. For details regarding the number of ADC channel available on a particular package, refer to the signal multiplexing chapter of this MCU.

10.7.1.2 DMA Support on ADC

Applications may require continuous sampling of the ADC (4K samples/sec) that may have considerable load on the CPU. Though using PDB to trigger ADC may reduce some CPU load, the ADC supports DMA request functionality for higher performance when the ADC is sampled at a very high rate or cases where PDB is bypassed. The ADC can trigger the DMA (via DMA req) on conversion completion.

10.7.1.3 Connections/channel assignment

10.7.1.3.1 ADC0 Connections/Channel Assignment

NOTE

As indicated by the following sections, each ADCx_DPx input and certain ADCx_DMx inputs may operate as single-ended ADC channels in single-ended mode.

10.7.1.3.1.1 ADC0 Channel Assignment for 121-Pin Packages

ADC Channel (SC1n[ADCH])	Channel	Input signal (SC1n[DIFF]= 1)	Input signal (SC1n[DIFF]= 0)
00000	DAD0	ADC0_DP0 and ADC0_DM0	ADC0_DP0
00001	DAD1	ADC0_DP1 and ADC0_DM1	ADC0_DP1
00010	DAD2	Reserved	Reserved
00011	DAD3	ADC0_DP3 and ADC0_DM3	ADC0_DP3
00100	AD4a	Reserved	ADC0_SE4a
00101[1]	AD5a	Reserved	ADC0_SE5a
00110[1]	AD6a	Reserved	ADC0_SE6a
00111[1]	AD7a	Reserved	ADC0_SE7a
00100[1]	AD4b	Reserved	ADC0_SE4b
00101[1]	AD5b	Reserved	ADC0_SE5b
00110[1]	AD6b	Reserved	ADC0_SE6b
00111[1]	AD7b	Reserved	ADC0_SE7b
01000	AD8	Reserved	ADC0_SE8
01001	AD9	Reserved	ADC0_SE9
01010	AD10	Reserved	ADC0_SE10
01011	AD11	Reserved	ADC0_SE11
01100	AD12	Reserved	ADC0_SE12
01101	AD13	Reserved	ADC0_SE13
01110	AD14	Reserved	ADC0_SE14
01111	AD15	Reserved	ADC0_SE15

Table continues on the next page...

ADC Channel (SC1n[ADCH])	Channel	Input signal (SC1n[DIFF]= 1)	Input signal (SC1n[DIFF]= 0)
10000	AD16	Reserved	Reserved
10001	AD17	Reserved	Reserved
10010	AD18	Reserved	Reserved
10011	AD19	Reserved	ADC0_DM0
10100	AD20	Reserved	ADC0_DM1
10101	AD21	Reserved	ADC0_SE21
10110	AD22	Reserved	ADC0_SE22
10111	AD23	Reserved	/ADC0_SE23
11000	AD24	Reserved	Reserved
11001	AD25	Reserved	Reserved
11010	AD26	Temperature Sensor (Diff)	Temperature Sensor (S.E)
11011	AD27	Bandgap (Diff)	Bandgap (S.E) [2]
11100	AD28	Reserved	Reserved
11101	AD29	VREFH (Diff)	VREFH (S.E)
11110	AD30	Reserved	VREFL
11111	AD31	Module Disabled	Module Disabled

[1] ADCx_CFG2[MUXSEL] bit selects between ADCx_SEn channels a and b. Refer to MUXSEL description in ADC chapter for details.

[2] This is the PMC bandgap 1V reference voltage. Prior to reading from this ADC channel, ensure that you enable the bandgap buffer by setting the PMC_REGSC[BGBE] bit. Refer to the device data sheet for the bandgap voltage (V_{BG}) specification.

10.7.1.4 ADC Channels MUX Selection

The following figure shows the assignment of ADCx_SEn channels a and b through a MUX selection to ADC. To select between alternate set of channels, refer to ADCx_CFG2[MUXSEL] bit settings for more details.

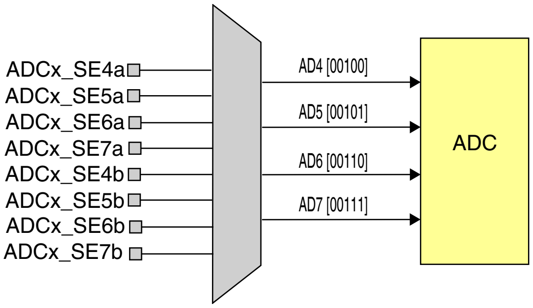


Figure 10-32. ADCx_SEn channels a and b selection

10.7.1.5 ADC Reference Options

The voltage reference of the ADC is VREFH/VREFL. The voltage reference selection bits ADCx_SC2[REFSEL] must be set to the default 00.

10.7.1.6 ADC triggers

The ADC supports both software and hardware triggers. The primary hardware mechanism for triggering the ADC is the PDB. The PDB itself can be triggered by other peripherals. For example: RTC (Alarm, Seconds) signal is connected to the PDB. The PDB input trigger can receive the RTC (alarm/seconds) trigger forcing ADC conversions in run mode (where PDB is enabled). On the other hand, the ADC can conduct conversions in low power modes, not triggered by PDB. This allows the ADC to do conversions in low power mode and store the output in the result register. The ADC generates interrupt when the data is ready in the result register that wakes the system from low power mode. The PDB can also be bypassed by using the ADCxTRGSEL bits in the SIM_SOPT7 register.

For operation of triggers in different modes, refer to Power Management chapter.

10.7.1.7 Alternate clock

For this device, the alternate clock is connected to OSCERCLK.

NOTE

> This clock option is only usable when OSCERCLK is in the MHz range. A system with OSCERCLK in the kHz range has the optional clock source below minimum ADC clock operating frequency.

10.7.1.8 ADC low-power modes

This table shows the ADC low-power modes and the corresponding chip low-power modes.

Table 10-40. ADC low-power modes

Module mode	Chip mode
Wait	Wait, VLPW
Normal Stop	Stop, VLPS
Low Power Stop	LLS, VLLS3, VLLS2, VLLS1, VLLS0

10.7.2 CMP Configuration

This section summarizes how the module has been configured in the chip. For a comprehensive description of the module itself, see the module's dedicated chapter.

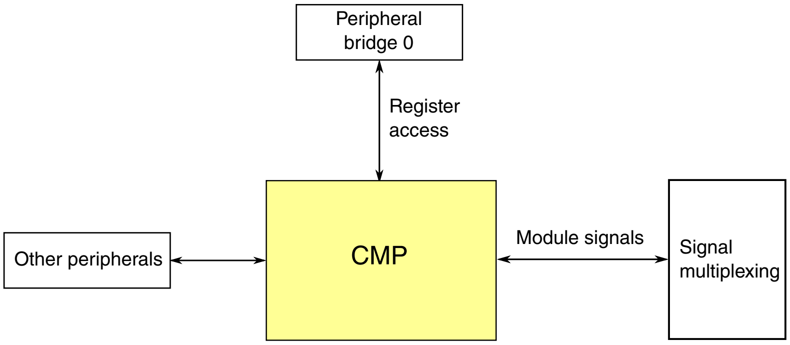


Figure 10-33. CMP configuration

Table 10-41. Reference links to related information

Topic	Related module	Reference
Full description	Comparator (CMP)	Comparator
System memory map		System memory map
Clocking		Clock distribution
Power management		Power management
Signal multiplexing	Port control	Signal multiplexing

10.7.2.1 CMP input connections

The following table shows the fixed internal connections to the CMP.

Table 10-42. CMP input connections

CMP Inputs	CMP0	CMP1
IN0	CMP0_IN0	CMP1_IN0
IN1	CMP0_IN1	CMP1_IN1
IN2	CMP0_IN2	—
IN3	CMP0_IN3	—
IN4	—	—
IN5	CMP0_IN5	CMP1_IN5
IN6	Bandgap	Bandgap
IN7	6b DAC0 Reference	6b DAC1 Reference

10.7.2.2 CMP external references

The 6-bit DAC sub-block can be only be configured to accept the voltage reference: VDD (the V_{in2} input). The V_{in1} input is not available and should not be selected.

10.7.2.3 External window/sample input

Individual PDB pulse-out signals control each CMP Sample/Window timing.

10.8 Timers

10.8.1 PDB Configuration

This section summarizes how the module has been configured in the chip. For a comprehensive description of the module itself, see the module's dedicated chapter.

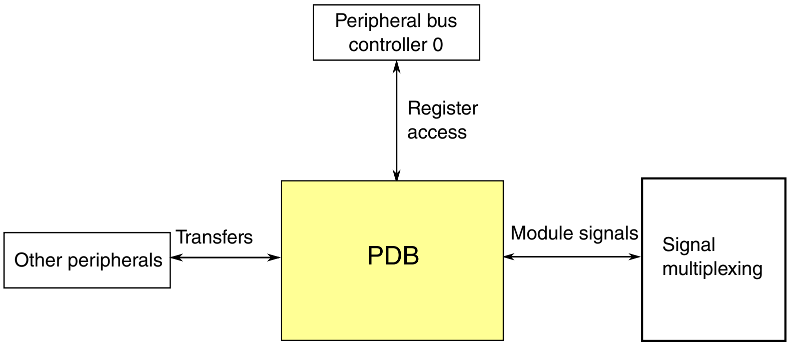


Figure 10-34. PDB configuration

Table 10-43. Reference links to related information

Topic	Related module	Reference
Full description	PDB	PDB
System memory map		System memory map
Clocking		Clock distribution
Power management		Power management
Signal multiplexing	Port control	Signal multiplexing

10.8.1.1 PDB Instantiation

10.8.1.1.1 PDB Output Triggers

Table 10-44. PDB output triggers

Number of PDB channels for ADC trigger	1
Number of PDB channels to inter-connect with FTM0	1[1]
Number of pre-triggers per PDB channel	2
Number of Pulse Out	2

[1] The PDB output trigger to FTM0 will occur only once. To have a reoccurring FTM0 trigger, disable and enable PDB_CH1C1[EN] or PDB_SC[PDBEN], after a trigger is sent to FTM0.

10.8.1.1.2 PDB Input Trigger Connections

Table 10-45. PDB Input Trigger Options

PDB Trigger	PDB Input
0000	External Trigger
0001	CMP 0
0010	CMP 1
0011	Reserved
0100	PIT Ch 0 Output
0101	PIT Ch 1 Output
0110	PIT Ch 2 Output
0111	PIT Ch 3 Output
1000	FTM0 initialization trigger and channel triggers, as programmed in the FTM external trigger register (EXTTRIG)
1001	FTM1 initialization trigger and channel triggers, as programmed in the FTM external trigger register (EXTTRIG)
1010	FTM2 initialization trigger and channel triggers, as programmed in the FTM external trigger register (EXTTRIG)
1011	Reserved
1100	RTC Alarm
1101	RTC Seconds
1110	LPTMR Output
1111	Software Trigger

10.8.1.2 PDB Module Interconnections

PDB trigger outputs	Connection
Channel 0 triggers	ADC0 trigger

Table continues on the next page...

PDB trigger outputs	Connection
Channel 1 triggers	synchronous input 1 of FTM0
Pulse-out	Pulse-out connected to each CMP module's sample/window input to control sample operation

10.8.1.3 Back-to-back acknowledgement connections

The application code can set the PDB*x*_CH*n*C1[BB] bits to configure the PDB pre-triggers as a single chain or several chains.

10.8.1.4 Pulse-Out Connection

10.8.1.5 Pulse-Out Enable Register Implementation

The following table shows the comparison of pulse-out enable register at the module and chip level.

Table 10-46. PDB pulse-out enable register

Register	Module implementation	Chip implementation
POnEN	7:0 - POEN 31:8 - Reserved	0 - POEN[0] for CMP0 1 - POEN[1] for CMP1 31:2 - Reserved

10.8.2 FlexTimer Configuration

This section summarizes how the module has been configured in the chip. For a comprehensive description of the module itself, see the module’s dedicated chapter.

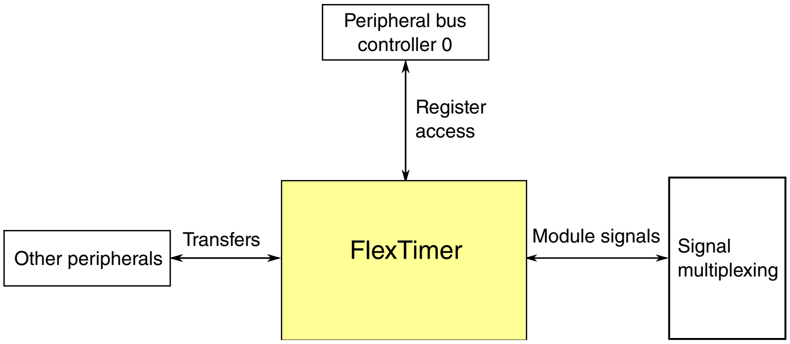


Figure 10-35. FlexTimer configuration

Table 10-47. Reference links to related information

Topic	Related module	Reference
Full description	FlexTimer	FlexTimer
System memory map		System memory map
Clocking		Clock distribution
Power management		Power management
Signal multiplexing	Port control	Signal multiplexing

10.8.2.1 Instantiation Information

This device contains three FlexTimer modules.

The following table shows how these modules are configured.

Table 10-48. FTM Instantiations

FTM instance	Number of channels	Features/usage
FTM0	8	3-phase motor + 2 general purpose or stepper motor
FTM1	2	Quadrature decoder or general purpose
FTM2	2[1]	General purpose

[1] Only channels 0 and 1 are available.

10.8.2.2 External Clock Options

By default each FTM is clocked by the internal bus clock (the FTM refers to it as system clock). Each module contains a register setting that allows the module to be clocked from an external clock instead. There are two external FTM_CLKINx pins that can be selected by any FTM module via the SIM_SOPT4 register.

10.8.2.3 Fixed frequency clock

The fixed frequency clock for each FTM is MCGFFCLK.

10.8.2.4 FTM Interrupts

The FlexTimer has multiple sources of interrupt. However, these sources are OR'd together to generate a single interrupt request to the interrupt controller. When an FTM interrupt occurs, read the FTM status registers (FMS, SC, and STATUS) to determine the exact interrupt source.

10.8.2.5 FTM Fault Detection Inputs

The following fault detection input options for the FTM modules are selected via the SIM_SOPT4 register. The external pin option is selected by default.

- FTM0 FAULT0 = FTM0_FLT0 pin or CMP0 output
- FTM0 FAULT1 = FTM0_FLT1 pin or CMP1 output
- FTM0 FAULT2 = FTM0_FLT2 pin
- FTM0 FAULT3 = FTM0_FLT3 pin

- FTM1 FAULT0 = FTM1_FLT0 pin or CMP0 output
- FTM1 FAULT1 = CMP1 output

- FTM2 FAULT0 = FTM2_FLT0 pin or CMP0 output
- FTM2 FAULT1 = CMP1 output

10.8.2.6 FTM Hardware Triggers

The FTM synchronization hardware triggers are connected in the chip as follows:

- FTM0 hardware trigger 0 = CMP0 Output or FTM1 Match (when enabled in the FTM1 External Trigger (EXTTRIG) register)
- FTM0 hardware trigger 1 = PDB channel 1 Trigger Output or FTM2 Match (when enabled in the FTM2 External Trigger (EXTTRIG) register)
- FTM0 hardware trigger 2 = FTM0_FLT0 pin

- FTM1 hardware trigger 0 = CMP0 Output
- FTM1 hardware trigger 1 = CMP1 Output
- FTM1 hardware trigger 2 = FTM1_FLT0 pin

- FTM2 hardware trigger 0 = CMP0 Output
- FTM2 hardware trigger 2 = FTM2_FLT0 pin

For the triggers with more than one option, SIM_SOPT4 controls the selection.

10.8.2.7 Input capture options for FTM module instances

The following channel 0 input capture source options are selected via SIM_SOPT4. The external pin option is selected by default.

- FTM1 channel 0 input capture = FTM1_CH0 pin or CMP0 output or CMP1 output or USB start of frame pulse
- FTM2 channel 0 input capture = FTM2_CH0 pin or CMP0 output or CMP1 output

NOTE

When the USB start of frame pulse option is selected as an FTM channel input capture, disable the USB SOF token interrupt in the USB Interrupt Enable register (INTEN[SOFTOKEN]) to avoid USB enumeration conflicts.

10.8.2.8 FTM output triggers for other modules

FTM output triggers can be selected as input triggers for the PDB and ADC modules. See PDB Instantiation and ADC triggers.

10.8.2.9 FTM Global Time Base

This chip provides the optional FTM global time base feature (see Global time base (GTB)).

FTM0 provides the only source for the FTM global time base. The other FTM modules can share the time base as shown in the following figure:

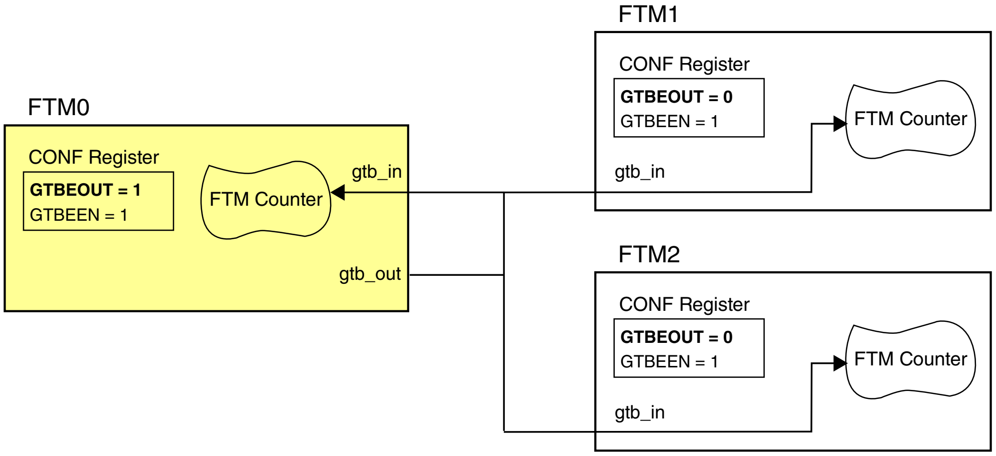


Figure 10-36. FTM Global Time Base Configuration

10.8.2.10 FTM BDM and debug halt mode

In the FTM chapter, references to the chip being in "BDM" are the same as the chip being in “debug halt mode".

10.8.3 PIT Configuration

This section summarizes how the module has been configured in the chip. For a comprehensive description of the module itself, see the module’s dedicated chapter.

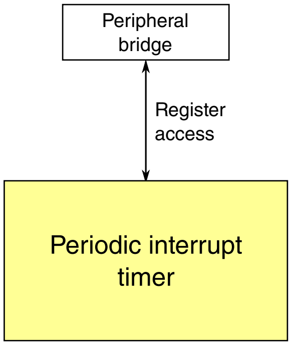


Figure 10-37. PIT configuration

Table 10-49. Reference links to related information

Topic	Related module	Reference
Full description	PIT	PIT
System memory map		System memory map
Clocking		Clock Distribution
Power management		Power management

10.8.3.1 PIT/DMA Periodic Trigger Assignments

The PIT generates periodic trigger events to the DMA Mux as shown in the table below.

Table 10-50. PIT channel assignments for periodic DMA triggering

DMA Channel Number	PIT Channel
DMA Channel 0	PIT Channel 0
DMA Channel 1	PIT Channel 1
DMA Channel 2	PIT Channel 2
DMA Channel 3	PIT Channel 3

10.8.3.2 PIT/ADC Triggers

PIT triggers are selected as ADCx trigger sources using the SIM_SOPT7[ADCxTRGSEL] fields. For more details, refer to SIM chapter.

10.8.4 Low-power timer configuration

This section summarizes how the module has been configured in the chip. For a comprehensive description of the module itself, see the module's dedicated chapter.

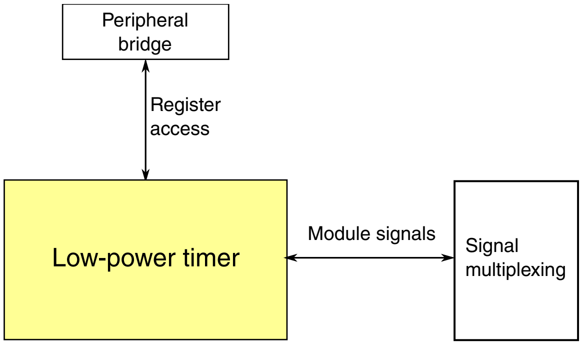


Figure 10-38. LPTMR configuration

Table 10-51. Reference links to related information

Topic	Related module	Reference
Full description	Low-power timer	Low-power timer
System memory map		System memory map
Clocking		Clock Distribution
Power management		Power management

Table continues on the next page...

Table 10-51. Reference links to related information (continued)

Topic	Related module	Reference
Signal Multiplexing	Port control	Signal Multiplexing

10.8.4.1 LPTMR prescaler/glitch filter clocking options

The prescaler and glitch filter of the LPTMR module can be clocked from one of four sources determined by the LPTMR0_PSR[PCS] bitfield. The following table shows the chip-specific clock assignments for this bitfield.

NOTE

The chosen clock must remain enabled if the LPTMR is to continue operating in all required low-power modes.

LPTMR0_PSR[PCS]	Prescaler/glitch filter clock number	Chip clock
00	0	MCGIRCLK — internal reference clock (not available in VLPS/LLS/VLLS modes)
01	1	LPO — 1 kHz clock (not available in VLLS0 mode)
10	2	ERCLK32K — secondary external reference clock
11	3	OSCERCLK — external reference clock (not available in VLLS0 mode)

See Clock Distribution for more details on these clocks.

10.8.4.2 LPTMR pulse counter input options

The LPTMR_CSR[TPS] bitfield configures the input source used in pulse counter mode. The following table shows the chip-specific input assignments for this bitfield.

LPTMR_CSR[TPS]	Pulse counter input number	Chip input
00	0	CMP0 output
01	1	LPTMR_ALT1 pin
10	2	LPTMR_ALT2 pin
11	3	LPTMR_ALT3 pin

10.8.5 CMT Configuration

This section summarizes how the module has been configured in the chip. For a comprehensive description of the module itself, see the module's dedicated chapter.

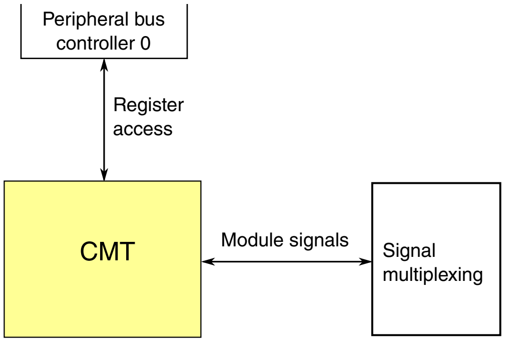


Figure 10-39. CMT configuration

Table 10-52. Reference links to related information

Topic	Related module	Reference
Full description	Carrier modulator transmitter (CMT)	CMT
System memory map		System memory map
Clocking		Clock distribution
Power management		Power management
Signal multiplexing	Port control	Signal multiplexing

10.8.5.1 Instantiation Information

This device contains one CMT module.

10.8.5.2 IRO Drive Strength

The IRO pad requires higher current drive than can be obtained from a single pad. For this device, the pin associated with the CMT_IRO signal is doubled bonded to two pads.

SIM_SOPT2[PTD7PAD] can be used to configure the pin associated with the CMT_IRO signal as a higher current output port pin.

10.8.6 RTC configuration

This section summarizes how the module has been configured in the chip. For a comprehensive description of the module itself, see the module's dedicated chapter.

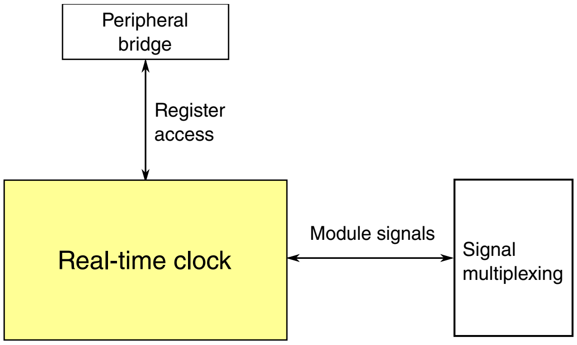


Figure 10-40. RTC configuration

Table 10-53. Reference links to related information

Topic	Related module	Reference
Full description	RTC	RTC
System memory map		System memory map
Clocking		Clock Distribution
Power management		Power management

10.8.6.1 RTC_CLKOUT signal

When the RTC is enabled and the port control module selects the RTC_CLKOUT function, the RTC_CLKOUT signal outputs a 1 Hz or 32 kHz output derived from RTC oscillator as shown below.

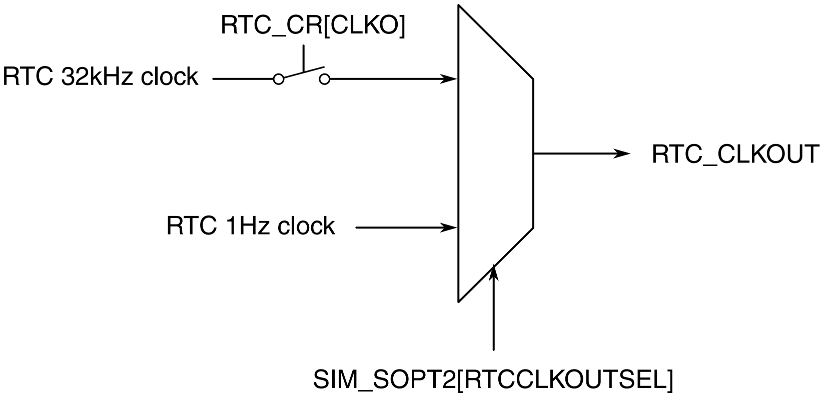


Figure 10-41. RTC_CLKOUT generation

10.9 Communication interfaces

10.9.1 Universal Serial Bus (USB) FS Subsystem

The USB FS subsystem includes these components:

- Dual-role USB OTG-capable (On-The-Go) controller that supports a full-speed (FS) device or FS/LS host. The module complies with the USB 2.0 specification.
- USB transceiver that includes internal 15 kΩ pulldowns on the D+ and D- lines for host mode functionality.
- A 3.3 V regulator.
- USB device charger detection module.
- VBUS detect signal: To detect a valid VBUS in device mode, use a GPIO signal that can wake the chip in all power modes. .

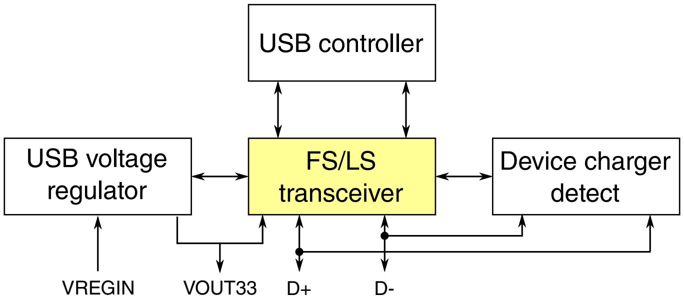


Figure 10-42. USB FS/LS Subsystem Overview

NOTE

Use the following code sequence to select USB clock source, USB clock divide ratio, and enable its clock gate to avoid potential clock glitches which may result in USB enumeration stage failure.

1. Select the USB clock source by configuring SIM_SOPT2.
2. Select the desired clock divide ratio by configuring SIM_CLKDIV2.
3. Enable USB clock gate by setting SIM_SCGC4.

10.9.1.1 USB Wakeup

When the USB detects that there is no activity on the USB bus for more than 3 ms, the INT_STAT[SLEEP] bit is set. This bit can cause an interrupt and software decides the appropriate action.

Waking from a low power mode (except in LLS/VLLS mode where USB is not powered) occurs through an asynchronous interrupt triggered by activity on the USB bus. Setting the USBTRC0[USBRESMEN] bit enables this function.

10.9.1.2 USB Power Distribution

This chip includes an internal 5 V to 3.3 V USB regulator that powers the USB transceiver or the MCU (depending on the application).

NOTE

In the following examples, VREGIN is used instead of VREG_IN0. Similarly, VOUT33 is used instead of VREGOUT. Please refer to the Signal multiplexing and signal description section for details on signals for this device.

10.9.1.2.1 AA/AAA cells power supply

The chip can be powered by two AA/AAA cells. In this case, the MCU is powered through VDD which is within the 1.8 to 3.0 V range. After USB cable insertion is detected, the USB regulator is enabled to power the USB transceiver.

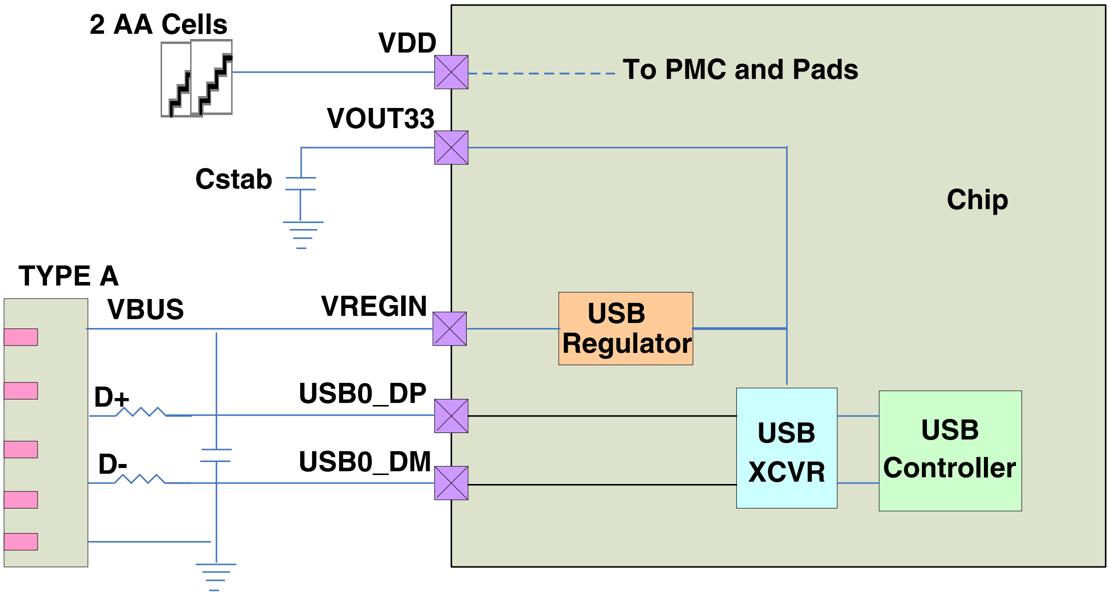


Figure 10-43. USB regulator AA cell usecase

10.9.1.2.2 Li-Ion battery power supply

The chip can also be powered by a single Li-ion battery. In this case, VOUT33 is connected to VDD. The USB regulator must be enabled by default to power the MCU. When connected to a USB host, the input source of this regulator is switched to the USB bus supply from the Li-ion battery. To charge the battery, the MCU can configure the battery charger according to the charger detection information.

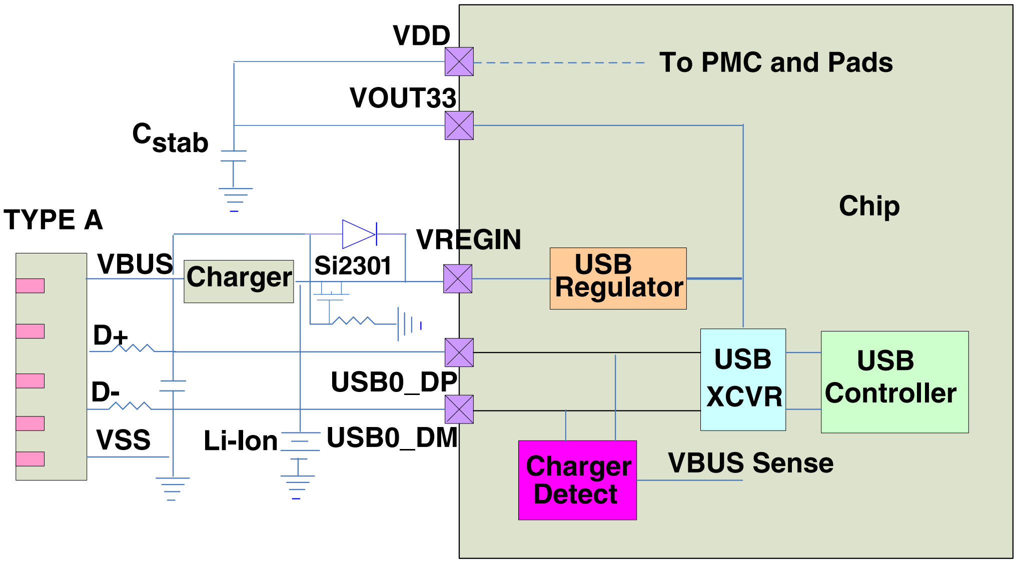


Figure 10-44. USB regulator Li-ion usecase

10.9.1.2.3 USB bus power supply

The chip can also be powered by the USB bus directly. In this case, VOUT33 is connected to VDD. The USB regulator must be enabled by default to power the MCU, then to power USB transceiver or external sensor.

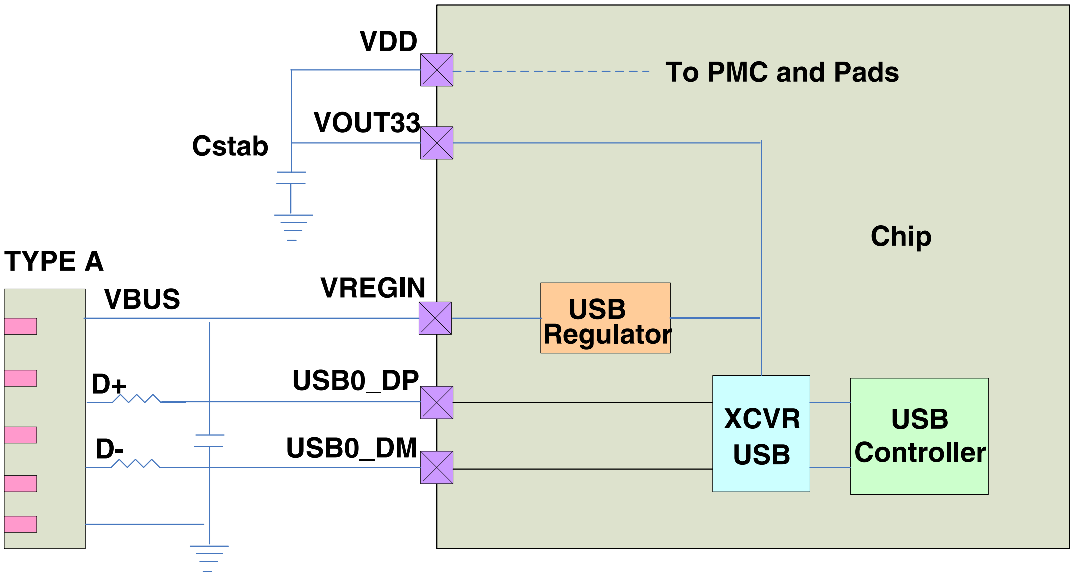


Figure 10-45. USB regulator bus supply

10.9.1.3 USB power management

The regulator should be put into STANDBY mode whenever the chip is in Stop mode.

10.9.1.4 USB controller configuration

This section summarizes how the module has been configured in the chip. For a comprehensive description of the module itself, see the module's dedicated chapter.

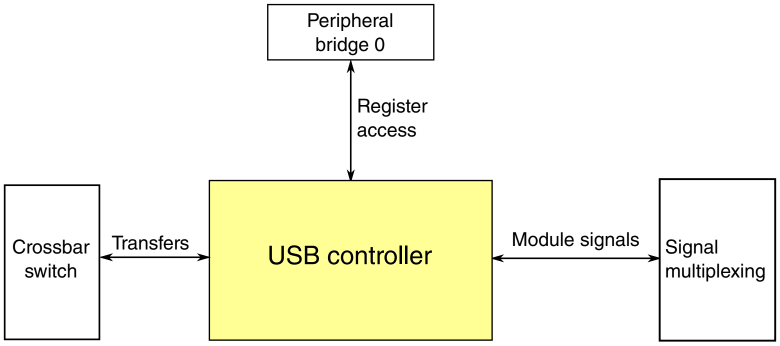


Figure 10-46. USB controller configuration

Table 10-54. Reference links to related information

Topic	Related module	Reference
Full description	USB controller	USB controller
System memory map		System memory map
Clocking		Clock Distribution
Transfers	Crossbar switch	Crossbar switch
Signal Multiplexing	Port control	Signal Multiplexing

NOTE

When USB is not used in the application, it is recommended that the USB regulator VREGIN and VOUT33 pins remain floating.

NOTE

When USB is not used in the application, it is recommended that the remain floating.

10.9.1.5 USB DCD Configuration

This section summarizes how the module has been configured in the chip. For a comprehensive description of the module itself, see the module's dedicated chapter.

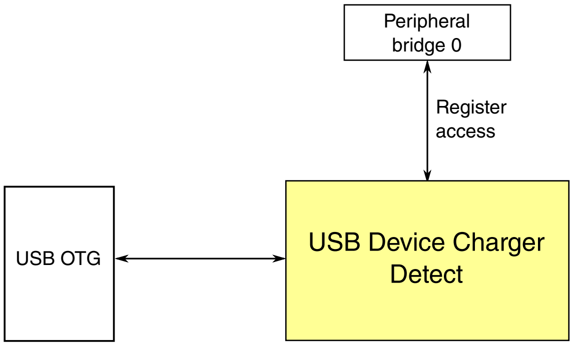


Figure 10-47. USB DCD configuration

Table 10-55. Reference links to related information

Topic	Related module	Reference
Full description	USB DCD	USB DCD
System memory map		System memory map
Clocking		Clock Distribution
	USB FS/LS controller	USB FS/LS controller

10.9.1.6 USB Voltage Regulator Configuration

This section summarizes how the module has been configured in the chip. For a comprehensive description of the module itself, see the module's dedicated chapter.

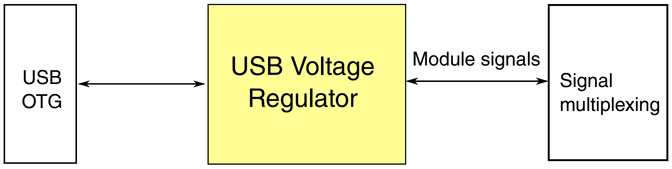


Figure 10-48. USB Voltage Regulator configuration

Table 10-56. Reference links to related information

Topic	Related module	Reference
Full description	USB Voltage Regulator	USB Voltage Regulator
System memory map		System memory map
Clocking		Clock Distribution
	USB controller	USB controller
Signal Multiplexing	Port control	Signal Multiplexing

NOTE

When USB is not used in the application, it is recommended that the USB regulator VREGIN and VOUT33 pins remain floating.

10.9.2 SPI configuration

This section summarizes how the module has been configured in the chip. For a comprehensive description of the module itself, see the module's dedicated chapter.

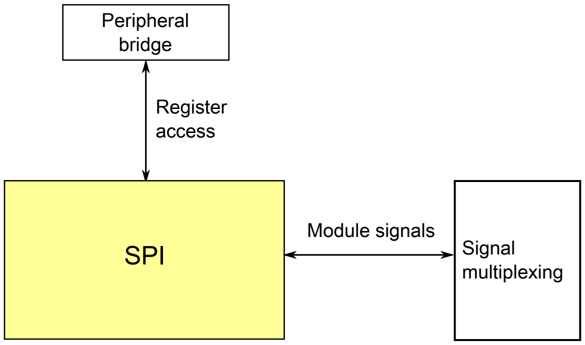


Figure 10-49. SPI configuration

Table 10-57. Reference links to related information

Topic	Related module	Reference
Full description	SPI	SPI
System memory map		System memory map
Clocking		Clock Distribution
Signal Multiplexing	Port control	Signal Multiplexing

10.9.2.1 SPI Modules Configuration

This device contains two SPI modules .

10.9.2.2 SPI clocking

The SPI module is clocked by the internal bus clock (the DSPI refers to it as system clock). The module has an internal divider, with a minimum divide is two. So, the SPI can run at a maximum frequency of bus clock/2.

10.9.2.3 Number of CTARs

SPI CTAR registers define different transfer attribute configurations. The SPI module supports up to eight CTAR registers. This device supports two CTARs on all instances of the SPI.

In master mode, the CTAR registers define combinations of transfer attributes, such as frame size, clock phase, clock polarity, data bit ordering, baud rate, and various delays. In slave mode only CTAR0 is used, and a subset of its bitfields sets the slave transfer attributes.

10.9.2.4 TX FIFO size

Table 10-58. SPI transmit FIFO size

SPI Module	Transmit FIFO size
SPI0	4
SPI1	4

10.9.2.5 RX FIFO Size

SPI supports up to 16-bit frame size during reception.

Table 10-59. SPI receive FIFO size

SPI Module	Receive FIFO size
SPI0	4
SPI1	4

10.9.2.6 Number of PCS signals

The following table shows the number of peripheral chip select signals available per SPI module.

Table 10-60. SPI PCS signals

SPI Module	PCS Signals
SPI0	SPI_PCS[4:0]
SPI1	SPI_PCS[2:0]

10.9.2.7 SPI Operation in Low Power Modes

In VLPR and VLPW modes the SPI is functional; however, the reduced system frequency also reduces the max frequency of operation for the SPI. In VLPR and VLPW modes the max SPI_CLK frequency is 2MHz.

In stop and VLPS modes, the clocks to the SPI module are disabled. The module is not functional, but it is powered so that it retains state.

There is one way to wake from stop mode via the SPI, which is explained in the following section.

10.9.2.7.1 Using GPIO Interrupt to Wake from stop mode

Here are the steps to use a GPIO to create a wakeup upon reception of SPI data in slave mode:

1. Point the GPIO interrupt vector to the desired interrupt handler.
2. Enable the GPIO input to generate an interrupt on either the rising or falling edge (depending on the polarity of the chip select signal).
3. Enter Stop or VLPS mode and Wait for the GPIO interrupt.

NOTE

It is likely that in using this approach the first word of data from the SPI host might not be received correctly. This is dependent on the transfer rate used for the SPI, the delay between chip select assertion and presentation of data, and the system interrupt latency.

10.9.2.8 SPI Doze Mode

The Doze mode for the SPI module is the same as the Wait and VLPW modes for the chip.

10.9.2.9 SPI Interrupts

The SPI has multiple sources of interrupt requests. However, these sources are OR'd together to generate a single interrupt request per SPI module to the interrupt controller. When an SPI interrupt occurs, read the SPI_SR to determine the exact interrupt source.

10.9.3 I2C Configuration

This section summarizes how the module has been configured in the chip. For a comprehensive description of the module itself, see the module's dedicated chapter.

NOTE

The Open Drain mode on SDA, SCL pins has to be enabled by setting PORTx_PCRn[ODE] bits.

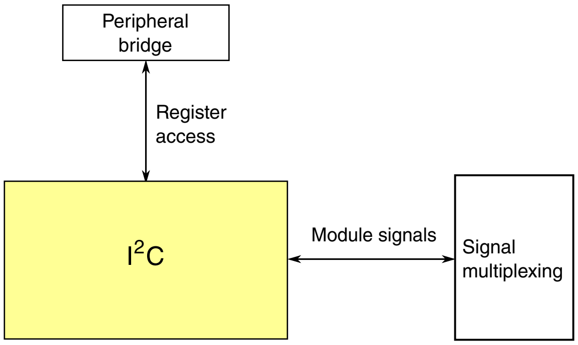


Figure 10-50. I2C configuration

Table 10-61. Reference links to related information

Topic	Related module	Reference
Full description	I2C	I2C
System memory map		System memory map
Clocking		Clock Distribution
Power management		Power management
Signal Multiplexing	Port control	Signal Multiplexing

10.9.3.1 Number of I2C modules

This device has two I^{2}C modules.

10.9.4 UART Configuration

This section summarizes how the module has been configured in the chip. For a comprehensive description of the module itself, see the module's dedicated chapter.

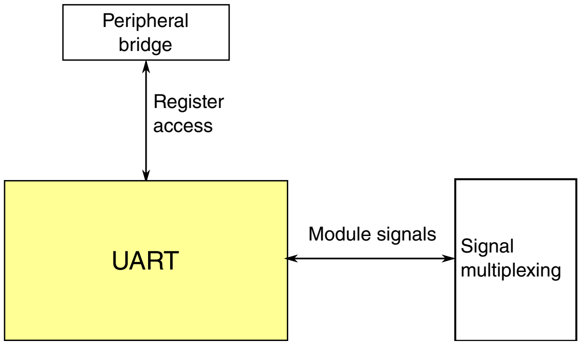


Figure 10-51. UART configuration

Table 10-62. Reference links to related information

Topic	Related module	Reference
Full description	UART	UART
System memory map		System memory map
Clocking		Clock Distribution
Power management		Power management
Signal Multiplexing	Port control	Signal Multiplexing

10.9.4.1 UART configuration information

This chip contains three UART modules. This section describes how each module is configured on this device.

1. Standard features of all UARTs:
   - RS-485 support
   - Hardware flow control (RTS/CTS)
   - 9-bit UART to support address mark with parity
   - MSB/LSB configuration on data
2. UART0 and UART1 are clocked from the core clock, the remaining UARTs are clocked on the bus clock. The maximum baud rate is 1/16 of related source clock frequency.
3. IrDA is available on all UARTs
4. UART0 contains the standard features plus ISO7816
5. UART0 contain 8-entry transmit and 8-entry receive FIFOs
6. All other UARTs contain a 1-entry transmit and receive FIFOs

10.9.4.2 UART wakeup

The UART can be configured to generate an interrupt/wakeup on the first active edge that it receives.

10.9.4.3 UART interrupts

The UART has multiple sources of interrupt requests. However, some of these sources are OR'd together to generate a single interrupt request. See below for the mapping of the individual interrupt sources to the interrupt request:

The status interrupt combines the following interrupt sources:

Source	UART 0	UART 1	UART 2
Transmit data empty	x	x	x
Transmit complete	x	x	x
Idle line	x	x	x
Receive data full	x	x	x
LIN break detect	x	x	x
RxD pin active edge	x	x	x
Initial character detect	x	—	—

The error interrupt combines the following interrupt sources:

Source	UART 0	UART 1	UART 2
Receiver overrun	x	x	x
Noise flag	x	x	x
Framing error	x	x	x
Parity error	x	x	x
Transmitter buffer overflow	x	x	x
Receiver buffer overflow	x	x	x
Receiver buffer underflow	x	x	x
Transmit threshold (ISO7816)	x	—	—
Receiver threshold (ISO7816)	x	—	—
Wait timer (ISO7816)	x	—	—
Character wait timer (ISO7816)	x	—	—
Block wait timer (ISO7816)	x	—	—
Guard time violation (ISO7816)	x	—	—

10.9.5 I²S configuration

This section summarizes how the module has been configured in the chip. For a comprehensive description of the module itself, see the module's dedicated chapter.

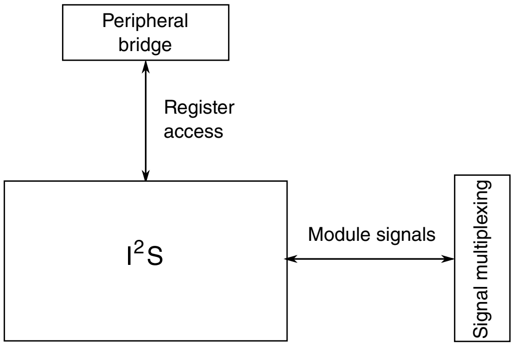


Figure 10-52. I²S configuration

Table 10-63. Reference links to related information

Topic	Related module	Reference
Full description	I²S	I2S
System memory map		System memory map
Clocking		Clock Distribution
Power management		Power management
Signal multiplexing	Port control	Signal Multiplexing

10.9.5.1 Instantiation information

This device contains one I²S module. Because of the limited set of pin availability in the SiP, the I2S/SAI block is usable only for receive mode and must be configured as a slave.

As configured on the device, module features include:

- RX data lines: 1
- FIFO size (words): 8
- Maximum words per frame: 16
- Maximum bit clock divider: 512

10.9.5.2 I²S/SAI clocking

10.9.5.2.1 Audio Master Clock

The audio master clock (MCLK) is used to generate the bit clock when the receiver or transmitter is configured for an internally generated bit clock. The audio master clock can also be output to or input from a pin. The transmitter and receiver have the same audio master clock inputs.

10.9.5.2.2 Bit Clock

The I^{2}S/SAI transmitter and receiver support asynchronous bit clocks (BCLKs) that can be generated internally from the audio master clock or supplied externally. The module also supports the option for synchronous operation between the receiver and transmitter product.

10.9.5.2.3 Bus Clock

The bus clock is used by the control registers and to generate synchronous interrupts and DMA requests.

10.9.5.2.4 I^{2}S/SAI clock generation

Each SAI peripheral can control the input clock selection, pin direction and divide ratio of one audio master clock.

The MCLK Input Clock Select bit of the MCLK Control Register (MCR[MICS]) selects the clock input to the I^{2}S/SAI module’s MCLK divider.

The module's MCLK Divide Register (MDR) configures the MCLK divide ratio.

The module's MCLK Output Enable bit of the MCLK Control Register (MCR[MOE]) controls the direction of the MCLK pin. The pin is the input from the pin when MOE is 0, and the pin is the output from the clock divider when MOE is 1.

The transmitter and receiver can independently select between the bus clock and the audio master clock to generate the bit clock. Each module's Clocking Mode field of the Transmit Configuration 2 Register and Receive Configuration 2 Register (TCR2[MSEL] and RCR2[MSEL]) selects the master clock.

10.9.5.2.5 Clock gating and I^{2}S/SAI initialization

The clock to the I^{2}S/SAI module can be gated using a bit in the SIM. To minimize power consumption, these bits are cleared after any reset, which disables the clock to the corresponding module. The clock enable bit should be set by software at the beginning of the module initialization routine to enable the module clock before initialization of any of the I^{2}S/SAI registers.

10.9.5.3 I^{2}S/SAI operation in low power modes

10.9.5.3.1 Stop and very low power modes

In Stop mode, the SAI transmitter and/or receiver can continue operating provided the appropriate Stop Enable bit is set (TCSR[STOPE] and/or RCSR[STOPE], respectively), and provided the transmitter and/or receiver is/are using an externally generated bit clock or an Audio Master Clock that remains operating in Stop mode. The SAI transmitter and/or receiver can generate an asynchronous interrupt to wake the CPU from Stop mode.

In VLPS mode, the module behaves as it does in stop mode if VLPS mode is entered from run mode. However, if VLPS mode is entered from VLPR mode, the FIFO might underflow or overflow before wakeup from stop mode due to the limits in bus bandwidth. In VLPW and VLPR modes, the module is limited by the maximum bus clock frequencies.

When operating from an internally generated bit clock or Audio Master Clock that is disabled in stop modes:

In Stop mode, if the Transmitter Stop Enable (TCSR[STOPE]) bit is clear, the transmitter is disabled after completing the current transmit frame, and, if the Receiver Stop Enable (RCSR[STOPE]) bit is clear, the receiver is disabled after completing the current receive frame. Entry into Stop mode is prevented–not acknowledged–while waiting for the transmitter and receiver to be disabled at the end of the current frame.

10.9.5.3.2 Low-leakage modes

When entering low-leakage modes, the Stop Enable (TCSR[STOPE] and RCSR[STOPE]) bits are ignored and the SAI is disabled after completing the current transmit and receive Frames. Entry into stop mode is prevented (not acknowledged) while waiting for the transmitter and receiver to be disabled at the end of the current frame.

10.10 Human-machine interfaces

10.10.1 GPIO configuration

This section summarizes how the module has been configured in the chip. For a comprehensive description of the module itself, see the module's dedicated chapter.

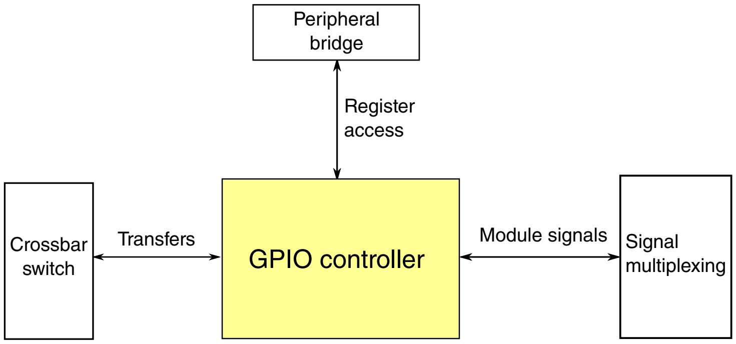


Figure 10-53. GPIO configuration

Table 10-64. Reference links to related information

Topic	Related module	Reference
Full description	GPIO	GPIO
System memory map		System memory map
Clocking		Clock Distribution
Power management		Power management
Transfers	Crossbar switch	Clock Distribution
Signal Multiplexing	Port control	Signal Multiplexing

10.10.1.1 GPIO access protection

The GPIO module does not have access protection because it is not connected to a peripheral bridge slot.

10.10.1.2 Number of GPIO signals

The number of GPIO signals available on the devices covered by this document are detailed in Orderable part numbers .

Chapter 11
MCU: Memory Map

11.1 Introduction

This device contains various memories and memory-mapped peripherals which are located in one 32-bit contiguous memory space. This chapter describes the memory and peripheral locations within that memory space.

11.2 System memory map

The following table shows the high-level device memory map. This map provides the complete architectural address space definition for the various sections. Based on the physical sizes of the memories and peripherals, the actual address regions used may be smaller.

Table 11-1. System memory map

System 32-bit Address Range	Destination Slave	Access
0x0000_0000–0x07FF_FFFF	Program flash and read-only data (Includes exception vectors in first 1024 bytes)	All masters
0x0800_0000–0x0FFF_FFFF	Reserved	—
0x1000_0000–0x13FF_FFFF	• For MK21DX128VMC5: FlexNVM • For MK21DX256VMC5: FlexNVM • For MK21DN512VMC5: Reserved	All masters
0x1400_0000–0x17FF_FFFF	For devices with FlexNVM: FlexRAM For devices with program flash only: Programming acceleration RAM	All masters
0x1FFF_8000-0x1FFF_FFFF	SRAM_L: Lower SRAM (ICODE/DCODE)	All masters
0x2000_0000–0x2000_7FFF	SRAM_U: Upper SRAM bitband region	All masters
0x2010_0000–0x21FF_FFFF	Reserved	–
0x2200_0000–0x23FF_FFFF	Aliased to SRAM_U bitband	Cortex-M4 core only
0x2400_0000–0x3FFF_FFFF	Reserved	–

Table continues on the next page...

Table 11-1. System memory map (continued)

System 32-bit Address Range	Destination Slave	Access
0x4000_0000–0x4007_FFFF	Bitband region for peripheral bridge 0 (AIPS-Lite0)	Cortex-M4 core & DMA/EzPort
0x4008_0000–0x400F_EFFF	Reserved	–
0x400F_F000–0x400F_FFFF	Bitband region for general purpose input/output (GPIO)	Cortex-M4 core & DMA/EzPort
0x4010_0000–0x41FF_FFFF	Reserved	–
0x4200_0000–0x43FF_FFFF	Aliased to peripheral bridge (AIPS-Lite) and general purpose input/output (GPIO) bitband	Cortex-M4 core only
0x4400_0000–0xDFFF_FFFF	Reserved	–
0xE000_0000–0xE00F_FFFF	Private peripherals	Cortex-M4 core only
0xE010_0000–0xFFFF_FFFF	Reserved	–

NOTE

1. EzPort master port is statically muxed with DMA master port. Access rights to AIPS-Lite peripheral bridge and general purpose input/output (GPIO) module address space is limited to the core, DMA and EzPort.
2. ARM Cortex-M4 core access privileges also includes accesses via the debug interface.

11.2.1 Aliased bit-band regions

The SRAM_U, AIPS-Lite, and general purpose input/output (GPIO) module resources reside in the Cortex-M4 processor bit-band regions.

The processor also includes two 32 MB aliased bit-band regions associated with the two 1 MB bit-band spaces. Each 32-bit location in the 32 MB space maps to an individual bit in the bit-band region. A 32-bit write in the alias region has the same effect as a read-modify-write operation on the targeted bit in the bit-band region.

Bit 0 of the value written to the alias region determines what value is written to the target bit:

- Writing a value with bit 0 set writes a 1 to the target bit.
- Writing a value with bit 0 clear writes a 0 to the target bit.

A 32-bit read in the alias region returns either:

- a value of 0x0000_0000 to indicate the target bit is clear
- a value of 0x0000_0001 to indicate the target bit is set

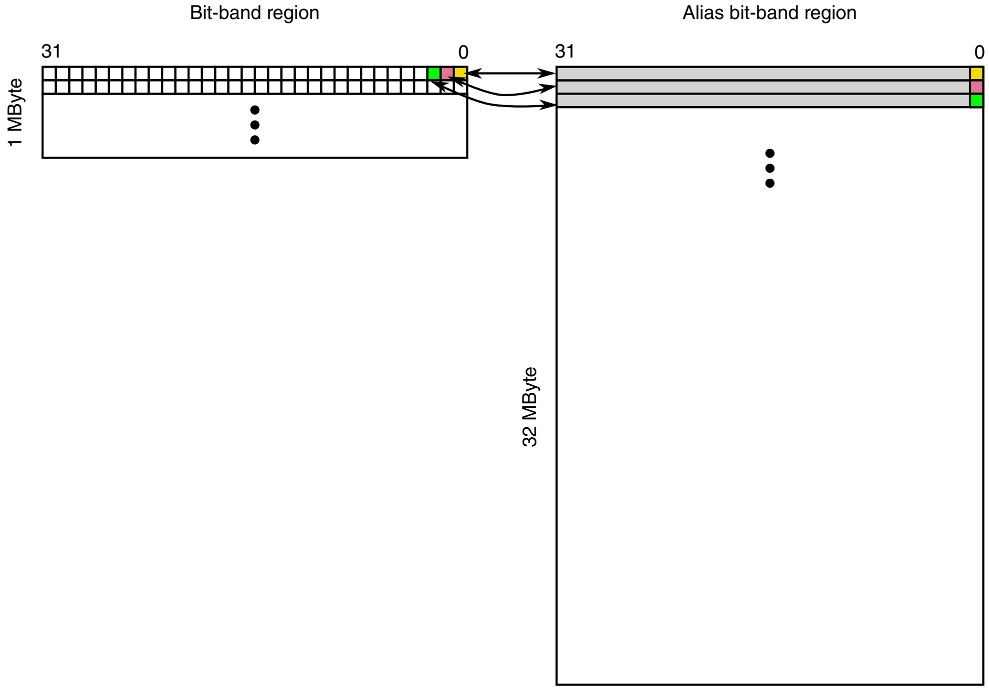


Figure 11-1. Alias bit-band mapping

NOTE

Each bit in bit-band region has an equivalent bit that can be manipulated through bit 0 in a corresponding long word in the alias bit-band region.

11.3 System memory map

The following table shows the high-level device memory map. This map provides the complete architectural address space definition for the various sections. Based on the physical sizes of the memories and peripherals, the actual address regions used may be smaller.

Table 11-2. System memory map

System 32-bit Address Range	Destination Slave	Access
0x0000_0000–0x07FF_FFFF	Program flash and read-only data (Includes exception vectors in first 1024 bytes)	All masters
0x0800_0000–0x0FFF_FFFF	Reserved	—
0x1000_0000–0x13FF_FFFF	• For MK21DX128VMC5: FlexNVM • For MK21DX256VMC5: FlexNVM • For MK21DN512VMC5: Reserved	All masters
0x1400_0000–0x17FF_FFFF	For devices with FlexNVM: FlexRAM	All masters

Table continues on the next page...

Table 11-2. System memory map (continued)

System 32-bit Address Range	Destination Slave	Access
	For devices with program flash only: Programming acceleration RAM	
0x1FFF_8000-0x1FFF_FFFF	SRAM_L: Lower SRAM (ICODE/DCODE)	All masters
0x2000_0000–0x2000_7FFF	SRAM_U: Upper SRAM bitband region	All masters
0x2010_0000–0x21FF_FFFF	Reserved	–
0x2200_0000–0x23FF_FFFF	Aliased to SRAM_U bitband	Cortex-M4 core only
0x2400_0000–0x3FFF_FFFF	Reserved	–
0x4000_0000–0x4007_FFFF	Bitband region for peripheral bridge 0 (AIPS-Lite0)	Cortex-M4 core & DMA/EzPort
0x4008_0000–0x400F_EFFF	Reserved	–
0x400F_F000–0x400F_FFFF	Bitband region for general purpose input/output (GPIO)	Cortex-M4 core & DMA/EzPort
0x4010_0000–0x41FF_FFFF	Reserved	–
0x4200_0000–0x43FF_FFFF	Aliased to peripheral bridge (AIPS-Lite) and general purpose input/output (GPIO) bitband	Cortex-M4 core only
0x4400_0000–0xDFFF_FFFF	Reserved	–
0xE000_0000–0xE00F_FFFF	Private peripherals	Cortex-M4 core only
0xE010_0000–0xFFFF_FFFF	Reserved	–

NOTE

1. EzPort master port is statically muxed with DMA master port. Access rights to AIPS-Lite peripheral bridge and general purpose input/output (GPIO) module address space is limited to the core, DMA and EzPort.
2. ARM Cortex-M4 core access privileges also includes accesses via the debug interface.

11.3.1 Alternate Non-Volatile IRC User Trim Description

The following non-volatile locations (4 bytes) are reserved for custom IRC user trim supported by some development tools. An alternate IRC trim to the factory loaded trim can be stored at this location. To override the factory trim, user software must load new values into the MCG trim registers.

Non-Volatile Byte Address	Alternate IRC Trim Value
0x0000_03FC	Reserved
0x0000_03FD	Reserved

Table continues on the next page...

Non-Volatile Byte Address	Alternate IRC Trim Value
0x0000_03FE (bit 0)	SCFTRIM
0x0000_03FE (bit 4:1)	FCTRIM
0x0000_03FF	SCTRIM

11.4 SRAM memory map

The on-chip RAM is split in two regions: SRAM_L and SRAM_U. The RAM is implemented such that the SRAM_L and SRAM_U ranges form a contiguous block in the memory map. See SRAM Configuration for details.

Accesses to the SRAM_L and SRAM_U memory ranges outside the amount of RAM on the device causes the bus cycle to be terminated with an error followed by the appropriate response in the requesting bus master.

11.5 Peripheral bridge (AIPS-Lite) memory map

The peripheral memory map is accessible via one slave port on the crossbar in the 0x4000_0000–0x4007_FFFF region. The device implements one peripheral bridge that defines a 512 KB address space.

Modules that are disabled via their clock gate control bits in the SIM registers disable the associated AIPS slots. Access to any address within an unimplemented or disabled peripheral bridge slot results in a transfer error termination.

For programming model accesses via the peripheral bridges, there is generally only a small range within the 4 KB slots that is implemented. Accessing an address that is not implemented in the peripheral results in a transfer error termination.

11.5.1 Read-after-write sequence and required serialization of memory operations

In some situations, a write to a peripheral must be completed fully before a subsequent action can occur. Examples of such situations include:

- Exiting an interrupt service routine (ISR)
- Changing a mode
- Configuring a function

In these situations, the application software must perform a read-after-write sequence to guarantee the required serialization of the memory operations:

1. Write the peripheral register.
2. Read the written peripheral register to verify the write.
3. Continue with subsequent operations.

11.5.2 Peripheral Bridge 0 (AIPS-Lite 0) Memory Map

Table 11-3. Peripheral bridge 0 slot assignments

System 32-bit base address	Slot number	Module
0x4000_0000	0	Peripheral bridge 0 (AIPS-Lite 0)
0x4000_1000	1	—
0x4000_2000	2	—
0x4000_3000	3	—
0x4000_4000	4	—
0x4000_5000	5	—
0x4000_6000	6	—
0x4000_7000	7	—
0x4000_8000	8	DMA controller
0x4000_9000	9	DMA controller transfer control descriptors
0x4000_A000	10	—
0x4000_B000	11	—
0x4000_C000	12	—
0x4000_D000	13	—
0x4000_E000	14	—
0x4000_F000	15	—
0x4001_0000	16	—
0x4001_1000	17	—
0x4001_2000	18	—
0x4001_3000	19	—
0x4001_4000	20	—
0x4001_5000	21	—
0x4001_6000	22	—
0x4001_7000	23	—
0x4001_8000	24	—
0x4001_9000	25	—
0x4001_A000	26	—
0x4001_B000	27	—
0x4001_C000	28	—
0x4001_D000	29	—

Table continues on the next page...

Table 11-3. Peripheral bridge 0 slot assignments (continued)

System 32-bit base address	Slot number	Module
0x4001_E000	30	—
0x4001_F000	31	Flash memory controller
0x4002_0000	32	Flash memory
0x4002_1000	33	DMA channel mutiplexer
0x4002_2000	34	—
0x4002_3000	35	—
0x4002_4000	36	—
0x4002_5000	37	—
0x4002_6000	38	—
0x4002_7000	39	—
0x4002_8000	40	—
0x4002_9000	41	Random Number Generator (RNGA)
0x4002_A000	42	—
0x4002_B000	43	—
0x4002_C000	44	SPI 0
0x4002_D000	45	SPI 1
0x4002_E000	46	—
0x4002_F000	47	I2S 0
0x4003_0000	48	—
0x4003_1000	49	—
0x4003_2000	50	CRC
0x4003_3000	51	—
0x4003_4000	52	—
0x4003_5000	53	USB DCD
0x4003_6000	54	Programmable delay block (PDB)
0x4003_7000	55	Periodic interrupt timers (PIT)
0x4003_8000	56	FlexTimer (FTM) 0
0x4003_9000	57	FlexTimer (FTM) 1
0x4003_A000	58	FlexTimer (FTM) 2
0x4003_B000	59	Analog-to-digital converter (ADC) 0
0x4003_C000	60	—
0x4003_D000	61	Real-time clock (RTC)
0x4003_E000	62	VBAT register file
0x4003_F000	63	—
0x4004_0000	64	Low-power timer (LPTMR)
0x4004_1000	65	System register file
0x4004_2000	66	—
0x4004_3000	67	—
0x4004_4000	68	—

Table continues on the next page...

Table 11-3. Peripheral bridge 0 slot assignments (continued)

System 32-bit base address	Slot number	Module
0x4004_5000	69	—
0x4004_6000	70	—
0x4004_7000	71	SIM low-power logic
0x4004_8000	72	System integration module (SIM)
0x4004_9000	73	Port A multiplexing control
0x4004_A000	74	Port B multiplexing control
0x4004_B000	75	Port C multiplexing control
0x4004_C000	76	Port D multiplexing control
0x4004_D000	77	Port E multiplexing control
0x4004_E000	78	—
0x4004_F000	79	—
0x4005_0000	80	—
0x4005_1000	81	—
0x4005_2000	82	Software watchdog
0x4005_3000	83	—
0x4005_4000	84	—
0x4005_5000	85	—
0x4005_6000	86	—
0x4005_7000	87	—
0x4005_8000	88	—
0x4005_9000	89	—
0x4005_A000	90	—
0x4005_B000	91	—
0x4005_C000	92	—
0x4005_D000	93	—
0x4005_E000	94	—
0x4005_F000	95	—
0x4006_0000	96	—
0x4006_1000	97	External watchdog
0x4006_2000	98	Carrier modulator timer (CMT)
0x4006_3000	99	—
0x4006_4000	100	Multi-purpose Clock Generator (MCG)
0x4006_5000	101	System oscillator (OSC)
0x4006_6000	102	I^{2}C 0
0x4006_7000	103	I^{2}C 1
0x4006_8000	104	—
0x4006_9000	105	—
0x4006_A000	106	UART 0
0x4006_B000	107	UART 1

Table continues on the next page...

Table 11-3. Peripheral bridge 0 slot assignments (continued)

System 32-bit base address	Slot number	Module
0x4006_C000	108	UART 2
0x4006_D000	109	—
0x4006_E000	110	—
0x4006_F000	111	—
0x4007_0000	112	—
0x4007_1000	113	—
0x4007_2000	114	USB OTG FS/LS
0x4007_3000	115	Analog comparator (CMP) / 6-bit digital-to-analog converter (DAC)
0x4007_4000	116	—
0x4007_5000	117	—
0x4007_6000	118	—
0x4007_7000	119	—
0x4007_8000	120	—
0x4007_9000	121	—
0x4007_A000	122	—
0x4007_B000	123	—
0x4007_C000	124	Low-leakage wakeup unit (LLWU)
0x4007_D000	125	Power management controller (PMC)
0x4007_E000	126	System Mode controller (SMC)
0x4007_F000	127	Reset Control Module (RCM)
0x400F_F000		GPIO controller

11.6 Private Peripheral Bus (PPB) memory map

The PPB is part of the defined ARM bus architecture and provides access to select processor-local modules. These resources are only accessible from the core; other system masters do not have access to them.

Table 11-4. PPB memory map

System 32-bit Address Range	Resource
0xE000_0000–0xE000_0FFF	Instrumentation Trace Macrocell (ITM)
0xE000_1000–0xE000_1FFF	Data Watchpoint and Trace (DWT)
0xE000_2000–0xE000_2FFF	Flash Patch and Breakpoint (FPB)
0xE000_3000–0xE000_DFFF	Reserved
0xE000_E000–0xE000_EFFF	System Control Space (SCS) (for NVIC)
0xE000_F000–0xE003_FFFF	Reserved

Table continues on the next page...

Table 11-4. PPB memory map (continued)

System 32-bit Address Range	Resource
0xE004_0000–0xE004_0FFF	Trace Port Interface Unit (TPIU)
0xE004_1000–0xE004_1FFF	Embedded Trace Macrocell (ETM)
0xE004_2000–0xE004_2FFF	Reserved
0xE004_3000–0xE004_3FFF	Reserved
0xE004_4000–0xE007_FFFF	Reserved
0xE008_0000–0xE008_0FFF	Miscellaneous Control Module (MCM)
0xE008_1000–0xE008_1FFF	Memory Mapped Cryptographic Acceleration Unit (MMCAU)
0xE008_2000–0xE00F_EFFF	Reserved
0xE00F_F000–0xE00F_FFFF	ROM Table - allows auto-detection of debug components

Chapter 12
MCU: Clock Distribution

12.1 Introduction

The MCG module controls which clock source is used to derive the system clocks. The clock generation logic divides the selected clock source into a variety of clock domains, including the clocks for the system bus masters, system bus slaves, and flash memory . The clock generation logic also implements module-specific clock gating to allow granular shutoff of modules.

The primary clocks for the system are generated from the MCGOUTCLK clock. The clock generation circuitry provides several clock dividers that allow different portions of the device to be clocked at different frequencies. This allows for trade-offs between performance and power dissipation.

Various modules, such as the USB OTG Controller, have module-specific clocks that can be generated from the MCGPLLCLK, or MCGFLLCLK clock. In addition, there are various other module-specific clocks that have other alternate sources. Clock selection for most modules is controlled by the SOPT registers in the SIM module.

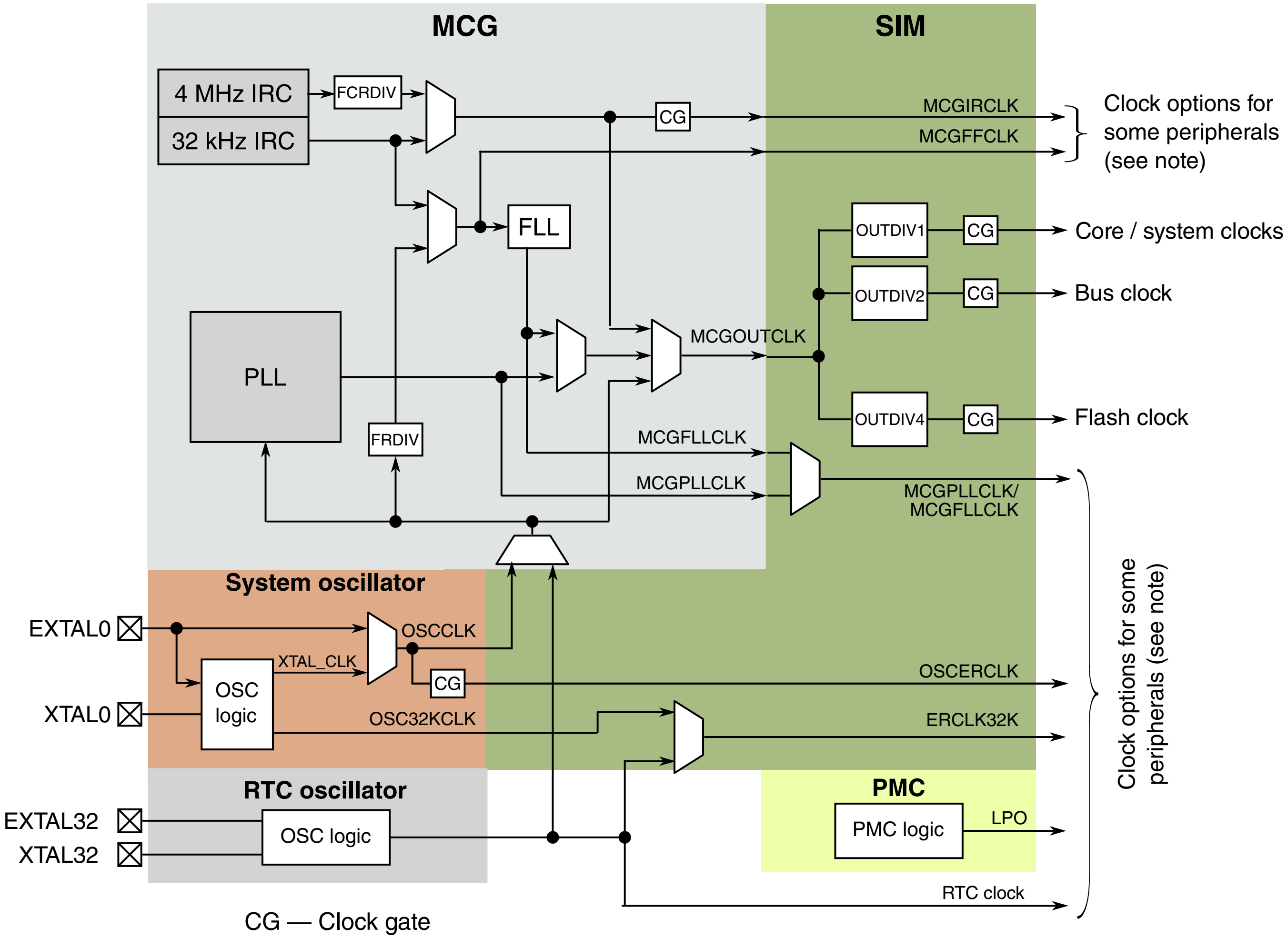
12.2 Programming model

The selection and multiplexing of system clock sources is controlled and programmed via the MCG module. The setting of clock dividers and module clock gating for the system are programmed via the SIM module. Reference those sections for detailed register and bit descriptions.

12.3 High-Level device clocking diagram

The following system oscillator, MCG, and SIM module registers control the multiplexers, dividers, and clock gates shown in the below figure:

	OSC	MCG	SIM
Multiplexers	MCG_C*x*	MCG_C*x*	SIM_SOPT1, SIM_SOPT2
Dividers	—	MCG_C*x*	SIM_CLKDIV*x*
Clock gates	OSC_CR	MCG_C1	SIM_SCGC*x*



CG — Clock gate

Note: See subsequent sections for details on where these clocks are used.

Figure 12-1. Clocking diagram

12.4 Clock definitions

The following table describes the clocks in the previous block diagram.

Clock name	Description
Core clock	MCGOUTCLK divided by OUTDIV1 clocks the ARM Cortex-M4 core

Table continues on the next page...

Clock name	Description
System clock	MCGOUTCLK divided by OUTDIV1 clocks the crossbar switch and bus masters directly connected to the crossbar. In addition, this clock is used for UART0 and UART1.
Bus clock	MCGOUTCLK divided by OUTDIV2 clocks the bus slaves and peripheral (excluding memories)
Flash clock	MCGOUTCLK divided by OUTDIV4 clocks the flash memory
MCGIRCLK	MCG output of the slow or fast internal reference clock
MCGFFCLK	MCG output of the slow internal reference clock or a divided MCG external reference clock.
MCGOUTCLK	MCG output of either IRC, MCGFLLCLK MCGPLLCLK, or MCG's external reference clock that sources the core, system, bus, and flash clock. It is also an option for the debug trace clock.
MCGFLLCLK	MCG output of the FLL. MCGFLLCLK may clock some modules.
MCGPLLCLK	MCG output of the PLL. MCGPLL0CLK may clock some modules.
OSCCLK	System oscillator output of the internal oscillator or sourced directly from EXTAL
OSCERCLK	System oscillator output sourced from OSCCLKthat may clock some on-chip modules
OSC32KCLK	System oscillator 32kHz output
ERCLK32K	Clock source for some modules that is chosen as OSC32KCLK or the RTC clock.
RTC clock	RTC oscillator output for the RTC module and the DryIce module
LPO	PMC 1kHz output

12.4.1 Device clock summary

The following table provides more information regarding the on-chip clocks.

Table 12-1. Clock Summary

Clock name	Run mode clock frequency	VLPR mode clock frequency	Clock source	Clock is disabled when…
MCGOUTCLK	Up to 50 MHz	Up to 4 MHz	MCG	In all stop modes
Core clock	Up to 50 MHz	Up to 4 MHz	MCGOUTCLK clock divider	In all wait and stop modes
System clock	Up to 50 MHz	Up to 4 MHz	MCGOUTCLK clock divider	In all stop modes
Bus clock	Up to 50 MHz	Up to 4 MHz	MCGOUTCLK clock divider	In all stop modes
Flash clock	Up to 25 MHz	Up to 1 MHz in BLPE, Up to 800 kHz in BLPI	MCGOUTCLK clock divider	In all stop modes

Table continues on the next page...

Table 12-1. Clock Summary (continued)

Clock name	Run mode clock frequency	VLPR mode clock frequency	Clock source	Clock is disabled when…
Internal reference (MCGIRCLK)	30-40 kHz or 4 MHz	4 MHz only	MCG	MCG_C1[IRCLKEN] cleared, Stop mode and MCG_C1[IREFSTEN] cleared, or VLPS/LLS/VLLS mode
External reference (OSCERCLK)	Up to 50 MHz (bypass), 30-40 kHz, or 3-32 MHz (crystal)	Up to 4 MHz (bypass), 30-40 kHz (low-range crystal) or Up to 4 MHz (high-range crystal)	System OSC	System OSC's OSC_CR[ERCLKEN] cleared, or Stop mode and OSC_CR[EREFSTEN] cleared
External reference 32kHz (ERCLK32K)	30-40 kHz	30-40 kHz	System OSC or RTC OSC depending on SIM_SOPT1[OSC32KSEL]	System OSC's OSC_CR[ERCLKEN] cleared or RTC's RTC_CR[OSCE] cleared
RTC_CLKOUT	1 Hz or 32 kHz	1 Hz or 32 kHz	RTC clock	Clock is disabled in LLS and VLLSx modes
LPO	1 kHz	1 kHz	PMC	in VLLS0
USB FS clock	48 MHz	N/A	MCGPLLCLK or MCGFLLCLK with fractional clock divider	USB FS OTG is disabled
I2S master clock	Up to 25 MHz	N/A	System clock , MCGPLLCLK , OSCERCLK with fractional clock divider, or I2S_CLKIN	I^{2}S is disabled
TRACE clock	Up to 50 MHz	Up to 4 MHz	System clock or MCGOUTCLK	Trace is disabled

12.5 Internal clocking requirements

The clock dividers are programmed via the SIM module's CLKDIV registers. Each divider is programmable from a divide-by-1 through divide-by-16 setting. The following requirements must be met when configuring the clocks for this device:

1. The core and system clock frequencies must be 50 MHz or slower.
2. The bus clock frequency must be programmed to 50 MHz or less and an integer divide of the core clock.

3. The flash clock frequency must be programmed to 25 MHz or less and an integer divide of the bus clock.

The following are a few of the more common clock configurations for this device:

Option 1:

Clock	Frequency
Core clock	50 MHz
System clock	50 MHz
Bus clock	50 MHz
Flash clock	25 MHz

12.5.1 Clock divider values after reset

Each clock divider is programmed via the SIM module's CLKDIV*n* registers. The flash memory's FTF_FOPT[LPBOOT] bit controls the reset value of the core clock, system clock, bus clock, and flash clock dividers as shown below:

FTF_FOPT [LPBOOT]	Core/system clock	Bus clock	Flash clock	Description
0	0x7 (divide by 8)	0x7 (divide by 8)	0xF (divide by 16)	Low power boot
1	0x0 (divide by 1)	0x0 (divide by 1)	0x1 (divide by 2)	Fast clock boot

This gives the user flexibility for a lower frequency, low-power boot option. The flash erased state defaults to fast clocking mode, since where the low power boot (FTF_FOPT[LPBOOT]) bit resides in flash is logic 1 in the flash erased state.

To enable the low power boot option program FTF_FOPT[LPBOOT] to zero. During the reset sequence, if LPBOOT is cleared, the system is in a slow clock configuration. Upon any system reset, the clock dividers return to this configurable reset state.

12.5.2 VLPR mode clocking

The clock dividers cannot be changed while in VLPR mode. They must be programmed prior to entering VLPR mode to guarantee:

- the core/system and bus clocks are less than or equal to 4 MHz, and
- the flash memory clock is less than or equal to 1 MHz

NOTE

When the MCG is in BLPI and clocking is derived from the Fast IRC, the clock divider controls, MCG_SC[FCRDIV] and SIM_CLKDIV1[OUTDIV4], must be programmed such that the resulting flash clock nominal frequency is 800 kHz or less. In this case, one example of correct configuration is MCG_SC[FCRDIV]=000b and SIM_CLKDIV1[OUTDIV4]=0100b, resulting in a divide by 5 setting.

12.6 Clock Gating

The clock to each module can be individually gated on and off using the SIM module's SCGC*x* registers. These bits are cleared after any reset, which disables the clock to the corresponding module to conserve power. Prior to initializing a module, set the corresponding bit in SCGC*x* register to enable the clock. Before turning off the clock, make sure to disable the module.

Any bus access to a peripheral that has its clock disabled generates an error termination.

12.7 Module clocks

The following table summarizes the clocks associated with each module.

Table 12-2. Module clocks

Module	Bus interface clock	Internal clocks	I/O interface clocks
Core modules			
ARM Cortex-M4 core	System clock	Core clock	—
NVIC	System clock	—	—
DAP	System clock	—	—
ITM	System clock	—	—
ETM	System clock	TRACE clock	TRACE_CLKOUT
cJTAG, JTAGC	—	—	JTAG_CLK
System modules			
DMA	System clock	—	—
DMA Mux	Bus clock	—	—
Port control	Bus clock	LPO	—
Crossbar Switch	System clock	—	—
Peripheral bridges	System clock	Bus clock, Flash clock	—

Table continues on the next page...

Table 12-2. Module clocks (continued)

Module	Bus interface clock	Internal clocks	I/O interface clocks
LLWU, PMC, SIM, RCM	Flash clock	LPO	—
Mode controller	Flash clock	—	—
MCM	System clock	—	—
EWM	Bus clock	LPO	—
Watchdog timer	Bus clock	LPO	—
Clocks			
MCG	Bus clock	MCGOUTCLK, MCGPLLCLK, MCGFLLCLK, MCGIRCLK, OSCERCLK	—
OSC	Bus clock	OSCERCLK	—
Memory and memory interfaces			
Flash Controller	System clock	Flash clock	—
Flash memory	Flash clock	—	—
EzPort	System clock	—	EZP_CLK
Security			
CRC	Bus clock	—	—
MMCAU	System clock	—	—
RNGA	Bus clock	—	—
Analog			
ADC	Bus clock	OSCERCLK	—
CMP	Bus clock	—	—
Timers			
PDB	Bus clock	—	—
FlexTimers	Bus clock	MCGFFCLK	FTM_CLKIN*x*
PIT	Bus clock	—	—
LPTMR	Flash clock	LPO, OSCERCLK, MCGIRCLK, ERCLK32K	—
CMT	Bus clock	—	—
RTC	Flash clock	EXTAL32	—
Communication interfaces			
USB FS OTG	System clock	USB FS clock	—
USB DCD	Bus clock	—	—
DSPI	Bus clock	—	DSPI_SCK
I^{2}C	Bus clock	—	I2C_SCL
UART0, UART1	System clock	—	—
UART2	Bus clock	—	—
I^{2}S	Bus clock	I^{2}S master clock	I2S_TX_BCLK, I2S_RX_BCLK
Human-machine interfaces			
GPIO	System clock	—	—

12.7.1 PMC 1-kHz LPO clock

The Power Management Controller (PMC) generates a 1-kHz clock that is enabled in all modes of operation, including all low power modes. This 1-kHz source is commonly referred to as LPO clock or 1-kHz LPO clock.

12.7.2 WDOG clocking

The WDOG may be clocked from two clock sources as shown in the following figure.

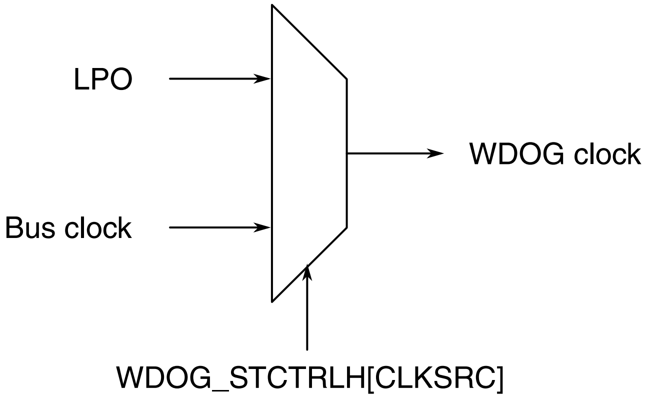


Figure 12-2. WDOG clock generation

12.7.3 Debug trace clock

The debug trace clock source can be clocked as shown in the following figure.

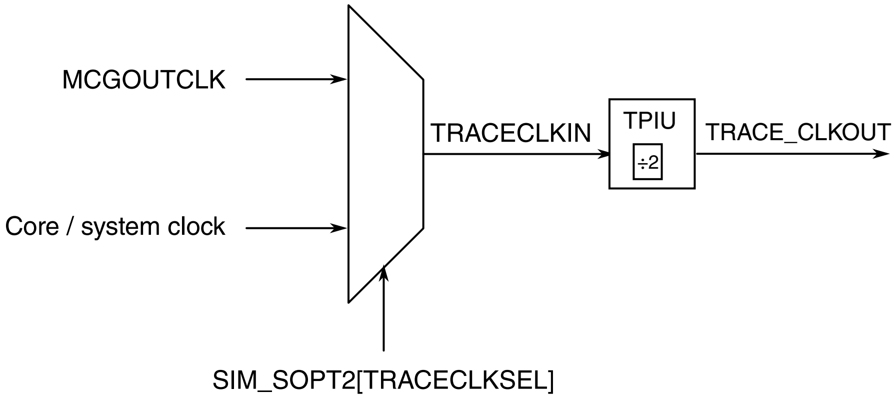


Figure 12-3. Trace clock generation

NOTE

The trace clock frequency observed at the TRACE_CLKOUT pin will be half that of the selected clock source.

12.7.4 PORT digital filter clocking

The digital filters in each of the PORT*x* modules can be clocked as shown in the following figure.

NOTE

In stop mode, the digital input filters are bypassed unless they are configured to run from the 1 kHz LPO clock source.

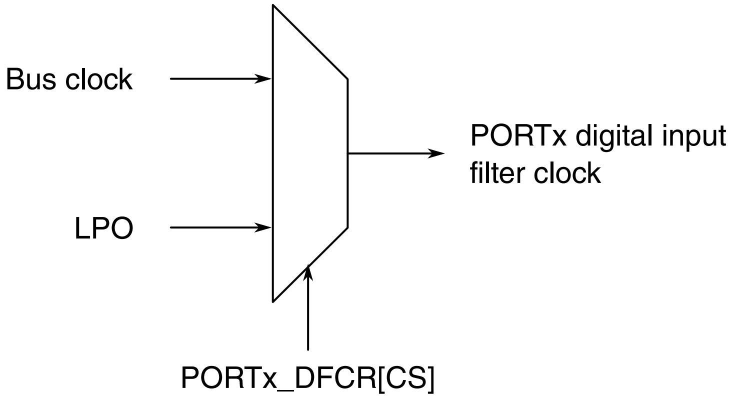


Figure 12-4. PORT*x* digital input filter clock generation

12.7.5 LPTMR clocking

The prescaler and glitch filters in each of the LPTMR*x* modules can be clocked as shown in the following figure.

NOTE

The chosen clock must remain enabled if the LPTMR*x* is to continue operating in all required low-power modes.

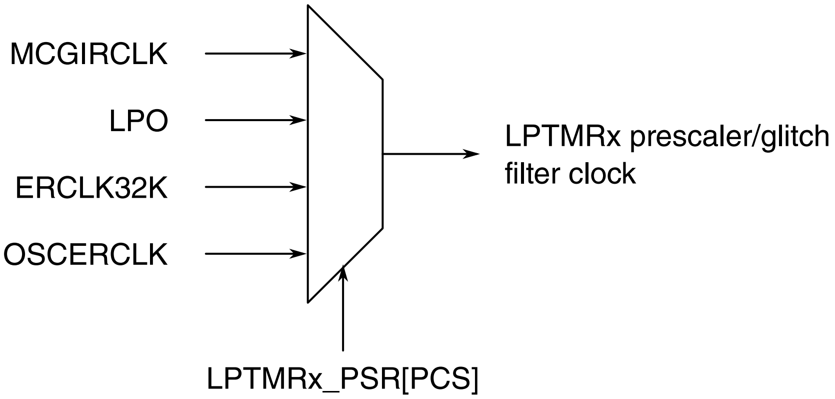


Figure 12-5. LPTMRx prescaler/glitch filter clock generation

12.7.6 USB FS OTG Controller clocking

NOTE

For the USB FS OTG controller to operate, the minimum system clock frequency is 20 MHz.

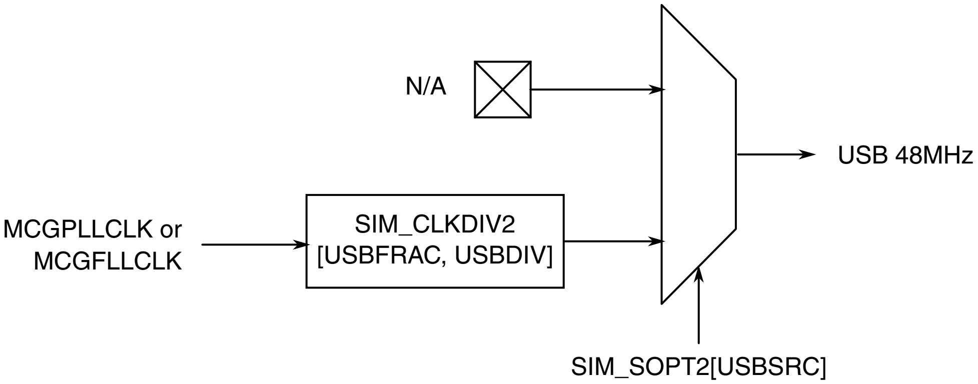


Figure 12-6. USB 48 MHz clock source

NOTE

The MCGFLLCLK does not meet the USB jitter specifications for certification.

12.7.7 UART clocking

UART0 and UART1 modules operate from the core/system clock, which provides higher performance level for these modules. All other UART modules operate from the bus clock.

12.7.8 I^{2}S/SAI clocking

The audio master clock (MCLK) is used to generate the bit clock when the receiver or transmitter is configured for an internally generated bit clock. The audio master clock can also be output to or input from a pin. The transmitter and receiver have the same audio master clock inputs.

Each SAI peripheral can control the input clock selection, pin direction and divide ratio of one audio master clock.

The I^{2}S/SAI transmitter and receiver support asynchronous bit clocks (BCLKs) that can be generated internally from the audio master clock or supplied externally. The module also supports the option for synchronous operation between the receiver and transmitter product.

The transmitter and receiver can independently select between the bus clock and the audio master clock to generate the bit clock.

The MCLK and BCLK source options appear in the following figure.

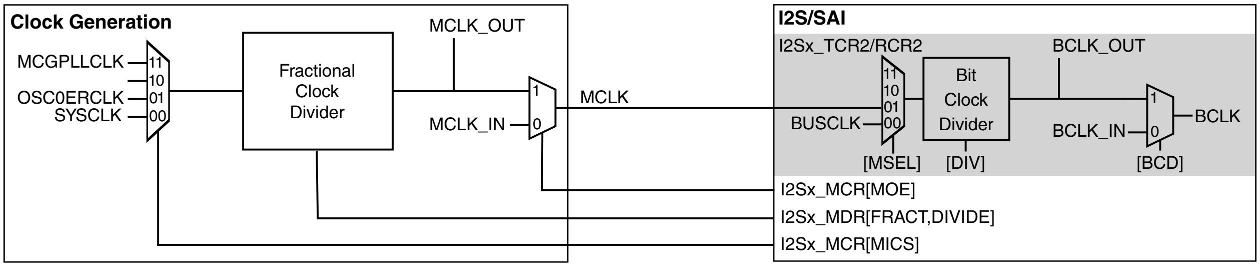


Figure 12-7. I^{2}S/SAI clock generation

Chapter 13
MCU: Reset and Boot

13.1 Introduction

The following reset sources are supported in this MCU:

Table 13-1. Reset sources

Reset sources	Description
POR reset	• Power-on reset (POR)
System resets	• External pin reset (PIN) • Low-voltage detect (LVD) • Computer operating properly (COP) watchdog reset • Low leakage wakeup (LLWU) reset • Multipurpose clock generator loss of clock (LOC) reset • Multipurpose clock generator loss of lock (LOL) reset • Stop mode acknowledge error (SACKERR) • Software reset (SW) • Lockup reset (LOCKUP) • EzPort reset • MDM DAP system reset
Debug reset	• JTAG reset • nTRST reset
Tamper Detect	• Tamper Reset

Each of the system reset sources has an associated bit in the system reset status (SRS) registers. See the Reset Control Module for register details.

The MCU exits reset in functional mode that is controlled by $\overline{\text{EZP\_CS}}$ pin to select between the single chip (default) or serial flash programming (EzPort) modes. See Boot options for more details.

13.2 Reset

This section discusses basic reset mechanisms and sources. Some modules that cause resets can be configured to cause interrupts instead. Consult the individual peripheral chapters for more information.

13.2.1 Power-on reset (POR)

When power is initially applied to the MCU or when the supply voltage drops below the power-on reset re-arm voltage level (V_{POR}), the POR circuit causes a POR reset condition.

As the supply voltage rises, the LVD circuit holds the MCU in reset until the supply has risen above the LVD low threshold (V_{LVDL}). The POR and LVD bits in SRS0 register are set following a POR.

13.2.2 System reset sources

Resetting the MCU provides a way to start processing from a known set of initial conditions. System reset begins with the on-chip regulator in full regulation and system clocking generation from an internal reference. When the processor exits reset, it performs the following:

- Reads the start SP (SP_main) from vector-table offset 0
- Reads the start PC from vector-table offset 4
- LR is set to 0xFFFF_FFFF

The on-chip peripheral modules are disabled and the non-analog I/O pins are initially configured as disabled. The pins with analog functions assigned to them default to their analog function after reset.

During and following a reset, the JTAG pins have their associated input pins configured as:

- TDI in pull-up (PU)
- TCK in pull-down (PD)
- TMS in PU

and associated output pin configured as:

- TDO with no pull-down or pull-up

Note that the nTRST signal is initially configured as disabled, however once configured to its JTAG functionality its associated input pin is configured as:

- nTRST in PU

13.2.2.1 External pin reset (PIN)

On this device, $\overline{\text{RESET}}$ is a dedicated pin. This pin is open drain and has an internal pullup device. Asserting $\overline{\text{RESET}}$ wakes the device from any mode. During a pin reset, the RCM's SRS0[PIN] bit is set.

13.2.2.1.1 Reset pin filter

The $\overline{\text{RESET}}$ pin filter supports filtering from both the 1 kHz LPO clock and the bus clock. A separate filter is implemented for each clock source. In stop and VLPS mode operation, this logic either switches to bypass operation or has continued filtering operation depending on the filtering mode selected. In low leakage stop modes, a separate LPO filter in the LLWU can continue filtering the $\overline{\text{RESET}}$ pin.

The RPFC[RSTFLTSS], RPFC[RSTFLTSRW], and RPFW[RSTFLTSEL] fields in the reset control (RCM) register set control this functionality; see the RCM chapter. The filters are asynchronously reset by Chip POR. The reset value for each filter assumes the RESET pin is negated.

The two clock options for the $\overline{\text{RESET}}$ pin filter when the chip is not in low leakage modes are the LPO (1 kHz) and bus clock. For low leakage modes VLLS3, VLLS2, VLLS1, VLLS0, the LLWU provides control (in the LLWU_RST register) of an optional fixed digital filter running the LPO. When entering VLLS0, the RESET pin filter is disabled and bypassed.

The LPO filter has a fixed filter value of 3. Due to a synchronizer on the input data, there is also some associated latency (2 cycles). As a result, 5 cycles are required to complete a transition from low to high or high to low.

The bus filter initializes to off (logic 1) when the bus filter is not enabled. The bus clock is used when the filter selects bus clock, and the number of counts is controlled by the RCM's RPFW[RSTFLTSEL] field.

13.2.2.2 Low-voltage detect (LVD)

The chip includes a system for managing low voltage conditions to protect memory contents and control MCU system states during supply voltage variations. The system consists of a power-on reset (POR) circuit and an LVD circuit with a user-selectable trip voltage. The LVD system is always enabled in normal run, wait, or stop mode. The LVD system is disabled when entering VLPx, LLS, or VLLSx modes.

The LVD can be configured to generate a reset upon detection of a low voltage condition by setting the PMC's LVDSC1[LVDRE] bit to 1. The low voltage detection threshold is determined by the PMC's LVDSC1[LVDV] field. After an LVD reset has occurred, the LVD system holds the MCU in reset until the supply voltage has risen above the low voltage detection threshold. The RCM's SRS0[LVD] bit is set following either an LVD reset or POR.

13.2.2.3 Computer operating properly (COP) watchdog timer

The computer operating properly (COP) watchdog timer (WDOG) monitors the operation of the system by expecting periodic communication from the software. This communication is generally known as servicing (or refreshing) the COP watchdog. If this periodic refreshing does not occur, the watchdog issues a system reset. The COP reset causes the RCM's SRS0[WDOG] bit to set.

13.2.2.4 Low leakage wakeup (LLWU)

The LLWU module provides the means for a number of external pins, the $\overline{\text{RESET}}$ pin, and a number of internal peripherals to wake the MCU from low leakage power modes. The LLWU module is functional only in low leakage power modes.

- In LLS mode, only the $\overline{\text{RESET}}$ pin via the LLWU can generate a system reset.
- In VLLSx modes, all enabled inputs to the LLWU can generate a system reset.

After a system reset, the LLWU retains the flags indicating the input source of the last wakeup until the user clears them.

NOTE

Some flags are cleared in the LLWU and some flags are required to be cleared in the peripheral module. Refer to the individual peripheral chapters for more information.

13.2.2.5 Multipurpose clock generator loss-of-clock (LOC)

The MCG module supports an external reference clock.

If the C6[CME] bit in the MCG module is set, the clock monitor is enabled. If the external reference falls below f_{loc_low} or f_{loc_high}, as controlled by the C2[RANGE] field in the MCG module, the MCU resets. The RCM's SRS0[LOC] bit is set to indicate this reset source.

NOTE

To prevent unexpected loss of clock reset events, all clock monitors should be disabled before entering any low power modes, including VLPR and VLPW.

13.2.2.6 MCG loss-of-lock (LOL) reset

The MCG includes a PLL loss-of-lock detector. The detector is enabled when configured for PEE and lock has been achieved. If the MCG_C8[LOLRE] bit in the MCG module is set and the PLL lock status bit (MCG_S[LOLS0]) becomes set, the MCU resets. The RCM_SRS0[LOL] bit is set to indicate this reset source.

NOTE

This reset source does not cause a reset if the chip is in any stop mode.

13.2.2.7 Stop mode acknowledge error (SACKERR)

This reset is generated if the core attempts to enter stop mode, but not all modules acknowledge stop mode within 1025 cycles of the 1 kHz LPO clock.

A module might not acknowledge the entry to stop mode if an error condition occurs. The error can be caused by a failure of an external clock input to a module.

13.2.2.8 Software reset (SW)

The SYSRESETREQ bit in the NVIC application interrupt and reset control register can be set to force a software reset on the device. (See ARM's NVIC documentation for the full description of the register fields, especially the VECTKEY field requirements.) Setting SYSRESETREQ generates a software reset request. This reset forces a system reset of all major components except for the debug module. A software reset causes the RCM's SRS1[SW] bit to set.

13.2.2.9 Lockup reset (LOCKUP)

The LOCKUP gives immediate indication of seriously errant kernel software. This is the result of the core being locked because of an unrecoverable exception following the activation of the processor’s built in system state protection hardware.

The LOCKUP condition causes a system reset and also causes the RCM's SRS1[LOCKUP] bit to set.

13.2.2.10 Tamper detect reset (TAMPER)

A tamper detect condition can optionally cause a system reset and also causes the RCM's SRS1[TAMPER] bit to set.

13.2.2.11 EzPort reset

The EzPort supports a system reset request via EzPort signaling. The EzPort generates a system reset request following execution of a Reset Chip (RESET) command via the EzPort interface. This method of reset allows the chip to boot from flash memory after it has been programmed by an external source. The EzPort is enabled or disabled by the $\overline{\text{EZP\_CS}}$ pin.

An EzPort reset causes the RCM's SRS1[EZPT] bit to set.

13.2.2.12 MDM-AP system reset request

Set the system reset request bit in the MDM-AP control register to initiate a system reset. This is the primary method for resets via the JTAG/SWD interface. The system reset is held until this bit is cleared.

Set the core hold reset bit in the MDM-AP control register to hold the core in reset as the rest of the chip comes out of system reset.

13.2.3 MCU Resets

A variety of resets are generated by the MCU to reset different modules.

13.2.3.1 VBAT POR

The VBAT POR asserts on a VBAT POR reset source. It affects only the modules within the VBAT power domain: RTC, DryIce, and VBAT Register File. These modules are not affected by the other reset types.

13.2.3.2 POR Only

The POR Only reset asserts on the POR reset source only. It resets the PMC and System Register File.

The POR Only reset also causes all other reset types (except VBAT POR) to occur.

13.2.3.3 Chip POR not VLLS

The Chip POR not VLLS reset asserts on POR and LVD reset sources. It resets parts of the SMC and SIM. It also resets the LPTMR.

The Chip POR not VLLS reset also causes these resets to occur: Chip POR, Chip Reset not VLLS, and Chip Reset (including Early Chip Reset).

13.2.3.4 Chip POR

The Chip POR asserts on POR, LVD, and VLLS Wakeup reset sources. It resets the Reset Pin Filter registers and parts of the SIM and MCG.

The Chip POR also causes the Chip Reset (including Early Chip Reset) to occur.

13.2.3.5 Chip Reset not VLLS

The Chip Reset not VLLS reset asserts on all reset sources except a VLLS Wakeup that does not occur via the $\overline{\text{RESET}}$ pin. It resets parts of the SMC, LLWU, and other modules that remain powered during VLLS mode.

The Chip Reset not VLLS reset also causes the Chip Reset (including Early Chip Reset) to occur.

13.2.3.6 Early Chip Reset

The Early Chip Reset asserts on all reset sources. It resets only the flash memory module. It negates before flash memory initialization begins ("earlier" than when the Chip Reset negates).

13.2.3.7 Chip Reset

Chip Reset asserts on all reset sources and only negates after flash initialization has completed and the $\overline{\text{RESET}}$ pin has also negated. It resets the remaining modules (the modules not reset by other reset types).

13.2.4 Reset Pin

For all reset sources except a VLLS Wakeup that does not occur via the $\overline{\text{RESET}}$ pin, the $\overline{\text{RESET}}$ pin is driven low by the MCU for at least 128 bus clock cycles and until flash initialization has completed.

After flash initialization has completed, the $\overline{\text{RESET}}$ pin is released, and the internal Chip Reset negates after the $\overline{\text{RESET}}$ pin is pulled high. Keeping the $\overline{\text{RESET}}$ pin asserted externally delays the negation of the internal Chip Reset.

13.2.5 Debug resets

The following sections detail the debug resets available on the device.

13.2.5.1 JTAG reset

The JTAG module generate a system reset when certain IR codes are selected. This functional reset is asserted when EzPort, EXTEST, HIGHZ and CLAMP instructions are active. The reset source from the JTAG module is released when any other IR code is selected. A JTAG reset causes the RCM's SRS1[JTAG] bit to set.

13.2.5.2 nTRST reset

The nTRST pin causes a reset of the JTAG logic when asserted. Asserting the nTRST pin allows the debugger to gain control of the TAP controller state machine (after exiting LLS or VLLSx) without resetting the state of the debug modules.

The nTRST pin does not cause a system reset.

13.2.5.3 Resetting the Debug subsystem

Use the CDBGRSTREQ bit within the SWJ-DP CTRL/STAT register to reset the debug modules. However, as explained below, using the CDBGRSTREQ bit does not reset all debug-related registers.

CDBGRSTREQ resets the debug-related registers within the following modules:

- SWJ-DP
- AHB-AP
- ETM
- TPIU
- MDM-AP (MDM control and status registers)

CDBGRSTREQ does not reset the debug-related registers within the following modules:

- CM4 core (core debug registers: DHCSR, DCRSR, DCRDR, DEMCR)
- FPB
- DWT
- ITM
- NVIC
- Crossbar bus switch
- AHB-AP[1]
- Private peripheral bus[1]

13.3 Boot

This section describes the boot sequence, including sources and options.

13.3.1 Boot sources

[1] CDBGRSTREQ does not affect AHB resources so that debug resources on the private peripheral bus are available during System Reset.

This device only supports booting from internal flash. Any secondary boot must go through an initialization sequence in flash.

13.3.2 Boot options

The device's functional mode is controlled by the state of the EzPort chip select ($\overline{\text{EZP\_CS}}$) pin during reset.

The device can be in single chip (default) or serial flash programming mode (EzPort). While in single chip mode the device can be in run or various low power modes mentioned in Power mode transitions.

Table 13-2. Mode select decoding

EzPort chip select ($\overline{\text{EZP\_CS}}$)	Description
0	Serial flash programming mode (EzPort)
1	Single chip (default)

13.3.3 FOPT boot options

The flash option register (FOPT) in the flash memory module allows the user to customize the operation of the MCU at boot time. The register contains read-only bits that are loaded from the NVM's option byte in the flash configuration field. The user can reprogram the option byte in flash to change the FOPT values that are used for subsequent resets. For more details on programming the option byte, refer to the flash memory chapter.

The MCU uses the FOPT register bits to configure the device at reset as shown in the following table.

Table 13-3. Flash Option Register Bit Definitions

Bit Num	Field	Value	Definition
7-3	Reserved	Reserved for future expansion.	
2	NMI_DIS	Enable/disable control for the NMI function.	
		0	NMI interrupts are always blocked. The associated pin continues to default to NMI pin controls with internal pullup enabled.
		1	NMI pin/interrupts reset default to enabled.
1	EZPORT_DIS	Enable/disable EzPort function.	

Table continues on the next page...

Table 13-3. Flash Option Register Bit Definitions (continued)

Bit Num	Field	Value	Definition
		0	EzPort operation is disabled. The device always boots to normal CPU execution and the state of $\overline{\text{EZP\_CS}}$ signal during reset is ignored. This option avoids inadvertent resets into EzPort mode if the $\overline{\text{EZP\_CS}}$/NMI pin is used for its NMI function.
		1	EzPort operation is enabled. The state of $\overline{\text{EZP\_CS}}$ pin during reset determines if device enters EzPort mode.
0	LPBOOT		Control the reset value of OUTDIVx values in SIM_CLKDIV1 register. Larger divide value selections produce lower average power consumption during POR, VLLSx recoveries and reset sequencing and after reset exit.
		0	Low-power boot: OUTDIVx values in SIM_CLKDIV1 register are auto-configured at reset exit for higher divide values that produce lower power consumption at reset exit. • Core and system clock divider (OUTDIV1) and bus clock divider (OUTDIV2) are 0x7 (divide by 8) • Flash clock divider (OUTDIV4)is 0xF (divide by 16)
		1	Normal boot: OUTDIVx values in SIM_CLKDIV1 register are auto-configured at reset exit for higher frequency values that produce faster operating frequencies at reset exit. • Core and system clock divider (OUTDIV1) and bus clock divider (OUTDIV2) are 0x0 (divide by 1) • Flash clock divider (OUTDIV4)is 0x1 (divide by 2)

13.3.4 Boot sequence

At power up, the on-chip regulator holds the system in a POR state until the input supply is above the POR threshold. The system continues to be held in this static state until the internally regulated supplies have reached a safe operating voltage as determined by the LVD. The Mode Controller reset logic then controls a sequence to exit reset.

1. A system reset is held on internal logic, the $\overline{\text{RESET}}$ pin is driven out low, and the MCG is enabled in its default clocking mode.
2. Required clocks are enabled (Core Clock, System Clock, Flash Clock, and any Bus Clocks that do not have clock gate control).
3. The system reset on internal logic continues to be held, but the Flash Controller is released from reset and begins initialization operation while the Mode Control logic continues to drive the $\overline{\text{RESET}}$ pin out low for a count of ~128 Bus Clock cycles.
4. The $\overline{\text{RESET}}$ pin is released, but the system reset of internal logic continues to be held until the Flash Controller finishes initialization. EzPort mode is selected instead of the normal CPU execution if $\overline{\text{EZP\_CS}}$ is low when the internal reset is deasserted. EzPort mode can be disabled by programming the FOPT[EZPORT_DIS] field in the Flash Memory module.

5. When Flash Initialization completes, the $\overline{\text{RESET}}$ pin is observed. If $\overline{\text{RESET}}$ continues to be asserted (an indication of a slow rise time on the $\overline{\text{RESET}}$ pin or external drive in low), the system continues to be held in reset. Once the $\overline{\text{RESET}}$ pin is detected high, the system is released from reset.
6. At release of system reset, clocking is switched to a slow clock if the FOPT[LPBOOT] field in the Flash Memory module is configured for Low Power Boot
7. When the system exits reset, the processor sets up the stack, program counter (PC), and link register (LR). The processor reads the start SP (SP_main) from vector-table offset 0. The core reads the start PC from vector-table offset 4. LR is set to 0xFFFF_FFFF. What happens next depends on the NMI input and the FOPT[NMI_DIS] field in the Flash Memory module:
    - If the NMI input is high or the NMI function is disabled in the NMI_DIS field, the CPU begins execution at the PC location.
    - If the NMI input is low and the NMI function is enabled in the NMI_DIS field, this results in an NMI interrupt. The processor executes an Exception Entry and reads the NMI interrupt handler address from vector-table offset 8. The CPU begins execution at the NMI interrupt handler.
8. If FlexNVM is enabled, the flash controller continues to restore the FlexNVM data. This data is not available immediately out of reset and the system should not access this data until the flash controller completes this initialization step as indicated by the EEERDY flag.

Subsequent system resets follow this reset flow beginning with the step where system clocks are enabled.

Chapter 14
MCU: Power Management

14.1 Introduction

This chapter describes the various chip power modes and functionality of the individual modules in these modes.

14.2 Power Modes Description

The power management controller (PMC) provides multiple power options to allow the user to optimize power consumption for the level of functionality needed.

Depending on the stop requirements of the user application, a variety of stop modes are available that provide state retention, partial power down or full power down of certain logic and/or memory. I/O states are held in all modes of operation. The following table compares the various power modes available.

For Run and VLPR mode there is a corresponding wait and stop mode. Wait modes are similar to ARM sleep modes. Stop modes (VLPS, STOP) are similar to ARM sleep deep mode. The very low power run (VLPR) operating mode can drastically reduce runtime power when the maximum bus frequency is not required to handle the application needs.

The three primary modes of operation are run, wait and stop. The WFI instruction invokes both wait and stop modes for the chip. The primary modes are augmented in a number of ways to provide lower power based on application needs.

Table 14-1. Chip power modes

Chip mode	Description	Core mode	Normal recovery method
Normal run	Allows maximum performance of chip. Default mode out of reset; on-chip voltage regulator is on.	Run	-

Table continues on the next page...

Table 14-1. Chip power modes (continued)

Chip mode	Description	Core mode	Normal recovery method
Normal Wait - via WFI	Allows peripherals to function while the core is in sleep mode, reducing power. NVIC remains sensitive to interrupts; peripherals continue to be clocked.	Sleep	Interrupt
Normal Stop - via WFI	Places chip in static state. Lowest power mode that retains all registers while maintaining LVD protection. NVIC is disabled; AWIC is used to wake up from interrupt; peripheral clocks are stopped.	Sleep Deep	Interrupt
VLPR (Very Low Power Run)	On-chip voltage regulator is in a low power mode that supplies only enough power to run the chip at a reduced frequency. Reduced frequency Flash access mode (1 MHz); LVD off; internal oscillator provides a low power 4 MHz source for the core, the bus and the peripheral clocks.	Run	Interrupt
VLPW (Very Low Power Wait) -via WFI	Same as VLPR but with the core in sleep mode to further reduce power; NVIC remains sensitive to interrupts (FCLK = ON). On-chip voltage regulator is in a low power mode that supplies only enough power to run the chip at a reduced frequency.	Sleep	Interrupt
VLPS (Very Low Power Stop)-via WFI	Places chip in static state with LVD operation off. Lowest power mode with ADC and pin interrupts functional. Peripheral clocks are stopped, but LPTimer, RTC, CMP can be used. NVIC is disabled (FCLK = OFF); AWIC is used to wake up from interrupt. On-chip voltage regulator is in a low power mode that supplies only enough power to run the chip at a reduced frequency. All SRAM is operating (content retained and I/O states held).	Sleep Deep	Interrupt
LLS (Low Leakage Stop)	State retention power mode. Most peripherals are in state retention mode (with clocks stopped), but LLWU, LPTimer, RTC, CMP can be used. NVIC is disabled; LLWU is used to wake up. **NOTE:** The LLWU interrupt must not be masked by the interrupt controller to avoid a scenario where the system does not fully exit stop mode on an LLS recovery. All SRAM is operating (content retained and I/O states held).	Sleep Deep	Wakeup Interrupt[1]
VLLS3 (Very Low Leakage Stop3)	Most peripherals are disabled (with clocks stopped), but LLWU, LPTimer, RTC, CMP can be used. NVIC is disabled; LLWU is used to wake up. SRAM_U and SRAM_L remain powered on (content retained and I/O states held).	Sleep Deep	Wakeup Reset
VLLS2 (Very Low Leakage Stop2)	Most peripherals are disabled (with clocks stopped), but LLWU, LPTimer, RTC, CMP can be used. NVIC is disabled; LLWU is used to wake up. SRAM_L is powered off. A portion of SRAM_U remains powered on (content retained and I/O states held).	Sleep Deep	Wakeup Reset[2]
VLLS1 (Very Low Leakage Stop1)	Most peripherals are disabled (with clocks stopped), but LLWU, LPTimer, RTC, CMP can be used. NVIC is disabled; LLWU is used to wake up. All of SRAM_U and SRAM_L are powered off. The 32-byte system register file and the 32-byteVBAT register file remain powered for customer-critical data.	Sleep Deep	Wakeup Reset[2]

Table continues on the next page...

Table 14-1. Chip power modes (continued)

Chip mode	Description	Core mode	Normal recovery method
VLLS0 (Very Low Leakage Stop 0)	Most peripherals are disabled (with clocks stopped), but LLWU and RTC can be used. NVIC is disabled; LLWU is used to wake up. All of SRAM_U and SRAM_L are powered off. The 32-byte system register file and the 32-byte VBAT register file remain powered for customer-critical data. The POR detect circuit can be optionally powered off.	Sleep Deep	Wakeup Reset[2]
BAT (backup battery only)	The chip is powered down except for the VBAT supply. The RTC and the 32-byte VBAT register file for customer-critical data remain powered.	Off	Power-up Sequence

1. Resumes normal run mode operation by executing the LLWU interrupt service routine.
2. Follows the reset flow with the LLWU interrupt flag set for the NVIC.

14.3 Entering and exiting power modes

The WFI instruction invokes wait and stop modes for the chip. The processor exits the low-power mode via an interrupt. The Nested Vectored Interrupt Controller (NVIC) describes interrupt operation and what peripherals can cause interrupts.

NOTE

The WFE instruction can have the side effect of entering a low-power mode, but that is not its intended usage. See ARM documentation for more on the WFE instruction.

Recovery from VLLSx is through the wake-up Reset event. The chip wake-ups from VLLSx by means of reset, an enabled pin or enabled module. See the table "LLWU inputs" in the LLWU configuration section for a list of the sources.

The wake-up flow from VLLSx is through reset. The wakeup bit in the SRS registers in the RCM is set indicating that the chip is recovering from a low power mode. Code execution begins; however, the I/O pins are held in their pre low power mode entry states, and the system oscillator and MCG registers are reset (even if EREFSTEN had been set before entering VLLSx). Software must clear this hold by writing a 1 to the ACKISO bit in the Regulator Status and Control Register in the PMC module.

NOTE

To avoid unwanted transitions on the pins, software must re-initialize the I/O pins to their pre-low-power mode entry states *before* releasing the hold.

If the oscillator was configured to continue running during VLLSx modes, it must be re-configured before the ACKISO bit is cleared. The oscillator configuration within the MCG is cleared after VLLSx recovery and the oscillator will stop when ACKISO is cleared unless the register is re-configured.

14.4 Power mode transitions

The following figure shows the power mode transitions. Any reset always brings the chip back to the normal run state. In run, wait, and stop modes active power regulation is enabled. The VLPx modes offer a lower power operating mode than normal modes. VLPR and VLPW are limited in frequency. The LLS and VLLSx modes are the lowest power stop modes based on amount of logic or memory that is required to be retained by the application.

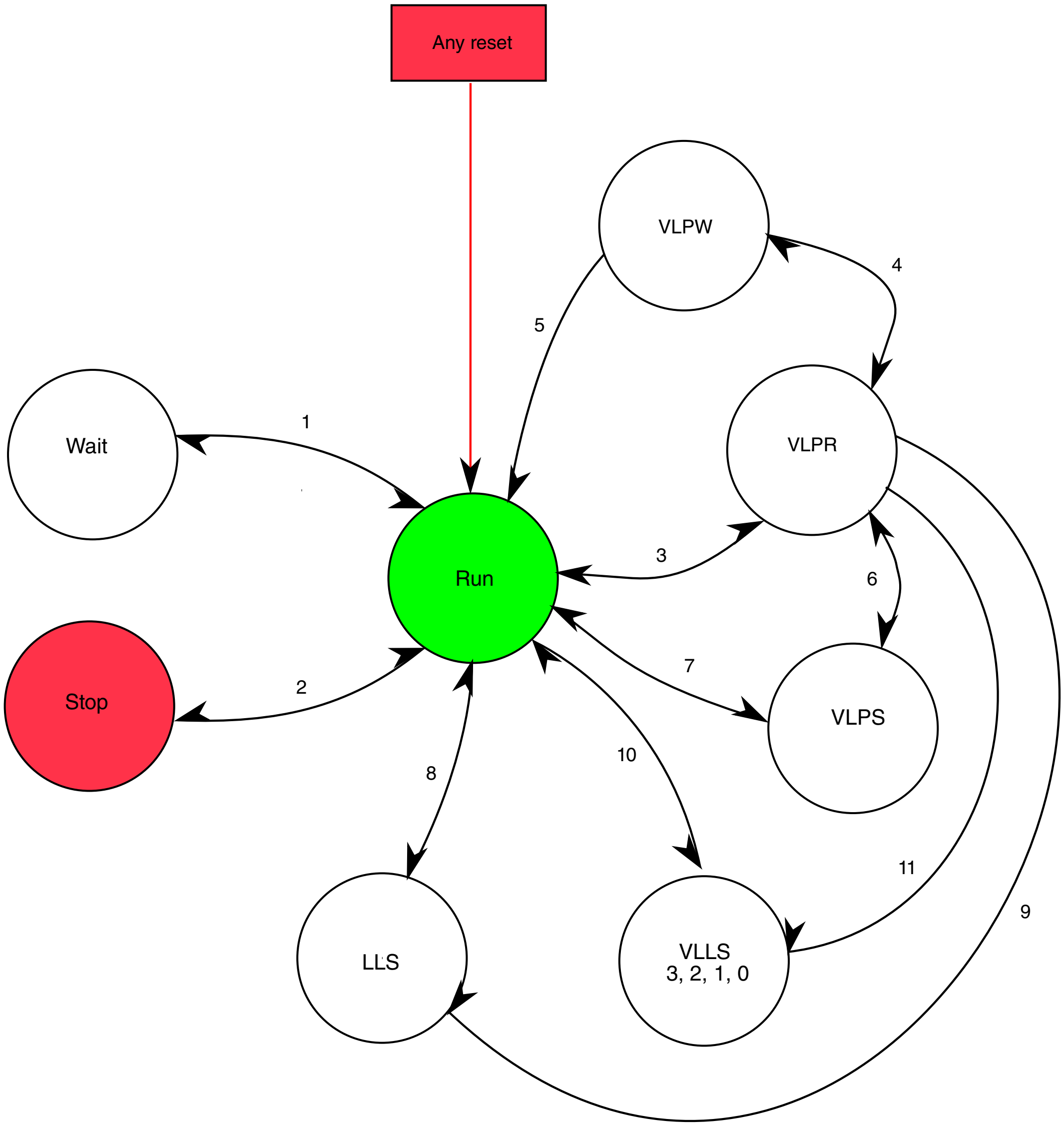


Figure 14-1. Power mode state transition diagram

14.5 Power modes shutdown sequencing

When entering stop or other low-power modes, the clocks are shut off in an orderly sequence to safely place the chip in the targeted low-power state. All low-power entry sequences are initiated by the core executing an WFI instruction. The ARM core's outputs, SLEEPDEEP and SLEEPING, trigger entry to the various low-power modes:

- System level wait and VLPW modes equate to: SLEEPING & $\overline{\text{SLEEPDEEP}}$
- All other low power modes equate to: SLEEPING & SLEEPDEEP

When entering the non-wait modes, the chip performs the following sequence:

- Shuts off Core Clock and System Clock to the ARM Cortex-M4 core immediately.
- Polls stop acknowledge indications from the non-core crossbar masters (DMA), supporting peripherals (SPI, PIT, RNG) and the Flash Controller for indications that System Clocks, Bus Clock and/or Flash Clock need to be left enabled to complete a previously initiated operation, effectively stalling entry to the targeted low power mode. When all acknowledges are detected, System Clock, Bus Clock and Flash Clock are turned off at the same time.
- MCG and Mode Controller shut off clock sources and/or the internal supplies driven from the on-chip regulator as defined for the targeted low power mode.

In wait modes, most of the system clocks are not affected by the low power mode entry. The Core Clock to the ARM Cortex-M4 core is shut off. Some modules support stop-in-wait functionality and have their clocks disabled under these configurations.

The debugger modules support a transition from stop, wait, VLPS, and VLPW back to a halted state when the debugger is enabled. This transition is initiated by setting the Debug Request bit in MDM-AP control register. As part of this transition, system clocking is re-established and is equivalent to normal run/VLPR mode clocking configuration.

14.6 Module Operation in Low Power Modes

The following table illustrates the functionality of each module while the chip is in each of the low power modes. (Debug modules are discussed separately; see Debug in Low Power Modes.) Number ratings (such as 2 MHz and 1 Mbit/s) represent the maximum frequencies or maximum data rates per mode. Also, these terms are used:

- FF = Full functionality. In VLPR and VLPW the system frequency is limited, but if a module does not have a limitation in its functionality, it is still listed as FF.
- static = Module register states and associated memories are retained.
- powered = Memory is powered to retain contents.
- low power = Flash has a low power state that retains configuration registers to support faster wakeup.
- OFF = Modules are powered off; module is in reset state upon wakeup.
- wakeup = Modules can serve as a wakeup source for the chip.

Table 14-2. Module operation in low power modes

Modules	Stop	VLPR	VLPW	VLPS	LLS	VLLSx
Core modules						
NVIC	static	FF	FF	static	static	OFF
System modules						
Mode Controller	FF	FF	FF	FF	FF	FF

Table continues on the next page...

Table 14-2. Module operation in low power modes (continued)

Modules	Stop	VLPR	VLPW	VLPS	LLS	VLLSx
LLWU[1]	static	static	static	static	FF	FF[2]
Regulator	ON	low power	low power	low power	low power	low power in VLLS2/3, OFF in VLLS0/1
LVD	ON	disabled	disabled	disabled	disabled	disabled
Brown-out Detection	ON	ON	ON	ON	ON	ON in VLLS1/2/3, optionally disabled in VLLS0[3]
DMA	static	FF	FF	static	static	OFF
Watchdog	FF	FF	FF	FF	static	OFF
EWM	static	FF	static	static	static	OFF
Clocks						
1kHz LPO	ON	ON	ON	ON	ON	ON in VLLS1/2/3, OFF in VLLS0
System oscillator (OSC)	OSCERCLK optional	OSCERCLK max of 4 MHz crystal	OSCERCLK max of 4 MHz crystal	OSCERCLK max of 4 MHz crystal	limited to low range/low power	limited to low range/low power in VLLS1/2/3, OFF in VLLS0
MCG	static - MCGIRCLK optional; PLL optionally on but gated	4 MHz IRC	4 MHz IRC	static - no clock output	static - no clock output	OFF
Core clock	OFF	4 MHz max	OFF	OFF	OFF	OFF
System clock	OFF	4 MHz max	4 MHz max	OFF	OFF	OFF
Bus clock	OFF	4 MHz max	4 MHz max	OFF	OFF	OFF
Memory and memory interfaces						
Flash	powered	1 MHz max access - no pgm	low power	low power	OFF	OFF
Portion of SRAM_U[4]	low power	low power	low power	low power	low power	low power in VLLS3,2; otherwise OFF
Remaining SRAM_U and all of SRAM_L	low power	low power	low power	low power	low power	low power in VLLS3; otherwise OFF
FlexMemory	low power	low power[5]	low power	low power	low power	OFF
VBAT Register file	powered	powered	powered	powered	powered	powered
System Register files	powered	powered	powered	powered	powered	powered
EzPort	disabled	disabled	disabled	disabled	disabled	disabled
Communication interfaces						
USB FS/LS	static	static	static	static	static	OFF

Table continues on the next page...

Table 14-2. Module operation in low power modes (continued)

Modules	Stop	VLPR	VLPW	VLPS	LLS	VLLSx
USB DCD	static	FF	FF	static	static	OFF
USB Voltage Regulator	optional	optional	optional	optional	optional	optional
UART	static, wakeup on edge	125 kbit/s	125 kbit/s	static, wakeup on edge	static	OFF
SPI	static	1 Mbit/s	1 Mbit/s	static	static	OFF
I^{2}C	static, address match wakeup	100 kbit/s	100 kbit/s	static, address match wakeup	static	OFF
I^{2}S	FF with external clock	FF	FF	FF with external clock[7]	static	OFF
Security						
CRC	static	FF	FF	static	static	OFF
RNG	static	FF	static	static	static	OFF
DryIce[6]	FF	FF	FF	FF	FF	FF
Timers						
FTM	static	FF	FF	static	static	OFF
PIT	static	FF	FF	static	static	OFF
PDB	static	FF	FF	static	static	OFF
LPTMR	FF	FF	FF	FF	FF	FF[8]
RTC - 32kHz OSC[6]	FF	FF	FF	FF	FF	FF[9]
CMT	static	FF	FF	static	static	OFF
Analog						
16-bit ADC	ADC internal clock only	FF	FF	ADC internal clock only	static	OFF
CMP[10]	HS or LS level compare	FF	FF	HS or LS level compare	LS level compare	LS level compare in VLLS1/2/3, OFF in VLLS0
6-bit DAC	static	FF	FF	static	static	static
Human-machine interfaces						
GPIO	wakeup	FF	FF	wakeup	static, pins latched	OFF, pins latched

1. Using the LLWU module, the external pins available for this chip do not require the associated peripheral function to be enabled. It only requires the function controlling the pin (GPIO or peripheral) to be configured as an input to allow a transition to occur to the LLWU.
2. Since LPO clock source is disabled, filters will be bypassed during VLLS0
3. The VLLSCTRL[PORPO] bit in the SMC module controls this option.
4. A KB portion of SRAM_U block is left powered on in low power mode VLLS2.
5. FlexRAM enabled as EEPROM is not writable in VLPR and writes are ignored. Read accesses to FlexRAM as EEPROM while in VLPR are allowed. There are no access restrictions for FlexRAM configured as traditional RAM.
6. These components remain powered in BAT power mode.
7. Use an externally generated bit clock or an externally generated audio master clock (including EXTAL).
8. System OSC and LPO clock sources are not available in VLLS0
9. RTC_CLKOUT is not available.

10. CMP in stop or VLPS supports high speed or low speed external pin to pin or external pin to DAC compares. CMP in LLS or VLLSx only supports low speed external pin to pin or external pin to DAC compares. Windowed, sampled & filtered modes of operation are not available while in stop, VLPS, LLS, or VLLSx modes.

14.7 Clock Gating

To conserve power, the clocks to most modules can be turned off using the SCGCx registers in the SIM module. These bits are cleared after any reset, which disables the clock to the corresponding module. Prior to initializing a module, set the corresponding bit in the SCGCx register to enable the clock. Before turning off the clock, make sure to disable the module. For more details, refer to the clock distribution and SIM chapters.

Chapter 15
MCU: Security

15.1 Introduction

This device implements security based on the mode selected from the flash module. The following sections provide an overview of flash security and details the effects of security on non-flash modules.

15.2 Flash Security

The flash module provides security information to the MCU based on the state held by the FSEC[SEC] bits. The MCU, in turn, confirms the security request and limits access to flash resources. During reset, the flash module initializes the FSEC register using data read from the security byte of the flash configuration field.

NOTE

> The security features apply only to external accesses via debug and EzPort. CPU accesses to the flash are not affected by the status of FSEC.

In the unsecured state all flash commands are available to the programming interfaces (JTAG and EzPort), as well as user code execution of Flash Controller commands. When the flash is secured (FSEC[SEC] = 00, 01, or 11), programmer interfaces are only allowed to launch mass erase operations and have no access to memory locations.

Further information regarding the flash security options and enabling/disabling flash security is available in the Flash Memory Module.

15.3 Security Interactions with other Modules

The flash security settings are used by the SoC to determine what resources are available. The following sections describe the interactions between modules and the flash security settings or the impact that the flash security has on non-flash modules.

15.3.1 Security Interactions with EzPort

When flash security is active the MCU can still boot in EzPort mode. The EzPort holds the flash logic in NVM special mode and thus limits flash operation when flash security is active. While in EzPort mode and security is active, flash bulk erase (BE) can still be executed. The write FCCOB registers (WRFCCOB) command is limited to the mass erase (Erase All Blocks) and verify all 1s (Read 1s All Blocks) commands. Read accesses to internal memories via the EzPort are blocked when security is enabled.

The mass erase can be used to disable flash security, but all of the flash contents are lost in the process. A mass erase via the EzPort is allowed even when some memory locations are protected.

When mass erase has been disabled, mass erase via the EzPort is blocked and cannot be defeated.

15.3.2 Security Interactions with Debug

When flash security is active the JTAG port cannot access the memory resources of the MCU. Boundary scan chain operations work, but debugging capabilities are disabled so that the debug port cannot read flash contents.

Although most debug functions are disabled, the debugger can write to the Flash Mass Erase in Progress bit in the MDM-AP Control register to trigger a mass erase (Erase All Blocks) command. A mass erase via the debugger is allowed even when some memory locations are protected.

When mass erase is disabled, mass erase via the debugger is blocked.

Chapter 16
MCU: Debug

16.1 Introduction

This device's debug is based on the ARM coresight architecture and is configured in each device to provide the maximum flexibility as allowed by the restrictions of the pinout and other available resources.

Four debug interfaces are supported:

- IEEE 1149.1 JTAG
- IEEE 1149.7 JTAG (cJTAG)
- Serial Wire Debug (SWD)
- ARM Real-Time Trace Interface

The basic Cortex-M4 debug architecture is very flexible. The following diagram shows the topology of the core debug architecture and its components.

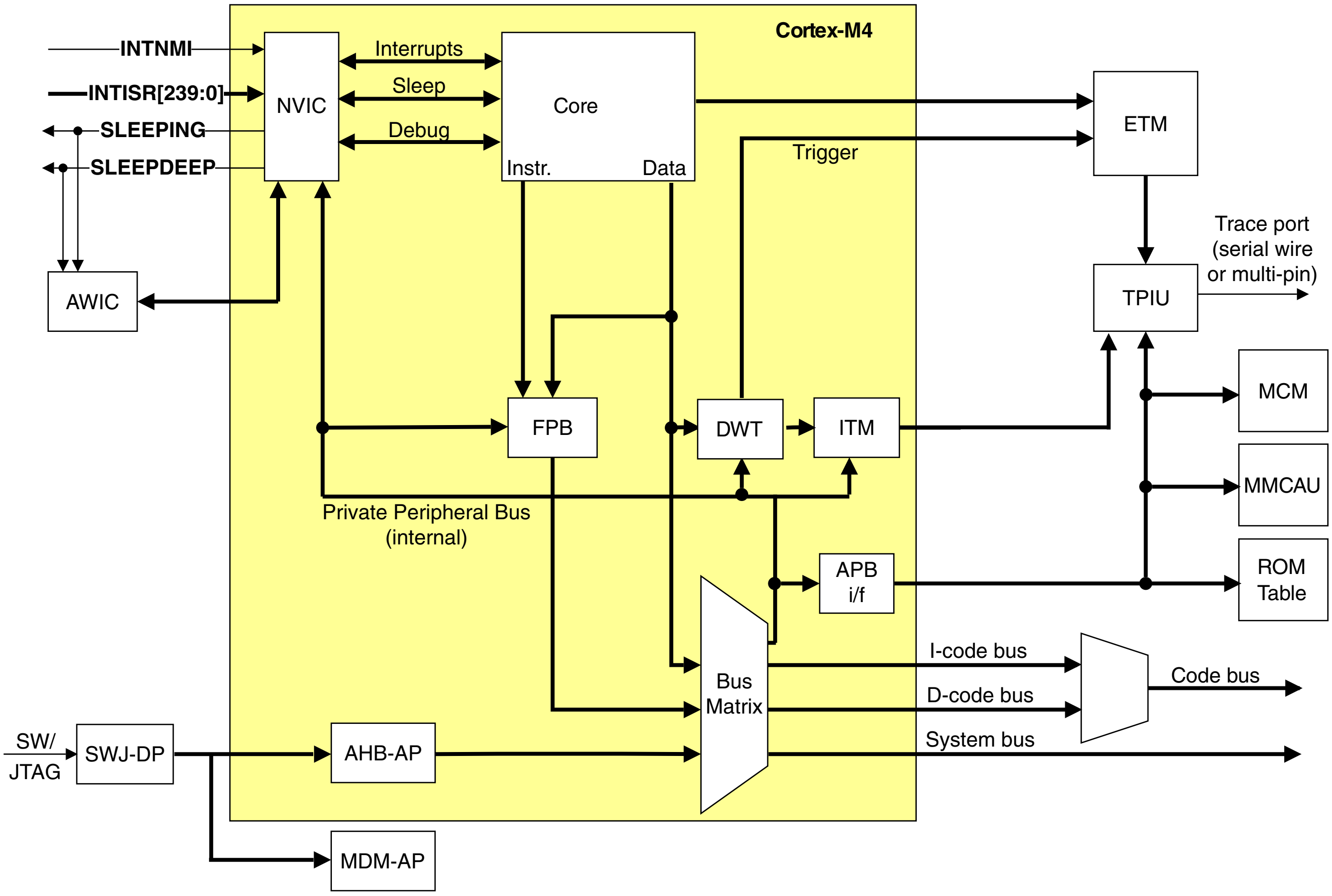


Figure 16-1. Cortex-M4 Debug Topology

The following table presents a brief description of each one of the debug components.

Table 16-1. Debug Components Description

Module	Description
SWJ-DP+ cJTAG	Modified Debug Port with support for SWD, JTAG, cJTAG
AHB-AP	AHB Master Interface from JTAG to debug module and SOC system memory maps
MDM-AP	Provides centralized control and status registers for an external debugger to control the device.
ROM Table	Identifies which debug IP is available.
Core Debug	Singlestep, Register Access, Run, Core Status
ETM (Embedded Trace Macrocell)	ETMv3.5 Architecture
ITM	S/W Instrumentation Messaging + Simple Data Trace Messaging + Watchpoint Messaging
DWT (Data and Address Watchpoints)	4 data and address watchpoints
FPB (Flash Patch and Breakpoints)	The FPB implements hardware breakpoints and patches code and data from code space to system space.

Table continues on the next page...

Table 16-1. Debug Components Description (continued)

Module	Description
	The FPB unit contains two literal comparators for matching against literal loads from Code space, and remapping to a corresponding area in System space. The FBP also contains six instruction comparators for matching against instruction fetches from Code space, and remapping to a corresponding area in System space. Alternatively, the six instruction comparators can individually configure the comparators to return a Breakpoint Instruction (BKPT) to the processor core on a match, so providing hardware breakpoint capability.
TPIU (Trace Port Inteface Unit)	Synchronous Mode (5-pin) = TRACE_D[3:0] + TRACE_CLKOUT Synchronous Mode (3-pin) = TRACE_D[1:0] + TRACE_CLKOUT Asynchronous Mode (1-pin) = TRACE_SWO (available on JTAG_TDO)
MCM (Miscellaneous Control Module)	The MCM provides miscellaneous control functions .

16.1.1 References

For more information on ARM debug components, see these documents:

- ARMv7-M Architecture Reference Manual
- ARM Debug Interface v5.1
- ARM CoreSight Architecture Specification
- ARM ETM Architecture Specification v3.5

16.2 The Debug Port

The configuration of the cJTAG module, JTAG controller, and debug port is illustrated in the following figure:

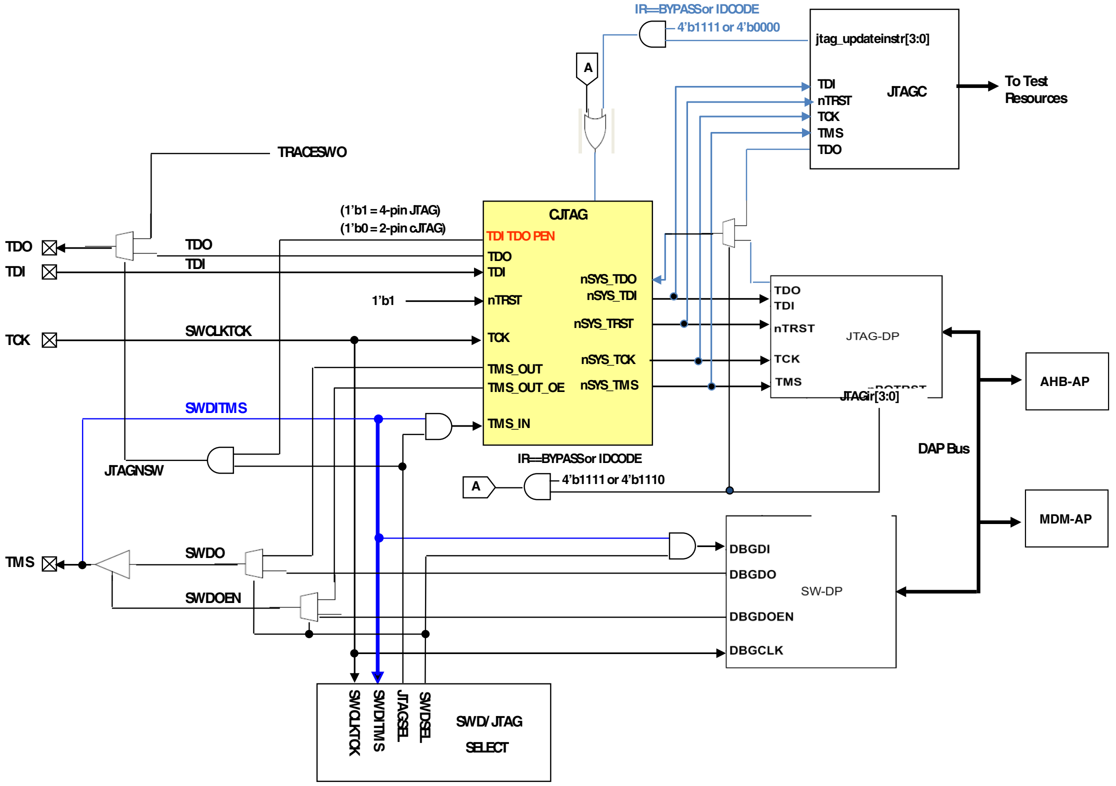


Figure 16-2. Modified Debug Port

The debug port comes out of reset in standard JTAG mode and is switched into either cJTAG or SWD mode by the following sequences. Once the mode has been changed, unused debug pins can be reassigned to any of their alternative muxed functions.

16.2.1 JTAG-to-SWD change sequence

1. Send more than 50 TCK cycles with TMS (SWDIO) =1
2. Send the 16-bit sequence on TMS (SWDIO) = 0111_1001_1110_0111 (MSB transmitted first)
3. Send more than 50 TCK cycles with TMS (SWDIO) =1

NOTE

See the ARM documentation for the CoreSight DAP Lite for restrictions.

16.2.2 JTAG-to-cJTAG change sequence

1. Reset the debug port

2. Set the control level to 2 via zero-bit scans
3. Execute the Store Format (STFMT) command (00011) to set the scan format register to 1149.7 scan format

16.3 Debug Port Pin Descriptions

The debug port pins default after POR to their JTAG functionality with the exception of JTAG_TRST_b and can be later reassigned to their alternate functionalities. In cJTAG and SWD modes JTAG_TDI and JTAG_TRST_b can be configured to alternate GPIO functions.

Table 16-2. Debug port pins

Pin Name	JTAG Debug Port		cJTAG Debug Port		SWD Debug Port		Internal Pull-up\Down
	Type	Description	Type	Description	Type	Description	
JTAG_TMS/ SWD_DIO	I/O	JTAG Test Mode Selection	I/O	cJTAG Data	I/O	Serial Wire Data	Pull-up
JTAG_TCLK/ SWD_CLK	I	JTAG Test Clock	I	cJTAG Clock	I	Serial Wire Clock	Pull-down
JTAG_TDI	I	JTAG Test Data Input	-	-	-	-	Pull-up
JTAG_TDO/ TRACE_SWO	O	JTAG Test Data Output	O	Trace output over a single pin	O	Trace output over a single pin	N/C
JTAG_TRST_b	I	JTAG Reset	I	cJTAG Reset	-	-	Pull-up

16.4 System TAP connection

The system JTAG controller is connected in parallel to the ARM TAP controller. The system JTAG controller IR codes overlay the ARM JTAG controller IR codes without conflict. Refer to the IR codes table for a list of the available IR codes. The output of the TAPs (TDO) are muxed based on the IR code which is selected. This design is fully JTAG compliant and appears to the JTAG chain as a single TAP. At power on reset, ARM's IDCODE (IR=4'b1110) is selected.

16.4.1 IR Codes

Table 16-3. JTAG Instructions

Instruction	Code[3:0]	Instruction Summary
IDCODE	0000	Selects device identification register for shift
SAMPLE/PRELOAD	0010	Selects boundary scan register for shifting, sampling, and preloading without disturbing functional operation
SAMPLE	0011	Selects boundary scan register for shifting and sampling without disturbing functional operation
EXTEST	0100	Selects boundary scan register while applying preloaded values to output pins and asserting functional reset
HIGHZ	1001	Selects bypass register while three-stating all output pins and asserting functional reset
CLAMP	1100	Selects bypass register while applying preloaded values to output pins and asserting functional reset
EZPORT	1101	Enables the EZPORT function for the SoC and asserts functional reset.
ARM_IDCODE	1110	ARM JTAG-DP Instruction
BYPASS	1111	Selects bypass register for data operations
Factory debug reserved	0101, 0110, 0111	Intended for factory debug only
ARM JTAG-DP Reserved	1000, 1010, 1011, 1110	These instructions will go the ARM JTAG-DP controller. Please look at ARM JTAG-DP documentation for more information on these instructions.
Reserved [3]	All other opcodes	Decoded to select bypass register

[3] The manufacturer reserves the right to change the decoding of reserved instruction codes in the future

16.5 JTAG status and control registers

Through the ARM Debug Access Port (DAP), the debugger has access to the status and control elements, implemented as registers on the DAP bus as shown in the following figure. These registers provide additional control and status for low power mode recovery and typical run-control scenarios. The status register bits also provide a means for the debugger to get updated status of the core without having to initiate a bus transaction across the crossbar switch, thus remaining less intrusive during a debug session.

It is important to note that these DAP control and status registers are not memory mapped within the system memory map and are only accessible via the Debug Access Port (DAP) using JTAG, cJTAG, or SWD. The MDM-AP is accessible as Debug Access Port 1 with the available registers shown in the table below.

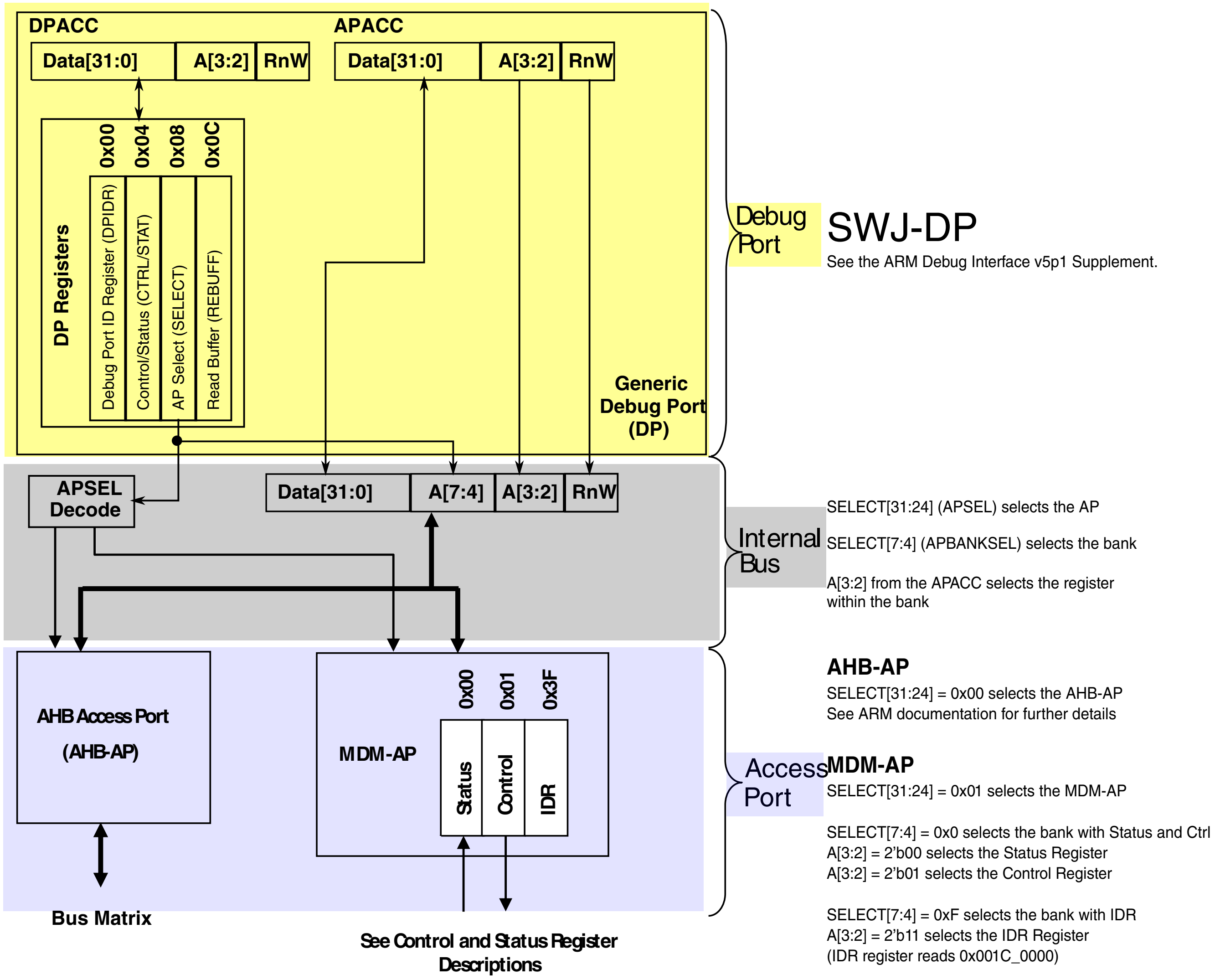
Table 16-4. MDM-AP Register Summary

Address	Register	Description

Table continues on the next page...

Table 16-4. MDM-AP Register Summary (continued)

0x0100_0000	Status	See MDM-AP Status Register
0x0100_0004	Control	See MDM-AP Control Register
0x0100_00FC	ID	Read-only identification register that always reads as 0x001C_0000



SWJ-DP

See the ARM Debug Interface v5p1 Supplement.

SELECT[31:24] (APSEL) selects the AP

SELECT[7:4] (APBANKSEL) selects the bank

A[3:2] from the APACC selects the register within the bank

AHB-AP

SELECT[31:24] = 0x00 selects the AHB-AP
See ARM documentation for further details

MDM-AP

SELECT[31:24] = 0x01 selects the MDM-AP

SELECT[7:4] = 0x0 selects the bank with Status and Ctrl
A[3:2] = 2'b00 selects the Status Register
A[3:2] = 2'b01 selects the Control Register

SELECT[7:4] = 0xF selects the bank with IDR
A[3:2] = 2'b11 selects the IDR Register
(IDR register reads 0x001C_0000)

Figure 16-3. MDM AP Addressing

16.5.1 MDM-AP Control Register

Table 16-5. MDM-AP Control register assignments

Bit	Name	Secure[1]	Description
0	Flash Mass Erase in Progress	Y	Set to cause mass erase. Cleared by hardware after mass erase operation completes. When mass erase is disabled (via MEEN and SEC settings), the erase request does not occur and the Flash Mass Erase in Progress bit continues to assert until the next system reset.
1	Debug Disable	N	Set to disable debug. Clear to allow debug operation. When set it overrides the C_DEBUGEN bit within the DHCSR and force disables Debug logic.
2	Debug Request	N	Set to force the Core to halt. If the Core is in a stop or wait mode, this bit can be used to wakeup the core and transition to a halted state.
3	System Reset Request	N	Set to force a system reset. The system remains held in reset until this bit is cleared.
4	Core Hold Reset	N	Configuration bit to control Core operation at the end of system reset sequencing. 0 Normal operation - release the Core from reset along with the rest of the system at the end of system reset sequencing. 1 Suspend operation - hold the Core in reset at the end of reset sequencing. Once the system enters this suspended state, clearing this control bit immediately releases the Core from reset and CPU operation begins.
5	VLLSx Debug Request (VLLDBGREQ)	N	Set to configure the system to be held in reset after the next recovery from a VLLSx mode. This bit holds the in reset when VLLSx modes are exited to allow the debugger time to re-initialize debug IP before the debug session continues. The Mode Controller captures this bit logic on entry to VLLSx modes. Upon exit from VLLSx modes, the Mode Controller will hold the in reset until VLLDBGACK is asserted. The VLLDBGREQ bit clears automatically due to the POR reset generated as part of the VLLSx recovery.
6	VLLSx Debug Acknowledge (VLLDBGACK)	N	Set to release a being held in reset following a VLLSx recovery This bit is used by the debugger to release the system reset when it is being held on VLLSx mode exit. The debugger re-initializes all debug IP and then assert this control bit to allow the Mode Controller to release the from reset and allow CPU operation to begin. The VLLDBGACK bit is cleared by the debugger or can be left set because it clears automatically due to the POR reset generated as part of the next VLLSx recovery.
7	LLS, VLLSx Status Acknowledge	N	Set this bit to acknowledge the DAP LLS and VLLS Status bits have been read. This acknowledge automatically clears the status bits. This bit is used by the debugger to clear the sticky LLS and VLLSx mode entry status bits. This bit is asserted and cleared by the debugger.

Table continues on the next page...

Table 16-5. MDM-AP Control register assignments (continued)

Bit	Name	Secure[1]	Description
8	Timestamp Disable	N	Set this bit to disable the 48-bit global trace timestamp counter during debug halt mode when the core is halted. 0 The timestamp counter continues to count assuming trace is enabled and the ETM is enabled. (default) 1 The timestamp counter freezes when the core has halted (debug halt mode).
9 – 31	Reserved for future use	N	

[1] Command available in secure mode

16.5.2 MDM-AP Status Register

Table 16-6. MDM-AP Status register assignments

Bit	Name	Description
0	Flash Mass Erase Acknowledge	The Flash Mass Erase Acknowledge bit is cleared after any system reset. The bit is also cleared at launch of a mass erase command due to write of Flash Mass Erase in Progress bit in MDM AP Control Register. The Flash Mass Erase Acknowledge is set after Flash control logic has started the mass erase operation. When mass erase is disabled (via MEEN and SEC settings), an erase request due to seting of Flash Mass Erase in Progress bit is not acknowledged.
1	Flash Ready	Indicate Flash has been initialized and debugger can be configured even if system is continuing to be held in reset via the debugger.
2	System Security	Indicates the security state. When secure, the debugger does not have access to the system bus or any memory mapped peripherals. This bit indicates when the part is locked and no system bus access is possible.
3	System Reset	Indicates the system reset state. 0 System is in reset 1 System is not in reset
4	Reserved	
5	Mass Erase Enable	Indicates if the MCU can be mass erased or not 0 Mass erase is disabled 1 Mass erase is enabled
6	Backdoor Access Key Enable	Indicates if the MCU has the backdoor access key enabled. 0 Disabled 1 Enabled
7	LP Enabled	Decode of LPLLSM control bits to indicate that VLPS, LLS, or VLLSx are the selected power mode the next time the ARM Core enters Deep Sleep. 0 Low Power Stop Mode is not enabled 1 Low Power Stop Mode is enabled

Table continues on the next page...

Table 16-6. MDM-AP Status register assignments (continued)

Bit	Name	Description
		Usage intended for debug operation in which Run to VLPS is attempted. Per debug definition, the system actually enters the Stop state. A debugger should interpret deep sleep indication (with SLEEPDEEP and SLEEPING asserted), in conjuntion with this bit asserted as the debugger-VLPS status indication.
8	Very Low Power Mode	Indicates current power mode is VLPx. This bit is not 'sticky' and should always represent whether VLPx is enabled or not. This bit is used to throttle JTAG TCK frequency up/down.
9	LLS Mode Exit	This bit indicates an exit from LLS mode has occurred. The debugger will lose communication while the system is in LLS (including access to this register). Once communication is reestablished, this bit indicates that the system had been in LLS. Since the debug modules held their state during LLS, they do not need to be reconfigured. This bit is set during the LLS recovery sequence. The LLS Mode Exit bit is held until the debugger has had a chance to recognize that LLS was exited and is cleared by a write of 1 to the LLS, VLLSx Status Acknowledge bit in MDM AP Control register.
10	VLLSx Modes Exit	This bit indicates an exit from VLLSx mode has occurred. The debugger will lose communication while the system is in VLLSx (including access to this register). Once communication is reestablished, this bit indicates that the system had been in VLLSx. Since the debug modules lose their state during VLLSx modes, they need to be reconfigured. This bit is set during the VLLSx recovery sequence. The VLLSx Mode Exit bit is held until the debugger has had a chance to recognize that a VLLS mode was exited and is cleared by a write of 1 to the LLS, VLLSx Status Acknowledge bit in MDM AP Control register.
11 – 15	Reserved for future use	Always read 0.
16	Core Halted	Indicates the Core has entered debug halt mode
17	Core SLEEPDEEP	Indicates the Core has entered a low power mode SLEEPING==1 and SLEEPDEEP==0 indicates wait or VLPW mode. SLEEPING==1 and SLEEPDEEP==1 indicates stop or VLPS mode.
18	Core SLEEPING	
19 – 31	Reserved for future use	Always read 0.

16.6 Debug Resets

The debug system receives the following sources of reset:

- JTAG_TRST_b from an external signal. This signal is optional and may not be available in all packages.
- Debug reset (CDBGRSTREQ bit within the SWJ-DP CTRL/STAT register) in the TCLK domain that allows the debugger to reset the debug logic.
- TRST asserted via the cJTAG escape command.
- System POR reset

Conversely the debug system is capable of generating system reset using the following mechanism:

- A system reset in the DAP control register which allows the debugger to hold the system in reset.
- SYSRESETREQ bit in the NVIC application interrupt and reset control register
- A system reset in the DAP control register which allows the debugger to hold the Core in reset.

16.7 AHB-AP

AHB-AP provides the debugger access to all memory and registers in the system, including processor registers through the NVIC. System access is independent of the processor status. AHB-AP does not do back-to-back transactions on the bus, so all transactions are non-sequential. AHB-AP can perform unaligned and bit-band transactions. AHB-AP transactions bypass the FPB, so the FPB cannot remap AHB-AP transactions. SWJ/SW-DP-initiated transaction aborts drive an AHB-AP-supported sideband signal called HABORT. This signal is driven into the Bus Matrix, which resets the Bus Matrix state, so that AHB-AP can access the Private Peripheral Bus for last ditch debugging such as read/stop/reset the core. AHB-AP transactions are little endian.

For a short period at the start of a system reset event the system security status is being determined and debugger access to all AHB-AP transactions is blocked. The MDM-AP Status register is accessible and can be monitored to determine when this initial period is completed. After this initial period, if system reset is held via assertion of the RESET pin, the debugger has access via the bus matrix to the private peripheral bus to configure the debug IP even while system reset is asserted. While in system reset, access to other memory and register resources, accessed over the Crossbar Switch, is blocked.

16.8 ITM

The ITM is an application-driven trace source that supports printf style debugging to trace Operating System (OS) and application events, and emits diagnostic system information. The ITM emits trace information as packets. There are four sources that can generate packets. If multiple sources generate packets at the same time, the ITM arbitrates the order in which packets are output. The four sources in decreasing order of priority are:

1. Software trace -- Software can write directly to ITM stimulus registers. This emits packets.

2. Hardware trace -- The DWT generates these packets, and the ITM emits them.
3. Time stamping -- Timestamps are emitted relative to packets. The ITM contains a 21-bit counter to generate the timestamp. The Cortex-M4 clock or the bitclock rate of the Serial Wire Viewer (SWV) output clocks the counter.
4. Global system timestamping. Timestamps can optionally be generated using a system-wide 48-bit count value. The same count value can be used to insert timestamps in the ETM trace stream, allowing coarse-grain correlation.

16.9 Core Trace Connectivity

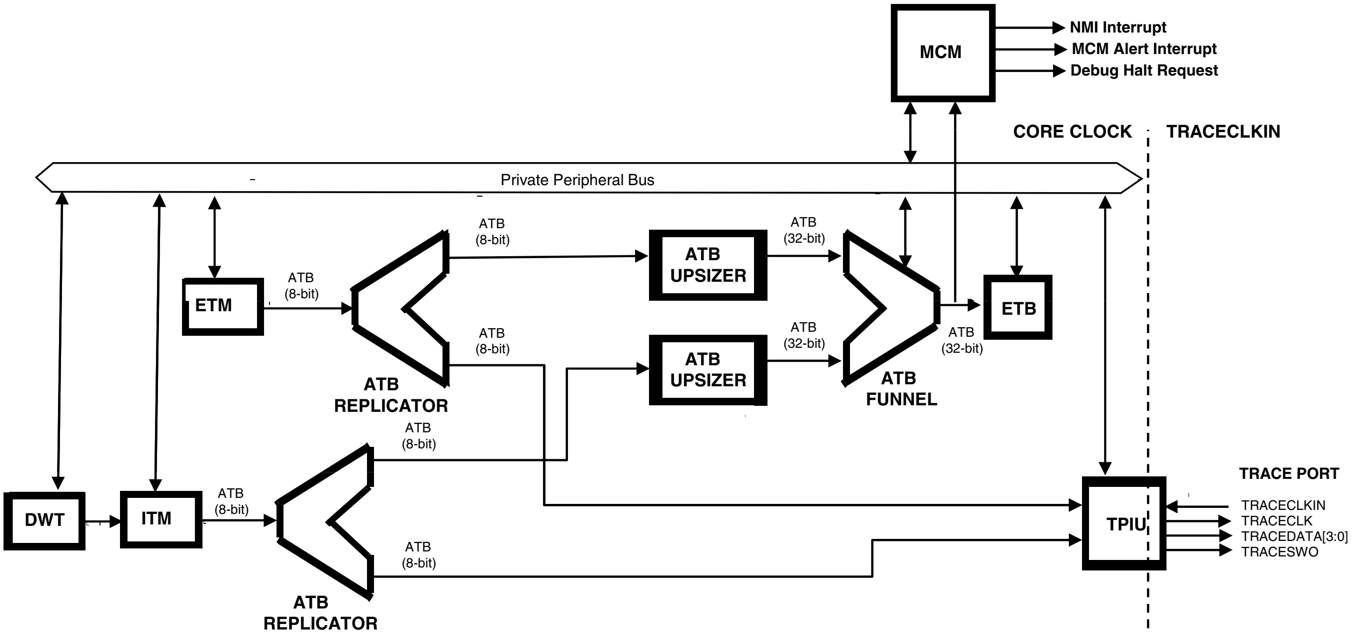


Figure 16-4. Core Trace Connectivity

The ETM and ITM can route its data to the ETB or the TPIU. (See the MCM (Miscellaneous Control Module) for controlling the routing to the TPIU.) This configuration enables the use of trace with low cost tools while maintaining the compatibility with trace probes. The arbitration between the ETM and ITM is performed inside the TPIU.

The ETB can not be configured with an interface smaller than 32 bits, making it necessary to add an ATB upsizer to make it compatible with the ETM operating with an 8-bit interface. The speed of the ETB 32 bit interface and its associated RAM is expected to be one quarter of the ETB clock.

The following combinations paths are supported:

1. ETM -> ETB

2. ETM -> TPIU(4 pin or 2 pin parallel)
3. ITM->ETB
4. ITM->TPIU(1 pin SWO, 2 pin or 4 pin parallel)
5. ETM & ITM -> ETB
6. ETM & ITM -> TPIU
7. ETM -> ETB & ITM -> TPIU

The following combination paths are NOT supported

16.10 Embedded Trace Macrocell v3.5 (ETM)

The Cortex-M4 Embedded Trace Macrocell (ETM-M4) is a debug component that enables a debugger to reconstruct program execution. The CoreSight ETM-M4 supports only instruction trace. You can use it either with the Cortex-M4 Trace Port Interface Unit (M4-TPIU).

The main features of an ETM are:

- tracing of 16-bit and 32-bit Thumb instructions
- four EmbeddedICE watchpoint inputs
- a Trace Start/Stop block with EmbeddedICE inputs
- one reduced function counter
- two external inputs
- a 24-byte FIFO queue
- global timestamping

16.11 TPIU

The TPIU acts as a bridge between the on-chip trace data from the Embedded Trace Macrocell (ETM) and the Instrumentation Trace Macrocell (ITM), with separate IDs, to a data stream, encapsulating IDs where required, that is then captured by a Trace Port Analyzer (TPA). The TPIU is specially designed for low-cost debug.

16.12 DWT

The DWT is a unit that performs the following debug functionality:

- It contains four comparators that you can configure as a hardware watchpoint, an ETM trigger, a PC sampler event trigger, or a data address sampler event trigger. The first comparator, DWT_COMP0, can also compare against the clock cycle counter, CYCCNT. The second comparator, DWT_COMP1, can also be used as a data comparator.
- The DWT contains counters for:
    - Clock cycles (CYCCNT)
    - Folded instructions
    - Load store unit (LSU) operations
    - Sleep cycles
    - CPI (all instruction cycles except for the first cycle)
    - Interrupt overhead

NOTE

An event is emitted each time a counter overflows.

- The DWT can be configured to emit PC samples at defined intervals, and to emit interrupt event information.

16.13 Debug in Low Power Modes

In low power modes in which the debug modules are kept static or powered off, the debugger cannot gather any debug data for the duration of the low power mode. In the case that the debugger is held static, the debug port returns to full functionality as soon as the low power mode exits and the system returns to a state with active debug. In the case that the debugger logic is powered off, the debugger is reset on recovery and must be reconfigured once the low power mode is exited.

Power mode entry logic monitors Debug Power Up and System Power Up signals from the debug port as indications that a debugger is active. These signals can be changed in RUN, VLPR, WAIT and VLPW. If the debug signal is active and the system attempts to enter stop or VLPS, FCLK continues to run to support core register access. In these modes in which FCLK is left active the debug modules have access to core registers but not to system memory resources accessed via the crossbar.

With debug enabled, transitions from Run directly to VLPS are not allowed and result in the system entering Stop mode instead. Status bits within the MDM-AP Status register can be evaluated to determine this pseudo-VLPS state. Note with the debug enabled, transitions from Run--> VLPR --> VLPS are still possible but also result in the system entering Stop mode instead.

In VLLS mode all debug modules are powered off and reset at wakeup. In LLS mode, the debug modules retain their state but no debug activity is possible.

NOTE

When using cJTAG and entering LLS mode, the cJTAG controller must be reset on exit from LLS mode.

Going into a VLLSx mode causes all the debug controls and settings to be reset. To give time to the debugger to sync up with the HW, the MDM-AP Control register can be configured hold the system in reset on recovery so that the debugger can regain control and reconfigure debug logic prior to the system exiting reset and resuming operation.

16.13.1 Debug Module State in Low Power Modes

The following table shows the state of the debug modules in low power modes. These terms are used:

- FF = Full functionality. In VLPR and VLPW the system frequency is limited, but if a module does not have a limitation in its functionality, it is still listed as FF.
- static = Module register states and associated memories are retained.
- OFF = Modules are powered off; module is in reset state upon wakeup.

Table 16-7. Debug Module State in Low Power Modes

Module	STOP	VLPR	VLPW	VLPS	LLS	VLLSx
Debug Port	FF	FF	FF	OFF	static	OFF
AHB-AP	FF	FF	FF	OFF	static	OFF
ITM	FF	FF	FF	OFF	static	OFF
TPIU	FF	FF	FF	OFF	static	OFF
DWT	FF	FF	FF	OFF	static	OFF

16.14 Debug & Security

When security is enabled (FSEC[SEC] != 10), the debug port capabilities are limited in order to prevent exploitation of secure data. In the secure state the debugger still has access to the MDM-AP Status Register and can determine the current security state of the device. In the case of a secure device, the debugger also has the capability of performing a mass erase operation via writes to the MDM-AP Control Register. In the case of a secure device that has mass erase disabled (FSEC[MEEN] = 10), attempts to mass erase via the debug interface are blocked.

Chapter 17
MCU: Signal Multiplexing and Signal Descriptions

17.1 Introduction

To optimize functionality in small packages, pins have several functions available via signal multiplexing. This chapter illustrates which of this device's signals are multiplexed on which external pin.

The Port Control block controls which signal is present on the external pin. Reference that chapter to find which register controls the operation of a specific pin.

17.2 Signal Multiplexing Integration

This section summarizes how the module is integrated into the device. For a comprehensive description of the module itself, see the module’s dedicated chapter.

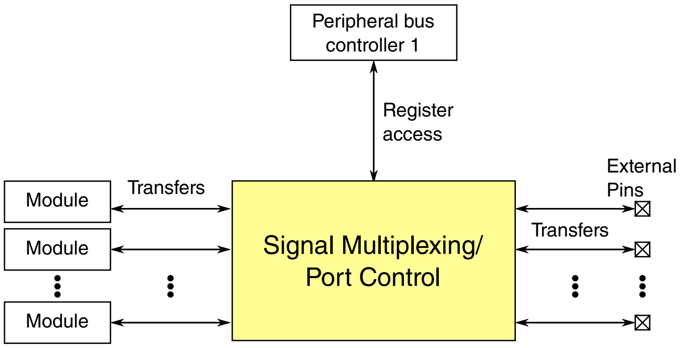


Figure 17-1. Signal multiplexing integration

Table 17-1. Reference links to related information

Topic	Related module	Reference
Full description	Port control	Port control
System memory map		System memory map

Table continues on the next page...

Table 17-1. Reference links to related information (continued)

Topic	Related module	Reference
Clocking		Clock Distribution
Register access	Peripheral bus controller	Peripheral bridge

17.2.1 Port control and interrupt module features

- Five 32-pin ports

NOTE

Not all pins are available on the device. See the following section for details.

- Each 32-pin port is assigned one interrupt.
- The digital filter option has two clock source options: bus clock and 1-kHz LPO. The 1-kHz LPO option gives users this feature in low power modes.
- The digital filter is configurable from 1 to 32 clock cycles when enabled.

17.2.2 PCRn reset values for port A

PCRn bit reset values for port A are 1 for the following bits:

- For PCR0: bits 1, 6, 8, 9, and 10.
- For PCR1 to PCR4: bits 0, 1, 6, 8, 9, and 10.
- For PCR5 : bits 0, 1, and 6.

All other PCRn bit reset values for port A are 0.

17.2.3 Clock gating

The clock to the port control module can be gated on and off using the SCGC5[PORTx] bits in the SIM module. These bits are cleared after any reset, which disables the clock to the corresponding module to conserve power. Prior to initializing the corresponding module, set SCGC5[PORTx] in the SIM module to enable the clock. Before turning off the clock, make sure to disable the module. For more details, refer to the clock distribution chapter.

17.2.4 Signal multiplexing constraints

1. A given peripheral function must be assigned to a maximum of one package pin. Do not program the same function to more than one pin.
2. To ensure the best signal timing for a given peripheral's interface, choose the pins in closest proximity to each other.

17.3 Module Signal Description Tables

The following sections correlate the chip-level signal name with the signal name used in the module's chapter. They also briefly describe the signal function and direction.

17.3.1 Core Modules

Table 17-2. JTAG Signal Descriptions

Chip signal name	Module signal name	Description	I/O
JTAG_TMS	JTAG_TMS/ SWD_DIO	JTAG Test Mode Selection	I/O
JTAG_TCLK	JTAG_TCLK/ SWD_CLK	JTAG Test Clock	I
JTAG_TDI	JTAG_TDI	JTAG Test Data Input	I
JTAG_TDO	JTAG_TDO/ TRACE_SWO	JTAG Test Data Output	O
JTAG_TRST	JTAG_TRST_b	JTAG Reset	I

Table 17-3. SWD Signal Descriptions

Chip signal name	Module signal name	Description	I/O
SWD_DIO	JTAG_TMS/ SWD_DIO	Serial Wire Data	I/O
SWD_CLK	JTAG_TCLK/ SWD_CLK	Serial Wire Clock	I

Table 17-4. TPIU Signal Descriptions

Chip signal name	Module signal name	Description	I/O
TRACE_CLKOUT	TRACECLK	Trace clock output from the ARM CoreSight debug block	O
TRACE_D[3:2]	TRACEDATA	Trace output data from the ARM CoreSight debug block used for 5-pin interface	O

Table continues on the next page...

Table 17-4. TPIU Signal Descriptions (continued)

Chip signal name	Module signal name	Description	I/O
TRACE_D[1:0]	TRACEDATA	Trace output data from the ARM CoreSight debug block used for both 5-pin and 3-pin interfaces	O
TRACE_SWO	JTAG_TDO/ TRACE_SWO	Trace output data from the ARM CoreSight debug block over a single pin	O

17.3.2 System Modules

Table 17-5. EWM Signal Descriptions

Chip signal name	Module signal name	Description	I/O
EWM_IN	EWM_in	EWM input for safety status of external safety circuits. The polarity of EWM_in is programmable using the EWM_CTRL[ASSIN] bit. The default polarity is active-low.	I
$\overline{\text{EWM\_OUT}}$	$\overline{\text{EWM\_out}}$	EWM reset out signal	O

17.3.3 Clock Modules

Table 17-6. OSC Signal Descriptions

Chip signal name	Module signal name	Description	I/O
EXTAL0	EXTAL	External clock/Oscillator input	I
XTAL0	XTAL	Oscillator output	O

Table 17-7. RTC OSC Signal Descriptions

Chip signal name	Module signal name	Description	I/O
EXTAL32	EXTAL32	32.768 kHz oscillator input	I
XTAL32	XTAL32	32.768 kHz oscillator output	O

17.3.4 Memories and Memory Interfaces

Table 17-8. EzPort Signal Descriptions

Chip signal name	Module signal name	Description	I/O
EZP_CLK	EZP_CK	EzPort Clock	Input
$\overline{\text{EZP\_CS}}$	$\overline{\text{EZP\_CS}}$	EzPort Chip Select	Input
EZP_DI	EZP_D	EzPort Serial Data In	Input
EZP_DO	EZP_Q	EzPort Serial Data Out	Output

17.3.5 Security Modules

Table 17-9. Tamper Detect Signal Descriptions

Chip signal name	Module signal name	Description	I/O
TAMPER[2:0]	TAMPER[7:0]	External tamper input or active tamper output	I/O

17.3.6 Analog

Table 17-10. ADC 0 Signal Descriptions

Chip signal name	Module signal name	Description	I/O
ADC0_DP1	DADP3–DADP0	Differential Analog Channel Inputs	I
ADC0_DM1	DADM3–DADM0	Differential Analog Channel Inputs	I
ADC0_SE[23:21,15:4]	AD*n*	Single-Ended Analog Channel Inputs	I
VREFH[1]	V_{REFSH}	Voltage Reference Select High	I
VREFL[2]	V_{REFSL}	Voltage Reference Select Low	I
VDDA	V_{DDA}	Analog Power Supply	I
VSSA	V_{SSA}	Analog Ground	I

1. VREFH is internally tied to VDDA.
2. VREFL is internally tied to VSSA.

Table 17-11. CMP 0 Signal Descriptions

Chip signal name	Module signal name	Description	I/O
CMP0_IN[5:0]	IN[5:0]	Analog voltage inputs	I
CMP0_OUT	CMPO	Comparator output	O

Table 17-12. CMP 1 Signal Descriptions

Chip signal name	Module signal name	Description	I/O
CMP1_IN[5:0]	IN[5:0]	Analog voltage inputs	I
CMP1_OUT	CMPO	Comparator output	O

17.3.7 Timer Modules

Table 17-13. FTM 0 Signal Descriptions

Chip signal name	Module signal name	Description	I/O
FTM_CLKIN[1:0]	EXTCLK	External clock. FTM external clock can be selected to drive the FTM counter.	I
FTM0_CH[7:0]	CHn	FTM channel (n), where n can be 7-0	I/O
FTM0_FLT[3:0]	FAULTj	Fault input (j), where j can be 3-0	I

Table 17-14. FTM 1 Signal Descriptions

Chip signal name	Module signal name	Description	I/O
FTM_CLKIN[1:0]	EXTCLK	External clock. FTM external clock can be selected to drive the FTM counter.	I
FTM1_CH[1:0]	CHn	FTM channel (n), where n can be 7-0	I/O
FTM1_FLT0	FAULTj	Fault input (j), where j can be 3-0	I
FTM1_QD_PHA	PHA	Quadrature decoder phase A input. Input pin associated with quadrature decoder phase A.	I
FTM1_QD_PHB	PHB	Quadrature decoder phase B input. Input pin associated with quadrature decoder phase B.	I

Table 17-15. FTM 2 Signal Descriptions

Chip signal name	Module signal name	Description	I/O
FTM_CLKIN[1:0]	EXTCLK	External clock. FTM external clock can be selected to drive the FTM counter.	I
FTM2_CH[1:0]	CHn	FTM channel (n), where n can be 7-0	I/O
FTM2_FLT0	FAULTj	Fault input (j), where j can be 3-0	I

Table 17-16. CMT Signal Descriptions

Chip signal name	Module signal name	Description	I/O
CMT_IRO	CMT_IRO	Infrared Output	O

Table 17-17. PDB 0 Signal Descriptions

Chip signal name	Module signal name	Description	I/O
PDB0_EXTRG	EXTRG	External Trigger Input Source If the PDB is enabled and external trigger input source is selected, a positive edge on the EXTRG signal resets and starts the counter.	I

Table 17-18. LPTMR 0 Signal Descriptions

Chip signal name	Module signal name	Description	I/O
LPTMR0_ALT[3:1]	LPTMR0_ALT*n*	Pulse Counter Input pin	I

Table 17-19. RTC Signal Descriptions

Chip signal name	Module signal name	Description	I/O
VBAT	—	Backup battery supply for RTC and VBAT register file	I
RTC_CLKOUT	RTC_CLKOUT	1 Hz square-wave output or OSCERCLK	O

17.3.8 Communication Interfaces

Table 17-20. USB FS OTG Signal Descriptions

Chip signal name	Module signal name	Description	I/O
USB0_DM	usb_dm	USB D- analog data signal on the USB bus.	I/O
USB0_DP	usb_dp	USB D+ analog data signal on the USB bus.	I/O
USB_SOF_OUT	—	USB start of frame signal. Can be used to make the USB start of frame available for external synchronization.	O

Table 17-21. USB VREG Signal Descriptions

Chip signal name	Module signal name	Description	I/O
VOUT33	reg33_out	Regulator output voltage	O
VREGIN	reg33_in	Unregulated power supply	I

Table 17-22. SPI 0 Signal Descriptions

Chip signal name	Module signal name	Description	I/O
SPI0_PCS0	PCS0/$\overline{\text{SS}}$	Peripheral Chip Select 0 (O)	I/O
SPI0_PCS[3:1]	PCS[1:3]	Peripheral Chip Selects 1–3	O
SPI0_PCS4	PCS4	Peripheral Chip Select 4	O
SPI0_SIN	SIN	Serial Data In	I
SPI0_SOUT	SOUT	Serial Data Out	O
SPI0_SCK	SCK	Serial Clock (O)	I/O

Table 17-23. SPI 1 Signal Descriptions

Chip signal name	Module signal name	Description	I/O
SPI1_PCS0	PCS0/$\overline{\text{SS}}$	Peripheral Chip Select 0 (O)	I/O
SPI1_PCS[2:1]	PCS[2:1]	Peripheral Chip Select 1 – 2	O
SPI1_SIN	SIN	Serial Data In	I
SPI1_SOUT	SOUT	Serial Data Out	O
SPI1_SCK	SCK	Serial Clock (O)	I/O

Table 17-24. I^{2}C 0 Signal Descriptions

Chip signal name	Module signal name	Description	I/O
I2C0_SCL	SCL	Bidirectional serial clock line of the I^{2}C system.	I/O
I2C0_SDA	SDA	Bidirectional serial data line of the I^{2}C system.	I/O

Table 17-25. I^{2}C 1 Signal Descriptions

Chip signal name	Module signal name	Description	I/O
I2C1_SCL	SCL	Bidirectional serial clock line of the I^{2}C system.	I/O
I2C1_SDA	SDA	Bidirectional serial data line of the I^{2}C system.	I/O

Table 17-26. UART 0 Signal Descriptions

Chip signal name	Module signal name	Description	I/O
$\overline{\text{UART0\_CTS}}$	$\overline{\text{CTS}}$	Clear to send	I
$\overline{\text{UART0\_RTS}}$	$\overline{\text{RTS}}$	Request to send	O
UART0_TX	TXD	Transmit data	O
UART0_RX	RXD	Receive data	I

Table 17-27. UART 1 Signal Descriptions

Chip signal name	Module signal name	Description	I/O
$\overline{\text{UART1\_CTS}}$	$\overline{\text{CTS}}$	Clear to send	I
$\overline{\text{UART1\_RTS}}$	$\overline{\text{RTS}}$	Request to send	O
UART1_TX	TXD	Transmit data	O
UART1_RX	RXD	Receive data	I

Table 17-28. UART 2 Signal Descriptions

Chip signal name	Module signal name	Description	I/O
$\overline{\text{UART2\_CTS}}$	$\overline{\text{CTS}}$	Clear to send	I
$\overline{\text{UART2\_RTS}}$	$\overline{\text{RTS}}$	Request to send	O
UART2_TX	TXD	Transmit data	O
UART2_RX	RXD	Receive data	I

Table 17-29. I^{2}S0 Signal Descriptions

Chip signal name	Module signal name	Description	I/O
I2S0_MCLK	SAI_MCLK	Audio Master Clock. The master clock is an input when externally generated and an output when internally generated.	I/O
I2S0_RX_BCLK	SAI_RX_BCLK	Receive Bit Clock. The bit clock is an input when externally generated and an output when internally generated.	I/O
I2S0_RX_FS	SAI_RX_SYNC	Receive Frame Sync. The frame sync is an input sampled synchronously by the bit clock when externally generated and an output generated synchronously by the bit clock when internally generated.	I/O
I2S0_RXD	SAI_RX_DATA	Receive Data. The receive data is sampled synchronously by the bit clock.	I
I2S0_TX_BCLK	SAI_TX_BCLK	Transmit Bit Clock. The bit clock is an input when externally generated and an output when internally generated.	I/O
I2S0_TX_FS	SAI_TX_SYNC	Transmit Frame Sync. The frame sync is an input sampled synchronously by the bit clock when externally generated and an output generated synchronously by the bit clock when internally generated.	I/O
I2S0_TXD	SAI_TX_DATA	Transmit Data. The transmit data is generated synchronously by the bit clock and is tristated whenever not transmitting a word.	O

17.3.9 Human-Machine Interfaces (HMI)

Table 17-30. GPIO Signal Descriptions

Chip signal name	Module signal name	Description	I/O
PTA[31:0]	PORTA31–PORTA0	General-purpose input/output	I/O
PTB[31:0][1]	PORTB31–PORTB0	General-purpose input/output	I/O
PTC[31:0][1]	PORTC31–PORTC0	General-purpose input/output	I/O
PTD[31:0][1]	PORTD31–PORTD0	General-purpose input/output	I/O
PTE[31:0][1]	PORTE31–PORTE0	General-purpose input/output	I/O

[1] The available GPIO pins depends on the specific package. See the signal multiplexing section for which exact GPIO signals are available.

Chapter 18
MCU: Port control and interrupts (PORT)

18.1 Introduction

NOTE

> For the chip-specific implementation details of this module's instances, see the chip configuration information.

18.2 Overview

The Port Control and Interrupt (PORT) module provides support for port control, digital filtering, and external interrupt functions.

Most functions can be configured independently for each pin in the 32-bit port and affect the pin regardless of its pin muxing state.

There is one instance of the PORT module for each port. Not all pins within each port are implemented on a specific device.

18.2.1 Features

The PORT module has the following features:

- Pin interrupt
  - Interrupt flag and enable registers for each pin
  - Support for edge sensitive (rising, falling, both) or level sensitive (low, high) configured per pin
  - Support for interrupt or DMA request configured per pin
  - Asynchronous wake-up in low-power modes
  - Pin interrupt is functional in all digital pin muxing modes
- Digital input filter on selected pins
  - Digital input filter for each pin

  - Individual enable or bypass control field per pin
  - Selectable clock source for digital input filter with a five bit resolution on filter size
- Port control
  - Individual pull control fields with pullup, pulldown, and pull-disable support
  - Individual drive strength field supporting high and low drive strength
  - Individual slew rate field supporting fast and slow slew rates
  - Individual input passive filter field supporting enable and disable of the individual input passive filter
  - Individual open drain field supporting enable and disable of the individual open drain output
  - Individual mux control field supporting analog or pin disabled, GPIO, and up to six chip-specific digital functions
  - Pad configuration fields are functional in all digital pin muxing modes.

18.2.2 Modes of operation

18.2.2.1 Run mode

In Run mode, the PORT operates normally.

18.2.2.2 Wait mode

In Wait mode, PORT continues to operate normally and may be configured to exit the Low-Power mode if an enabled interrupt is detected. DMA requests are still generated during the Wait mode, but do not cause an exit from the Low-Power mode.

18.2.2.3 Stop mode

In Stop mode, the PORT can be configured to exit the Low-Power mode via an asynchronous wake-up signal if an enabled interrupt is detected.

In Stop mode, the digital input filters are bypassed unless they are configured to run from the clock source.

18.2.2.4 Debug mode

In Debug mode, PORT operates normally.

18.3 External signal description

The table found here describes the PORT external signal.

Table 18-1. Signal properties

Name	Function	I/O	Reset	Pull
PORTx[31:0]	External interrupt	I/O	0	-

NOTE

Not all pins within each port are implemented on each device.

18.4 Detailed signal description

The table found here contains the detailed signal description for the PORT interface.

Table 18-2. PORT interface—detailed signal description

<table>
<tr><th>Signal</th><th>I/O</th><th colspan="2">Description</th></tr>
<tr><td rowspan="3">PORTx[31:0]</td><td rowspan="3">I/O</td><td colspan="2">External interrupt.</td></tr>
<tr><td>State meaning</td><td>Asserted—pin is logic 1.
Negated—pin is logic 0.</td></tr>
<tr><td>Timing</td><td>Assertion—may occur at any time and can assert asynchronously to the system clock.
Negation—may occur at any time and can assert asynchronously to the system clock.</td></tr>
</table>

18.5 Memory map and register definition

Any read or write access to the PORT memory space that is outside the valid memory map results in a bus error. All register accesses complete with zero wait states.

PORT memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4004_9000	Pin Control Register n (PORTA_PCR0)	32	R/W	See section	18.5.1/410
4004_9004	Pin Control Register n (PORTA_PCR1)	32	R/W	See section	18.5.1/410
4004_9008	Pin Control Register n (PORTA_PCR2)	32	R/W	See section	18.5.1/410
4004_900C	Pin Control Register n (PORTA_PCR3)	32	R/W	See section	18.5.1/410
4004_9010	Pin Control Register n (PORTA_PCR4)	32	R/W	See section	18.5.1/410
4004_9014	Pin Control Register n (PORTA_PCR5)	32	R/W	See section	18.5.1/410
4004_9018	Pin Control Register n (PORTA_PCR6)	32	R/W	See section	18.5.1/410
4004_901C	Pin Control Register n (PORTA_PCR7)	32	R/W	See section	18.5.1/410
4004_9020	Pin Control Register n (PORTA_PCR8)	32	R/W	See section	18.5.1/410
4004_9024	Pin Control Register n (PORTA_PCR9)	32	R/W	See section	18.5.1/410
4004_9028	Pin Control Register n (PORTA_PCR10)	32	R/W	See section	18.5.1/410
4004_902C	Pin Control Register n (PORTA_PCR11)	32	R/W	See section	18.5.1/410
4004_9030	Pin Control Register n (PORTA_PCR12)	32	R/W	See section	18.5.1/410
4004_9034	Pin Control Register n (PORTA_PCR13)	32	R/W	See section	18.5.1/410
4004_9038	Pin Control Register n (PORTA_PCR14)	32	R/W	See section	18.5.1/410
4004_903C	Pin Control Register n (PORTA_PCR15)	32	R/W	See section	18.5.1/410
4004_9040	Pin Control Register n (PORTA_PCR16)	32	R/W	See section	18.5.1/410
4004_9044	Pin Control Register n (PORTA_PCR17)	32	R/W	See section	18.5.1/410
4004_9048	Pin Control Register n (PORTA_PCR18)	32	R/W	See section	18.5.1/410
4004_904C	Pin Control Register n (PORTA_PCR19)	32	R/W	See section	18.5.1/410
4004_9050	Pin Control Register n (PORTA_PCR20)	32	R/W	See section	18.5.1/410
4004_9054	Pin Control Register n (PORTA_PCR21)	32	R/W	See section	18.5.1/410
4004_9058	Pin Control Register n (PORTA_PCR22)	32	R/W	See section	18.5.1/410
4004_905C	Pin Control Register n (PORTA_PCR23)	32	R/W	See section	18.5.1/410
4004_9060	Pin Control Register n (PORTA_PCR24)	32	R/W	See section	18.5.1/410
4004_9064	Pin Control Register n (PORTA_PCR25)	32	R/W	See section	18.5.1/410
4004_9068	Pin Control Register n (PORTA_PCR26)	32	R/W	See section	18.5.1/410
4004_906C	Pin Control Register n (PORTA_PCR27)	32	R/W	See section	18.5.1/410
4004_9070	Pin Control Register n (PORTA_PCR28)	32	R/W	See section	18.5.1/410
4004_9074	Pin Control Register n (PORTA_PCR29)	32	R/W	See section	18.5.1/410
4004_9078	Pin Control Register n (PORTA_PCR30)	32	R/W	See section	18.5.1/410
4004_907C	Pin Control Register n (PORTA_PCR31)	32	R/W	See section	18.5.1/410
4004_9080	Global Pin Control Low Register (PORTA_GPCLR)	32	W (always reads 0)	0000_0000h	18.5.2/413
4004_9084	Global Pin Control High Register (PORTA_GPCHR)	32	W (always reads 0)	0000_0000h	18.5.3/413
4004_90A0	Interrupt Status Flag Register (PORTA_ISFR)	32	w1c	0000_0000h	18.5.4/414

Table continues on the next page...

PORT memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4004_90C0	Digital Filter Enable Register (PORTA_DFER)	32	R/W	0000_0000h	18.5.5/414
4004_90C4	Digital Filter Clock Register (PORTA_DFCR)	32	R/W	0000_0000h	18.5.6/415
4004_90C8	Digital Filter Width Register (PORTA_DFWR)	32	R/W	0000_0000h	18.5.7/415
4004_A000	Pin Control Register n (PORTB_PCR0)	32	R/W	See section	18.5.1/410
4004_A004	Pin Control Register n (PORTB_PCR1)	32	R/W	See section	18.5.1/410
4004_A008	Pin Control Register n (PORTB_PCR2)	32	R/W	See section	18.5.1/410
4004_A00C	Pin Control Register n (PORTB_PCR3)	32	R/W	See section	18.5.1/410
4004_A010	Pin Control Register n (PORTB_PCR4)	32	R/W	See section	18.5.1/410
4004_A014	Pin Control Register n (PORTB_PCR5)	32	R/W	See section	18.5.1/410
4004_A018	Pin Control Register n (PORTB_PCR6)	32	R/W	See section	18.5.1/410
4004_A01C	Pin Control Register n (PORTB_PCR7)	32	R/W	See section	18.5.1/410
4004_A020	Pin Control Register n (PORTB_PCR8)	32	R/W	See section	18.5.1/410
4004_A024	Pin Control Register n (PORTB_PCR9)	32	R/W	See section	18.5.1/410
4004_A028	Pin Control Register n (PORTB_PCR10)	32	R/W	See section	18.5.1/410
4004_A02C	Pin Control Register n (PORTB_PCR11)	32	R/W	See section	18.5.1/410
4004_A030	Pin Control Register n (PORTB_PCR12)	32	R/W	See section	18.5.1/410
4004_A034	Pin Control Register n (PORTB_PCR13)	32	R/W	See section	18.5.1/410
4004_A038	Pin Control Register n (PORTB_PCR14)	32	R/W	See section	18.5.1/410
4004_A03C	Pin Control Register n (PORTB_PCR15)	32	R/W	See section	18.5.1/410
4004_A040	Pin Control Register n (PORTB_PCR16)	32	R/W	See section	18.5.1/410
4004_A044	Pin Control Register n (PORTB_PCR17)	32	R/W	See section	18.5.1/410
4004_A048	Pin Control Register n (PORTB_PCR18)	32	R/W	See section	18.5.1/410
4004_A04C	Pin Control Register n (PORTB_PCR19)	32	R/W	See section	18.5.1/410
4004_A050	Pin Control Register n (PORTB_PCR20)	32	R/W	See section	18.5.1/410
4004_A054	Pin Control Register n (PORTB_PCR21)	32	R/W	See section	18.5.1/410
4004_A058	Pin Control Register n (PORTB_PCR22)	32	R/W	See section	18.5.1/410
4004_A05C	Pin Control Register n (PORTB_PCR23)	32	R/W	See section	18.5.1/410
4004_A060	Pin Control Register n (PORTB_PCR24)	32	R/W	See section	18.5.1/410
4004_A064	Pin Control Register n (PORTB_PCR25)	32	R/W	See section	18.5.1/410
4004_A068	Pin Control Register n (PORTB_PCR26)	32	R/W	See section	18.5.1/410
4004_A06C	Pin Control Register n (PORTB_PCR27)	32	R/W	See section	18.5.1/410
4004_A070	Pin Control Register n (PORTB_PCR28)	32	R/W	See section	18.5.1/410
4004_A074	Pin Control Register n (PORTB_PCR29)	32	R/W	See section	18.5.1/410
4004_A078	Pin Control Register n (PORTB_PCR30)	32	R/W	See section	18.5.1/410
4004_A07C	Pin Control Register n (PORTB_PCR31)	32	R/W	See section	18.5.1/410
4004_A080	Global Pin Control Low Register (PORTB_GPCLR)	32	W (always reads 0)	0000_0000h	18.5.2/413

Table continues on the next page...

PORT memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4004_A084	Global Pin Control High Register (PORTB_GPCHR)	32	W (always reads 0)	0000_0000h	18.5.3/413
4004_A0A0	Interrupt Status Flag Register (PORTB_ISFR)	32	w1c	0000_0000h	18.5.4/414
4004_A0C0	Digital Filter Enable Register (PORTB_DFER)	32	R/W	0000_0000h	18.5.5/414
4004_A0C4	Digital Filter Clock Register (PORTB_DFCR)	32	R/W	0000_0000h	18.5.6/415
4004_A0C8	Digital Filter Width Register (PORTB_DFWR)	32	R/W	0000_0000h	18.5.7/415
4004_B000	Pin Control Register n (PORTC_PCR0)	32	R/W	See section	18.5.1/410
4004_B004	Pin Control Register n (PORTC_PCR1)	32	R/W	See section	18.5.1/410
4004_B008	Pin Control Register n (PORTC_PCR2)	32	R/W	See section	18.5.1/410
4004_B00C	Pin Control Register n (PORTC_PCR3)	32	R/W	See section	18.5.1/410
4004_B010	Pin Control Register n (PORTC_PCR4)	32	R/W	See section	18.5.1/410
4004_B014	Pin Control Register n (PORTC_PCR5)	32	R/W	See section	18.5.1/410
4004_B018	Pin Control Register n (PORTC_PCR6)	32	R/W	See section	18.5.1/410
4004_B01C	Pin Control Register n (PORTC_PCR7)	32	R/W	See section	18.5.1/410
4004_B020	Pin Control Register n (PORTC_PCR8)	32	R/W	See section	18.5.1/410
4004_B024	Pin Control Register n (PORTC_PCR9)	32	R/W	See section	18.5.1/410
4004_B028	Pin Control Register n (PORTC_PCR10)	32	R/W	See section	18.5.1/410
4004_B02C	Pin Control Register n (PORTC_PCR11)	32	R/W	See section	18.5.1/410
4004_B030	Pin Control Register n (PORTC_PCR12)	32	R/W	See section	18.5.1/410
4004_B034	Pin Control Register n (PORTC_PCR13)	32	R/W	See section	18.5.1/410
4004_B038	Pin Control Register n (PORTC_PCR14)	32	R/W	See section	18.5.1/410
4004_B03C	Pin Control Register n (PORTC_PCR15)	32	R/W	See section	18.5.1/410
4004_B040	Pin Control Register n (PORTC_PCR16)	32	R/W	See section	18.5.1/410
4004_B044	Pin Control Register n (PORTC_PCR17)	32	R/W	See section	18.5.1/410
4004_B048	Pin Control Register n (PORTC_PCR18)	32	R/W	See section	18.5.1/410
4004_B04C	Pin Control Register n (PORTC_PCR19)	32	R/W	See section	18.5.1/410
4004_B050	Pin Control Register n (PORTC_PCR20)	32	R/W	See section	18.5.1/410
4004_B054	Pin Control Register n (PORTC_PCR21)	32	R/W	See section	18.5.1/410
4004_B058	Pin Control Register n (PORTC_PCR22)	32	R/W	See section	18.5.1/410
4004_B05C	Pin Control Register n (PORTC_PCR23)	32	R/W	See section	18.5.1/410
4004_B060	Pin Control Register n (PORTC_PCR24)	32	R/W	See section	18.5.1/410
4004_B064	Pin Control Register n (PORTC_PCR25)	32	R/W	See section	18.5.1/410
4004_B068	Pin Control Register n (PORTC_PCR26)	32	R/W	See section	18.5.1/410
4004_B06C	Pin Control Register n (PORTC_PCR27)	32	R/W	See section	18.5.1/410
4004_B070	Pin Control Register n (PORTC_PCR28)	32	R/W	See section	18.5.1/410
4004_B074	Pin Control Register n (PORTC_PCR29)	32	R/W	See section	18.5.1/410
4004_B078	Pin Control Register n (PORTC_PCR30)	32	R/W	See section	18.5.1/410

Table continues on the next page...

PORT memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4004_B07C	Pin Control Register n (PORTC_PCR31)	32	R/W	See section	18.5.1/410
4004_B080	Global Pin Control Low Register (PORTC_GPCLR)	32	W (always reads 0)	0000_0000h	18.5.2/413
4004_B084	Global Pin Control High Register (PORTC_GPCHR)	32	W (always reads 0)	0000_0000h	18.5.3/413
4004_B0A0	Interrupt Status Flag Register (PORTC_ISFR)	32	w1c	0000_0000h	18.5.4/414
4004_B0C0	Digital Filter Enable Register (PORTC_DFER)	32	R/W	0000_0000h	18.5.5/414
4004_B0C4	Digital Filter Clock Register (PORTC_DFCR)	32	R/W	0000_0000h	18.5.6/415
4004_B0C8	Digital Filter Width Register (PORTC_DFWR)	32	R/W	0000_0000h	18.5.7/415
4004_C000	Pin Control Register n (PORTD_PCR0)	32	R/W	See section	18.5.1/410
4004_C004	Pin Control Register n (PORTD_PCR1)	32	R/W	See section	18.5.1/410
4004_C008	Pin Control Register n (PORTD_PCR2)	32	R/W	See section	18.5.1/410
4004_C00C	Pin Control Register n (PORTD_PCR3)	32	R/W	See section	18.5.1/410
4004_C010	Pin Control Register n (PORTD_PCR4)	32	R/W	See section	18.5.1/410
4004_C014	Pin Control Register n (PORTD_PCR5)	32	R/W	See section	18.5.1/410
4004_C018	Pin Control Register n (PORTD_PCR6)	32	R/W	See section	18.5.1/410
4004_C01C	Pin Control Register n (PORTD_PCR7)	32	R/W	See section	18.5.1/410
4004_C020	Pin Control Register n (PORTD_PCR8)	32	R/W	See section	18.5.1/410
4004_C024	Pin Control Register n (PORTD_PCR9)	32	R/W	See section	18.5.1/410
4004_C028	Pin Control Register n (PORTD_PCR10)	32	R/W	See section	18.5.1/410
4004_C02C	Pin Control Register n (PORTD_PCR11)	32	R/W	See section	18.5.1/410
4004_C030	Pin Control Register n (PORTD_PCR12)	32	R/W	See section	18.5.1/410
4004_C034	Pin Control Register n (PORTD_PCR13)	32	R/W	See section	18.5.1/410
4004_C038	Pin Control Register n (PORTD_PCR14)	32	R/W	See section	18.5.1/410
4004_C03C	Pin Control Register n (PORTD_PCR15)	32	R/W	See section	18.5.1/410
4004_C040	Pin Control Register n (PORTD_PCR16)	32	R/W	See section	18.5.1/410
4004_C044	Pin Control Register n (PORTD_PCR17)	32	R/W	See section	18.5.1/410
4004_C048	Pin Control Register n (PORTD_PCR18)	32	R/W	See section	18.5.1/410
4004_C04C	Pin Control Register n (PORTD_PCR19)	32	R/W	See section	18.5.1/410
4004_C050	Pin Control Register n (PORTD_PCR20)	32	R/W	See section	18.5.1/410
4004_C054	Pin Control Register n (PORTD_PCR21)	32	R/W	See section	18.5.1/410
4004_C058	Pin Control Register n (PORTD_PCR22)	32	R/W	See section	18.5.1/410
4004_C05C	Pin Control Register n (PORTD_PCR23)	32	R/W	See section	18.5.1/410
4004_C060	Pin Control Register n (PORTD_PCR24)	32	R/W	See section	18.5.1/410
4004_C064	Pin Control Register n (PORTD_PCR25)	32	R/W	See section	18.5.1/410
4004_C068	Pin Control Register n (PORTD_PCR26)	32	R/W	See section	18.5.1/410
4004_C06C	Pin Control Register n (PORTD_PCR27)	32	R/W	See section	18.5.1/410

Table continues on the next page...

PORT memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4004_C070	Pin Control Register n (PORTD_PCR28)	32	R/W	See section	18.5.1/410
4004_C074	Pin Control Register n (PORTD_PCR29)	32	R/W	See section	18.5.1/410
4004_C078	Pin Control Register n (PORTD_PCR30)	32	R/W	See section	18.5.1/410
4004_C07C	Pin Control Register n (PORTD_PCR31)	32	R/W	See section	18.5.1/410
4004_C080	Global Pin Control Low Register (PORTD_GPCLR)	32	W (always reads 0)	0000_0000h	18.5.2/413
4004_C084	Global Pin Control High Register (PORTD_GPCHR)	32	W (always reads 0)	0000_0000h	18.5.3/413
4004_C0A0	Interrupt Status Flag Register (PORTD_ISFR)	32	w1c	0000_0000h	18.5.4/414
4004_C0C0	Digital Filter Enable Register (PORTD_DFER)	32	R/W	0000_0000h	18.5.5/414
4004_C0C4	Digital Filter Clock Register (PORTD_DFCR)	32	R/W	0000_0000h	18.5.6/415
4004_C0C8	Digital Filter Width Register (PORTD_DFWR)	32	R/W	0000_0000h	18.5.7/415
4004_D000	Pin Control Register n (PORTE_PCR0)	32	R/W	See section	18.5.1/410
4004_D004	Pin Control Register n (PORTE_PCR1)	32	R/W	See section	18.5.1/410
4004_D008	Pin Control Register n (PORTE_PCR2)	32	R/W	See section	18.5.1/410
4004_D00C	Pin Control Register n (PORTE_PCR3)	32	R/W	See section	18.5.1/410
4004_D010	Pin Control Register n (PORTE_PCR4)	32	R/W	See section	18.5.1/410
4004_D014	Pin Control Register n (PORTE_PCR5)	32	R/W	See section	18.5.1/410
4004_D018	Pin Control Register n (PORTE_PCR6)	32	R/W	See section	18.5.1/410
4004_D01C	Pin Control Register n (PORTE_PCR7)	32	R/W	See section	18.5.1/410
4004_D020	Pin Control Register n (PORTE_PCR8)	32	R/W	See section	18.5.1/410
4004_D024	Pin Control Register n (PORTE_PCR9)	32	R/W	See section	18.5.1/410
4004_D028	Pin Control Register n (PORTE_PCR10)	32	R/W	See section	18.5.1/410
4004_D02C	Pin Control Register n (PORTE_PCR11)	32	R/W	See section	18.5.1/410
4004_D030	Pin Control Register n (PORTE_PCR12)	32	R/W	See section	18.5.1/410
4004_D034	Pin Control Register n (PORTE_PCR13)	32	R/W	See section	18.5.1/410
4004_D038	Pin Control Register n (PORTE_PCR14)	32	R/W	See section	18.5.1/410
4004_D03C	Pin Control Register n (PORTE_PCR15)	32	R/W	See section	18.5.1/410
4004_D040	Pin Control Register n (PORTE_PCR16)	32	R/W	See section	18.5.1/410
4004_D044	Pin Control Register n (PORTE_PCR17)	32	R/W	See section	18.5.1/410
4004_D048	Pin Control Register n (PORTE_PCR18)	32	R/W	See section	18.5.1/410
4004_D04C	Pin Control Register n (PORTE_PCR19)	32	R/W	See section	18.5.1/410
4004_D050	Pin Control Register n (PORTE_PCR20)	32	R/W	See section	18.5.1/410
4004_D054	Pin Control Register n (PORTE_PCR21)	32	R/W	See section	18.5.1/410
4004_D058	Pin Control Register n (PORTE_PCR22)	32	R/W	See section	18.5.1/410
4004_D05C	Pin Control Register n (PORTE_PCR23)	32	R/W	See section	18.5.1/410
4004_D060	Pin Control Register n (PORTE_PCR24)	32	R/W	See section	18.5.1/410

Table continues on the next page...

PORT memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4004_D064	Pin Control Register n (PORTE_PCR25)	32	R/W	See section	18.5.1/410
4004_D068	Pin Control Register n (PORTE_PCR26)	32	R/W	See section	18.5.1/410
4004_D06C	Pin Control Register n (PORTE_PCR27)	32	R/W	See section	18.5.1/410
4004_D070	Pin Control Register n (PORTE_PCR28)	32	R/W	See section	18.5.1/410
4004_D074	Pin Control Register n (PORTE_PCR29)	32	R/W	See section	18.5.1/410
4004_D078	Pin Control Register n (PORTE_PCR30)	32	R/W	See section	18.5.1/410
4004_D07C	Pin Control Register n (PORTE_PCR31)	32	R/W	See section	18.5.1/410
4004_D080	Global Pin Control Low Register (PORTE_GPCLR)	32	W (always reads 0)	0000_0000h	18.5.2/413
4004_D084	Global Pin Control High Register (PORTE_GPCHR)	32	W (always reads 0)	0000_0000h	18.5.3/413
4004_D0A0	Interrupt Status Flag Register (PORTE_ISFR)	32	w1c	0000_0000h	18.5.4/414
4004_D0C0	Digital Filter Enable Register (PORTE_DFER)	32	R/W	0000_0000h	18.5.5/414
4004_D0C4	Digital Filter Clock Register (PORTE_DFCR)	32	R/W	0000_0000h	18.5.6/415
4004_D0C8	Digital Filter Width Register (PORTE_DFWR)	32	R/W	0000_0000h	18.5.7/415

18.5.1 Pin Control Register n (PORTx_PCR*n*)

NOTE

See the Signal Multiplexing and Pin Assignment chapter for the reset value of this device.

See the GPIO Configuration section for details on the available functions for each pin.

Do not modify pin configuration registers associated with pins not available in your selected package. All unbonded pins not available in your package will default to DISABLE state for lowest power consumption.

Address: Base address + 0h offset + (4d × i), where i=0d to 31d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0							ISF	0				IRQC			
W								w1c								
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	LK	0				MUX			0	DSE	ODE	PFE	0	SRE	PE	PS
W																
Reset	0	0	0	0	0	*	*	*	0	*	0	*	0	*	*	*

* Notes:

- MUX field: Varies by port. See Signal Multiplexing and Signal Descriptions chapter for reset values per port.
- DSE field: Varies by port. See the Signal Multiplexing and Signal Descriptions chapter for reset values per port.
- PFE field: Varies by port. See Signal Multiplexing and Signal Descriptions chapter for reset values per port.
- SRE field: Varies by port. See Signal Multiplexing and Signal Descriptions chapter for reset values per port.
- PE field: Varies by port. See Signal Multiplexing and Signal Descriptions chapter for reset values per port.
- PS field: Varies by port. See Signal Multiplexing and Signal Descriptions chapter for reset values per port.

PORTx_PCR*n* field descriptions

Field	Description
31–25 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
24 ISF	Interrupt Status Flag The pin interrupt configuration is valid in all digital pin muxing modes.

Table continues on the next page...

PORTx_PCR*n* field descriptions (continued)

Field	Description
	0 Configured interrupt is not detected. 1 Configured interrupt is detected. If the pin is configured to generate a DMA request, then the corresponding flag will be cleared automatically at the completion of the requested DMA transfer. Otherwise, the flag remains set until a logic 1 is written to the flag. If the pin is configured for a level sensitive interrupt and the pin remains asserted, then the flag is set again immediately after it is cleared.
23–20 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
19–16 IRQC	Interrupt Configuration The pin interrupt configuration is valid in all digital pin muxing modes. The corresponding pin is configured to generate interrupt/DMA request as follows: 0000 Interrupt Status Flag (ISF) is disabled. 0001 ISF flag and DMA request on rising edge. 0010 ISF flag and DMA request on falling edge. 0011 ISF flag and DMA request on either edge. 0100 Reserved. 0101 Reserved. 0110 Reserved. 0111 Reserved. 1000 ISF flag and Interrupt when logic 0. 1001 ISF flag and Interrupt on rising-edge. 1010 ISF flag and Interrupt on falling-edge. 1011 ISF flag and Interrupt on either edge. 1100 ISF flag and Interrupt when logic 1. 1101 Reserved. 1110 Reserved. 1111 Reserved.
15 LK	Lock Register 0 Pin Control Register fields [15:0] are not locked. 1 Pin Control Register fields [15:0] are locked and cannot be updated until the next system reset.
14–11 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
10–8 MUX	Pin Mux Control Not all pins support all pin muxing slots. Unimplemented pin muxing slots are reserved and may result in configuring the pin for a different pin muxing slot. The corresponding pin is configured in the following pin muxing slot as follows: 000 Pin disabled (Alternative 0) (analog). 001 Alternative 1 (GPIO). 010 Alternative 2 (chip-specific). 011 Alternative 3 (chip-specific). 100 Alternative 4 (chip-specific). 101 Alternative 5 (chip-specific). 110 Alternative 6 (chip-specific). 111 Alternative 7 (chip-specific).

Table continues on the next page...

PORTx_PCR*n* field descriptions (continued)

Field	Description
7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
6 DSE	Drive Strength Enable Drive strength configuration is valid in all digital pin muxing modes. 0 Low drive strength is configured on the corresponding pin, if pin is configured as a digital output. 1 High drive strength is configured on the corresponding pin, if pin is configured as a digital output.
5 ODE	Open Drain Enable Open drain configuration is valid in all digital pin muxing modes. 0 Open drain output is disabled on the corresponding pin. 1 Open drain output is enabled on the corresponding pin, if the pin is configured as a digital output.
4 PFE	Passive Filter Enable Passive filter configuration is valid in all digital pin muxing modes. 0 Passive input filter is disabled on the corresponding pin. 1 Passive input filter is enabled on the corresponding pin, if the pin is configured as a digital input. Refer to the device data sheet for filter characteristics.
3 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
2 SRE	Slew Rate Enable Slew rate configuration is valid in all digital pin muxing modes. 0 Fast slew rate is configured on the corresponding pin, if the pin is configured as a digital output. 1 Slow slew rate is configured on the corresponding pin, if the pin is configured as a digital output.
1 PE	Pull Enable Pull configuration is valid in all digital pin muxing modes. 0 Internal pullup or pulldown resistor is not enabled on the corresponding pin. 1 Internal pullup or pulldown resistor is enabled on the corresponding pin, if the pin is configured as a digital input.
0 PS	Pull Select Pull configuration is valid in all digital pin muxing modes. 0 Internal pulldown resistor is enabled on the corresponding pin, if the corresponding PE field is set. 1 Internal pullup resistor is enabled on the corresponding pin, if the corresponding PE field is set.

18.5.2 Global Pin Control Low Register (PORTx_GPCLR)

Only 32-bit writes are supported to this register.

Address: Base address + 80h offset

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

PORTx_GPCLR field descriptions

Field	Description
31–16 GPWE	Global Pin Write Enable Selects which Pin Control Registers (15 through 0) bits [15:0] update with the value in GPWD. If a selected Pin Control Register is locked then the write to that register is ignored. 0 Corresponding Pin Control Register is not updated with the value in GPWD. 1 Corresponding Pin Control Register is updated with the value in GPWD.
GPWD	Global Pin Write Data Write value that is written to all Pin Control Registers bits [15:0] that are selected by GPWE.

18.5.3 Global Pin Control High Register (PORTx_GPCHR)

Only 32-bit writes are supported to this register.

Address: Base address + 84h offset

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

PORTx_GPCHR field descriptions

Field	Description
31–16 GPWE	Global Pin Write Enable Selects which Pin Control Registers (31 through 16) bits [15:0] update with the value in GPWD. If a selected Pin Control Register is locked then the write to that register is ignored. 0 Corresponding Pin Control Register is not updated with the value in GPWD. 1 Corresponding Pin Control Register is updated with the value in GPWD.
GPWD	Global Pin Write Data Write value that is written to all Pin Control Registers bits [15:0] that are selected by GPWE.

18.5.4 Interrupt Status Flag Register (PORTx_ISFR)

The pin interrupt configuration is valid in all digital pin muxing modes. The Interrupt Status Flag for each pin is also visible in the corresponding Pin Control Register, and each flag can be cleared in either location.

Address: Base address + A0h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	ISF																															
W	w1c																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PORTx_ISFR field descriptions

Field	Description
ISF	Interrupt Status Flag Each bit in the field indicates the detection of the configured interrupt of the same number as the field. 0 Configured interrupt is not detected. 1 Configured interrupt is detected. If the pin is configured to generate a DMA request, then the corresponding flag will be cleared automatically at the completion of the requested DMA transfer. Otherwise, the flag remains set until a logic 1 is written to the flag. If the pin is configured for a level sensitive interrupt and the pin remains asserted, then the flag is set again immediately after it is cleared.

18.5.5 Digital Filter Enable Register (PORTx_DFER)

The corresponding bit is read only for pins that do not support a digital filter. Refer to the Chapter of Signal Multiplexing and Signal Descriptions for the pins that support digital filter.

The digital filter configuration is valid in all digital pin muxing modes.

Address: Base address + C0h offset

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

PORTx_DFER field descriptions

Field	Description
DFE	Digital Filter Enable

PORTx_DFER field descriptions (continued)

Field	Description
	The digital filter configuration is valid in all digital pin muxing modes. The output of each digital filter is reset to zero at system reset and whenever the digital filter is disabled. Each bit in the field enables the digital filter of the same number as the field. 0 Digital filter is disabled on the corresponding pin and output of the digital filter is reset to zero. 1 Digital filter is enabled on the corresponding pin, if the pin is configured as a digital input.

18.5.6 Digital Filter Clock Register (PORTx_DFCR)

This register is read only for ports that do not support a digital filter.

The digital filter configuration is valid in all digital pin muxing modes.

Address: Base address + C4h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0															CS
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PORTx_DFCR field descriptions

Field	Description
31–1 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
0 CS	Clock Source The digital filter configuration is valid in all digital pin muxing modes. Configures the clock source for the digital input filters. Changing the filter clock source must be done only when all digital filters are disabled. 0 Digital filters are clocked by the bus clock. 1 Digital filters are clocked by the clock.

18.5.7 Digital Filter Width Register (PORTx_DFWR)

This register is read only for ports that do not support a digital filter.

The digital filter configuration is valid in all digital pin muxing modes.

Address: Base address + C8h offset

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

PORTx_DFWR field descriptions

Field	Description
31–5 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
FILT	Filter Length The digital filter configuration is valid in all digital pin muxing modes. Configures the maximum size of the glitches, in clock cycles, that the digital filter absorbs for the enabled digital filters. Glitches that are longer than this register setting will pass through the digital filter, and glitches that are equal to or less than this register setting are filtered. Changing the filter length must be done only after all filters are disabled.

18.6 Functional description

18.6.1 Pin control

Each port pin has a corresponding Pin Control register, PORT_PCRn, associated with it.

The upper half of the Pin Control register configures the pin's capability to either interrupt the CPU or request a DMA transfer, on a rising/falling edge or both edges as well as a logic level occurring on the port pin. It also includes a flag to indicate that an interrupt has occurred.

The lower half of the Pin Control register configures the following functions for each pin within the 32-bit port.

- Pullup or pulldown enable
- Drive strength and slew rate configuration
- Open drain enable
- Passive input filter enable
- Pin Muxing mode

The functions apply across all digital pin muxing modes and individual peripherals do not override the configuration in the Pin Control register. For example, if an I^2C function is enabled on a pin, that does not override the pullup or open drain configuration for that pin.

When the Pin Muxing mode is configured for analog or is disabled, all the digital functions on that pin are disabled. This includes the pullup and pulldown enables, output buffer enable, input buffer enable, and passive filter enable.

The LK bit (bit 15 of Pin Control Register PCR*n*) allows the configuration for each pin to be locked until the next system reset. When locked, writes to the lower half of that pin control register are ignored, although a bus error is not generated on an attempted write to a locked register.

The configuration of each Pin Control register is retained when the PORT module is disabled.

Whenever a pin is configured in any digital pin muxing mode, the input buffer for that pin is enabled allowing the pin state to be read via the corresponding GPIO Port Data Input Register (GPIO_PDIR) or allowing a pin interrupt or DMA request to be generated. If a pin is ever floating when its input buffer is enabled, then this can cause an increase in power consumption and must be avoided. A pin can be floating due to an input pin that is not connected or an output pin that has tri-stated (output buffer is disabled).

Enabling the internal pull resistor (or implementing an external pull resistor) will ensure a pin does not float when its input buffer is enabled; note that the internal pull resistor is automatically disabled whenever the output buffer is enabled allowing the Pull Enable bit to remain set. Configuring the Pin Muxing mode to disabled or analog will disable the pin’s input buffer and results in the lowest power consumption.

18.6.2 Global pin control

The two global pin control registers allow a single register write to update the lower half of the pin control register on up to 16 pins, all with the same value. Registers that are locked cannot be written using the global pin control registers.

The global pin control registers are designed to enable software to quickly configure multiple pins within the one port for the same peripheral function. However, the interrupt functions cannot be configured using the global pin control registers.

The global pin control registers are write-only registers, that always read as 0.

18.6.3 External interrupts

The external interrupt capability of the PORT module is available in all digital pin muxing modes provided the PORT module is enabled.

Each pin can be individually configured for any of the following external interrupt modes:

- Interrupt disabled, default out of reset
- Active high level sensitive interrupt
- Active low level sensitive interrupt
- Rising edge sensitive interrupt
- Falling edge sensitive interrupt
- Rising and falling edge sensitive interrupt
- Rising edge sensitive DMA request
- Falling edge sensitive DMA request
- Rising and falling edge sensitive DMA request

The interrupt status flag is set when the configured edge or level is detected on the pin or at the output of the digital input filter, if the digital input digital filter is enabled. When not in Stop mode, the input is first synchronized to the bus clock to detect the configured level or edge transition.

The PORT module generates a single interrupt that asserts when the interrupt status flag is set for any enabled interrupt for that port. The interrupt negates after the interrupt status flags for all enabled interrupts have been cleared by writing a logic 1 to the ISF flag in either the PORT_ISFR or PORT_PCRn registers.

The PORT module generates a single DMA request that asserts when the interrupt status flag is set for any enabled DMA request in that port. The DMA request negates after the DMA transfer is completed, because that clears the interrupt status flags for all enabled DMA requests.

During Stop mode, the interrupt status flag for any enabled interrupt is asynchronously set if the required level or edge is detected. This also generates an asynchronous wake-up signal to exit the Low-Power mode.

18.6.4 Digital filter

The clock used for all digital filters within one port can be configured between the bus clock or the clock. This selection must be changed only when all digital filters for that port are disabled. If the digital filters for a port are configured to use the bus clock, then the digital filters are bypassed for the duration of Stop mode. While the digital filters are bypassed, the output of each digital filter always equals the input pin, but the internal state of the digital filters remains static and does not update due to any change on the input pin.

The filter width in clock size is the same for all enabled digital filters within one port and must be changed only when all digital filters for that port are disabled.

The output of each digital filter is logic zero after system reset and whenever a digital filter is disabled. After a digital filter is enabled, the input is synchronized to the filter clock, either the bus clock or the clock. If the synchronized input and the output of the digital filter remain different for a number of filter clock cycles equal to the filter width register configuration, then the output of the digital filter updates to equal the synchronized filter input.

The maximum latency through a digital filter equals three filter clock cycles plus the filter width configuration register.

Chapter 19
MCU: System Integration Module (SIM)

19.1 Introduction

NOTE

> For the chip-specific implementation details of this module's instances, see the chip configuration information.

The System Integration Module (SIM) provides system control and chip configuration registers.

19.1.1 Features

Features of the SIM include:

- System clocking configuration
  - System clock divide values
  - Architectural clock gating control
  - USB clock selection and divide values
- Flash and system RAM size configuration
- USB regulator configuration
- FlexTimer external clock, hardware trigger, and fault source selection
- UART0 and UART1 receive/transmit source selection/configuration

19.2 Memory map and register definition

The SIM module contains many fields for selecting the clock source and dividers for various module clocks. See the Clock Distribution chapter for more information, including block diagrams and clock definitions.

NOTE

The SIM_SOPT1 and SIM_SOPT1CFG registers are located at a different base address than the other SIM registers.

SIM memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4004_7000	System Options Register 1 (SIM_SOPT1)	32	R/W	See section	19.2.1/423
4004_7004	SOPT1 Configuration Register (SIM_SOPT1CFG)	32	R/W	0000_0000h	19.2.2/425
4004_8004	System Options Register 2 (SIM_SOPT2)	32	R/W	0000_1000h	19.2.3/426
4004_800C	System Options Register 4 (SIM_SOPT4)	32	R/W	0000_0000h	19.2.4/428
4004_8010	System Options Register 5 (SIM_SOPT5)	32	R/W	0000_0000h	19.2.5/431
4004_8018	System Options Register 7 (SIM_SOPT7)	32	R/W	0000_0000h	19.2.6/432
4004_8024	System Device Identification Register (SIM_SDID)	32	R	See section	19.2.7/433
4004_8034	System Clock Gating Control Register 4 (SIM_SCGC4)	32	R/W	F010_0030h	19.2.8/435
4004_8038	System Clock Gating Control Register 5 (SIM_SCGC5)	32	R/W	0004_0182h	19.2.9/437
4004_803C	System Clock Gating Control Register 6 (SIM_SCGC6)	32	R/W	4000_0001h	19.2.10/438
4004_8040	System Clock Gating Control Register 7 (SIM_SCGC7)	32	R/W	0000_0002h	19.2.11/441
4004_8044	System Clock Divider Register 1 (SIM_CLKDIV1)	32	R/W	See section	19.2.12/442
4004_8048	System Clock Divider Register 2 (SIM_CLKDIV2)	32	R/W	0000_0000h	19.2.13/444
4004_804C	Flash Configuration Register 1 (SIM_FCFG1)	32	R	See section	19.2.14/444
4004_8050	Flash Configuration Register 2 (SIM_FCFG2)	32	R	See section	19.2.15/447
4004_8054	Unique Identification Register High (SIM_UIDH)	32	R	See section	19.2.16/448
4004_8058	Unique Identification Register Mid-High (SIM_UIDMH)	32	R	See section	19.2.17/448
4004_805C	Unique Identification Register Mid Low (SIM_UIDML)	32	R	See section	19.2.18/449
4004_8060	Unique Identification Register Low (SIM_UIDL)	32	R	See section	19.2.19/449

19.2.1 System Options Register 1 (SIM_SOPT1)

NOTE

The SOPT1 register is only reset on POR or LVD.

Address: 4004_7000h base + 0h offset = 4004_7000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	USBREGEN	USBSSTBY	USBVSTBY	0									OSC32KSEL		0	
W																
Reset	1*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RAMSIZE				0						Reserved					
W																
Reset	x*	x*	x*	x*	0*	0*	0*	0*	0*	0*	x*	x*	x*	x*	x*	x*

* Notes:

- x = Undefined at reset.

SIM_SOPT1 field descriptions

Field	Description
31 USBREGEN	USB voltage regulator enable Controls whether the USB voltage regulator is enabled. 0 USB voltage regulator is disabled. 1 USB voltage regulator is enabled.
30 USBSSTBY	USB voltage regulator in standby mode during Stop, VLPS, LLS and VLLS modes. Controls whether the USB voltage regulator is placed in standby mode during Stop, VLPS, LLS and VLLS modes. 0 USB voltage regulator not in standby during Stop, VLPS, LLS and VLLS modes. 1 USB voltage regulator in standby during Stop, VLPS, LLS and VLLS modes.

Table continues on the next page...

SIM_SOPT1 field descriptions (continued)

Field	Description
29 USBVSTBY	USB voltage regulator in standby mode during VLPR and VLPW modes Controls whether the USB voltage regulator is placed in standby mode during VLPR and VLPW modes. 0 USB voltage regulator not in standby during VLPR and VLPW modes. 1 USB voltage regulator in standby during VLPR and VLPW modes.
28–20 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
19–18 OSC32KSEL	32K oscillator clock select Selects the 32 kHz clock source (ERCLK32K) for LPTMR. This field is reset only on POR/LVD. 00 System oscillator (OSC32KCLK) 01 Reserved 10 RTC 32.768kHz oscillator 11 LPO 1 kHz
17–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
15–12 RAMSIZE	RAM size This field specifies the amount of system RAM available on the device. 0001 8 KB 0011 16 KB 0100 24 KB 0101 32 KB 0110 48 KB 0111 64 KB 1000 96 KB 1001 128 KB 1011 256 KB 1101 512 KB 1111 1024 KB
11–6 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
Reserved	This field is reserved.

19.2.2 SOPT1 Configuration Register (SIM_SOPT1CFG)

NOTE

The SOPT1CFG register is reset on System Reset not VLLS.

Address: 4004_7000h base + 4h offset = 4004_7004h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0					USSWE	UVSWE	URWE	0							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0						0		0							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SIM_SOPT1CFG field descriptions

Field	Description
31–27 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
26 USSWE	USB voltage regulator stop standby write enable Writing one to the USSWE bit allows the SOPT1 USBSSTBY bit to be written. This register bit clears after a write to USBSSTBY. 0 SOPT1 USBSSTBY cannot be written. 1 SOPT1 USBSSTBY can be written.
25 UVSWE	USB voltage regulator VLP standby write enable Writing one to the UVSWE bit allows the SOPT1 USBVSTBY bit to be written. This register bit clears after a write to USBVSTBY. 0 SOPT1 USBVSTBY cannot be written. 1 SOPT1 USBVSTBY can be written.
24 URWE	USB voltage regulator enable write enable Writing one to the URWE bit allows the SOPT1 USBREGEN bit to be written. This register bit clears after a write to USBREGEN. 0 SOPT1 USBREGEN cannot be written. 1 SOPT1 USBREGEN can be written.

Table continues on the next page...

SIM_SOPT1CFG field descriptions (continued)

Field	Description
23–10 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
9–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

19.2.3 System Options Register 2 (SIM_SOPT2)

SOPT2 contains the controls for selecting many of the module clock source options on this device. See the Clock Distribution chapter for more information including clocking diagrams and definitions of device clocks.

Address: 4004_7000h base + 1004h offset = 4004_8004h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0		0		0		0		0		0			USBSRC	0	PLLFLLSEL
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0			TRACECLKSEL	PTD7PAD	0	0		CLKOUTSEL			RTCCLKOUTSEL	0			
W																
Reset	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0

SIM_SOPT2 field descriptions

Field	Description
31–30 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
29–28 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
27–26 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
25–24 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

Table continues on the next page...

SIM_SOPT2 field descriptions (continued)

Field	Description
23–22 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
21–19 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
18 USBSRC	USB clock source select Selects the clock source for the USB 48 MHz clock. 0 Reserved 1 MCGFLLCLK, or MCGPLLCLK clock as selected by SOPT2[PLLFLLSEL], and then divided by the USB fractional divider as configured by SIM_CLKDIV2[USBFRAC, USBDIV].
17 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
16 PLLFLLSEL	PLL/FLL clock select Selects the MCGPLLCLK or MCGFLLCLK clock for various peripheral clocking options. 0 MCGFLLCLK clock 1 MCGPLLCLK clock
15–13 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
12 TRACECLKSEL	Debug trace clock select Selects the core/system clock, or MCG output clock (MCGOUTCLK) as the trace clock source. 0 MCGOUTCLK 1 Core/system clock
11 PTD7PAD	PTD7 pad drive strength Controls the output drive strength of the PTD7 pin by selecting either one or two pads to drive it. 0 Single-pad drive strength for PTD7. 1 Double pad drive strength for PTD7.
10 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
9–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7–5 CLKOUTSEL	CLKOUT select Selects the clock to output on the CLKOUT pin. 000 Reserved 001 Reserved 010 Flash clock 011 LPO clock (1 kHz) 100 MCGIRCLK 101 RTC 32.768kHz clock 110 OSCERCLK0 111 Reserved

Table continues on the next page...

SIM_SOPT2 field descriptions (continued)

Field	Description
4 RTCCLKOUTSEL	RTC clock out select Selects either the RTC 1 Hz clock or the 32.768kHz clock to be output on the RTC_CLKOUT pin. 0 RTC 1 Hz clock is output on the RTC_CLKOUT pin. 1 RTC 32.768kHz clock is output on the RTC_CLKOUT pin.
Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

19.2.4 System Options Register 4 (SIM_SOPT4)

Address: 4004_7000h base + 100Ch offset = 4004_800Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0		FTM0TRG1SRC	FTM0TRG0SRC	0	FTM2CLKSEL	FTM1CLKSEL	FTM0CLKSEL	0	0	FTM2CH0SRC		FTM1CH0SRC		0	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0							FTM2FLT0	0			FTM1FLT0	0	0	FTM0FLT1	FTM0FLT0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SIM_SOPT4 field descriptions

Field	Description
31–30 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
29 FTM0TRG1SRC	FlexTimer 0 Hardware Trigger 1 Source Select Selects the source of FTM0 hardware trigger 1. 0 PDB output trigger 1 drives FTM0 hardware trigger 1 1 FTM2 channel match drives FTM0 hardware trigger 1
28 FTM0TRG0SRC	FlexTimer 0 Hardware Trigger 0 Source Select Selects the source of FTM0 hardware trigger 0. 0 HSCMP0 output drives FTM0 hardware trigger 0 1 FTM1 channel match drives FTM0 hardware trigger 0

Table continues on the next page...

SIM_SOPT4 field descriptions (continued)

Field	Description
27 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
26 FTM2CLKSEL	FlexTimer 2 External Clock Pin Select Selects the external pin used to drive the clock to the FTM2 module. **NOTE:** The selected pin must also be configured for the FTM2 module external clock function through the appropriate pin control register in the port control module. 0 FTM2 external clock driven by FTM_CLK0 pin. 1 FTM2 external clock driven by FTM_CLK1 pin.
25 FTM1CLKSEL	FTM1 External Clock Pin Select Selects the external pin used to drive the clock to the FTM1 module. **NOTE:** The selected pin must also be configured for the FTM external clock function through the appropriate pin control register in the port control module. 0 FTM_CLK0 pin 1 FTM_CLK1 pin
24 FTM0CLKSEL	FlexTimer 0 External Clock Pin Select Selects the external pin used to drive the clock to the FTM0 module. **NOTE:** The selected pin must also be configured for the FTM external clock function through the appropriate pin control register in the port control module. 0 FTM_CLK0 pin 1 FTM_CLK1 pin
23 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
22 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
21–20 FTM2CH0SRC	FTM2 channel 0 input capture source select Selects the source for FTM2 channel 0 input capture. **NOTE:** When the FTM is not in input capture mode, clear this field. 00 FTM2_CH0 signal 01 CMP0 output 10 CMP1 output 11 Reserved
19–18 FTM1CH0SRC	FTM1 channel 0 input capture source select Selects the source for FTM1 channel 0 input capture. **NOTE:** When the FTM is not in input capture mode, clear this field. 00 FTM1_CH0 signal 01 CMP0 output 10 CMP1 output 11 USB start of frame pulse

Table continues on the next page...

SIM_SOPT4 field descriptions (continued)

Field	Description
17–9 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
8 FTM2FLT0	FTM2 Fault 0 Select Selects the source of FTM2 fault 0. **NOTE:** The pin source for fault 0 must be configured for the FTM module fault function through the appropriate PORTx pin control register. 0 FTM2_FLT0 pin 1 CMP0 out
7–5 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
4 FTM1FLT0	FTM1 Fault 0 Select Selects the source of FTM1 fault 0. **NOTE:** The pin source for fault 0 must be configured for the FTM module fault function through the appropriate pin control register in the port control module. 0 FTM1_FLT0 pin 1 CMP0 out
3 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
2 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
1 FTM0FLT1	FTM0 Fault 1 Select Selects the source of FTM0 fault 1. **NOTE:** The pin source for fault 1 must be configured for the FTM module fault function through the appropriate pin control register in the port control module. 0 FTM0_FLT1 pin 1 CMP1 out
0 FTM0FLT0	FTM0 Fault 0 Select Selects the source of FTM0 fault 0. **NOTE:** The pin source for fault 0 must be configured for the FTM module fault function through the appropriate pin control register in the port control module. 0 FTM0_FLT0 pin 1 CMP0 out

19.2.5 System Options Register 5 (SIM_SOPT5)

Address: 4004_7000h base + 1010h offset = 4004_8010h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0														0	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								UART1RXSRC		UART1TXSRC		UART0RXSRC		UART0TXSRC	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SIM_SOPT5 field descriptions

Field	Description
31–18 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
17–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
15–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7–6 UART1RXSRC	UART 1 receive data source select Selects the source for the UART 1 receive data. 00 UART1_RX pin 01 CMP0 10 CMP1 11 Reserved
5–4 UART1TXSRC	UART 1 transmit data source select Selects the source for the UART 1 transmit data. 00 UART1_TX pin 01 UART1_TX pin modulated with FTM1 channel 0 output 10 UART1_TX pin modulated with FTM2 channel 0 output 11 Reserved
3–2 UART0RXSRC	UART 0 receive data source select Selects the source for the UART 0 receive data. 00 UART0_RX pin 01 CMP0 10 CMP1 11 Reserved
UART0TXSRC	UART 0 transmit data source select Selects the source for the UART 0 transmit data.

Table continues on the next page...

SIM_SOPT5 field descriptions (continued)

Field	Description
	00 UART0_TX pin 01 UART0_TX pin modulated with FTM1 channel 0 output 10 UART0_TX pin modulated with FTM2 channel 0 output 11 Reserved

19.2.6 System Options Register 7 (SIM_SOPT7)

Address: 4004_7000h base + 1018h offset = 4004_8018h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								ADC0ALTTRGEN	0		ADC0PRETRGSEL	ADC0TRGSEL			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SIM_SOPT7 field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
15–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7 ADC0ALTTRGEN	ADC0 alternate trigger enable Enable alternative conversion triggers for ADC0. 0 PDB trigger selected for ADC0. 1 Alternate trigger selected for ADC0.
6–5 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
4 ADC0PRETRGSEL	ADC0 pretrigger select Selects the ADC0 pre-trigger source when alternative triggers are enabled through ADC0ALTTRGEN.

Table continues on the next page...

SIM_SOPT7 field descriptions (continued)

Field	Description
	0 Pre-trigger A 1 Pre-trigger B
ADC0TRGSEL	ADC0 trigger select Selects the ADC0 trigger source when alternative triggers are functional in stop and VLPS modes. . 0000 PDB external trigger pin input (PDB0_EXTRG) 0001 High speed comparator 0 output 0010 High speed comparator 1 output 0011 Reserved 0100 PIT trigger 0 0101 PIT trigger 1 0110 PIT trigger 2 0111 PIT trigger 3 1000 FTM0 trigger 1001 FTM1 trigger 1010 FTM2 trigger 1011 Reserved 1100 RTC alarm 1101 RTC seconds 1110 Low-power timer (LPTMR) trigger 1111 Reserved

19.2.7 System Device Identification Register (SIM_SDID)

Address: 4004_7000h base + 1024h offset = 4004_8024h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																REVID				DIEID					FAMID			PINID			
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	x*	x*	x*	x*	0	0	1	0	0	x*	x*	x*	x*	x*	x*	x*

* Notes:

- x = Undefined at reset.

SIM_SDID field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
15–12 REVID	Device revision number Specifies the silicon implementation number for the device.
11–7 DIEID	Device die number Specifies the silicon implementation number for the device.

Table continues on the next page...

SIM_SDID field descriptions (continued)

Field	Description
6–4 FAMID	Kinetis family identification Specifies the Kinetis family of the device. 000 K10 or K12 Family 001 K20 or K22 Family 010 K30 Family or K11 Family or K61 Family 011 K40 Family or K21 Family 100 K60 or K62 Family 101 K70 Family 110 KW24 Family 111 Reserved
PINID	Pincount identification Specifies the pincount of the device. 0000 Reserved 0001 Reserved 0010 32-pin 0011 Reserved 0100 48-pin 0101 64-pin 0110 80-pin 0111 81-pin or 121-pin 1000 100-pin 1001 121-pin 1010 144-pin 1011 Custom pinout (WLCSP) 1100 Reserved 1101 Reserved 1110 256-pin 1111 Reserved

19.2.8 System Clock Gating Control Register 4 (SIM_SCGC4)

Address: 4004_7000h base + 1034h offset = 4004_8034h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	1				0							0	CMP	USBOTG	0	
W																
Reset	1	1	1	1	0	0	0	0	0	0	0	1	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0		0	UART2	UART1	UART0	0		I2C1	I2C0	1		0	CMT	EWM	0
W																
Reset	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0

SIM_SCGC4 field descriptions

Field	Description
31–28 Reserved	This field is reserved. This read-only field is reserved and always has the value 1.
27–21 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
20 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
19 CMP	Comparator Clock Gate Control This bit controls the clock gate to the comparator module. 0 Clock disabled 1 Clock enabled
18 USBOTG	USB Clock Gate Control This bit controls the clock gate to the USB module. 0 Clock disabled 1 Clock enabled
17–14 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
13 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
12 UART2	UART2 Clock Gate Control This bit controls the clock gate to the UART2 module. 0 Clock disabled 1 Clock enabled

Table continues on the next page...

SIM_SCGC4 field descriptions (continued)

Field	Description
11 UART1	UART1 Clock Gate Control This bit controls the clock gate to the UART1 module. 0 Clock disabled 1 Clock enabled
10 UART0	UART0 Clock Gate Control This bit controls the clock gate to the UART0 module. 0 Clock disabled 1 Clock enabled
9–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7 I2C1	I2C1 Clock Gate Control This bit controls the clock gate to the I ^{2} C1 module. 0 Clock disabled 1 Clock enabled
6 I2C0	I2C0 Clock Gate Control This bit controls the clock gate to the I ^{2} C0 module. 0 Clock disabled 1 Clock enabled
5–4 Reserved	This field is reserved. This read-only field is reserved and always has the value 1.
3 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
2 CMT	CMT Clock Gate Control This bit controls the clock gate to the CMT module. 0 Clock disabled 1 Clock enabled
1 EWM	EWM Clock Gate Control This bit controls the clock gate to the EWM module. 0 Clock disabled 1 Clock enabled
0 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

19.2.9 System Clock Gating Control Register 5 (SIM_SCGC5)

Address: 4004_7000h base + 1038h offset = 4004_8038h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0													1	0	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0		PORTE	PORTD	PORTC	PORTB	PORTA	1		0	0	0	0		1	LPTMR
W																
Reset	0	0	0	0	0	0	0	1	1	0	0	0	0	0	1	0

SIM_SCGC5 field descriptions

Field	Description
31–19 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
18 Reserved	This field is reserved. This read-only field is reserved and always has the value 1.
17–14 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
13 PORTE	Port E Clock Gate Control This bit controls the clock gate to the Port E module. 0 Clock disabled 1 Clock enabled
12 PORTD	Port D Clock Gate Control This bit controls the clock gate to the Port D module. 0 Clock disabled 1 Clock enabled
11 PORTC	Port C Clock Gate Control This bit controls the clock gate to the Port C module. 0 Clock disabled 1 Clock enabled
10 PORTB	Port B Clock Gate Control This bit controls the clock gate to the Port B module.

Table continues on the next page...

SIM_SCGC5 field descriptions (continued)

Field	Description
	0 Clock disabled 1 Clock enabled
9 PORTA	Port A Clock Gate Control This bit controls the clock gate to the Port A module. 0 Clock disabled 1 Clock enabled
8–7 Reserved	This field is reserved. This read-only field is reserved and always has the value 1.
6 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
5 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
4 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
3–2 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
1 Reserved	This field is reserved. This read-only field is reserved and always has the value 1.
0 LPTMR	Low Power Timer Access Control This bit controls software access to the Low Power Timer module. 0 Access disabled 1 Access enabled

19.2.10 System Clock Gating Control Register 6 (SIM_SCGC6)

Address: 4004_7000h base + 103Ch offset = 4004_803Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	1	RTC	0	ADC0	FTM2	FTM1	FTM0	PIT	PDB	USBDCD	0		CRC	0	
W																
Reset	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	I2S	0	SPI1	SPI0	0	0	RNGA	0	0	0	0	0	0		DMAMUX	FTF
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

SIM_SCGC6 field descriptions

Field	Description
31 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
30 Reserved	This field is reserved. This read-only field is reserved and always has the value 1.
29 RTC	RTC Access Control This bit controls software access and interrupts to the RTC module. 0 Access and interrupts disabled 1 Access and interrupts enabled
28 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
27 ADC0	ADC0 Clock Gate Control This bit controls the clock gate to the ADC0 module. 0 Clock disabled 1 Clock enabled
26 FTM2	FTM2 Clock Gate Control This bit controls the clock gate to the FTM2 module. 0 Clock disabled 1 Clock enabled
25 FTM1	FTM1 Clock Gate Control This bit controls the clock gate to the FTM1 module. 0 Clock disabled 1 Clock enabled
24 FTM0	FTM0 Clock Gate Control This bit controls the clock gate to the FTM0 module. 0 Clock disabled 1 Clock enabled
23 PIT	PIT Clock Gate Control This bit controls the clock gate to the PIT module. 0 Clock disabled 1 Clock enabled
22 PDB	PDB Clock Gate Control This bit controls the clock gate to the PDB module. 0 Clock disabled 1 Clock enabled
21 USBDCD	USB DCD Clock Gate Control This bit controls the clock gate to the USB DCD module.

Table continues on the next page...

SIM_SCGC6 field descriptions (continued)

Field	Description
	0 Clock disabled 1 Clock enabled
20–19 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
18 CRC	CRC Clock Gate Control This bit controls the clock gate to the CRC module. 0 Clock disabled 1 Clock enabled
17–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
15 I2S	I2S Clock Gate Control This bit controls the clock gate to the I ^{2} S module. 0 Clock disabled 1 Clock enabled
14 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
13 SPI1	SPI1 Clock Gate Control This bit controls the clock gate to the SPI1 module. 0 Clock disabled 1 Clock enabled
12 SPI0	SPI0 Clock Gate Control This bit controls the clock gate to the SPI0 module. 0 Clock disabled 1 Clock enabled
11 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
10 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
9 RNGA	RNGA Clock Gate Control This bit controls the clock gate to the RNGA module.
8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
6 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
5 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
4 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

Table continues on the next page...

SIM_SCGC6 field descriptions (continued)

Field	Description
3–2 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
1 DMAMUX	DMA Mux Clock Gate Control This bit controls the clock gate to the DMA Mux module. 0 Clock disabled 1 Clock enabled
0 FTF	Flash Memory Clock Gate Control This bit controls the clock gate to the flash memory. Flash reads are still supported while the flash memory is clock gated, but entry into low power modes is blocked. 0 Clock disabled 1 Clock enabled

19.2.11 System Clock Gating Control Register 7 (SIM_SCGC7)

Address: 4004_7000h base + 1040h offset = 4004_8040h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0												0	0	DMA	0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

SIM_SCGC7 field descriptions

Field	Description
31–4 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
3 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
2 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
1 DMA	DMA Clock Gate Control This bit controls the clock gate to the DMA module. 0 Clock disabled 1 Clock enabled
0 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

19.2.12 System Clock Divider Register 1 (SIM_CLKDIV1)

When updating CLKDIV1, update all fields using the one write command.

NOTE

The CLKDIV1 register cannot be written to when the device is in VLPR mode.

Address: 4004_7000h base + 1044h offset = 4004_8044h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	OUTDIV1				OUTDIV2				0				OUTDIV4				0															
W																																
Reset	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	1*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*

SIM_CLKDIV1 field descriptions

Field	Description
31–28 OUTDIV1	Clock 1 output divider value This field sets the divide value for the core/system clock from MCGOUTCLK. At the end of reset, it is loaded with either 0000 or 0111 depending on FTF_FOPT[LPBOOT]. 0000 Divide-by-1. 0001 Divide-by-2. 0010 Divide-by-3. 0011 Divide-by-4. 0100 Divide-by-5. 0101 Divide-by-6. 0110 Divide-by-7. 0111 Divide-by-8. 1000 Divide-by-9. 1001 Divide-by-10. 1010 Divide-by-11. 1011 Divide-by-12. 1100 Divide-by-13. 1101 Divide-by-14. 1110 Divide-by-15. 1111 Divide-by-16.
27–24 OUTDIV2	Clock 2 output divider value This field sets the divide value for the bus clock from MCGOUTCLK. At the end of reset, it is loaded with either 0000 or 0111 depending on FTF_FOPT[LPBOOT]. The bus clock frequency must be an integer divide of the core/system clock frequency. 0000 Divide-by-1. 0001 Divide-by-2. 0010 Divide-by-3. 0011 Divide-by-4. 0100 Divide-by-5.

Table continues on the next page...

SIM_CLKDIV1 field descriptions (continued)

Field	Description
	0101 Divide-by-6. 0110 Divide-by-7. 0111 Divide-by-8. 1000 Divide-by-9. 1001 Divide-by-10. 1010 Divide-by-11. 1011 Divide-by-12. 1100 Divide-by-13. 1101 Divide-by-14. 1110 Divide-by-15. 1111 Divide-by-16.
23–20 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
19–16 OUTDIV4	Clock 4 output divider value This field sets the divide value for the flash clock from MCGOUTCLK. At the end of reset, it is loaded with either 0001 or 1111 depending on FTF_FOPT[LPBOOT]. The flash clock frequency must be an integer divide of the system clock frequency. 0000 Divide-by-1. 0001 Divide-by-2. 0010 Divide-by-3. 0011 Divide-by-4. 0100 Divide-by-5. 0101 Divide-by-6. 0110 Divide-by-7. 0111 Divide-by-8. 1000 Divide-by-9. 1001 Divide-by-10. 1010 Divide-by-11. 1011 Divide-by-12. 1100 Divide-by-13. 1101 Divide-by-14. 1110 Divide-by-15. 1111 Divide-by-16.
Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

19.2.13 System Clock Divider Register 2 (SIM_CLKDIV2)

Address: 4004_7000h base + 1048h offset = 4004_8048h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0												USBDIV			USBFRAC
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SIM_CLKDIV2 field descriptions

Field	Description
31–4 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
3–1 USBDIV	USB clock divider divisor This field sets the divide value for the fractional clock divider when the MCGFLLCLK, or MCGPLLCLK clock is the USB clock source (SOPT2[USBSRC] = 1). Divider output clock = Divider input clock × [(USBFRAC+1) / (USBDIV+1)]
0 USBFRAC	USB clock divider fraction This field sets the fraction multiply value for the fractional clock divider when the MCGFLLCLK, or MCGPLLCLK clock is the USB clock source (SOPT2[USBSRC] = 1). Divider output clock = Divider input clock × [(USBFRAC+1) / (USBDIV+1)]

19.2.14 Flash Configuration Register 1 (SIM_FCFG1)

For devices with FlexNVM: The reset value of EESIZE and DEPART are based on user programming in user IFR via the PGMPART flash command.

For devices with program flash only:

Address: 4004_7000h base + 104Ch offset = 4004_804Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	NVMSIZE				PFSIZE				0				EESIZE			
W																
Reset	1*	1*	1*	1*	1*	1*	1*	1*	0*	0*	0*	0*	1*	1*	1*	1*

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0				DEPART				0						FLASHDOZE	FLASHDIS
W																
Reset	0*	0*	0*	0*	1*	1*	1*	1*	0*	0*	0*	0*	0*	0*	0*	0*

SIM_FCFG1 field descriptions

Field	Description
31–28 NVMSIZE	FlexNVM size This field specifies the amount of FlexNVM memory available on the device . Undefined values are reserved. 0000 0 KB of FlexNVM 0011 32 KB of FlexNVM 0101 64 KB of FlexNVM 0111 128 KB of FlexNVM 1001 256 KB of FlexNVM 1011 512 KB of FlexNVM 1111 For devices with FlexNVM (SIM_FCFG2[PFLSH]=0): 256 KB of FlexNVM, 32 KB protection region. For devices without FlexNVM (SIM_FCFG2[PFLSH]=1): 0 KB of FlexNVM
27–24 PFSIZE	Program flash size This field specifies the amount of program flash memory available on the device . Undefined values are reserved. 0011 32 KB of program flash memory 0101 64 KB of program flash memory 0111 128 KB of program flash memory 1001 256 KB of program flash memory

Table continues on the next page...

SIM_FCFG1 field descriptions (continued)

Field	Description
	1011 512 KB of program flash memory 1101 1024 KB of program flash memory 1111 512 KB of program flash memory
23–20 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
19–16 EESIZE	EEPROM size EEPROM data size . 0000 16 KB 0001 8 KB 0010 4 KB 0011 2 KB 0100 1 KB 0101 512 Bytes 0110 256 Bytes 0111 128 Bytes 1000 64 Bytes 1001 32 Bytes 1010-1110 Reserved 1111 0 Bytes
15–12 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
11–8 DEPART	FlexNVM partition For devices with FlexNVM: Data flash / EEPROM backup split . See DEPART bit description in FTFL chapter. For devices without FlexNVM: Reserved
7–2 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
1 FLASHDOZE	Flash Doze When set, Flash memory is disabled for the duration of Wait mode. An attempt by the DMA or other bus master to access the Flash when the Flash is disabled will result in a bus error. This bit should be clear during VLP modes. The Flash will be automatically enabled again at the end of Wait mode so interrupt vectors do not need to be relocated out of Flash memory. The wakeup time from Wait mode is extended when this bit is set. 0 Flash remains enabled during Wait mode 1 Flash is disabled for the duration of Wait mode
0 FLASHDIS	Flash Disable Flash accesses are disabled (and generate a bus error) and the Flash memory is placed in a low power state. This bit should not be changed during VLP modes. Relocate the interrupt vectors out of Flash memory before disabling the Flash. 0 Flash is enabled 1 Flash is disabled

19.2.15 Flash Configuration Register 2 (SIM_FCFG2)

Address: 4004_7000h base + 1050h offset = 4004_8050h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	SWAPPFLSH	MAXADDR0							PFLSH	MAXADDR1						
W																
Reset	0*	1*	1*	1*	1*	1*	1*	1*	0*	1*	1*	1*	1*	1*	1*	1*

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0															
W																
Reset	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*

SIM_FCFG2 field descriptions

Field	Description
31 SWAPPFLSH	Swap program flash For devices without FlexNVM: Indicates that swap is active . 0 Swap is not active. 1 Swap is active.
30–24 MAXADDR0	Max address block 0 This field concatenated with 13 trailing zeros indicates the first invalid address of each program flash block. For example, if MAXADDR0 = 0x20 the first invalid address of flash block 0 is 0x0004_0000. This would be the MAXADDR0 value for a device with 256 KB program flash in flash block 0.

Table continues on the next page...

SIM_FCFG2 field descriptions (continued)

Field	Description
23 PFLSH	Program flash only For devices with FlexNVM, this bit is always clear. For devices without FlexNVM, this bit is always set. 0 Device supports FlexNVM 1 Program Flash only, device does not support FlexNVM
22–16 MAXADDR1	Max address block 1 For devices with FlexNVM: This field concatenated with 13 trailing zeros plus the FlexNVM base address indicates the first invalid address of the FlexNVM flash block. For example, if MAXADDR1 = 0x20 the first invalid address of FlexNVM flash block is 0x4_0000 + 0x1000_0000 . This would be the MAXADDR1 value for a device with 256 KB FlexNVM. For devices with program flash only: This field equals zero if there is only one program flash block, otherwise it equals the value of the MAXADDR0 field. For example, with MAXADDR0 = MAXADDR1 = 0x20 the first invalid address of flash block 1 is 0x4_0000 + 0x4_0000. This would be the MAXADDR1 value for a device with 512 KB program flash memory across two flash blocks and no FlexNVM.
Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

19.2.16 Unique Identification Register High (SIM_UIDH)

Address: 4004_7000h base + 1054h offset = 4004_8054h

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

SIM_UIDH field descriptions

Field	Description
UID	Unique Identification Unique identification for the device.

19.2.17 Unique Identification Register Mid-High (SIM_UIDMH)

Address: 4004_7000h base + 1058h offset = 4004_8058h

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

SIM_UIDMH field descriptions

Field	Description
UID	Unique Identification Unique identification for the device.

19.2.18 Unique Identification Register Mid Low (SIM_UIDML)

Address: 4004_7000h base + 105Ch offset = 4004_805Ch

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

SIM_UIDML field descriptions

Field	Description
UID	Unique Identification Unique identification for the device.

19.2.19 Unique Identification Register Low (SIM_UIDL)

Address: 4004_7000h base + 1060h offset = 4004_8060h

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

SIM_UIDL field descriptions

Field	Description
UID	Unique Identification Unique identification for the device.

19.3 Functional description

For more information about the functions of SIM, see the Introduction section.

Chapter 20
MCU: Reset Control Module (RCM)

20.1 Introduction

Information found here describes the registers of the Reset Control Module (RCM). The RCM implements many of the reset functions for the chip. See the chip's reset chapter for more information.

See AN4503: Power Management for Kinetis MCUs for further details on using the RCM.

20.2 Reset memory map and register descriptions

The RCM Memory Map/Register Definition can be found here.

The Reset Control Module (RCM) registers provide reset status information and reset filter control.

NOTE

> The RCM registers can be written only in supervisor mode. Write accesses in user mode are blocked and will result in a bus error.

RCM memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4007_F000	System Reset Status Register 0 (RCM_SRS0)	8	R	82h	20.2.1/452
4007_F001	System Reset Status Register 1 (RCM_SRS1)	8	R	00h	20.2.2/453
4007_F004	Reset Pin Filter Control register (RCM_RPFC)	8	R/W	00h	20.2.3/455
4007_F005	Reset Pin Filter Width register (RCM_RPFW)	8	R/W	00h	20.2.4/456
4007_F007	Mode Register (RCM_MR)	8	R	00h	20.2.5/457

20.2.1 System Reset Status Register 0 (RCM_SRS0)

This register includes read-only status flags to indicate the source of the most recent reset. The reset state of these bits depends on what caused the MCU to reset.

NOTE

The reset value of this register depends on the reset source:

- POR (including LVD) — 0x82
- LVD (without POR) — 0x02
- VLLS mode wakeup due to $\overline{\text{RESET}}$ pin assertion — 0x41
- VLLS mode wakeup due to other wakeup sources — 0x01
- Other reset — a bit is set if its corresponding reset source caused the reset

Address: 4007_F000h base + 0h offset = 4007_F000h

Bit	7	6	5	4	3	2	1	0
Read	POR	PIN	WDOG	0	LOL	LOC	LVD	WAKEUP
Write								
Reset	1	0	0	0	0	0	1	0

RCM_SRS0 field descriptions

Field	Description
7 POR	Power-On Reset Indicates a reset has been caused by the power-on detection logic. Because the internal supply voltage was ramping up at the time, the low-voltage reset (LVD) status bit is also set to indicate that the reset occurred while the internal supply was below the LVD threshold. 0 Reset not caused by POR 1 Reset caused by POR
6 PIN	External Reset Pin Indicates a reset has been caused by an active-low level on the external $\overline{\text{RESET}}$ pin. 0 Reset not caused by external reset pin 1 Reset caused by external reset pin
5 WDOG	Watchdog Indicates a reset has been caused by the watchdog timer timing out. 0 Reset not caused by watchdog timeout 1 Reset caused by watchdog timeout
4 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
3 LOL	Loss-of-Lock Reset

Table continues on the next page...

RCM_SRS0 field descriptions (continued)

Field	Description
	Indicates a reset has been caused by a loss of lock in the MCG PLL. See the MCG description for information on the loss-of-clock event. 0 Reset not caused by a loss of lock in the PLL 1 Reset caused by a loss of lock in the PLL
2 LOC	Loss-of-Clock Reset Indicates a reset has been caused by a loss of external clock. The MCG clock monitor must be enabled for a loss of clock to be detected. Refer to the detailed MCG description for information on enabling the clock monitor. 0 Reset not caused by a loss of external clock. 1 Reset caused by a loss of external clock.
1 LVD	Low-Voltage Detect Reset If PMC_LVDSC1[LVDRE] is set and the supply drops below the LVD trip voltage, an LVD reset occurs. This field is also set by POR. 0 Reset not caused by LVD trip or POR 1 Reset caused by LVD trip or POR
0 WAKEUP	Low Leakage Wakeup Reset Indicates a reset has been caused by an enabled LLWU module wakeup source while the chip was in a low leakage mode. In LLS mode, the $\overline{\text{RESET}}$ pin is the only wakeup source that can cause this reset. Any enabled wakeup source in a VLLSx mode causes a reset. This bit is cleared by any reset except WAKEUP. 0 Reset not caused by LLWU module wakeup source 1 Reset caused by LLWU module wakeup source

20.2.2 System Reset Status Register 1 (RCM_SRS1)

This register includes read-only status flags to indicate the source of the most recent reset. The reset state of these bits depends on what caused the MCU to reset.

NOTE

The reset value of this register depends on the reset source:

- POR (including LVD) — 0x00
- LVD (without POR) — 0x00
- VLLS mode wakeup — 0x00
- Other reset — a bit is set if its corresponding reset source caused the reset

Address: 4007_F000h base + 1h offset = 4007_F001h

Bit	7	6	5	4	3	2	1	0
Read	TAMPER	0	SACKERR	EZPT	MDM_AP	SW	LOCKUP	JTAG
Write								
Reset	0	0	0	0	0	0	0	0

RCM_SRS1 field descriptions

Field	Description
7 TAMPER	Tamper detect Indicates a reset was caused by tamper detect. 0 Reset not caused by tamper detect 1 Reset caused by tamper detect.
6 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
5 SACKERR	Stop Mode Acknowledge Error Reset Indicates that after an attempt to enter Stop mode, a reset has been caused by a failure of one or more peripherals to acknowledge within approximately one second to enter stop mode. 0 Reset not caused by peripheral failure to acknowledge attempt to enter stop mode 1 Reset caused by peripheral failure to acknowledge attempt to enter stop mode
4 EZPT	EzPort Reset Indicates a reset has been caused by EzPort receiving the RESET command while the device is in EzPort mode. 0 Reset not caused by EzPort receiving the RESET command while the device is in EzPort mode 1 Reset caused by EzPort receiving the RESET command while the device is in EzPort mode
3 MDM_AP	MDM-AP System Reset Request Indicates a reset has been caused by the host debugger system setting of the System Reset Request bit in the MDM-AP Control Register. 0 Reset not caused by host debugger system setting of the System Reset Request bit 1 Reset caused by host debugger system setting of the System Reset Request bit
2 SW	Software Indicates a reset has been caused by software setting of SYSRESETREQ bit in Application Interrupt and Reset Control Register in the ARM core. 0 Reset not caused by software setting of SYSRESETREQ bit 1 Reset caused by software setting of SYSRESETREQ bit
1 LOCKUP	Core Lockup Indicates a reset has been caused by the ARM core indication of a LOCKUP event. 0 Reset not caused by core LOCKUP event 1 Reset caused by core LOCKUP event
0 JTAG	JTAG Generated Reset

Table continues on the next page...

RCM_SRS1 field descriptions (continued)

Field	Description
	Indicates a reset has been caused by JTAG selection of certain IR codes: EZPORT, EXTEST, HIGHZ, and CLAMP. 0 Reset not caused by JTAG 1 Reset caused by JTAG

20.2.3 Reset Pin Filter Control register (RCM_RPFC)

NOTE

The reset values of bits 2-0 are for Chip POR only. They are unaffected by other reset types.

NOTE

The bus clock filter is reset when disabled or when entering stop mode. The LPO filter is reset when disabled or when entering any low leakage stop mode .

Address: 4007_F000h base + 4h offset = 4007_F004h

<table>
<tr><td>Bit</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>Read</td><td colspan="5">0</td><td rowspan="2">RSTFLTSS</td><td colspan="2" rowspan="2">RSTFLTSRW</td></tr>
<tr><td>Write</td><td colspan="5"></td></tr>
<tr><td>Reset</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
</table>

RCM_RPFC field descriptions

Field	Description
7–3 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
2 RSTFLTSS	Reset Pin Filter Select in Stop Mode Selects how the reset pin filter is enabled in Stop and VLPS modes 0 All filtering disabled 1 LPO clock filter enabled
RSTFLTSRW	Reset Pin Filter Select in Run and Wait Modes Selects how the reset pin filter is enabled in run and wait modes. 00 All filtering disabled 01 Bus clock filter enabled for normal operation 10 LPO clock filter enabled for normal operation 11 Reserved

20.2.4 Reset Pin Filter Width register (RCM_RPFW)

NOTE

The reset values of the bits in the RSTFLTSEL field are for Chip POR only. They are unaffected by other reset types.

Address: 4007_F000h base + 5h offset = 4007_F005h

Bit	7	6	5	4	3	2	1	0
Read	0			RSTFLTSEL				
Write								
Reset	0	0	0	0	0	0	0	0

RCM_RPFW field descriptions

Field	Description
7–5 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
RSTFLTSEL	Reset Pin Filter Bus Clock Select Selects the reset pin bus clock filter width. 00000 Bus clock filter count is 1 00001 Bus clock filter count is 2 00010 Bus clock filter count is 3 00011 Bus clock filter count is 4 00100 Bus clock filter count is 5 00101 Bus clock filter count is 6 00110 Bus clock filter count is 7 00111 Bus clock filter count is 8 01000 Bus clock filter count is 9 01001 Bus clock filter count is 10 01010 Bus clock filter count is 11 01011 Bus clock filter count is 12 01100 Bus clock filter count is 13 01101 Bus clock filter count is 14 01110 Bus clock filter count is 15 01111 Bus clock filter count is 16 10000 Bus clock filter count is 17 10001 Bus clock filter count is 18 10010 Bus clock filter count is 19 10011 Bus clock filter count is 20 10100 Bus clock filter count is 21 10101 Bus clock filter count is 22 10110 Bus clock filter count is 23 10111 Bus clock filter count is 24 11000 Bus clock filter count is 25

Table continues on the next page...

RCM_RPFW field descriptions (continued)

Field	Description
	11001 Bus clock filter count is 26 11010 Bus clock filter count is 27 11011 Bus clock filter count is 28 11100 Bus clock filter count is 29 11101 Bus clock filter count is 30 11110 Bus clock filter count is 31 11111 Bus clock filter count is 32

20.2.5 Mode Register (RCM_MR)

This register includes read-only status flags to indicate the state of the mode pins during the last Chip Reset.

Address: 4007_F000h base + 7h offset = 4007_F007h

Bit	7	6	5	4	3	2	1	0
Read	0						EZP_MS	0
Write								
Reset	0	0	0	0	0	0	0	0

RCM_MR field descriptions

Field	Description
7–2 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
1 EZP_MS	EZP_MS_B pin state Reflects the state of the $\overline{\text{EZP\_MS}}$ pin during the last Chip Reset 0 Pin deasserted (logic 1) 1 Pin asserted (logic 0)
0 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

Chapter 21
MCU: System Mode Controller (SMC)

21.1 Introduction

NOTE

> For the chip-specific implementation details of this module's instances, see the chip configuration information.

The System Mode Controller (SMC) is responsible for sequencing the system into and out of all low-power Stop and Run modes.

Specifically, it monitors events to trigger transitions between power modes while controlling the power, clocks, and memories of the system to achieve the power consumption and functionality of that mode.

This chapter describes all the available low-power modes, the sequence followed to enter/ exit each mode, and the functionality available while in each of the modes.

The SMC is able to function during even the deepest low power modes.

See AN4503: Power Management for Kinetis MCUs for further details on using the SMC.

21.2 Modes of operation

The ARM CPU has three primary modes of operation:

- Run
- Sleep
- Deep Sleep

The WFI or WFE instruction is used to invoke Sleep and Deep Sleep modes. Run, Wait, and Stop are the common terms used for the primary operating modes of Kinetis microcontrollers.

The following table shows the translation between the ARM CPU modes and the Kinetis MCU power modes.

ARM CPU mode	MCU mode
Sleep	Wait
Deep Sleep	Stop

Accordingly, the ARM CPU documentation refers to sleep and deep sleep, while the Kinetis MCU documentation normally uses wait and stop.

In addition, Kinetis MCUs also augment Stop, Wait, and Run modes in a number of ways. The power management controller (PMC) contains a run and a stop mode regulator. Run regulation is used in normal run, wait and stop modes. Stop mode regulation is used during all very low power and low leakage modes. During stop mode regulation, the bus frequencies are limited in the very low power modes.

The SMC provides the user with multiple power options. The Very Low Power Run (VLPR) mode can drastically reduce run time power when maximum bus frequency is not required to handle the application needs. From Normal Run mode, the Run Mode (RUNM) field can be modified to change the MCU into VLPR mode when limited frequency is sufficient for the application. From VLPR mode, a corresponding wait (VLPW) and stop (VLPS) mode can be entered.

Depending on the needs of the user application, a variety of stop modes are available that allow the state retention, partial power down or full power down of certain logic and/or memory. I/O states are held in all modes of operation. Several registers are used to configure the various modes of operation for the device.

The following table describes the power modes available for the device.

Table 21-1. Power modes

Mode	Description
RUN	The MCU can be run at full speed and the internal supply is fully regulated, that is, in run regulation. This mode is also referred to as Normal Run mode.
WAIT	The core clock is gated off. The system clock continues to operate. Bus clocks, if enabled, continue to operate. Run regulation is maintained.
STOP	The core clock is gated off. System clocks to other masters and bus clocks are gated off after all stop acknowledge signals from supporting peripherals are valid.
VLPR	The core, system, bus, and flash clock maximum frequencies are restricted in this mode. See the Power Management chapter for details about the maximum allowable frequencies.
VLPW	The core clock is gated off. The system, bus, and flash clocks continue to operate, although their maximum frequency is restricted. See the Power Management chapter for details on the maximum allowable frequencies.
VLPS	The core clock is gated off. System clocks to other masters and bus clocks are gated off after all stop acknowledge signals from supporting peripherals are valid.

Table continues on the next page...

Table 21-1. Power modes (continued)

Mode	Description
LLS	The core clock is gated off. System clocks to other masters and bus clocks are gated off after all stop acknowledge signals from supporting peripherals are valid. The MCU is placed in a low leakage mode by reducing the voltage to internal logic. All system RAM contents, internal logic and I/O states are retained.
VLLS2	The core clock is gated off. System clocks to other masters and bus clocks are gated off after all stop acknowledge signals from supporting peripherals are valid. The MCU is placed in a low leakage mode by powering down the internal logic and the system RAM3 partition. The system RAM2 partition can be optionally retained using VLLS CTRL[RAM2PO].
VLLS0	The core clock is gated off. System clocks to other masters and bus clocks are gated off after all stop acknowledge signals from supporting peripherals are valid. The MCU is placed in a low leakage mode by powering down the internal logic and all system RAM. I/O states are held. Internal logic states are not retained. The 1kHz LPO clock is disabled and the power on reset (POR) circuit can be optionally enabled using VLLSCTRL[PORPO].

21.3 Memory map and register descriptions

Information about the registers related to the system mode controller can be found here.

Different SMC registers reset on different reset types. Each register's description provides details. For more information about the types of reset on this chip, refer to the Reset section details.

NOTE

The SMC registers can be written only in supervisor mode. Write accesses in user mode are blocked and will result in a bus error.

NOTE

Before executing the WFI instruction, the last register written to must be read back. This ensures that all register writes associated with setting up the low power mode being entered have completed before the MCU enters the low power mode. Failure to do this may result in the low power mode not being entered correctly.

SMC memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4007_E000	Power Mode Protection register (SMC_PMPROT)	8	R/W	00h	21.3.1/462
4007_E001	Power Mode Control register (SMC_PMCTRL)	8	R/W	00h	21.3.2/463

Table continues on the next page...

SMC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4007_E002	VLLS Control Register (SMC_VLLSCTRL)	8	R/W	03h	21.3.3/465
4007_E003	Power Mode Status register (SMC_PMSTAT)	8	R	01h	21.3.4/466

21.3.1 Power Mode Protection register (SMC_PMPROT)

This register provides protection for entry into any low-power run or stop mode. The enabling of the low-power run or stop mode occurs by configuring the Power Mode Control register (PMCTRL).

The PMPROT register can be written only once after any system reset.

If the MCU is configured for a disallowed or reserved power mode, the MCU remains in its current power mode. For example, if the MCU is in normal RUN mode and AVLP is 0, an attempt to enter VLPR mode using PMCTRL[RUNM] is blocked and PMCTRL[RUNM] remains 00b, indicating the MCU is still in Normal Run mode.

NOTE

> This register is reset on Chip Reset not VLLS and by reset types that trigger Chip Reset not VLLS. It is unaffected by reset types that do not trigger Chip Reset not VLLS. See the Reset section details for more information.

Address: 4007_E000h base + 0h offset = 4007_E000h

Bit	7	6	5	4	3	2	1	0
Read	0	0	AVLP	0	ALLS	0	AVLLS	0
Write	▨	▨		▨		▨		▨
Reset	0	0	0	0	0	0	0	0

SMC_PMPROT field descriptions

Field	Description
7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
6 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
5 AVLP	Allow Very-Low-Power Modes Provided the appropriate control bits are set up in PMCTRL, this write-once field allows the MCU to enter any very-low-power mode (VLPR, VLPW, and VLPS). 0 VLPR, VLPW, and VLPS are not allowed. 1 VLPR, VLPW, and VLPS are allowed.

Table continues on the next page...

SMC_PMPROT field descriptions (continued)

Field	Description
4 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
3 ALLS	Allow Low-Leakage Stop Mode Provided the appropriate control bits are set up in PMCTRL, this write-once field allows the MCU to enter any low-leakage stop mode (LLS). 0 LLS is not allowed 1 LLS is allowed
2 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
1 AVLLS	Allow Very-Low-Leakage Stop Mode Provided the appropriate control bits are set up in PMCTRL, this write once bit allows the MCU to enter any very-low-leakage stop mode (VLLSx). 0 Any VLLSx mode is not allowed 1 Any VLLSx mode is allowed
0 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

21.3.2 Power Mode Control register (SMC_PMCTRL)

The PMCTRL register controls entry into low-power Run and Stop modes, provided that the selected power mode is allowed via an appropriate setting of the protection (PMPROT) register.

NOTE

This register is reset on Chip POR not VLLS and by reset types that trigger Chip POR not VLLS. It is unaffected by reset types that do not trigger Chip POR not VLLS. See the Reset section details for more information.

Address: 4007_E000h base + 1h offset = 4007_E001h

Bit	7	6	5	4	3	2	1	0
Read	LPWUI	RUNM		0	STOPA	STOPM		
Write								
Reset	0	0	0	0	0	0	0	0

SMC_PMCTRL field descriptions

Field	Description
7 LPWUI	Low-Power Wake Up On Interrupt

Table continues on the next page...

SMC_PMCTRL field descriptions (continued)

Field	Description
	Causes the SMC to exit to normal RUN mode when any active MCU interrupt occurs while in a VLP mode (VLPR, VLPW or VLPS). **NOTE:** If VLPS mode was entered directly from RUN mode, the SMC will always exit back to normal RUN mode regardless of the LPWUI setting. **NOTE:** LPWUI must be modified only while the system is in RUN mode, that is, when PMSTAT=RUN. 0 The system remains in a VLP mode on an interrupt 1 The system exits to Normal RUN mode on an interrupt
6–5 RUNM	Run Mode Control When written, causes entry into the selected run mode. Writes to this field are blocked if the protection level has not been enabled using the PMPROT register. **NOTE:** RUNM may be set to VLPR only when PMSTAT=RUN. After being written to VLPR, RUNM should not be written back to RUN until PMSTAT=VLPR. 00 Normal Run mode (RUN) 01 Reserved 10 Very-Low-Power Run mode (VLPR) 11 Reserved
4 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
3 STOPA	Stop Aborted When set, this read-only status bit indicates an interrupt or reset occured during the previous stop mode entry sequence, preventing the system from entering that mode. This field is cleared by hardware at the beginning of any stop mode entry sequence and is set if the sequence was aborted. 0 The previous stop mode entry was successful. 1 The previous stop mode entry was aborted.
STOPM	Stop Mode Control When written, controls entry into the selected stop mode when Sleep-Now or Sleep-On-Exit mode is entered with SLEEPDEEP=1 . Writes to this field are blocked if the protection level has not been enabled using the PMPROT register. After any system reset, this field is cleared by hardware on any successful write to the PMPROT register. **NOTE:** When set to VLLSx, the VLLSM field in the VLLSCTRL register is used to further select the particular VLLS submode which will be entered. **NOTE:** 000 Normal Stop (STOP) 001 Reserved 010 Very-Low-Power Stop (VLPS) 011 Low-Leakage Stop (LLS) 100 Very-Low-Leakage Stop (VLLSx) 101 Reserved 110 Reseved 111 Reserved

21.3.3 VLLS Control Register (SMC_VLLSCTRL)

The VLLSCTRL register provides various control bits allowing the user to fine tune power consumption during the stop mode selected by the STOPM field.

NOTE

> This register is reset on Chip POR not VLLS and by reset types that trigger Chip POR not VLLS. It is unaffected by reset types that do not trigger Chip POR not VLLS. See the Reset section details for more information.

Address: 4007_E000h base + 2h offset = 4007_E002h

Bit	7	6	5	4	3	2	1	0
Read	0		PORPO	RAM2PO	Reserved	VLLSM		
Write								
Reset	0	0	0	0	0	0	1	1

SMC_VLLSCTRL field descriptions

Field	Description
7–6 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
5 PORPO	POR Power Option This bit controls whether the POR detect circuit is enabled in VLLS0 mode. 0 POR detect circuit is enabled in VLLS0 1 POR detect circuit is disabled in VLLS0
4 RAM2PO	RAM2 Power Option This bit controls powering of RAM partition 2 in VLLS2 mode. **NOTE:** See the device's Chip Configuration details for the size and location of RAM partition 2 0 RAM2 not powered in VLLS2 1 RAM2 powered in VLLS2
3 Reserved	This field is reserved. This bit is reserved for future expansion and should always be written zero.
VLLSM	VLLS Mode Control This field controls which VLLS sub-mode to enter if STOPM = VLLSx. 000 VLLS0 001 VLLS1 010 VLLS2 011 VLLS3 100 Reserved 101 Reserved

Table continues on the next page...

SMC_VLLSCTRL field descriptions (continued)

Field	Description
	110 Reserved 111 Reserved

21.3.4 Power Mode Status register (SMC_PMSTAT)

PMSTAT is a read-only, one-hot register which indicates the current power mode of the system.

NOTE

> This register is reset on Chip POR not VLLS and by reset types that trigger Chip POR not VLLS. It is unaffected by reset types that do not trigger Chip POR not VLLS. See the Reset section details for more information.

Address: 4007_E000h base + 3h offset = 4007_E003h

Bit	7	6	5	4	3	2	1	0
Read	PMSTAT							
Write								
Reset	0	0	0	0	0	0	0	1

SMC_PMSTAT field descriptions

Field	Description
PMSTAT	Power Mode Status **NOTE:** When debug is enabled, the PMSTAT will not update to STOP or VLPS 0000_0001 Current power mode is RUN. 0000_0010 Current power mode is STOP. 0000_0100 Current power mode is VLPR. 0000_1000 Current power mode is VLPW. 0001_0000 Current power mode is VLPS. 0010_0000 Current power mode is LLS. 0100_0000 Current power mode is VLLS. 1000_0000 Reserved

21.4 Functional description

21.4.1 Power mode transitions

The following figure shows the power mode state transitions available on the chip. Any reset always brings the MCU back to the normal RUN state.

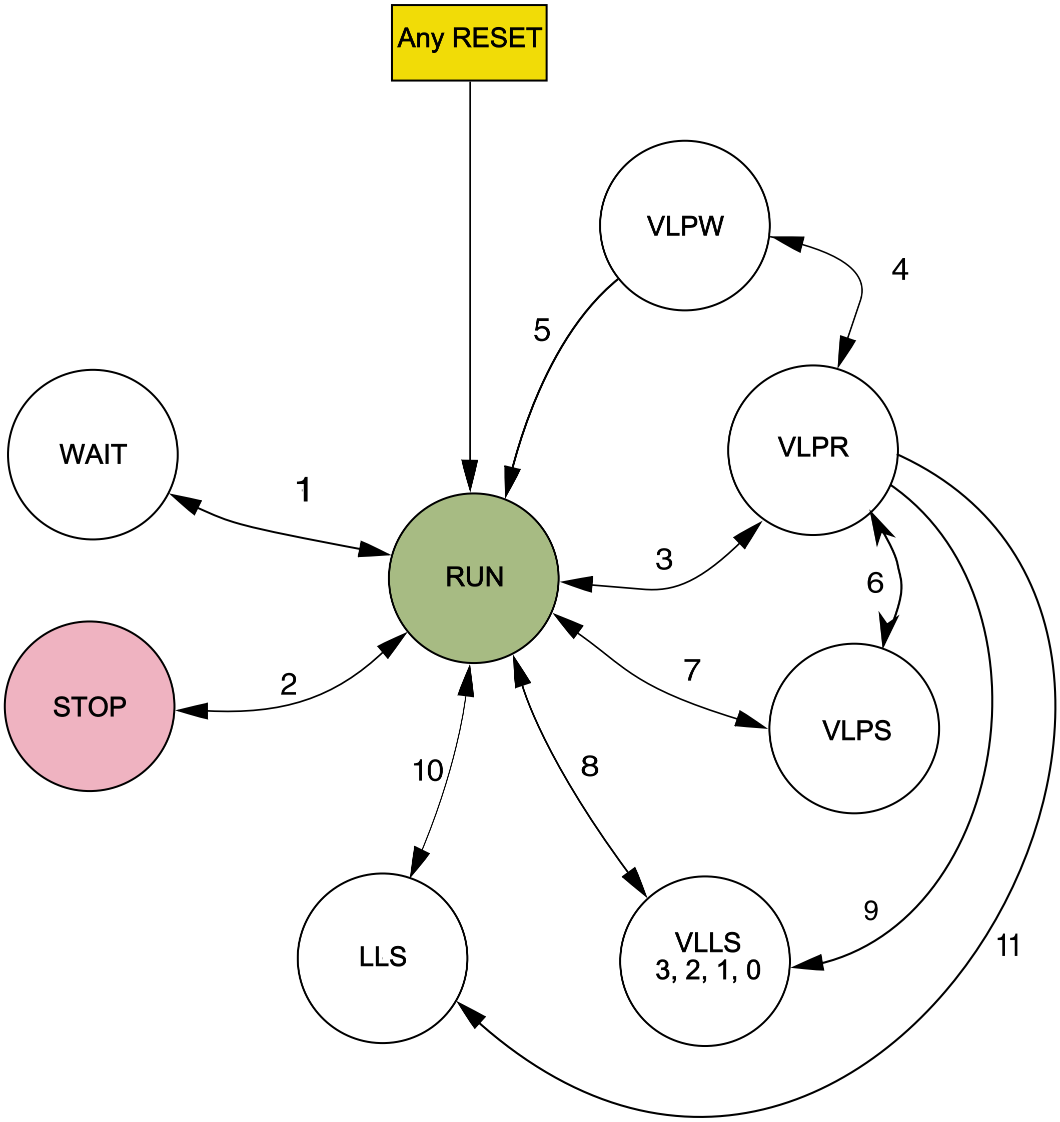


Figure 21-1. Power mode state diagram

The following table defines triggers for the various state transitions shown in the previous figure.

Table 21-2. Power mode transition triggers

Transition #	From	To	Trigger conditions
1	RUN	WAIT	Sleep-now or sleep-on-exit modes entered with SLEEPDEEP clear, controlled in System Control Register in ARM core. See note.
	WAIT	RUN	Interrupt or Reset
2	RUN	STOP	Sleep-now or sleep-on-exit modes entered with SLEEPDEEP set, which is controlled in System Control Register in ARM core. See note.[1]
	STOP	RUN	Interrupt or Reset
3	RUN	VLPR	The core, system, bus and flash clock frequencies and clocking mode are restricted in this mode. See the Power Management chapter for the maximum allowable frequencies and modes supported.
	VLPR	RUN	Set or Interrupt with PMCTRL[LPWUI] =1 or Reset.
4	VLPR	VLPW	Sleep-now or sleep-on-exit modes entered with SLEEPDEEP clear, which is controlled in System Control Register in ARM core. See note.[1]
	VLPW	VLPR	Interrupt with PMCTRL[LPWUI]=0
5	VLPW	RUN	Interrupt with PMCTRL[LPWUI]=1 or Reset
6	VLPR	VLPS	or 010, Sleep-now or sleep-on-exit modes entered with SLEEPDEEP set, which is controlled in System Control Register in ARM core. See note.[1]
	VLPS	VLPR	Interrupt with PMCTRL[LPWUI]=0 **NOTE:** If VLPS was entered directly from RUN (transition #7), hardware forces exit back to RUN and does not allow a transition to VLPR.
7	RUN	VLPS	Sleep-now or sleep-on-exit modes entered with SLEEPDEEP set, which is controlled in System Control Register in ARM core. See note.[1]
	VLPS	RUN	Interrupt with PMCTRL[LPWUI]=1 or Interrupt with PMCTRL[LPWUI]=0 and VLPS mode was entered directly from RUN or Reset
8	RUN	VLLSx	VLLSCTRL[LLSM]=x (VLLSx), Sleep-now or sleep-on-exit modes entered with SLEEPDEEP set, which is controlled in System Control Register in ARM core.

Table continues on the next page...

Table 21-2. Power mode transition triggers (continued)

Transition #	From	To	Trigger conditions
	VLLSx	RUN	Wakeup from enabled LLWU input source or RESET pin
9	VLPR	VLLSx	VLLSCTRL[LLSM]=x (VLLSx), Sleep-now or sleep-on-exit modes entered with SLEEPDEEP set, which is controlled in System Control Register in ARM core.
10	RUN	LLSx	Sleep-now or sleep-on-exit modes entered with SLEEPDEEP set, which is controlled in System Control Register in ARM core.
	LLSx	RUN	Wakeup from enabled LLWU input source or RESET pin.
11	VLPR	LLSx	Sleep-now or sleep-on-exit modes entered with SLEEPDEEP set, which is controlled in System Control Register in ARM core.

1. If debug is enabled, the core clock remains to support debug.

21.4.2 Power mode entry/exit sequencing

When entering or exiting low-power modes, the system must conform to an orderly sequence to manage transitions safely.

The SMC manages the system's entry into and exit from all power modes. This diagram illustrates the connections of the SMC with other system components in the chip that are necessary to sequence the system through all power modes.

21.4.2.1 Stop mode entry sequence

Entry into a low-power stop mode (Stop, VLPS, LLS, VLLSx) is initiated by a CPU executing the WFI instruction. After the instruction is executed, the following sequence occurs:

1. The CPU clock is gated off immediately.
2. Requests are made to all non-CPU bus masters to enter Stop mode.
3. After all masters have acknowledged they are ready to enter Stop mode, requests are made to all bus slaves to enter Stop mode.
4. After all slaves have acknowledged they are ready to enter Stop mode, all system and bus clocks are gated off.
5. Clock generators are disabled in the
6. The on-chip regulator in the PMC and internal power switches are configured to meet the power consumption goals for the targeted low-power mode.

21.4.2.2 Stop mode exit sequence

Exit from a low-power stop mode is initiated either by a reset or an interrupt event. The following sequence then executes to restore the system to a run mode (RUN or VLPR):

1. The on-chip regulator in the PMC and internal power switches are restored.
2. Clock generators are enabled in the
3. System and bus clocks are enabled to all masters and slaves.
4. The CPU clock is enabled and the CPU begins servicing the reset or interrupt that initiated the exit from the low-power stop mode.

21.4.2.3 Aborted stop mode entry

If an interrupt or a reset occurs during a stop entry sequence, the SMC can abort the transition early and return to RUN mode without completely entering the stop mode. An aborted entry is possible only if the reset or interrupt occurs before the PMC begins the transition to stop mode regulation. After this point, the interrupt or reset is ignored until the PMC has completed its transition to stop mode regulation. When an aborted stop mode entry sequence occurs, SMC_PMCTRL[STOPA] is set to 1.

21.4.2.4 Transition to wait modes

For wait modes (WAIT and VLPW), the CPU clock is gated off while all other clocking continues, as in RUN and VLPR mode operation. Some modules that support stop-in-wait functionality have their clocks disabled in these configurations.

21.4.2.5 Transition from stop modes to Debug mode

The debugger module supports a transition from STOP, WAIT, VLPS, and VLPW back to a Halted state when the debugger has been enabled. As part of this transition, system clocking is re-established and is equivalent to the normal RUN and VLPR mode clocking configuration.

21.4.3 Run modes

The run modes supported by this device can be found here.

- Run (RUN)
- Very Low-Power Run (VLPR)

21.4.3.1 RUN mode

This is the normal operating mode for the device.

This mode is selected after any reset. When the ARM processor exits reset, it sets up the stack, program counter (PC), and link register (LR):

- The processor reads the start SP (SP_main) from vector-table offset 0x000
- The processor reads the start PC from vector-table offset 0x004
- LR is set to 0xFFFF_FFFF.

To reduce power in this mode, disable the clocks to unused modules.

21.4.3.2 Very-Low Power Run (VLPR) mode

In VLPR mode, the on-chip voltage regulator is put into a stop mode regulation state. In this state, the regulator is designed to supply enough current to the MCU over a reduced frequency. To further reduce power in this mode, disable the clocks to unused modules using their corresponding clock gating control bits in the registers.

Before entering this mode, the following conditions must be met:

- All clock monitors in the must be disabled.
- The maximum frequencies of the system, bus, flash, and core are restricted. See the Power Management details about which frequencies are supported.
- Mode protection must be set to allow VLP modes, that is, PMPROT[AVLP] is 1.
- PMCTRL[RUNM] must be set to 10b to enter VLPR.
- Flash programming/erasing is not allowed.

NOTE

> Do not increase the clock frequency while in VLPR mode, because the regulator is slow in responding and cannot manage fast load transitions. In addition, do not modify the clock source in the module or any clock divider registers. Module clock enables in the can be set, but not cleared.

To reenter Normal Run mode, clear PMCTRL[RUNM]. PMSTAT is a read-only status register that can be used to determine when the system has completed an exit to RUN mode. When PMSTAT=RUN, the system is in run regulation and the MCU can run at full speed in any clock mode. If a higher execution frequency is desired, poll PMSTAT until it is set to RUN when returning from VLPR mode.

VLPR mode also provides the option to return to run regulation if any interrupt occurs. Implement this option by setting Low-Power Wakeup On Interrupt (LPWUI) in the PMCTRL register. Any reset always causes an exit from VLPR and returns the device to RUN mode after the MCU exits its reset flow. The RUNM bits are cleared by hardware on any interrupt when LPWUI is set or on any reset.

21.4.4 Wait modes

This device contains two different wait modes which are listed here.

- Wait
- Very-Low Power Wait (VLPW)

21.4.4.1 WAIT mode

WAIT mode is entered when the ARM core enters the Sleep-Now or Sleep-On-Exit modes while SLEEPDEEP is cleared. The ARM CPU enters a low-power state in which it is not clocked, but peripherals continue to be clocked provided they are enabled.

When an interrupt request occurs, the CPU exits WAIT mode and resumes processing in RUN mode, beginning with the stacking operations leading to the interrupt service routine.

A system reset causes an exit from WAIT mode, returning the device to normal RUN mode.

21.4.4.2 Very-Low-Power Wait (VLPW) mode

VLPW mode is entered by entering the Sleep-Now or Sleep-On-Exit mode while SLEEPDEEP is cleared and the device is in VLPR mode.

In VLPW, the on-chip voltage regulator remains in its stop regulation state. In this state, the regulator is designed to supply enough current to the device at a reduced frequency. To further reduce power in this mode, disable the clocks to unused modules.

VLPR mode restrictions also apply to VLPW.

VLPW mode provides the option to return to fully-regulated normal RUN mode if any enabled interrupt occurs. This is done by setting PMCTRL[LPWUI]. Wait for the PMSTAT register to set to RUN before increasing the frequency.

If the LPWUI bit is clear, when an interrupt from VLPW occurs, the device returns to VLPR mode to execute the interrupt service routine.

A system reset causes an exit from VLPW mode, returning the device to normal RUN mode.

21.4.5 Stop modes

This device contains a variety of stop modes to meet your application needs.

The stop modes range from:

- a stopped CPU, with all I/O, logic, and memory states retained, and certain asynchronous mode peripherals operating

to:

- a powered down CPU, with only I/O and a small register file retained, very few asynchronous mode peripherals operating, while the remainder of the MCU is powered down.

The choice of stop mode depends upon the user's application, and how power usage and state retention versus functional needs and recovery time may be traded off.

NOTE

> All clock monitors must be disabled before entering these low-power modes: Stop, VLPS, VLPR, VLPW, LLSand VLLSx.

The various stop modes are selected by setting the appropriate fields in PMPROT and PMCTRL. The selected stop mode is entered during the sleep-now or sleep-on-exit entry with the SLEEPDEEP bit set in the System Control Register in the ARM core.

The available stop modes are:

- Normal Stop (STOP)
- Very-Low Power Stop (VLPS)
- Low-Leakage Stop (LLS)
- Very-Low-Leakage Stop (VLLSx)

21.4.5.1 STOP mode

STOP mode is entered via the sleep-now or sleep-on-exit with the SLEEPDEEP bit set in the System Control Register in the ARM core.

The module can be configured to leave the reference clocks running.

A module capable of providing an asynchronous interrupt to the device takes the device out of STOP mode and returns the device to normal RUN mode. Refer to the device's Power Management chapter for peripheral, I/O, and memory operation in STOP mode. When an interrupt request occurs, the CPU exits STOP mode and resumes processing, beginning with the stacking operations leading to the interrupt service routine.

A system reset will cause an exit from STOP mode, returning the device to normal RUN mode via an MCU reset.

21.4.5.2 Very-Low-Power Stop (VLPS) mode

The two ways in which VLPS mode can be entered are listed here.

- Entry into stop via the sleep-now or sleep-on-exit with the SLEEPDEEP bit set in the System Control Register in the ARM core while the MCU is in VLPR mode and PMCTRL[STOPM] = 010 or 000.
- Entry into stop via the sleep-now or sleep-on-exit with the SLEEPDEEP bit set in the System Control Register in the ARM core while the MCU is in normal RUN mode and PMCTRL[STOPM] = 010. When VLPS is entered directly from RUN mode, exit to VLPR is disabled by hardware and the system will always exit back to RUN.

In VLPS, the on-chip voltage regulator remains in its stop regulation state as in VLPR.

A module capable of providing an asynchronous interrupt to the device takes the device out of VLPS and returns the device to VLPR mode, provided LPWUI is clear.

If LPWUI is set, the device returns to normal RUN mode upon an interrupt request. PMSTAT must be set to RUN before allowing the system to return to a frequency higher than that allowed in VLPR mode.

A system reset will also cause a VLPS exit, returning the device to normal RUN mode.

21.4.5.3 Low-Leakage Stop (LLS) mode

Low-Leakage Stop (LLS) mode can be entered from normal RUN or VLPR modes.

The MCU enters LLS mode if:

- In Sleep-Now or Sleep-On-Exit mode, SLEEPDEEP is set in the System Control Register in the ARM core, and
- The device is configured as shown in Table 21-2.

In LLS, the on-chip voltage regulator is in stop regulation. Most of the peripherals are put in a state-retention mode that does not allow them to operate while in LLS.

Before entering LLS mode, the user should configure the Low-Leakage Wake-up (LLWU) module to enable the desired wake-up sources. The available wake-up sources in LLS are detailed in the chip configuration details for this device.

After wakeup from LLS, the device returns to normal RUN mode with a pending LLWU module interrupt. In the LLWU interrupt service routine (ISR), the user can poll the LLWU module wake-up flags to determine the source of the wakeup.

NOTE

> The LLWU interrupt must not be masked by the interrupt controller to avoid a scenario where the system does not fully exit Stop mode on an LLS recovery.

An asserted $\overline{\text{RESET}}$ pin will cause an exit from LLS mode, returning the device to normal RUN mode. When LLS is exiting via the $\overline{\text{RESET}}$ pin, RCM_SRS[PIN] and RCM_SRS[WAKEUP] are set.

21.4.5.4 Very-Low-Leakage Stop (VLLSx) modes

This device contains these very low leakage modes:

- VLLS3
- VLLS2
- VLLS1
- VLLS0

VLLSx is often used in this document to refer to all of these modes.

All VLLSx modes can be entered from normal RUN or VLPR modes.

The MCU enters the configured VLLS mode if:

- In Sleep-Now or Sleep-On-Exit mode, the SLEEPDEEP bit is set in the System Control Register in the ARM core, and
- The device is configured as shown in Table 21-2.

In VLLS, the on-chip voltage regulator is in its stop-regulation state while most digital logic is powered off.

Before entering VLLS mode, the user should configure the Low-Leakage Wake-up (LLWU) module to enable the desired wakeup sources. The available wake-up sources in VLLS are detailed in the chip configuration details for this device.

After wakeup from VLLS, the device returns to normal RUN mode with a pending LLWU interrupt. In the LLWU interrupt service routine (ISR), the user can poll the LLWU module wake-up flags to determine the source of the wake-up.

When entering VLLS, each I/O pin is latched as configured before executing VLLS. Because all digital logic in the MCU is powered off, all port and peripheral data is lost during VLLS. This information must be restored before PMC_REGSC[ACKISO] is set.

An asserted $\overline{\text{RESET}}$ pin will cause an exit from any VLLS mode, returning the device to normal RUN mode. When exiting VLLS via the $\overline{\text{RESET}}$ pin, RCM_SRS[PIN] and RCM_SRS[WAKEUP] are set.

21.4.6 Debug in low power modes

When the MCU is secure, the device disables/limits debugger operation. When the MCU is unsecure, the ARM debugger can assert two power-up request signals:

- System power up, via SYSPWR in the Debug Port Control/Stat register
- Debug power up, via CDBGPWRUPREQ in the Debug Port Control/Stat register

When asserted while in RUN, WAIT, VLPR, or VLPW, the mode controller drives a corresponding acknowledge for each signal, that is, both CDBGPWRUPACK and CSYSPWRUPACK. When both requests are asserted, the mode controller handles attempts to enter STOP and VLPS by entering an emulated stop state. In this emulated stop state:

- the regulator is in run regulation,
- the clock source is enabled,
- all system clocks, except the core clock, are disabled,
- the debug module has access to core registers, and
- access to the on-chip peripherals is blocked.

No debug is available while the MCU is in LLS or VLLS modes. LLS is a state-retention mode and all debug operation can continue after waking from LLS, even in cases where system wakeup is due to a system reset event.

Entering into a VLLS mode causes all of the debug controls and settings to be powered off. To give time to the debugger to sync with the MCU, the MDM AP Control Register includes a Very-Low-Leakage Debug Request (VLLDBGREQ) bit that is set to configure the Reset Controller logic to hold the system in reset after the next recovery from a VLLS mode. This bit allows the debugger time to reinitialize the debug module before the debug session continues.

The MDM AP Control Register also includes a Very Low Leakage Debug Acknowledge (VLLDBGACK) bit that is set to release the ARM core being held in reset following a VLLS recovery. The debugger reinitializes all debug IP, and then asserts the VLLDBGACK control bit to allow the RCM to release the ARM core from reset and allow CPU operation to begin.

The VLLDBGACK bit is cleared by the debugger (or can be left set as is) or clears automatically due to the reset generated as part of the next VLLS recovery.

Chapter 22
MCU: Power Management Controller (PMC)

22.1 Introduction

NOTE

> For the chip-specific implementation details of this module's instances, see the chip configuration information.

The power management controller (PMC) contains the internal voltage regulator, power on reset (POR), and low voltage detect system (LVD).

See AN4503: Power Management for Kinetis MCUs for further details on using the PMC.

22.2 Features

A list of included PMC features can be found here.

- Internal voltage regulator
- Active POR providing brown-out detect
- Low-voltage detect supporting two low-voltage trip points with four warning levels per trip point

22.3 Low-voltage detect (LVD) system

This device includes a system to guard against low-voltage conditions. This protects memory contents and controls MCU system states during supply voltage variations.

The system is comprised of a power-on reset (POR) circuit and a LVD circuit with a user-selectable trip voltage: high (V_{LVDH}) or low (V_{LVDL}). The trip voltage is selected by LVDSC1[LVDV]. The LVD is disabled upon entering VLPx, LLS, and VLLSx modes.

Two flags are available to indicate the status of the low-voltage detect system:

- The Low Voltage Detect Flag in the Low Voltage Status and Control 1 Register (LVDSC1[LVDF]) operates in a level sensitive manner. LVDSC1[LVDF] is set when the supply voltage falls below the selected trip point (VLVD). LVDSC1[LVDF] is cleared by writing 1 to LVDSC1[LVDACK], but only if the internal supply has returned above the trip point; otherwise, LVDSC1[LVDF] remains set.
- The Low Voltage Warning Flag (LVWF) in the Low Voltage Status and Control 2 Register (LVDSC2[LVWF]) operates in a level sensitive manner. LVDSC2[LVWF] is set when the supply voltage falls below the selected monitor trip point (VLVW). LVDSC2[LVWF] is cleared by writing one to LVDSC2[LVWACK], but only if the internal supply has returned above the trip point; otherwise, LVDSC2[LVWF] remains set.

22.3.1 LVD reset operation

By setting LVDSC1[LVDRE], the LVD generates a reset upon detection of a low-voltage condition. The low-voltage detection threshold is determined by LVDSC1[LVDV]. After an LVD reset occurs, the LVD system holds the MCU in reset until the supply voltage rises above this threshold. The LVD field in the SRS register of the RCM module (RCM_SRS[LVD]) is set following an LVD or power-on reset.

22.3.2 LVD interrupt operation

By configuring the LVD circuit for interrupt operation (LVDSC1[LVDIE] set and LVDSC1[LVDRE] clear), LVDSC1[LVDF] is set and an LVD interrupt request occurs upon detection of a low voltage condition. LVDSC1[LVDF] is cleared by writing 1 to LVDSC1[LVDACK].

22.3.3 Low-voltage warning (LVW) interrupt operation

The LVD system contains a Low-Voltage Warning Flag (LVWF) in the Low Voltage Detect Status and Control 2 Register to indicate that the supply voltage is approaching, but is above, the LVD voltage. The LVW also has an interrupt, which is enabled by setting LVDSC2[LVWIE]. If enabled, an LVW interrupt request occurs when LVDSC2[LVWF] is set. LVDSC2[LVWF] is cleared by writing 1 to LVDSC2[LVWACK].

LVDSC2[LVWV] selects one of the four trip voltages:

- Highest: V_{LVW4}
- Two mid-levels: V_{LVW3} and V_{LVW2}
- Lowest: V_{LVW1}

22.4 I/O retention

When in LLS mode, the I/O pins are held in their input or output state.

Upon wakeup, the PMC is re-enabled, goes through a power up sequence to full regulation, and releases the logic from state retention mode. The I/O are released immediately after a wake-up or reset event. In the case of LLS exit via a RESET pin, the I/O default to their reset state.

When in VLLS modes, the I/O states are held on a wake-up event (with the exception of wake-up by reset event) until the wake-up has been acknowledged via a write to REGSC[ACKISO]. In the case of VLLS exit via a RESET pin, the I/O are released and default to their reset state. In this case, no write to REGSC[ACKISO] is needed.

22.5 Memory map and register descriptions

Details about the PMC registers can be found here.

NOTE

Different portions of PMC registers are reset only by particular reset types. Each register's description provides details. For more information about the types of reset on this chip, refer to the Reset section details.

The PMC registers can be written only in supervisor mode. Write accesses in user mode are blocked and will result in a bus error.

PMC memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4007_D000	Low Voltage Detect Status And Control 1 register (PMC_LVDSC1)	8	R/W	10h	22.5.1/482

Table continues on the next page...

PMC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4007_D001	Low Voltage Detect Status And Control 2 register (PMC_LVDSC2)	8	R/W	00h	22.5.2/483
4007_D002	Regulator Status And Control register (PMC_REGSC)	8	R/W	04h	22.5.3/484

22.5.1 Low Voltage Detect Status And Control 1 register (PMC_LVDSC1)

This register contains status and control bits to support the low voltage detect function. This register should be written during the reset initialization program to set the desired controls even if the desired settings are the same as the reset settings.

While the device is in the very low power or low leakage modes, the LVD system is disabled regardless of LVDSC1 settings. To protect systems that must have LVD always on, configure the Power Mode Protection (PMPROT) register of the SMC module (SMC_PMPROT) to disallow any very low power or low leakage modes from being enabled.

See the device's data sheet for the exact LVD trip voltages.

NOTE

> The LVDV bits are reset solely on a POR Only event. The register's other bits are reset on Chip Reset Not VLLS. For more information about these reset types, refer to the Reset section details.

Address: 4007_D000h base + 0h offset = 4007_D000h

Bit	7	6	5	4	3	2	1	0
Read	LVDF	0	LVDIE	LVDRE	0		LVDV	
Write		LVDACK						
Reset	0	0	0	1	0	0	0	0

PMC_LVDSC1 field descriptions

Field	Description
7 LVDF	Low-Voltage Detect Flag This read-only status field indicates a low-voltage detect event. 0 Low-voltage event not detected 1 Low-voltage event detected

Table continues on the next page...

PMC_LVDSC1 field descriptions (continued)

Field	Description
6 LVDACK	Low-Voltage Detect Acknowledge This write-only field is used to acknowledge low voltage detection errors. Write 1 to clear LVDF. Reads always return 0.
5 LVDIE	Low-Voltage Detect Interrupt Enable Enables hardware interrupt requests for LVDF. 0 Hardware interrupt disabled (use polling) 1 Request a hardware interrupt when LVDF = 1
4 LVDRE	Low-Voltage Detect Reset Enable This write-once bit enables LVDF events to generate a hardware reset. Additional writes are ignored. 0 LVDF does not generate hardware resets 1 Force an MCU reset when LVDF = 1
3–2 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
LVDV	Low-Voltage Detect Voltage Select Selects the LVD trip point voltage (V $_{LVD}$). 00 Low trip point selected (V $_{LVD}$ = V $_{LVDL}$) 01 High trip point selected (V $_{LVD}$ = V $_{LVDH}$) 10 Reserved 11 Reserved

22.5.2 Low Voltage Detect Status And Control 2 register (PMC_LVDSC2)

This register contains status and control bits to support the low voltage warning function.

While the device is in the very low power or low leakage modes, the LVD system is disabled regardless of LVDSC2 settings.

See the device's data sheet for the exact LVD trip voltages.

NOTE

The LVW trip voltages depend on LVWV and LVDV.

NOTE

LVWV is reset solely on a POR Only event. The other fields of the register are reset on Chip Reset Not VLLS. For more information about these reset types, refer to the Reset section details.

Address: 4007_D000h base + 1h offset = 4007_D001h

Bit	7	6	5	4	3	2	1	0
Read	LVWF	0	LVWIE		0		LVWV	
Write		LVWACK						
Reset	0	0	0	0	0	0	0	0

PMC_LVDSC2 field descriptions

Field	Description
7 LVWF	Low-Voltage Warning Flag This read-only status field indicates a low-voltage warning event. LVWF is set when V_{Supply} transitions below the trip point, or after reset and V_{Supply} is already below V_{LVW}. LVWF may be 1 after power-on reset, therefore, to use LVW interrupt function, before enabling LVWIE, LVWF must be cleared by writing LVWACK first. 0 Low-voltage warning event not detected 1 Low-voltage warning event detected
6 LVWACK	Low-Voltage Warning Acknowledge This write-only field is used to acknowledge low voltage warning errors. Write 1 to clear LVWF. Reads always return 0.
5 LVWIE	Low-Voltage Warning Interrupt Enable Enables hardware interrupt requests for LVWF. 0 Hardware interrupt disabled (use polling) 1 Request a hardware interrupt when LVWF = 1
4–2 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
LVWV	Low-Voltage Warning Voltage Select Selects the LVW trip point voltage (V_{LVW}). The actual voltage for the warning depends on LVDSC1[LVDV]. 00 Low trip point selected ($V_{LVW} = V_{LVW1}$) 01 Mid 1 trip point selected ($V_{LVW} = V_{LVW2}$) 10 Mid 2 trip point selected ($V_{LVW} = V_{LVW3}$) 11 High trip point selected ($V_{LVW} = V_{LVW4}$)

22.5.3 Regulator Status And Control register (PMC_REGSC)

The PMC contains an internal voltage regulator. The voltage regulator design uses a bandgap reference that is also available through a buffer as input to certain internal peripherals, such as the CMP and ADC. The internal regulator provides a status bit (REGONS) indicating the regulator is in run regulation.

NOTE

This register is reset on Chip Reset Not VLLS and by reset types that trigger Chip Reset not VLLS. See the Reset section details for more information.

Address: 4007_D000h base + 2h offset = 4007_D002h

<table>
<tr><td>Bit</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>Read</td><td>0</td><td>0</td><td rowspan="2">Reserved</td><td rowspan="2">BGEN</td><td>ACKISO</td><td>REGONS</td><td rowspan="2">Reserved</td><td rowspan="2">BGBE</td></tr>
<tr><td>Write</td><td></td><td></td><td>w1c</td><td></td></tr>
<tr><td>Reset</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td></tr>
</table>

PMC_REGSC field descriptions

Field	Description
7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
6 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
5 Reserved	This field is reserved.
4 BGEN	Bandgap Enable In VLPx Operation BGEN controls whether the bandgap is enabled in lower power modes of operation (VLPx, LLS, and VLLSx). When on-chip peripherals require the bandgap voltage reference in low power modes of operation, set BGEN to continue to enable the bandgap operation. **NOTE:** When the bandgap voltage reference is not needed in low power modes, clear BGEN to avoid excess power consumption. 0 Bandgap voltage reference is disabled in VLPx , LLS , and VLLSx modes. 1 Bandgap voltage reference is enabled in VLPx , LLS , and VLLSx modes.
3 ACKISO	Acknowledge Isolation Reading this field indicates whether certain peripherals and the I/O pads are in a latched state as a result of having been in a VLLS mode. Writing 1 to this field when it is set releases the I/O pads and certain peripherals to their normal run mode state. **NOTE:** After recovering from a VLLS mode, user should restore chip configuration before clearing ACKISO. In particular, pin configuration for enabled LLWU wakeup pins should be restored to avoid any LLWU flag from being falsely set when ACKISO is cleared. 0 Peripherals and I/O pads are in normal run state. 1 Certain peripherals and I/O pads are in an isolated and latched state.
2 REGONS	Regulator In Run Regulation Status This read-only field provides the current status of the internal voltage regulator. 0 Regulator is in stop regulation or in transition to/from it 1 Regulator is in run regulation
1 Reserved	This field is reserved. **NOTE:** This reserved bit must remain cleared (set to 0).

Table continues on the next page...

PMC_REGSC field descriptions (continued)

Field	Description
0 BGBE	Bandgap Buffer Enable Enables the bandgap buffer. 0 Bandgap buffer not enabled 1 Bandgap buffer enabled

Chapter 23
MCU: Low-Leakage Wakeup Unit (LLWU)

23.1 Introduction

NOTE

> For the chip-specific implementation details of this module's instances, see the chip configuration information.

The LLWU module allows the user to select up to 16 external pins and up to 8 internal modules as interrupt wake-up sources from low-leakage power modes.

The input sources are described in the device's chip configuration details. Each of the available wake-up sources can be individually enabled.

The RESET pin is an additional source for triggering an exit from low-leakage power modes, and causes the MCU to exit both LLS and VLLS through a reset flow.

The LLWU_RST[LLRSTE] bit must be set to allow an exit from low-leakage modes via the RESET pin. On a device where the RESET pin is shared with other functions, the explicit port mux control register must be set for the RESET pin before the RESET pin can be used as a low-leakage reset source.

The LLWU module also includes optional digital pin filters: two for the external wakeup pins and one for the RESET pin.

23.1.1 Features

The LLWU module features include:

- Support for up to 16 external input pins and up to 8 internal modules with individual enable bits for MCU interrupt from low leakage modes
- Input sources may be external pins or from internal peripherals capable of running in LLS or VLLS. See the chip configuration information for wakeup input sources for this device.

- External pin wake-up inputs, each of which is programmable as falling-edge, rising-edge, or any change
- Wake-up inputs that are activated after MCU enters a low-leakage power mode
- Optional digital filters provided to qualify an external pin detect and $\overline{\text{RESET}}$ pin detect. Note that when the LPO clock is disabled, the filters are disabled and bypassed.

23.1.2 Modes of operation

The LLWU module becomes functional on entry into a low-leakage power mode. After recovery from LLS, the LLWU is immediately disabled. After recovery from VLLS, the LLWU continues to detect wake-up events until the user has acknowledged the wake-up via a write to PMC_REGSC[ACKISO].

23.1.2.1 LLS mode

Wake-up events due to external pin inputs (LLWU_Px) and internal module interrupt inputs (LLWU_MxIF) result in an interrupt flow when exiting LLS. A reset event due to $\overline{\text{RESET}}$ pin assertion results in a reset flow when exiting LLS.

NOTE

The LLWU interrupt must not be masked by the interrupt controller to avoid a scenario where the system does not fully exit Stop mode on an LLS recovery.

23.1.2.2 VLLS modes

All wakeup and reset events result in VLLS exit via a reset flow.

23.1.2.3 Non-low leakage modes

The LLWU is not active in all non-low leakage modes where detection and control logic are in a static state. The LLWU registers are accessible in non-low leakage modes and are available for configuring and reading status when bus transactions are possible.

When the $\overline{\text{RESET}}$ pin filter or wake-up pin filters are enabled, filter operation begins immediately. If a low leakage mode is entered within five LPO clock cycles of an active edge, the edge event will be detected by the LLWU. For $\overline{\text{RESET}}$ pin filtering, this means that there is no restart to the minimum LPO cycle duration as the filtering transitions from a non-low leakage filter, which is implemented in the RCM, to the LLWU filter.

23.1.2.4 Debug mode

When the chip is in Debug mode and then enters LLS or a VLLSx mode, no debug logic works in the fully-functional low-leakage mode. Upon an exit from the LLS or VLLSx mode, the LLWU becomes inactive.

23.1.3 Block diagram

The following figure is the block diagram for the LLWU module.

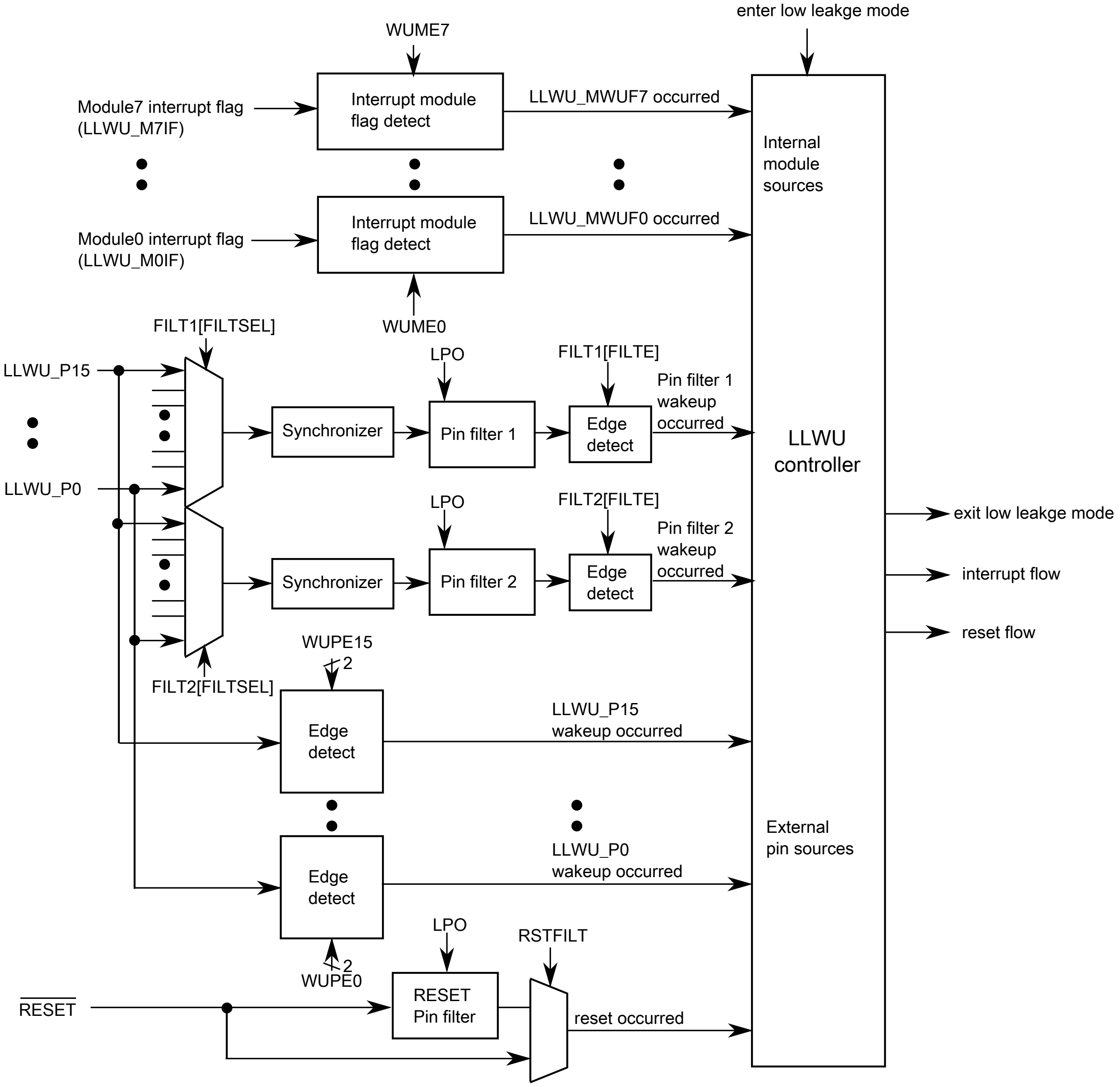


Figure 23-1. LLWU block diagram

23.2 LLWU signal descriptions

The signal properties of LLWU are shown in the table found here.

The external wakeup input pins can be enabled to detect either rising-edge, falling-edge, or on any change.

Table 23-1. LLWU signal descriptions

Signal	Description	I/O
LLWU_Pn	Wakeup inputs (n = 0-15)	I

23.3 Memory map/register definition

The LLWU includes the following registers:

- Wake-up source enable registers
  - Enable external pin input sources
  - Enable internal peripheral interrupt sources
- Wake-up flag registers
  - Indication of wakeup source that caused exit from a low-leakage power mode includes external pin or internal module interrupt
- Wake-up pin filter enable registers
- $\overline{\text{RESET}}$ pin filter enable register

NOTE

The LLWU registers can be written only in supervisor mode. Write accesses in user mode are blocked and will result in a bus error.

All LLWU registers are reset by Chip Reset not VLLS and by reset types that trigger Chip Reset not VLLS. Each register's displayed reset value represents this subset of reset types. LLWU registers are unaffected by reset types that do not trigger Chip Reset not VLLS. For more information about the types of reset on this chip, refer to the Introduction details.

LLWU memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4007_C000	LLWU Pin Enable 1 register (LLWU_PE1)	8	R/W	00h	23.3.1/492
4007_C001	LLWU Pin Enable 2 register (LLWU_PE2)	8	R/W	00h	23.3.2/493
4007_C002	LLWU Pin Enable 3 register (LLWU_PE3)	8	R/W	00h	23.3.3/494
4007_C003	LLWU Pin Enable 4 register (LLWU_PE4)	8	R/W	00h	23.3.4/495
4007_C004	LLWU Module Enable register (LLWU_ME)	8	R/W	00h	23.3.5/496
4007_C005	LLWU Flag 1 register (LLWU_F1)	8	R/W	00h	23.3.6/498
4007_C006	LLWU Flag 2 register (LLWU_F2)	8	R/W	00h	23.3.7/499
4007_C007	LLWU Flag 3 register (LLWU_F3)	8	R	00h	23.3.8/501
4007_C008	LLWU Pin Filter 1 register (LLWU_FILT1)	8	R/W	00h	23.3.9/503
4007_C009	LLWU Pin Filter 2 register (LLWU_FILT2)	8	R/W	00h	23.3.10/504
4007_C00A	LLWU Reset Enable register (LLWU_RST)	8	R/W	02h	23.3.11/505

23.3.1 LLWU Pin Enable 1 register (LLWU_PE1)

LLWU_PE1 contains the field to enable and select the edge detect type for the external wakeup input pins LLWU_P3–LLWU_P0.

NOTE

This register is reset on Chip Reset not VLLS and by reset types that trigger Chip Reset not VLLS. It is unaffected by reset types that do not trigger Chip Reset not VLLS. See the Introduction details for more information.

Address: 4007_C000h base + 0h offset = 4007_C000h

Bit	7	6	5	4	3	2	1	0
Read Write	WUPE3		WUPE2		WUPE1		WUPE0	
Reset	0	0	0	0	0	0	0	0

LLWU_PE1 field descriptions

Field	Description
7–6 WUPE3	Wakeup Pin Enable For LLWU_P3 Enables and configures the edge detection for the wakeup pin. 00 External input pin disabled as wakeup input 01 External input pin enabled with rising edge detection 10 External input pin enabled with falling edge detection 11 External input pin enabled with any change detection
5–4 WUPE2	Wakeup Pin Enable For LLWU_P2 Enables and configures the edge detection for the wakeup pin. 00 External input pin disabled as wakeup input 01 External input pin enabled with rising edge detection 10 External input pin enabled with falling edge detection 11 External input pin enabled with any change detection
3–2 WUPE1	Wakeup Pin Enable For LLWU_P1 Enables and configures the edge detection for the wakeup pin. 00 External input pin disabled as wakeup input 01 External input pin enabled with rising edge detection 10 External input pin enabled with falling edge detection 11 External input pin enabled with any change detection
WUPE0	Wakeup Pin Enable For LLWU_P0 Enables and configures the edge detection for the wakeup pin. 00 External input pin disabled as wakeup input 01 External input pin enabled with rising edge detection

Table continues on the next page...

LLWU_PE1 field descriptions (continued)

Field	Description
	10 External input pin enabled with falling edge detection 11 External input pin enabled with any change detection

23.3.2 LLWU Pin Enable 2 register (LLWU_PE2)

LLWU_PE2 contains the field to enable and select the edge detect type for the external wakeup input pins LLWU_P7–LLWU_P4.

NOTE

This register is reset on Chip Reset not VLLS and by reset types that trigger Chip Reset not VLLS. It is unaffected by reset types that do not trigger Chip Reset not VLLS. See the Introduction details for more information.

Address: 4007_C000h base + 1h offset = 4007_C001h

Bit	7	6	5	4	3	2	1	0
Read Write	WUPE7		WUPE6		WUPE5		WUPE4	
Reset	0	0	0	0	0	0	0	0

LLWU_PE2 field descriptions

Field	Description
7–6 WUPE7	Wakeup Pin Enable For LLWU_P7 Enables and configures the edge detection for the wakeup pin. 00 External input pin disabled as wakeup input 01 External input pin enabled with rising edge detection 10 External input pin enabled with falling edge detection 11 External input pin enabled with any change detection
5–4 WUPE6	Wakeup Pin Enable For LLWU_P6 Enables and configures the edge detection for the wakeup pin. 00 External input pin disabled as wakeup input 01 External input pin enabled with rising edge detection 10 External input pin enabled with falling edge detection 11 External input pin enabled with any change detection
3–2 WUPE5	Wakeup Pin Enable For LLWU_P5 Enables and configures the edge detection for the wakeup pin. 00 External input pin disabled as wakeup input 01 External input pin enabled with rising edge detection

Table continues on the next page...

LLWU_PE2 field descriptions (continued)

Field	Description
	10 External input pin enabled with falling edge detection 11 External input pin enabled with any change detection
WUPE4	Wakeup Pin Enable For LLWU_P4 Enables and configures the edge detection for the wakeup pin. 00 External input pin disabled as wakeup input 01 External input pin enabled with rising edge detection 10 External input pin enabled with falling edge detection 11 External input pin enabled with any change detection

23.3.3 LLWU Pin Enable 3 register (LLWU_PE3)

LLWU_PE3 contains the field to enable and select the edge detect type for the external wakeup input pins LLWU_P11–LLWU_P8.

NOTE

> This register is reset on Chip Reset not VLLS and by reset types that trigger Chip Reset not VLLS. It is unaffected by reset types that do not trigger Chip Reset not VLLS. See the Introduction details for more information.

Address: 4007_C000h base + 2h offset = 4007_C002h

Bit	7	6	5	4	3	2	1	0
Read Write	WUPE11		WUPE10		WUPE9		WUPE8	
Reset	0	0	0	0	0	0	0	0

LLWU_PE3 field descriptions

Field	Description
7–6 WUPE11	Wakeup Pin Enable For LLWU_P11 Enables and configures the edge detection for the wakeup pin. 00 External input pin disabled as wakeup input 01 External input pin enabled with rising edge detection 10 External input pin enabled with falling edge detection 11 External input pin enabled with any change detection
5–4 WUPE10	Wakeup Pin Enable For LLWU_P10 Enables and configures the edge detection for the wakeup pin. 00 External input pin disabled as wakeup input 01 External input pin enabled with rising edge detection

Table continues on the next page...

LLWU_PE3 field descriptions (continued)

Field	Description
	10 External input pin enabled with falling edge detection 11 External input pin enabled with any change detection
3–2 WUPE9	Wakeup Pin Enable For LLWU_P9 Enables and configures the edge detection for the wakeup pin. 00 External input pin disabled as wakeup input 01 External input pin enabled with rising edge detection 10 External input pin enabled with falling edge detection 11 External input pin enabled with any change detection
WUPE8	Wakeup Pin Enable For LLWU_P8 Enables and configures the edge detection for the wakeup pin. 00 External input pin disabled as wakeup input 01 External input pin enabled with rising edge detection 10 External input pin enabled with falling edge detection 11 External input pin enabled with any change detection

23.3.4 LLWU Pin Enable 4 register (LLWU_PE4)

LLWU_PE4 contains the field to enable and select the edge detect type for the external wakeup input pins LLWU_P15–LLWU_P12.

NOTE

This register is reset on Chip Reset not VLLS and by reset types that trigger Chip Reset not VLLS. It is unaffected by reset types that do not trigger Chip Reset not VLLS. See the Introduction details for more information.

Address: 4007_C000h base + 3h offset = 4007_C003h

Bit	7	6	5	4	3	2	1	0
Read Write	WUPE15		WUPE14		WUPE13		WUPE12	
Reset	0	0	0	0	0	0	0	0

LLWU_PE4 field descriptions

Field	Description
7–6 WUPE15	Wakeup Pin Enable For LLWU_P15 Enables and configures the edge detection for the wakeup pin. 00 External input pin disabled as wakeup input 01 External input pin enabled with rising edge detection

Table continues on the next page...

LLWU_PE4 field descriptions (continued)

Field	Description
	10 External input pin enabled with falling edge detection 11 External input pin enabled with any change detection
5–4 WUPE14	Wakeup Pin Enable For LLWU_P14 Enables and configures the edge detection for the wakeup pin. 00 External input pin disabled as wakeup input 01 External input pin enabled with rising edge detection 10 External input pin enabled with falling edge detection 11 External input pin enabled with any change detection
3–2 WUPE13	Wakeup Pin Enable For LLWU_P13 Enables and configures the edge detection for the wakeup pin. 00 External input pin disabled as wakeup input 01 External input pin enabled with rising edge detection 10 External input pin enabled with falling edge detection 11 External input pin enabled with any change detection
WUPE12	Wakeup Pin Enable For LLWU_P12 Enables and configures the edge detection for the wakeup pin. 00 External input pin disabled as wakeup input 01 External input pin enabled with rising edge detection 10 External input pin enabled with falling edge detection 11 External input pin enabled with any change detection

23.3.5 LLWU Module Enable register (LLWU_ME)

LLWU_ME contains the bits to enable the internal module flag as a wakeup input source for inputs MWUF7–MWUF0.

NOTE

This register is reset on Chip Reset not VLLS and by reset types that trigger Chip Reset not VLLS. It is unaffected by reset types that do not trigger Chip Reset not VLLS. See the Introduction details for more information.

Address: 4007_C000h base + 4h offset = 4007_C004h

Bit	7	6	5	4	3	2	1	0
Read Write	WUME7	WUME6	WUME5	WUME4	WUME3	WUME2	WUME1	WUME0
Reset	0	0	0	0	0	0	0	0

LLWU_ME field descriptions

Field	Description
7 WUME7	Wakeup Module Enable For Module 7 Enables an internal module as a wakeup source input. 0 Internal module flag not used as wakeup source 1 Internal module flag used as wakeup source
6 WUME6	Wakeup Module Enable For Module 6 Enables an internal module as a wakeup source input. 0 Internal module flag not used as wakeup source 1 Internal module flag used as wakeup source
5 WUME5	Wakeup Module Enable For Module 5 Enables an internal module as a wakeup source input. 0 Internal module flag not used as wakeup source 1 Internal module flag used as wakeup source
4 WUME4	Wakeup Module Enable For Module 4 Enables an internal module as a wakeup source input. 0 Internal module flag not used as wakeup source 1 Internal module flag used as wakeup source
3 WUME3	Wakeup Module Enable For Module 3 Enables an internal module as a wakeup source input. 0 Internal module flag not used as wakeup source 1 Internal module flag used as wakeup source
2 WUME2	Wakeup Module Enable For Module 2 Enables an internal module as a wakeup source input. 0 Internal module flag not used as wakeup source 1 Internal module flag used as wakeup source
1 WUME1	Wakeup Module Enable for Module 1 Enables an internal module as a wakeup source input. 0 Internal module flag not used as wakeup source 1 Internal module flag used as wakeup source
0 WUME0	Wakeup Module Enable For Module 0 Enables an internal module as a wakeup source input. 0 Internal module flag not used as wakeup source 1 Internal module flag used as wakeup source

23.3.6 LLWU Flag 1 register (LLWU_F1)

LLWU_F1 contains the wakeup flags indicating which wakeup source caused the MCU to exit LLS or VLLS mode. For LLS, this is the source causing the CPU interrupt flow. For VLLS, this is the source causing the MCU reset flow.

The external wakeup flags are read-only and clearing a flag is accomplished by a write of a 1 to the corresponding WUFx bit. The wakeup flag (WUFx), if set, will remain set if the associated WUPEx bit is cleared.

NOTE

> This register is reset on Chip Reset not VLLS and by reset types that trigger Chip Reset not VLLS. It is unaffected by reset types that do not trigger Chip Reset not VLLS. See the Introduction details for more information.

Address: 4007_C000h base + 5h offset = 4007_C005h

Bit	7	6	5	4	3	2	1	0
Read	WUF7	WUF6	WUF5	WUF4	WUF3	WUF2	WUF1	WUF0
Write	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c
Reset	0	0	0	0	0	0	0	0

LLWU_F1 field descriptions

Field	Description
7 WUF7	Wakeup Flag For LLWU_P7 Indicates that an enabled external wakeup pin was a source of exiting a low-leakage power mode. To clear the flag, write a 1 to WUF7. 0 LLWU_P7 input was not a wakeup source 1 LLWU_P7 input was a wakeup source
6 WUF6	Wakeup Flag For LLWU_P6 Indicates that an enabled external wakeup pin was a source of exiting a low-leakage power mode. To clear the flag, write a 1 to WUF6. 0 LLWU_P6 input was not a wakeup source 1 LLWU_P6 input was a wakeup source
5 WUF5	Wakeup Flag For LLWU_P5 Indicates that an enabled external wakeup pin was a source of exiting a low-leakage power mode. To clear the flag, write a 1 to WUF5. 0 LLWU_P5 input was not a wakeup source 1 LLWU_P5 input was a wakeup source

Table continues on the next page...

LLWU_F1 field descriptions (continued)

Field	Description
4 WUF4	Wakeup Flag For LLWU_P4 Indicates that an enabled external wake-up pin was a source of exiting a low-leakage power mode. To clear the flag, write a 1 to WUF4. 0 LLWU_P4 input was not a wakeup source 1 LLWU_P4 input was a wakeup source
3 WUF3	Wakeup Flag For LLWU_P3 Indicates that an enabled external wakeup pin was a source of exiting a low-leakage power mode. To clear the flag, write a 1 to WUF3. 0 LLWU_P3 input was not a wake-up source 1 LLWU_P3 input was a wake-up source
2 WUF2	Wakeup Flag For LLWU_P2 Indicates that an enabled external wakeup pin was a source of exiting a low-leakage power mode. To clear the flag, write a 1 to WUF2. 0 LLWU_P2 input was not a wakeup source 1 LLWU_P2 input was a wakeup source
1 WUF1	Wakeup Flag For LLWU_P1 Indicates that an enabled external wakeup pin was a source of exiting a low-leakage power mode. To clear the flag, write a 1 to WUF1. 0 LLWU_P1 input was not a wakeup source 1 LLWU_P1 input was a wakeup source
0 WUF0	Wakeup Flag For LLWU_P0 Indicates that an enabled external wake-up pin was a source of exiting a low-leakage power mode. To clear the flag, write a 1 to WUF0. 0 LLWU_P0 input was not a wakeup source 1 LLWU_P0 input was a wakeup source

23.3.7 LLWU Flag 2 register (LLWU_F2)

LLWU_F2 contains the wakeup flags indicating which wakeup source caused the MCU to exit LLS or VLLS mode. For LLS, this is the source causing the CPU interrupt flow. For VLLS, this is the source causing the MCU reset flow.

The external wakeup flags are read-only and clearing a flag is accomplished by a write of a 1 to the corresponding WUFx bit. The wakeup flag (WUFx), if set, will remain set if the associated WUPEx bit is cleared.

NOTE

This register is reset on Chip Reset not VLLS and by reset types that trigger Chip Reset not VLLS. It is unaffected by reset types that do not trigger Chip Reset not VLLS. See the Introduction details for more information.

Address: 4007_C000h base + 6h offset = 4007_C006h

Bit	7	6	5	4	3	2	1	0
Read	WUF15	WUF14	WUF13	WUF12	WUF11	WUF10	WUF9	WUF8
Write	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c
Reset	0	0	0	0	0	0	0	0

LLWU_F2 field descriptions

Field	Description
7 WUF15	Wakeup Flag For LLWU_P15 Indicates that an enabled external wakeup pin was a source of exiting a low-leakage power mode. To clear the flag, write a 1 to WUF15. 0 LLWU_P15 input was not a wakeup source 1 LLWU_P15 input was a wakeup source
6 WUF14	Wakeup Flag For LLWU_P14 Indicates that an enabled external wakeup pin was a source of exiting a low-leakage power mode. To clear the flag, write a 1 to WUF14. 0 LLWU_P14 input was not a wakeup source 1 LLWU_P14 input was a wakeup source
5 WUF13	Wakeup Flag For LLWU_P13 Indicates that an enabled external wakeup pin was a source of exiting a low-leakage power mode. To clear the flag, write a 1 to WUF13. 0 LLWU_P13 input was not a wakeup source 1 LLWU_P13 input was a wakeup source
4 WUF12	Wakeup Flag For LLWU_P12 Indicates that an enabled external wakeup pin was a source of exiting a low-leakage power mode. To clear the flag, write a 1 to WUF12. 0 LLWU_P12 input was not a wakeup source 1 LLWU_P12 input was a wakeup source
3 WUF11	Wakeup Flag For LLWU_P11 Indicates that an enabled external wakeup pin was a source of exiting a low-leakage power mode. To clear the flag, write a 1 to WUF11. 0 LLWU_P11 input was not a wakeup source 1 LLWU_P11 input was a wakeup source
2 WUF10	Wakeup Flag For LLWU_P10

Table continues on the next page...

LLWU_F2 field descriptions (continued)

Field	Description
	Indicates that an enabled external wakeup pin was a source of exiting a low-leakage power mode. To clear the flag, write a 1 to WUF10. 0 LLWU_P10 input was not a wakeup source 1 LLWU_P10 input was a wakeup source
1 WUF9	Wakeup Flag For LLWU_P9 Indicates that an enabled external wakeup pin was a source of exiting a low-leakage power mode. To clear the flag, write a 1 to WUF9. 0 LLWU_P9 input was not a wakeup source 1 LLWU_P9 input was a wakeup source
0 WUF8	Wakeup Flag For LLWU_P8 Indicates that an enabled external wakeup pin was a source of exiting a low-leakage power mode. To clear the flag, write a 1 to WUF8. 0 LLWU_P8 input was not a wakeup source 1 LLWU_P8 input was a wakeup source

23.3.8 LLWU Flag 3 register (LLWU_F3)

LLWU_F3 contains the wakeup flags indicating which internal wakeup source caused the MCU to exit LLS or VLLS mode. For LLS, this is the source causing the CPU interrupt flow. For VLLS, this is the source causing the MCU reset flow.

For internal peripherals that are capable of running in a low-leakage power mode, such as a real time clock module or CMP module, the flag from the associated peripheral is accessible as the MWUFx bit. The flag will need to be cleared in the peripheral instead of writing a 1 to the MWUFx bit.

NOTE

This register is reset on Chip Reset not VLLS and by reset types that trigger Chip Reset not VLLS. It is unaffected by reset types that do not trigger Chip Reset not VLLS. See the Introduction details for more information.

Address: 4007_C000h base + 7h offset = 4007_C007h

Bit	7	6	5	4	3	2	1	0
Read	MWUF7	MWUF6	MWUF5	MWUF4	MWUF3	MWUF2	MWUF1	MWUF0
Write								
Reset	0	0	0	0	0	0	0	0

LLWU_F3 field descriptions

Field	Description
7 MWUF7	Wakeup flag For module 7 Indicates that an enabled internal peripheral was a source of exiting a low-leakage power mode. To clear the flag, follow the internal peripheral flag clearing mechanism. 0 Module 7 input was not a wakeup source 1 Module 7 input was a wakeup source
6 MWUF6	Wakeup flag For module 6 Indicates that an enabled internal peripheral was a source of exiting a low-leakage power mode. To clear the flag, follow the internal peripheral flag clearing mechanism. 0 Module 6 input was not a wakeup source 1 Module 6 input was a wakeup source
5 MWUF5	Wakeup flag For module 5 Indicates that an enabled internal peripheral was a source of exiting a low-leakage power mode. To clear the flag, follow the internal peripheral flag clearing mechanism. 0 Module 5 input was not a wakeup source 1 Module 5 input was a wakeup source
4 MWUF4	Wakeup flag For module 4 Indicates that an enabled internal peripheral was a source of exiting a low-leakage power mode. To clear the flag, follow the internal peripheral flag clearing mechanism. 0 Module 4 input was not a wakeup source 1 Module 4 input was a wakeup source
3 MWUF3	Wakeup flag For module 3 Indicates that an enabled internal peripheral was a source of exiting a low-leakage power mode. To clear the flag, follow the internal peripheral flag clearing mechanism. 0 Module 3 input was not a wakeup source 1 Module 3 input was a wakeup source
2 MWUF2	Wakeup flag For module 2 Indicates that an enabled internal peripheral was a source of exiting a low-leakage power mode. To clear the flag, follow the internal peripheral flag clearing mechanism. 0 Module 2 input was not a wakeup source 1 Module 2 input was a wakeup source
1 MWUF1	Wakeup flag For module 1 Indicates that an enabled internal peripheral was a source of exiting a low-leakage power mode. To clear the flag, follow the internal peripheral flag clearing mechanism. 0 Module 1 input was not a wakeup source 1 Module 1 input was a wakeup source
0 MWUF0	Wakeup flag For module 0

Table continues on the next page...

LLWU_F3 field descriptions (continued)

Field	Description
	Indicates that an enabled internal peripheral was a source of exiting a low-leakage power mode. To clear the flag, follow the internal peripheral flag clearing mechanism. 0 Module 0 input was not a wakeup source 1 Module 0 input was a wakeup source

23.3.9 LLWU Pin Filter 1 register (LLWU_FILT1)

LLWU_FILT1 is a control and status register that is used to enable/disable the digital filter 1 features for an external pin.

NOTE

> This register is reset on Chip Reset not VLLS and by reset types that trigger Chip Reset not VLLS. It is unaffected by reset types that do not trigger Chip Reset not VLLS. See the Introduction details for more information.

Address: 4007_C000h base + 8h offset = 4007_C008h

Bit	7	6	5	4	3	2	1	0
Read	FILTF	FILTE		0	FILTSEL			
Write	w1c							
Reset	0	0	0	0	0	0	0	0

LLWU_FILT1 field descriptions

Field	Description
7 FILTF	Filter Detect Flag Indicates that the filtered external wakeup pin, selected by FILTSEL, was a source of exiting a low-leakage power mode. To clear the flag write a one to FILTF. 0 Pin Filter 1 was not a wakeup source 1 Pin Filter 1 was a wakeup source
6–5 FILTE	Digital Filter On External Pin Controls the digital filter options for the external pin detect. 00 Filter disabled 01 Filter posedge detect enabled 10 Filter negedge detect enabled 11 Filter any edge detect enabled
4 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
FILTSEL	Filter Pin Select

Table continues on the next page...

LLWU_FILT1 field descriptions (continued)

Field	Description
	Selects 1 out of the 16 wakeup pins to be muxed into the filter. 0000 Select LLWU_P0 for filter ... ... 1111 Select LLWU_P15 for filter

23.3.10 LLWU Pin Filter 2 register (LLWU_FILT2)

LLWU_FILT2 is a control and status register that is used to enable/disable the digital filter 2 features for an external pin.

NOTE

> This register is reset on Chip Reset not VLLS and by reset types that trigger Chip Reset not VLLS. It is unaffected by reset types that do not trigger Chip Reset not VLLS. See the Introduction details for more information.

Address: 4007_C000h base + 9h offset = 4007_C009h

Bit	7	6	5	4	3	2	1	0
Read	FILTF	FILTE		0	FILTSEL			
Write	w1c							
Reset	0	0	0	0	0	0	0	0

LLWU_FILT2 field descriptions

Field	Description
7 FILTF	Filter Detect Flag Indicates that the filtered external wakeup pin, selected by FILTSEL, was a source of exiting a low-leakage power mode. To clear the flag write a one to FILTF. 0 Pin Filter 2 was not a wakeup source 1 Pin Filter 2 was a wakeup source
6–5 FILTE	Digital Filter On External Pin Controls the digital filter options for the external pin detect. 00 Filter disabled 01 Filter posedge detect enabled 10 Filter negedge detect enabled 11 Filter any edge detect enabled
4 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
FILTSEL	Filter Pin Select

Table continues on the next page...

LLWU_FILT2 field descriptions (continued)

Field	Description
	Selects 1 out of the 16 wakeup pins to be muxed into the filter. 0000 Select LLWU_P0 for filter ... ... 1111 Select LLWU_P15 for filter

23.3.11 LLWU Reset Enable register (LLWU_RST)

LLWU_RST is a control register that is used to enable/disable the digital filter for the external pin detect and RESET pin.

NOTE

This register is reset on Chip Reset not VLLS and by reset types that trigger Chip Reset not VLLS. It is unaffected by reset types that do not trigger Chip Reset not VLLS. See the Introduction details for more information.

Address: 4007_C000h base + Ah offset = 4007_C00Ah

Bit	7	6	5	4	3	2	1	0
Read	0						LLRSTE	RSTFILT
Write								
Reset	0	0	0	0	0	0	1	0

LLWU_RST field descriptions

Field	Description
7–2 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
1 LLRSTE	Low-Leakage Mode RESET Enable This bit must be set to allow the device to be reset while in a low-leakage power mode. On devices where Reset is not a dedicated pin, the RESET pin must also be enabled in the explicit port mux control. 0 RESET pin not enabled as a leakage mode exit source 1 RESET pin enabled as a low leakage mode exit source
0 RSTFILT	Digital Filter On RESET Pin Enables the digital filter for the RESET pin during LLS, VLLS3, VLLS2, or VLLS1 modes. 0 Filter not enabled 1 Filter enabled

23.4 Functional description

Thie low-leakage wakeup unit (LLWU) module allows internal peripherals and external input pins as a source of wakeup from low-leakage modes.

It is operational only in LLS and VLLSx modes.

The LLWU module contains pin enables for each external pin and internal module. For each external pin, the user can disable or select the edge type for the wakeup with the following options:

- Falling-edge
- Rising-edge
- Either-edge

When an external pin is enabled as a wakeup source, the pin must be configured as an input pin.

The LLWU implements optional 3-cycle glitch filters, based on the LPO clock. A detected external pin, either wakeup or $\overline{\text{RESET}}$, is required to remain asserted until the enabled glitch filter times out. Additional latency of up to 2 cycles is due to synchronization, which results in a total of up to 5 cycles of delay before the detect circuit alerts the system to the wakeup or reset event when the filter function is enabled. Two wakeup detect filters are available for selected external pins. A separate reset filter is on the $\overline{\text{RESET}}$ pin. Glitch filtering is not provided on the internal modules.

For internal module interrupts, the WUMEx bit enables the associated module interrupt as a wakeup source.

23.4.1 LLS mode

Wakeup events triggered from either an external pin input or an internal module interrupt, result in a CPU interrupt flow to begin user code execution.

An LLS reset event due to RESET pin assertion causes an exit via a system reset. State retention data is lost, and the I/O states return to their reset state. The RCM_SRS[WAKEUP] and RCM_SRS[PIN] bits are set and the system executes a reset flow before CPU operation begins with a reset vector fetch.

23.4.2 VLLS modes

For any wakeup from VLLS, recovery is always via a reset flow and RCM_SRS[WAKEUP] is set indicating the low-leakage mode was active. State retention data is lost and I/O will be restored after PMC_REGSC[ACKISO] has been written.

A VLLS exit event due to $\overline{\text{RESET}}$ pin assertion causes an exit via a system reset. State retention data is lost and the I/O states immediately return to their reset state. The RCM_SRS[WAKEUP] and RCM_SRS[PIN] bits are set and the system executes a reset flow before CPU operation begins with a reset vector fetch.

23.4.3 Initialization

For an enabled peripheral wakeup input, the peripheral flag must be cleared by software before entering LLS or VLLSx mode to avoid an immediate exit from the mode.

Flags associated with external input pins, filtered and unfiltered, must also be cleared by software prior to entry to LLS or VLLSx mode.

After enabling an external pin filter or changing the source pin, wait at least five LPO clock cycles before entering LLS or VLLSx mode to allow the filter to initialize.

NOTE

After recovering from a VLLS mode, user must restore chip configuration before clearing PMC_REGSC[ACKISO]. In particular, pin configuration for enabled LLWU wake-up pins must be restored to avoid any LLWU flag from being falsely set when PMC_REGSC[ACKISO] is cleared.

The signal selected as a wake-up source pin must be a digital pin, as selected in the pin mux control.

Chapter 24
MCU: Miscellaneous Control Module (MCM)

24.1 Introduction

The Miscellaneous Control Module (MCM) provides miscellaneous control functions and contains Cortex-M7 local memory descriptors.

24.1.1 Features

The MCM includes the following features:

- Program-visible information on the platform configuration and revision
- Floating Point Exception monitor and interrupt control
- Local memory descriptors (ITCM, D0TCM, D1TCM, ICACHE, and DCACHE)

24.2 Memory map/register descriptions

The memory map and register descriptions below describe the registers using byte addresses.

MCM memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
E008_0008	Crossbar Switch (AXBS) Slave Configuration (MCM_PLASC)	16	R	000Fh	24.2.1/510
E008_000A	Crossbar Switch (AXBS) Master Configuration (MCM_PLAMC)	16	R	0017h	24.2.2/510
E008_000C	Crossbar Switch (AXBS) Control Register (MCM_PLACR)	32	R/W	0000_0000h	24.2.3/511

24.2.1 Crossbar Switch (AXBS) Slave Configuration (MCM_PLASC)

PLASC is a 16-bit read-only register identifying the presence/absence of bus slave connections to the device’s crossbar switch.

Address: E008_0000h base + 8h offset = E008_0008h

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	0								ASC							
Write																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1

MCM_PLASC field descriptions

Field	Description
15–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
ASC	Each bit in the ASC field indicates whether there is a corresponding connection to the crossbar switch's slave input port. 0 A bus slave connection to AXBS input port *n* is absent 1 A bus slave connection to AXBS input port *n* is present

24.2.2 Crossbar Switch (AXBS) Master Configuration (MCM_PLAMC)

PLAMC is a 16-bit read-only register identifying the presence/absence of bus master connections to the device's crossbar switch.

Address: E008_0000h base + Ah offset = E008_000Ah

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	0								AMC							
Write																
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	1

MCM_PLAMC field descriptions

Field	Description
15–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
AMC	Each bit in the AMC field indicates whether there is a corresponding connection to the AXBS master input port.

Table continues on the next page...

MCM_PLAMC field descriptions (continued)

Field	Description
	0 A bus master connection to AXBS input port *n* is absent 1 A bus master connection to AXBS input port *n* is present

24.2.3 Crossbar Switch (AXBS) Control Register (MCM_PLACR)

The PLACR register selects the arbitration policy for the crossbar masters.

Address: E008_0000h base + Ch offset = E008_000Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0						ARB	Reserved								
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MCM_PLACR field descriptions

Field	Description
31–10 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
9 ARB	Arbitration select 0 Fixed-priority arbitration for the crossbar masters 1 Round-robin arbitration for the crossbar masters
Reserved	This field is reserved.

Chapter 25
MCU: Crossbar Switch Lite (AXBS-Lite)

25.1 Introduction

NOTE

> For the chip-specific implementation details of this module's instances, see the chip configuration information.

The information found here provides information on the layout, configuration, and programming of the crossbar switch.

The crossbar switch connects bus masters and bus slaves using a crossbar switch structure. This structure allows up to four bus masters to access different bus slaves simultaneously, while providing arbitration among the bus masters when they access the same slave.

25.1.1 Features

The crossbar switch includes these features:

- Symmetric crossbar bus switch implementation
  - Allows concurrent accesses from different masters to different slaves
- Up to single-clock 32-bit transfer
- Programmable configuration for fixed-priority or round-robin slave port arbitration (see the chip-specific information).

25.2 Memory Map / Register Definition

This crossbar switch is designed for minimal gate count. It, therefore, has no memory-mapped configuration registers.

Please see the chip-specific information for information on whether the arbitration method in the crossbar switch is programmable, and by which module.

25.3 Functional Description

25.3.1 General operation

When a master accesses the crossbar switch, the access is immediately taken. If the targeted slave port of the access is available, then the access is immediately presented on the slave port. Single-clock or zero-wait-state accesses are possible through the crossbar. If the targeted slave port of the access is busy or parked on a different master port, the requesting master simply sees wait states inserted until the targeted slave port can service the master's request. The latency in servicing the request depends on each master's priority level and the responding slave's access time.

Because the crossbar switch appears to be just another slave to the master device, the master device has no knowledge of whether it actually owns the slave port it is targeting. While the master does not have control of the slave port it is targeting, it simply waits.

After the master has control of the slave port it is targeting, the master remains in control of the slave port until it relinquishes the slave port by running an IDLE cycle or by targeting a different slave port for its next access.

The master can also lose control of the slave port if another higher-priority master makes a request to the slave port.

The crossbar terminates all master IDLE transfers, as opposed to allowing the termination to come from one of the slave buses. Additionally, when no master is requesting access to a slave port, the crossbar drives IDLE transfers onto the slave bus, even though a default master may be granted access to the slave port.

When a slave bus is being idled by the crossbar, it remains parked with the last master to use the slave port. This is done to save the initial clock of arbitration delay that otherwise would be seen if the same master had to arbitrate to gain control of the slave port.

Chapter 26 MCU: Peripheral Bridge (AIPS-Lite)

26.1 Introduction

NOTE

> For the chip-specific implementation details of this module's instances, see the chip configuration information.

The peripheral bridge converts the crossbar switch interface to an interface that can access most of the slave peripherals on this chip.

The peripheral bridge occupies 64 MB of the address space, which is divided into peripheral slots of 4 KB. (It might be possible that all the peripheral slots are not used. See the memory map chapter for details on slot assignments.) The bridge includes separate clock enable inputs for each of the slots to accommodate slower peripherals.

26.1.1 Features

Key features of the peripheral bridge are:

- Supports peripheral slots with 8-, 16-, and 32-bit datapath width

26.1.2 General operation

The slave devices connected to the peripheral bridge are modules which contain a programming model of control and status registers. The system masters read and write these registers through the peripheral bridge.

The register maps of the peripherals are located on 4-KB boundaries. Each peripheral is allocated one or more 4-KB block(s) of the memory map.

26.2 Memory map/register definition

AIPS memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4000_0000	Master Privilege Register A (AIPS0_MPRA)	32	R/W	See section	26.2.1/516
4000_0020	Peripheral Access Control Register (AIPS0_PACRA)	32	R/W	See section	26.2.2/517
4000_0024	Peripheral Access Control Register (AIPS0_PACRB)	32	R/W	See section	26.2.2/517
4000_0028	Peripheral Access Control Register (AIPS0_PACRC)	32	R/W	See section	26.2.2/517
4000_002C	Peripheral Access Control Register (AIPS0_PACRD)	32	R/W	See section	26.2.2/517
4000_0040	Peripheral Access Control Register (AIPS0_PACRE)	32	R/W	See section	26.2.3/523
4000_0044	Peripheral Access Control Register (AIPS0_PACRF)	32	R/W	See section	26.2.3/523
4000_0048	Peripheral Access Control Register (AIPS0_PACRG)	32	R/W	See section	26.2.3/523
4000_004C	Peripheral Access Control Register (AIPS0_PACRH)	32	R/W	See section	26.2.3/523
4000_0050	Peripheral Access Control Register (AIPS0_PACRI)	32	R/W	See section	26.2.3/523
4000_0054	Peripheral Access Control Register (AIPS0_PACRJ)	32	R/W	See section	26.2.3/523
4000_0058	Peripheral Access Control Register (AIPS0_PACRK)	32	R/W	See section	26.2.3/523
4000_005C	Peripheral Access Control Register (AIPS0_PACRL)	32	R/W	See section	26.2.3/523
4000_0060	Peripheral Access Control Register (AIPS0_PACRM)	32	R/W	See section	26.2.3/523
4000_0064	Peripheral Access Control Register (AIPS0_PACRN)	32	R/W	See section	26.2.3/523
4000_0068	Peripheral Access Control Register (AIPS0_PACRO)	32	R/W	See section	26.2.3/523
4000_006C	Peripheral Access Control Register (AIPS0_PACRP)	32	R/W	See section	26.2.3/523

26.2.1 Master Privilege Register A (AIPSx_MPRA)

The MPRA specifies identical 4-bit fields defining the access-privilege level associated with a bus master to various peripherals on the chip. The register provides one field per bus master.

NOTE

> At reset, the default value loaded into the MPRA fields is chip-specific. See the chip configuration details for the value of a particular device.

A register field that maps to an unimplemented master or peripheral behaves as read-only-zero.

Each master is assigned a logical ID from 0 to 15. See the master logical ID assignment table in the chip-specific AIPS information.

Address: 4000_0000h base + 0h offset = 4000_0000h

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

* Notes:
- The reset value is chip-dependent and can be found in the chip-specific AIPS information.
- Reserved field: The reset value is chip-dependent and can be found in the chip-specific AIPS information.

AIPSx_MPRA field descriptions

Field	Description
31–28 Reserved	This field is reserved.
27–24 Reserved	This field is reserved.
23–20 Reserved	This field is reserved.
19–16 Reserved	This field is reserved.
15–12 Reserved	This field is reserved.
11–8 Reserved	This field is reserved.
7–4 Reserved	This field is reserved.
Reserved	This field is reserved.

26.2.2 Peripheral Access Control Register (AIPS*x*_PACR*n*)

Each PACR register consists of eight 4-bit PACR fields. Each PACR field defines the access levels for a particular peripheral. The mapping between a peripheral and its PACR field is shown in the table below. The peripheral assignment to each PACR is defined by the memory map slot that the peripheral is assigned to. See this chip's memory map for the assignment of a particular peripheral.

The following table shows the location of each peripheral slot's PACR field in the PACR registers.

Offset	Register	[31:28]	[27:24]	[23:20]	[19:16]	[15:12]	[11:8]	[7:4]	[3:0]
0x20	PACRA	PACR0	PACR1	PACR2	PACR3	PACR4	PACR5	PACR6	PACR7
0x24	PACRB	PACR8	PACR9	PACR10	PACR11	PACR12	PACR13	PACR14	PACR15
0x28	PACRC	PACR16	PACR17	PACR18	PACR19	PACR20	PACR21	PACR22	PACR23
0x2C	PACRD	PACR24	PACR25	PACR26	PACR27	PACR28	PACR29	PACR30	PACR31
0x30	Reserved								

Table continues on the next page...

Offset	Register	[31:28]	[27:24]	[23:20]	[19:16]	[15:12]	[11:8]	[7:4]	[3:0]
0x34	Reserved								
0x38	Reserved								
0x3C	Reserved								
0x40	PACRE	PACR32	PACR33	PACR34	PACR35	PACR36	PACR37	PACR38	PACR39
0x44	PACRF	PACR40	PACR41	PACR42	PACR43	PACR44	PACR45	PACR46	PACR47
0x48	PACRG	PACR48	PACR49	PACR50	PACR51	PACR52	PACR53	PACR54	PACR55
0x4C	PACRH	PACR56	PACR57	PACR58	PACR59	PACR60	PACR61	PACR62	PACR63
0x50	PACRI	PACR64	PACR65	PACR66	PACR67	PACR68	PACR69	PACR70	PACR71
0x54	PACRJ	PACR72	PACR73	PACR74	PACR75	PACR76	PACR77	PACR78	PACR79
0x58	PACRK	PACR80	PACR81	PACR82	PACR83	PACR84	PACR85	PACR86	PACR87
0x5C	PACRL	PACR88	PACR89	PACR90	PACR91	PACR92	PACR93	PACR94	PACR95
0x60	PACRM	PACR96	PACR97	PACR98	PACR99	PACR100	PACR101	PACR102	PACR103
0x64	PACRN	PACR104	PACR105	PACR106	PACR107	PACR108	PACR109	PACR110	PACR111
0x68	PACRO	PACR112	PACR113	PACR114	PACR115	PACR116	PACR117	PACR118	PACR119
0x6C	PACRP	PACR120	PACR121	PACR122	PACR123	PACR124	PACR125	PACR126	PACR127

NOTE

The register field descriptions for PACR A-D, which control peripheral slots 0-31, are shown below. The following section, Peripheral Access Control Register (AIPS_PACR*n*), shows the register field descriptions for PACR E-P. All PACR registers are identical. They are divided into two sections because they occupy two non-contiguous address spaces.

Address: 4000_0000h base + 20h offset + (4d × i), where i=0d to 3d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	SP0	WP0	TP0	0	SP1	WP1	TP1	0	SP2	WP2	TP2	0	SP3	WP3	TP3
W																
Reset	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	SP4	WP4	TP4	0	SP5	WP5	TP5	0	SP6	WP6	TP6	0	SP7	WP7	TP7
W																
Reset	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*

* Notes:
- The reset value is chip-dependent and can be found in the AIPS chip-specific information.

AIPSx_PACR*n* field descriptions

Field	Description
31 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

Table continues on the next page...

AIPSx_PACR*n* field descriptions (continued)

Field	Description
30 SP0	Supervisor Protect Determines whether the peripheral requires supervisor privilege level for accesses. When this field is set, the master privilege level must indicate the supervisor access attribute, and the MPR*x*[MPL*n*] control field for the master must be set. If not, access terminates with an error response and no peripheral access initiates. 0 This peripheral does not require supervisor privilege level for accesses. 1 This peripheral requires supervisor privilege level for accesses.
29 WP0	Write Protect Determines whether the peripheral allows write access. When this bit is set and a write access is attempted, access terminates with an error response and no peripheral access initiates. 0 This peripheral allows write accesses. 1 This peripheral is write protected.
28 TP0	Trusted Protect Determines whether the peripheral allows accesses from an untrusted master. When this field is set and an access is attempted by an untrusted master, the access terminates with an error response and no peripheral access initiates. 0 Accesses from an untrusted master are allowed. 1 Accesses from an untrusted master are not allowed.
27 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
26 SP1	Supervisor Protect Determines whether the peripheral requires supervisor privilege level for accesses. When this field is set, the master privilege level must indicate the supervisor access attribute, and the MPR*x*[MPL*n*] control field for the master must be set. If not, access terminates with an error response and no peripheral access initiates. 0 This peripheral does not require supervisor privilege level for accesses. 1 This peripheral requires supervisor privilege level for accesses.
25 WP1	Write Protect Determines whether the peripheral allows write accesses. When this field is set and a write access is attempted, access terminates with an error response and no peripheral access initiates. 0 This peripheral allows write accesses. 1 This peripheral is write protected.
24 TP1	Trusted Protect Determines whether the peripheral allows accesses from an untrusted master. When this bit is set and an access is attempted by an untrusted master, the access terminates with an error response and no peripheral access initiates. 0 Accesses from an untrusted master are allowed. 1 Accesses from an untrusted master are not allowed.
23 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

Table continues on the next page...

AIPSx_PACR*n* field descriptions (continued)

Field	Description
22 SP2	Supervisor Protect Determines whether the peripheral requires supervisor privilege level for accesses. When this field is set, the master privilege level must indicate the supervisor access attribute, and the MPR*x*[MPL*n*] control field for the master must be set. If not, access terminates with an error response and no peripheral access initiates. 0 This peripheral does not require supervisor privilege level for accesses. 1 This peripheral requires supervisor privilege level for accesses.
21 WP2	Write Protect Determines whether the peripheral allows write access. When this bit is set and a write access is attempted, access terminates with an error response and no peripheral access initiates. 0 This peripheral allows write accesses. 1 This peripheral is write protected.
20 TP2	Trusted Protect Determines whether the peripheral allows accesses from an untrusted master. When this field is set and an access is attempted by an untrusted master, the access terminates with an error response and no peripheral access initiates. 0 Accesses from an untrusted master are allowed. 1 Accesses from an untrusted master are not allowed.
19 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
18 SP3	Supervisor Protect Determines whether the peripheral requires supervisor privilege level for access. When this bit is set, the master privilege level must indicate the supervisor access attribute, and the MPR*x*[MPL*n*] control bit for the master must be set. If not, access terminates with an error response and no peripheral access initiates. 0 This peripheral does not require supervisor privilege level for accesses. 1 This peripheral requires supervisor privilege level for accesses.
17 WP3	Write Protect Determines whether the peripheral allows write accesses. When this field is set and a write access is attempted, access terminates with an error response and no peripheral access initiates. 0 This peripheral allows write accesses. 1 This peripheral is write protected.
16 TP3	Trusted Protect Determines whether the peripheral allows accesses from an untrusted master. When this bit is set and an access is attempted by an untrusted master, the access terminates with an error response and no peripheral access initiates. 0 Accesses from an untrusted master are allowed. 1 Accesses from an untrusted master are not allowed.
15 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

Table continues on the next page...

AIPSx_PACR*n* field descriptions (continued)

Field	Description
14 SP4	Supervisor Protect Determines whether the peripheral requires supervisor privilege level for accesses. When this field is set, the master privilege level must indicate the supervisor access attribute, and the MPR*x*[MPL*n*] control field for the master must be set. If not, access terminates with an error response and no peripheral access initiates. 0 This peripheral does not require supervisor privilege level for accesses. 1 This peripheral requires supervisor privilege level for accesses.
13 WP4	Write Protect Determines whether the peripheral allows write access. When this bit is set and a write access is attempted, access terminates with an error response and no peripheral access initiates. 0 This peripheral allows write accesses. 1 This peripheral is write protected.
12 TP4	Trusted Protect Determines whether the peripheral allows accesses from an untrusted master. When this field is set and an access is attempted by an untrusted master, the access terminates with an error response and no peripheral access initiates. 0 Accesses from an untrusted master are allowed. 1 Accesses from an untrusted master are not allowed.
11 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
10 SP5	Supervisor Protect Determines whether the peripheral requires supervisor privilege level for accesses. When this field is set, the master privilege level must indicate the supervisor access attribute, and the MPR*x*[MPL*n*] control field for the master must be set. If not, access terminates with an error response and no peripheral access initiates. 0 This peripheral does not require supervisor privilege level for accesses. 1 This peripheral requires supervisor privilege level for accesses.
9 WP5	Write Protect Determines whether the peripheral allows write accesses. When this field is set and a write access is attempted, access terminates with an error response and no peripheral access initiates. 0 This peripheral allows write accesses. 1 This peripheral is write protected.
8 TP5	Trusted Protect Determines whether the peripheral allows accesses from an untrusted master. When this field is set and an access is attempted by an untrusted master, the access terminates with an error response and no peripheral access initiates. 0 Accesses from an untrusted master are allowed. 1 Accesses from an untrusted master are not allowed.
7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

Table continues on the next page...

AIPSx_PACR*n* field descriptions (continued)

Field	Description
6 SP6	Supervisor Protect Determines whether the peripheral requires supervisor privilege level for accesses. When this field is set, the master privilege level must indicate the supervisor access attribute, and the MPR*x*[MPL*n*] control field for the master must be set. If not, access terminates with an error response and no peripheral access initiates. 0 This peripheral does not require supervisor privilege level for accesses. 1 This peripheral requires supervisor privilege level for accesses.
5 WP6	Write Protect Determines whether the peripheral allows write accesses. When this field is set and a write access is attempted, access terminates with an error response and no peripheral access initiates. 0 This peripheral allows write accesses. 1 This peripheral is write protected.
4 TP6	Trusted Protect Determines whether the peripheral allows accesses from an untrusted master. When this field is set and an access is attempted by an untrusted master, the access terminates with an error response and no peripheral access initiates. 0 Accesses from an untrusted master are allowed. 1 Accesses from an untrusted master are not allowed.
3 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
2 SP7	Supervisor Protect Determines whether the peripheral requires supervisor privilege level for accesses. When this field is set, the master privilege level must indicate the supervisor access attribute, and the MPR*x*[MPL*n*] control field for the master must be set. If not, access terminates with an error response and no peripheral access initiates. 0 This peripheral does not require supervisor privilege level for accesses. 1 This peripheral requires supervisor privilege level for accesses.
1 WP7	Write Protect Determines whether the peripheral allows write accesses. When this field is set and a write access is attempted, access terminates with an error response and no peripheral access initiates. 0 This peripheral allows write accesses. 1 This peripheral is write protected.
0 TP7	Trusted Protect Determines whether the peripheral allows accesses from an untrusted master. When this field is set and an access is attempted by an untrusted master, the access terminates with an error response and no peripheral access initiates. 0 Accesses from an untrusted master are allowed. 1 Accesses from an untrusted master are not allowed.

26.2.3 Peripheral Access Control Register (AIPSx_PACR*n*)

This section describes PACR registers E-P, which control peripheral slots 32-127. See Peripheral Access Control Register (AIPS_PACR*n*) for the description of these registers.

Address: 4000_0000h base + 40h offset + (4d × i), where i=0d to 11d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	SP0	WP0	TP0	0	SP1	WP1	TP1	0	SP2	WP2	TP2	0	SP3	WP3	TP3
W	▨				▨				▨				▨			
Reset	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	SP4	WP4	TP4	0	SP5	WP5	TP5	0	SP6	WP6	TP6	0	SP7	WP7	TP7
W	▨				▨				▨				▨			
Reset	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*

* Notes:

- The reset value is chip-dependent and can be found in the AIPS chip-specific information.

AIPSx_PACR*n* field descriptions

Field	Description
31 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
30 SP0	Supervisor Protect Determines whether the peripheral requires supervisor privilege level for accesses. When this field is set, the master privilege level must indicate the supervisor access attribute, and the MPR*x*[MPL*n*] control field for the master must be set. If not, access terminates with an error response and no peripheral access initiates. 0 This peripheral does not require supervisor privilege level for accesses. 1 This peripheral requires supervisor privilege level for accesses.
29 WP0	Write Protect Determines whether the peripheral allows write accesses. When this field is set and a write access is attempted, access terminates with an error response and no peripheral access initiates. 0 This peripheral allows write accesses. 1 This peripheral is write protected.
28 TP0	Trusted Protect Determines whether the peripheral allows accesses from an untrusted master. When this bit is set and an access is attempted by an untrusted master, the access terminates with an error response and no peripheral access initiates. 0 Accesses from an untrusted master are allowed. 1 Accesses from an untrusted master are not allowed.

Table continues on the next page...

AIPSx_PACR*n* field descriptions (continued)

Field	Description
27 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
26 SP1	Supervisor Protect Determines whether the peripheral requires supervisor privilege level for access. When this field is set, the master privilege level must indicate the supervisor access attribute, and the MPR*x*[MPL*n*] control field for the master must be set. If not, access terminates with an error response and no peripheral access initiates. 0 This peripheral does not require supervisor privilege level for accesses. 1 This peripheral requires supervisor privilege level for accesses.
25 WP1	Write Protect Determines whether the peripheral allows write accesses. When this field is set and a write access is attempted, access terminates with an error response and no peripheral access initiates. 0 This peripheral allows write accesses. 1 This peripheral is write protected.
24 TP1	Trusted Protect Determines whether the peripheral allows accesses from an untrusted master. When this field is set and an access is attempted by an untrusted master, the access terminates with an error response and no peripheral access initiates. 0 Accesses from an untrusted master are allowed. 1 Accesses from an untrusted master are not allowed.
23 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
22 SP2	Supervisor Protect Determines whether the peripheral requires supervisor privilege level for access. When this bit is set, the master privilege level must indicate the supervisor access attribute, and the MPR*x*[MPL*n*] control bit for the master must be set. If not, access terminates with an error response and no peripheral access initiates. 0 This peripheral does not require supervisor privilege level for accesses. 1 This peripheral requires supervisor privilege level for accesses.
21 WP2	Write Protect Determines whether the peripheral allows write accesses. When this field is set and a write access is attempted, access terminates with an error response and no peripheral access initiates. 0 This peripheral allows write accesses. 1 This peripheral is write protected.
20 TP2	Trusted Protect Determines whether the peripheral allows accesses from an untrusted master. When this bit is set and an access is attempted by an untrusted master, the access terminates with an error response and no peripheral access initiates. 0 Accesses from an untrusted master are allowed. 1 Accesses from an untrusted master are not allowed.

Table continues on the next page...

AIPSx_PACR*n* field descriptions (continued)

Field	Description
19 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
18 SP3	Supervisor Protect Determines whether the peripheral requires supervisor privilege level for accesses. When this field is set, the master privilege level must indicate the supervisor access attribute, and the MPR*x*[MPL*n*] control field for the master must be set. If not, access terminates with an error response and no peripheral access initiates. 0 This peripheral does not require supervisor privilege level for accesses. 1 This peripheral requires supervisor privilege level for accesses.
17 WP3	Write Protect Determines whether the peripheral allows write access. When this bit is set and a write access is attempted, access terminates with an error response and no peripheral access initiates. 0 This peripheral allows write accesses. 1 This peripheral is write protected.
16 TP3	Trusted Protect Determines whether the peripheral allows accesses from an untrusted master. When this field is set and an access is attempted by an untrusted master, the access terminates with an error response and no peripheral access initiates. 0 Accesses from an untrusted master are allowed. 1 Accesses from an untrusted master are not allowed.
15 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
14 SP4	Supervisor Protect Determines whether the peripheral requires supervisor privilege level for access. When this bit is set, the master privilege level must indicate the supervisor access attribute, and the MPR*x*[MPL*n*] control bit for the master must be set. If not, access terminates with an error response and no peripheral access initiates. 0 This peripheral does not require supervisor privilege level for accesses. 1 This peripheral requires supervisor privilege level for accesses.
13 WP4	Write Protect Determines whether the peripheral allows write accesses. When this field is set and a write access is attempted, access terminates with an error response and no peripheral access initiates. 0 This peripheral allows write accesses. 1 This peripheral is write protected.
12 TP4	Trusted Protect Determines whether the peripheral allows accesses from an untrusted master. When this bit is set and an access is attempted by an untrusted master, the access terminates with an error response and no peripheral access initiates. 0 Accesses from an untrusted master are allowed. 1 Accesses from an untrusted master are not allowed.

Table continues on the next page...

AIPSx_PACR*n* field descriptions (continued)

Field	Description
11 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
10 SP5	Supervisor Protect Determines whether the peripheral requires supervisor privilege level for accesses. When this field is set, the master privilege level must indicate the supervisor access attribute, and the MPR*x*[MPL*n*] control field for the master must be set. If not, access terminates with an error response and no peripheral access initiates. 0 This peripheral does not require supervisor privilege level for accesses. 1 This peripheral requires supervisor privilege level for accesses.
9 WP5	Write Protect Determines whether the peripheral allows write accesses. When this field is set and a write access is attempted, access terminates with an error response and no peripheral access initiates. 0 This peripheral allows write accesses. 1 This peripheral is write protected.
8 TP5	Trusted Protect Determines whether the peripheral allows accesses from an untrusted master. When this field is set and an access is attempted by an untrusted master, the access terminates with an error response and no peripheral access initiates. 0 Accesses from an untrusted master are allowed. 1 Accesses from an untrusted master are not allowed.
7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
6 SP6	Supervisor Protect Determines whether the peripheral requires supervisor privilege level for accesses. When this field is set, the master privilege level must indicate the supervisor access attribute, and the MPR*x*[MPL*n*] control field for the master must be set. If not, access terminates with an error response and no peripheral access initiates. 0 This peripheral does not require supervisor privilege level for accesses. 1 This peripheral requires supervisor privilege level for accesses.
5 WP6	Write Protect Determines whether the peripheral allows write accesses. When this field is set and a write access is attempted, access terminates with an error response and no peripheral access initiates. 0 This peripheral allows write accesses. 1 This peripheral is write protected.
4 TP6	Trusted Protect Determines whether the peripheral allows accesses from an untrusted master. When this field is set and an access is attempted by an untrusted master, the access terminates with an error response and no peripheral access initiates. 0 Accesses from an untrusted master are allowed. 1 Accesses from an untrusted master are not allowed.

Table continues on the next page...

AIPSx_PACR*n* field descriptions (continued)

Field	Description
3 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
2 SP7	Supervisor Protect Determines whether the peripheral requires supervisor privilege level for accesses. When this field is set, the master privilege level must indicate the supervisor access attribute, and the MPR*x*[MPL*n*] control field for the master must be set. If not, access terminates with an error response and no peripheral access initiates. 0 This peripheral does not require supervisor privilege level for accesses. 1 This peripheral requires supervisor privilege level for accesses.
1 WP7	Write Protect Determines whether the peripheral allows write accesses. When this field is set and a write access is attempted, access terminates with an error response and no peripheral access initiates. 0 This peripheral allows write accesses. 1 This peripheral is write protected.
0 TP7	Trusted Protect Determines whether the peripheral allows accesses from an untrusted master. When this field is set and an access is attempted by an untrusted master, the access terminates with an error response and no peripheral access initiates. 0 Accesses from an untrusted master are allowed. 1 Accesses from an untrusted master are not allowed.

26.3 Functional description

The peripheral bridge functions as a bus protocol translator between the crossbar switch and the slave peripheral bus.

The peripheral bridge manages all transactions destined for the attached slave devices and generates select signals for modules on the peripheral bus by decoding accesses within the attached address space.

26.3.1 Access support

Aligned and misaligned 32-bit, 16-bit, and byte accesses are supported for 32-bit peripherals. Misaligned accesses are supported to allow memory to be placed on the slave peripheral bus. Peripheral registers must not be misaligned, although no explicit checking is performed by the peripheral bridge. All accesses are performed with a single transfer.

All accesses to the peripheral slots must be sized less than or equal to the designated peripheral slot size. If an access is attempted that is larger than the targeted port, an error response is generated.

Chapter 27
MCU: Direct Memory Access Multiplexer (DMAMUX)

27.1 Introduction

NOTE

For the chip-specific implementation details of this module's instances, see the chip configuration information.

27.1.1 Overview

The Direct Memory Access Multiplexer (DMAMUX) routes DMA sources, called slots, to any of the 16 DMA channels. This process is illustrated in the following figure.

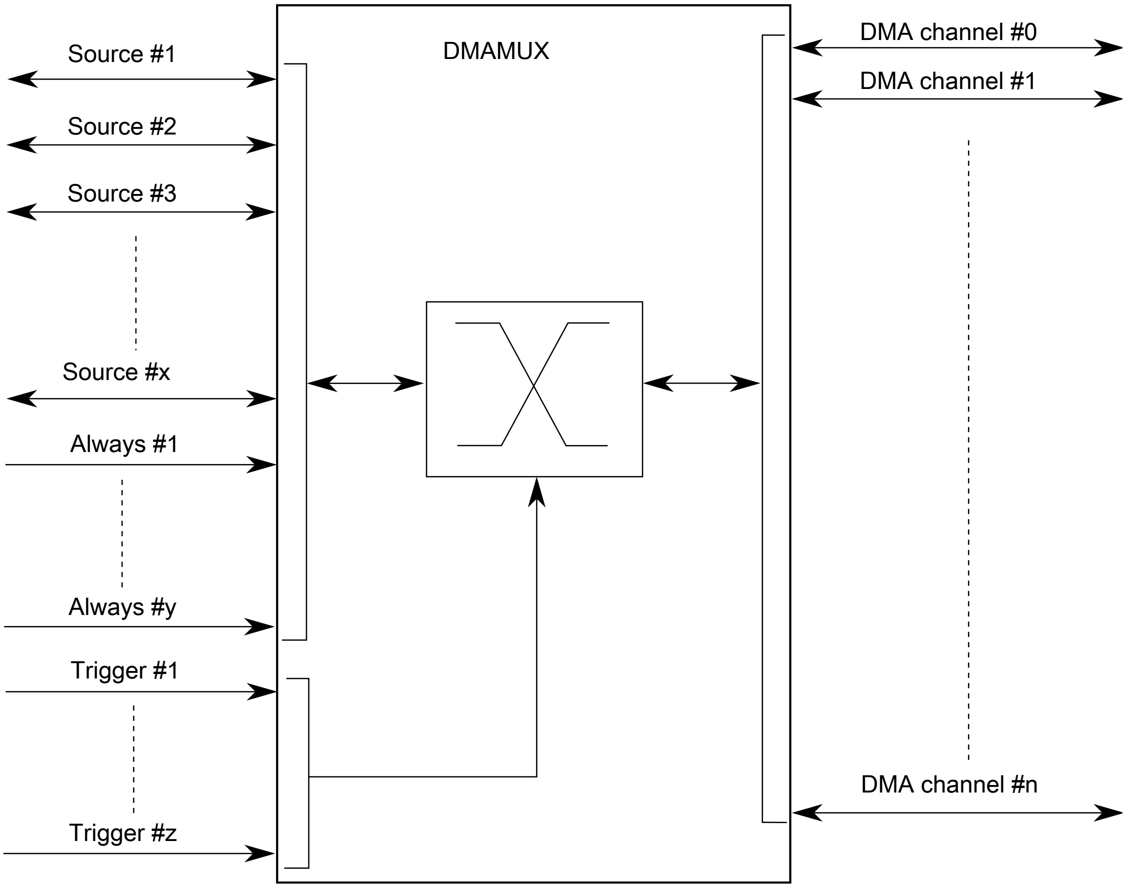


Figure 27-1. DMAMUX block diagram

27.1.2 Features

The DMAMUX module provides these features:

- Up to 52 peripheral slots and up to 10 always-on slots can be routed to 16 channels.
- 16 independently selectable DMA channel routers.
  - The first four channels additionally provide a trigger functionality.
- Each channel router can be assigned to one of the possible peripheral DMA slots or to one of the always-on slots.

27.1.3 Modes of operation

The following operating modes are available:

- Disabled mode

In this mode, the DMA channel is disabled. Because disabling and enabling of DMA channels is done primarily via the DMA configuration registers, this mode is used mainly as the reset state for a DMA channel in the DMA channel MUX. It may also be used to temporarily suspend a DMA channel while reconfiguration of the system takes place, for example, changing the period of a DMA trigger.

- Normal mode

  In this mode, a DMA source is routed directly to the specified DMA channel. The operation of the DMAMUX in this mode is completely transparent to the system.

- Periodic Trigger mode

  In this mode, a DMA source may only request a DMA transfer, such as when a transmit buffer becomes empty or a receive buffer becomes full, periodically.

27.2 External signal description

The DMAMUX has no external pins.

27.3 Memory map/register definition

This section provides a detailed description of all memory-mapped registers in the DMAMUX.

DMAMUX memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4002_1000	Channel Configuration register (DMAMUX_CHCFG)	8	R/W	00h	27.3.1/ 0

27.4 Functional description

The primary purpose of the DMAMUX is to provide flexibility in the system's use of the available DMA channels.

As such, configuration of the DMAMUX is intended to be a static procedure done during execution of the system boot code. However, if the procedure outlined in Enabling and configuring sources is followed, the configuration of the DMAMUX may be changed during the normal operation of the system.

Functionally, the DMAMUX channels may be divided into two classes:

- Channels that implement the normal routing functionality plus periodic triggering capability
- Channels that implement only the normal routing functionality

27.4.1 DMA channels with periodic triggering capability

Besides the normal routing functionality, the first 4 channels of the DMAMUX provide a special periodic triggering capability that can be used to provide an automatic mechanism to transmit bytes, frames, or packets at fixed intervals without the need for processor intervention.

Note

Because of the dynamic nature of the system (due to DMA channel priorities, bus arbitration, interrupt service routine lengths, etc.), the number of clock cycles between a trigger and the actual DMA transfer cannot be guaranteed.

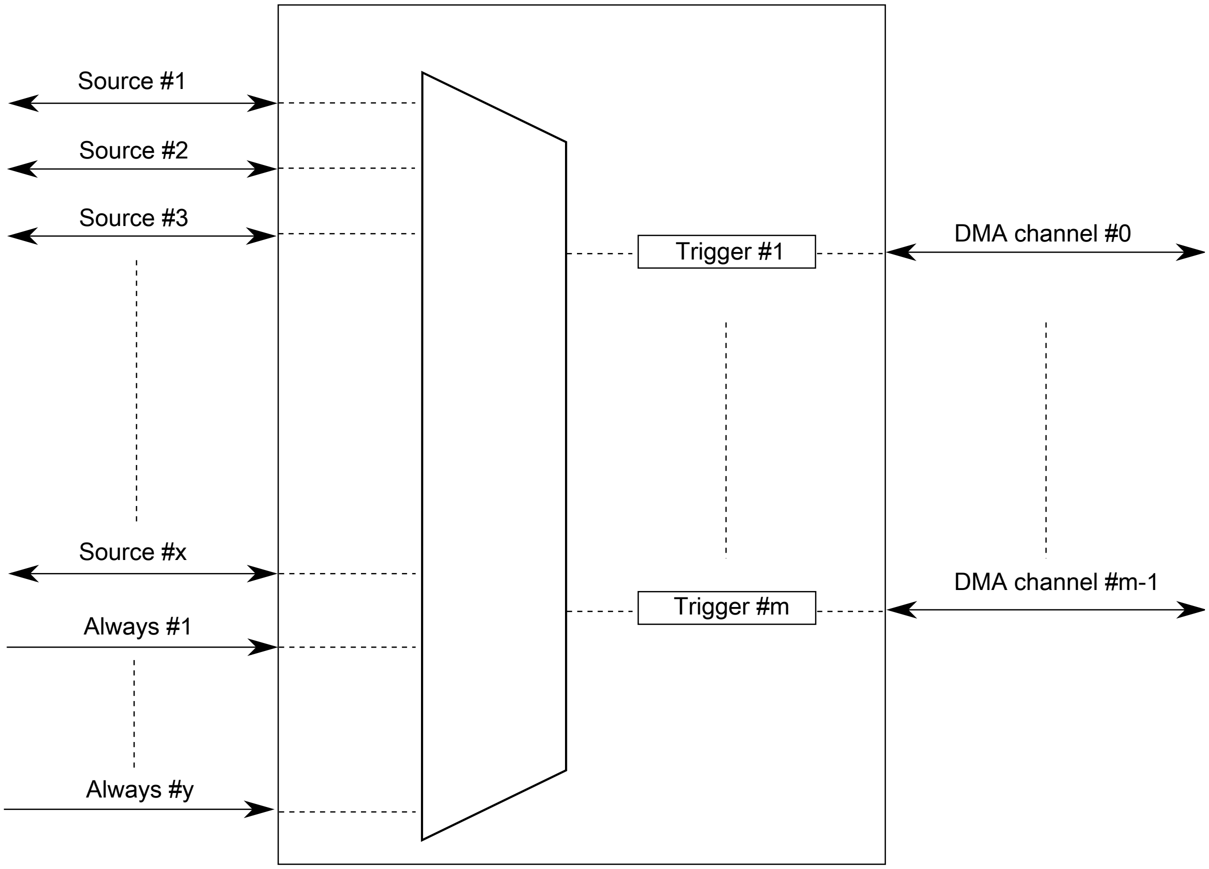


Figure 27-2. DMAMUX triggered channels

The DMA channel triggering capability allows the system to schedule regular DMA transfers, usually on the transmit side of certain peripherals, without the intervention of the processor. This trigger works by gating the request from the peripheral to the DMA until a trigger event has been seen. This is illustrated in the following figure.

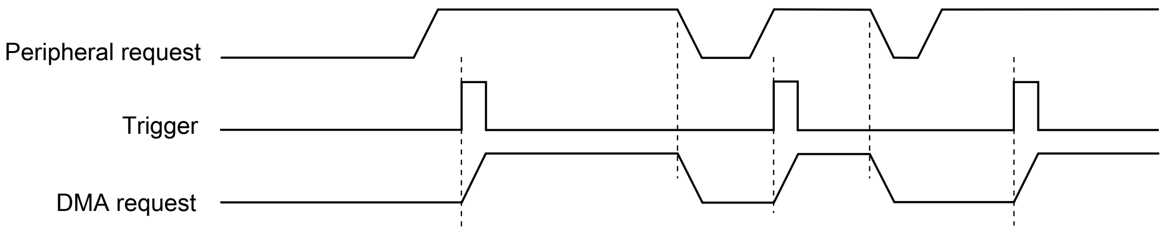


Figure 27-3. DMAMUX channel triggering: normal operation

After the DMA request has been serviced, the peripheral will negate its request, effectively resetting the gating mechanism until the peripheral reasserts its request and the next trigger event is seen. This means that if a trigger is seen, but the peripheral is not requesting a transfer, then that trigger will be ignored. This situation is illustrated in the following figure.

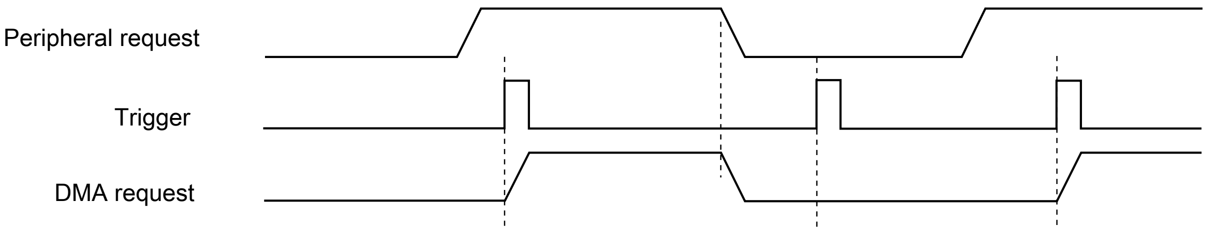


Figure 27-4. DMAMUX channel triggering: ignored trigger

This triggering capability may be used with any peripheral that supports DMA transfers, and is most useful for two types of situations:

- Periodically polling external devices on a particular bus

  As an example, the transmit side of an SPI is assigned to a DMA channel with a trigger, as described above. After it has been set up, the SPI will request DMA transfers, presumably from memory, as long as its transmit buffer is empty. By using a trigger on this channel, the SPI transfers can be automatically performed every 5 μs (as an example). On the receive side of the SPI, the SPI and DMA can be configured to transfer receive data into memory, effectively implementing a method to periodically read data from external devices and transfer the results into memory without processor intervention.

- Using the GPIO ports to drive or sample waveforms

By configuring the DMA to transfer data to one or more GPIO ports, it is possible to create complex waveforms using tabular data stored in on-chip memory. Conversely, using the DMA to periodically transfer data from one or more GPIO ports, it is possible to sample complex waveforms and store the results in tabular form in on-chip memory.

A more detailed description of the capability of each trigger, including resolution, range of values, and so on, may be found in the periodic interrupt timer section.

27.4.2 DMA channels with no triggering capability

The other channels of the DMAMUX provide the normal routing functionality as described in Modes of operation.

27.4.3 Always-enabled DMA sources

In addition to the peripherals that can be used as DMA sources, there are 10 additional DMA sources that are always enabled. Unlike the peripheral DMA sources, where the peripheral controls the flow of data during DMA transfers, the sources that are always enabled provide no such "throttling" of the data transfers. These sources are most useful in the following cases:

- Performing DMA transfers to/from GPIO—Moving data from/to one or more GPIO pins, either unthrottled (that is, as fast as possible), or periodically (using the DMA triggering capability).
- Performing DMA transfers from memory to memory—Moving data from memory to memory, typically as fast as possible, sometimes with software activation.
- Performing DMA transfers from memory to the external bus, or vice-versa—Similar to memory to memory transfers, this is typically done as quickly as possible.
- Any DMA transfer that requires software activation—Any DMA transfer that should be explicitly started by software.

In cases where software should initiate the start of a DMA transfer, an always-enabled DMA source can be used to provide maximum flexibility. When activating a DMA channel via software, subsequent executions of the minor loop require that a new start event be sent. This can either be a new software activation, or a transfer request from the DMA channel MUX. The options for doing this are:

- Transfer all data in a single minor loop.

By configuring the DMA to transfer all of the data in a single minor loop (that is, major loop counter = 1), no reactivation of the channel is necessary. The disadvantage to this option is the reduced granularity in determining the load that the DMA transfer will impose on the system. For this option, the DMA channel must be disabled in the DMA channel MUX.

- Use explicit software reactivation.

  In this option, the DMA is configured to transfer the data using both minor and major loops, but the processor is required to reactivate the channel by writing to the DMA registers *after every minor loop*. For this option, the DMA channel must be disabled in the DMA channel MUX.

- Use an always-enabled DMA source.

  In this option, the DMA is configured to transfer the data using both minor and major loops, and the DMA channel MUX does the channel reactivation. For this option, the DMA channel should be enabled and pointing to an "always enabled" source. Note that the reactivation of the channel can be continuous (DMA triggering is disabled) or can use the DMA triggering capability. In this manner, it is possible to execute periodic transfers of packets of data from one source to another, without processor intervention.

27.5 Initialization/application information

This section provides instructions for initializing the DMA channel MUX.

27.5.1 Reset

The reset state of each individual bit is shown in Memory map/register definition. In summary, after reset, all channels are disabled and must be explicitly enabled before use.

27.5.2 Enabling and configuring sources

To enable a source with periodic triggering:

1. Determine with which DMA channel the source will be associated. Note that only the first 4 DMA channels have periodic triggering capability.
2. Clear the CHCFG[ENBL] and CHCFG[TRIG] fields of the DMA channel.

3. Ensure that the DMA channel is properly configured in the DMA. The DMA channel may be enabled at this point.
4. Configure the corresponding timer.
5. Select the source to be routed to the DMA channel. Write to the corresponding CHCFG register, ensuring that the CHCFG[ENBL] and CHCFG[TRIG] fields are set.

NOTE

> The following is an example. See the chip configuration details for the number of this device's DMA channels that have triggering capability.

To configure source #5 transmit for use with DMA channel 1, with periodic triggering capability:

1. Write 0x00 to CHCFG1.
2. Configure channel 1 in the DMA, including enabling the channel.
3. Configure a timer for the desired trigger interval.
4. Write 0xC5 to CHCFG1.

The following code example illustrates steps 1 and 4 above:

```
void DMAMUX_Init(uint8_t DMA_CH, uint8_t DMAMUX_SOURCE)
{
    DMAMUX_0.CHCFG[DMA_CH].B.SOURCE = DMAMUX_SOURCE;
    DMAMUX_0.CHCFG[DMA_CH].B.ENBL   = 1;
    DMAMUX_0.CHCFG[DMA_CH].B.TRIG   = 1;
}
```

To enable a source, without periodic triggering:

1. Determine with which DMA channel the source will be associated. Note that only the first 4 DMA channels have periodic triggering capability.
2. Clear the CHCFG[ENBL] and CHCFG[TRIG] fields of the DMA channel.
3. Ensure that the DMA channel is properly configured in the DMA. The DMA channel may be enabled at this point.
4. Select the source to be routed to the DMA channel. Write to the corresponding CHCFG register, ensuring that CHCFG[ENBL] is set while CHCFG[TRIG] is cleared.

NOTE

> The following is an example. See the chip configuration details for the number of this device's DMA channels that have triggering capability.

To configure source #5 transmit for use with DMA channel 1, with no periodic triggering capability:

1. Write 0x00 to CHCFG1.

2. Configure channel 1 in the DMA, including enabling the channel.
3. Write 0x85 to CHCFG1.

The following code example illustrates steps 1 and 3 above:

To disable a source:

A particular DMA source may be disabled by not writing the corresponding source value into any of the CHCFG registers. Additionally, some module-specific configuration may be necessary. See the appropriate section for more details.

To switch the source of a DMA channel:

1. Disable the DMA channel in the DMA and reconfigure the channel for the new source.
2. Clear the CHCFG[ENBL] and CHCFG[TRIG] bits of the DMA channel.
3. Select the source to be routed to the DMA channel. Write to the corresponding CHCFG register, ensuring that the CHCFG[ENBL] and CHCFG[TRIG] fields are set.

To switch DMA channel 8 from source #5 transmit to source #7 transmit:

1. In the DMA configuration registers, disable DMA channel 8 and reconfigure it to handle the transfers to peripheral slot 7. This example assumes channel 8 doesn't have triggering capability.
2. Write 0x00 to CHCFG8.
3. Write 0x87 to CHCFG8. (In this example, setting CHCFG[TRIG] would have no effect due to the assumption that channel 8 does not support the periodic triggering functionality.)

The following code example illustrates steps 2 and 3 above:

Chapter 28
MCU: Direct Memory Access Controller (eDMA)

28.1 Introduction

NOTE

> For the chip-specific implementation details of this module's instances, see the chip configuration information.

The enhanced direct memory access (eDMA) controller is a second-generation module capable of performing complex data transfers with minimal intervention from a host processor. The hardware microarchitecture includes:

- A DMA engine that performs:
  - Source address and destination address calculations
  - Data-movement operations
- Local memory containing transfer control descriptors for each of the 16 channels

28.1.1 eDMA system block diagram

Figure 28-1 illustrates the components of the eDMA system, including the eDMA module ("engine").

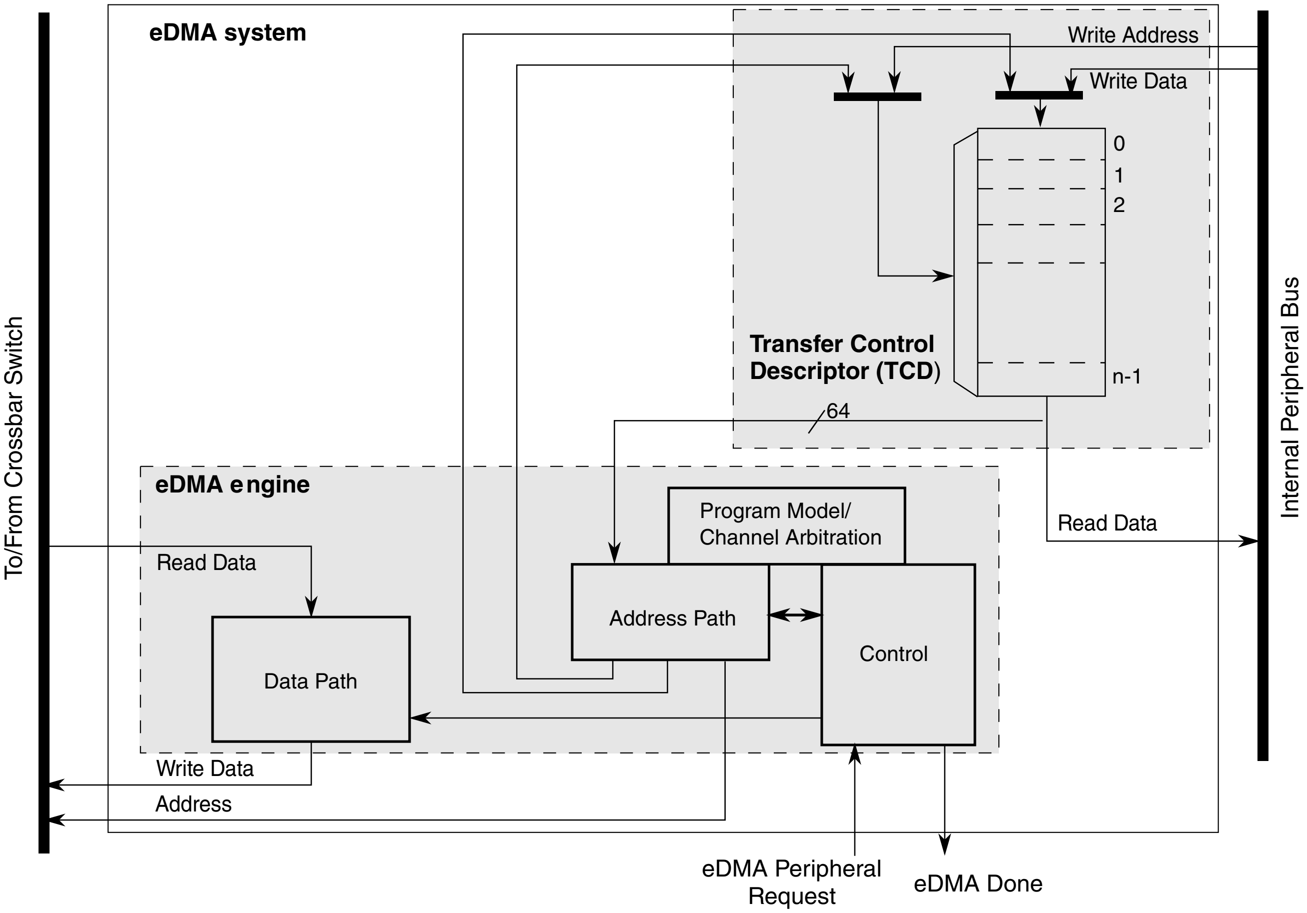


Figure 28-1. eDMA system block diagram

28.1.2 Block parts

The eDMA module is partitioned into two major modules: the eDMA engine and the transfer-control descriptor local memory.

The eDMA engine is further partitioned into four submodules:

Table 28-1. eDMA engine submodules

Submodule	Function
Address path	This block implements registered versions of two channel transfer control descriptors, channel x and channel y, and manages all master bus-address calculations. All the channels provide the same functionality. This structure allows data transfers associated with one channel to be preempted after the completion of a read/write sequence if a higher priority channel activation is asserted while the first channel is active. After a channel is activated, it runs until the minor loop is completed, unless preempted by a higher priority channel. This provides a mechanism (enabled by DCHPRI*n*[ECP]) where a large data move operation can be preempted to minimize the time another channel is blocked from execution. When any channel is selected to execute, the contents of its TCD are read from local memory and loaded into the address path channel x registers for a normal start and into channel y registers for a preemption start. After the minor loop completes execution, the address path hardware writes

Table continues on the next page...

Table 28-1. eDMA engine submodules (continued)

Submodule	Function
	the new values for the TCD*n*_{SADDR, DADDR, CITER} back to local memory. If the major iteration count is exhausted, additional processing is performed, including the final address pointer updates, reloading the TCD*n*_CITER field, and a possible fetch of the next TCD*n* from memory as part of a scatter/gather operation.
Data path	This block implements the bus master read/write datapath. It includes a data buffer and the necessary multiplex logic to support any required data alignment. The internal read data bus is the primary input, and the internal write data bus is the primary output. The address and data path modules directly support the 2-stage pipelined internal bus. The address path module represents the 1st stage of the bus pipeline (address phase), while the data path module implements the 2nd stage of the pipeline (data phase).
Program model/channel arbitration	This block implements the first section of the eDMA programming model as well as the channel arbitration logic. The programming model registers are connected to the internal peripheral bus. The eDMA peripheral request inputs and interrupt request outputs are also connected to this block (via control logic).
Control	This block provides all the control functions for the eDMA engine. For data transfers where the source and destination sizes are equal, the eDMA engine performs a series of source read/destination write operations until the number of bytes specified in the minor loop byte count has moved. For descriptors where the sizes are not equal, multiple accesses of the smaller size data are required for each reference of the larger size. As an example, if the source size references 16-bit data and the destination is 32-bit data, two reads are performed, then one 32-bit write.

The transfer-control descriptor local memory is further partitioned into:

Table 28-2. Transfer control descriptor memory

Submodule	Description
Memory controller	This logic implements the required dual-ported controller, managing accesses from the eDMA engine as well as references from the internal peripheral bus. As noted earlier, in the event of simultaneous accesses, the eDMA engine is given priority and the peripheral transaction is stalled.
Memory array	TCD storage for each channel's transfer profile.

28.1.3 Features

The eDMA is a highly programmable data-transfer engine optimized to minimize any required intervention from the host processor. It is intended for use in applications where the data size to be transferred is statically known and not defined within the transferred data itself. The eDMA module features:

- All data movement via dual-address transfers: read from source, write to destination
  - Programmable source and destination addresses and transfer size
  - Support for enhanced addressing modes

- 16-channel implementation that performs complex data transfers with minimal intervention from a host processor
    - Internal data buffer, used as temporary storage to support 16- and 32-byte transfers
    - Connections to the crossbar switch for bus mastering the data movement
- Transfer control descriptor (TCD) organized to support two-deep, nested transfer operations
    - 32-byte TCD stored in local memory for each channel
    - An inner data transfer loop defined by a minor byte transfer count
    - An outer data transfer loop defined by a major iteration count
- Channel activation via one of three methods:
    - Explicit software initiation
    - Initiation via a channel-to-channel linking mechanism for continuous transfers
    - Peripheral-paced hardware requests, one per channel
- Fixed-priority and round-robin channel arbitration
- Channel completion reported via programmable interrupt requests
    - One interrupt per channel, which can be asserted at completion of major iteration count
    - Programmable error terminations per channel and logically summed together to form one error interrupt to the interrupt controller
- Programmable support for scatter/gather DMA processing
- Support for complex data structures

In the discussion of this module, *n* is used to reference the channel number.

28.2 Modes of operation

The eDMA operates in the following modes:

Table 28-3. Modes of operation

Mode	Description
Normal	In Normal mode, the eDMA transfers data between a source and a destination. The source and destination can be a memory block or an I/O block capable of operation with the eDMA. A service request initiates a transfer of a specific number of bytes (NBYTES) as specified in the transfer control descriptor (TCD). The minor loop is the sequence of read-write operations that transfers these NBYTES per service request. Each service request executes one iteration of the major loop, which transfers NBYTES of data.
Debug	DMA operation is configurable in Debug mode via the control register: • If CR[EDBG] is cleared, the DMA continues to operate. • If CR[EDBG] is set, the eDMA stops transferring data. If Debug mode is entered while a channel is active, the eDMA continues operation until the channel retires.
Wait	Before entering Wait mode, the DMA attempts to complete its current transfer. After the transfer completes, the device enters Wait mode.

28.3 Memory map/register definition

The eDMA's programming model is partitioned into two regions:

- The first region defines a number of registers providing control functions
- The second region corresponds to the local transfer control descriptor (TCD) memory

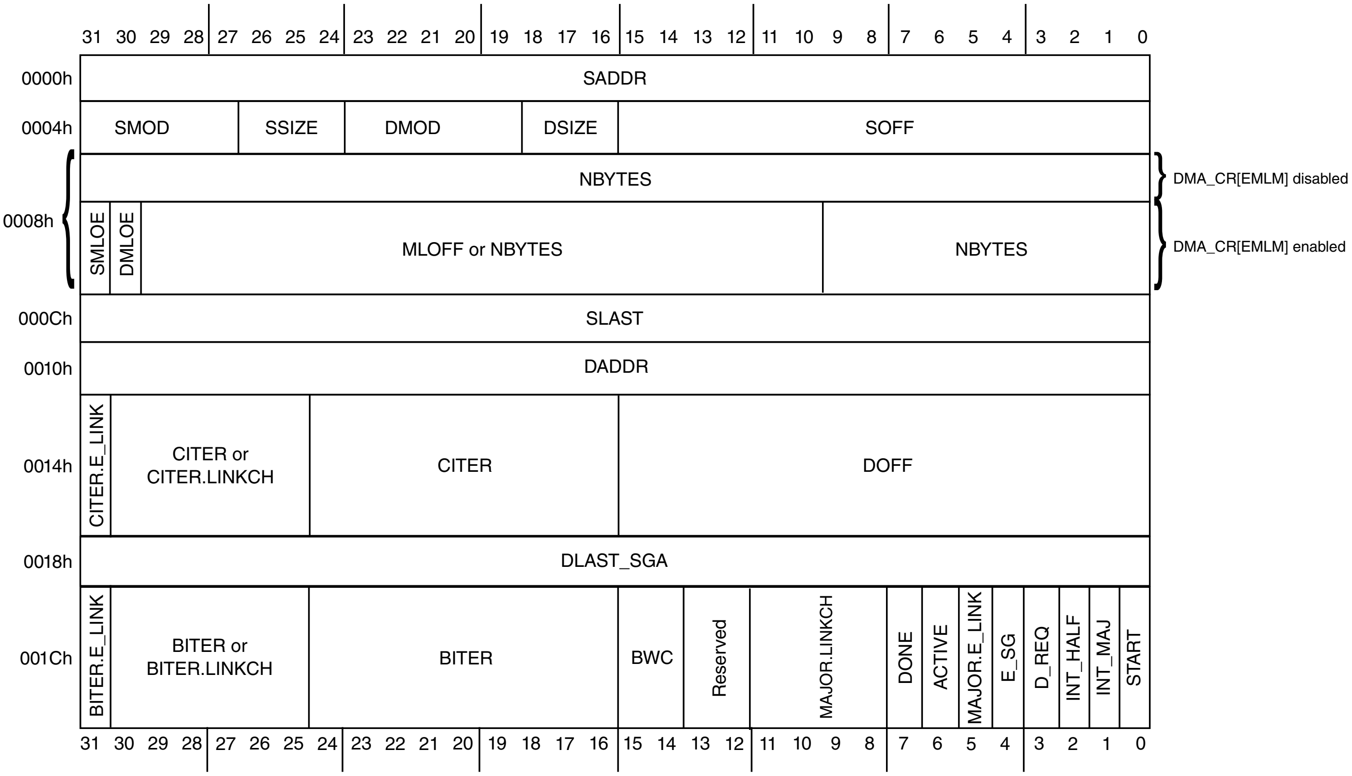
28.3.1 TCD memory

Each channel requires a 32-byte transfer control descriptor for defining the desired data movement operation. The channel descriptors are stored in the local memory in sequential order: channel 0, channel 1, ... channel 15. Each TCD*n* definition is presented as 11 registers of 16 or 32 bits.

28.3.2 TCD initialization

Prior to activating a channel, you must initialize its TCD with the appropriate transfer profile.

28.3.3 TCD structure



28.3.4 Reserved memory and bit fields

- Reading reserved bits in a register returns the value of zero.
- Writes to reserved bits in a register are ignored.
- Reading or writing a reserved memory location generates a bus error.

DMA memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4000_8000	Control Register (DMA_CR)	32	R/W	0000_0000h	28.3.1/557
4000_8004	Error Status Register (DMA_ES)	32	R	0000_0000h	28.3.2/560
4000_800C	Enable Request Register (DMA_ERQ)	32	R/W	0000_0000h	28.3.3/562
4000_8014	Enable Error Interrupt Register (DMA_EEI)	32	R/W	0000_0000h	28.3.4/564
4000_8018	Clear Enable Error Interrupt Register (DMA_CEEI)	8	W (always reads 0)	00h	28.3.5/566
4000_8019	Set Enable Error Interrupt Register (DMA_SEEI)	8	W (always reads 0)	00h	28.3.6/567

Table continues on the next page...

DMA memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4000_801A	Clear Enable Request Register (DMA_CERQ)	8	W (always reads 0)	00h	28.3.7/568
4000_801B	Set Enable Request Register (DMA_SERQ)	8	W (always reads 0)	00h	28.3.8/569
4000_801C	Clear DONE Status Bit Register (DMA_CDNE)	8	W (always reads 0)	00h	28.3.9/570
4000_801D	Set START Bit Register (DMA_SSRT)	8	W (always reads 0)	00h	28.3.10/ 571
4000_801E	Clear Error Register (DMA_CERR)	8	W (always reads 0)	00h	28.3.11/ 572
4000_801F	Clear Interrupt Request Register (DMA_CINT)	8	W (always reads 0)	00h	28.3.12/ 573
4000_8024	Interrupt Request Register (DMA_INT)	32	R/W	0000_0000h	28.3.13/ 574
4000_802C	Error Register (DMA_ERR)	32	R/W	0000_0000h	28.3.14/ 576
4000_8034	Hardware Request Status Register (DMA_HRS)	32	R	0000_0000h	28.3.15/ 579
4000_8100	Channel n Priority Register (DMA_DCHPRI3)	8	R/W	See section	28.3.16/ 582
4000_8101	Channel n Priority Register (DMA_DCHPRI2)	8	R/W	See section	28.3.16/ 582
4000_8102	Channel n Priority Register (DMA_DCHPRI1)	8	R/W	See section	28.3.16/ 582
4000_8103	Channel n Priority Register (DMA_DCHPRI0)	8	R/W	See section	28.3.16/ 582
4000_8104	Channel n Priority Register (DMA_DCHPRI7)	8	R/W	See section	28.3.16/ 582
4000_8105	Channel n Priority Register (DMA_DCHPRI6)	8	R/W	See section	28.3.16/ 582
4000_8106	Channel n Priority Register (DMA_DCHPRI5)	8	R/W	See section	28.3.16/ 582
4000_8107	Channel n Priority Register (DMA_DCHPRI4)	8	R/W	See section	28.3.16/ 582
4000_8108	Channel n Priority Register (DMA_DCHPRI11)	8	R/W	See section	28.3.16/ 582
4000_8109	Channel n Priority Register (DMA_DCHPRI10)	8	R/W	See section	28.3.16/ 582

Table continues on the next page...

DMA memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4000_810A	Channel n Priority Register (DMA_DCHPRI9)	8	R/W	See section	28.3.16/ 582
4000_810B	Channel n Priority Register (DMA_DCHPRI8)	8	R/W	See section	28.3.16/ 582
4000_810C	Channel n Priority Register (DMA_DCHPRI15)	8	R/W	See section	28.3.16/ 582
4000_810D	Channel n Priority Register (DMA_DCHPRI14)	8	R/W	See section	28.3.16/ 582
4000_810E	Channel n Priority Register (DMA_DCHPRI13)	8	R/W	See section	28.3.16/ 582
4000_810F	Channel n Priority Register (DMA_DCHPRI12)	8	R/W	See section	28.3.16/ 582
4000_9000	TCD Source Address (DMA_TCD0_SADDR)	32	R/W	Undefined	28.3.17/ 583
4000_9004	TCD Signed Source Address Offset (DMA_TCD0_SOFF)	16	R/W	Undefined	28.3.18/ 583
4000_9006	TCD Transfer Attributes (DMA_TCD0_ATTR)	16	R/W	Undefined	28.3.19/ 584
4000_9008	TCD Minor Byte Count (Minor Loop Mapping Disabled) (DMA_TCD0_NBYTES_MLNO)	32	R/W	Undefined	28.3.20/ 585
4000_9008	TCD Signed Minor Loop Offset (Minor Loop Mapping Enabled and Offset Disabled) (DMA_TCD0_NBYTES_MLOFFNO)	32	R/W	Undefined	28.3.21/ 586
4000_9008	TCD Signed Minor Loop Offset (Minor Loop Mapping and Offset Enabled) (DMA_TCD0_NBYTES_MLOFFYES)	32	R/W	Undefined	28.3.22/ 587
4000_900C	TCD Last Source Address Adjustment (DMA_TCD0_SLAST)	32	R/W	Undefined	28.3.23/ 588
4000_9010	TCD Destination Address (DMA_TCD0_DADDR)	32	R/W	Undefined	28.3.24/ 589
4000_9014	TCD Signed Destination Address Offset (DMA_TCD0_DOFF)	16	R/W	Undefined	28.3.25/ 589
4000_9016	TCD Current Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA_TCD0_CITER_ELINKYES)	16	R/W	Undefined	28.3.26/ 590
4000_9016	DMA_TCD0_CITER_ELINKNO	16	R/W	Undefined	28.3.27/ 591
4000_9018	TCD Last Destination Address Adjustment/Scatter Gather Address (DMA_TCD0_DLASTSGA)	32	R/W	Undefined	28.3.28/ 592
4000_901C	TCD Control and Status (DMA_TCD0_CSR)	16	R/W	Undefined	28.3.29/ 593
4000_901E	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA_TCD0_BITER_ELINKYES)	16	R/W	Undefined	28.3.30/ 595
4000_901E	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Disabled) (DMA_TCD0_BITER_ELINKNO)	16	R/W	Undefined	28.3.31/ 596

Table continues on the next page...

DMA memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4000_9020	TCD Source Address (DMA_TCD1_SADDR)	32	R/W	Undefined	28.3.17/ 583
4000_9024	TCD Signed Source Address Offset (DMA_TCD1_SOFF)	16	R/W	Undefined	28.3.18/ 583
4000_9026	TCD Transfer Attributes (DMA_TCD1_ATTR)	16	R/W	Undefined	28.3.19/ 584
4000_9028	TCD Minor Byte Count (Minor Loop Mapping Disabled) (DMA_TCD1_NBYTES_MLNO)	32	R/W	Undefined	28.3.20/ 585
4000_9028	TCD Signed Minor Loop Offset (Minor Loop Mapping Enabled and Offset Disabled) (DMA_TCD1_NBYTES_MLOFFNO)	32	R/W	Undefined	28.3.21/ 586
4000_9028	TCD Signed Minor Loop Offset (Minor Loop Mapping and Offset Enabled) (DMA_TCD1_NBYTES_MLOFFYES)	32	R/W	Undefined	28.3.22/ 587
4000_902C	TCD Last Source Address Adjustment (DMA_TCD1_SLAST)	32	R/W	Undefined	28.3.23/ 588
4000_9030	TCD Destination Address (DMA_TCD1_DADDR)	32	R/W	Undefined	28.3.24/ 589
4000_9034	TCD Signed Destination Address Offset (DMA_TCD1_DOFF)	16	R/W	Undefined	28.3.25/ 589
4000_9036	TCD Current Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA_TCD1_CITER_ELINKYES)	16	R/W	Undefined	28.3.26/ 590
4000_9036	DMA_TCD1_CITER_ELINKNO	16	R/W	Undefined	28.3.27/ 591
4000_9038	TCD Last Destination Address Adjustment/Scatter Gather Address (DMA_TCD1_DLASTSGA)	32	R/W	Undefined	28.3.28/ 592
4000_903C	TCD Control and Status (DMA_TCD1_CSR)	16	R/W	Undefined	28.3.29/ 593
4000_903E	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA_TCD1_BITER_ELINKYES)	16	R/W	Undefined	28.3.30/ 595
4000_903E	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Disabled) (DMA_TCD1_BITER_ELINKNO)	16	R/W	Undefined	28.3.31/ 596
4000_9040	TCD Source Address (DMA_TCD2_SADDR)	32	R/W	Undefined	28.3.17/ 583
4000_9044	TCD Signed Source Address Offset (DMA_TCD2_SOFF)	16	R/W	Undefined	28.3.18/ 583
4000_9046	TCD Transfer Attributes (DMA_TCD2_ATTR)	16	R/W	Undefined	28.3.19/ 584
4000_9048	TCD Minor Byte Count (Minor Loop Mapping Disabled) (DMA_TCD2_NBYTES_MLNO)	32	R/W	Undefined	28.3.20/ 585
4000_9048	TCD Signed Minor Loop Offset (Minor Loop Mapping Enabled and Offset Disabled) (DMA_TCD2_NBYTES_MLOFFNO)	32	R/W	Undefined	28.3.21/ 586
4000_9048	TCD Signed Minor Loop Offset (Minor Loop Mapping and Offset Enabled) (DMA_TCD2_NBYTES_MLOFFYES)	32	R/W	Undefined	28.3.22/ 587

Table continues on the next page...

DMA memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4000_904C	TCD Last Source Address Adjustment (DMA_TCD2_SLAST)	32	R/W	Undefined	28.3.23/ 588
4000_9050	TCD Destination Address (DMA_TCD2_DADDR)	32	R/W	Undefined	28.3.24/ 589
4000_9054	TCD Signed Destination Address Offset (DMA_TCD2_DOFF)	16	R/W	Undefined	28.3.25/ 589
4000_9056	TCD Current Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA_TCD2_CITER_ELINKYES)	16	R/W	Undefined	28.3.26/ 590
4000_9056	DMA_TCD2_CITER_ELINKNO	16	R/W	Undefined	28.3.27/ 591
4000_9058	TCD Last Destination Address Adjustment/Scatter Gather Address (DMA_TCD2_DLASTSGA)	32	R/W	Undefined	28.3.28/ 592
4000_905C	TCD Control and Status (DMA_TCD2_CSR)	16	R/W	Undefined	28.3.29/ 593
4000_905E	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA_TCD2_BITER_ELINKYES)	16	R/W	Undefined	28.3.30/ 595
4000_905E	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Disabled) (DMA_TCD2_BITER_ELINKNO)	16	R/W	Undefined	28.3.31/ 596
4000_9060	TCD Source Address (DMA_TCD3_SADDR)	32	R/W	Undefined	28.3.17/ 583
4000_9064	TCD Signed Source Address Offset (DMA_TCD3_SOFF)	16	R/W	Undefined	28.3.18/ 583
4000_9066	TCD Transfer Attributes (DMA_TCD3_ATTR)	16	R/W	Undefined	28.3.19/ 584
4000_9068	TCD Minor Byte Count (Minor Loop Mapping Disabled) (DMA_TCD3_NBYTES_MLNO)	32	R/W	Undefined	28.3.20/ 585
4000_9068	TCD Signed Minor Loop Offset (Minor Loop Mapping Enabled and Offset Disabled) (DMA_TCD3_NBYTES_MLOFFNO)	32	R/W	Undefined	28.3.21/ 586
4000_9068	TCD Signed Minor Loop Offset (Minor Loop Mapping and Offset Enabled) (DMA_TCD3_NBYTES_MLOFFYES)	32	R/W	Undefined	28.3.22/ 587
4000_906C	TCD Last Source Address Adjustment (DMA_TCD3_SLAST)	32	R/W	Undefined	28.3.23/ 588
4000_9070	TCD Destination Address (DMA_TCD3_DADDR)	32	R/W	Undefined	28.3.24/ 589
4000_9074	TCD Signed Destination Address Offset (DMA_TCD3_DOFF)	16	R/W	Undefined	28.3.25/ 589
4000_9076	TCD Current Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA_TCD3_CITER_ELINKYES)	16	R/W	Undefined	28.3.26/ 590
4000_9076	DMA_TCD3_CITER_ELINKNO	16	R/W	Undefined	28.3.27/ 591
4000_9078	TCD Last Destination Address Adjustment/Scatter Gather Address (DMA_TCD3_DLASTSGA)	32	R/W	Undefined	28.3.28/ 592

Table continues on the next page...

DMA memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4000_907C	TCD Control and Status (DMA_TCD3_CSR)	16	R/W	Undefined	28.3.29/ 593
4000_907E	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA_TCD3_BITER_ELINKYES)	16	R/W	Undefined	28.3.30/ 595
4000_907E	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Disabled) (DMA_TCD3_BITER_ELINKNO)	16	R/W	Undefined	28.3.31/ 596
4000_9080	TCD Source Address (DMA_TCD4_SADDR)	32	R/W	Undefined	28.3.17/ 583
4000_9084	TCD Signed Source Address Offset (DMA_TCD4_SOFF)	16	R/W	Undefined	28.3.18/ 583
4000_9086	TCD Transfer Attributes (DMA_TCD4_ATTR)	16	R/W	Undefined	28.3.19/ 584
4000_9088	TCD Minor Byte Count (Minor Loop Mapping Disabled) (DMA_TCD4_NBYTES_MLNO)	32	R/W	Undefined	28.3.20/ 585
4000_9088	TCD Signed Minor Loop Offset (Minor Loop Mapping Enabled and Offset Disabled) (DMA_TCD4_NBYTES_MLOFFNO)	32	R/W	Undefined	28.3.21/ 586
4000_9088	TCD Signed Minor Loop Offset (Minor Loop Mapping and Offset Enabled) (DMA_TCD4_NBYTES_MLOFFYES)	32	R/W	Undefined	28.3.22/ 587
4000_908C	TCD Last Source Address Adjustment (DMA_TCD4_SLAST)	32	R/W	Undefined	28.3.23/ 588
4000_9090	TCD Destination Address (DMA_TCD4_DADDR)	32	R/W	Undefined	28.3.24/ 589
4000_9094	TCD Signed Destination Address Offset (DMA_TCD4_DOFF)	16	R/W	Undefined	28.3.25/ 589
4000_9096	TCD Current Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA_TCD4_CITER_ELINKYES)	16	R/W	Undefined	28.3.26/ 590
4000_9096	DMA_TCD4_CITER_ELINKNO	16	R/W	Undefined	28.3.27/ 591
4000_9098	TCD Last Destination Address Adjustment/Scatter Gather Address (DMA_TCD4_DLASTSGA)	32	R/W	Undefined	28.3.28/ 592
4000_909C	TCD Control and Status (DMA_TCD4_CSR)	16	R/W	Undefined	28.3.29/ 593
4000_909E	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA_TCD4_BITER_ELINKYES)	16	R/W	Undefined	28.3.30/ 595
4000_909E	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Disabled) (DMA_TCD4_BITER_ELINKNO)	16	R/W	Undefined	28.3.31/ 596
4000_90A0	TCD Source Address (DMA_TCD5_SADDR)	32	R/W	Undefined	28.3.17/ 583
4000_90A4	TCD Signed Source Address Offset (DMA_TCD5_SOFF)	16	R/W	Undefined	28.3.18/ 583
4000_90A6	TCD Transfer Attributes (DMA_TCD5_ATTR)	16	R/W	Undefined	28.3.19/ 584

Table continues on the next page...

DMA memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4000_90A8	TCD Minor Byte Count (Minor Loop Mapping Disabled) (DMA_TCD5_NBYTES_MLNO)	32	R/W	Undefined	28.3.20/ 585
4000_90A8	TCD Signed Minor Loop Offset (Minor Loop Mapping Enabled and Offset Disabled) (DMA_TCD5_NBYTES_MLOFFNO)	32	R/W	Undefined	28.3.21/ 586
4000_90A8	TCD Signed Minor Loop Offset (Minor Loop Mapping and Offset Enabled) (DMA_TCD5_NBYTES_MLOFFYES)	32	R/W	Undefined	28.3.22/ 587
4000_90AC	TCD Last Source Address Adjustment (DMA_TCD5_SLAST)	32	R/W	Undefined	28.3.23/ 588
4000_90B0	TCD Destination Address (DMA_TCD5_DADDR)	32	R/W	Undefined	28.3.24/ 589
4000_90B4	TCD Signed Destination Address Offset (DMA_TCD5_DOFF)	16	R/W	Undefined	28.3.25/ 589
4000_90B6	TCD Current Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA_TCD5_CITER_ELINKYES)	16	R/W	Undefined	28.3.26/ 590
4000_90B6	DMA_TCD5_CITER_ELINKNO	16	R/W	Undefined	28.3.27/ 591
4000_90B8	TCD Last Destination Address Adjustment/Scatter Gather Address (DMA_TCD5_DLASTSGA)	32	R/W	Undefined	28.3.28/ 592
4000_90BC	TCD Control and Status (DMA_TCD5_CSR)	16	R/W	Undefined	28.3.29/ 593
4000_90BE	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA_TCD5_BITER_ELINKYES)	16	R/W	Undefined	28.3.30/ 595
4000_90BE	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Disabled) (DMA_TCD5_BITER_ELINKNO)	16	R/W	Undefined	28.3.31/ 596
4000_90C0	TCD Source Address (DMA_TCD6_SADDR)	32	R/W	Undefined	28.3.17/ 583
4000_90C4	TCD Signed Source Address Offset (DMA_TCD6_SOFF)	16	R/W	Undefined	28.3.18/ 583
4000_90C6	TCD Transfer Attributes (DMA_TCD6_ATTR)	16	R/W	Undefined	28.3.19/ 584
4000_90C8	TCD Minor Byte Count (Minor Loop Mapping Disabled) (DMA_TCD6_NBYTES_MLNO)	32	R/W	Undefined	28.3.20/ 585
4000_90C8	TCD Signed Minor Loop Offset (Minor Loop Mapping Enabled and Offset Disabled) (DMA_TCD6_NBYTES_MLOFFNO)	32	R/W	Undefined	28.3.21/ 586
4000_90C8	TCD Signed Minor Loop Offset (Minor Loop Mapping and Offset Enabled) (DMA_TCD6_NBYTES_MLOFFYES)	32	R/W	Undefined	28.3.22/ 587
4000_90CC	TCD Last Source Address Adjustment (DMA_TCD6_SLAST)	32	R/W	Undefined	28.3.23/ 588
4000_90D0	TCD Destination Address (DMA_TCD6_DADDR)	32	R/W	Undefined	28.3.24/ 589
4000_90D4	TCD Signed Destination Address Offset (DMA_TCD6_DOFF)	16	R/W	Undefined	28.3.25/ 589

Table continues on the next page...

DMA memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4000_90D6	TCD Current Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA_TCD6_CITER_ELINKYES)	16	R/W	Undefined	28.3.26/ 590
4000_90D6	DMA_TCD6_CITER_ELINKNO	16	R/W	Undefined	28.3.27/ 591
4000_90D8	TCD Last Destination Address Adjustment/Scatter Gather Address (DMA_TCD6_DLASTSGA)	32	R/W	Undefined	28.3.28/ 592
4000_90DC	TCD Control and Status (DMA_TCD6_CSR)	16	R/W	Undefined	28.3.29/ 593
4000_90DE	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA_TCD6_BITER_ELINKYES)	16	R/W	Undefined	28.3.30/ 595
4000_90DE	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Disabled) (DMA_TCD6_BITER_ELINKNO)	16	R/W	Undefined	28.3.31/ 596
4000_90E0	TCD Source Address (DMA_TCD7_SADDR)	32	R/W	Undefined	28.3.17/ 583
4000_90E4	TCD Signed Source Address Offset (DMA_TCD7_SOFF)	16	R/W	Undefined	28.3.18/ 583
4000_90E6	TCD Transfer Attributes (DMA_TCD7_ATTR)	16	R/W	Undefined	28.3.19/ 584
4000_90E8	TCD Minor Byte Count (Minor Loop Mapping Disabled) (DMA_TCD7_NBYTES_MLNO)	32	R/W	Undefined	28.3.20/ 585
4000_90E8	TCD Signed Minor Loop Offset (Minor Loop Mapping Enabled and Offset Disabled) (DMA_TCD7_NBYTES_MLOFFNO)	32	R/W	Undefined	28.3.21/ 586
4000_90E8	TCD Signed Minor Loop Offset (Minor Loop Mapping and Offset Enabled) (DMA_TCD7_NBYTES_MLOFFYES)	32	R/W	Undefined	28.3.22/ 587
4000_90EC	TCD Last Source Address Adjustment (DMA_TCD7_SLAST)	32	R/W	Undefined	28.3.23/ 588
4000_90F0	TCD Destination Address (DMA_TCD7_DADDR)	32	R/W	Undefined	28.3.24/ 589
4000_90F4	TCD Signed Destination Address Offset (DMA_TCD7_DOFF)	16	R/W	Undefined	28.3.25/ 589
4000_90F6	TCD Current Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA_TCD7_CITER_ELINKYES)	16	R/W	Undefined	28.3.26/ 590
4000_90F6	DMA_TCD7_CITER_ELINKNO	16	R/W	Undefined	28.3.27/ 591
4000_90F8	TCD Last Destination Address Adjustment/Scatter Gather Address (DMA_TCD7_DLASTSGA)	32	R/W	Undefined	28.3.28/ 592
4000_90FC	TCD Control and Status (DMA_TCD7_CSR)	16	R/W	Undefined	28.3.29/ 593
4000_90FE	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA_TCD7_BITER_ELINKYES)	16	R/W	Undefined	28.3.30/ 595
4000_90FE	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Disabled) (DMA_TCD7_BITER_ELINKNO)	16	R/W	Undefined	28.3.31/ 596

Table continues on the next page...

DMA memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4000_9100	TCD Source Address (DMA_TCD8_SADDR)	32	R/W	Undefined	28.3.17/ 583
4000_9104	TCD Signed Source Address Offset (DMA_TCD8_SOFF)	16	R/W	Undefined	28.3.18/ 583
4000_9106	TCD Transfer Attributes (DMA_TCD8_ATTR)	16	R/W	Undefined	28.3.19/ 584
4000_9108	TCD Minor Byte Count (Minor Loop Mapping Disabled) (DMA_TCD8_NBYTES_MLNO)	32	R/W	Undefined	28.3.20/ 585
4000_9108	TCD Signed Minor Loop Offset (Minor Loop Mapping Enabled and Offset Disabled) (DMA_TCD8_NBYTES_MLOFFNO)	32	R/W	Undefined	28.3.21/ 586
4000_9108	TCD Signed Minor Loop Offset (Minor Loop Mapping and Offset Enabled) (DMA_TCD8_NBYTES_MLOFFYES)	32	R/W	Undefined	28.3.22/ 587
4000_910C	TCD Last Source Address Adjustment (DMA_TCD8_SLAST)	32	R/W	Undefined	28.3.23/ 588
4000_9110	TCD Destination Address (DMA_TCD8_DADDR)	32	R/W	Undefined	28.3.24/ 589
4000_9114	TCD Signed Destination Address Offset (DMA_TCD8_DOFF)	16	R/W	Undefined	28.3.25/ 589
4000_9116	TCD Current Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA_TCD8_CITER_ELINKYES)	16	R/W	Undefined	28.3.26/ 590
4000_9116	DMA_TCD8_CITER_ELINKNO	16	R/W	Undefined	28.3.27/ 591
4000_9118	TCD Last Destination Address Adjustment/Scatter Gather Address (DMA_TCD8_DLASTSGA)	32	R/W	Undefined	28.3.28/ 592
4000_911C	TCD Control and Status (DMA_TCD8_CSR)	16	R/W	Undefined	28.3.29/ 593
4000_911E	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA_TCD8_BITER_ELINKYES)	16	R/W	Undefined	28.3.30/ 595
4000_911E	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Disabled) (DMA_TCD8_BITER_ELINKNO)	16	R/W	Undefined	28.3.31/ 596
4000_9120	TCD Source Address (DMA_TCD9_SADDR)	32	R/W	Undefined	28.3.17/ 583
4000_9124	TCD Signed Source Address Offset (DMA_TCD9_SOFF)	16	R/W	Undefined	28.3.18/ 583
4000_9126	TCD Transfer Attributes (DMA_TCD9_ATTR)	16	R/W	Undefined	28.3.19/ 584
4000_9128	TCD Minor Byte Count (Minor Loop Mapping Disabled) (DMA_TCD9_NBYTES_MLNO)	32	R/W	Undefined	28.3.20/ 585
4000_9128	TCD Signed Minor Loop Offset (Minor Loop Mapping Enabled and Offset Disabled) (DMA_TCD9_NBYTES_MLOFFNO)	32	R/W	Undefined	28.3.21/ 586
4000_9128	TCD Signed Minor Loop Offset (Minor Loop Mapping and Offset Enabled) (DMA_TCD9_NBYTES_MLOFFYES)	32	R/W	Undefined	28.3.22/ 587

Table continues on the next page...

DMA memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4000_912C	TCD Last Source Address Adjustment (DMA_TCD9_SLAST)	32	R/W	Undefined	28.3.23/ 588
4000_9130	TCD Destination Address (DMA_TCD9_DADDR)	32	R/W	Undefined	28.3.24/ 589
4000_9134	TCD Signed Destination Address Offset (DMA_TCD9_DOFF)	16	R/W	Undefined	28.3.25/ 589
4000_9136	TCD Current Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA_TCD9_CITER_ELINKYES)	16	R/W	Undefined	28.3.26/ 590
4000_9136	DMA_TCD9_CITER_ELINKNO	16	R/W	Undefined	28.3.27/ 591
4000_9138	TCD Last Destination Address Adjustment/Scatter Gather Address (DMA_TCD9_DLASTSGA)	32	R/W	Undefined	28.3.28/ 592
4000_913C	TCD Control and Status (DMA_TCD9_CSR)	16	R/W	Undefined	28.3.29/ 593
4000_913E	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA_TCD9_BITER_ELINKYES)	16	R/W	Undefined	28.3.30/ 595
4000_913E	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Disabled) (DMA_TCD9_BITER_ELINKNO)	16	R/W	Undefined	28.3.31/ 596
4000_9140	TCD Source Address (DMA_TCD10_SADDR)	32	R/W	Undefined	28.3.17/ 583
4000_9144	TCD Signed Source Address Offset (DMA_TCD10_SOFF)	16	R/W	Undefined	28.3.18/ 583
4000_9146	TCD Transfer Attributes (DMA_TCD10_ATTR)	16	R/W	Undefined	28.3.19/ 584
4000_9148	TCD Minor Byte Count (Minor Loop Mapping Disabled) (DMA_TCD10_NBYTES_MLNO)	32	R/W	Undefined	28.3.20/ 585
4000_9148	TCD Signed Minor Loop Offset (Minor Loop Mapping Enabled and Offset Disabled) (DMA_TCD10_NBYTES_MLOFFNO)	32	R/W	Undefined	28.3.21/ 586
4000_9148	TCD Signed Minor Loop Offset (Minor Loop Mapping and Offset Enabled) (DMA_TCD10_NBYTES_MLOFFYES)	32	R/W	Undefined	28.3.22/ 587
4000_914C	TCD Last Source Address Adjustment (DMA_TCD10_SLAST)	32	R/W	Undefined	28.3.23/ 588
4000_9150	TCD Destination Address (DMA_TCD10_DADDR)	32	R/W	Undefined	28.3.24/ 589
4000_9154	TCD Signed Destination Address Offset (DMA_TCD10_DOFF)	16	R/W	Undefined	28.3.25/ 589
4000_9156	TCD Current Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA_TCD10_CITER_ELINKYES)	16	R/W	Undefined	28.3.26/ 590
4000_9156	DMA_TCD10_CITER_ELINKNO	16	R/W	Undefined	28.3.27/ 591
4000_9158	TCD Last Destination Address Adjustment/Scatter Gather Address (DMA_TCD10_DLASTSGA)	32	R/W	Undefined	28.3.28/ 592

Table continues on the next page...

DMA memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4000_915C	TCD Control and Status (DMA_TCD10_CSR)	16	R/W	Undefined	28.3.29/ 593
4000_915E	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA_TCD10_BITER_ELINKYES)	16	R/W	Undefined	28.3.30/ 595
4000_915E	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Disabled) (DMA_TCD10_BITER_ELINKNO)	16	R/W	Undefined	28.3.31/ 596
4000_9160	TCD Source Address (DMA_TCD11_SADDR)	32	R/W	Undefined	28.3.17/ 583
4000_9164	TCD Signed Source Address Offset (DMA_TCD11_SOFF)	16	R/W	Undefined	28.3.18/ 583
4000_9166	TCD Transfer Attributes (DMA_TCD11_ATTR)	16	R/W	Undefined	28.3.19/ 584
4000_9168	TCD Minor Byte Count (Minor Loop Mapping Disabled) (DMA_TCD11_NBYTES_MLNO)	32	R/W	Undefined	28.3.20/ 585
4000_9168	TCD Signed Minor Loop Offset (Minor Loop Mapping Enabled and Offset Disabled) (DMA_TCD11_NBYTES_MLOFFNO)	32	R/W	Undefined	28.3.21/ 586
4000_9168	TCD Signed Minor Loop Offset (Minor Loop Mapping and Offset Enabled) (DMA_TCD11_NBYTES_MLOFFYES)	32	R/W	Undefined	28.3.22/ 587
4000_916C	TCD Last Source Address Adjustment (DMA_TCD11_SLAST)	32	R/W	Undefined	28.3.23/ 588
4000_9170	TCD Destination Address (DMA_TCD11_DADDR)	32	R/W	Undefined	28.3.24/ 589
4000_9174	TCD Signed Destination Address Offset (DMA_TCD11_DOFF)	16	R/W	Undefined	28.3.25/ 589
4000_9176	TCD Current Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA_TCD11_CITER_ELINKYES)	16	R/W	Undefined	28.3.26/ 590
4000_9176	DMA_TCD11_CITER_ELINKNO	16	R/W	Undefined	28.3.27/ 591
4000_9178	TCD Last Destination Address Adjustment/Scatter Gather Address (DMA_TCD11_DLASTSGA)	32	R/W	Undefined	28.3.28/ 592
4000_917C	TCD Control and Status (DMA_TCD11_CSR)	16	R/W	Undefined	28.3.29/ 593
4000_917E	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA_TCD11_BITER_ELINKYES)	16	R/W	Undefined	28.3.30/ 595
4000_917E	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Disabled) (DMA_TCD11_BITER_ELINKNO)	16	R/W	Undefined	28.3.31/ 596
4000_9180	TCD Source Address (DMA_TCD12_SADDR)	32	R/W	Undefined	28.3.17/ 583
4000_9184	TCD Signed Source Address Offset (DMA_TCD12_SOFF)	16	R/W	Undefined	28.3.18/ 583

Table continues on the next page...

DMA memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4000_9186	TCD Transfer Attributes (DMA_TCD12_ATTR)	16	R/W	Undefined	28.3.19/ 584
4000_9188	TCD Minor Byte Count (Minor Loop Mapping Disabled) (DMA_TCD12_NBYTES_MLNO)	32	R/W	Undefined	28.3.20/ 585
4000_9188	TCD Signed Minor Loop Offset (Minor Loop Mapping Enabled and Offset Disabled) (DMA_TCD12_NBYTES_MLOFFNO)	32	R/W	Undefined	28.3.21/ 586
4000_9188	TCD Signed Minor Loop Offset (Minor Loop Mapping and Offset Enabled) (DMA_TCD12_NBYTES_MLOFFYES)	32	R/W	Undefined	28.3.22/ 587
4000_918C	TCD Last Source Address Adjustment (DMA_TCD12_SLAST)	32	R/W	Undefined	28.3.23/ 588
4000_9190	TCD Destination Address (DMA_TCD12_DADDR)	32	R/W	Undefined	28.3.24/ 589
4000_9194	TCD Signed Destination Address Offset (DMA_TCD12_DOFF)	16	R/W	Undefined	28.3.25/ 589
4000_9196	TCD Current Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA_TCD12_CITER_ELINKYES)	16	R/W	Undefined	28.3.26/ 590
4000_9196	DMA_TCD12_CITER_ELINKNO	16	R/W	Undefined	28.3.27/ 591
4000_9198	TCD Last Destination Address Adjustment/Scatter Gather Address (DMA_TCD12_DLASTSGA)	32	R/W	Undefined	28.3.28/ 592
4000_919C	TCD Control and Status (DMA_TCD12_CSR)	16	R/W	Undefined	28.3.29/ 593
4000_919E	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA_TCD12_BITER_ELINKYES)	16	R/W	Undefined	28.3.30/ 595
4000_919E	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Disabled) (DMA_TCD12_BITER_ELINKNO)	16	R/W	Undefined	28.3.31/ 596
4000_91A0	TCD Source Address (DMA_TCD13_SADDR)	32	R/W	Undefined	28.3.17/ 583
4000_91A4	TCD Signed Source Address Offset (DMA_TCD13_SOFF)	16	R/W	Undefined	28.3.18/ 583
4000_91A6	TCD Transfer Attributes (DMA_TCD13_ATTR)	16	R/W	Undefined	28.3.19/ 584
4000_91A8	TCD Minor Byte Count (Minor Loop Mapping Disabled) (DMA_TCD13_NBYTES_MLNO)	32	R/W	Undefined	28.3.20/ 585
4000_91A8	TCD Signed Minor Loop Offset (Minor Loop Mapping Enabled and Offset Disabled) (DMA_TCD13_NBYTES_MLOFFNO)	32	R/W	Undefined	28.3.21/ 586
4000_91A8	TCD Signed Minor Loop Offset (Minor Loop Mapping and Offset Enabled) (DMA_TCD13_NBYTES_MLOFFYES)	32	R/W	Undefined	28.3.22/ 587
4000_91AC	TCD Last Source Address Adjustment (DMA_TCD13_SLAST)	32	R/W	Undefined	28.3.23/ 588

Table continues on the next page...

DMA memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4000_91B0	TCD Destination Address (DMA_TCD13_DADDR)	32	R/W	Undefined	28.3.24/ 589
4000_91B4	TCD Signed Destination Address Offset (DMA_TCD13_DOFF)	16	R/W	Undefined	28.3.25/ 589
4000_91B6	TCD Current Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA_TCD13_CITER_ELINKYES)	16	R/W	Undefined	28.3.26/ 590
4000_91B6	DMA_TCD13_CITER_ELINKNO	16	R/W	Undefined	28.3.27/ 591
4000_91B8	TCD Last Destination Address Adjustment/Scatter Gather Address (DMA_TCD13_DLASTSGA)	32	R/W	Undefined	28.3.28/ 592
4000_91BC	TCD Control and Status (DMA_TCD13_CSR)	16	R/W	Undefined	28.3.29/ 593
4000_91BE	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA_TCD13_BITER_ELINKYES)	16	R/W	Undefined	28.3.30/ 595
4000_91BE	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Disabled) (DMA_TCD13_BITER_ELINKNO)	16	R/W	Undefined	28.3.31/ 596
4000_91C0	TCD Source Address (DMA_TCD14_SADDR)	32	R/W	Undefined	28.3.17/ 583
4000_91C4	TCD Signed Source Address Offset (DMA_TCD14_SOFF)	16	R/W	Undefined	28.3.18/ 583
4000_91C6	TCD Transfer Attributes (DMA_TCD14_ATTR)	16	R/W	Undefined	28.3.19/ 584
4000_91C8	TCD Minor Byte Count (Minor Loop Mapping Disabled) (DMA_TCD14_NBYTES_MLNO)	32	R/W	Undefined	28.3.20/ 585
4000_91C8	TCD Signed Minor Loop Offset (Minor Loop Mapping Enabled and Offset Disabled) (DMA_TCD14_NBYTES_MLOFFNO)	32	R/W	Undefined	28.3.21/ 586
4000_91C8	TCD Signed Minor Loop Offset (Minor Loop Mapping and Offset Enabled) (DMA_TCD14_NBYTES_MLOFFYES)	32	R/W	Undefined	28.3.22/ 587
4000_91CC	TCD Last Source Address Adjustment (DMA_TCD14_SLAST)	32	R/W	Undefined	28.3.23/ 588
4000_91D0	TCD Destination Address (DMA_TCD14_DADDR)	32	R/W	Undefined	28.3.24/ 589
4000_91D4	TCD Signed Destination Address Offset (DMA_TCD14_DOFF)	16	R/W	Undefined	28.3.25/ 589
4000_91D6	TCD Current Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA_TCD14_CITER_ELINKYES)	16	R/W	Undefined	28.3.26/ 590
4000_91D6	DMA_TCD14_CITER_ELINKNO	16	R/W	Undefined	28.3.27/ 591
4000_91D8	TCD Last Destination Address Adjustment/Scatter Gather Address (DMA_TCD14_DLASTSGA)	32	R/W	Undefined	28.3.28/ 592
4000_91DC	TCD Control and Status (DMA_TCD14_CSR)	16	R/W	Undefined	28.3.29/ 593

Table continues on the next page...

DMA memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4000_91DE	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA_TCD14_BITER_ELINKYES)	16	R/W	Undefined	28.3.30/ 595
4000_91DE	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Disabled) (DMA_TCD14_BITER_ELINKNO)	16	R/W	Undefined	28.3.31/ 596
4000_91E0	TCD Source Address (DMA_TCD15_SADDR)	32	R/W	Undefined	28.3.17/ 583
4000_91E4	TCD Signed Source Address Offset (DMA_TCD15_SOFF)	16	R/W	Undefined	28.3.18/ 583
4000_91E6	TCD Transfer Attributes (DMA_TCD15_ATTR)	16	R/W	Undefined	28.3.19/ 584
4000_91E8	TCD Minor Byte Count (Minor Loop Mapping Disabled) (DMA_TCD15_NBYTES_MLNO)	32	R/W	Undefined	28.3.20/ 585
4000_91E8	TCD Signed Minor Loop Offset (Minor Loop Mapping Enabled and Offset Disabled) (DMA_TCD15_NBYTES_MLOFFNO)	32	R/W	Undefined	28.3.21/ 586
4000_91E8	TCD Signed Minor Loop Offset (Minor Loop Mapping and Offset Enabled) (DMA_TCD15_NBYTES_MLOFFYES)	32	R/W	Undefined	28.3.22/ 587
4000_91EC	TCD Last Source Address Adjustment (DMA_TCD15_SLAST)	32	R/W	Undefined	28.3.23/ 588
4000_91F0	TCD Destination Address (DMA_TCD15_DADDR)	32	R/W	Undefined	28.3.24/ 589
4000_91F4	TCD Signed Destination Address Offset (DMA_TCD15_DOFF)	16	R/W	Undefined	28.3.25/ 589
4000_91F6	TCD Current Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA_TCD15_CITER_ELINKYES)	16	R/W	Undefined	28.3.26/ 590
4000_91F6	DMA_TCD15_CITER_ELINKNO	16	R/W	Undefined	28.3.27/ 591
4000_91F8	TCD Last Destination Address Adjustment/Scatter Gather Address (DMA_TCD15_DLASTSGA)	32	R/W	Undefined	28.3.28/ 592
4000_91FC	TCD Control and Status (DMA_TCD15_CSR)	16	R/W	Undefined	28.3.29/ 593
4000_91FE	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA_TCD15_BITER_ELINKYES)	16	R/W	Undefined	28.3.30/ 595
4000_91FE	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Disabled) (DMA_TCD15_BITER_ELINKNO)	16	R/W	Undefined	28.3.31/ 596

28.3.1 Control Register (DMA_CR)

The CR defines the basic operating configuration of the DMA.

Arbitration can be configured to use either a fixed-priority or a round-robin scheme. For fixed-priority arbitration, the highest priority channel requesting service is selected to execute. The channel priority registers assign the priorities; see the DCHPRIn registers. For round-robin arbitration, the channel priorities are ignored and channels are cycled through (from high to low channel number) without regard to priority.

NOTE

> For correct operation, writes to the CR register must be performed only when the DMA channels are inactive; that is, when TCDn_CSR[ACTIVE] bits are cleared.

Minor loop offsets are address offset values added to the final source address (TCDn_SADDR) or destination address (TCDn_DADDR) upon minor loop completion. When minor loop offsets are enabled, the minor loop offset (MLOFF) is added to the final source address (TCDn_SADDR), to the final destination address (TCDn_DADDR), or to both prior to the addresses being written back into the TCD. If the major loop is complete, the minor loop offset is ignored and the major loop address offsets (TCDn_SLAST and TCDn_DLAST_SGA) are used to compute the next TCDn_SADDR and TCDn_DADDR values.

When minor loop mapping is enabled (EMLM is 1), TCDn word2 is redefined. A portion of TCDn word2 is used to specify multiple fields: a source enable bit (SMLOE) to specify the minor loop offset should be applied to the source address (TCDn_SADDR) upon minor loop completion, a destination enable bit (DMLOE) to specify the minor loop offset should be applied to the destination address (TCDn_DADDR) upon minor loop completion, and the sign extended minor loop offset value (MLOFF). The same offset value (MLOFF) is used for both source and destination minor loop offsets. When either minor loop offset is enabled (SMLOE set or DMLOE set), the NBYTES field is reduced to 10 bits. When both minor loop offsets are disabled (SMLOE cleared and DMLOE cleared), the NBYTES field is a 30-bit vector.

When minor loop mapping is disabled (EMLM is 0), all 32 bits of TCDn word2 are assigned to the NBYTES field.

Address: 4000_8000h base + 0h offset = 4000_8000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0														CX	ECX
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								EMLM	CLM	HALT	HOE	Reserved	ERCA	EDBG	Reserved
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DMA_CR field descriptions

Field	Description
31–18 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
17 CX	Cancel Transfer 0 Normal operation 1 Cancel the remaining data transfer. Stop the executing channel and force the minor loop to finish. The cancel takes effect after the last write of the current read/write sequence. The CX bit clears itself after the cancel has been honored. This cancel retires the channel normally as if the minor loop was completed.
16 ECX	Error Cancel Transfer 0 Normal operation 1 Cancel the remaining data transfer in the same fashion as the CX bit. Stop the executing channel and force the minor loop to finish. The cancel takes effect after the last write of the current read/write sequence. The ECX bit clears itself after the cancel is honored. In addition to cancelling the transfer, ECX treats the cancel as an error condition, thus updating the Error Status register (DMAx_ES) and generating an optional error interrupt.
15–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7 EMLM	Enable Minor Loop Mapping 0 Disabled. TCDn.word2 is defined as a 32-bit NBYTES field. 1 Enabled. TCDn.word2 is redefined to include individual enable fields, an offset field, and the NBYTES field. The individual enable fields allow the minor loop offset to be applied to the source address, the destination address, or both. The NBYTES field is reduced when either offset is enabled.
6 CLM	Continuous Link Mode **NOTE:** Do not use continuous link mode with a channel linking to itself if there is only one minor loop iteration per service request, e.g., if the channel's NBYTES value is the same as either the source or destination size. The same data transfer profile can be achieved by simply increasing the NBYTES value, which provides more efficient, faster processing. 0 A minor loop channel link made to itself goes through channel arbitration before being activated again. 1 A minor loop channel link made to itself does not go through channel arbitration before being activated again. Upon minor loop completion, the channel activates again if that channel has a minor loop channel link enabled and the link channel is itself. This effectively applies the minor loop offsets and restarts the next minor loop.
5 HALT	Halt DMA Operations 0 Normal operation 1 Stall the start of any new channels. Executing channels are allowed to complete. Channel execution resumes when this bit is cleared.

Table continues on the next page...

DMA_CR field descriptions (continued)

Field	Description
4 HOE	Halt On Error 0 Normal operation 1 Any error causes the HALT bit to set. Subsequently, all service requests are ignored until the HALT bit is cleared.
3 Reserved	This field is reserved. Reserved
2 ERCA	Enable Round Robin Channel Arbitration 0 Fixed priority arbitration is used for channel selection . 1 Round robin arbitration is used for channel selection .
1 EDBG	Enable Debug 0 When in debug mode, the DMA continues to operate. 1 When in debug mode, the DMA stalls the start of a new channel. Executing channels are allowed to complete. Channel execution resumes when the system exits debug mode or the EDBG bit is cleared.
0 Reserved	This field is reserved. Reserved

28.3.2 Error Status Register (DMA_ES)

The ES provides information concerning the last recorded channel error. Channel errors can be caused by:

- A configuration error, that is:
  - An illegal setting in the transfer-control descriptor, or
  - An illegal priority register setting in fixed-arbitration
- An error termination to a bus master read or write cycle
- A cancel transfer with error bit that will be set when a transfer is canceled via the corresponding cancel transfer control bit

See the Error Reporting and Handling section for more details.

Address: 4000_8000h base + 4h offset = 4000_8004h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	VLD	0														ECX
W	▨	▨														▨
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	CPE	0		ERRCHN				SAE	SOE	DAE	DOE	NCE	SGE	SBE	DBE
W	▨	▨	▨		▨				▨	▨	▨	▨	▨	▨	▨	▨
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DMA_ES field descriptions

Field	Description
31 VLD	Logical OR of all ERR status bits 0 No ERR bits are set. 1 At least one ERR bit is set indicating a valid error exists that has not been cleared.
30–17 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
16 ECX	Transfer Canceled 0 No canceled transfers 1 The last recorded entry was a canceled transfer by the error cancel transfer input
15 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
14 CPE	Channel Priority Error 0 No channel priority error 1 The last recorded error was a configuration error in the channel priorities . Channel priorities are not unique.
13–12 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
11–8 ERRCHN	Error Channel Number or Canceled Channel Number The channel number of the last recorded error, excluding CPE errors, or last recorded error canceled transfer.
7 SAE	Source Address Error 0 No source address configuration error. 1 The last recorded error was a configuration error detected in the TCDn_SADDR field. TCDn_SADDR is inconsistent with TCDn_ATTR[SSIZE].
6 SOE	Source Offset Error 0 No source offset configuration error 1 The last recorded error was a configuration error detected in the TCDn_SOFF field. TCDn_SOFF is inconsistent with TCDn_ATTR[SSIZE].
5 DAE	Destination Address Error 0 No destination address configuration error 1 The last recorded error was a configuration error detected in the TCDn_DADDR field. TCDn_DADDR is inconsistent with TCDn_ATTR[DSIZE].
4 DOE	Destination Offset Error 0 No destination offset configuration error 1 The last recorded error was a configuration error detected in the TCDn_DOFF field. TCDn_DOFF is inconsistent with TCDn_ATTR[DSIZE].
3 NCE	NBYTES/CITER Configuration Error 0 No NBYTES/CITER configuration error 1 The last recorded error was a configuration error detected in the TCDn_NBYTES or TCDn_CITER fields. • TCDn_NBYTES is not a multiple of TCDn_ATTR[SSIZE] and TCDn_ATTR[DSIZE], or

Table continues on the next page...

DMA_ES field descriptions (continued)

Field	Description
	• TCDn_CITER[CITER] is equal to zero, or • TCDn_CITER[ELINK] is not equal to TCDn_BITER[ELINK]
2 SGE	Scatter/Gather Configuration Error 0 No scatter/gather configuration error 1 The last recorded error was a configuration error detected in the TCDn_DLASTSGA field. This field is checked at the beginning of a scatter/gather operation after major loop completion if TCDn_CSR[ESG] is enabled. TCDn_DLASTSGA is not on a 32 byte boundary.
1 SBE	Source Bus Error 0 No source bus error 1 The last recorded error was a bus error on a source read
0 DBE	Destination Bus Error 0 No destination bus error 1 The last recorded error was a bus error on a destination write

28.3.3 Enable Request Register (DMA_ERQ)

The ERQ register provides a bit map for the 16 channels to enable the request signal for each channel. The state of any given channel enable is directly affected by writes to this register; it is also affected by writes to the SERQ and CERQ registers. These registers are provided so the request enable for a single channel can easily be modified without needing to perform a read-modify-write sequence to the ERQ.

DMA request input signals and this enable request flag must be asserted before a channel's hardware service request is accepted. The state of the DMA enable request flag does not affect a channel service request made explicitly through software or a linked channel request.

Address: 4000_8000h base + Ch offset = 4000_800Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ERQ15	ERQ14	ERQ13	ERQ12	ERQ11	ERQ10	ERQ9	ERQ8	ERQ7	ERQ6	ERQ5	ERQ4	ERQ3	ERQ2	ERQ1	ERQ0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DMA_ERQ field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
15 ERQ15	Enable DMA Request 15 0 The DMA request signal for the corresponding channel is disabled 1 The DMA request signal for the corresponding channel is enabled
14 ERQ14	Enable DMA Request 14 0 The DMA request signal for the corresponding channel is disabled 1 The DMA request signal for the corresponding channel is enabled
13 ERQ13	Enable DMA Request 13 0 The DMA request signal for the corresponding channel is disabled 1 The DMA request signal for the corresponding channel is enabled
12 ERQ12	Enable DMA Request 12 0 The DMA request signal for the corresponding channel is disabled 1 The DMA request signal for the corresponding channel is enabled
11 ERQ11	Enable DMA Request 11 0 The DMA request signal for the corresponding channel is disabled 1 The DMA request signal for the corresponding channel is enabled
10 ERQ10	Enable DMA Request 10 0 The DMA request signal for the corresponding channel is disabled 1 The DMA request signal for the corresponding channel is enabled
9 ERQ9	Enable DMA Request 9 0 The DMA request signal for the corresponding channel is disabled 1 The DMA request signal for the corresponding channel is enabled
8 ERQ8	Enable DMA Request 8 0 The DMA request signal for the corresponding channel is disabled 1 The DMA request signal for the corresponding channel is enabled
7 ERQ7	Enable DMA Request 7 0 The DMA request signal for the corresponding channel is disabled 1 The DMA request signal for the corresponding channel is enabled
6 ERQ6	Enable DMA Request 6 0 The DMA request signal for the corresponding channel is disabled 1 The DMA request signal for the corresponding channel is enabled
5 ERQ5	Enable DMA Request 5 0 The DMA request signal for the corresponding channel is disabled 1 The DMA request signal for the corresponding channel is enabled
4 ERQ4	Enable DMA Request 4

Table continues on the next page...

DMA_ERQ field descriptions (continued)

Field	Description
	0 The DMA request signal for the corresponding channel is disabled 1 The DMA request signal for the corresponding channel is enabled
3 ERQ3	Enable DMA Request 3 0 The DMA request signal for the corresponding channel is disabled 1 The DMA request signal for the corresponding channel is enabled
2 ERQ2	Enable DMA Request 2 0 The DMA request signal for the corresponding channel is disabled 1 The DMA request signal for the corresponding channel is enabled
1 ERQ1	Enable DMA Request 1 0 The DMA request signal for the corresponding channel is disabled 1 The DMA request signal for the corresponding channel is enabled
0 ERQ0	Enable DMA Request 0 0 The DMA request signal for the corresponding channel is disabled 1 The DMA request signal for the corresponding channel is enabled

28.3.4 Enable Error Interrupt Register (DMA_EEI)

The EEI register provides a bit map for the 16 channels to enable the error interrupt signal for each channel. The state of any given channel's error interrupt enable is directly affected by writes to this register; it is also affected by writes to the SEEI and CEEI. These registers are provided so that the error interrupt enable for a single channel can easily be modified without the need to perform a read-modify-write sequence to the EEI register.

The DMA error indicator and the error interrupt enable flag must be asserted before an error interrupt request for a given channel is asserted to the interrupt controller.

Address: 4000_8000h base + 14h offset = 4000_8014h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	EEI15	EEI14	EEI13	EEI12	EEI11	EEI10	EEI9	EEI8	EEI7	EEI6	EEI5	EEI4	EEI3	EEI2	EEI1	EEI0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DMA_EEI field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
15 EEI15	Enable Error Interrupt 15 0 The error signal for corresponding channel does not generate an error interrupt 1 The assertion of the error signal for corresponding channel generates an error interrupt request
14 EEI14	Enable Error Interrupt 14 0 The error signal for corresponding channel does not generate an error interrupt 1 The assertion of the error signal for corresponding channel generates an error interrupt request
13 EEI13	Enable Error Interrupt 13 0 The error signal for corresponding channel does not generate an error interrupt 1 The assertion of the error signal for corresponding channel generates an error interrupt request
12 EEI12	Enable Error Interrupt 12 0 The error signal for corresponding channel does not generate an error interrupt 1 The assertion of the error signal for corresponding channel generates an error interrupt request
11 EEI11	Enable Error Interrupt 11 0 The error signal for corresponding channel does not generate an error interrupt 1 The assertion of the error signal for corresponding channel generates an error interrupt request
10 EEI10	Enable Error Interrupt 10 0 The error signal for corresponding channel does not generate an error interrupt 1 The assertion of the error signal for corresponding channel generates an error interrupt request
9 EEI9	Enable Error Interrupt 9 0 The error signal for corresponding channel does not generate an error interrupt 1 The assertion of the error signal for corresponding channel generates an error interrupt request
8 EEI8	Enable Error Interrupt 8 0 The error signal for corresponding channel does not generate an error interrupt 1 The assertion of the error signal for corresponding channel generates an error interrupt request
7 EEI7	Enable Error Interrupt 7 0 The error signal for corresponding channel does not generate an error interrupt 1 The assertion of the error signal for corresponding channel generates an error interrupt request
6 EEI6	Enable Error Interrupt 6

Table continues on the next page...

DMA_EEI field descriptions (continued)

Field	Description
	0 The error signal for corresponding channel does not generate an error interrupt 1 The assertion of the error signal for corresponding channel generates an error interrupt request
5 EEI5	Enable Error Interrupt 5 0 The error signal for corresponding channel does not generate an error interrupt 1 The assertion of the error signal for corresponding channel generates an error interrupt request
4 EEI4	Enable Error Interrupt 4 0 The error signal for corresponding channel does not generate an error interrupt 1 The assertion of the error signal for corresponding channel generates an error interrupt request
3 EEI3	Enable Error Interrupt 3 0 The error signal for corresponding channel does not generate an error interrupt 1 The assertion of the error signal for corresponding channel generates an error interrupt request
2 EEI2	Enable Error Interrupt 2 0 The error signal for corresponding channel does not generate an error interrupt 1 The assertion of the error signal for corresponding channel generates an error interrupt request
1 EEI1	Enable Error Interrupt 1 0 The error signal for corresponding channel does not generate an error interrupt 1 The assertion of the error signal for corresponding channel generates an error interrupt request
0 EEI0	Enable Error Interrupt 0 0 The error signal for corresponding channel does not generate an error interrupt 1 The assertion of the error signal for corresponding channel generates an error interrupt request

28.3.5 Clear Enable Error Interrupt Register (DMA_CEEI)

The CEEI provides a simple memory-mapped mechanism to clear a given bit in the EEI to disable the error interrupt for a given channel. The data value on a register write causes the corresponding bit in the EEI to be cleared. Setting the CAEE bit provides a global clear function, forcing the EEI contents to be cleared, disabling all DMA request inputs. If the NOP bit is set, the command is ignored. This allows you to write multiple-byte registers as a 32-bit word. Reads of this register return all zeroes.

Address: 4000_8000h base + 18h offset = 4000_8018h

Bit	7	6	5	4	3	2	1	0
Read	0	0			0			
Write	NOP	CAEE	0		CEEI			
Reset	0	0	0	0	0	0	0	0

DMA_CEEI field descriptions

Field	Description
7 NOP	No Op enable 0 Normal operation 1 No operation, ignore the other bits in this register
6 CAEE	Clear All Enable Error Interrupts 0 Clear only the EEI bit specified in the CEEI field 1 Clear all bits in EEI
5–4 Reserved	This field is reserved.
CEEI	Clear Enable Error Interrupt Clears the corresponding bit in EEI

28.3.6 Set Enable Error Interrupt Register (DMA_SEEI)

The SEEI provides a simple memory-mapped mechanism to set a given bit in the EEI to enable the error interrupt for a given channel. The data value on a register write causes the corresponding bit in the EEI to be set. Setting the SAEE bit provides a global set function, forcing the entire EEI contents to be set. If the NOP bit is set, the command is ignored. This allows you to write multiple-byte registers as a 32-bit word. Reads of this register return all zeroes.

Address: 4000_8000h base + 19h offset = 4000_8019h

Bit	7	6	5	4	3	2	1	0
Read	0	0			0			
Write	NOP	SAEE	0		SEEI			
Reset	0	0	0	0	0	0	0	0

DMA_SEEI field descriptions

Field	Description
7 NOP	No Op enable 0 Normal operation 1 No operation, ignore the other bits in this register
6 SAEE	Sets All Enable Error Interrupts 0 Set only the EEI bit specified in the SEEI field. 1 Sets all bits in EEI
5–4 Reserved	This field is reserved.

Table continues on the next page...

DMA_SEEI field descriptions (continued)

Field	Description
SEEI	Set Enable Error Interrupt Sets the corresponding bit in EEI

28.3.7 Clear Enable Request Register (DMA_CERQ)

The CERQ provides a simple memory-mapped mechanism to clear a given bit in the ERQ to disable the DMA request for a given channel. The data value on a register write causes the corresponding bit in the ERQ to be cleared. Setting the CAER bit provides a global clear function, forcing the entire contents of the ERQ to be cleared, disabling all DMA request inputs. If NOP is set, the command is ignored. This allows you to write multiple-byte registers as a 32-bit word. Reads of this register return all zeroes.

Address: 4000_8000h base + 1Ah offset = 4000_801Ah

Bit	7	6	5	4	3	2	1	0
Read	0	0			0			
Write	NOP	CAER	0		CERQ			
Reset	0	0	0	0	0	0	0	0

DMA_CERQ field descriptions

Field	Description
7 NOP	No Op enable 0 Normal operation 1 No operation, ignore the other bits in this register
6 CAER	Clear All Enable Requests 0 Clear only the ERQ bit specified in the CERQ field 1 Clear all bits in ERQ
5–4 Reserved	This field is reserved.
CERQ	Clear Enable Request Clears the corresponding bit in ERQ.

28.3.8 Set Enable Request Register (DMA_SERQ)

The SERQ provides a simple memory-mapped mechanism to set a given bit in the ERQ to enable the DMA request for a given channel. The data value on a register write causes the corresponding bit in the ERQ to be set. Setting the SAER bit provides a global set function, forcing the entire contents of ERQ to be set. If the NOP bit is set, the command is ignored. This allows you to write multiple-byte registers as a 32-bit word. Reads of this register return all zeroes.

Address: 4000_8000h base + 1Bh offset = 4000_801Bh

Bit	7	6	5	4	3	2	1	0
Read	0	0			0			
Write	NOP	SAER	0		SERQ			
Reset	0	0	0	0	0	0	0	0

DMA_SERQ field descriptions

Field	Description
7 NOP	No Op enable 0 Normal operation 1 No operation, ignore the other bits in this register
6 SAER	Set All Enable Requests 0 Set only the ERQ bit specified in the SERQ field 1 Set all bits in ERQ
5–4 Reserved	This field is reserved.
SERQ	Set Enable Request Sets the corresponding bit in ERQ.

28.3.9 Clear DONE Status Bit Register (DMA_CDNE)

The CDNE provides a simple memory-mapped mechanism to clear the DONE bit in the TCD of the given channel. The data value on a register write causes the DONE bit in the corresponding transfer control descriptor to be cleared. Setting the CADN bit provides a global clear function, forcing all DONE bits to be cleared. If the NOP bit is set, the command is ignored. This allows you to write multiple-byte registers as a 32-bit word. Reads of this register return all zeroes.

Address: 4000_8000h base + 1Ch offset = 4000_801Ch

Bit	7	6	5	4	3	2	1	0
Read	0	0			0			
Write	NOP	CADN	0		CDNE			
Reset	0	0	0	0	0	0	0	0

DMA_CDNE field descriptions

Field	Description
7 NOP	No Op enable 0 Normal operation 1 No operation, ignore the other bits in this register
6 CADN	Clears All DONE Bits 0 Clears only the TCDn_CSR[DONE] bit specified in the CDNE field 1 Clears all bits in TCDn_CSR[DONE]
5–4 Reserved	This field is reserved.
CDNE	Clear DONE Bit Clears the corresponding bit in TCDn_CSR[DONE]

28.3.10 Set START Bit Register (DMA_SSRT)

The SSRT provides a simple memory-mapped mechanism to set the START bit in the TCD of the given channel. The data value on a register write causes the START bit in the corresponding transfer control descriptor to be set. Setting the SAST bit provides a global set function, forcing all START bits to be set. If the NOP bit is set, the command is ignored. This allows you to write multiple-byte registers as a 32-bit word. Reads of this register return all zeroes.

Address: 4000_8000h base + 1Dh offset = 4000_801Dh

Bit	7	6	5	4	3	2	1	0
Read	0	0			0			
Write	NOP	SAST	0		SSRT			
Reset	0	0	0	0	0	0	0	0

DMA_SSRT field descriptions

Field	Description
7 NOP	No Op enable 0 Normal operation 1 No operation, ignore the other bits in this register
6 SAST	Set All START Bits (activates all channels) 0 Set only the TCDn_CSR[START] bit specified in the SSRT field 1 Set all bits in TCDn_CSR[START]
5–4 Reserved	This field is reserved.
SSRT	Set START Bit Sets the corresponding bit in TCDn_CSR[START]

28.3.11 Clear Error Register (DMA_CERR)

The CERR provides a simple memory-mapped mechanism to clear a given bit in the ERR to disable the error condition flag for a given channel. The given value on a register write causes the corresponding bit in the ERR to be cleared. Setting the CAEI bit provides a global clear function, forcing the ERR contents to be cleared, clearing all channel error indicators. If the NOP bit is set, the command is ignored. This allows you to write multiple-byte registers as a 32-bit word. Reads of this register return all zeroes.

Address: 4000_8000h base + 1Eh offset = 4000_801Eh

Bit	7	6	5	4	3	2	1	0
Read	0	0			0			
Write	NOP	CAEI	0		CERR			
Reset	0	0	0	0	0	0	0	0

DMA_CERR field descriptions

Field	Description
7 NOP	No Op enable 0 Normal operation 1 No operation, ignore the other bits in this register
6 CAEI	Clear All Error Indicators 0 Clear only the ERR bit specified in the CERR field 1 Clear all bits in ERR
5–4 Reserved	This field is reserved.
CERR	Clear Error Indicator Clears the corresponding bit in ERR

28.3.12 Clear Interrupt Request Register (DMA_CINT)

The CINT provides a simple, memory-mapped mechanism to clear a given bit in the INT to disable the interrupt request for a given channel. The given value on a register write causes the corresponding bit in the INT to be cleared. Setting the CAIR bit provides a global clear function, forcing the entire contents of the INT to be cleared, disabling all DMA interrupt requests. If the NOP bit is set, the command is ignored. This allows you to write multiple-byte registers as a 32-bit word. Reads of this register return all zeroes.

Address: 4000_8000h base + 1Fh offset = 4000_801Fh

Bit	7	6	5	4	3	2	1	0
Read	0	0			0			
Write	NOP	CAIR	0		CINT			
Reset	0	0	0	0	0	0	0	0

DMA_CINT field descriptions

Field	Description
7 NOP	No Op enable 0 Normal operation 1 No operation, ignore the other bits in this register
6 CAIR	Clear All Interrupt Requests 0 Clear only the INT bit specified in the CINT field 1 Clear all bits in INT
5–4 Reserved	This field is reserved.
CINT	Clear Interrupt Request Clears the corresponding bit in INT

28.3.13 Interrupt Request Register (DMA_INT)

The INT register provides a bit map for the 16 channels signaling the presence of an interrupt request for each channel. Depending on the appropriate bit setting in the transfer-control descriptors, the eDMA engine generates an interrupt on data transfer completion. The outputs of this register are directly routed to the interrupt controller. During the interrupt-service routine associated with any given channel, it is the software's responsibility to clear the appropriate bit, negating the interrupt request. Typically, a write to the CINT register in the interrupt service routine is used for this purpose.

The state of any given channel's interrupt request is directly affected by writes to this register; it is also affected by writes to the CINT register. On writes to INT, a 1 in any bit position clears the corresponding channel's interrupt request. A zero in any bit position has no affect on the corresponding channel's current interrupt status. The CINT register is provided so the interrupt request for a single channel can easily be cleared without the need to perform a read-modify-write sequence to the INT register.

Address: 4000_8000h base + 24h offset = 4000_8024h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	INT15	INT14	INT13	INT12	INT11	INT10	INT9	INT8	INT7	INT6	INT5	INT4	INT3	INT2	INT1	INT0
W	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DMA_INT field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

Table continues on the next page...

DMA_INT field descriptions (continued)

Field	Description
15 INT15	Interrupt Request 15 0 The interrupt request for corresponding channel is cleared 1 The interrupt request for corresponding channel is active
14 INT14	Interrupt Request 14 0 The interrupt request for corresponding channel is cleared 1 The interrupt request for corresponding channel is active
13 INT13	Interrupt Request 13 0 The interrupt request for corresponding channel is cleared 1 The interrupt request for corresponding channel is active
12 INT12	Interrupt Request 12 0 The interrupt request for corresponding channel is cleared 1 The interrupt request for corresponding channel is active
11 INT11	Interrupt Request 11 0 The interrupt request for corresponding channel is cleared 1 The interrupt request for corresponding channel is active
10 INT10	Interrupt Request 10 0 The interrupt request for corresponding channel is cleared 1 The interrupt request for corresponding channel is active
9 INT9	Interrupt Request 9 0 The interrupt request for corresponding channel is cleared 1 The interrupt request for corresponding channel is active
8 INT8	Interrupt Request 8 0 The interrupt request for corresponding channel is cleared 1 The interrupt request for corresponding channel is active
7 INT7	Interrupt Request 7 0 The interrupt request for corresponding channel is cleared 1 The interrupt request for corresponding channel is active
6 INT6	Interrupt Request 6 0 The interrupt request for corresponding channel is cleared 1 The interrupt request for corresponding channel is active
5 INT5	Interrupt Request 5 0 The interrupt request for corresponding channel is cleared 1 The interrupt request for corresponding channel is active
4 INT4	Interrupt Request 4 0 The interrupt request for corresponding channel is cleared 1 The interrupt request for corresponding channel is active

Table continues on the next page...

DMA_INT field descriptions (continued)

Field	Description
3 INT3	Interrupt Request 3 0 The interrupt request for corresponding channel is cleared 1 The interrupt request for corresponding channel is active
2 INT2	Interrupt Request 2 0 The interrupt request for corresponding channel is cleared 1 The interrupt request for corresponding channel is active
1 INT1	Interrupt Request 1 0 The interrupt request for corresponding channel is cleared 1 The interrupt request for corresponding channel is active
0 INT0	Interrupt Request 0 0 The interrupt request for corresponding channel is cleared 1 The interrupt request for corresponding channel is active

28.3.14 Error Register (DMA_ERR)

The ERR provides a bit map for the 16 channels, signaling the presence of an error for each channel. The eDMA engine signals the occurrence of an error condition by setting the appropriate bit in this register. The outputs of this register are enabled by the contents of the EEI, and then routed to the interrupt controller. During the execution of the interrupt-service routine associated with any DMA errors, it is software's responsibility to clear the appropriate bit, negating the error-interrupt request. Typically, a write to the CERR in the interrupt-service routine is used for this purpose. The normal DMA channel completion indicators (setting the transfer control descriptor DONE flag and the possible assertion of an interrupt request) are not affected when an error is detected.

The contents of this register can also be polled because a non-zero value indicates the presence of a channel error regardless of the state of the EEI. The state of any given channel's error indicators is affected by writes to this register; it is also affected by writes to the CERR. On writes to the ERR, a one in any bit position clears the corresponding channel's error status. A zero in any bit position has no affect on the corresponding channel's current error status. The CERR is provided so the error indicator for a single channel can easily be cleared.

Address: 4000_8000h base + 2Ch offset = 4000_802Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	ERR15	ERR14	ERR13	ERR12	ERR11	ERR10	ERR9	ERR8	ERR7	ERR6	ERR5	ERR4	ERR3	ERR2	ERR1	ERR0
W	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DMA_ERR field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
15 ERR15	Error In Channel 15 0 An error in this channel has not occurred 1 An error in this channel has occurred
14 ERR14	Error In Channel 14 0 An error in this channel has not occurred 1 An error in this channel has occurred
13 ERR13	Error In Channel 13 0 An error in this channel has not occurred 1 An error in this channel has occurred
12 ERR12	Error In Channel 12 0 An error in this channel has not occurred 1 An error in this channel has occurred
11 ERR11	Error In Channel 11 0 An error in this channel has not occurred 1 An error in this channel has occurred
10 ERR10	Error In Channel 10 0 An error in this channel has not occurred 1 An error in this channel has occurred

Table continues on the next page...

DMA_ERR field descriptions (continued)

Field	Description
9 ERR9	Error In Channel 9 0 An error in this channel has not occurred 1 An error in this channel has occurred
8 ERR8	Error In Channel 8 0 An error in this channel has not occurred 1 An error in this channel has occurred
7 ERR7	Error In Channel 7 0 An error in this channel has not occurred 1 An error in this channel has occurred
6 ERR6	Error In Channel 6 0 An error in this channel has not occurred 1 An error in this channel has occurred
5 ERR5	Error In Channel 5 0 An error in this channel has not occurred 1 An error in this channel has occurred
4 ERR4	Error In Channel 4 0 An error in this channel has not occurred 1 An error in this channel has occurred
3 ERR3	Error In Channel 3 0 An error in this channel has not occurred 1 An error in this channel has occurred
2 ERR2	Error In Channel 2 0 An error in this channel has not occurred 1 An error in this channel has occurred
1 ERR1	Error In Channel 1 0 An error in this channel has not occurred 1 An error in this channel has occurred
0 ERR0	Error In Channel 0 0 An error in this channel has not occurred 1 An error in this channel has occurred

28.3.15 Hardware Request Status Register (DMA_HRS)

The HRS register provides a bit map for the DMA channels, signaling the presence of a hardware request for each channel. The hardware request status bits reflect the current state of the register and qualified (via the ERQ fields) DMA request signals as seen by the DMA's arbitration logic. This view into the hardware request signals may be used for debug purposes.

NOTE

These bits reflect the state of the request as seen by the arbitration logic. Therefore, this status is affected by the ERQ bits.

Address: 4000_8000h base + 34h offset = 4000_8034h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	HRS15	HRS14	HRS13	HRS12	HRS11	HRS10	HRS9	HRS8	HRS7	HRS6	HRS5	HRS4	HRS3	HRS2	HRS1	HRS0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DMA_HRS field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
15 HRS15	Hardware Request Status Channel 15

Table continues on the next page...

DMA_HRS field descriptions (continued)

Field	Description
	The HRS bit for its respective channel remains asserted for the period when a Hardware Request is Present on the Channel. After the Request is completed and Channel is free, the HRS bit is automatically cleared by hardware. 0 A hardware service request for channel 15 is not present 1 A hardware service request for channel 15 is present
14 HRS14	Hardware Request Status Channel 14 The HRS bit for its respective channel remains asserted for the period when a Hardware Request is Present on the Channel. After the Request is completed and Channel is free, the HRS bit is automatically cleared by hardware. 0 A hardware service request for channel 14 is not present 1 A hardware service request for channel 14 is present
13 HRS13	Hardware Request Status Channel 13 The HRS bit for its respective channel remains asserted for the period when a Hardware Request is Present on the Channel. After the Request is completed and Channel is free, the HRS bit is automatically cleared by hardware. 0 A hardware service request for channel 13 is not present 1 A hardware service request for channel 13 is present
12 HRS12	Hardware Request Status Channel 12 The HRS bit for its respective channel remains asserted for the period when a Hardware Request is Present on the Channel. After the Request is completed and Channel is free, the HRS bit is automatically cleared by hardware. 0 A hardware service request for channel 12 is not present 1 A hardware service request for channel 12 is present
11 HRS11	Hardware Request Status Channel 11 The HRS bit for its respective channel remains asserted for the period when a Hardware Request is Present on the Channel. After the Request is completed and Channel is free, the HRS bit is automatically cleared by hardware. 0 A hardware service request for channel 11 is not present 1 A hardware service request for channel 11 is present
10 HRS10	Hardware Request Status Channel 10 The HRS bit for its respective channel remains asserted for the period when a Hardware Request is Present on the Channel. After the Request is completed and Channel is free, the HRS bit is automatically cleared by hardware. 0 A hardware service request for channel 10 is not present 1 A hardware service request for channel 10 is present
9 HRS9	Hardware Request Status Channel 9 The HRS bit for its respective channel remains asserted for the period when a Hardware Request is Present on the Channel. After the Request is completed and Channel is free, the HRS bit is automatically cleared by hardware.

Table continues on the next page...

DMA_HRS field descriptions (continued)

Field	Description
	0 A hardware service request for channel 9 is not present 1 A hardware service request for channel 9 is present
8 HRS8	Hardware Request Status Channel 8 The HRS bit for its respective channel remains asserted for the period when a Hardware Request is Present on the Channel. After the Request is completed and Channel is free, the HRS bit is automatically cleared by hardware. 0 A hardware service request for channel 8 is not present 1 A hardware service request for channel 8 is present
7 HRS7	Hardware Request Status Channel 7 The HRS bit for its respective channel remains asserted for the period when a Hardware Request is Present on the Channel. After the Request is completed and Channel is free, the HRS bit is automatically cleared by hardware. 0 A hardware service request for channel 7 is not present 1 A hardware service request for channel 7 is present
6 HRS6	Hardware Request Status Channel 6 The HRS bit for its respective channel remains asserted for the period when a Hardware Request is Present on the Channel. After the Request is completed and Channel is free, the HRS bit is automatically cleared by hardware. 0 A hardware service request for channel 6 is not present 1 A hardware service request for channel 6 is present
5 HRS5	Hardware Request Status Channel 5 The HRS bit for its respective channel remains asserted for the period when a Hardware Request is Present on the Channel. After the Request is completed and Channel is free, the HRS bit is automatically cleared by hardware. 0 A hardware service request for channel 5 is not present 1 A hardware service request for channel 5 is present
4 HRS4	Hardware Request Status Channel 4 The HRS bit for its respective channel remains asserted for the period when a Hardware Request is Present on the Channel. After the Request is completed and Channel is free, the HRS bit is automatically cleared by hardware. 0 A hardware service request for channel 4 is not present 1 A hardware service request for channel 4 is present
3 HRS3	Hardware Request Status Channel 3 The HRS bit for its respective channel remains asserted for the period when a Hardware Request is Present on the Channel. After the Request is completed and Channel is free, the HRS bit is automatically cleared by hardware. 0 A hardware service request for channel 3 is not present 1 A hardware service request for channel 3 is present
2 HRS2	Hardware Request Status Channel 2

Table continues on the next page...

DMA_HRS field descriptions (continued)

Field	Description
	The HRS bit for its respective channel remains asserted for the period when a Hardware Request is Present on the Channel. After the Request is completed and Channel is free, the HRS bit is automatically cleared by hardware. 0 A hardware service request for channel 2 is not present 1 A hardware service request for channel 2 is present
1 HRS1	Hardware Request Status Channel 1 The HRS bit for its respective channel remains asserted for the period when a Hardware Request is Present on the Channel. After the Request is completed and Channel is free, the HRS bit is automatically cleared by hardware. 0 A hardware service request for channel 1 is not present 1 A hardware service request for channel 1 is present
0 HRS0	Hardware Request Status Channel 0 The HRS bit for its respective channel remains asserted for the period when a Hardware Request is Present on the Channel. After the Request is completed and Channel is free, the HRS bit is automatically cleared by hardware. 0 A hardware service request for channel 0 is not present 1 A hardware service request for channel 0 is present

28.3.16 Channel n Priority Register (DMA_DCHPRI*n*)

When fixed-priority channel arbitration is enabled (CR[ERCA] = 0), the contents of these registers define the unique priorities associated with each channel. The channel priorities are evaluated by numeric value; for example, 0 is the lowest priority, 1 is the next higher priority, then 2, 3, etc. Software must program the channel priorities with unique values; otherwise, a configuration error is reported. The range of the priority value is limited to the values of 0 through 15.

Address: 4000_8000h base + 100h offset + (1d × i), where i=0d to 15d

Bit	7	6	5	4	3	2	1	0
Read	ECP	DPA	0		CHPRI			
Write								
Reset	0	0	0	0	*	*	*	*

* Notes:
- CHPRI field: See bit field description.

DMA_DCHPRI*n* field descriptions

Field	Description
7 ECP	Enable Channel Preemption. 0 Channel n cannot be suspended by a higher priority channel's service request. 1 Channel n can be temporarily suspended by the service request of a higher priority channel.
6 DPA	Disable Preempt Ability. 0 Channel n can suspend a lower priority channel. 1 Channel n cannot suspend any channel, regardless of channel priority.
5–4 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
CHPRI	Channel n Arbitration Priority Channel priority when fixed-priority arbitration is enabled **NOTE:** Reset value for the channel priority field, CHPRI, is equal to the corresponding channel number for each priority register, that is, DCHPRI15[CHPRI] = 0b1111.

28.3.17 TCD Source Address (DMA_TCD*n*_SADDR)

Address: 4000_8000h base + 1000h offset + (32d × i), where i=0d to 15d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SADDR																															
Reset	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*

* Notes:

- x = Undefined at reset.

DMA_TCD*n*_SADDR field descriptions

Field	Description
SADDR	Source Address Memory address pointing to the source data.

28.3.18 TCD Signed Source Address Offset (DMA_TCD*n*_SOFF)

Address: 4000_8000h base + 1004h offset + (32d × i), where i=0d to 15d

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read Write	SOFF															
Reset	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*

* Notes:

- x = Undefined at reset.

DMA_TCD*n*_SOFF field descriptions

Field	Description
SOFF	Source address signed offset Sign-extended offset applied to the current source address to form the next-state value as each source read is completed.

28.3.19 TCD Transfer Attributes (DMA_TCD*n*_ATTR)

Address: 4000_8000h base + 1006h offset + (32d × i), where i=0d to 15d

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read Write	SMOD					SSIZE			DMOD					DSIZE		
Reset	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*

* Notes:

- x = Undefined at reset.

DMA_TCD*n*_ATTR field descriptions

Field	Description
15–11 SMOD	Source Address Modulo 0 Source address modulo feature is disabled ≠0 This value defines a specific address range specified to be the value after SADDR + SOFF calculation is performed on the original register value. Setting this field provides the ability to implement a circular data queue easily. For data queues requiring power-of-2 size bytes, the queue should start at a 0-modulo-size address and the SMOD field should be set to the appropriate value for the queue, freezing the desired number of upper address bits. The value programmed into this field specifies the number of lower address bits allowed to change. For a circular queue application, the SOFF is typically set to the transfer size to implement post-increment addressing with the SMOD function constraining the addresses to a 0-modulo-size range.
10–8 SSIZE	Source data transfer size **NOTE:** Using a Reserved value causes a configuration error. 000 8-bit 001 16-bit 010 32-bit 011 Reserved 100 16-byte burst 101 32-byte burst 110 Reserved 111 Reserved
7–3 DMOD	Destination Address Modulo See the SMOD definition
DSIZE	Destination data transfer size

Table continues on the next page...

DMA_TCD*n*_ATTR field descriptions (continued)

Field	Description
	See the SSIZE definition

28.3.20 TCD Minor Byte Count (Minor Loop Mapping Disabled) (DMA_TCD*n*_NBYTES_MLNO)

This register, or one of the next two registers (TCD_NBYTES_MLOFFNO, TCD_NBYTES_MLOFFYES), defines the number of bytes to transfer per request. Which register to use depends on whether minor loop mapping is disabled, enabled but not used for this channel, or enabled and used.

TCD word 2 is defined as follows if:

- Minor loop mapping is disabled (CR[EMLM] = 0)

If minor loop mapping is enabled, see the TCD_NBYTES_MLOFFNO and TCD_NBYTES_MLOFFYES register descriptions for the definition of TCD word 2.

Address: 4000_8000h base + 1008h offset + (32d × i), where i=0d to 15d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	NBYTES																															
Reset	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*

* Notes:

- x = Undefined at reset.

DMA_TCD*n*_NBYTES_MLNO field descriptions

Field	Description
NBYTES	Minor Byte Transfer Count Number of bytes to be transferred in each service request of the channel. As a channel activates, the appropriate TCD contents load into the eDMA engine, and the appropriate reads and writes perform until the minor byte transfer count has transferred. This is an indivisible operation and cannot be halted. It can, however, be stalled by using the bandwidth control field, or via preemption. After the minor count is exhausted, the SADDR and DADDR values are written back into the TCD memory, the major iteration count is decremented and restored to the TCD memory. If the major iteration count is completed, additional processing is performed. **NOTE:** An NBYTES value of 0x0000_0000 is interpreted as a 4 GB transfer.

28.3.21 TCD Signed Minor Loop Offset (Minor Loop Mapping Enabled and Offset Disabled) (DMA_TCD*n*_NBYTES_MLOFFNO)

One of three registers (this register, TCD_NBYTES_MLNO, or TCD_NBYTES_MLOFFYES), defines the number of bytes to transfer per request. Which register to use depends on whether minor loop mapping is disabled, enabled but not used for this channel, or enabled and used.

TCD word 2 is defined as follows if:

- Minor loop mapping is enabled (CR[EMLM] = 1) and
- SMLOE = 0 and DMLOE = 0

If minor loop mapping is enabled and SMLOE or DMLOE is set, then refer to the TCD_NBYTES_MLOFFYES register description. If minor loop mapping is disabled, then refer to the TCD_NBYTES_MLNO register description.

Address: 4000_8000h base + 1008h offset + (32d × i), where i=0d to 15d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SMLOE	DMLOE	NBYTES													
Reset	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	NBYTES															
Reset	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*

* Notes:

- x = Undefined at reset.

DMA_TCD*n*_NBYTES_MLOFFNO field descriptions

Field	Description
31 SMLOE	Source Minor Loop Offset Enable Selects whether the minor loop offset is applied to the source address upon minor loop completion. 0 The minor loop offset is not applied to the SADDR 1 The minor loop offset is applied to the SADDR
30 DMLOE	Destination Minor Loop Offset enable Selects whether the minor loop offset is applied to the destination address upon minor loop completion.

Table continues on the next page...

DMA_TCD*n*_NBYTES_MLOFFNO field descriptions (continued)

Field	Description
	0 The minor loop offset is not applied to the DADDR 1 The minor loop offset is applied to the DADDR
NBYTES	Minor Byte Transfer Count Number of bytes to be transferred in each service request of the channel. As a channel activates, the appropriate TCD contents load into the eDMA engine, and the appropriate reads and writes perform until the minor byte transfer count has transferred. This is an indivisible operation and cannot be halted. It can, however, be stalled by using the bandwidth control field, or via preemption. After the minor count is exhausted, the SADDR and DADDR values are written back into the TCD memory, the major iteration count is decremented and restored to the TCD memory. If the major iteration count is completed, additional processing is performed.

28.3.22 TCD Signed Minor Loop Offset (Minor Loop Mapping and Offset Enabled) (DMA_TCD*n*_NBYTES_MLOFFYES)

One of three registers (this register, TCD_NBYTES_MLNO, or TCD_NBYTES_MLOFFNO), defines the number of bytes to transfer per request. Which register to use depends on whether minor loop mapping is disabled, enabled but not used for this channel, or enabled and used.

TCD word 2 is defined as follows if:

- Minor loop mapping is enabled (CR[EMLM] = 1) and
- Minor loop offset is enabled (SMLOE or DMLOE = 1)

If minor loop mapping is enabled and SMLOE and DMLOE are cleared, then refer to the TCD_NBYTES_MLOFFNO register description. If minor loop mapping is disabled, then refer to the TCD_NBYTES_MLNO register description.

Address: 4000_8000h base + 1008h offset + (32d × i), where i=0d to 15d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SMLOE	DMLOE	MLOFF													
Reset	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	MLOFF						NBYTES									
Reset	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*

* Notes:

- x = Undefined at reset.

DMA_TCD*n*_NBYTES_MLOFFYES field descriptions

Field	Description
31 SMLOE	Source Minor Loop Offset Enable Selects whether the minor loop offset is applied to the source address upon minor loop completion. 0 The minor loop offset is not applied to the SADDR 1 The minor loop offset is applied to the SADDR
30 DMLOE	Destination Minor Loop Offset enable Selects whether the minor loop offset is applied to the destination address upon minor loop completion. 0 The minor loop offset is not applied to the DADDR 1 The minor loop offset is applied to the DADDR
29–10 MLOFF	If SMLOE or DMLOE is set, this field represents a sign-extended offset applied to the source or destination address to form the next-state value after the minor loop completes.
NBYTES	Minor Byte Transfer Count Number of bytes to be transferred in each service request of the channel. As a channel activates, the appropriate TCD contents load into the eDMA engine, and the appropriate reads and writes perform until the minor byte transfer count has transferred. This is an indivisible operation and cannot be halted. It can, however, be stalled by using the bandwidth control field, or via preemption. After the minor count is exhausted, the SADDR and DADDR values are written back into the TCD memory, the major iteration count is decremented and restored to the TCD memory. If the major iteration count is completed, additional processing is performed.

28.3.23 TCD Last Source Address Adjustment (DMA_TCD*n*_SLAST)

Address: 4000_8000h base + 100Ch offset + (32d × i), where i=0d to 15d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SLAST																															
Reset	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*

* Notes:

- x = Undefined at reset.

DMA_TCD*n*_SLAST field descriptions

Field	Description
SLAST	Last Source Address Adjustment Adjustment value added to the source address at the completion of the major iteration count. This value can be applied to restore the source address to the initial value, or adjust the address to reference the next data structure. This register uses two's complement notation; the overflow bit is discarded.

28.3.24 TCD Destination Address (DMA_TCD*n*_DADDR)

Address: 4000_8000h base + 1010h offset + (32d × i), where i=0d to 15d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DADDR																															
Reset	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*

* Notes:
- x = Undefined at reset.

DMA_TCD*n*_DADDR field descriptions

Field	Description
DADDR	Destination Address Memory address pointing to the destination data.

28.3.25 TCD Signed Destination Address Offset (DMA_TCD*n*_DOFF)

Address: 4000_8000h base + 1014h offset + (32d × i), where i=0d to 15d

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read Write	DOFF															
Reset	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*

* Notes:
- x = Undefined at reset.

DMA_TCD*n*_DOFF field descriptions

Field	Description
DOFF	Destination Address Signed Offset Sign-extended offset applied to the current destination address to form the next-state value as each destination write is completed.

28.3.26 TCD Current Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA_TCD*n*_CITER_ELINKYES)

If TCDn_CITER[ELINK] is set, the TCDn_CITER register is defined as follows.

Address: 4000_8000h base + 1016h offset + (32d × i), where i=0d to 15d

Bit	15	14	13	12	11	10	9	8
Read	ELINK			LINKCH				CITER
Write		0						
Reset	x*	x*	x*	x*	x*	x*	x*	x*

Bit	7	6	5	4	3	2	1	0
Read Write	CITER							
Reset	x*	x*	x*	x*	x*	x*	x*	x*

* Notes:
- x = Undefined at reset.

DMA_TCD*n*_CITER_ELINKYES field descriptions

Field	Description
15 ELINK	Enable channel-to-channel linking on minor-loop complete As the channel completes the minor loop, this flag enables linking to another channel, defined by the LINKCH field. The link target channel initiates a channel service request via an internal mechanism that sets the TCDn_CSR[START] bit of the specified channel. If channel linking is disabled, the CITER value is extended to 15 bits in place of a link channel number. If the major loop is exhausted, this link mechanism is suppressed in favor of the MAJORELINK channel linking. **NOTE:** This bit must be equal to the BITER[ELINK] bit; otherwise, a configuration error is reported. 0 The channel-to-channel linking is disabled 1 The channel-to-channel linking is enabled
14–13 Reserved	This field is reserved.
12–9 LINKCH	Minor Loop Link Channel Number If channel-to-channel linking is enabled (ELINK = 1), then after the minor loop is exhausted, the eDMA engine initiates a channel service request to the channel defined by this field by setting that channel's TCDn_CSR[START] bit.
CITER	Current Major Iteration Count This 9-bit (ELINK = 1) or 15-bit (ELINK = 0) count represents the current major loop count for the channel. It is decremented each time the minor loop is completed and updated in the transfer control descriptor memory. After the major iteration count is exhausted, the channel performs a number of operations, for example, final source and destination address calculations, optionally generating an interrupt to signal channel completion before reloading the CITER field from the Beginning Iteration Count (BITER) field.

Table continues on the next page...

DMA_TCD*n*_CITER_ELINKYES field descriptions (continued)

Field	Description
	NOTE: When the CITER field is initially loaded by software, it must be set to the same value as that contained in the BITER field. **NOTE:** If the channel is configured to execute a single service request, the initial values of BITER and CITER should be 0x0001.

28.3.27 TCD Current Minor Loop Link, Major Loop Count (Channel Linking Disabled) (DMA_TCD*n*_CITER_ELINKNO)

If TCDn_CITER[ELINK] is cleared, the TCDn_CITER register is defined as follows.

Address: 4000_8000h base + 1016h offset + (32d × i), where i=0d to 15d

Bit	15	14	13	12	11	10	9	8
Read Write	ELINK	CITER						
Reset	x*	x*	x*	x*	x*	x*	x*	x*

Bit	7	6	5	4	3	2	1	0
Read Write	CITER							
Reset	x*	x*	x*	x*	x*	x*	x*	x*

* Notes:
- x = Undefined at reset.

DMA_TCD*n*_CITER_ELINKNO field descriptions

Field	Description
15 ELINK	Enable channel-to-channel linking on minor-loop complete As the channel completes the minor loop, this flag enables linking to another channel, defined by the LINKCH field. The link target channel initiates a channel service request via an internal mechanism that sets the TCDn_CSR[START] bit of the specified channel. If channel linking is disabled, the CITER value is extended to 15 bits in place of a link channel number. If the major loop is exhausted, this link mechanism is suppressed in favor of the MAJORELINK channel linking. **NOTE:** This bit must be equal to the BITER[ELINK] bit; otherwise, a configuration error is reported. 0 The channel-to-channel linking is disabled 1 The channel-to-channel linking is enabled
CITER	Current Major Iteration Count This 9-bit (ELINK = 1) or 15-bit (ELINK = 0) count represents the current major loop count for the channel. It is decremented each time the minor loop is completed and updated in the transfer control descriptor memory. After the major iteration count is exhausted, the channel performs a number of operations, for

Table continues on the next page...

DMA_TCD*n*_CITER_ELINKNO field descriptions (continued)

Field	Description
	example, final source and destination address calculations, optionally generating an interrupt to signal channel completion before reloading the CITER field from the Beginning Iteration Count (BITER) field. **NOTE:** When the CITER field is initially loaded by software, it must be set to the same value as that contained in the BITER field. **NOTE:** If the channel is configured to execute a single service request, the initial values of BITER and CITER should be 0x0001.

28.3.28 TCD Last Destination Address Adjustment/Scatter Gather Address (DMA_TCD*n*_DLASTSGA)

Address: 4000_8000h base + 1018h offset + (32d × i), where i=0d to 15d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DLASTSGA																															
Reset	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*

* Notes:

- x = Undefined at reset.

DMA_TCD*n*_DLASTSGA field descriptions

Field	Description
DLASTSGA	Destination last address adjustment or the memory address for the next transfer control descriptor to be loaded into this channel (scatter/gather). If (TCDn_CSR[ESG] = 0) then: • Adjustment value added to the destination address at the completion of the major iteration count. This value can apply to restore the destination address to the initial value or adjust the address to reference the next data structure. • This field uses two's complement notation for the final destination address adjustment. Otherwise: • This address points to the beginning of a 0-modulo-32-byte region containing the next transfer control descriptor to be loaded into this channel. This channel reload is performed as the major iteration count completes. The scatter/gather address must be 0-modulo-32-byte, otherwise a configuration error is reported.

28.3.29 TCD Control and Status (DMA_TCD*n*_CSR)

Address: 4000_8000h base + 101Ch offset + (32d × i), where i=0d to 15d

<table>
<tr><td>Bit</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td>Read</td><td colspan="2" rowspan="2">BWC</td><td colspan="2">▨</td><td colspan="4" rowspan="2">MAJORLINKCH</td></tr>
<tr><td>Write</td><td colspan="2">0</td></tr>
<tr><td>Reset</td><td>x*</td><td>x*</td><td>x*</td><td>x*</td><td>x*</td><td>x*</td><td>x*</td><td>x*</td></tr>
</table>

<table>
<tr><td>Bit</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>Read</td><td rowspan="2">DONE</td><td>ACTIVE</td><td rowspan="2">MAJORELI NK</td><td rowspan="2">ESG</td><td rowspan="2">DREQ</td><td rowspan="2">INTHALF</td><td rowspan="2">INTMAJOR</td><td rowspan="2">START</td></tr>
<tr><td>Write</td><td>▨</td></tr>
<tr><td>Reset</td><td>x*</td><td>x*</td><td>x*</td><td>x*</td><td>x*</td><td>x*</td><td>x*</td><td>x*</td></tr>
</table>

* Notes:
- x = Undefined at reset.

DMA_TCD*n*_CSR field descriptions

Field	Description
15–14 BWC	Bandwidth Control Throttles the amount of bus bandwidth consumed by the eDMA. Generally, as the eDMA processes the minor loop, it continuously generates read/write sequences until the minor count is exhausted. This field forces the eDMA to stall after the completion of each read/write access to control the bus request bandwidth seen by the crossbar switch. **NOTE:** If the source and destination sizes are equal, this field is ignored between the first and second transfers and after the last write of each minor loop. This behavior is a side effect of reducing start-up latency. 00 No eDMA engine stalls. 10 eDMA engine stalls for 4 cycles after each R/W. 11 eDMA engine stalls for 8 cycles after each R/W.
13–12 Reserved	This field is reserved.
11–8 MAJORLINKCH	Major Loop Link Channel Number If (MAJORELINK = 0) then: • No channel-to-channel linking, or chaining, is performed after the major loop counter is exhausted. Otherwise: • After the major loop counter is exhausted, the eDMA engine initiates a channel service request at the channel defined by this field by setting that channel's TCDn_CSR[START] bit.
7 DONE	Channel Done This flag indicates the eDMA has completed the major loop. The eDMA engine sets it as the CITER count reaches zero. The software clears it, or the hardware when the channel is activated. **NOTE:** This bit must be cleared to write the MAJORELINK or ESG bits.

Table continues on the next page...

DMA_TCD*n*_CSR field descriptions (continued)

Field	Description
6 ACTIVE	Channel Active This flag signals the channel is currently in execution. It is set when channel service begins, and is cleared by the eDMA as the minor loop completes or when any error condition is detected.
5 MAJORELINK	Enable channel-to-channel linking on major loop complete As the channel completes the major loop, this flag enables the linking to another channel, defined by MAJORLINKCH. The link target channel initiates a channel service request via an internal mechanism that sets the TCDn_CSR[START] bit of the specified channel. **NOTE:** To support the dynamic linking coherency model, this field is forced to zero when written to while the TCDn_CSR[DONE] bit is set. 0 The channel-to-channel linking is disabled. 1 The channel-to-channel linking is enabled.
4 ESG	Enable Scatter/Gather Processing As the channel completes the major loop, this flag enables scatter/gather processing in the current channel. If enabled, the eDMA engine uses DLASTSGA as a memory pointer to a 0-modulo-32 address containing a 32-byte data structure loaded as the transfer control descriptor into the local memory. **NOTE:** To support the dynamic scatter/gather coherency model, this field is forced to zero when written to while the TCDn_CSR[DONE] bit is set. 0 The current channel’s TCD is normal format. 1 The current channel’s TCD specifies a scatter gather format. The DLASTSGA field provides a memory pointer to the next TCD to be loaded into this channel after the major loop completes its execution.
3 DREQ	Disable Request If this flag is set, the eDMA hardware automatically clears the corresponding ERQ bit when the current major iteration count reaches zero. 0 The channel’s ERQ bit is not affected. 1 The channel’s ERQ bit is cleared when the major loop is complete.
2 INTHALF	Enable an interrupt when major counter is half complete. If this flag is set, the channel generates an interrupt request by setting the appropriate bit in the INT register when the current major iteration count reaches the halfway point. Specifically, the comparison performed by the eDMA engine is (CITER == (BITER >> 1)). This halfway point interrupt request is provided to support double-buffered, also known as ping-pong, schemes or other types of data movement where the processor needs an early indication of the transfer’s progress. **NOTE:** If BITER = 1, do not use INTHALF. Use INTMAJOR instead. 0 The half-point interrupt is disabled. 1 The half-point interrupt is enabled.
1 INTMAJOR	Enable an interrupt when major iteration count completes. If this flag is set, the channel generates an interrupt request by setting the appropriate bit in the INT when the current major iteration count reaches zero. 0 The end-of-major loop interrupt is disabled. 1 The end-of-major loop interrupt is enabled.

Table continues on the next page...

DMA_TCD*n*_CSR field descriptions (continued)

Field	Description
0 START	Channel Start If this flag is set, the channel is requesting service. The eDMA hardware automatically clears this flag after the channel begins execution. 0 The channel is not explicitly started. 1 The channel is explicitly started via a software initiated service request.

28.3.30 TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA_TCD*n*_BITER_ELINKYES)

If the TCDn_BITER[ELINK] bit is set, the TCDn_BITER register is defined as follows.

Address: 4000_8000h base + 101Eh offset + (32d × i), where i=0d to 15d

<table>
<tr><td>Bit</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td>Read
Write</td><td>ELINK</td><td colspan="2">0</td><td colspan="4">LINKCH</td><td>BITER</td></tr>
<tr><td>Reset</td><td>x*</td><td>x*</td><td>x*</td><td>x*</td><td>x*</td><td>x*</td><td>x*</td><td>x*</td></tr>
</table>

<table>
<tr><td>Bit</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>Read
Write</td><td colspan="8">BITER</td></tr>
<tr><td>Reset</td><td>x*</td><td>x*</td><td>x*</td><td>x*</td><td>x*</td><td>x*</td><td>x*</td><td>x*</td></tr>
</table>

* Notes:

- x = Undefined at reset.

DMA_TCD*n*_BITER_ELINKYES field descriptions

Field	Description
15 ELINK	Enables channel-to-channel linking on minor loop complete As the channel completes the minor loop, this flag enables the linking to another channel, defined by BITER[LINKCH]. The link target channel initiates a channel service request via an internal mechanism that sets the TCDn_CSR[START] bit of the specified channel. If channel linking disables, the BITER value extends to 15 bits in place of a link channel number. If the major loop is exhausted, this link mechanism is suppressed in favor of the MAJORELINK channel linking. **NOTE:** When the software loads the TCD, this field must be set equal to the corresponding CITER field; otherwise, a configuration error is reported. As the major iteration count is exhausted, the contents of this field are reloaded into the CITER field. 0 The channel-to-channel linking is disabled 1 The channel-to-channel linking is enabled
14–13 Reserved	This field is reserved.
12–9 LINKCH	Link Channel Number

Table continues on the next page...

DMA_TCD*n*_BITER_ELINKYES field descriptions (continued)

Field	Description
	If channel-to-channel linking is enabled (ELINK = 1), then after the minor loop is exhausted, the eDMA engine initiates a channel service request at the channel defined by this field by setting that channel's TCDn_CSR[START] bit. **NOTE:** When the software loads the TCD, this field must be set equal to the corresponding CITER field; otherwise, a configuration error is reported. As the major iteration count is exhausted, the contents of this field are reloaded into the CITER field.
BITER	Starting major iteration count As the transfer control descriptor is first loaded by software, this 9-bit (ELINK = 1) or 15-bit (ELINK = 0) field must be equal to the value in the CITER field. As the major iteration count is exhausted, the contents of this field are reloaded into the CITER field. **NOTE:** When the software loads the TCD, this field must be set equal to the corresponding CITER field; otherwise, a configuration error is reported. As the major iteration count is exhausted, the contents of this field are reloaded into the CITER field. If the channel is configured to execute a single service request, the initial values of BITER and CITER should be 0x0001.

28.3.31 TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Disabled) (DMA_TCD*n*_BITER_ELINKNO)

If the TCDn_BITER[ELINK] bit is cleared, the TCDn_BITER register is defined as follows.

Address: 4000_8000h base + 101Eh offset + (32d × i), where i=0d to 15d

Bit	15	14	13	12	11	10	9	8
Read Write	ELINK	BITER						
Reset	x*	x*	x*	x*	x*	x*	x*	x*

Bit	7	6	5	4	3	2	1	0
Read Write	BITER							
Reset	x*	x*	x*	x*	x*	x*	x*	x*

* Notes:

- x = Undefined at reset.

DMA_TCD*n*_BITER_ELINKNO field descriptions

Field	Description
15 ELINK	Enables channel-to-channel linking on minor loop complete As the channel completes the minor loop, this flag enables the linking to another channel, defined by BITER[LINKCH]. The link target channel initiates a channel service request via an internal mechanism that sets the TCDn_CSR[START] bit of the specified channel. If channel linking is disabled, the BITER value extends to 15 bits in place of a link channel number. If the major loop is exhausted, this link mechanism is suppressed in favor of the MAJORELINK channel linking.

Table continues on the next page...

DMA_TCD*n*_BITER_ELINKNO field descriptions (continued)

Field	Description
	NOTE: When the software loads the TCD, this field must be set equal to the corresponding CITER field; otherwise, a configuration error is reported. As the major iteration count is exhausted, the contents of this field are reloaded into the CITER field. 0 The channel-to-channel linking is disabled 1 The channel-to-channel linking is enabled
BITER	Starting Major Iteration Count As the transfer control descriptor is first loaded by software, this 9-bit (ELINK = 1) or 15-bit (ELINK = 0) field must be equal to the value in the CITER field. As the major iteration count is exhausted, the contents of this field are reloaded into the CITER field. **NOTE:** When the software loads the TCD, this field must be set equal to the corresponding CITER field; otherwise, a configuration error is reported. As the major iteration count is exhausted, the contents of this field is reloaded into the CITER field. If the channel is configured to execute a single service request, the initial values of BITER and CITER should be 0x0001.

28.4 Functional description

The operation of the eDMA is described in the following subsections.

28.4.1 eDMA basic data flow

The basic flow of a data transfer can be partitioned into three segments.

As shown in the following diagram, the first segment involves the channel activation:

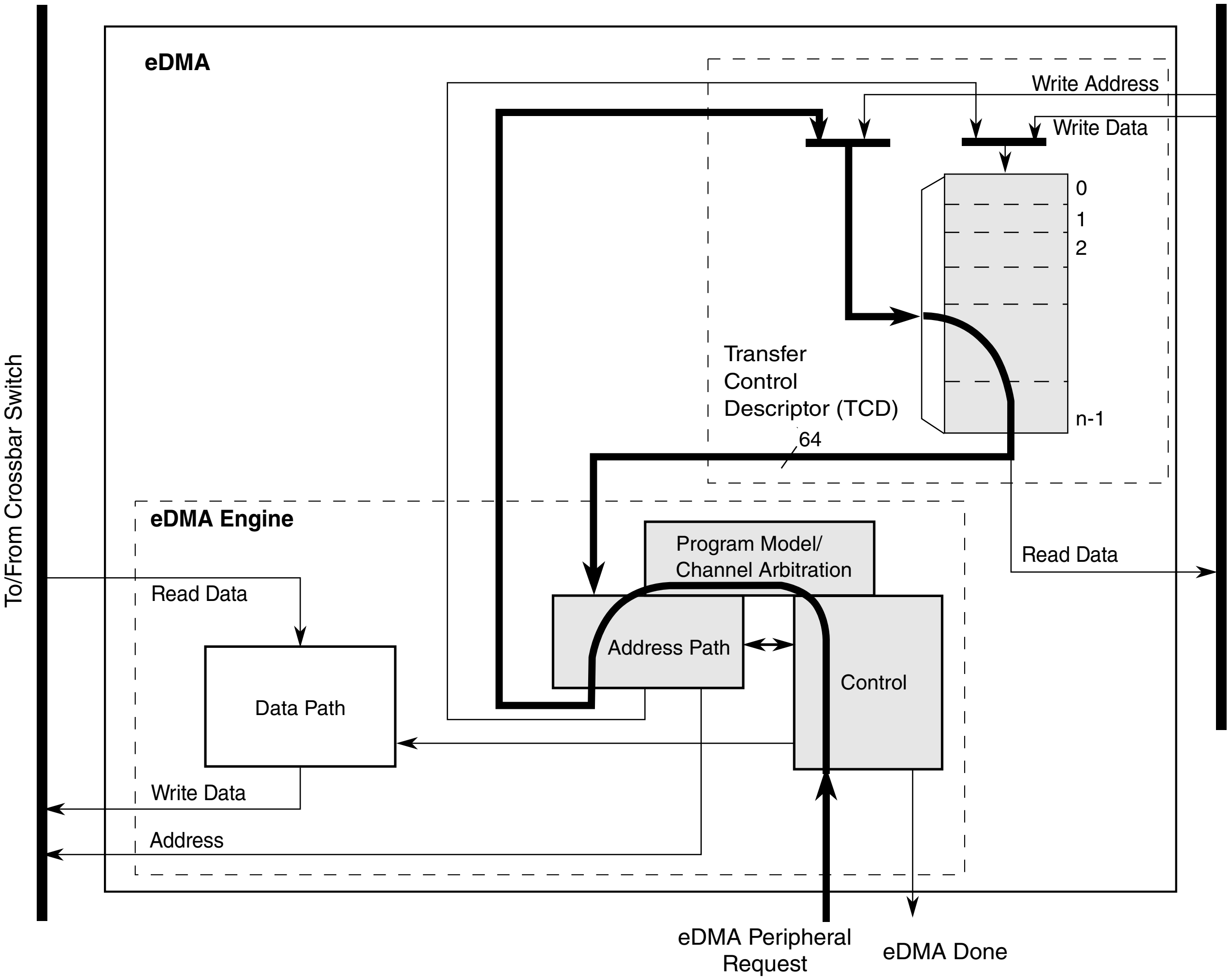


Figure 28-2. eDMA operation, part 1

This example uses the assertion of the eDMA peripheral request signal to request service for channel *n*. Channel activation via software and the TCD*n*_CSR[START] bit follows the same basic flow as peripheral requests. The eDMA request input signal is registered internally and then routed through the eDMA engine: first through the control module, then into the program model and channel arbitration. In the next cycle, the channel arbitration performs, using the fixed-priority or round-robin algorithm. After arbitration is complete, the activated channel number is sent through the address path and converted into the required address to access the local memory for TCD*n*. Next, the TCD memory is accessed and the required descriptor read from the local memory and loaded into the eDMA engine address path channel x or y registers. The TCD memory is 64 bits wide to minimize the time needed to fetch the activated channel descriptor and load it into the address path channel x or y registers.

The following diagram illustrates the second part of the basic data flow:

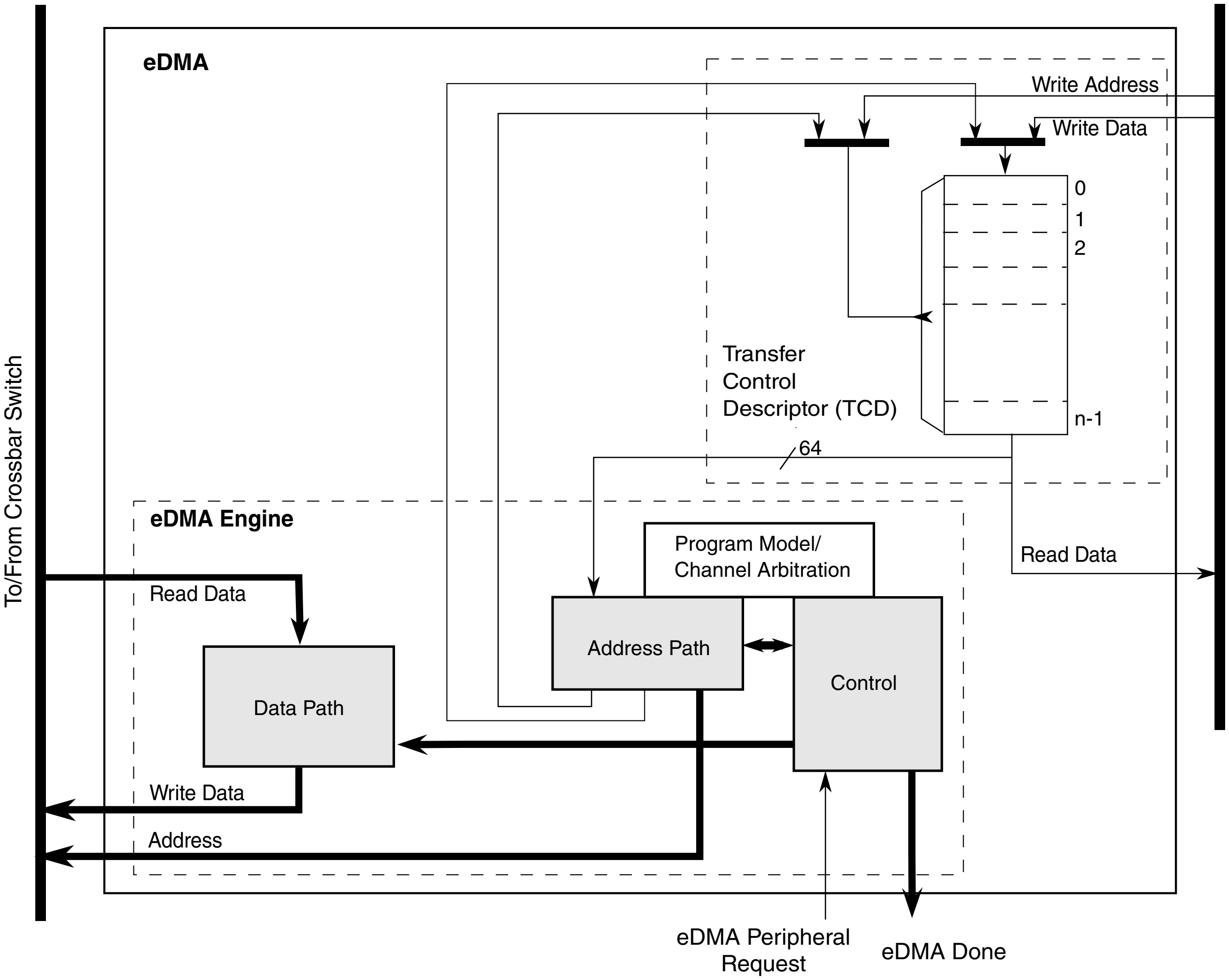


Figure 28-3. eDMA operation, part 2

The modules associated with the data transfer (address path, data path, and control) sequence through the required source reads and destination writes to perform the actual data movement. The source reads are initiated and the fetched data is temporarily stored in the data path block until it is gated onto the internal bus during the destination write. This source read/destination write processing continues until the minor byte count has transferred.

After the minor byte count has moved, the final phase of the basic data flow is performed. In this segment, the address path logic performs the required updates to certain fields in the appropriate TCD, for example, SADDR, DADDR, CITER. If the major iteration count is exhausted, additional operations are performed. These include the final address adjustments and reloading of the BITER field into the CITER. Assertion of an optional interrupt request also occurs at this time, as does a possible fetch of a new TCD from memory using the scatter/gather address pointer included in the descriptor (if scatter/gather is enabled). The updates to the TCD memory and the assertion of an interrupt request are shown in the following diagram.

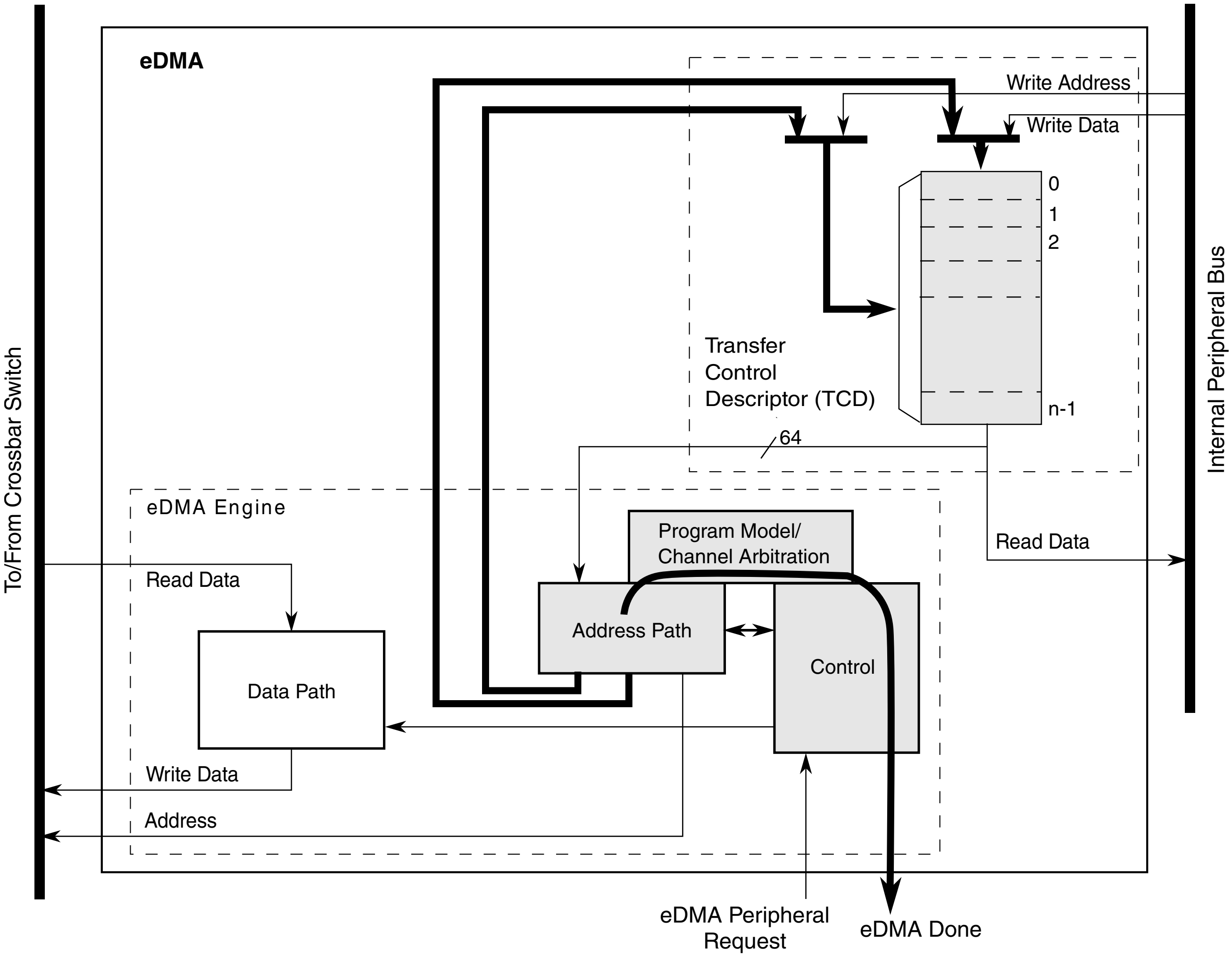


Figure 28-4. eDMA operation, part 3

28.4.2 Fault reporting and handling

Channel errors are reported in the Error Status register (DMAx_ES) and can be caused by:

- A configuration error, which is an illegal setting in the transfer-control descriptor or an illegal priority register setting in Fixed-Arbitration mode, or
- An error termination to a bus master read or write cycle

A configuration error is reported when the starting source or destination address, source or destination offsets, minor loop byte count, or the transfer size represent an inconsistent state. Each of these possible causes are detailed below:

- The addresses and offsets must be aligned on 0-modulo-transfer-size boundaries.
- The minor loop byte count must be a multiple of the source and destination transfer sizes.

- All source reads and destination writes must be configured to the natural boundary of the programmed transfer size respectively.
- In fixed arbitration mode, a configuration error is caused by any two channel priorities being equal. All channel priority levels must be unique when fixed arbitration mode is enabled.

NOTE

When two channels have the same priority, a channel priority error exists and will be reported in the Error Status register. However, the channel number will not be reported in the Error Status register. When all of the channel priorities within a group are not unique, the channel number selected by arbitration is undetermined.

To aid in Channel Priority Error (CPE) debug, set the Halt On Error bit in the DMA's Control Register. If all of the channel priorities within a group are not unique, the DMA will be halted after the CPE error is recorded. The DMA will remain halted and will not process any channel service requests. Once all of the channel priorities are set to unique numbers, the DMA may be enabled again by clearing the Halt bit.

- If a scatter/gather operation is enabled upon channel completion, a configuration error is reported if the scatter/gather address (DLAST_SGA) is not aligned on a 32-byte boundary.
- If minor loop channel linking is enabled upon channel completion, a configuration error is reported when the link is attempted if the TCDn_CITER[E_LINK] bit does not equal the TCDn_BITER[E_LINK] bit.

If enabled, all configuration error conditions, except the scatter/gather and minor-loop link errors, report as the channel activates and asserts an error interrupt request. A scatter/gather configuration error is reported when the scatter/gather operation begins at major loop completion when properly enabled. A minor loop channel link configuration error is reported when the link operation is serviced at minor loop completion.

If a system bus read or write is terminated with an error, the data transfer is stopped and the appropriate bus error flag set. In this case, the state of the channel's transfer control descriptor is updated by the eDMA engine with the current source address, destination address, and current iteration count at the point of the fault. When a system bus error occurs, the channel terminates after the next transfer. Due to pipeline effect, the next transfer is already in progress when the bus error is received by the eDMA. If a bus error occurs on the last read prior to beginning the write sequence, the write executes using the data captured during the bus error. If a bus error occurs on the last write prior to switching to the next read sequence, the read sequence executes before the channel terminates due to the destination bus error.

A transfer may be cancelled by software with the CR[CX] bit. When a cancel transfer request is recognized, the DMA engine stops processing the channel. The current read-write sequence is allowed to finish. If the cancel occurs on the last read-write sequence of a major or minor loop, the cancel request is discarded and the channel retires normally.

The error cancel transfer is the same as a cancel transfer except the Error Status register (DMAx_ES) is updated with the cancelled channel number and ECX is set. The TCD of a cancelled channel contains the source and destination addresses of the last transfer saved in the TCD. If the channel needs to be restarted, you must re-initialize the TCD because the aforementioned fields no longer represent the original parameters. When a transfer is cancelled by the error cancel transfer mechanism, the channel number is loaded into DMA_ES[ERRCHN] and ECX and VLD are set. In addition, an error interrupt may be generated if enabled.

NOTE

The cancel transfer request allows the user to stop a large data transfer in the event the full data transfer is no longer needed. The cancel transfer bit does not abort the channel. It simply stops the transferring of data and then retires the channel through its normal shutdown sequence. The application software must handle the context of the cancel. If an interrupt is desired (or not), then the interrupt should be enabled (or disabled) before the cancel request. The application software must clean up the transfer control descriptor since the full transfer did not occur.

The occurrence of any error causes the eDMA engine to stop normal processing of the active channel immediately (it goes to its error processing states and the transaction to the system bus still has pipeline effect), and the appropriate channel bit in the eDMA error register is asserted. At the same time, the details of the error condition are loaded into the Error Status register (DMAx_ES). The major loop complete indicators, setting the transfer control descriptor DONE flag and the possible assertion of an interrupt request, are not affected when an error is detected. After the error status has been updated, the eDMA engine continues operating by servicing the next appropriate channel. A channel that experiences an error condition is not automatically disabled. If a channel is terminated by an error and then issues another service request before the error is fixed, that channel executes and terminates with the same error condition.

28.4.3 Channel preemption

Channel preemption is enabled on a per-channel basis by setting the DCHPRIn[ECP] bit. Channel preemption allows the executing channel's data transfers to temporarily suspend in favor of starting a higher priority channel. After the preempting channel has completed all its minor loop data transfers, the preempted channel is restored and resumes execution. After the restored channel completes one read/write sequence, it is again eligible for preemption. If any higher priority channel is requesting service, the restored channel is suspended and the higher priority channel is serviced. Nested preemption, that is, attempting to preempt a preempting channel, is not supported. After a preempting channel begins execution, it cannot be preempted. Preemption is available only when fixed arbitration is selected.

A channel's ability to preempt another channel can be disabled by setting DCHPRIn[DPA]. When a channel's preempt ability is disabled, that channel cannot suspend a lower priority channel's data transfer, regardless of the lower priority channel's ECP setting. This allows for a pool of low priority, large data-moving channels to be defined. These low priority channels can be configured to not preempt each other, thus preventing a low priority channel from consuming the preempt slot normally available to a true, high priority channel.

28.4.4 Performance

This section addresses the performance of the eDMA module, focusing on two separate metrics:

- In the traditional data movement context, performance is best expressed as the peak data transfer rates achieved using the eDMA. In most implementations, this transfer rate is limited by the speed of the source and destination address spaces.
- In a second context where device-paced movement of single data values to/from peripherals is dominant, a measure of the requests that can be serviced in a fixed time is a more relevant metric. In this environment, the speed of the source and destination address spaces remains important. However, the microarchitecture of the eDMA also factors significantly into the resulting metric.

28.4.4.1 Peak transfer rates

The peak transfer rates for several different source and destination transfers are shown in the following tables. These tables assume:

- Internal SRAM can be accessed with zero wait-states when viewed from the system bus data phase
- All internal peripheral bus reads require two wait-states, and internal peripheral bus writes three wait-states, when viewed from the system bus data phase
- All internal peripheral bus accesses are 32-bits in size

NOTE

All architectures will not meet the assumptions listed above. See the SRAM configuration section for more information.

This table compares peak transfer rates based on different possible system speeds. Specific chips/devices may not support all system speeds listed.

Table 28-4. eDMA peak transfer rates (Mbytes/sec)

System Speed, Width	Internal SRAM-to- Internal SRAM	32 bit internal peripheral bus-to-Internal SRAM	Internal SRAM-to-32 bit internal peripheral bus
66.7 MHz, 32 bit	133.3	66.7	53.3
83.3 MHz, 32 bit	166.7	83.3	66.7
100.0 MHz, 32 bit	200.0	100.0	80.0
133.3 MHz, 32 bit	266.7	133.3	106.7
150.0 MHz, 32 bit	300.0	150.0	120.0

Internal-SRAM-to-internal-SRAM transfers occur at the core's datapath width. For all transfers involving the internal peripheral bus, 32-bit transfer sizes are used. In all cases, the transfer rate includes the time to read the source plus the time to write the destination.

28.4.4.2 Peak request rates

The second performance metric is a measure of the number of DMA requests that can be serviced in a given amount of time. For this metric, assume that the peripheral request causes the channel to move a single internal peripheral bus-mapped operand to/from internal SRAM. The same timing assumptions used in the previous example apply to this calculation. In particular, this metric also reflects the time required to activate the channel.

The eDMA design supports the following hardware service request sequence. Note that the exact timing from Cycle 7 is a function of the response times for the channel's read and write accesses. In the case of an internal peripheral bus read and internal SRAM write, the combined data phase time is 4 cycles. For an SRAM read and internal peripheral bus write, it is 5 cycles.

Table 28-5. Hardware service request process

Cycle		Description
With internal peripheral bus read and internal SRAM write	With SRAM read and internal peripheral bus write	
1		eDMA peripheral request is asserted.
2		The eDMA peripheral request is registered locally in the eDMA module and qualified. TCD*n*_CSR[START] bit initiated requests start at this point with the registering of the user write to TCD*n* word 7.
3		Channel arbitration begins.
4		Channel arbitration completes. The transfer control descriptor local memory read is initiated.
5–6		The first two parts of the activated channel's TCD is read from the local memory. The memory width to the eDMA engine is 64 bits, so the entire descriptor can be accessed in four cycles
7		The first system bus read cycle is initiated, as the third part of the channel's TCD is read from the local memory. Depending on the state of the crossbar switch, arbitration at the system bus may insert an additional cycle of delay here.
8–11	8–12	The last part of the TCD is read in. This cycle represents the first data phase for the read, and the address phase for the destination write.
12	13	This cycle represents the data phase of the last destination write.
13	14	The eDMA engine completes the execution of the inner minor loop and prepares to write back the required TCD*n* fields into the local memory. The TCD*n* word 7 is read and checked for channel linking or scatter/gather requests.
14	15	The appropriate fields in the first part of the TCD*n* are written back into the local memory.
15	16	The fields in the second part of the TCD*n* are written back into the local memory. This cycle coincides with the next channel arbitration cycle start.
16	17	The next channel to be activated performs the read of the first part of its TCD from the local memory. This is equivalent to Cycle 4 for the first channel's service request.

Assuming zero wait states on the system bus, DMA requests can be processed every 9 cycles. Assuming an average of the access times associated with internal peripheral bus-to-SRAM (4 cycles) and SRAM-to-internal peripheral bus (5 cycles), DMA requests can be processed every 11.5 cycles (4 + (4+5)/2 + 3). This is the time from Cycle 4 to Cycle x +5. The resulting peak request rate, as a function of the system frequency, is shown in the following table.

Table 28-6. eDMA peak request rate (MReq/sec)

System frequency (MHz)	Request rate with zero wait states	Request rate with wait states
66.6	7.4	5.8
83.3	9.2	7.2
100.0	11.1	8.7
133.3	14.8	11.6
150.0	16.6	13.0

A general formula to compute the peak request rate with overlapping requests is:

PEAKreq = freq / [entry + (1 + read_ws) + (1 + write_ws) + exit]

where:

Table 28-7. Peak request formula operands

Operand	Description
PEAKreq	Peak request rate
freq	System frequency
entry	Channel startup (4 cycles)
read_ws	Wait states seen during the system bus read data phase
write_ws	Wait states seen during the system bus write data phase
exit	Channel shutdown (3 cycles)

28.4.4.3 eDMA performance example

Consider a system with the following characteristics:

- Internal SRAM can be accessed with one wait-state when viewed from the system bus data phase
- All internal peripheral bus reads require two wait-states, and internal peripheral bus writes three wait-states viewed from the system bus data phase
- System operates at 150 MHz

For an SRAM to internal peripheral bus transfer,

PEAKreq = 150 MHz / [4 + (1 + 1) + (1 + 3) + 3] cycles = 11.5 Mreq/sec

For an internal peripheral bus to SRAM transfer, PEAKreq = 150 MHz / [4 + (1 + 2) + (1 + 1) + 3] cycles = 12.5 Mreq/sec

Assuming an even distribution of the two transfer types, the average peak request rate would be:

PEAKreq = (11.5 Mreq/sec + 12.5 Mreq/sec) / 2 = 12.0 Mreq/sec

The minimum number of cycles to perform a single read/write, zero wait states on the system bus, from a cold start where no channel is executing and eDMA is idle are:

- 11 cycles for a software, that is, a TCD*n*_CSR[START] bit, request
- 12 cycles for a hardware, that is, an eDMA peripheral request signal, request

Two cycles account for the arbitration pipeline and one extra cycle on the hardware request resulting from the internal registering of the eDMA peripheral request signals. For the peak request rate calculations above, the arbitration and request registering is absorbed in or overlaps the previous executing channel.

Note

When channel linking or scatter/gather is enabled, a two cycle delay is imposed on the next channel selection and startup. This allows the link channel or the scatter/gather channel to be eligible and considered in the arbitration pool for next channel selection.

28.5 Initialization/application information

The following sections discuss initialization of the eDMA and programming considerations.

28.5.1 eDMA initialization

To initialize the eDMA:

1. Write to the CR if a configuration other than the default is desired.
2. Write the channel priority levels to the DCHPRI*n* registers if a configuration other than the default is desired.
3. Enable error interrupts in the EEI register if so desired.
4. Write the 32-byte TCD for each channel that may request service.

5. Enable any hardware service requests via the ERQH and ERQL registers.
6. Request channel service via either:
   - Software: setting the TCD*n*_CSR[START]
   - Hardware: slave device asserting its eDMA peripheral request signal

After any channel requests service, a channel is selected for execution based on the arbitration and priority levels written into the programmer's model. The eDMA engine reads the entire TCD, including the TCD control and status fields, as shown in the following table, for the selected channel into its internal address path module.

As the TCD is read, the first transfer is initiated on the internal bus, unless a configuration error is detected. Transfers from the source, as defined by TCD*n*_SADDR, to the destination, as defined by TCD*n*_DADDR, continue until the number of bytes specified by TCD*n*_NBYTES are transferred.

When the transfer is complete, the eDMA engine's local TCD*n*_SADDR, TCD*n*_DADDR, and TCD*n*_CITER are written back to the main TCD memory and any minor loop channel linking is performed, if enabled. If the major loop is exhausted, further post processing executes, such as interrupts, major loop channel linking, and scatter/gather operations, if enabled.

Table 28-8. TCD Control and Status fields

TCD*n*_CSR field name	Description
START	Control bit to start channel explicitly when using a software initiated DMA service (Automatically cleared by hardware)
ACTIVE	Status bit indicating the channel is currently in execution
DONE	Status bit indicating major loop completion (cleared by software when using a software initiated DMA service)
D_REQ	Control bit to disable DMA request at end of major loop completion when using a hardware initiated DMA service
BWC	Control bits for throttling bandwidth control of a channel
E_SG	Control bit to enable scatter-gather feature
INT_HALF	Control bit to enable interrupt when major loop is half complete
INT_MAJ	Control bit to enable interrupt when major loop completes

The following figure shows how each DMA request initiates one minor-loop transfer, or iteration, without CPU intervention. DMA arbitration can occur after each minor loop, and one level of minor loop DMA preemption is allowed. The number of minor loops in a major loop is specified by the beginning iteration count (BITER).

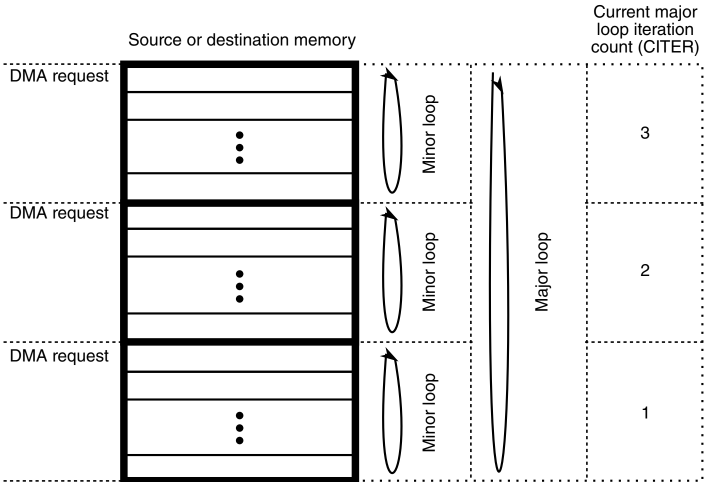


Figure 28-5. Example of multiple loop iterations

The following figure lists the memory array terms and how the TCD settings interrelate.

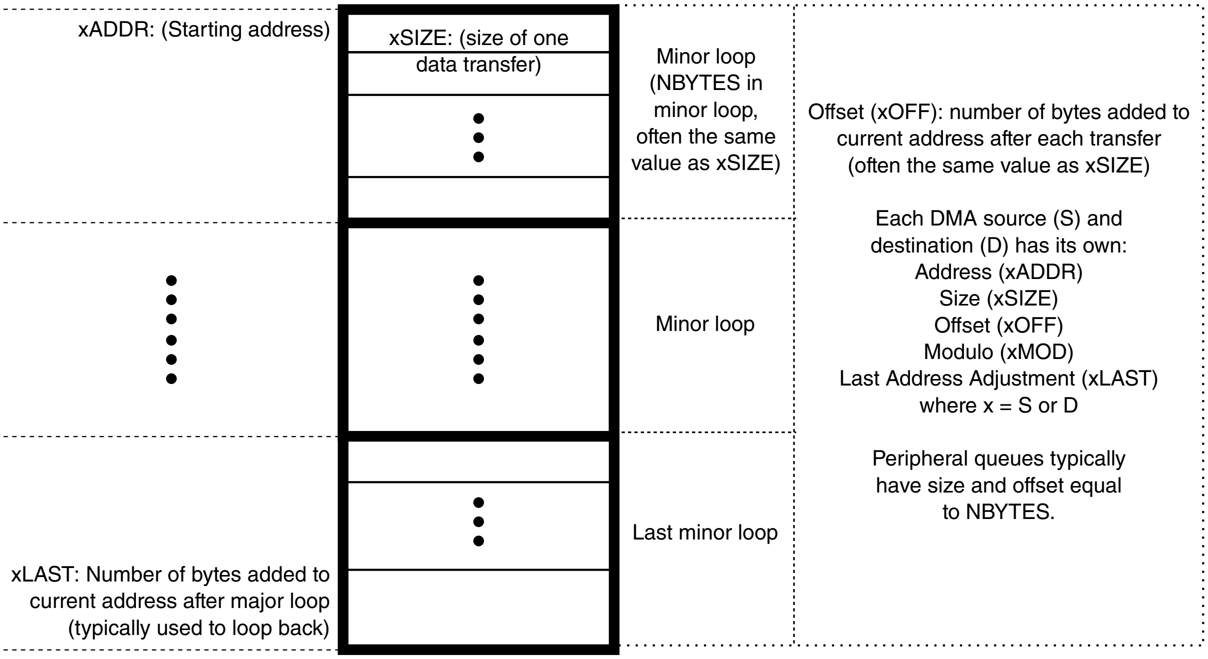

xADDR: (Starting address)
xSIZE: (size of one data transfer)
Minor loop (NBYTES in minor loop, often the same value as xSIZE)
Offset (xOFF): number of bytes added to current address after each transfer (often the same value as xSIZE)
Each DMA source (S) and destination (D) has its own:
Address (xADDR)
Size (xSIZE)
Offset (xOFF)
Modulo (xMOD)
Last Address Adjustment (xLAST)
where x = S or D
Peripheral queues typically have size and offset equal to NBYTES.
Minor loop
Last minor loop
xLAST: Number of bytes added to current address after major loop (typically used to loop back)

Figure 28-6. Memory array terms

28.5.2 Programming errors

The eDMA performs various tests on the transfer control descriptor to verify consistency in the descriptor data. Most programming errors are reported on a per channel basis with the exception of channel priority error (ES[CPE]).

For all error types other than channel priority error, the channel number causing the error is recorded in the Error Status register (DMAx_ES). If the error source is not removed before the next activation of the problem channel, the error is detected and recorded again.

If priority levels are not unique, when any channel requests service, a channel priority error is reported. The highest channel priority with an active request is selected, but the lowest numbered channel with that priority is selected by arbitration and executed by the eDMA engine. The hardware service request handshake signals, error interrupts, and error reporting is associated with the selected channel.

28.5.3 Arbitration mode considerations

This section discusses arbitration considerations for the eDMA.

28.5.3.1 Fixed channel arbitration

In this mode, the channel service request from the highest priority channel is selected to execute.

28.5.3.2 Round-robin channel arbitration

Channels are serviced starting with the highest channel number and rotating through to the lowest channel number without regard to the channel priority levels.

28.5.4 Performing DMA transfers

This section presents examples on how to perform DMA transfers with the eDMA.

28.5.4.1 Single request

To perform a simple transfer of n bytes of data with one activation, set the major loop to one (TCD*n*_CITER = TCD*n*_BITER = 1). The data transfer begins after the channel service request is acknowledged and the channel is selected to execute. After the transfer is complete, the TCD*n*_CSR[DONE] bit is set and an interrupt generates if properly enabled.

For example, the following TCD entry is configured to transfer 16 bytes of data. The eDMA is programmed for one iteration of the major loop transferring 16 bytes per iteration. The source memory has a byte wide memory port located at 0x1000. The destination memory has a 32-bit port located at 0x2000. The address offsets are programmed in increments to match the transfer size: one byte for the source and four bytes for the destination. The final source and destination addresses are adjusted to return to their beginning values.

```
TCDn_CITER = TCDn_BITER = 1
TCDn_NBYTES = 16
TCDn_SADDR = 0x1000
TCDn_SOFF = 1
TCDn_ATTR[SSIZE] = 0
TCDn_SLAST = -16
TCDn_DADDR = 0x2000
TCDn_DOFF = 4
TCDn_ATTR[DSIZE] = 2
TCDn_DLAST_SGA= -16
TCDn_CSR[INT_MAJ] = 1
TCDn_CSR[START] = 1 (Should be written last after all other fields have been initialized)
All other TCDn fields = 0
```

This generates the following event sequence:

1. User write to the TCD*n*_CSR[START] bit requests channel service.
2. The channel is selected by arbitration for servicing.
3. eDMA engine writes: TCD*n*_CSR[DONE] = 0, TCD*n*_CSR[START] = 0, TCD*n*_CSR[ACTIVE] = 1.
4. eDMA engine reads: channel TCD data from local memory to internal register file.
5. The source-to-destination transfers are executed as follows:
    a. Read byte from location 0x1000, read byte from location 0x1001, read byte from 0x1002, read byte from 0x1003.
    b. Write 32-bits to location 0x2000 → first iteration of the minor loop.
    c. Read byte from location 0x1004, read byte from location 0x1005, read byte from 0x1006, read byte from 0x1007.
    d. Write 32-bits to location 0x2004 → second iteration of the minor loop.
    e. Read byte from location 0x1008, read byte from location 0x1009, read byte from 0x100A, read byte from 0x100B.
    f. Write 32-bits to location 0x2008 → third iteration of the minor loop.
    g. Read byte from location 0x100C, read byte from location 0x100D, read byte from 0x100E, read byte from 0x100F.

   h. Write 32-bits to location 0x200C → last iteration of the minor loop → major loop complete.

6. The eDMA engine writes: TCD*n*_SADDR = 0x1000, TCD*n*_DADDR = 0x2000, TCD*n*_CITER = 1 (TCD*n*_BITER).
7. The eDMA engine writes: TCD*n*_CSR[ACTIVE] = 0, TCD*n*_CSR[DONE] = 1, INT[*n*] = 1.
8. The channel retires and the eDMA goes idle or services the next channel.

28.5.4.2 Multiple requests

The following example transfers 32 bytes via two hardware requests, but is otherwise the same as the previous example. The only fields that change are the major loop iteration count and the final address offsets. The eDMA is programmed for two iterations of the major loop transferring 16 bytes per iteration. After the channel's hardware requests are enabled in the ERQ register, the slave device initiates channel service requests.

```
TCDn_CITER = TCDn_BITER = 2
TCDn_SLAST = -32
TCDn_DLAST_SGA = -32
```

This would generate the following sequence of events:

1. First hardware, that is, eDMA peripheral, request for channel service.
2. The channel is selected by arbitration for servicing.
3. eDMA engine writes: TCD*n*_CSR[DONE] = 0, TCD*n*_CSR[START] = 0, TCD*n*_CSR[ACTIVE] = 1.
4. eDMA engine reads: channel TCD*n* data from local memory to internal register file.
5. The source to destination transfers are executed as follows:
   a. Read byte from location 0x1000, read byte from location 0x1001, read byte from 0x1002, read byte from 0x1003.
   b. Write 32-bits to location 0x2000 → first iteration of the minor loop.
   c. Read byte from location 0x1004, read byte from location 0x1005, read byte from 0x1006, read byte from 0x1007.
   d. Write 32-bits to location 0x2004 → second iteration of the minor loop.
   e. Read byte from location 0x1008, read byte from location 0x1009, read byte from 0x100A, read byte from 0x100B.

   f. Write 32-bits to location 0x2008 → third iteration of the minor loop.
   g. Read byte from location 0x100C, read byte from location 0x100D, read byte from 0x100E, read byte from 0x100F.
   h. Write 32-bits to location 0x200C → last iteration of the minor loop.

6. eDMA engine writes: TCD*n*_SADDR = 0x1010, TCD*n*_DADDR = 0x2010, TCD*n*_CITER = 1.
7. eDMA engine writes: TCD*n*_CSR[ACTIVE] = 0.
8. The channel retires → one iteration of the major loop. The eDMA goes idle or services the next channel.
9. Second hardware, that is, eDMA peripheral, requests channel service.
10. The channel is selected by arbitration for servicing.
11. eDMA engine writes: TCD*n*_CSR[DONE] = 0, TCD*n*_CSR[START] = 0, TCD*n*_CSR[ACTIVE] = 1.
12. eDMA engine reads: channel TCD data from local memory to internal register file.
13. The source to destination transfers are executed as follows:
   a. Read byte from location 0x1010, read byte from location 0x1011, read byte from 0x1012, read byte from 0x1013.
   b. Write 32-bits to location 0x2010 → first iteration of the minor loop.
   c. Read byte from location 0x1014, read byte from location 0x1015, read byte from 0x1016, read byte from 0x1017.
   d. Write 32-bits to location 0x2014 → second iteration of the minor loop.
   e. Read byte from location 0x1018, read byte from location 0x1019, read byte from 0x101A, read byte from 0x101B.
   f. Write 32-bits to location 0x2018 → third iteration of the minor loop.
   g. Read byte from location 0x101C, read byte from location 0x101D, read byte from 0x101E, read byte from 0x101F.
   h. Write 32-bits to location 0x201C → last iteration of the minor loop → major loop complete.
14. eDMA engine writes: TCD*n*_SADDR = 0x1000, TCD*n*_DADDR = 0x2000, TCD*n*_CITER = 2 (TCD*n*_BITER).

15. eDMA engine writes: TCD*n*_CSR[ACTIVE] = 0, TCD*n*_CSR[DONE] = 1, INT[n] = 1.
16. The channel retires → major loop complete. The eDMA goes idle or services the next channel.

28.5.4.3 Using the modulo feature

The modulo feature of the eDMA provides the ability to implement a circular data queue in which the size of the queue is a power of 2. MOD is a 5-bit field for the source and destination in the TCD, and it specifies which lower address bits increment from their original value after the address+offset calculation. All upper address bits remain the same as in the original value. A setting of 0 for this field disables the modulo feature.

The following table shows how the transfer addresses are specified based on the setting of the MOD field. Here a circular buffer is created where the address wraps to the original value while the 28 upper address bits (0x1234567*x*) retain their original value. In this example the source address is set to 0x12345670, the offset is set to 4 bytes and the MOD field is set to 4, allowing for a 2^4 byte (16-byte) size queue.

Table 28-9. Modulo example

Transfer Number	Address
1	0x12345670
2	0x12345674
3	0x12345678
4	0x1234567C
5	0x12345670
6	0x12345674

28.5.5 Monitoring transfer descriptor status

This section discusses how to monitor eDMA status.

28.5.5.1 Testing for minor loop completion

There are two methods to test for minor loop completion when using software initiated service requests. The first is to read the TCD*n*_CITER field and test for a change. Another method may be extracted from the sequence shown below. The second method is to test the TCD*n*_CSR[START] bit and the TCD*n*_CSR[ACTIVE] bit. The minor-loop-complete condition is indicated by both bits reading zero after the TCD*n*_CSR[START] was set. Polling the TCD*n*_CSR[ACTIVE] bit may be inconclusive, because the active status may be missed if the channel execution is short in duration.

The TCD status bits execute the following sequence for a software activated channel:

Stage	TCD*n*_CSR bits			State
	START	ACTIVE	DONE	
1	1	0	0	Channel service request via software
2	0	1	0	Channel is executing
3a	0	0	0	Channel has completed the minor loop and is idle
3b	0	0	1	Channel has completed the major loop and is idle

The best method to test for minor-loop completion when using hardware, that is, peripheral, initiated service requests is to read the TCD*n*_CITER field and test for a change. The hardware request and acknowledge handshake signals are not visible in the programmer's model.

The TCD status bits execute the following sequence for a hardware-activated channel:

Stage	TCD*n*_CSR bits			State
	START	ACTIVE	DONE	
1	0	0	0	Channel service request via hardware (peripheral request asserted)
2	0	1	0	Channel is executing
3a	0	0	0	Channel has completed the minor loop and is idle
3b	0	0	1	Channel has completed the major loop and is idle

For both activation types, the major-loop-complete status is explicitly indicated via the TCD*n*_CSR[DONE] bit.

The TCD*n*_CSR[START] bit is cleared automatically when the channel begins execution regardless of how the channel activates.

28.5.5.2 Reading the transfer descriptors of active channels

The eDMA reads back the true TCD*n*_SADDR, TCD*n*_DADDR, and TCD*n*_NBYTES values if read while a channel executes. The true values of the SADDR, DADDR, and NBYTES are the values the eDMA engine currently uses in its internal register file and not the values in the TCD local memory for that channel. The addresses, SADDR and

DADDR, and NBYTES, which decrement to zero as the transfer progresses, can give an indication of the progress of the transfer. All other values are read back from the TCD local memory.

28.5.5.3 Checking channel preemption status

Preemption is available only when fixed arbitration is selected as the channel arbitration mode. A preemptive situation is one in which a preempt-enabled channel runs and a higher priority request becomes active. When the eDMA engine is not operating in fixed channel arbitration mode, the determination of the actively running relative priority outstanding requests become undefined. Channel priorities are treated as equal, that is, constantly rotating, when Round-Robin Arbitration mode is selected.

The TCD*n*_CSR[ACTIVE] bit for the preempted channel remains asserted throughout the preemption. The preempted channel is temporarily suspended while the preempting channel executes one major loop iteration. If two TCD*n*_CSR[ACTIVE] bits are set simultaneously in the global TCD map, a higher priority channel is actively preempting a lower priority channel.

28.5.6 Channel Linking

Channel linking (or chaining) is a mechanism where one channel sets the TCD*n*_CSR[START] bit of another channel (or itself), therefore initiating a service request for that channel. When properly enabled, the EDMA engine automatically performs this operation at the major or minor loop completion.

The minor loop channel linking occurs at the completion of the minor loop (or one iteration of the major loop). The TCD*n*_CITER[E_LINK] field determines whether a minor loop link is requested. When enabled, the channel link is made after each iteration of the major loop except for the last. When the major loop is exhausted, only the major loop channel link fields are used to determine if a channel link should be made. For example, the initial fields of:

```
TCDn_CITER[E_LINK] = 1
TCDn_CITER[LINKCH] = 0xC
TCDn_CITER[CITER] value = 0x4
TCDn_CSR[MAJOR_E_LINK] = 1
TCDn_CSR[MAJOR_LINKCH] = 0x7
```

executes as:

1. Minor loop done → set TCD12_CSR[START] bit

2. Minor loop done → set TCD12_CSR[START] bit
3. Minor loop done → set TCD12_CSR[START] bit
4. Minor loop done, major loop done→ set TCD7_CSR[START] bit

When minor loop linking is enabled (TCD*n*_CITER[E_LINK] = 1), the TCD*n*_CITER[CITER] field uses a nine bit vector to form the current iteration count. When minor loop linking is disabled (TCD*n*_CITER[E_LINK] = 0), the TCD*n*_CITER[CITER] field uses a 15-bit vector to form the current iteration count. The bits associated with the TCD*n*_CITER[LINKCH] field are concatenated onto the CITER value to increase the range of the CITER.

Note

> The TCD*n*_CITER[E_LINK] bit and the TCD*n*_BITER[E_LINK] bit must equal or a configuration error is reported. The CITER and BITER vector widths must be equal to calculate the major loop, half-way done interrupt point.

The following table summarizes how a DMA channel can link to another DMA channel, i.e, use another channel's TCD, at the end of a loop.

Table 28-10. Channel Linking Parameters

Desired Link Behavior	TCD Control Field Name	Description
Link at end of Minor Loop	CITER[E_LINK]	Enable channel-to-channel linking on minor loop completion (current iteration)
	CITER[LINKCH]	Link channel number when linking at end of minor loop (current iteration)
Link at end of Major Loop	CSR[MAJOR_E_LINK]	Enable channel-to-channel linking on major loop completion
	CSR[MAJOR_LINKCH]	Link channel number when linking at end of major loop

28.5.7 Dynamic programming

This section provides recommended methods to change the programming model during channel execution.

28.5.7.1 Dynamically changing the channel priority

The following two options are recommended for dynamically changing channel priority levels:

1. Switch to Round-Robin Channel Arbitration mode, change the channel priorities, then switch back to Fixed Arbitration mode,
2. Disable all the channels, change the channel priorities, then enable the appropriate channels.

28.5.7.2 Dynamic channel linking

Dynamic channel linking is the process of setting the TCD.major.e_link bit during channel execution (see the diagram in TCD structure). This bit is read from the TCD local memory at the end of channel execution, thus allowing the user to enable the feature during channel execution.

Because the user is allowed to change the configuration during execution, a coherency model is needed. Consider the scenario where the user attempts to execute a dynamic channel link by enabling the TCD.major.e_link bit at the same time the eDMA engine is retiring the channel. The TCD.major.e_link would be set in the programmer's model, but it would be unclear whether the actual link was made before the channel retired.

The following coherency model is recommended when executing a dynamic channel link request.

1. Write 1 to the TCD.major.e_link bit.
2. Read back the TCD.major.e_link bit.
3. Test the TCD.major.e_link request status:
   - If TCD.major.e_link = 1, the dynamic link attempt was successful.
   - If TCD.major.e_link = 0, the attempted dynamic link did not succeed (the channel was already retiring).

For this request, the TCD local memory controller forces the TCD.major.e_link bit to zero on any writes to a channel's TCD.word7 after that channel's TCD.done bit is set, indicating the major loop is complete.

NOTE

The user must clear the TCD.done bit before writing the TCD.major.e_link bit. The TCD.done bit is cleared automatically by the eDMA engine after a channel begins execution.

28.5.7.3 Dynamic scatter/gather

Scatter/gather is the process of automatically loading a new TCD into a channel. It allows a DMA channel to use multiple TCDs; this enables a DMA channel to scatter the DMA data to multiple destinations or gather it from multiple sources.When scatter/gather is enabled and the channel has finished its major loop, a new TCD is fetched from system memory and loaded into that channel's descriptor location in eDMA programmer's model, thus replacing the current descriptor.

Because the user is allowed to change the configuration during execution, a coherency model is needed. Consider the scenario where the user attempts to execute a dynamic scatter/gather operation by enabling the TCD.e_sg bit at the same time the eDMA engine is retiring the channel. The TCD.e_sg would be set in the programmer's model, but it would be unclear whether the actual scatter/gather request was honored before the channel retired.

Two methods for this coherency model are shown in the following subsections. Method 1 has the advantage of reading the major.linkch field and the e_sg bit with a single read. For both dynamic channel linking and scatter/gather requests, the TCD local memory controller forces the TCD.major.e_link and TCD.e_sg bits to zero on any writes to a channel's TCD.word7 if that channel's TCD.done bit is set indicating the major loop is complete.

NOTE

The user must clear the TCD.done bit before writing the TCD.major.e_link or TCD.e_sg bits. The TCD.done bit is cleared automatically by the eDMA engine after a channel begins execution.

28.5.7.3.1 Method 1 (channel not using major loop channel linking)

For a channel not using major loop channel linking, the coherency model described here may be used for a dynamic scatter/gather request.

When the TCD.major.e_link bit is zero, the TCD.major.linkch field is not used by the eDMA. In this case, the TCD.major.linkch bits may be used for other purposes. This method uses the TCD.major.linkch field as a TCD indentification (ID).

1. When the descriptors are built, write a unique TCD ID in the TCD.major.linkch field for each TCD associated with a channel using dynamic scatter/gather.
2. Write 1b to the TCD.d_req bit.

Should a dynamic scatter/gather attempt fail, setting the TCD.d_req bit will prevent a future hardware activation of this channel. This stops the channel from executing with a destination address (daddr) that was calculated using a scatter/gather address (written in the next step) instead of a dlast final offest value.

3. Write the TCD.dlast_sga field with the scatter/gather address.
4. Write 1b to the TCD.e_sg bit.
5. Read back the 16 bit TCD control/status field.
6. Test the TCD.e_sg request status and TCD.major.linkch value:

   If e_sg = 1b, the dynamic link attempt was successful.

   If e_sg = 0b and the major.linkch (ID) did not change, the attempted dynamic link did not succeed (the channel was already retiring).

   If e_sg = 0b and the major.linkch (ID) changed, the dynamic link attempt was successful (the new TCD's e_sg value cleared the e_sg bit).

28.5.7.3.2 Method 2 (channel using major loop channel linking)

For a channel using major loop channel linking, the coherency model described here may be used for a dynamic scatter/gather request. This method uses the TCD.dlast_sga field as a TCD indentification (ID).

1. Write 1b to the TCD.d_req bit.

   Should a dynamic scatter/gather attempt fail, setting the d_req bit will prevent a future hardware activation of this channel. This stops the channel from executing with a destination address (daddr) that was calculated using a scatter/gather address (written in the next step) instead of a dlast final offest value.

2. Write theTCD.dlast_sga field with the scatter/gather address.
3. Write 1b to the TCD.e_sg bit.
4. Read back the TCD.e_sg bit.
5. Test the TCD.e_sg request status:

   If e_sg = 1b, the dynamic link attempt was successful.

   If e_sg = 0b, read the 32 bit TCD dlast_sga field.

   If e_sg = 0b and the dlast_sga did not change, the attempted dynamic link did not succeed (the channel was already retiring).

If e_sg = 0b and the dlast_sga changed, the dynamic link attempt was successful (the new TCD’s e_sg value cleared the e_sg bit).

Chapter 29
External Watchdog Monitor (EWM)

External Watchdog Monitor (EWM)External Watchdog Monitor EWM

29.1 Features

Features of EWM module include:

- Independent LPO_CLK clock source
- Programmable time-out period specified in terms of number of EWM LPO_CLK clock cycles.
- Windowed refresh option
  - Provides robust check that program flow is faster than expected.
  - Programmable window.
  - Refresh outside window leads to assertion of $\overline{\text{EWM\_out}}$.
- Robust refresh mechanism
  - Write values of 0xB4 and 0x2C to EWM Refresh Register within 15 (*EWM_refresh_time*) peripheral bus clock cycles.
- One output port, $\overline{\text{EWM\_out}}$, when asserted is used to reset or place the external circuit into safe mode.
- One Input port, EWM_in, allows an external circuit to control the assertion of the $\overline{\text{EWM\_out}}$ signal.

29.2 Modes of Operation

This section describes the module's operating modes.

29.2.1 Stop Mode

When the EWM is in stop mode, the CPU refreshes to the EWM cannot occur. On entry to stop mode, the EWM's counter freezes.

There are two possible ways to exit from Stop mode:

- On exit from stop mode through a reset, the EWM remains disabled.
- On exit from stop mode by an interrupt, the EWM is re-enabled, and the counter continues to be clocked from the same value prior to entry to stop mode.

Note the following if the EWM enters the stop mode during CPU refresh mechanism: At the exit from stop mode by an interrupt, refresh mechanism state machine starts from the previous state which means, if first refresh command is written correctly and EWM enters the stop mode immediately, the next command has to be written within the next 15 (*EWM_refresh_time*) peripheral bus clocks after exiting from stop mode. User must mask all interrupts prior to executing EWM refresh instructions.

29.2.2 Wait Mode

The EWM module treats the stop and wait modes as the same. EWM functionality remains the same in both of these modes.

29.2.3 Debug Mode

Entry to debug mode has no effect on the EWM.

- If the EWM is enabled prior to entry of debug mode, it remains enabled.
- If the EWM is disabled prior to entry of debug mode, it remains disabled.

29.3 Block Diagram

This figure shows the EWM block diagram.

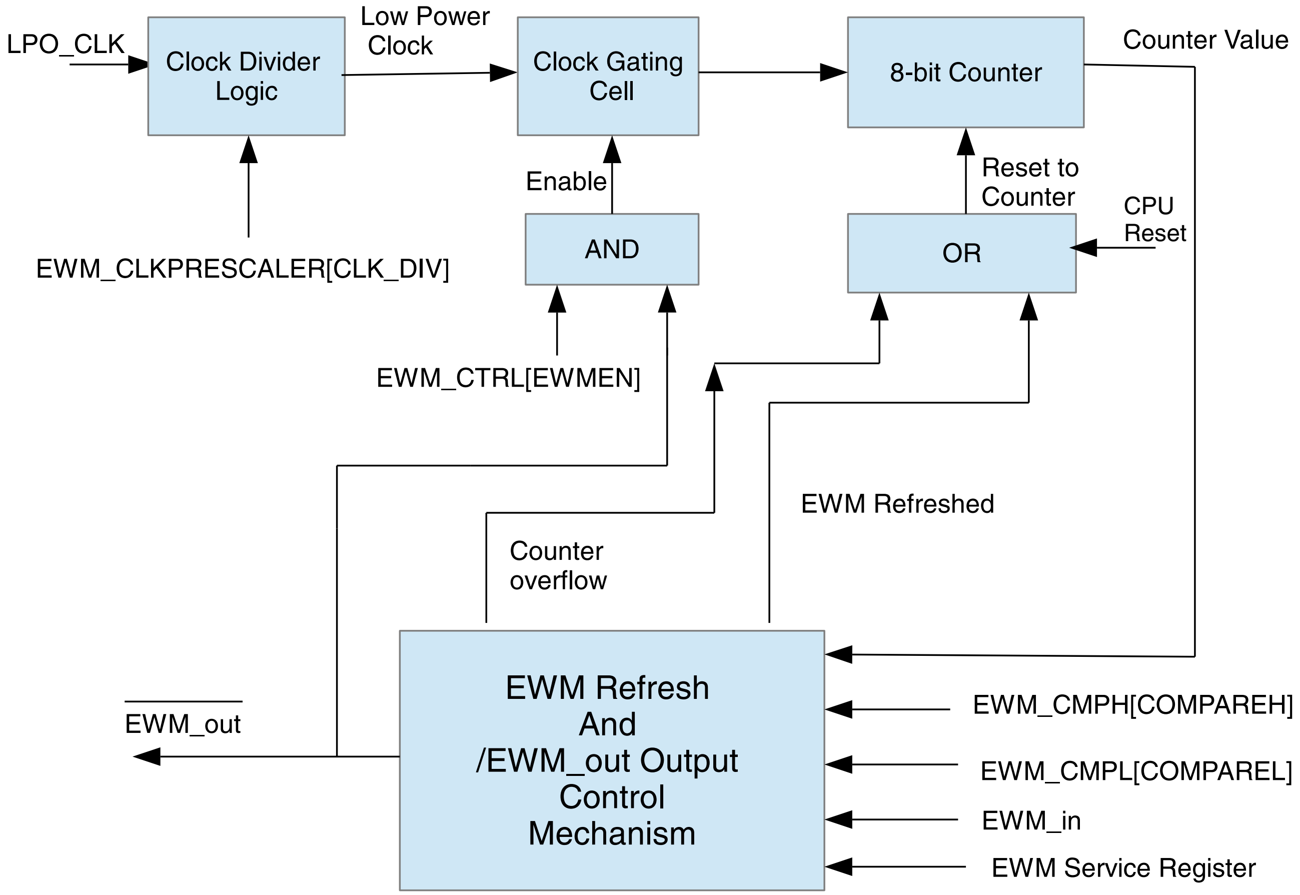


Figure 29-1. EWM Block Diagram

29.4 External Watchdog Monitor (EWM)

29.4.1 Features

29.4.2 Modes of Operation

29.4.2.1 Stop Mode

29.4.2.2 Wait Mode

29.4.2.3 Debug Mode

29.4.3 Block Diagram

29.5 External Signal Description

EWM memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4006_1000	Control Register (EWM_CTRL)	8	R/W	00h	29.6.6/626
4006_1001	Service Register (EWM_SERV)	8	W (always reads 0)	00h	29.6.7/627
4006_1002	Compare Low Register (EWM_CMPL)	8	R/W	00h	29.6.8/627
4006_1003	Compare High Register (EWM_CMPH)	8	R/W	FFh	29.6.9/628

29.6.6 Control Register (EWM_CTRL)

The CTRL register is cleared by any reset.

NOTE

INEN, ASSIN and EWMEN bits can be written once after a CPU reset. Modifying these bits more than once, generates a bus transfer error.

Address: 4006_1000h base + 0h offset = 4006_1000h

Bit	7	6	5	4	3	2	1	0
Read	0				INTEN	INEN	ASSIN	EWMEN
Write								
Reset	0	0	0	0	0	0	0	0

EWM_CTRL field descriptions

Field	Description
7–4 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
3 INTEN	Interrupt Enable. This bit when set and $\overline{\text{EWM\_out}}$ is asserted, an interrupt request is generated. To de-assert interrupt request, user should clear this bit by writing 0.
2 INEN	Input Enable. This bit when set, enables the EWM_in port.
1 ASSIN	EWM_in's Assertion State Select. Default assert state of the EWM_in signal is logic zero. Setting the ASSIN bit inverts the assert state of EWM_in signal to a logic one.
0 EWMEN	EWM enable. This bit when set, enables the EWM module. This resets the EWM counter to zero and deasserts the $\overline{\text{EWM\_out}}$ signal. This bit when unset, keeps the EWM module disabled. It cannot be re-enabled until a next reset, due to the write-once nature of this bit.

29.6.7 Service Register (EWM_SERV)

The SERV register provides the interface from the CPU to the EWM module. It is write-only and reads of this register return zero.

Address: 4006_1000h base + 1h offset = 4006_1001h

Bit	7	6	5	4	3	2	1	0
Read	0							
Write	SERVICE							
Reset	0	0	0	0	0	0	0	0

EWM_SERV field descriptions

Field	Description
SERVICE	The EWM refresh mechanism requires the CPU to write two values to the SERV register: a first data byte of 0xB4, followed by a second data byte of 0x2C. The EWM refresh is invalid if either of the following conditions is true. • The first or second data byte is not written correctly. • The second data byte is not written within a fixed number of peripheral bus cycles of the first data byte. This fixed number of cycles is called *EWM_refresh_time*.

29.6.8 Compare Low Register (EWM_CMPL)

The CMPL register is reset to zero after a CPU reset. This provides no minimum time for the CPU to refresh the EWM counter.

NOTE

This register can be written only once after a CPU reset. Writing this register more than once generates a bus transfer error.

Address: 4006_1000h base + 2h offset = 4006_1002h

Bit	7	6	5	4	3	2	1	0
Read Write	COMPAREL							
Reset	0	0	0	0	0	0	0	0

EWM_CMPL field descriptions

Field	Description
COMPAREL	To prevent runaway code from changing this field, software should write to this field after a CPU reset even if the (default) minimum refresh time is required.

29.6.9 Compare High Register (EWM_CMPH)

The CMPH register is reset to 0xFF after a CPU reset. This provides a maximum of 256 clocks time, for the CPU to refresh the EWM counter.

NOTE

This register can be written only once after a CPU reset. Writing this register more than once generates a bus transfer error.

NOTE

The valid values for CMPH are up to 0xFE because the EWM counter never expires when CMPH = 0xFF. The expiration happens only if EWM counter is greater than CMPH.

Address: 4006_1000h base + 3h offset = 4006_1003h

Bit	7	6	5	4	3	2	1	0
Read Write	COMPAREH							
Reset	1	1	1	1	1	1	1	1

EWM_CMPH field descriptions

Field	Description
COMPAREH	To prevent runaway code from changing this field, software should write to this field after a CPU reset even if the (default) maximum refresh time is required.

29.10 Memory Map

29.11 The $\overline{\text{EWM\_out}}$ Signal

The $\overline{\text{EWM\_out}}$ is a digital output signal used to gate an external circuit (application specific) that controls critical safety functions. For example, the $\overline{\text{EWM\_out}}$ could be connected to the high voltage transistors circuits that control an AC motor in a large appliance.

The $\overline{\text{EWM\_out}}$ signal remains deasserted when the EWM is being regularly refreshed by the CPU within the programmable refresh window, indicating that the application code is executed as expected.

The $\overline{\text{EWM\_out}}$ signal is asserted in any of the following conditions:

- The EWM refresh occurs when the counter value is less than CMPL value.
- The EWM counter value reaches the CMPH value, and no EWM refresh has occurred.
- If functionality of EWM_in pin is enabled and EWM_in pin is asserted while refreshing the EWM.
- After any reset (by the virtue of the external pull-down mechanism on the $\overline{\text{EWM\_out}}$ pin)

The $\overline{\text{EWM\_out}}$ is asserted after any reset by the virtue of the external pull-down mechanism on the $\overline{\text{EWM\_out}}$ signal. Then, to deassert the $\overline{\text{EWM\_out}}$ signal, set EWMEN bit in the CTRL register to enable the EWM.

If the $\overline{\text{EWM\_out}}$ signal shares its pad with a digital I/O pin, on reset this actual pad defers to being an input signal. The pad state is controlled by the $\overline{\text{EWM\_out}}$ signal only after the EWM is enabled by the EWMEN bit in the CTRL register.

Note

> $\overline{\text{EWM\_out}}$ pad must be in pull down state when EWM functionality is used and when EWM is under Reset.

29.12 The EWM_in Signal

The EWM_in is a digital input signal for safety status of external safety circuits, that allows an external circuit to control the assertion of the $\overline{\text{EWM\_out}}$ signal. For example, in the application, an external circuit monitors a critical safety function, and if there is fault with safety function, the external circuit can then actively initiate the $\overline{\text{EWM\_out}}$ signal that controls the gating circuit.

The EWM_in signal is ignored if the EWM is disabled, or if INEN bit of CTRL register is cleared, as after any reset.

On enabling the EWM (setting the CTRL[EWMEN] bit) and enabling EWM_in functionality (setting the CTRL[INEN] bit), the EWM_in signal must be in the deasserted state prior to the CPU start refreshing the EWM. This ensures that the $\overline{\text{EWM\_out}}$ stays in the deasserted state; otherwise, the $\overline{\text{EWM\_out}}$ output signal is asserted.

Note

> The user must update the CMPH and CMPL registers prior to enabling the EWM. After enabling the EWM, the counter resets to zero, therefore the user shall provide a reasonable time after a power-on reset for the external monitoring circuit to stabilize. The user shall also ensure that the EWM_in pin is deasserted.

29.13 EWM Counter

It is an 8-bit ripple counter fed from a clock source that is independent of the peripheral bus clock source. As the preferred time-out is between 1 ms and 100 ms the actual clock source should be in the kHz range.

The counter is reset to zero after the CPU reset, or when EWM refresh action completes, or at counter overflow. The counter value is not accessible to the CPU.

29.14 EWM Compare Registers

The compare registers CMPL and CMPH are write-once after a CPU reset and cannot be modified until another CPU reset occurs.

The EWM compare registers are used to create a refresh window to refresh the EWM module.

It is illegal to program CMPL and CMPH with same value. In this case, as soon as counter reaches (CMPL + 1), $\overline{\mathrm{EWM\_out}}$ is asserted.

29.15 EWM Refresh Mechanism

Other than the initial configuration of the EWM, the CPU can only access the EWM by the EWM Service Register. The CPU must access the EWM service register with correct write of unique data within the windowed time frame as determined by the CMPL and CMPH registers for correct EWM refresh operation. Therefore, three possible conditions can occur:

Table 29-1. EWM Refresh Mechanisms

Condition	Mechanism
An EWM refresh action completes when: CMPL < Counter < CMPH.	The software behaves as expected and the EWM counter is reset to zero. The $\overline{\mathrm{EWM\_out}}$ output signal remains in the deasserted state if, during the EWM refresh action, the EWM_in input has been in deasserted state..
An EWM refresh action completes when Counter < CMPL	The software refreshes the EWM before the windowed time frame, the counter is reset to zero and the $\overline{\mathrm{EWM\_out}}$ output signal is asserted irrespective of the input $\overline{\mathrm{EWM\_in}}$.
Counter value reaches CMPH prior to completion of EWM refresh action.	Software has not refreshed the EWM. The EWM counter is reset to zero and the $\overline{\mathrm{EWM\_out}}$ output signal is asserted irrespective of the input $\overline{\mathrm{EWM\_in}}$.

29.16 EWM Interrupt

When $\overline{\mathrm{EWM\_out}}$ is asserted, an interrupt request is generated to indicate the assertion of the EWM reset out signal. This interrupt is enabled when CTRL[INTEN] is set. Clearing this bit clears the interrupt request but does not affect $\overline{\mathrm{EWM\_out}}$. The $\overline{\mathrm{EWM\_out}}$ signal can be deasserted only by forcing a system reset.

29.17 Functional Description

29.17.1 The EWM_out Signal

29.17.2 The EWM_in Signal

29.17.3 EWM Counter

29.17.4 EWM Compare Registers

29.17.5 EWM Refresh Mechanism

29.17.6 EWM Interrupt

Chapter 30
MCU: Watchdog Timer (WDOG)

30.1 Introduction

NOTE

> For the chip-specific implementation details of this module's instances, see the chip configuration information.

The Watchdog Timer (WDOG) keeps a watch on the system functioning and resets it in case of its failure. Reasons for failure include run-away software code and the stoppage of the system clock that in a safety critical system can lead to serious consequences. In such cases, the watchdog brings the system into a safe state of operation. The watchdog monitors the operation of the system by expecting periodic communication from the software, generally known as servicing or refreshing the watchdog. If this periodic refreshing does not occur, the watchdog resets the system.

30.2 Features

The features of the Watchdog Timer (WDOG) include:

- Clock source input independent from CPU/bus clock. Choice between two clock sources:
  - Low-power oscillator (LPO)
  - External system clock
- Unlock sequence for allowing updates to write-once WDOG control/configuration bits.
- All WDOG control/configuration bits are writable once only within 256 bus clock cycles of being unlocked.

    - You need to always update these bits after unlocking within 256 bus clock cycles. Failure to update these bits resets the system.
- Programmable time-out period specified in terms of number of WDOG clock cycles.
- Ability to test WDOG timer and reset with a flag indicating watchdog test.
    - Quick test—Small time-out value programmed for quick test.
    - Byte test—Individual bytes of timer tested one at a time.
    - Read-only access to the WDOG timer—Allows dynamic check that WDOG timer is operational.

NOTE

Reading the watchdog timer counter while running the watchdog on the bus clock might not give the accurate counter value.

- Windowed refresh option
    - Provides robust check that program flow is faster than expected.
    - Programmable window.
    - Refresh outside window leads to reset.
- Robust refresh mechanism
    - Write values of 0xA602 and 0xB480 to WDOG Refresh Register within 20 bus clock cycles.
- Count of WDOG resets as they occur.
- Configurable interrupt on time-out to provide debug breadcrumbs. This is followed by a reset after 256 bus clock cycles.

30.3 Functional overview

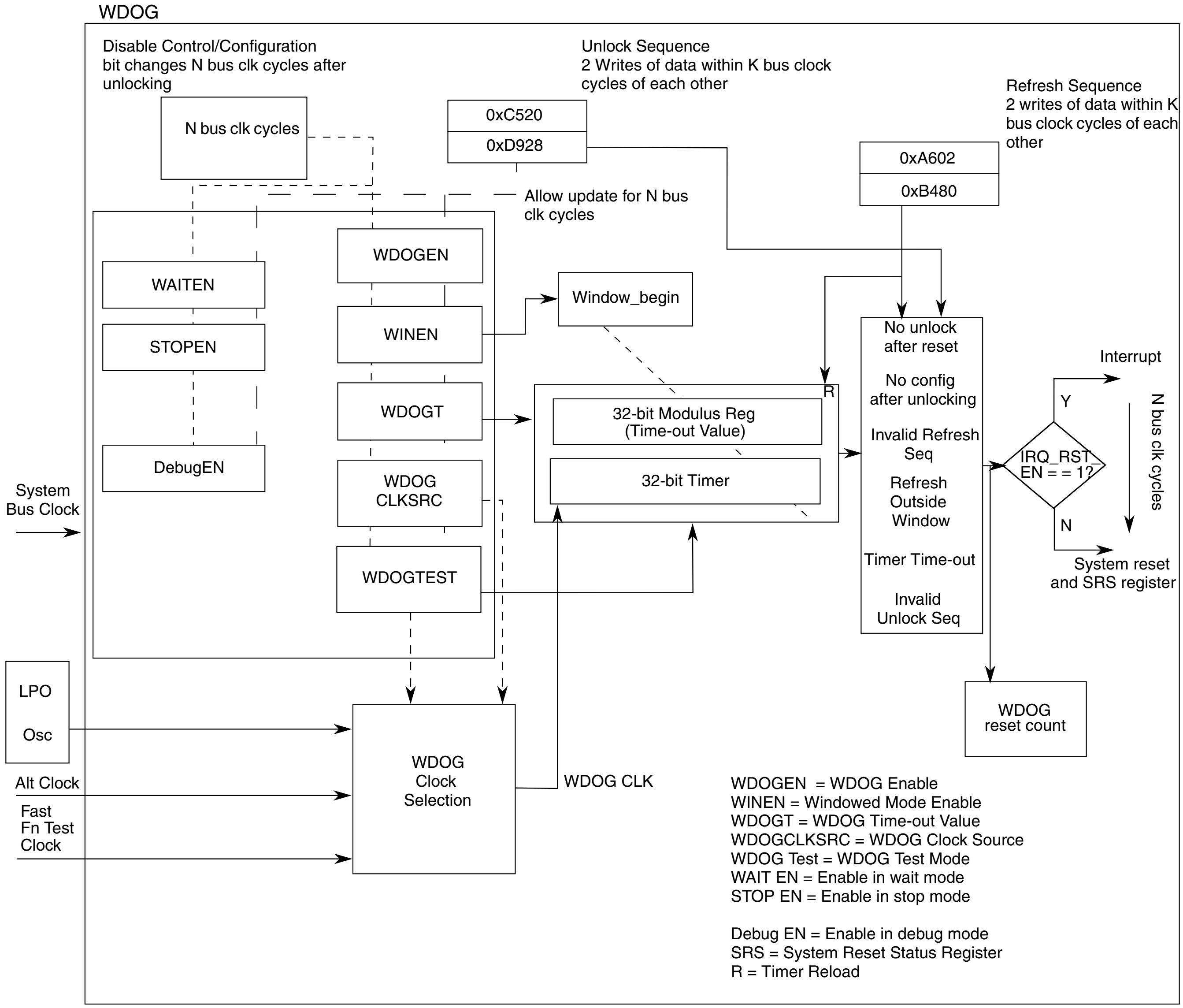

WDOG
Disable Control/Configuration bit changes N bus clk cycles after unlocking
N bus clk cycles
Unlock Sequence
2 Writes of data within K bus clock cycles of each other
0xC520
0xD928
Refresh Sequence
2 writes of data within K bus clock cycles of each other
0xA602
0xB480
Allow update for N bus clk cycles
WDOGEN
WAITEN
WINEN
Window_begin
STOPEN
WDOGT
32-bit Modulus Reg (Time-out Value)
R
32-bit Timer
DebugEN
WDOG CLKSRC
WDOGTEST
System Bus Clock
No unlock after reset
No config after unlocking
Invalid Refresh Seq
Refresh Outside Window
Timer Time-out
Invalid Unlock Seq
Interrupt
Y
IRQ_RST_ EN = = 1?
N
N bus clk cycles
System reset and SRS register
WDOG reset count
LPO
Osc
Alt Clock
Fast Fn Test Clock
WDOG Clock Selection
WDOG CLK
WDOGEN = WDOG Enable
WINEN = Windowed Mode Enable
WDOGT = WDOG Time-out Value
WDOGCLKSRC = WDOG Clock Source
WDOG Test = WDOG Test Mode
WAIT EN = Enable in wait mode
STOP EN = Enable in stop mode
Debug EN = Enable in debug mode
SRS = System Reset Status Register
R = Timer Reload

Figure 30-1. WDOG operation

The preceding figure shows the operation of the watchdog. The values for N and K are:

- N = 256
- K = 20

The watchdog is a fail safe mechanism that brings the system into a known initial state in case of its failure due to CPU clock stopping or a run-away condition in code execution. In its simplest form, the watchdog timer runs continuously off a clock source and expects to be serviced periodically, failing which it resets the system. This ensures that the software is executing correctly and has not run away in an unintended direction. Software can adjust the period of servicing or the time-out value for the watchdog timer to meet the needs of the application.

You can select a windowed mode of operation that expects the servicing to be done only in a particular window of the time-out period. An attempted servicing of the watchdog outside this window results in a reset. By operating in this mode, you can get an indication of whether the code is running faster than expected. The window length is also user programmable.

If a system fails to update/refresh the watchdog due to an unknown and persistent cause, it will be caught in an endless cycle of resets from the watchdog. To analyze the cause of such conditions, you can program the watchdog to first issue an interrupt, followed by a reset. In the interrupt service routine, the software can analyze the system stack to aid debugging.

To enhance the independence of watchdog from the system, it runs off an independent LPO oscillator clock. You can also switch over to an alternate clock source if required, through a control register bit.

30.3.1 Unlocking and updating the watchdog

As long as ALLOW_UPDATE in the watchdog control register is set, you can unlock and modify the write-once-only control and configuration registers:

1. Write 0xC520 followed by 0xD928 within 20 bus clock cycles to a specific unlock register (WDOG_UNLOCK).
2. Wait one bus clock cycle. You cannot update registers on the bus clock cycle immediately following the write of the unlock sequence.
3. An update window equal in length to the watchdog configuration time (WCT) opens. Within this window, you can update the configuration and control register bits.

These register bits can be modified only once after unlocking.

If none of the configuration and control registers is updated within the update window, the watchdog issues a reset, that is, interrupt-then-reset, to the system. Trying to unlock the watchdog within the WCT after an initial unlock has no effect. During the update operation, the watchdog timer is not paused and continues running in the background. After the update window closes, the watchdog timer restarts and the watchdog functions according to the new configuration.

The update feature is useful for applications that have an initial, non-safety critical part, where the watchdog is kept disabled or with a conveniently long time-out period. This means the application coder does not have to frequently service the watchdog. After the critical part of the application begins, the watchdog can be reconfigured as needed.

The watchdog issues a reset, that is, interrupt-then-reset if enabled, to the system for any of these invalid unlock sequences:

- Write any value other than 0xC520 or 0xD928 to the unlock register.
- ALLOW_UPDATE is set and a gap of more than 20 bus clock cycles is inserted between the writing of the unlock sequence values.

An attempted refresh operation between the two writes of the unlock sequence and in the WCT time following a successful unlock, goes undetected. Also, see Watchdog Operation with 8-bit access for guidelines related to 8-bit accesses to the unlock register.

Note

A context switch during unlocking and refreshing may lead to a watchdog reset.

30.3.2 Watchdog configuration time (WCT)

To prevent unintended modification of the watchdog's control and configuration register bits, you are allowed to update them only within a period of 256 bus clock cycles after unlocking. This period is known as the watchdog configuration time (WCT). In addition, these register bits can be modified only once after unlocking them for editing, even after reset.

You must unlock the registers within WCT after system reset, failing which the WDOG issues a reset to the system. In other words, you must write at least the first word of the unlocking sequence within the WCT after reset. After this is done, you have a further 20 bus clock cycles, the maximum allowed gap between the words of the unlock sequence, to complete the unlocking operation. Thereafter, to make sure that you do not forget to configure the watchdog, the watchdog issues a reset if none of the WDOG control and configuration registers is updated in the WCT after unlock. After the close of this window or after the first write, these register bits are locked out from any further changes.

The watchdog timer keeps running according to its default configuration through unlocking and update operations that can extend up to a maximum total of 2xWCT + 20 bus clock cycles. Therefore, it must be ensured that the time-out value for the watchdog is always greater than 2xWCT time + 20 bus clock cycles.

Updates in the write-once registers take effect only after the WCT window closes with the following exceptions for which changes take effect immediately:

- Stop, Wait, and Debug mode enable
- IRQ_RST_EN

The operations of refreshing the watchdog goes undetected during the WCT.

30.3.3 Refreshing the watchdog

A robust refreshing mechanism has been chosen for the watchdog. A valid refresh is a write of 0xA602 followed by 0xB480 within 20 bus clock cycles to watchdog refresh register. If these two values are written more than 20 bus cycles apart or if something other than these two values is written to the register, a watchdog reset, or interrupt-then-reset if enabled, is issued to the system. A valid refresh makes the watchdog timer restart on the next bus clock. Also, an attempted unlock operation in between the two writes of the refresh sequence goes undetected. See Watchdog Operation with 8-bit access for guidelines related to 8-bit accesses to the refresh register.

30.3.4 Windowed mode of operation

In this mode of operation, a restriction is placed on the point in time within the time-out period at which the watchdog can be refreshed. The refresh is considered valid only when the watchdog timer increments beyond a certain count as specified by the watchdog window register. This is known as refreshing the watchdog within a window of the total time-out period. If a refresh is attempted before the timer reaches the window value, the watchdog generates a reset, or interrupt-then-reset if enabled. If there is no refresh at all, the watchdog times out and generates a reset or interrupt-then-reset if enabled.

30.3.5 Watchdog disabled mode of operation

When the watchdog is disabled through the WDOG_EN bit in the watchdog status and control register, the watchdog timer is reset to zero and is disabled from counting until you enable it or it is enabled again by the system reset. In this mode, the watchdog timer cannot be refreshed–there is no requirement to do so while the timer is disabled. However, the watchdog still generates a reset, or interrupt-then-reset if enabled, on a non-time-out exception. See Generated Resets and Interrupts. You need to unlock the watchdog before enabling it. A system reset brings the watchdog out of the disabled mode.

30.3.6 Low-power modes of operation

The low-power modes of operation of the watchdog are described in the following table:

Table 30-1. Low-power modes of operation

Mode	Behavior
Wait	If the WDOG is enabled (WAIT_EN = 1), it can run on bus clock or low-power oscillator clock (CLK_SRC = x) to generate interrupt (IRQ_RST_EN=1) followed by a reset on time-out. After reset the WDOG reset counter increments by one.
Stop	Where the bus clock is gated, the WDOG can run only on low-power oscillator clock (CLK_SRC=0) if it is enabled in stop (STOP_EN=1). In this case, the WDOG runs to time-out twice, and then generates a reset from its backup circuitry. Therefore, if you program the watchdog to time-out after 100 ms and then enter such a stop mode, the reset will occur after 200 ms. Also, in this case, no interrupt will be generated irrespective of the value of IRQ_RST_EN bit. After WDOG reset, the WDOG reset counter will also not increment.
Power-Down	The watchdog is • static in LLS mode • powered off in VLLSx mode

30.3.7 Debug modes of operation

You can program the watchdog to disable in debug modes through DBG_EN in the watchdog control register. This results in the watchdog timer pausing for the duration of the mode. Register read/writes are still allowed, which means that operations like refresh, unlock, and so on are allowed. Upon exit from the mode, the timer resumes its operation from the point of pausing.

The entry of the system into the debug mode does not excuse it from compulsorily configuring the watchdog in the WCT time after unlock, unless the system bus clock is gated off, in which case the internal state machine pauses too. Failing to do so still results in a reset, or interrupt-then-reset, if enabled, to the system. Also, all of the exception conditions that result in a reset to the system, as described in Generated Resets and Interrupts, are still valid in this mode. So, if an exception condition occurs and the system bus clock is on, a reset occurs, or interrupt-then-reset, if enabled.

The entry into Debug mode within WCT after reset is treated differently. The WDOG timer is kept reset to zero and there is no need to unlock and configure it within WCT. You must not try to refresh or unlock the WDOG in this state or unknown behavior may result. Upon exit from this mode, the WDOG timer restarts and the WDOG has to be unlocked and configured within WCT.

30.4 Testing the watchdog

For IEC 60730 and other safety standards, the expectation is that anything that monitors a safety function must be tested, and this test is required to be fault tolerant. To test the watchdog, its main timer and its associated compare and reset logic must be tested. To this end, two tests are implemented for the watchdog, as described in Quick Test and Byte Test. A control bit is provided to put the watchdog into functional test mode. There is also an overriding test-disable control bit which allows the functional test mode to be disabled permanently. After it is set, this test-disable bit can only be cleared by a reset.

These two tests achieve the overall aim of testing the counter functioning and the compare and reset logic.

Note

> Do not enable the watchdog interrupt during these tests. If required, you must ensure that the effective time-out value is greater than WCT time. See Generated Resets and Interrupts for more details.

To run a particular test:

1. Select either quick test or byte test..
2. Set a certain test mode bit to put the watchdog in the functional test mode. Setting this bit automatically switches the watchdog timer to a fast clock source. The switching of the clock source is done to achieve a faster time-out and hence a faster test.

In a successful test, the timer times out after reaching the programmed time-out value and generates a system reset.

Note

> After emerging from a reset due to a watchdog test, unlock and configure the watchdog. The refresh and unlock operations and interrupt are not automatically disabled in the test mode.

30.4.1 Quick test

In this test, the time-out value of watchdog timer is programmed to a very low value to achieve quick time-out. The only difference between the quick test and the normal mode of the watchdog is that TESTWDOG is set for the quick test. This allows for a faster test of the watchdog reset mechanism.

30.4.2 Byte test

The byte test is a more thorough a test of the watchdog timer. In this test, the timer is split up into its constituent byte-wide stages that are run independently and tested for time-out against the corresponding byte of the time-out value register. The following figure explains the splitting concept:

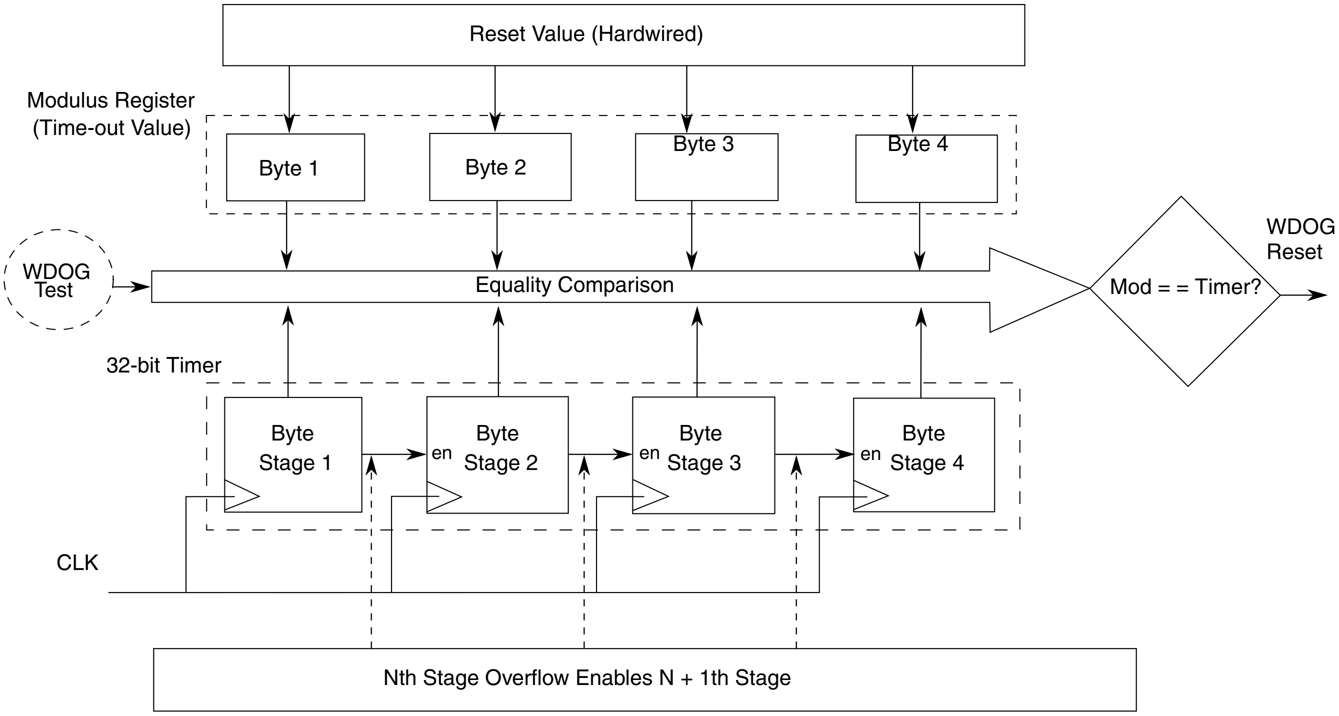


Figure 30-2. Watchdog timer byte splitting

Each stage is an 8-bit synchronous counter followed by combinational logic that generates an overflow signal. The overflow signal acts as an enable to the N + 1th stage.

In the test mode, when an individual byte, N, is tested, byte N – 1 is loaded forcefully with 0xFF, and both these bytes are allowed to run off the clock source. By doing so, the overflow signal from stage N – 1 is generated immediately, enabling counter stage N. The Nth stage runs and compares with the Nth byte of the time-out value register. In this way, the byte N is also tested along with the link between it and the preceding stage. No other stages, N – 2, N – 3... and N + 1, N + 2... are enabled for the test on byte N. These disabled stages, except the most significant stage of the counter, are loaded with a value of 0xFF.

30.5 Backup reset generator

The backup reset generator generates the final reset which goes out to the system. It has a backup mechanism which ensures that in case the bus clock stops and prevents the main state machine from generating a reset exception/interrupt, the watchdog timer's time-out is separately routed out as a reset to the system. Two successive timer time-outs without an intervening system reset result in the backup reset generator routing out the time-out signal as a reset to the system.

30.6 Generated resets and interrupts

The watchdog generates a reset in the following events, also referred to as exceptions:

- A watchdog time-out
- Failure to unlock the watchdog within WCT time after system reset deassertion
- No update of the control and configuration registers within the WCT window after unlocking. At least one of the following registers must be written to within the WCT window to avoid reset:
  - WDOG_ST_CTRL_H, WDOG_ST_CTRL_L
  - WDOG_TO_VAL_H, WDOG_TO_VAL_L
  - WDOG_WIN_H, WDOG_WIN_L
  - WDOG_PRESCALER
- A value other than the unlock sequence or the refresh sequence is written to the unlock and/or refresh registers, respectively.
- A gap of more than 20 bus cycles exists between the writes of two values of the unlock sequence.
- A gap of more than 20 bus cycles exists between the writes of two values of the refresh sequence.

The watchdog can also generate an interrupt. If IRQ_RST_EN is set, then on the above mentioned events WDOG_ST_CTRL_L[INT_FLG] is set, generating an interrupt. A watchdog reset is also generated WCT time later to ensure the watchdog is fault tolerant. The interrupt can be cleared by writing 1 to INT_FLG.

The gap of WCT between interrupt and reset means that the WDOG time-out value must be greater than WCT. Otherwise, if the interrupt was generated due to a time-out, a second consecutive time-out will occur in that WCT gap. This will trigger the backup reset generator to generate a reset to the system, prematurely ending the interrupt service routine execution. Also, jobs such as counting the number of watchdog resets would not be done.

30.7 Memory map and register definition

This section consists of the memory map and register descriptions.

WDOG memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4005_2000	Watchdog Status and Control Register High (WDOG_STCTRLH)	16	R/W	01D3h	30.7.1/644
4005_2002	Watchdog Status and Control Register Low (WDOG_STCTRLL)	16	R/W	0001h	30.7.2/645
4005_2004	Watchdog Time-out Value Register High (WDOG_TOVALH)	16	R/W	004Ch	30.7.3/646
4005_2006	Watchdog Time-out Value Register Low (WDOG_TOVALL)	16	R/W	4B4Ch	30.7.4/646
4005_2008	Watchdog Window Register High (WDOG_WINH)	16	R/W	0000h	30.7.5/647
4005_200A	Watchdog Window Register Low (WDOG_WINL)	16	R/W	0010h	30.7.6/647
4005_200C	Watchdog Refresh register (WDOG_REFRESH)	16	R/W	B480h	30.7.7/648
4005_200E	Watchdog Unlock register (WDOG_UNLOCK)	16	R/W	D928h	30.7.8/648
4005_2010	Watchdog Timer Output Register High (WDOG_TMROUTH)	16	R/W	0000h	30.7.9/648
4005_2012	Watchdog Timer Output Register Low (WDOG_TMROUTL)	16	R/W	0000h	30.7.10/ 649
4005_2014	Watchdog Reset Count register (WDOG_RSTCNT)	16	R/W	0000h	30.7.11/ 649
4005_2016	Watchdog Prescaler register (WDOG_PRESC)	16	R/W	0400h	30.7.12/ 649

30.7.1 Watchdog Status and Control Register High (WDOG_STCTRLH)

Address: 4005_2000h base + 0h offset = 4005_2000h

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	0	DISTESTWDOG	BYTESEL[1:0]		TESTSEL	TESTWDOG	0	Reserved	WAITEN	STOPEN	DBGEN	ALLOWUPDATE	WINEN	IRQRSTEN	CLKSRC	WDOGEN
Write	▨						▨									
Reset	0	0	0	0	0	0	0	1	1	1	0	1	0	0	1	1

WDOG_STCTRLH field descriptions

Field	Description
15 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
14 DISTESTWDOG	Allows the WDOG's functional test mode to be disabled permanently. After it is set, it can only be cleared by a reset. It cannot be unlocked for editing after it is set. 0 WDOG functional test mode is not disabled. 1 WDOG functional test mode is disabled permanently until reset.
13–12 BYTESEL[1:0]	This 2-bit field selects the byte to be tested when the watchdog is in the byte test mode. 00 Byte 0 selected 01 Byte 1 selected 10 Byte 2 selected 11 Byte 3 selected
11 TESTSEL	Effective only if TESTWDOG is set. Selects the test to be run on the watchdog timer. 0 Quick test. The timer runs in normal operation. You can load a small time-out value to do a quick test. 1 Byte test. Puts the timer in the byte test mode where individual bytes of the timer are enabled for operation and are compared for time-out against the corresponding byte of the programmed time-out value. Select the byte through BYTESEL[1:0] for testing.
10 TESTWDOG	Puts the watchdog in the functional test mode. In this mode, the watchdog timer and the associated compare and reset generation logic is tested for correct operation. The clock for the timer is switched from the main watchdog clock to the fast clock input for watchdog functional test. The TESTSEL bit selects the test to be run.
9 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
8 Reserved	This field is reserved.
7 WAITEN	Enables or disables WDOG in Wait mode. 0 WDOG is disabled in CPU Wait mode. 1 WDOG is enabled in CPU Wait mode.
6 STOPEN	Enables or disables WDOG in Stop mode.

Table continues on the next page...

WDOG_STCTRLH field descriptions (continued)

Field	Description
	0 WDOG is disabled in CPU Stop mode. 1 WDOG is enabled in CPU Stop mode.
5 DBGEN	Enables or disables WDOG in Debug mode. 0 WDOG is disabled in CPU Debug mode. 1 WDOG is enabled in CPU Debug mode.
4 ALLOWUPDATE	Enables updates to watchdog write-once registers, after the reset-triggered initial configuration window (WCT) closes, through unlock sequence. 0 No further updates allowed to WDOG write-once registers. 1 WDOG write-once registers can be unlocked for updating.
3 WINEN	Enables Windowing mode. 0 Windowing mode is disabled. 1 Windowing mode is enabled.
2 IRQRSTEN	Used to enable the debug breadcrumbs feature. A change in this bit is updated immediately, as opposed to updating after WCT. 0 WDOG time-out generates reset only. 1 WDOG time-out initially generates an interrupt. After WCT, it generates a reset.
1 CLKSRC	Selects clock source for the WDOG timer and other internal timing operations. 0 WDOG clock sourced from LPO . 1 WDOG clock sourced from alternate clock source.
0 WDOGEN	Enables or disables the WDOG's operation. In the disabled state, the watchdog timer is kept in the reset state, but the other exception conditions can still trigger a reset/interrupt. A change in the value of this bit must be held for more than one WDOG_CLK cycle for the WDOG to be enabled or disabled. 0 WDOG is disabled. 1 WDOG is enabled.

30.7.2 Watchdog Status and Control Register Low (WDOG_STCTRLL)

Address: 4005_2000h base + 2h offset = 4005_2002h

Bit	15	14	13	12	11	10	9	8
Read Write	INTFLG	Reserved						
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Read Write	Reserved							
Reset	0	0	0	0	0	0	0	1

WDOG_STCTRLL field descriptions

Field	Description
15 INTFLG	Interrupt flag. It is set when an exception occurs. IRQRSTEN = 1 is a precondition to set this flag. INTFLG = 1 results in an interrupt being issued followed by a reset, WCT later. The interrupt can be cleared by writing 1 to this bit. It also gets cleared on a system reset.
Reserved	This field is reserved. **NOTE:** Do not modify this field value.

30.7.3 Watchdog Time-out Value Register High (WDOG_TOVALH)

Address: 4005_2000h base + 4h offset = 4005_2004h

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read Write	TOVALHIGH															
Reset	0	0	0	0	0	0	0	0	0	1	0	0	1	1	0	0

WDOG_TOVALH field descriptions

Field	Description
TOVALHIGH	Defines the upper 16 bits of the 32-bit time-out value for the watchdog timer. It is defined in terms of cycles of the watchdog clock.

30.7.4 Watchdog Time-out Value Register Low (WDOG_TOVALL)

The time-out value of the watchdog must be set to a minimum of four watchdog clock cycles. This is to take into account the delay in new settings taking effect in the watchdog clock domain.

Address: 4005_2000h base + 6h offset = 4005_2006h

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read Write	TOVALLOW															
Reset	0	1	0	0	1	0	1	1	0	1	0	0	1	1	0	0

WDOG_TOVALL field descriptions

Field	Description
TOVALLOW	Defines the lower 16 bits of the 32-bit time-out value for the watchdog timer. It is defined in terms of cycles of the watchdog clock.

30.7.5 Watchdog Window Register High (WDOG_WINH)

NOTE

You must set the Window Register value lower than the Time-out Value Register.

Address: 4005_2000h base + 8h offset = 4005_2008h

<table>
<tr><th>Bit</th><th>15</th><th>14</th><th>13</th><th>12</th><th>11</th><th>10</th><th>9</th><th>8</th><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th></tr>
<tr><td>Read
Write</td><td colspan="16">WINHIGH</td></tr>
<tr><td>Reset</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
</table>

WDOG_WINH field descriptions

Field	Description
WINHIGH	Defines the upper 16 bits of the 32-bit window for the windowed mode of operation of the watchdog. It is defined in terms of cycles of the watchdog clock. In this mode, the watchdog can be refreshed only when the timer has reached a value greater than or equal to this window length. A refresh outside this window resets the system or if IRQRSTEN is set, it interrupts and then resets the system.

30.7.6 Watchdog Window Register Low (WDOG_WINL)

NOTE

You must set the Window Register value lower than the Time-out Value Register.

Address: 4005_2000h base + Ah offset = 4005_200Ah

<table>
<tr><th>Bit</th><th>15</th><th>14</th><th>13</th><th>12</th><th>11</th><th>10</th><th>9</th><th>8</th><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th></tr>
<tr><td>Read
Write</td><td colspan="16">WINLOW</td></tr>
<tr><td>Reset</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
</table>

WDOG_WINL field descriptions

Field	Description
WINLOW	Defines the lower 16 bits of the 32-bit window for the windowed mode of operation of the watchdog. It is defined in terms of cycles of the pre-scaled watchdog clock. In this mode, the watchdog can be refreshed only when the timer reaches a value greater than or equal to this window length value. A refresh outside of this window resets the system or if IRQRSTEN is set, it interrupts and then resets the system.

30.7.7 Watchdog Refresh register (WDOG_REFRESH)

Address: 4005_2000h base + Ch offset = 4005_200Ch

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read Write	WDOGREFRESH															
Reset	1	0	1	1	0	1	0	0	1	0	0	0	0	0	0	0

WDOG_REFRESH field descriptions

Field	Description
WDOGREFRESH	Watchdog refresh register. A sequence of 0xA602 followed by 0xB480 within 20 bus clock cycles written to this register refreshes the WDOG and prevents it from resetting the system. Writing a value other than the above mentioned sequence or if the sequence is longer than 20 bus cycles, resets the system, or if IRQRSTEN is set, it interrupts and then resets the system.

30.7.8 Watchdog Unlock register (WDOG_UNLOCK)

Address: 4005_2000h base + Eh offset = 4005_200Eh

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read Write	WDOGUNLOCK															
Reset	1	1	0	1	1	0	0	1	0	0	1	0	1	0	0	0

WDOG_UNLOCK field descriptions

Field	Description
WDOGUNLOCK	Writing the unlock sequence values to this register to makes the watchdog write-once registers writable again. The required unlock sequence is 0xC520 followed by 0xD928 within 20 bus clock cycles. A valid unlock sequence opens a window equal in length to the WCT within which you can update the registers. Writing a value other than the above mentioned sequence or if the sequence is longer than 20 bus cycles, resets the system or if IRQRSTEN is set, it interrupts and then resets the system. The unlock sequence is effective only if ALLOWUPDATE is set.

30.7.9 Watchdog Timer Output Register High (WDOG_TMROUTH)

Address: 4005_2000h base + 10h offset = 4005_2010h

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read Write	TIMEROUTHIGH															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

WDOG_TMROUTH field descriptions

Field	Description
TIMEROUTHIGH	Shows the value of the upper 16 bits of the watchdog timer.

30.7.10 Watchdog Timer Output Register Low (WDOG_TMROUTL)

During Stop mode, the WDOG_TIMER_OUT will be caught at the pre-stop value of the watchdog timer. After exiting Stop mode, a maximum delay of 1 WDOG_CLK cycle + 3 bus clock cycles will occur before the WDOG_TIMER_OUT starts following the watchdog timer.

Address: 4005_2000h base + 12h offset = 4005_2012h

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read Write	TIMEROUTLOW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

WDOG_TMROUTL field descriptions

Field	Description
TIMEROUTLOW	Shows the value of the lower 16 bits of the watchdog timer.

30.7.11 Watchdog Reset Count register (WDOG_RSTCNT)

Address: 4005_2000h base + 14h offset = 4005_2014h

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read Write	RSTCNT															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

WDOG_RSTCNT field descriptions

Field	Description
RSTCNT	Counts the number of times the watchdog resets the system. This register is reset only on a POR. Writing 1 to the bit to be cleared enables you to clear the contents of this register.

30.7.12 Watchdog Prescaler register (WDOG_PRESC)

Address: 4005_2000h base + 16h offset = 4005_2016h

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	0					PRESCVAL			0							
Write	▨	▨	▨	▨	▨				▨	▨	▨	▨	▨	▨	▨	▨
Reset	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0

WDOG_PRESC field descriptions

Field	Description
15–11 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
10–8 PRESCVAL	3-bit prescaler for the watchdog clock source. A value of zero indicates no division of the input WDOG clock. The watchdog clock is divided by (PRESCVAL + 1) to provide the prescaled WDOG_CLK.
Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

30.8 Watchdog operation with 8-bit access

30.8.1 General guideline

When performing 8-bit accesses to the watchdog's 16-bit registers where the intention is to access both the bytes of a register, place the two 8-bit accesses one after the other in your code.

30.8.2 Refresh and unlock operations with 8-bit access

One exception condition that generates a reset to the system is the write of any value other than those required for a legal refresh/update sequence to the respective refresh and unlock registers.

For an 8-bit access to these registers, writing a correct value requires at least two bus clock cycles, resulting in an invalid value in the registers for one cycle. Therefore, the system is reset even if the intention is to write a correct value to the refresh/unlock register. Keeping this in mind, the exception condition for 8-bit accesses is slightly modified.

Whereas the match for a correct value for a refresh/unlock sequence is as according to the original definition, the match for an incorrect value is done byte-wise on the refresh/unlock rather than for the whole 16-bit value. This means that if the high byte of the refresh/unlock register contains any value other than high bytes of the two values that make up the sequence, it is treated as an exception condition, leading to a reset or interrupt-then-reset. The same holds true for the lower byte of the refresh or unlock register. Take the refresh operation that expects a write of 0xA602 followed by 0xB480 to the refresh register, as an example.

Table 30-2. Refresh for 8-bit access

	WDOG_REFRESH[15:8]	WDOG_REFRESH[7:0]	Sequence value1 or value2 match	Mismatch exception
Current Value	0xB4	0x80	Value2 match	No
Write 1	0xB4	0x02	No match	No
Write 2	0xA6	0x02	Value1 match	No
Write 3	0xB4	0x02	No match	No
Write 4	0xB4	0x80	Value2 match. Sequence complete.	No
Write 5	0x02	0x80	No match	Yes

As shown in the preceding table, the refresh register holds its reset value initially. Thereafter, two 8-bit accesses are performed on the register to write the first value of the refresh sequence. No mismatch exception is registered on the intermediate write, Write1. The sequence is completed by performing two more 8-bit accesses, writing in the second value of the sequence for a successful refresh. It must be noted that the match of value2 takes place only when the complete 16-bit value is correctly written, write4. Hence, the requirement of writing value2 of the sequence within 20 bus clock cycles of value1 is checked by measuring the gap between write2 and write4.

It is reiterated that the condition for matching values 1 and 2 of the refresh or unlock sequence remains unchanged. The difference for 8-bit accesses is that the criterion for detecting a mismatch is less strict. Any 16-bit access still needs to adhere to the original guidelines, mentioned in the sections Refreshing the Watchdog.

30.9 Restrictions on watchdog operation

This section mentions some exceptions to the watchdog operation that may not be apparent to you.

- Restriction on unlock/refresh operations—In the period between the closure of the WCT window after unlock and the actual reload of the watchdog timer, unlock and refresh operations need not be attempted.
- The update and reload of the watchdog timer happens two to three watchdog clocks after WCT window closes, following a successful configuration on unlock.
- Clock Switching Delay—The watchdog uses glitch-free multiplexers at two places – one to choose between the LPO oscillator input and alternate clock input, and the other to choose between the watchdog functional clock and fast clock input for

watchdog functional test. A maximum time period of ~2 clock A cycles plus ~2 clock B cycles elapses from the time a switch is requested to the occurrence of the actual clock switch, where clock A and B are the two input clocks to the clock mux.

- For the windowed mode, there is a two to three bus clock latency between the watchdog counter going past the window value and the same registering in the bus clock domain.
- For proper operation of the watchdog, the watchdog clock must be at least five times slower than the system bus clock at all times. An exception is when the watchdog clock is synchronous to the bus clock wherein the watchdog clock can be as fast as the bus clock.
- WCT must be equivalent to at least three watchdog clock cycles. If not ensured, this means that even after the close of the WCT window, you have to wait for the synchronized system reset to deassert in the watchdog clock domain, before expecting the configuration updates to take effect.
- The time-out value of the watchdog should be set to a minimum of four watchdog clock cycles. This is to take into account the delay in new settings taking effect in the watchdog clock domain.
- You must take care not only to refresh the watchdog within the watchdog timer's actual time-out period, but also provide enough allowance for the time it takes for the refresh sequence to be detected by the watchdog timer, on the watchdog clock.
- Updates cannot be made in the bus clock cycle immediately following the write of the unlock sequence, but one bus clock cycle later.
- It should be ensured that the time-out value for the watchdog is always greater than 2xWCT time + 20 bus clock cycles.
- An attempted refresh operation, in between the two writes of the unlock sequence and in the WCT time following a successful unlock, will go undetected.
- Trying to unlock the watchdog within the WCT time after an initial unlock has no effect.
- The refresh and unlock operations and interrupt are not automatically disabled in the watchdog functional test mode.
- After emerging from a reset due to a watchdog functional test, you are still expected to go through the mandatory steps of unlocking and configuring the watchdog. The watchdog continues to be in its functional test mode and therefore you should pull the watchdog out of the functional test mode within WCT time of reset.

- After emerging from a reset due to a watchdog functional test, you still need to go through the mandatory steps of unlocking and configuring the watchdog.
- You must ensure that both the clock inputs to the glitchless clock multiplexers are alive during the switching of clocks. Failure to do so results in a loss of clock at their outputs.
- There is a gap of two to three watchdog clock cycles from the point that stop mode is entered to the watchdog timer actually pausing, due to synchronization. The same holds true for an exit from the stop mode, this time resulting in a two to three watchdog clock cycle delay in the timer restarting. In case the duration of the stop mode is less than one watchdog clock cycle, the watchdog timer is not guaranteed to pause.
- Consider the case when the first refresh value is written, following which the system enters stop mode with system bus clk still on. If the second refresh value is not written within 20 bus cycles of the first value, the system is reset, or interrupt-then-reset if enabled.

Chapter 31
MCU: Multipurpose Clock Generator (MCG)

31.1 Introduction

NOTE

For the chip-specific implementation details of this module's instances, see the chip configuration information.

The multipurpose clock generator (MCG) module provides several clock source choices for the MCU.

The module contains a frequency-locked loop (FLL) and a phase-locked loop (PLL). The FLL is controllable by either an internal or an external reference clock. The PLL is controllable by the external reference clock. The module can select either an FLL or PLL output clock, or a reference clock (internal or external) as a source for the MCU system clock. The MCG operates in conjuction with a crystal oscillator, which allows an external crystal, ceramic resonator, or another external clock source to produce the external reference clock.

31.1.1 Features

Key features of the MCG module are:

- Frequency-locked loop (FLL):
  - Digitally-controlled oscillator (DCO)
  - DCO frequency range is programmable for up to four different frequency ranges.
  - Option to program and maximize DCO output frequency for a low frequency external reference clock source.
  - Option to prevent FLL from resetting its current locked frequency when switching clock modes if FLL reference frequency is not changed.

  - Internal or external reference clock can be used as the FLL source.
  - Can be used as a clock source for other on-chip peripherals.
- Phase-locked loop (PLL):
  - Voltage-controlled oscillator (VCO)
  - External reference clock is used as the PLL source.
  - Modulo VCO frequency divider
  - Phase/Frequency detector
  - Integrated loop filter
  - Can be used as a clock source for other on-chip peripherals.
- Internal reference clock generator:
  - Slow clock with nine trim bits for accuracy
  - Fast clock with four trim bits
  - Can be used as source clock for the FLL. In FEI mode, only the slow Internal Reference Clock (IRC) can be used as the FLL source.
  - Either the slow or the fast clock can be selected as the clock source for the MCU.
  - Can be used as a clock source for other on-chip peripherals.
- Control signals for the MCG external reference low power oscillator clock generators are provided:
  - HGO0, RANGE0, EREFS0

    HGO1, RANGE1, EREFS1
- External clock from the Crystal Oscillator (OSC0):
  - Can be used as a source for the FLL and/or the PLL.
  - Can be selected as the clock source for the MCU.
- External clock from the Real Time Counter (RTC):
  - Can be used as a source for the FLL only.
  - Can be selected as the clock source for the MCU.
- External clock from the Crystal Oscillator (OSC1)
  - Can be used as a source for the PLL only.

- External clock monitor with reset and interrupt request capability to check for external clock failure when running in FBE, PEE, BLPE, or FEE modes
- Lock detector with interrupt request capability for use with the PLL
- Internal Reference Clocks Auto Trim Machine (ATM) capability using an external clock as a reference
- Reference dividers for both the FLL and the PLL are provided
- Reference dividers for the Fast Internal Reference Clock are provided
- MCG PLL Clock (MCGPLLCLK) is provided as a clock source for other on-chip peripherals
- MCG FLL Clock (MCGFLLCLK) is provided as a clock source for other on-chip peripherals
- MCG Fixed Frequency Clock (MCGFFCLK) is provided as a clock source for other on-chip peripherals
- MCG Internal Reference Clock (MCGIRCLK) is provided as a clock source for other on-chip peripherals

This figure presents the block diagram of the MCG module.

Figure 31-1. Multipurpose Clock Generator (MCG) block diagram

NOTE

Refer to the chip configuration chapter to identify the oscillator used in this MCU.

31.1.2 Modes of Operation

The MCG has the following modes of operation: FEI, FEE, FBI, FBE, PBE, PEE, BLPI, BLPE, and Stop. For details, see MCG modes of operation.

31.2 External Signal Description

There are no MCG signals that connect off chip.

31.3 Memory Map/Register Definition

This section includes the memory map and register definition.

The MCG registers can only be written when in supervisor mode. Write accesses when in user mode will result in a bus error. Read accesses may be performed in both supervisor and user mode.

MCG memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4006_4000	MCG Control 1 Register (MCG_C1)	8	R/W	04h	31.3.1/658
4006_4001	MCG Control 2 Register (MCG_C2)	8	R/W	80h	31.3.2/660
4006_4002	MCG Control 3 Register (MCG_C3)	8	R/W	Undefined	31.3.3/661
4006_4003	MCG Control 4 Register (MCG_C4)	8	R/W	See section	31.3.4/661
4006_4004	MCG Control 5 Register (MCG_C5)	8	R/W	00h	31.3.5/663
4006_4005	MCG Control 6 Register (MCG_C6)	8	R/W	00h	31.3.6/664
4006_4006	MCG Status Register (MCG_S)	8	R	10h	31.3.7/665
4006_4008	MCG Status and Control Register (MCG_SC)	8	R/W	02h	31.3.8/667
4006_400A	MCG Auto Trim Compare Value High Register (MCG_ATCVH)	8	R/W	00h	31.3.9/668
4006_400B	MCG Auto Trim Compare Value Low Register (MCG_ATCVL)	8	R/W	00h	31.3.10/ 668
4006_400C	MCG Control 7 Register (MCG_C7)	8	R/W	00h	31.3.11/ 669
4006_400D	MCG Control 8 Register (MCG_C8)	8	R/W	80h	31.3.12/ 669
4006_400F	MCG Control 10 Register (MCG_C10)	8	R/W	00h	31.3.13/ 670
4006_4011	MCG Control 12 Register (MCG_C12)	8	R/W	00h	31.3.14/ 671
4006_4012	MCG Status 2 Register (MCG_S2)	8	R/W	00h	31.3.14/ 671
4006_4013	MCG Test 3 Register (MCG_T3)	8	R/W	00h	31.3.14/ 671

31.3.1 MCG Control 1 Register (MCG_C1)

Address: 4006_4000h base + 0h offset = 4006_4000h

Bit	7	6	5	4	3	2	1	0
Read Write	CLKS		FRDIV			IREFS	IRCLKEN	IREFSTEN
Reset	0	0	0	0	0	1	0	0

MCG_C1 field descriptions

Field	Description
7–6 CLKS	Clock Source Select Selects the clock source for MCGOUTCLK . 00 01 Encoding 1 — Internal reference clock is selected. 10 Encoding 2 — External reference clock is selected. 11 Encoding 3 — Reserved.
5–3 FRDIV	FLL External Reference Divider Selects the amount to divide down the external reference clock for the FLL. The resulting frequency must be in the range 31.25 kHz to 39.0625 kHz (This is required when FLL/DCO is the clock source for MCGOUTCLK . In FBE mode, it is not required to meet this range, but it is recommended in the cases when trying to enter a FLL mode from FBE). 000 If RANGE 0 = 0 or OSCSEL=1 , Divide Factor is 1; for all other RANGE 0 values, Divide Factor is 32. 001 If RANGE 0 = 0 or OSCSEL=1 , Divide Factor is 2; for all other RANGE 0 values, Divide Factor is 64. 010 If RANGE 0 = 0 or OSCSEL=1 , Divide Factor is 4; for all other RANGE 0 values, Divide Factor is 128. 011 If RANGE 0 = 0 or OSCSEL=1 , Divide Factor is 8; for all other RANGE 0 values, Divide Factor is 256. 100 If RANGE 0 = 0 or OSCSEL=1 , Divide Factor is 16; for all other RANGE 0 values, Divide Factor is 512. 101 If RANGE 0 = 0 or OSCSEL=1 , Divide Factor is 32; for all other RANGE 0 values, Divide Factor is 1024. 110 If RANGE 0 = 0 or OSCSEL=1 , Divide Factor is 64; for all other RANGE 0 values, Divide Factor is 1280 . 111 If RANGE 0 = 0 or OSCSEL=1 , Divide Factor is 128; for all other RANGE 0 values, Divide Factor is 1536 .
2 IREFS	Internal Reference Select Selects the reference clock source for the FLL. 0 External reference clock is selected. 1 The slow internal reference clock is selected.
1 IRCLKEN	Internal Reference Clock Enable Enables the internal reference clock for use as MCGIRCLK. 0 MCGIRCLK inactive. 1 MCGIRCLK active.
0 IREFSTEN	Internal Reference Stop Enable Controls whether or not the internal reference clock remains enabled when the MCG enters Stop mode. 0 Internal reference clock is disabled in Stop mode. 1 Internal reference clock is enabled in Stop mode if IRCLKEN is set or if MCG is in FEI, FBI, or BLPI modes before entering Stop mode.

31.3.2 MCG Control 2 Register (MCG_C2)

Address: 4006_4000h base + 1h offset = 4006_4001h

Bit	7	6	5	4	3	2	1	0
Read	LOCRE0	0	RANGE0		HGO0	EREFS0	LP	IRCS
Write		▨						
Reset	1	0	0	0	0	0	0	0

MCG_C2 field descriptions

Field	Description
7 LOCRE0	Loss of Clock Reset Enable Determines whether an interrupt or a reset request is made following a loss of OSC0 external reference clock. The LOCRE0 only has an affect when CME0 is set. 0 Interrupt request is generated on a loss of OSC0 external reference clock. 1 Generate a reset request on a loss of OSC0 external reference clock.
6 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
5–4 RANGE0	Frequency Range Select Selects the frequency range for the crystal oscillator or external clock source. See the Oscillator (OSC) chapter for more details and the device data sheet for the frequency ranges used. 00 Encoding 0 — Low frequency range selected for the crystal oscillator . 01 Encoding 1 — High frequency range selected for the crystal oscillator . 1X Encoding 2 — Very high frequency range selected for the crystal oscillator .
3 HGO0	High Gain Oscillator Select Controls the crystal oscillator mode of operation. See the Oscillator (OSC) chapter for more details. 0 Configure crystal oscillator for low-power operation. 1 Configure crystal oscillator for high-gain operation.
2 EREFS0	External Reference Select Selects the source for the external reference clock. See the Oscillator (OSC) chapter for more details. 0 External reference clock requested. 1 Oscillator requested.
1 LP	Low Power Select Controls whether the FLL or PLL is disabled in BLPI and BLPE modes. In FBE or PBE modes, setting this bit to 1 will transition the MCG into BLPE mode; in FBI mode, setting this bit to 1 will transition the MCG into BLPI mode. In any other MCG mode, LP bit has no affect. 0 FLL or PLL is not disabled in bypass modes. 1 FLL or PLL is disabled in bypass modes (lower power)
0 IRCS	Internal Reference Clock Select Selects between the fast or slow internal reference clock source.

Table continues on the next page...

MCG_C2 field descriptions (continued)

Field	Description
	0 Slow internal reference clock selected. 1 Fast internal reference clock selected.

31.3.3 MCG Control 3 Register (MCG_C3)

Address: 4006_4000h base + 2h offset = 4006_4002h

Bit	7	6	5	4	3	2	1	0
Read Write	SCTRIM							
Reset	x*	x*	x*	x*	x*	x*	x*	x*

* Notes:
- x = Undefined at reset.

MCG_C3 field descriptions

Field	Description
SCTRIM	Slow Internal Reference Clock Trim Setting SCTRIM [1] controls the slow internal reference clock frequency by controlling the slow internal reference clock period. The SCTRIM bits are binary weighted, that is, bit 1 adjusts twice as much as bit 0. Increasing the binary value increases the period, and decreasing the value decreases the period. An additional fine trim bit is available in C4 register as the SCFTRIM bit. Upon reset, this value is loaded with a factory trim value. If an SCTRIM value stored in nonvolatile memory is to be used, it is your responsibility to copy that value from the nonvolatile memory location to this register.

[1] A value for SCTRIM is loaded during reset from a factory programmed location.

31.3.4 MCG Control 4 Register (MCG_C4)

Address: 4006_4000h base + 3h offset = 4006_4003h

Bit	7	6	5	4	3	2	1	0
Read Write	DMX32	DRST_DRS		FCTRIM				SCFTRIM
Reset	0	0	0	x*	x*	x*	x*	x*

* Notes:
- x = Undefined at reset.

MCG_C4 field descriptions

Field	Description
7 DMX32	DCO Maximum Frequency with 32.768 kHz Reference The DMX32 bit controls whether the DCO frequency range is narrowed to its maximum frequency with a 32.768 kHz reference.

Table continues on the next page...

MCG_C4 field descriptions (continued)

<table>
<tr><th>Field</th><th>Description</th></tr>
<tr><td></td><td>The following table identifies settings for the DCO frequency range.

<b>NOTE:</b> The system clocks derived from this source should not exceed their specified maximums.

<table>
<tr><td>DRST_DRS</td><td>DMX32</td><td>Reference Range</td><td>FLL Factor</td><td>DCO Range</td></tr>
<tr><td>00</td><td>0</td><td>31.25–39.0625 kHz</td><td>640</td><td>20–25 MHz</td></tr>
<tr><td></td><td>1</td><td>32.768 kHz</td><td>732</td><td>24 MHz</td></tr>
<tr><td>01</td><td>0</td><td>31.25–39.0625 kHz</td><td>1280</td><td>40–50 MHz</td></tr>
<tr><td></td><td>1</td><td>32.768 kHz</td><td>1464</td><td>48 MHz</td></tr>
<tr><td>10</td><td>0</td><td>31.25–39.0625 kHz</td><td>1920</td><td>60–75 MHz</td></tr>
<tr><td></td><td>1</td><td>32.768 kHz</td><td>2197</td><td>72 MHz</td></tr>
<tr><td>11</td><td>0</td><td>31.25–39.0625 kHz</td><td>2560</td><td>80–100 MHz</td></tr>
<tr><td></td><td>1</td><td>32.768 kHz</td><td>2929</td><td>96 MHz</td></tr>
</table>

0 DCO has a default range of 25%.
1 DCO is fine-tuned for maximum frequency with 32.768 kHz reference.</td></tr>
<tr><td>6–5
DRST_DRS</td><td>DCO Range Select

The DRS bits select the frequency range for the FLL output, DCOOUT. When the LP bit is set, writes to the DRS bits are ignored. The DRST read field indicates the current frequency range for DCOOUT. The DRST field does not update immediately after a write to the DRS field due to internal synchronization between clock domains. See the DCO Frequency Range table for more details.

00 Encoding 0 — Low range (reset default).
01 Encoding 1 — Mid range.
10 Encoding 2 — Mid-high range.
11 Encoding 3 — High range.</td></tr>
<tr><td>4–1
FCTRIM</td><td>Fast Internal Reference Clock Trim Setting

FCTRIM [1] controls the fast internal reference clock frequency by controlling the fast internal reference clock period. The FCTRIM bits are binary weighted, that is, bit 1 adjusts twice as much as bit 0. Increasing the binary value increases the period, and decreasing the value decreases the period.

If an FCTRIM[3:0] value stored in nonvolatile memory is to be used, it is your responsibility to copy that value from the nonvolatile memory location to this register.</td></tr>
<tr><td>0
SCFTRIM</td><td>Slow Internal Reference Clock Fine Trim

SCFTRIM [2] controls the smallest adjustment of the slow internal reference clock frequency. Setting SCFTRIM increases the period and clearing SCFTRIM decreases the period by the smallest amount possible.

If an SCFTRIM value stored in nonvolatile memory is to be used, it is your responsibility to copy that value from the nonvolatile memory location to this bit.</td></tr>
</table>

[1] A value for FCTRIM is loaded during reset from a factory programmed location.

[2] A value for SCFTRIM is loaded during reset from a factory programmed location.

31.3.5 MCG Control 5 Register (MCG_C5)

Address: 4006_4000h base + 4h offset = 4006_4004h

Bit	7	6	5	4	3	2	1	0
Read	0	PLLCLKEN0	PLLSTEN0	PRDIV0				
Write	▨							
Reset	0	0	0	0	0	0	0	0

MCG_C5 field descriptions

Field	Description
7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
6 PLLCLKEN0	PLL Clock Enable Enables the PLL independent of PLLS and enables the PLL clock for use as MCGPLLCLK. (PRDIV 0 needs to be programmed to the correct divider to generate a PLL reference clock in the range of 2 - 4 MHz range prior to setting the PLLCLKEN 0 bit). Setting PLLCLKEN 0 will enable the external oscillator if not already enabled. Whenever the PLL is being enabled by means of the PLLCLKEN 0 bit, and the external oscillator is being used as the reference clock, the OSCINIT 0 bit should be checked to make sure it is set. 0 MCGPLLCLK is inactive. 1 MCGPLLCLK is active.
5 PLLSTEN0	PLL Stop Enable Enables the PLL Clock during Normal Stop. In Low Power Stop mode, the PLL clock gets disabled even if PLLSTEN 0 =1. All other power modes, PLLSTEN 0 bit has no affect and does not enable the PLL Clock to run if it is written to 1. 0 MCGPLLCLK is disabled in any of the Stop modes. 1 MCGPLLCLK is enabled if system is in Normal Stop mode.
PRDIV0	PLL External Reference Divider Selects the amount to divide down the external reference clock for the PLL. The resulting frequency must be in the range of 2 MHz to 4 MHz. After the PLL is enabled (by setting either PLLCLKEN 0 or PLLS), the PRDIV 0 value must not be changed when LOCK0 is zero.

Table 31-1. PLL External Reference Divide Factor

PRDIV 0	Divide Factor		PRDIV 0	Divide Factor		PRDIV 0	Divide Factor		PRDIV 0	Divide Factor
00000	1		01000	9		10000	17		11000	25
00001	2		01001	10		10001	18		11001	Reserved
00010	3		01010	11		10010	19		11010	Reserved
00011	4		01011	12		10011	20		11011	Reserved

Table continues on the next page...

MCG_C5 field descriptions (continued)

Field	Description
	Table 31-1. PLL External Reference Divide Factor (continued)

00100	5		01100	13		10100	21		11100	Reserved
00101	6		01101	14		10101	22		11101	Reserved
00110	7		01110	15		10110	23		11110	Reserved
00111	8		01111	16		10111	24		11111	Reserved

31.3.6 MCG Control 6 Register (MCG_C6)

Address: 4006_4000h base + 5h offset = 4006_4005h

Bit	7	6	5	4	3	2	1	0
Read Write	LOLIE0	PLLS	CME0	VDIV0				
Reset	0	0	0	0	0	0	0	0

MCG_C6 field descriptions

Field	Description
7 LOLIE0	Loss of Lock Interrrupt Enable Determines if an interrupt request is made following a loss of lock indication. This bit only has an effect when LOLS 0 is set. 0 No interrupt request is generated on loss of lock. 1 Generate an interrupt request on loss of lock.
6 PLLS	PLL Select Controls whether the PLL or FLL output is selected as the MCG source when CLKS[1:0]=00. If the PLLS bit is cleared and PLLCLKEN 0 is not set, the PLL is disabled in all modes. If the PLLS is set, the FLL is disabled in all modes. 0 FLL is selected. 1 PLL is selected (PRDIV 0 need to be programmed to the correct divider to generate a PLL reference clock in the range of 2–4 MHz prior to setting the PLLS bit).
5 CME0	Clock Monitor Enable Enables the loss of clock monitoring circuit for the OSC0 external reference mux select. The LOCRE0 bit will determine if a interrupt or a reset request is generated following a loss of OSC0 indication. The CME0 bit must only be set to a logic 1 when the MCG is in an operational mode that uses the external clock (FEE, FBE, PEE, PBE, or BLPE) . Whenever the CME0 bit is set to a logic 1, the value of the RANGE0 bits in the C2 register should not be changed. CME0 bit should be set to a logic 0 before the MCG enters

Table continues on the next page...

MCG_C6 field descriptions (continued)

Field	Description
	any Stop mode. Otherwise, a reset request may occur while in Stop mode. CME0 should also be set to a logic 0 before entering VLPR or VLPW power modes if the MCG is in BLPE mode. 0 External clock monitor is disabled for OSC0. 1 External clock monitor is enabled for OSC0.
VDIV0	VCO 0 Divider Selects the amount to divide the VCO output of the PLL. The VDIV 0 bits establish the multiplication factor (M) applied to the reference clock frequency. After the PLL is enabled (by setting either PLLCLKEN 0 or PLLS), the VDIV 0 value must not be changed when LOCK 0 is zero.

Table 31-2. PLL VCO Divide Factor

VDIV 0	Multiply Factor		VDIV 0	Multiply Factor		VDIV 0	Multiply Factor		VDIV 0	Multiply Factor
00000	24		01000	32		10000	40		11000	48
00001	25		01001	33		10001	41		11001	49
00010	26		01010	34		10010	42		11010	50
00011	27		01011	35		10011	43		11011	51
00100	28		01100	36		10100	44		11100	52
00101	29		01101	37		10101	45		11101	53
00110	30		01110	38		10110	46		11110	54
00111	31		01111	39		10111	47		11111	55

31.3.7 MCG Status Register (MCG_S)

Address: 4006_4000h base + 6h offset = 4006_4006h

Bit	7	6	5	4	3	2	1	0
Read	LOLS0	LOCK0	PLLST	IREFST	CLKST		OSCINIT0	IRCST
Write								
Reset	0	0	0	1	0	0	0	0

MCG_S field descriptions

Field	Description
7 LOLS0	Loss of Lock Status This bit is a sticky bit indicating the lock status for the PLL. LOLS is set if after acquiring lock, the PLL output frequency has fallen outside the lock exit frequency tolerance, D $_{unl}$. LOLIE determines whether an interrupt request is made when LOLS is set. LOLRE determines whether a reset request is made when LOLS is set. This bit is cleared by reset or by writing a logic 1 to it when set. Writing a logic 0 to this bit has no effect.

Table continues on the next page...

MCG_S field descriptions (continued)

Field	Description
	0 PLL has not lost lock since LOLS 0 was last cleared. 1 PLL has lost lock since LOLS 0 was last cleared.
6 LOCK0	Lock Status This bit indicates whether the PLL has acquired lock. Lock detection is only enabled when the PLL is enabled (either through clock mode selection or PLLCLKEN0=1 setting). While the PLL clock is locking to the desired frequency, the MCG PLL clock (MCGPLLCLK) will be gated off until the LOCK bit gets asserted. If the lock status bit is set, changing the value of the PRDIV0 [4:0] bits in the C5 register or the VDIV0[4:0] bits in the C6 register causes the lock status bit to clear and stay cleared until the PLL has reacquired lock. Loss of PLL reference clock will also cause the LOCK0 bit to clear until the PLL has reacquired lock. Entry into LLS, VLPS, or regular Stop with PLLSTEN=0 also causes the lock status bit to clear and stay cleared until the Stop mode is exited and the PLL has reacquired lock. Any time the PLL is enabled and the LOCK0 bit is cleared, the MCGPLLCLK will be gated off until the LOCK0 bit is asserted again. 0 PLL is currently unlocked. 1 PLL is currently locked.
5 PLLST	PLL Select Status This bit indicates the clock source selected by PLLS . The PLLST bit does not update immediately after a write to the PLLS bit due to internal synchronization between clock domains. 0 Source of PLLS clock is FLL clock. 1 Source of PLLS clock is PLL output clock.
4 IREFST	Internal Reference Status This bit indicates the current source for the FLL reference clock. The IREFST bit does not update immediately after a write to the IREFS bit due to internal synchronization between clock domains. 0 Source of FLL reference clock is the external reference clock. 1 Source of FLL reference clock is the internal reference clock.
3–2 CLKST	Clock Mode Status These bits indicate the current clock mode. The CLKST bits do not update immediately after a write to the CLKS bits due to internal synchronization between clock domains. 00 Encoding 0 — Output of the FLL is selected (reset default). 01 Encoding 1 — Internal reference clock is selected. 10 Encoding 2 — External reference clock is selected. 11 Encoding 3 — Output of the PLL is selected.
1 OSCINIT0	OSC Initialization This bit, which resets to 0, is set to 1 after the initialization cycles of the crystal oscillator clock have completed. After being set, the bit is cleared to 0 if the OSC is subsequently disabled. See the OSC module's detailed description for more information.
0 IRCST	Internal Reference Clock Status The IRCST bit indicates the current source for the internal reference clock select clock (IRCSCLK). The IRCST bit does not update immediately after a write to the IRCS bit due to internal synchronization between clock domains. The IRCST bit will only be updated if the internal reference clock is enabled, either by the MCG being in a mode that uses the IRC or by setting the C1[IRCLKEN] bit .

Table continues on the next page...

MCG_S field descriptions (continued)

Field	Description
	0 Source of internal reference clock is the slow clock (32 kHz IRC). 1 Source of internal reference clock is the fast clock (4 MHz IRC).

31.3.8 MCG Status and Control Register (MCG_SC)

Address: 4006_4000h base + 8h offset = 4006_4008h

<table>
<tr><td>Bit</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>Read</td><td rowspan="2">ATME</td><td rowspan="2">ATMS</td><td>ATMF</td><td rowspan="2">FLTPRSRV</td><td colspan="3" rowspan="2">FCRDIV</td><td>LOCS0</td></tr>
<tr><td>Write</td><td>w1c</td><td>w1c</td></tr>
<tr><td>Reset</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td></tr>
</table>

MCG_SC field descriptions

Field	Description
7 ATME	Automatic Trim Machine Enable Enables the Auto Trim Machine to start automatically trimming the selected Internal Reference Clock. **NOTE:** ATME deasserts after the Auto Trim Machine has completed trimming all trim bits of the IRCS clock selected by the ATMS bit. Writing to C1, C3, C4, and SC registers or entering Stop mode aborts the auto trim operation and clears this bit. 0 Auto Trim Machine disabled. 1 Auto Trim Machine enabled.
6 ATMS	Automatic Trim Machine Select Selects the IRCS clock for Auto Trim Test. 0 32 kHz Internal Reference Clock selected. 1 4 MHz Internal Reference Clock selected.
5 ATMF	Automatic Trim Machine Fail Flag Fail flag for the Automatic Trim Machine (ATM). This bit asserts when the Automatic Trim Machine is enabled, ATME=1, and a write to the C1, C3, C4, and SC registers is detected or the MCG enters into any Stop mode. A write to ATMF clears the flag. 0 Automatic Trim Machine completed normally. 1 Automatic Trim Machine failed.
4 FLTPRSRV	FLL Filter Preserve Enable This bit will prevent the FLL filter values from resetting allowing the FLL output frequency to remain the same during clock mode changes where the FLL/DCO output is still valid. (Note: This requires that the FLL reference frequency to remain the same as what it was prior to the new clock mode switch. Otherwise FLL filter and frequency values will change.) 0 FLL filter and FLL frequency will reset on changes to currect clock mode. 1 Fll filter and FLL frequency retain their previous values during new clock mode change.

Table continues on the next page...

MCG_SC field descriptions (continued)

Field	Description
3–1 FCRDIV	Fast Clock Internal Reference Divider Selects the amount to divide down the fast internal reference clock. The resulting frequency will be in the range 31.25 kHz to 4 MHz (Note: Changing the divider when the Fast IRC is enabled is not supported). 000 Divide Factor is 1 001 Divide Factor is 2. 010 Divide Factor is 4. 011 Divide Factor is 8. 100 Divide Factor is 16 101 Divide Factor is 32 110 Divide Factor is 64 111 Divide Factor is 128.
0 LOCS0	OSC0 Loss of Clock Status The LOCS0 indicates when a loss of OSC0 reference clock has occurred. The LOCS0 bit only has an effect when CME0 is set. This bit is cleared by writing a logic 1 to it when set. 0 Loss of OSC0 has not occurred. 1 Loss of OSC0 has occurred.

31.3.9 MCG Auto Trim Compare Value High Register (MCG_ATCVH)

Address: 4006_4000h base + Ah offset = 4006_400Ah

Bit	7	6	5	4	3	2	1	0
Read Write	ATCVH							
Reset	0	0	0	0	0	0	0	0

MCG_ATCVH field descriptions

Field	Description
ATCVH	ATM Compare Value High Values are used by Auto Trim Machine to compare and adjust Internal Reference trim values during ATM SAR conversion.

31.3.10 MCG Auto Trim Compare Value Low Register (MCG_ATCVL)

Address: 4006_4000h base + Bh offset = 4006_400Bh

Bit	7	6	5	4	3	2	1	0
Read Write	ATCVL							
Reset	0	0	0	0	0	0	0	0

MCG_ATCVL field descriptions

Field	Description
ATCVL	ATM Compare Value Low Values are used by Auto Trim Machine to compare and adjust Internal Reference trim values during ATM SAR conversion.

31.3.11 MCG Control 7 Register (MCG_C7)

Address: 4006_4000h base + Ch offset = 4006_400Ch

Bit	7	6	5	4	3	2	1	0
Read	0		0				0	OSCSEL
Write								
Reset	0	0	0	0	0	0	0	0

MCG_C7 field descriptions

Field	Description
7–6 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
5–2 Reserved	Reserved This field is reserved. This read-only field is reserved and always has the value 0.
1 Reserved	Reserved This field is reserved. This read-only field is reserved and always has the value 0.
0 OSCSEL	MCG OSC Clock Select Selects the MCG FLL external reference clock 0 Selects Oscillator (OSCCLK). 1 Selects 32 kHz RTC Oscillator.

31.3.12 MCG Control 8 Register (MCG_C8)

Address: 4006_4000h base + Dh offset = 4006_400Dh

Bit	7	6	5	4	3	2	1	0
Read	LOCRE1	LOLRE	CME1	0				LOCS1
Write								w1c
Reset	1	0	0	0	0	0	0	0

MCG_C8 field descriptions

Field	Description
7 LOCRE1	Loss of Clock Reset Enable Determines if a interrupt or a reset request is made following a loss of RTC external reference clock. The LOCRE1 only has an affect when CME1 is set. 0 Interrupt request is generated on a loss of RTC external reference clock. 1 Generate a reset request on a loss of RTC external reference clock
6 LOLRE	PLL Loss of Lock Reset Enable Determines if an interrupt or a reset request is made following a PLL loss of lock. 0 Interrupt request is generated on a PLL loss of lock indication. The PLL loss of lock interrupt enable bit must also be set to generate the interrupt request. 1 Generate a reset request on a PLL loss of lock indication.
5 CME1	Clock Monitor Enable1 Enables the loss of clock monitoring circuit for the output of the RTC external reference clock. The LOCRE1 bit will determine whether an interrupt or a reset request is generated following a loss of RTC clock indication. The CME1 bit should be set to a logic 1 when the MCG is in an operational mode that uses the RTC as its external reference clock or if the RTC is operational. CME1 bit must be set to a logic 0 before the MCG enters any Stop mode. Otherwise, a reset request may occur when in Stop mode. CME1 should also be set to a logic 0 before entering VLPR or VLPW power modes. 0 External clock monitor is disabled for RTC clock. 1 External clock monitor is enabled for RTC clock.
4–1 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
0 LOCS1	RTC Loss of Clock Status This bit indicates when a loss of clock has occurred. This bit is cleared by writing a logic 1 to it when set. 0 Loss of RTC has not occur. 1 Loss of RTC has occur

31.3.13 MCG Control 10 Register (MCG_C10)

Address: 4006_4000h base + Fh offset = 4006_400Fh

Bit	7	6	5	4	3	2	1	0
Read	0				0			
Write								
Reset	0	0	0	0	0	0	0	0

MCG_C10 field descriptions

Field	Description
7–4 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

31.3.14 MCG Control 12 Register (MCG_C12)

Address: 4006_4000h base + 11h offset = 4006_4011h

Bit	7	6	5	4	3	2	1	0
Read	0							
Write								
Reset	0	0	0	0	0	0	0	0

MCG_C12 field descriptions

Field	Description
Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

31.3.14 MCG Status 2 Register (MCG_S2)

Address: 4006_4000h base + 12h offset = 4006_4012h

Bit	7	6	5	4	3	2	1	0
Read	0							
Write								
Reset	0	0	0	0	0	0	0	0

MCG_S2 field descriptions

Field	Description
Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

31.3.14 MCG Test 3 Register (MCG_T3)

Address: 4006_4000h base + 13h offset = 4006_4013h

Bit	7	6	5	4	3	2	1	0
Read	0							
Write								
Reset	0	0	0	0	0	0	0	0

MCG_T3 field descriptions

Field	Description
Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

31.4 Functional description

31.4.1 MCG mode state diagram

The nine states of the MCG are shown in the following figure and are described in Table 31-3. The arrows indicate the permitted MCG mode transitions.

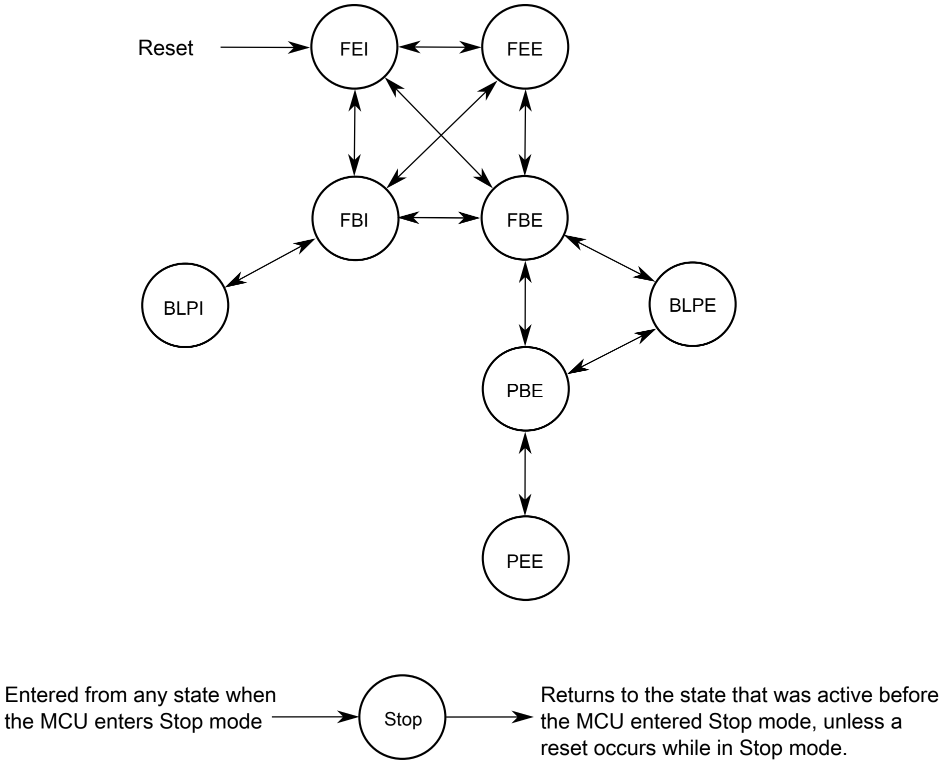

Reset
FEI
FEE
FBI
FBE
BLPI
BLPE
PBE
PEE
Entered from any state when the MCU enters Stop mode
Stop
Returns to the state that was active before the MCU entered Stop mode, unless a reset occurs while in Stop mode.

Figure 31-2. MCG mode state diagram

NOTE

- During exits from LLS or VLPS when the MCG is in PEE mode, the MCG will reset to PBE clock mode and the C1[CLKS] and S[CLKST] will automatically be set to 2’b10.
- If entering Normal Stop mode when the MCG is in PEE mode with PLLSTEN=0, the MCG will reset to PBE clock mode and C1[CLKS] and S[CLKST] will automatically be set to 2’b10.

31.4.1.1 MCG modes of operation

The MCG operates in one of the following modes.

Note

> The MCG restricts transitions between modes. For the permitted transitions, see Figure 31-2.

Table 31-3. MCG modes of operation

Mode	Description
FLL Engaged Internal (FEI)	FLL engaged internal (FEI) is the default mode of operation and is entered when all the following condtions occur: • 00 is written to C1[CLKS]. • 1 is written to C1[IREFS]. • 0 is written to C6[PLLS]. In FEI mode, MCGOUTCLK is derived from the FLL clock (DCOCLK) that is controlled by the 32 kHz Internal Reference Clock (IRC). The FLL loop will lock the DCO frequency to the FLL factor, as selected by C4[DRST_DRS] and C4[DMX32] bits, times the internal reference frequency. See the C4[DMX32] bit description for more details. In FEI mode, disabled in a low-power state unless C5[PLLCLKEN] is set .
FLL Engaged External (FEE)	FLL engaged external (FEE) mode is entered when all the following conditions occur: • 00 is written to C1[CLKS]. • 0 is written to C1[IREFS]. • C1[FRDIV] must be written to divide external reference clock to be within the range of 31.25 kHz to 39.0625 kHz • 0 is written to C6[PLLS]. In FEE mode, MCGOUTCLK is derived from the FLL clock (DCOCLK) that is controlled by the external reference clock. The FLL loop will lock the DCO frequency to the FLL factor, as selected by C4[DRST_DRS] and C4[DMX32] bits, times the external reference frequency, as specified by C1[FRDIV] and C2[RANGE0]. See the C4[DMX32] bit description for more details. In FEE mode, disabled in a low-power state unless C5[PLLCLKEN] is set .
FLL Bypassed Internal (FBI)	FLL bypassed internal (FBI) mode is entered when all the following conditions occur: • 01 is written to C1[CLKS]. • 1 is written to C1[IREFS]. • 0 is written to C6[PLLS] • 0 is written to C2[LP]. In FBI mode, the MCGOUTCLK is derived either from the slow (32 kHz IRC) or fast (4 MHz IRC) internal reference clock, as selected by the C2[IRCS] bit. The FLL is operational but its output is not used. This mode is useful to allow the FLL to acquire its target frequency while the MCGOUTCLK is driven from the C2[IRCS] selected internal reference clock. The FLL clock (DCOCLK) is controlled by the slow internal reference clock, and the DCO clock frequency locks to a multiplication factor, as selected by C4[DRST_DRS] and C4[DMX32] bits, times the internal reference frequency. See the C4[DMX32] bit description for more details. In FBI mode, disabled in a low-power state unless C5[PLLCLKEN] is set .

Table continues on the next page...

Table 31-3. MCG modes of operation (continued)

Mode	Description
FLL Bypassed External (FBE)	FLL bypassed external (FBE) mode is entered when all the following conditions occur: • 10 is written to C1[CLKS]. • 0 is written to C1[IREFS]. • C1[FRDIV] must be written to divide external reference clock to be within the range of 31.25 kHz to 39.0625 kHz. • 0 is written to C6[PLLS]. • 0 is written to C2[LP]. In FBE mode, the MCGOUTCLK is derived from the OSCSEL external reference clock. The FLL is operational but its output is not used. This mode is useful to allow the FLL to acquire its target frequency while the MCGOUTCLK is driven from the external reference clock. The FLL clock (DCOCLK) is controlled by the external reference clock, and the DCO clock frequency locks to a multiplication factor, as selected by C4[DRST_DRS] and C4[DMX32] bits, times the divided external reference frequency. See the C4[DMX32] bit description for more details. In FBE mode, disabled in a low-power state unless C5[PLLCLKEN] is set .
PLL Engaged External (PEE)	PLL Engaged External (PEE) mode is entered when all the following conditions occur: • 00 is written to C1[CLKS]. • 0 is written to C1[IREFS]. • 1 is written to C6[PLLS]. In PEE mode, the MCGOUTCLK is derived from the output of PLL controlled by a external reference clock. The PLL clock frequency locks to a multiplication factor, as specified by its corresponding VDIV, times the selected PLL reference frequency, as specified by its corresponding PRDIV. The PLL's programmable reference divider must be configured to produce a valid PLL reference clock. The FLL is disabled in a low-power state.
PLL Bypassed External (PBE)	PLL Bypassed External (PBE) mode is entered when all the following conditions occur: • 10 is written to C1[CLKS]. • 0 is written to C1[IREFS]. • 1 is written to C6[PLLS]. • 0 is written to C2[LP]. In PBE mode, MCGOUTCLK is derived from the OSCSEL external reference clock; the PLL is operational, but its output clock is not used. This mode is useful to allow the PLL to acquire its target frequency while MCGOUTCLK is driven from the external reference clock. The PLL clock frequency locks to a multiplication factor, as specified by its [VDIV], times the PLL reference frequency, as specified by its [PRDIV]. In preparation for transition to PEE, the PLL's programmable reference divider must be configured to produce a valid PLL reference clock. The FLL is disabled in a low-power state.
Bypassed Low Power Internal (BLPI)	Bypassed Low Power Internal (BLPI) mode is entered when all the following conditions occur: • 01 is written to C1[CLKS]. • 1 is written to C1[IREFS]. • 0 is written to C6[PLLS]. • 1 is written to C2[LP].

Table continues on the next page...

Table 31-3. MCG modes of operation (continued)

Mode	Description
	In BLPI mode, MCGOUTCLK is derived from the internal reference clock. The FLL is disabled and PLL is disabled even if C5[PLLCLKEN] is set to 1.
Bypassed Low Power External (BLPE)	Bypassed Low Power External (BLPE) mode is entered when all the following conditions occur: • 10 is written to C1[CLKS]. • 0 is written to C1[IREFS]. • 1 is written to C2[LP]. In BLPE mode, MCGOUTCLK is derived from the OSCSEL external reference clock. The FLL is disabled and PLL is disabled even if the C5[PLLCLKEN] is set to 1.
Stop	Entered whenever the MCU enters a Stop state. The power modes are chip specific. For power mode assignments, see the chapter that describes how modules are configured and MCG behavior during Stop recovery. Entering Stop mode, the FLL is disabled, and all MCG clock signals are static except in the following case: MCGPLLCLK is active in Normal Stop mode when PLLSTEN=1 MCGPLL1CLK is active in Normal Stop mode when PLLSTEN1=1 MCGIRCLK is active in Normal Stop mode when all the following conditions become true: • C1[IRCLKEN] = 1 • C1[IREFSTEN] = 1 **NOTE:** • When entering Low Power Stop modes (LLS or VLPS) from PEE mode, on exit the MCG clock mode is forced to PBE clock mode. C1[CLKS] and S[CLKST] will be configured to 2’b10 and S[LOCK] bit will be cleared without setting S[LOLS]. • When entering Normal Stop mode from PEE mode and if C5[PLLSTEN]=0, on exit the MCG clock mode is forced to PBE mode, the C1[CLKS] and S[CLKST] will be configured to 2’b10 and S[LOCK] bit will clear without setting S[LOLS]. If C5[PLLSTEN]=1, the S[LOCK] bit will not get cleared and on exit the MCG will continue to run in PEE mode.

NOTE

For the chip-specific modes of operation, see the power management chapter of this MCU.

31.4.1.2 MCG mode switching

C1[IREFS] can be changed at any time, but the actual switch to the newly selected reference clocks is shown by S[IREFST]. When switching between engaged internal and engaged external modes, the FLL will begin locking again after the switch is completed.

C1[CLKS] can also be changed at any time, but the actual switch to the newly selected clock is shown by S[CLKST]. If the newly selected clock is not available, the previous clock will remain selected.

The C4[DRST_DRS] write bits can be changed at any time except when C2[LP] bit is 1. If C4[DRST_DRS] write bits are changed while in FLL engaged internal (FEI) or FLL engaged external (FEE) mode, the MCGOUTCLK switches to the new selected DCO range within three clocks of the selected DCO clock. After switching to the new DCO (indicated by the updated C4[DRST_DRS] read bits), the FLL remains unlocked for several reference cycles. The FLL lock time is provided in the device data sheet as $t_{fll\_acquire}$.

31.4.2 Low-power bit usage

C2[LP] is provided to allow the FLL or PLL to be disabled and thus conserve power when these systems are not being used. C4[DRST_DRS] can not be written while C2[LP] is 1. However, in some applications, it may be desirable to enable the FLL or PLL and allow it to lock for maximum accuracy before switching to an engaged mode. Do this by writing 0 to C2[LP].

31.4.3 MCG Internal Reference Clocks

This module supports two internal reference clocks with nominal frequencies of 32 kHz (slow IRC) and 4 MHz (fast IRC). The fast IRC frequency can be divided down by programming of the FCRDIV to produce a frequency range of 32 kHz to 4 MHz.

31.4.3.1 MCG Internal Reference Clock

The MCG Internal Reference Clock (MCGIRCLK) provides a clock source for other on-chip peripherals and is enabled when C1[IRCLKEN]=1. When enabled, MCGIRCLK is driven by either the fast internal reference clock (4 MHz IRC which can be divided down by the FRDIV factors) or the slow internal reference clock (32 kHz IRC). The IRCS clock frequency can be re-targeted by trimming the period of its IRCS selected internal reference clock. This can be done by writing a new trim value to the C3[SCTRIM]:C4[SCFTRIM] bits when the slow IRC clock is selected or by writing a new trim value to C4[FCTRIM] when the fast IRC clock is selected. The internal reference clock period is proportional to the trim value written. C3[SCTRIM]:C4[SCFTRIM] (if C2[IRCS]=0) and C4[FCTRIM] (if C2[IRCS]=1) bits affect the MCGOUTCLK frequency if the MCG is in FBI or BLPI modes. C3[SCTRIM]:C4[SCFTRIM] (if C2[IRCS]=0) bits also affect the MCGOUTCLK frequency if the MCG is in FEI mode.

Additionally, this clock can be enabled in Stop mode by setting C1[IRCLKEN] and C1[IREFSTEN], otherwise this clock is disabled in Stop mode.

31.4.4 External Reference Clock

The MCG module can support an external reference clock in all modes. See the device datasheet for external reference frequency range. When C1[IREFS] is set, the external reference clock will not be used by the FLL or PLL. In these modes, the frequency can be equal to the maximum frequency the chip-level timing specifications will support.

If any of the CME bits are asserted the slow internal reference clock is enabled along with the enabled external clock monitor. For the case when C6[CME0]=1, a loss of clock is detected if the OSC0 external reference falls below a minimum frequency (f_{loc_high} or f_{loc_low} depending on C2[RANGE0]). For the case when C8[CME1]=1, a loss of clock is detected if the RTC external reference falls below a minimum frequency (f_{loc_low}).

NOTE

All clock monitors must be disabled before entering these low-power modes: Stop, VLPS, VLPR, VLPW, LLS, and VLLSx.

On detecting a loss-of-clock event, the MCU generates a system reset if the respective LOCRE bit is set. Otherwise the MCG sets the respective LOCS bit and the MCG generates a LOCS interrupt request. In the case where a OSC loss of clock is detected, the PLL LOCK status bit is cleared.

31.4.5 MCG Fixed Frequency Clock

The MCG Fixed Frequency Clock (MCGFFCLK) provides a fixed frequency clock source for other on-chip peripherals; see the block diagram. This clock is driven by either the slow clock from the internal reference clock generator or the external reference clock from the Crystal Oscillator, divided by the FLL reference clock divider. The source of MCGFFCLK is selected by C1[IREFS].

This clock is synchronized to the peripheral bus clock and is valid only when its frequency is not more than 1/8 of the MCGOUTCLK frequency. When it is not valid, it is disabled and held high. The MCGFFCLK is not available when the MCG is in BLPI mode. This clock is also disabled in Stop mode. The FLL reference clock must be set within the valid frequency range for the MCGFFCLK.

31.4.6 MCG PLL clock

The MCG PLL Clock (MCGPLLCLK) is available depending on the device's configuration of the MCG module. For more details, see the clock distribution chapter of this MCU. The MCGPLLCLK is prevented from coming out of the MCG until it is enabled and S[LOCK0] is set.

31.4.7 MCG Auto TRIM (ATM)

The MCG Auto Trim (ATM) is a MCG feature that when enabled, it configures the MCG hardware to automatically trim the MCG Internal Reference Clocks using an external clock as a reference. The selection between which MCG IRC clock gets tested and enabled is controlled by the ATC[ATMS] control bit (ATC[ATMS]=0 selects the 32 kHz IRC and ATC[ATMS]=1 selects the 4 MHz IRC). If 4 MHz IRC is selected for the ATM, a divide by 128 is enabled to divide down the 4 MHz IRC to a range of 31.250 kHz.

When MCG ATM is enabled by writing ATC[ATME] bit to 1, The ATM machine will start auto trimming the selected IRC clock. During the autotrim process, ATC[ATME] will remain asserted and will deassert after ATM is completed or an abort occurs. The MCG ATM is aborted if a write to any of the following control registers is detected : C1, C3, C4, or ATC or if Stop mode is entered. If an abort occurs, ATC[ATMF] fail flag is asserted.

The ATM machine uses the bus clock as the external reference clock to perform the IRC auto-trim. Therefore, it is required that the MCG is configured in a clock mode where the reference clock used to generate the system clock is the external reference clock such as FBE clock mode. The MCG must not be configured in a clock mode where selected IRC ATM clock is used to generate the system clock. The bus clock is also required to be running with in the range of 8–16 MHz.

To perform the ATM on the selected IRC, the ATM machine uses the successive approximation technique to adjust the IRC trim bits to generate the desired IRC trimmed frequency. The ATM SARs each of the ATM IRC trim bits starting with the MSB. For each trim bit test, the ATM uses a pulse that is generated by the ATM selected IRC clock to enable a counter that counts number of ATM external clocks. At end of each trim bit, the ATM external counter value is compared to the ATCV[15:0] register value. Based on the comparison result, the ATM trim bit under test will get cleared or stay asserted. This is done until all trim bits have been tested by ATM SAR machine.

Before the ATM can be enabled, the ATM expected count needs to be derived and stored into the ATCV register. The ATCV expected count is derived based on the required target Internal Reference Clock (IRC) frequency, and the frequency of the external reference clock using the following formula:

$$\text{ATCV Expected Count Value} = 21 * (Fe/Fr)$$

- Fr = Target Internal Reference Clock (IRC) Trimmed Frequency
- Fe = External Clock Frequency

If the auto trim is being performed on the 4 MHz IRC, the calculated expected count value must be multiplied by 128 before storing it in the ATCV register. Therefore, the ATCV Expected Count Value for trimming the 4 MHz IRC is calculated using the following formula.

$$\text{Expected Count Value} = (Fe/Fr) * 21 * (128)$$

31.5 Initialization / Application information

This section describes how to initialize and configure the MCG module in an application.

The following sections include examples on how to initialize the MCG and properly switch between the various available modes.

31.5.1 MCG module initialization sequence

The MCG comes out of reset configured for FEI mode.

The internal reference will stabilize in $t_{irefsts}$ microseconds before the FLL can acquire lock. As soon as the internal reference is stable, the FLL will acquire lock in $t_{fll\_acquire}$ milliseconds.

31.5.1.1 Initializing the MCG

Because the MCG comes out of reset in FEI mode, the only MCG modes that can be directly switched to upon reset are FEE, FBE, and FBI modes (see Figure 31-2). Reaching any of the other modes requires first configuring the MCG for one of these three intermediate modes. Care must be taken to check relevant status bits in the MCG status register reflecting all configuration changes within each mode.

To change from FEI mode to FEE or FBE modes, follow this procedure:

1. Enable the external clock source by setting the appropriate bits in C2 register.
2. Write to C1 register to select the clock mode.

- If entering FEE mode, set C1[FRDIV] appropriately, clear C1[IREFS] bit to switch to the external reference, and leave C1[CLKS] at 2'b00 so that the output of the FLL is selected as the system clock source.
- If entering FBE, clear C1[IREFS] to switch to the external reference and change C1[CLKS] to 2'b10 so that the external reference clock is selected as the system clock source. The C1[FRDIV] bits should also be set appropriately here according to the external reference frequency to keep the FLL reference clock in the range of 31.25 kHz to 39.0625 kHz. Although the FLL is bypassed, it is still on in FBE mode.
- The internal reference can optionally be kept running by setting C1[IRCLKEN]. This is useful if the application will switch back and forth between internal and external modes. For minimum power consumption, leave the internal reference disabled while in an external clock mode.

3. Once the proper configuration bits have been set, wait for the affected bits in the MCG status register to be changed appropriately, reflecting that the MCG has moved into the proper mode.
    - If the MCG is in FEE, FBE, PEE, PBE, or BLPE mode, and C2[EREFS0] was also set in step 1, wait here for S[OSCINIT0] bit to become set indicating that the external clock source has finished its initialization cycles and stabilized.
    - If in FEE mode, check to make sure S[IREFST] is cleared before moving on.
    - If in FBE mode, check to make sure S[IREFST] is cleared and S[CLKST] bits have changed to 2'b10 indicating the external reference clock has been appropriately selected. Although the FLL is bypassed, it is still on in FBE mode.
4. Write to the C4 register to determine the DCO output (MCGFLLCLK) frequency range.
    - By default, with C4[DMX32] cleared to 0, the FLL multiplier for the DCO output is 640. For greater flexibility, if a mid-low-range FLL multiplier of 1280 is desired instead, set C4[DRST_DRS] bits to 2'b01 for a DCO output frequency of 40 MHz. If a mid high-range FLL multiplier of 1920 is desired instead, set the C4[DRST_DRS] bits to 2'b10 for a DCO output frequency of 60 MHz. If a high-range FLL multiplier of 2560 is desired instead, set the C4[DRST_DRS] bits to 2'b11 for a DCO output frequency of 80 MHz.

- When using a 32.768 kHz external reference, if the maximum low-range DCO frequency that can be achieved with a 32.768 kHz reference is desired, set C4[DRST_DRS] bits to 2'b00 and set C4[DMX32] bit to 1. The resulting DCO output (MCGOUTCLK) frequency with the new multiplier of 732 will be 24 MHz.
- When using a 32.768 kHz external reference, if the maximum mid-range DCO frequency that can be achieved with a 32.768 kHz reference is desired, set C4[DRST_DRS] bits to 2'b01 and set C4[DMX32] bit to 1. The resulting DCO output (MCGOUTCLK) frequency with the new multiplier of 1464 will be 48 MHz.
- When using a 32.768 kHz external reference, if the maximum mid high-range DCO frequency that can be achieved with a 32.768 kHz reference is desired, set C4[DRST_DRS] bits to 2'b10 and set C4[DMX32] bit to 1. The resulting DCO output (MCGOUTCLK) frequency with the new multiplier of 2197 will be 72 MHz.
- When using a 32.768 kHz external reference, if the maximum high-range DCO frequency that can be achieved with a 32.768 kHz reference is desired, set C4[DRST_DRS] bits to 2'b11 and set C4[DMX32] bit to 1. The resulting DCO output (MCGOUTCLK) frequency with the new multiplier of 2929 will be 96 MHz.

5. Wait for the FLL lock time to guarantee FLL is running at new C4[DRST_DRS] and C4[DMX32] programmed frequency.

To change from FEI clock mode to FBI clock mode, follow this procedure:

1. Change C1[CLKS] bits in C1 register to 2'b01 so that the internal reference clock is selected as the system clock source.
2. Wait for S[CLKST] bits in the MCG status register to change to 2'b01, indicating that the internal reference clock has been appropriately selected.
3. Write to the C2 register to determine the IRCS output (IRCSCLK) frequency range.
   - By default, with C2[IRCS] cleared to 0, the IRCS selected output clock is the slow internal reference clock (32 kHz IRC). If the faster IRC is desired, set C2[IRCS] to 1 for a IRCS clock derived from the 4 MHz IRC source.

31.5.2 Using a 32.768 kHz reference

In FEE and FBE modes, if using a 32.768 kHz external reference, at the default FLL multiplication factor of 640, the DCO output (MCGFLLCLK) frequency is 20.97 MHz at low-range.

If C4[DRST_DRS] bits are set to 2'b01, the multiplication factor is doubled to 1280, and the resulting DCO output frequency is 41.94 MHz at mid-low-range. If C4[DRST_DRS] bits are set to 2'b10, the multiplication factor is set to 1920, and the resulting DCO output frequency is 62.91 MHz at mid high-range. If C4[DRST_DRS] bits are set to 2'b11, the multiplication factor is set to 2560, and the resulting DCO output frequency is 83.89 MHz at high-range.

In FBI and FEI modes, setting C4[DMX32] bit is not recommended. If the internal reference is trimmed to a frequency above 32.768 kHz, the greater FLL multiplication factor could potentially push the microcontroller system clock out of specification and damage the part.

31.5.3 MCG mode switching

When switching between operational modes of the MCG, certain configuration bits must be changed in order to properly move from one mode to another.

Each time any of these bits are changed (C6[PLLS], C1[IREFS], C1[CLKS], C2[IRCS], or C2[EREFS0], the corresponding bits in the MCG status register (PLLST, IREFST, CLKST, IRCST, or OSCINIT) must be checked before moving on in the application software.

Additionally, care must be taken to ensure that the reference clock divider (C1[FRDIV] and C5[PRDIV0]) is set properly for the mode being switched to. For instance, in PEE mode, if using a 4 MHz crystal, C5[PRDIV0] must be set to 5'b000 (divide-by-1) or 5'b001 (divide-by-2) to divide the external reference down to the required frequency between 2 and 4 MHz

In FBE, FEE, FBI, and FEI modes, at any time, the application can switch the FLL multiplication factor between 640, 1280, 1920, and 2560 with C4[DRST_DRS] bits. Writes to C4[DRST_DRS] bits will be ignored if C2[LP]=1.

The table below shows MCGOUTCLK frequency calculations using C1[FRDIV], C5[PRDIV0], and C6[VDIV0] settings for each clock mode.

Table 31-4. MCGOUTCLK Frequency Calculation Options

Clock Mode	$f_{MCGOUTCLK}$[1]	Note
FEI (FLL engaged internal)	$f_{int} \times F$	Typical $f_{MCGOUTCLK}$ = 21 MHz immediately after reset.
FEE (FLL engaged external)	$(f_{ext} / FLL\_R) \times F$	f_{ext} / FLL_R must be in the range of 31.25 kHz to 39.0625 kHz
FBE (FLL bypassed external)	OSCCLK	OSCCLK / FLL_R must be in the range of 31.25 kHz to 39.0625 kHz
FBI (FLL bypassed internal)	MCGIRCLK	Selectable between slow and fast IRC
PEE (PLL engaged external)	(OSCCLK / PLL_R) × M	OSCCLK / PLL_R must be in the range of 2 – 4 MHz
PBE (PLL bypassed external)	OSCCLK	OSCCLK / PLL_R must be in the range of 2 – 4 MHz
BLPI (Bypassed low power internal)	MCGIRCLK	Selectable between slow and fast IRC
BLPE (Bypassed low power external)	OSCCLK	

[1] FLL_R is the reference divider selected by the C1[FRDIV] bits, F is the FLL factor selected by C4[DRST_DRS] and C4[DMX32] bits , PLL_R is the reference divider selected by C5[PRDIV0] bits, and M is the multiplier selected by C6[VDIV0] bits.

This section will include several mode switching examples, using an 4 MHz external crystal. If using an external clock source less than 2 MHz, the MCG must not be configured for any of the PLL modes (PEE and PBE).

31.5.3.1 Example 1: Moving from FEI to PEE mode: External Crystal = 4 MHz, MCGOUTCLK frequency = 48 MHz

In this example, the MCG will move through the proper operational modes from FEI to PEE to achieve 48 MHz MCGOUTCLK frequency from 4 MHz external crystal reference. First, the code sequence will be described. Then there is a flowchart that illustrates the sequence.

1. First, FEI must transition to FBE mode:
   a. C2 = 0x2C
      - C2[RANGE0] set to 2'b01 because the frequency of 4 MHz is within the high frequency range.
      - C2[HGO0] set to 1 to configure the crystal oscillator for high gain operation.
      - C2[EREFS0] set to 1, because a crystal is being used.

   b. C1 = 0x90

- C1[CLKS] set to 2'b10 to select external reference clock as system clock source
- C1[FRDIV] set to 3'b010, or divide-by-128 because 4 MHz / 128 = 31.25 kHz which is in the 31.25 kHz to 39.0625 kHz range required by the FLL
- C1[IREFS] cleared to 0, selecting the external reference clock and enabling the external oscillator.

c. Loop until S[OSCINIT0] is 1, indicating the crystal selected by C2[EREFS0] has been initialized.

d. Loop until S[IREFST] is 0, indicating the external reference is the current source for the reference clock.

e. Loop until S[CLKST] is 2'b10, indicating that the external reference clock is selected to feed MCGOUTCLK.

2. Then configure C5[PRDIV0] to generate correct PLL reference frequency.

   a. C5 = 0x01

      - C5[PRDIV] set to 5'b00001, or divide-by-2 resulting in a pll reference frequency of 4MHz/2 = 2 MHz.

3. Then, FBE must transition either directly to PBE mode or first through BLPE mode and then to PBE mode:

   a. BLPE: If a transition through BLPE mode is desired, first set C2[LP] to 1.

   b. BLPE/PBE: C6 = 0x40

      - C6[PLLS] set to 1, selects the PLL. At this time, with a C1[PRDIV] value of 2'b001, the PLL reference divider is 2 (see PLL External Reference Divide Factor table), resulting in a reference frequency of 4 MHz/ 2 = 2 MHz. In BLPE mode, changing the C6[PLLS] bit only prepares the MCG for PLL usage in PBE mode.
      - C6[VDIV] set to 5'b00000, or multiply-by-24 because 2 MHz reference * 24 = 48 MHz. In BLPE mode, the configuration of the VDIV bits does not matter because the PLL is disabled. Changing them only sets up the multiply value for PLL usage in PBE mode.

   c. BLPE: If transitioning through BLPE mode, clear C2[LP] to 0 here to switch to PBE mode.

   d. PBE: Loop until S[PLLST] is set, indicating that the current source for the PLLS clock is the PLL.

   e. PBE: Then loop until S[LOCK0] is set, indicating that the PLL has acquired lock.

4. Lastly, PBE mode transitions into PEE mode:

   a. C1 = 0x10

      - C1[CLKS] set to 2'b00 to select the output of the PLL as the system clock source.

   b. Loop until S[CLKST] are 2'b11, indicating that the PLL output is selected to feed MCGOUTCLK in the current clock mode.

      - Now, with PRDIV of divide-by-2, and C6[VDIV] of multiply-by-24, MCGOUTCLK = [(4 MHz / 2) * 24] = 48 MHz.

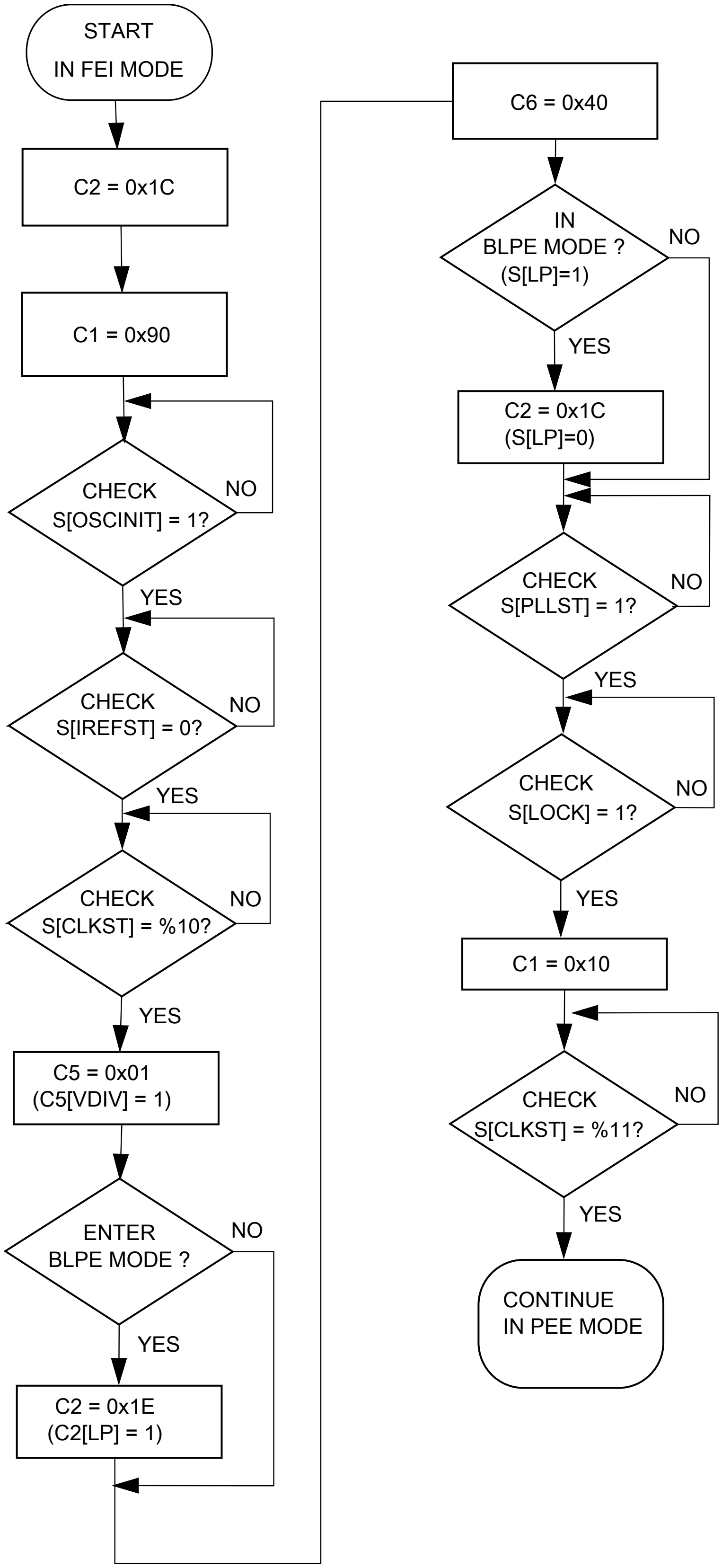


Figure 31-3. Flowchart of FEI to PEE mode transition using an 4 MHz crystal

31.5.3.2 Example 2: Moving from PEE to BLPI mode: MCGOUTCLK frequency =32 kHz

In this example, the MCG will move through the proper operational modes from PEE mode with a 4 MHz crystal configured for a 48 MHz MCGOUTCLK frequency (see previous example) to BLPI mode with a 32 kHz MCGOUTCLK frequency. First, the code sequence will be described. Then there is a flowchart that illustrates the sequence.

1. First, PEE must transition to PBE mode:
    a. C1 = 0x90
        - C1[CLKS] set to 2'b10 to switch the system clock source to the external reference clock.

    b. Loop until S[CLKST] are 2'b10, indicating that the external reference clock is selected to feed MCGOUTCLK.
2. Then, PBE must transition either directly to FBE mode or first through BLPE mode and then to FBE mode:
    a. BLPE: If a transition through BLPE mode is desired, first set C2[LP] to 1.

    b. BLPE/FBE: C6 = 0x00
        - C6[PLLS] clear to 0 to select the FLL. At this time, with C1[FRDIV] value of 3'b010, the FLL divider is set to 128, resulting in a reference frequency of 4 MHz / 128 = 31.25 kHz. If C1[FRDIV] was not previously set to 3'b010 (necessary to achieve required 31.25–39.06 kHz FLL reference frequency with an 4 MHz external source frequency), it must be changed prior to clearing C6[PLLS] bit. In BLPE mode,changing this bit only prepares the MCG for FLL usage in FBE mode. With C6[PLLS] = 0, the C6[VDIV] value does not matter.

    c. BLPE: If transitioning through BLPE mode, clear C2[LP] to 0 here to switch to FBE mode.

    d. FBE: Loop until S[PLLST] is cleared, indicating that the current source for the PLLS clock is the FLL.
3. Next, FBE mode transitions into FBI mode:
    a. C1 = 0x54
        - C1[CLKS] set to 2'b01 to switch the system clock to the internal reference clock.

- C1[IREFS] set to 1 to select the internal reference clock as the reference clock source.
- C1[FRDIV] remain unchanged because the reference divider does not affect the internal reference.

b. Loop until S[IREFST] is 1, indicating the internal reference clock has been selected as the reference clock source.

c. Loop until S[CLKST] are 2'b01, indicating that the internal reference clock is selected to feed MCGOUTCLK.

4. Lastly, FBI transitions into BLPI mode.

a. C2 = 0x02

- C2[LP] is 1
- C2[RANGE0], C2[HGO0], C2[EREFS0], C1[IRCLKEN], and C1[IREFSTEN] bits are ignored when the C1[IREFS] bit is set. They can remain set, or be cleared at this point.

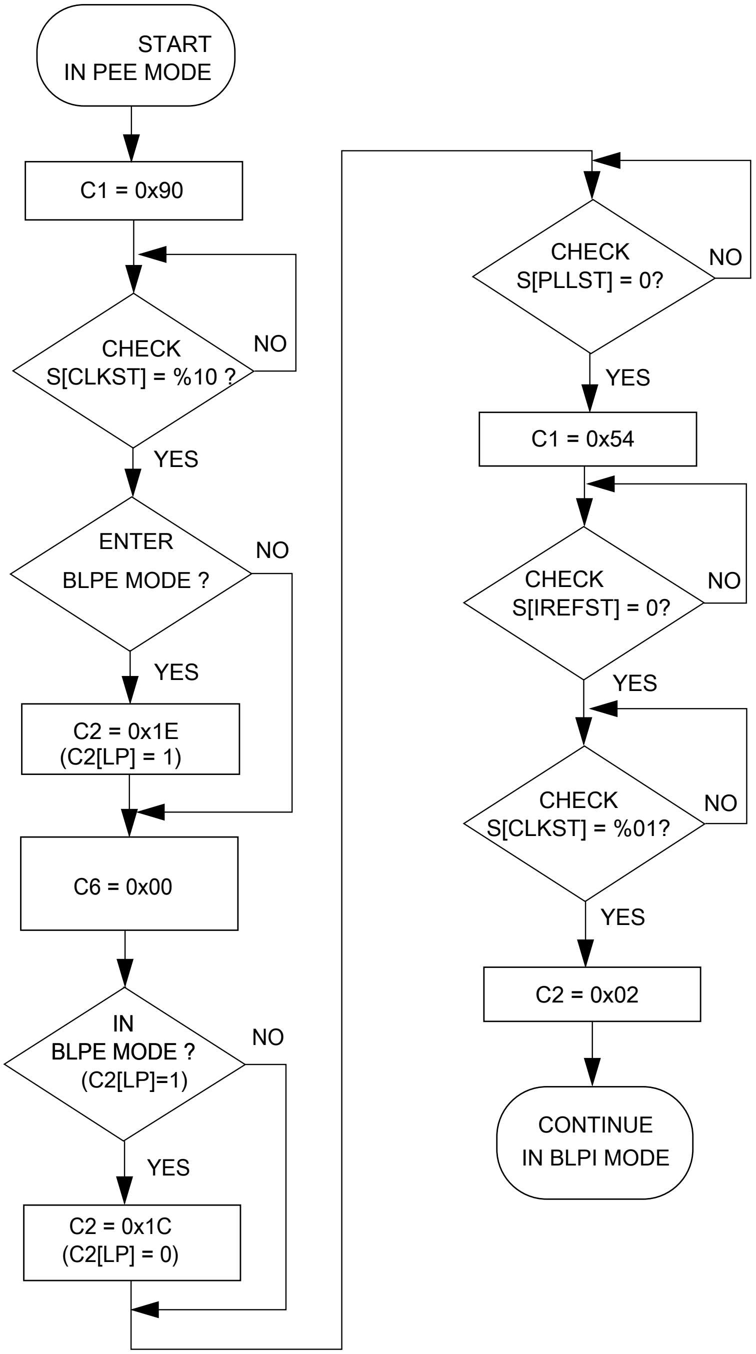


Figure 31-4. Flowchart of PEE to BLPI mode transition using an 4 MHz crystal

Chapter 32
MCU: Oscillator (OSC)

32.1 Introduction

NOTE

> For the chip-specific implementation details of this module's instances, see the chip configuration information.

The OSC module is a crystal oscillator. The module, in conjunction with an external crystal or resonator, generates a reference clock for the MCU.

32.2 Features and Modes

Key features of the module are listed here.

- Supports 32 kHz crystals (Low Range mode)
- Supports 3–8 MHz crystals and resonators (High Range mode)
- Automatic Gain Control (AGC) to optimize power consumption in high frequency ranges 3–8 MHz using low-power mode
- High gain option in frequency ranges: 32 kHz, 3–8 MHz
- Voltage and frequency filtering to guarantee clock frequency and stability
- Optionally external input bypass clock from EXTAL signal directly
- One clock for MCU clock system
- Two clocks for on-chip peripherals that can work in Stop modes

Functional Description describes the module's operation in more detail.

32.3 Block Diagram

The OSC module uses a crystal or resonator to generate three filtered oscillator clock signals.Three clocks are output from OSC module: OSCCLK for MCU system, OSCERCLK for on-chip peripherals, and . The OSCCLK can only work in run mode. OSCERCLK and can work in low power modes. For the clock source assignments, refer to the clock distribution information of this MCU.

Refer to the chip configuration details for the external reference clock source in this MCU.

The figure found here shows the block diagram of the OSC module.

Figure 32-1. OSC Module Block Diagram

32.4 OSC Signal Descriptions

The table found here shows the user-accessible signals available for the OSC module.

Refer to signal multiplexing information for this MCU for more details.

Table 32-1. OSC Signal Descriptions

Signal	Description	I/O
EXTAL	External clock/Oscillator input	I
XTAL	Oscillator output	O

32.5 External Crystal / Resonator Connections

The connections for a crystal/resonator frequency reference are shown in the figures found here.

When using low-frequency, low-power mode, the only external component is the crystal or ceramic resonator itself. In the other oscillator modes, load capacitors (C_x, C_y) and feedback resistor (R_F) are required. The following table shows all possible connections.

Table 32-2. External Caystal/Resonator Connections

Oscillator Mode	Connections
Low-frequency (32 kHz), low-power	Connection 1

Table continues on the next page...

Table 32-2. External Caystal/Resonator Connections (continued)

Oscillator Mode	Connections
Low-frequency (32 kHz), high-gain	Connection 2/Connection 3
High-frequency (3~32 MHz), low-power	Connection 1/Connection 3[2,2]
High-frequency (3~32 MHz), high-gain	Connection 2/Connection 3[2]

1. When the load capacitors (Cx, Cy) are greater than 30 pF, use Connection 3.
2. With the low-power mode, the oscillator has the internal feedback resistor R_F. Therefore, the feedback resistor must not be externally with the Connection 3.

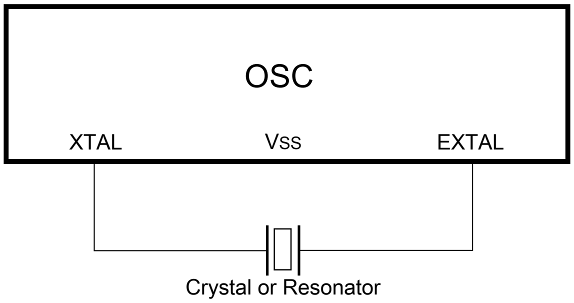


Figure 32-2. Crystal/Ceramic Resonator Connections - Connection 1

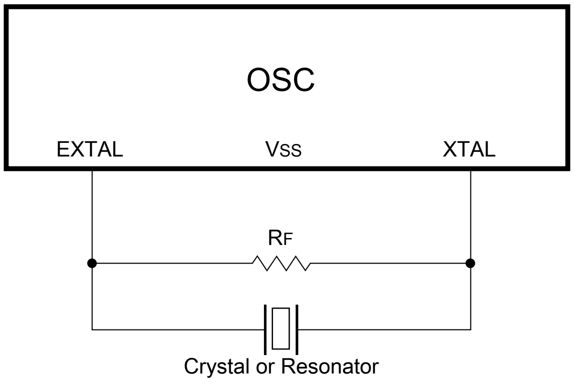


Figure 32-3. Crystal/Ceramic Resonator Connections - Connection 2

NOTE

Connection 1 and Connection 2 should use internal capacitors as the load of the oscillator by configuring the CR[SCxP] bits.

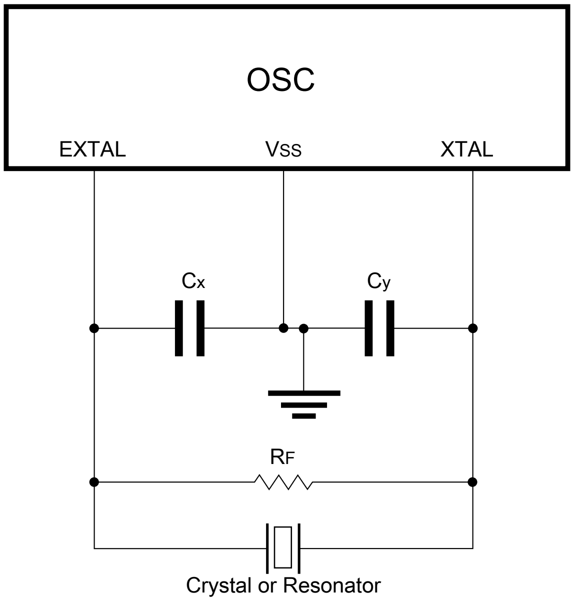


Figure 32-4. Crystal/Ceramic Resonator Connections - Connection 3

32.6 External Clock Connections

In external clock mode, the pins can be connected as shown in the figure found here.

NOTE

XTAL can be used as a GPIO when the GPIO alternate function is configured for it.

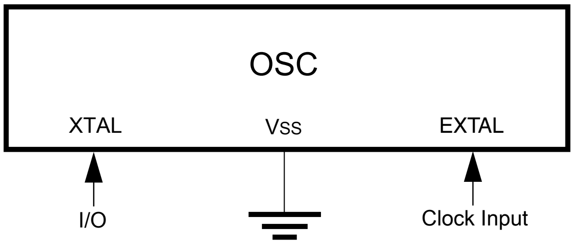


Figure 32-5. External Clock Connections

32.7 Memory Map/Register Definitions

Some oscillator module register bits are typically incorporated into other peripherals such as MCG or SIM.

32.7.1 OSC Memory Map/Register Definition

OSC memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4006_5000	OSC Control Register (OSC_CR)	8	R/W	00h	32.7.1.1/ 695

32.7.1.1 OSC Control Register (OSC_CR)

NOTE

> After OSC is enabled and starts generating the clocks, the configurations such as low power and frequency range, must not be changed.

Address: 4006_5000h base + 0h offset = 4006_5000h

Bit	7	6	5	4	3	2	1	0
Read	ERCLKEN	0	EREFSTEN	0	SC2P	SC4P	SC8P	SC16P
Write								
Reset	0	0	0	0	0	0	0	0

OSC_CR field descriptions

Field	Description
7 ERCLKEN	External Reference Enable Enables external reference clock (OSCERCLK) . 0 External reference clock is inactive. 1 External reference clock is enabled.
6 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
5 EREFSTEN	External Reference Stop Enable Controls whether or not the external reference clock (OSCERCLK) remains enabled when MCU enters Stop mode. 0 External reference clock is disabled in Stop mode. 1 External reference clock stays enabled in Stop mode if ERCLKEN is set before entering Stop mode.
4 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
3 SC2P	Oscillator 2 pF Capacitor Load Configure

Table continues on the next page...

OSC_CR field descriptions (continued)

Field	Description
	Configures the oscillator load. 0 Disable the selection. 1 Add 2 pF capacitor to the oscillator load.
2 SC4P	Oscillator 4 pF Capacitor Load Configure Configures the oscillator load. 0 Disable the selection. 1 Add 4 pF capacitor to the oscillator load.
1 SC8P	Oscillator 8 pF Capacitor Load Configure Configures the oscillator load. 0 Disable the selection. 1 Add 8 pF capacitor to the oscillator load.
0 SC16P	Oscillator 16 pF Capacitor Load Configure Configures the oscillator load. 0 Disable the selection. 1 Add 16 pF capacitor to the oscillator load.

32.8 Functional Description

Functional details of the module can be found here.

32.8.1 OSC module states

The states of the OSC module are shown in the following figure. The states and their transitions between each other are described in this section.

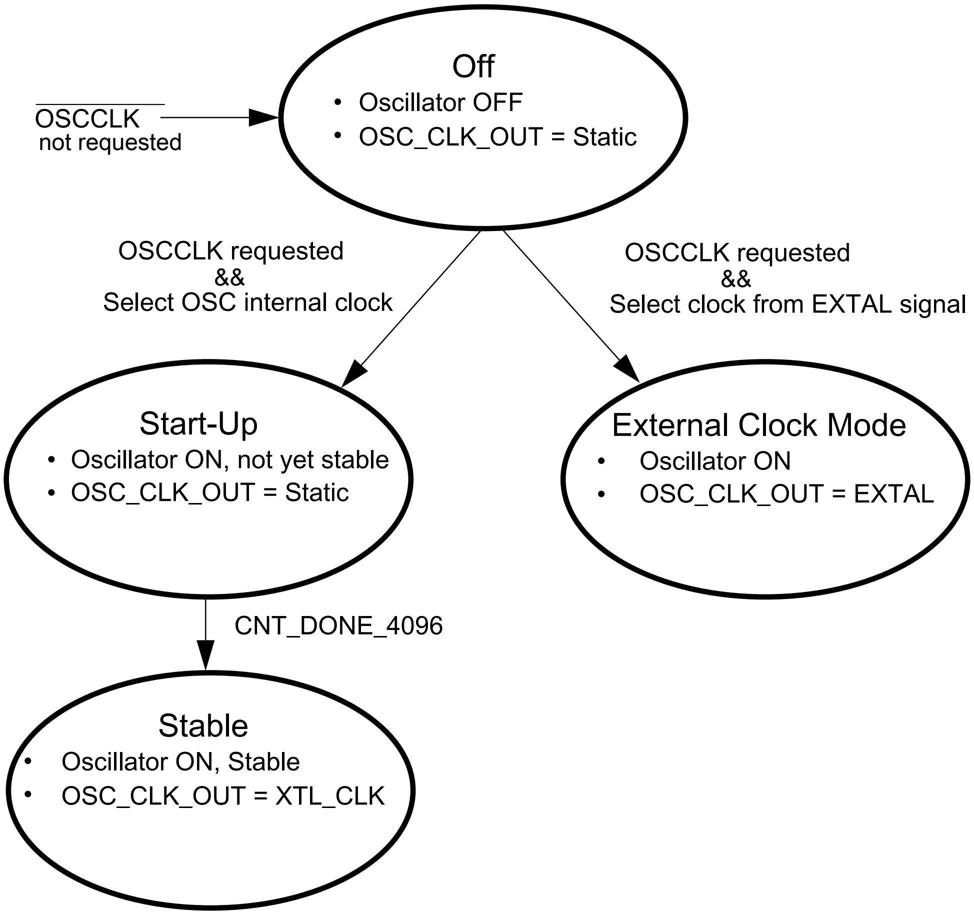


Figure 32-6. OSC Module state diagram

NOTE

XTL_CLK is the clock generated internally from OSC circuits.

32.8.1.1 Off

The OSC enters the Off state when the system does not require OSC clocks. Upon entering this state, XTL_CLK is static unless OSC is configured to select the clock from the EXTAL pad by clearing the external reference clock selection bit. For details regarding the external reference clock source in this MCU, refer to the chip configuration details. The EXTAL and XTAL pins are also decoupled from all other oscillator circuitry in this state. The OSC module circuitry is configured to draw minimal current.

32.8.1.2 Oscillator startup

The OSC enters startup state when it is configured to generate clocks (internally the OSC_EN transitions high) using the internal oscillator circuits by setting the external reference clock selection bit. In this state, the OSC module is enabled and oscillations are starting up, but have not yet stabilized. When the oscillation amplitude becomes large enough to pass through the input buffer, XTL_CLK begins clocking the counter. When the counter reaches 4096 cycles of XTL_CLK, the oscillator is considered stable and XTL_CLK is passed to the output clock OSC_CLK_OUT.

32.8.1.3 Oscillator Stable

The OSC enters stable state when it is configured to generate clocks (internally the OSC_EN transitions high) using the internal oscillator circuits by setting the external reference clock selection bit and the counter reaches 4096 cycles of XTL_CLK (when CNT_DONE_4096 is high). In this state, the OSC module is producing a stable output clock on OSC_CLK_OUT. Its frequency is determined by the external components being used.

32.8.1.4 External Clock mode

The OSC enters external clock state when it is enabled and external reference clock selection bit is cleared. For details regarding external reference clock source in this MCU, see the chip configuration details. In this state, the OSC module is set to buffer (with hysteresis) a clock from EXTAL onto the OSC_CLK_OUT. Its frequency is determined by the external clock being supplied.

32.8.2 OSC module modes

The OSC is a pierce-type oscillator that supports external crystals or resonators operating over the frequency ranges shown in Table 32-3. These modes assume the following conditions: OSC is enabled to generate clocks (OSC_EN=1), configured to generate clocks internally (MCG_C2[EREFS] = 1), and some or one of the other peripherals (MCG, Timer, and so on) is configured to use the oscillator output clock (OSC_CLK_OUT).

Table 32-3. Oscillator modes

Mode	Frequency Range
Low-frequency, high-gain	f_{osc_lo} (32.768 kHz) up to f_{osc_lo} (39.0625 kHz)

Table continues on the next page...

Table 32-3. Oscillator modes (continued)

Mode	Frequency Range
High-frequency mode1, high-gain	$f_{osc\_hi\_1}$ (3 MHz) up to $f_{osc\_hi\_1}$ (8 MHz)
High-frequency mode1, low-power	
High-frequency mode2, high-gain	$f_{osc\_hi\_2}$ (8 MHz) up to $f_{osc\_hi\_2}$ (32 MHz)
High-frequency mode2, low-power	

NOTE

For information about low power modes of operation used in this chip and their alignment with some OSC modes, see the chip's Power Management details.

32.8.2.1 Low-Frequency, High-Gain Mode

In Low-frequency, high-gain mode, the oscillator uses a simple inverter-style amplifier. The gain is set to achieve rail-to-rail oscillation amplitudes.

The oscillator input buffer in this mode is single-ended. It provides low pass frequency filtering as well as hysteresis for voltage filtering and converts the output to logic levels. In this mode, the internal capacitors could be used.

32.8.2.2 Low-Frequency, Low-Power Mode

In low-frequency, low-power mode, the oscillator uses a gain control loop to minimize power consumption. As the oscillation amplitude increases, the amplifier current is reduced. This continues until a desired amplitude is achieved at steady-state. This mode provides low pass frequency filtering as well as hysteresis for voltage filtering and converts the output to logic levels. In this mode, the internal capacitors could be used, the internal feedback resistor is connected, and no external resistor should be used.

In this mode, the amplifier inputs, gain-control input, and input buffer input are all capacitively coupled for leakage tolerance (not sensitive to the DC level of EXTAL).

Also in this mode, all external components except for the resonator itself are integrated, which includes the load capacitors and feeback resistor that biases EXTAL.

32.8.2.3 High-Frequency, High-Gain Mode

In high-frequency, high-gain mode, the oscillator uses a simple inverter-style amplifier. The gain is set to achieve rail-to-rail oscillation amplitudes. This mode provides low pass frequency filtering as well as hysteresis for voltage filtering and converts the output to logic levels. In this mode, the internal capacitors could be used.

32.8.2.4 High-Frequency, Low-Power Mode

In high-frequency, low-power mode, the oscillator uses a gain control loop to minimize power consumption. As the oscillation amplitude increases, the amplifier current is reduced. This continues until a desired amplitude is achieved at steady-state. In this mode, the internal capacitors could be used, the internal feedback resistor is connected, and no external resistor should be used.

The oscillator input buffer in this mode is differential. It provides low pass frequency filtering as well as hysteresis for voltage filtering and converts the output to logic levels.

32.8.3 Counter

The oscillator output clock (OSC_CLK_OUT) is gated off until the counter has detected 4096 cycles of its input clock (XTL_CLK). After 4096 cycles are completed, the counter passes XTL_CLK onto OSC_CLK_OUT. This counting timeout is used to guarantee output clock stability.

32.8.4 Reference clock pin requirements

The OSC module requires use of both the EXTAL and XTAL pins to generate an output clock in Oscillator mode, but requires only the EXTAL pin in External clock mode. The EXTAL and XTAL pins are available for I/O. For the implementation of these pins on this device, refer to the Signal Multiplexing chapter.

32.9 Reset

There is no reset state associated with the OSC module. The counter logic is reset when the OSC is not configured to generate clocks.

There are no sources of reset requests for the OSC module.

32.10 Low power modes operation

When the MCU enters Stop modes, the OSC is functional depending on CR[ERCLKEN] and CR[EREFSETN] bit settings. If both these bits are set, the OSC is in operation.

In Low Leakage Stop (LLS) modes, the OSC holds all register settings. If CR[ERCLKEN] and CR[EREFSTEN] are set before entry to Low Leakage Stop modes, the OSC is still functional in these modes. After waking up from Very Low Leakage Stop (VLLSx) modes, all OSC register bits are reset and initialization is required through software.

32.11 Interrupts

The OSC module does not generate any interrupts.

Chapter 33
MCU: RTC Oscillator

33.1 Introduction

NOTE

> For the chip-specific implementation details of this module's instances, see the chip configuration information.

The RTC oscillator module provides the clock source for the RTC. The RTC oscillator module, in conjunction with an external crystal, generates a reference clock for the RTC.

33.1.1 Features and Modes

The key features of the RTC oscillator are as follows:

- Supports 32 kHz crystals with very low power
- Consists of internal feed back resistor
- Consists of internal programmable capacitors as the C_{load} of the oscillator
- Automatic Gain Control (AGC) to optimize power consumption

The RTC oscillator operations are described in detail in Functional Description .

33.1.2 Block Diagram

The following is the block diagram of the RTC oscillator.

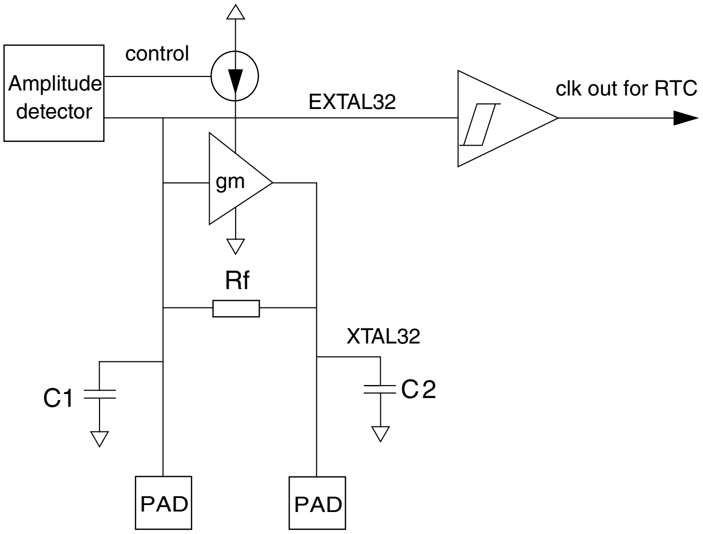


Figure 33-1. RTC Oscillator Block Diagram

33.2 RTC Signal Descriptions

The following table shows the user-accessible signals available for the RTC oscillator. See the chip-level specification to find out which signals are actually connected to the external pins.

Table 33-1. RTC Signal Descriptions

Signal	Description	I/O
EXTAL32	Oscillator Input	I
XTAL32	Oscillator Output	O

33.2.1 EXTAL32 — Oscillator Input

This signal is the analog input of the RTC oscillator.

33.2.2 XTAL32 — Oscillator Output

This signal is the analog output of the RTC oscillator module.

33.3 External Crystal Connections

The connections with a crystal is shown in the following figure. External load capacitors and feedback resistor are not required.

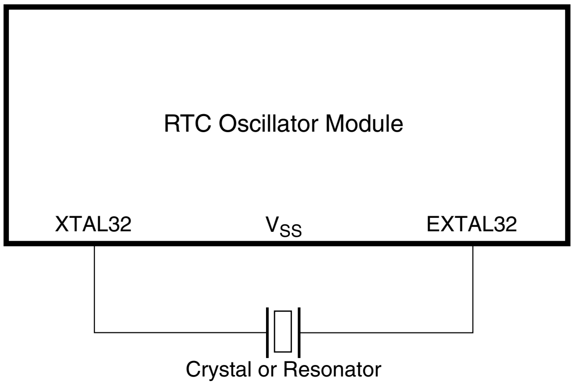


Figure 33-2. Crystal Connections

33.4 Memory Map/Register Descriptions

RTC oscillator control bits are part of the RTC registers. Refer to RTC Control Register (RTC_CR) , or RTC_GP_DATA_REG in the chip-specific information section, for more details.

33.5 Functional Description

As shown in Figure 33-1, the module includes an amplifier which supplies the negative resistor for the RTC oscillator. The gain of the amplifier is controlled by the amplitude detector, which optimizes the power consumption. A schmitt trigger is used to translate the sine-wave generated by this oscillator to a pulse clock out, which is a reference clock for the RTC digital core.

The oscillator includes an internal feedback resistor of approximately 100 MΩ between EXTAL32 and XTAL32.

In addition, there are two programmable capacitors with this oscillator, which can be used as the Cload of the oscillator. The programmable range is from 0pF to 30pF.

33.6 Reset Overview

There is no reset state associated with the RTC oscillator.

33.7 Interrupts

The RTC oscillator does not generate any interrupts.

Chapter 34
MCU: Flash Memory Controller (FMC)

34.1 Introduction

NOTE

> For the chip-specific implementation details of this module's instances, see the chip configuration information.

The Flash Memory Controller (FMC) is a memory acceleration unit that provides:

- an interface between the device and the dual-bank nonvolatile memory. Bank 0 consists of program flash memory, and bank 1 consists of FlexNVM.
- buffers that can accelerate flash memory and FlexNVM data transfers.

34.1.1 Overview

The Flash Memory Controller manages the interface between the device and the dual-bank flash memory. The FMC receives status information detailing the configuration of the memory and uses this information to ensure a proper interface. The following table shows the supported read/write operations.

Flash memory type	Read	Write
Program flash memory	8-bit, 16-bit, and 32-bit reads	—
FlexNVM used as Data flash memory	8-bit, 16-bit, and 32-bit reads	—[1]
FlexNVM and FlexRAM used as EEPROM	8-bit, 16-bit, and 32-bit reads	8-bit, 16-bit, and 32-bit writes

[1] A write operation to program flash memory or to FlexNVM used as data flash memory results in a bus error.

In addition, for bank 0 and bank 1, the FMC provides three separate mechanisms for accelerating the interface between the device and the flash memory. A 64-bit speculation buffer can prefetch the next 64-bit flash memory location, and both a 4-way, 2-set cache and a single-entry 64-bit buffer can store previously accessed flash memory or FlexNVM data for quick access times.

34.1.2 Features

The FMC's features include:

- Interface between the device and the dual-bank flash memory and FlexMemory:
    - 8-bit, 16-bit, and 32-bit read operations to program flash memory and FlexNVM used as data flash memory.
    - 8-bit, 16-bit, and 32-bit read and write operations to FlexNVM and FlexRAM used as EEPROM.
    - For bank 0 and bank 1: Read accesses to consecutive 32-bit spaces in memory return the second read data with no wait states. The memory returns 64 bits via the 32-bit bus access.
    - Crossbar master access protection for setting no access, read-only access, write-only access, or read/write access for each crossbar master.
- For bank 0 and bank 1: Acceleration of data transfer from program flash memory and FlexMemory to the device:
    - 64-bit prefetch speculation buffer with controls for instruction/data access per master and bank
    - 4-way, 2-set, 64-bit line size cache for a total of eight 64-bit entries with controls for replacement algorithm and lock per way for each bank
    - Single-entry buffer per bank
    - Invalidation control for the speculation buffer and the single-entry buffer

34.2 Modes of operation

The FMC only operates when a bus master accesses the flash memory or FlexMemory.

In terms of device power modes, the FMC only operates in run and wait modes, including VLPR and VLPW modes.

For any device power mode where the flash memory or FlexMemory cannot be accessed, the FMC is disabled.

34.3 External signal description

The FMC has no external signals.

34.4 Memory map and register descriptions

The programming model consists of the FMC control registers and the program visible cache (data and tag/valid entries).

NOTE

Program the registers only while the flash controller is idle (for example, execute from RAM). Changing configuration settings while a flash access is in progress can lead to non-deterministic behavior.

Table 34-1. FMC register access

Registers	Read access		Write access	
	Mode	Length	Mode	Length
Control registers: PFAPR, PFB0CR, PFB1CR	Supervisor (privileged) mode or user mode	32 bits	Supervisor (privileged) mode only	32 bits
Cache registers	Supervisor (privileged) mode or user mode	32 bits	Supervisor (privileged) mode only	32 bits

NOTE

Accesses to unimplemented registers within the FMC's 4 KB address space return a bus error.

The cache entries, both data and tag/valid, can be read at any time.

NOTE

System software is required to maintain memory coherence when any segment of the flash cache is programmed. For example, all buffer data associated with the reprogrammed flash should be invalidated. Accordingly, cache program visible writes must occur after a programming or erase event is completed and before the new memory image is accessed.

The cache is a 4-way, set-associative cache with 2 sets. The ways are numbered 0-3 and the sets are numbered 0-1. The following table elaborates on the tag/valid and data entries.

Table 34-2. Program visible cache registers

Cache storage	Based at offset	Contents of 32-bit read	Nomenclature	Nomenclature example
Tag	100h	13'h0, tag[18:4], 3'h0, valid	In TAGVDWxSy, x denotes the way and y denotes the set.	TAGVDW2S0 is the 15-bit tag and 1-bit valid for cache entry way 2, set 0.
Data	200h	Upper or lower longword of data	In DATAWxSyU and DATAWxSyL, x denotes the way, y denotes the set, and U and L represent upper and lower word, respectively.	DATAW1S0U represents bits [63:32] of data entry way 1, set 0, and DATAW1S0L represents bits [31:0] of data entry way 1, set 0.

FMC memory map

Address offset (hex)	Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
0	4001_F000	Flash Access Protection Register (FMC_PFAPR)	32	R/W	00F8_003Fh	34.4.1/711
4	4001_F004	Flash Bank 0 Control Register (FMC_PFB0CR)	32	R/W	3002_001Fh	34.4.2/714
8	4001_F008	Flash Bank 1 Control Register (FMC_PFB1CR)	32	R/W	3002_001Fh	34.4.3/717
100	4001_F100	Cache Tag Storage (FMC_TAGVDW0S0)	32	R/W	0000_0000h	34.4.4/719
104	4001_F104	Cache Tag Storage (FMC_TAGVDW0S1)	32	R/W	0000_0000h	34.4.4/719
108	4001_F108	Cache Tag Storage (FMC_TAGVDW1S0)	32	R/W	0000_0000h	34.4.5/720
10C	4001_F10C	Cache Tag Storage (FMC_TAGVDW1S1)	32	R/W	0000_0000h	34.4.5/720
110	4001_F110	Cache Tag Storage (FMC_TAGVDW2S0)	32	R/W	0000_0000h	34.4.6/721
114	4001_F114	Cache Tag Storage (FMC_TAGVDW2S1)	32	R/W	0000_0000h	34.4.6/721
118	4001_F118	Cache Tag Storage (FMC_TAGVDW3S0)	32	R/W	0000_0000h	34.4.7/722
11C	4001_F11C	Cache Tag Storage (FMC_TAGVDW3S1)	32	R/W	0000_0000h	34.4.7/722
200	4001_F200	Cache Data Storage (upper word) (FMC_DATAW0S0U)	32	R/W	0000_0000h	34.4.8/722
204	4001_F204	Cache Data Storage (lower word) (FMC_DATAW0S0L)	32	R/W	0000_0000h	34.4.9/723
208	4001_F208	Cache Data Storage (upper word) (FMC_DATAW0S1U)	32	R/W	0000_0000h	34.4.8/722
20C	4001_F20C	Cache Data Storage (lower word) (FMC_DATAW0S1L)	32	R/W	0000_0000h	34.4.9/723
210	4001_F210	Cache Data Storage (upper word) (FMC_DATAW1S0U)	32	R/W	0000_0000h	34.4.10/ 723
214	4001_F214	Cache Data Storage (lower word) (FMC_DATAW1S0L)	32	R/W	0000_0000h	34.4.11/ 724
218	4001_F218	Cache Data Storage (upper word) (FMC_DATAW1S1U)	32	R/W	0000_0000h	34.4.10/ 723
21C	4001_F21C	Cache Data Storage (lower word) (FMC_DATAW1S1L)	32	R/W	0000_0000h	34.4.11/ 724
220	4001_F220	Cache Data Storage (upper word) (FMC_DATAW2S0U)	32	R/W	0000_0000h	34.4.12/ 724

Table continues on the next page...

FMC memory map (continued)

Address offset (hex)	Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
224	4001_F224	Cache Data Storage (lower word) (FMC_DATAW2S0L)	32	R/W	0000_0000h	34.4.13/ 725
228	4001_F228	Cache Data Storage (upper word) (FMC_DATAW2S1U)	32	R/W	0000_0000h	34.4.12/ 724
22C	4001_F22C	Cache Data Storage (lower word) (FMC_DATAW2S1L)	32	R/W	0000_0000h	34.4.13/ 725
230	4001_F230	Cache Data Storage (upper word) (FMC_DATAW3S0U)	32	R/W	0000_0000h	34.4.14/ 725
234	4001_F234	Cache Data Storage (lower word) (FMC_DATAW3S0L)	32	R/W	0000_0000h	34.4.15/ 726
238	4001_F238	Cache Data Storage (upper word) (FMC_DATAW3S1U)	32	R/W	0000_0000h	34.4.14/ 725
23C	4001_F23C	Cache Data Storage (lower word) (FMC_DATAW3S1L)	32	R/W	0000_0000h	34.4.15/ 726

34.4.1 Flash Access Protection Register (FMC_PFAPR)

Address: 4001_F000h base + 0h offset = 4001_F000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved								M7PFD	M6PFD	M5PFD	M4PFD	M3PFD	M2PFD	M1PFD	M0PFD
W																
Reset	0	0	0	0	0	0	0	0	1	1	1	1	1	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	M7AP[1:0]		M6AP[1:0]		M5AP[1:0]		M4AP[1:0]		M3AP[1:0]		M2AP[1:0]		M1AP[1:0]		M0AP[1:0]	
W																
Reset	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1

FMC_PFAPR field descriptions

Field	Description
31–24 Reserved	This field is reserved.
23 M7PFD	Master 7 Prefetch Disable These bits control whether prefetching is enabled, based on the logical number of the requesting crossbar switch master. This field is further qualified by the PFBnCR[BxDPE,BxIPE] bits.

Table continues on the next page...

FMC_PFAPR field descriptions (continued)

Field	Description
	0 Prefetching for this master is enabled. 1 Prefetching for this master is disabled.
22 M6PFD	Master 6 Prefetch Disable These bits control whether prefetching is enabled, based on the logical number of the requesting crossbar switch master. This field is further qualified by the PFBnCR[BxDPE,BxIPE] bits. 0 Prefetching for this master is enabled. 1 Prefetching for this master is disabled.
21 M5PFD	Master 5 Prefetch Disable These bits control whether prefetching is enabled, based on the logical number of the requesting crossbar switch master. This field is further qualified by the PFBnCR[BxDPE,BxIPE] bits. 0 Prefetching for this master is enabled. 1 Prefetching for this master is disabled.
20 M4PFD	Master 4 Prefetch Disable These bits control whether prefetching is enabled, based on the logical number of the requesting crossbar switch master. This field is further qualified by the PFBnCR[BxDPE,BxIPE] bits. 0 Prefetching for this master is enabled. 1 Prefetching for this master is disabled.
19 M3PFD	Master 3 Prefetch Disable These bits control whether prefetching is enabled, based on the logical number of the requesting crossbar switch master. This field is further qualified by the PFBnCR[BxDPE,BxIPE] bits. 0 Prefetching for this master is enabled. 1 Prefetching for this master is disabled.
18 M2PFD	Master 2 Prefetch Disable These bits control whether prefetching is enabled, based on the logical number of the requesting crossbar switch master. This field is further qualified by the PFBnCR[BxDPE,BxIPE] bits. 0 Prefetching for this master is enabled. 1 Prefetching for this master is disabled.
17 M1PFD	Master 1 Prefetch Disable These bits control whether prefetching is enabled, based on the logical number of the requesting crossbar switch master. This field is further qualified by the PFBnCR[BxDPE,BxIPE] bits. 0 Prefetching for this master is enabled. 1 Prefetching for this master is disabled.
16 M0PFD	Master 0 Prefetch Disable These bits control whether prefetching is enabled, based on the logical number of the requesting crossbar switch master. This field is further qualified by the PFBnCR[BxDPE,BxIPE] bits. 0 Prefetching for this master is enabled. 1 Prefetching for this master is disabled.

Table continues on the next page...

FMC_PFAPR field descriptions (continued)

Field	Description
15–14 M7AP[1:0]	Master 7 Access Protection This field controls whether read and write access to the flash is allowed, based on the logical master number of the requesting crossbar switch master. 00 No access may be performed by this master. 01 Only read accesses may be performed by this master. 10 Only write accesses may be performed by this master. 11 Both read and write accesses may be performed by this master.
13–12 M6AP[1:0]	Master 6 Access Protection This field controls whether read and write access to the flash is allowed, based on the logical master number of the requesting crossbar switch master. 00 No access may be performed by this master 01 Only read accesses may be performed by this master 10 Only write accesses may be performed by this master 11 Both read and write accesses may be performed by this master
11–10 M5AP[1:0]	Master 5 Access Protection This field controls whether read and write access to the flash is allowed, based on the logical master number of the requesting crossbar switch master. 00 No access may be performed by this master 01 Only read accesses may be performed by this master 10 Only write accesses may be performed by this master 11 Both read and write accesses may be performed by this master
9–8 M4AP[1:0]	Master 4 Access Protection This field controls whether read and write access to the flash is allowed, based on the logical master number of the requesting crossbar switch master. 00 No access may be performed by this master 01 Only read accesses may be performed by this master 10 Only write accesses may be performed by this master 11 Both read and write accesses may be performed by this master
7–6 M3AP[1:0]	Master 3 Access Protection This field controls whether read and write access to the flash is allowed, based on the logical master number of the requesting crossbar switch master. 00 No access may be performed by this master 01 Only read accesses may be performed by this master 10 Only write accesses may be performed by this master 11 Both read and write accesses may be performed by this master
5–4 M2AP[1:0]	Master 2 Access Protection This field controls whether read and write access to the flash is allowed, based on the logical master number of the requesting crossbar switch master. 00 No access may be performed by this master

Table continues on the next page...

FMC_PFAPR field descriptions (continued)

Field	Description
	01 Only read accesses may be performed by this master 10 Only write accesses may be performed by this master 11 Both read and write accesses may be performed by this master
3–2 M1AP[1:0]	Master 1 Access Protection This field controls whether read and write access to the flash is allowed, based on the logical master number of the requesting crossbar switch master. 00 No access may be performed by this master 01 Only read accesses may be performed by this master 10 Only write accesses may be performed by this master 11 Both read and write accesses may be performed by this master
M0AP[1:0]	Master 0 Access Protection This field controls whether read and write access to the flash is allowed, based on the logical master number of the requesting crossbar switch master. 00 No access may be performed by this master 01 Only read accesses may be performed by this master 10 Only write accesses may be performed by this master 11 Both read and write accesses may be performed by this master

34.4.2 Flash Bank 0 Control Register (FMC_PFB0CR)

Address: 4001_F000h base + 4h offset = 4001_F004h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	B0RWSC[3:0]				CLCK_WAY[3:0]				0				0	B0MW[1:0]		0
W									CINV_WAY[3:0]				S_B_INV			
Reset	0	0	1	1	0	0	0	0	0	0	0	0	0	0	1	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								CRC[2:0]			B0DCE	B0ICE	B0DPE	B0IPE	B0SEBE
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1

FMC_PFB0CR field descriptions

Field	Description
31–28 B0RWSC[3:0]	Bank 0 Read Wait State Control This read-only field defines the number of wait states required to access the bank 0 flash memory. The relationship between the read access time of the flash array (expressed in system clock cycles) and RWSC is defined as: Access time of flash array [system clocks] = RWSC + 1 The FMC automatically calculates this value based on the ratio of the system clock speed to the flash clock speed. For example, when this ratio is 4:1, the field's value is 3h.
27–24 CLCK_WAY[3:0]	Cache Lock Way x These bits determine if the given cache way is locked such that its contents will not be displaced by future misses. The bit setting definitions are for each bit in the field. 0 Cache way is unlocked and may be displaced 1 Cache way is locked and its contents are not displaced
23–20 CINV_WAY[3:0]	Cache Invalidate Way x These bits determine if the given cache way is to be invalidated (cleared). When a bit within this field is written, the corresponding cache way is immediately invalidated: the way's tag, data, and valid contents are cleared. This field always reads as zero. Cache invalidation takes precedence over locking. The cache is invalidated by system reset. System software is required to maintain memory coherency when any segment of the flash memory is programmed or erased. Accordingly, cache invalidations must occur after a programming or erase event is completed and before the new memory image is accessed. The bit setting definitions are for each bit in the field. 0 No cache way invalidation for the corresponding cache 1 Invalidate cache way for the corresponding cache: clear the tag, data, and vld bits of ways selected
19 S_B_INV	Invalidate Prefetch Speculation Buffer This bit determines if the FMC's prefetch speculation buffer and the single entry page buffer are to be invalidated (cleared). When this bit is written, the speculation buffer and single entry buffer are immediately cleared. This bit always reads as zero. 0 Speculation buffer and single entry buffer are not affected. 1 Invalidate (clear) speculation buffer and single entry buffer.
18–17 B0MW[1:0]	Bank 0 Memory Width This read-only field defines the width of the bank 0 memory. 00 32 bits 01 64 bits 10 Reserved 11 Reserved
16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
15–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

Table continues on the next page...

FMC_PFB0CR field descriptions (continued)

Field	Description
7–5 CRC[2:0]	Cache Replacement Control This 3-bit field defines the replacement algorithm for accesses that are cached. 000 LRU replacement algorithm per set across all four ways 001 Reserved 010 Independent LRU with ways [0-1] for ifetches, [2-3] for data 011 Independent LRU with ways [0-2] for ifetches, [3] for data 1xx Reserved
4 B0DCE	Bank 0 Data Cache Enable This bit controls whether data references are loaded into the cache. 0 Do not cache data references. 1 Cache data references.
3 B0ICE	Bank 0 Instruction Cache Enable This bit controls whether instruction fetches are loaded into the cache. 0 Do not cache instruction fetches. 1 Cache instruction fetches.
2 B0DPE	Bank 0 Data Prefetch Enable This bit controls whether prefetches (or speculative accesses) are initiated in response to data references. 0 Do not prefetch in response to data references. 1 Enable prefetches in response to data references.
1 B0IPE	Bank 0 Instruction Prefetch Enable This bit controls whether prefetches (or speculative accesses) are initiated in response to instruction fetches. 0 Do not prefetch in response to instruction fetches. 1 Enable prefetches in response to instruction fetches.
0 B0SEBE	Bank 0 Single Entry Buffer Enable This bit controls whether the single entry page buffer is enabled in response to flash read accesses. Its operation is independent from bank 1's cache. A high-to-low transition of this enable forces the page buffer to be invalidated. 0 Single entry buffer is disabled. 1 Single entry buffer is enabled.

34.4.3 Flash Bank 1 Control Register (FMC_PFB1CR)

This register has a format similar to that for PFB0CR, except it controls the operation of flash bank 1, and the "global" cache control fields are empty.

Address: 4001_F000h base + 8h offset = 4001_F008h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	B1RWSC[3:0]				0									B1MW[1:0]		0
W																
Reset	0	0	1	1	0	0	0	0	0	0	0	0	0	0	1	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								0			B1DCE	B1ICE	B1DPE	B1IPE	B1SEBE
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1

FMC_PFB1CR field descriptions

Field	Description
31–28 B1RWSC[3:0]	Bank 1 Read Wait State Control This read-only field defines the number of wait states required to access the bank 1 flash memory. The relationship between the read access time of the flash array (expressed in system clock cycles) and RWSC is defined as: Access time of flash array [system clocks] = RWSC + 1 The FMC automatically calculates this value based on the ratio of the system clock speed to the flash clock speed. For example, when this ratio is 4:1, the field's value is 3h.
27–19 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
18–17 B1MW[1:0]	Bank 1 Memory Width This read-only field defines the width of the bank 1 memory.

Table continues on the next page...

FMC_PFB1CR field descriptions (continued)

Field	Description
	00 32 bits 01 64 bits 10 Reserved 11 Reserved
16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
15–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7–5 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
4 B1DCE	Bank 1 Data Cache Enable This bit controls whether data references are loaded into the cache. 0 Do not cache data references. 1 Cache data references.
3 B1ICE	Bank 1 Instruction Cache Enable This bit controls whether instruction fetches are loaded into the cache. 0 Do not cache instruction fetches. 1 Cache instruction fetches.
2 B1DPE	Bank 1 Data Prefetch Enable This bit controls whether prefetches (or speculative accesses) are initiated in response to data references. 0 Do not prefetch in response to data references. 1 Enable prefetches in response to data references.
1 B1IPE	Bank 1 Instruction Prefetch Enable This bit controls whether prefetches (or speculative accesses) are initiated in response to instruction fetches. 0 Do not prefetch in response to instruction fetches. 1 Enable prefetches in response to instruction fetches.
0 B1SEBE	Bank 1 Single Entry Buffer Enable This bit controls whether the single entry buffer is enabled in response to flash read accesses. Its operation is independent from bank 0's cache. A high-to-low transition of this enable forces the page buffer to be invalidated. 0 Single entry buffer is disabled. 1 Single entry buffer is enabled.

34.4.4 Cache Tag Storage (FMC_TAGVDW0S*n*)

The cache is a 4-way, set-associative cache with 2 sets. The ways are numbered 0-3 and the sets are numbered 0-1. In TAGVDWxSy, x denotes the way, and y denotes the set. This section represents tag/vld information for all sets in the indicated way.

Address: 4001_F000h base + 100h offset + (4d × i), where i=0d to 1d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0													tag[18:4]		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	tag[18:4]												0			valid
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

FMC_TAGVDW0S*n* field descriptions

Field	Description
31–19 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
18–4 tag[18:4]	15-bit tag for cache entry
3–1 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
0 valid	1-bit valid for cache entry

34.4.5 Cache Tag Storage (FMC_TAGVDW1S*n*)

The cache is a 4-way, set-associative cache with 2 sets. The ways are numbered 0-3 and the sets are numbered 0-1. In TAGVDWxSy, x denotes the way, and y denotes the set. This section represents tag/vld information for all sets in the indicated way.

Address: 4001_F000h base + 108h offset + (4d × i), where i=0d to 1d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0													tag[18:4]		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	tag[18:4]												0			valid
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

FMC_TAGVDW1S*n* field descriptions

Field	Description
31–19 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
18–4 tag[18:4]	15-bit tag for cache entry
3–1 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
0 valid	1-bit valid for cache entry

34.4.6 Cache Tag Storage (FMC_TAGVDW2S*n*)

The cache is a 4-way, set-associative cache with 2 sets. The ways are numbered 0-3 and the sets are numbered 0-1. In TAGVDWxSy, x denotes the way, and y denotes the set. This section represents tag/vld information for all sets in the indicated way.

Address: 4001_F000h base + 110h offset + (4d × i), where i=0d to 1d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0													tag[18:4]		
W	▨															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	tag[18:4]												0			valid
W													▨			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

FMC_TAGVDW2S*n* field descriptions

Field	Description
31–19 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
18–4 tag[18:4]	15-bit tag for cache entry
3–1 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
0 valid	1-bit valid for cache entry

34.4.7 Cache Tag Storage (FMC_TAGVDW3S*n*)

The cache is a 4-way, set-associative cache with 2 sets. The ways are numbered 0-3 and the sets are numbered 0-1. In TAGVDWxSy, x denotes the way, and y denotes the set. This section represents tag/vld information for all sets in the indicated way.

Address: 4001_F000h base + 118h offset + (4d × i), where i=0d to 1d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0													tag[18:4]		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	tag[18:4]												0			valid
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

FMC_TAGVDW3S*n* field descriptions

Field	Description
31–19 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
18–4 tag[18:4]	15-bit tag for cache entry
3–1 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
0 valid	1-bit valid for cache entry

34.4.8 Cache Data Storage (upper word) (FMC_DATAW0S*n*U)

The cache of 64-bit entries is a 4-way, set-associative cache with 2 sets. The ways are numbered 0-3 and the sets are numbered 0-1. In DATAWxSyU and DATAWxSyL, x denotes the way, y denotes the set, and U and L represent upper and lower word, respectively. This section represents data for the upper word (bits [63:32]) of all sets in the indicated way.

Address: 4001_F000h base + 200h offset + (8d × i), where i=0d to 1d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	data[63:32]																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

FMC_DATAW0S*n*U field descriptions

Field	Description
data[63:32]	Bits [63:32] of data entry

34.4.9 Cache Data Storage (lower word) (FMC_DATAW0S*n*L)

The cache of 64-bit entries is a 4-way, set-associative cache with 2 sets. The ways are numbered 0-3 and the sets are numbered 0-1. In DATAWxSyU and DATAWxSyL, x denotes the way, y denotes the set, and U and L represent upper and lower word, respectively. This section represents data for the lower word (bits [31:0]) of all sets in the indicated way.

Address: 4001_F000h base + 204h offset + (8d × i), where i=0d to 1d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	data[31:0]																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

FMC_DATAW0S*n*L field descriptions

Field	Description
data[31:0]	Bits [31:0] of data entry

34.4.10 Cache Data Storage (upper word) (FMC_DATAW1S*n*U)

The cache of 64-bit entries is a 4-way, set-associative cache with 2 sets. The ways are numbered 0-3 and the sets are numbered 0-1. In DATAWxSyU and DATAWxSyL, x denotes the way, y denotes the set, and U and L represent upper and lower word, respectively. This section represents data for the upper word (bits [63:32]) of all sets in the indicated way.

Address: 4001_F000h base + 210h offset + (8d × i), where i=0d to 1d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	data[63:32]																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

FMC_DATAW1S*n*U field descriptions

Field	Description
data[63:32]	Bits [63:32] of data entry

34.4.11 Cache Data Storage (lower word) (FMC_DATAW1S*n*L)

The cache of 64-bit entries is a 4-way, set-associative cache with 2 sets. The ways are numbered 0-3 and the sets are numbered 0-1. In DATAWxSyU and DATAWxSyL, x denotes the way, y denotes the set, and U and L represent upper and lower word, respectively. This section represents data for the lower word (bits [31:0]) of all sets in the indicated way.

Address: 4001_F000h base + 214h offset + (8d × i), where i=0d to 1d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	data[31:0]																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

FMC_DATAW1S*n*L field descriptions

Field	Description
data[31:0]	Bits [31:0] of data entry

34.4.12 Cache Data Storage (upper word) (FMC_DATAW2S*n*U)

The cache of 64-bit entries is a 4-way, set-associative cache with 2 sets. The ways are numbered 0-3 and the sets are numbered 0-1. In DATAWxSyU and DATAWxSyL, x denotes the way, y denotes the set, and U and L represent upper and lower word, respectively. This section represents data for the upper word (bits [63:32]) of all sets in the indicated way.

Address: 4001_F000h base + 220h offset + (8d × i), where i=0d to 1d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	data[63:32]																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

FMC_DATAW2S*n*U field descriptions

Field	Description
data[63:32]	Bits [63:32] of data entry

34.4.13 Cache Data Storage (lower word) (FMC_DATAW2S*n*L)

The cache of 64-bit entries is a 4-way, set-associative cache with 2 sets. The ways are numbered 0-3 and the sets are numbered 0-1. In DATAWxSyU and DATAWxSyL, x denotes the way, y denotes the set, and U and L represent upper and lower word, respectively. This section represents data for the lower word (bits [31:0]) of all sets in the indicated way.

Address: 4001_F000h base + 224h offset + (8d × i), where i=0d to 1d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	data[31:0]																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

FMC_DATAW2S*n*L field descriptions

Field	Description
data[31:0]	Bits [31:0] of data entry

34.4.14 Cache Data Storage (upper word) (FMC_DATAW3S*n*U)

The cache of 64-bit entries is a 4-way, set-associative cache with 2 sets. The ways are numbered 0-3 and the sets are numbered 0-1. In DATAWxSyU and DATAWxSyL, x denotes the way, y denotes the set, and U and L represent upper and lower word, respectively. This section represents data for the upper word (bits [63:32]) of all sets in the indicated way.

Address: 4001_F000h base + 230h offset + (8d × i), where i=0d to 1d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	data[63:32]																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

FMC_DATAW3S*n*U field descriptions

Field	Description
data[63:32]	Bits [63:32] of data entry

34.4.15 Cache Data Storage (lower word) (FMC_DATAW3S*n*L)

The cache of 64-bit entries is a 4-way, set-associative cache with 2 sets. The ways are numbered 0-3 and the sets are numbered 0-1. In DATAWxSyU and DATAWxSyL, x denotes the way, y denotes the set, and U and L represent upper and lower word, respectively. This section represents data for the lower word (bits [31:0]) of all sets in the indicated way.

Address: 4001_F000h base + 234h offset + (8d × i), where i=0d to 1d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	data[31:0]																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

FMC_DATAW3S*n*L field descriptions

Field	Description
data[31:0]	Bits [31:0] of data entry

34.5 Functional description

The FMC is a flash acceleration unit with flexible buffers for user configuration. Besides managing the interface between the device and the flash memory and FlexMemory, the FMC can be used to restrict access from crossbar switch masters and —for program flash only—to customize the cache and buffers to provide single-cycle system-clock data-access times. Whenever a hit occurs for the prefetch speculation buffer, the cache, or the single-entry buffer, the requested data is transferred within a single system clock.

34.5.1 Default configuration

Upon system reset, the FMC is configured to provide a significant level of buffering for transfers from the flash memory or FlexMemory:

- Crossbar masters 0, 1, 2 have read access to bank 0 and bank 1.
- These masters have write access to a portion of bank 1 when FlexNVM is used with FlexRAM as EEPROM.
- For bank 0 and bank 1:
  - Prefetch support for data and instructions is enabled for crossbar masters 0, 1, 2.
  - The cache is configured for least recently used (LRU) replacement for all four ways.

- The cache is configured for data or instruction replacement.
- The single-entry buffer is enabled.

34.5.2 Configuration options

Though the default configuration provides a high degree of flash acceleration, advanced users may desire to customize the FMC buffer configurations to maximize throughput for their use cases. When reconfiguring the FMC for custom use cases, do not program the FMC's control registers while the flash memory or FlexMemory is being accessed. Instead, change the control registers with a routine executing from RAM in supervisor mode.

The FMC's cache and buffering controls within PFB0CR and PFB1CR allow the tuning of resources to suit particular applications' needs. The cache and buffer are each controlled individually. The register controls enable buffering and prefetching per memory bank and access type (instruction fetch or data reference). The cache also supports 3 types of LRU replacement algorithms:

- LRU per set across all 4 ways,
- LRU with ways [0-1] for instruction fetches and ways [2-3] for data fetches, and
- LRU with ways [0-2] for instruction fetches and way [3] for data fetches.

As an application example: if both instruction fetches and data references are accessing flash memory, then control is available to send instruction fetches, data references, or both to the cache or the single-entry buffer. Likewise, speculation can be enabled or disabled for either type of access. If both instruction fetches and data references are cached, then the cache's way resources may be divided in several ways between the instruction fetches and data references.

In another application example, the cache can be configured for replacement from bank 0, while the single-entry buffer can be enabled for bank 1 only. This configuration is ideal for applications that use bank 0 for program space and bank 1 for data space.

34.5.3 Wait states

Because the core, crossbar switch, and bus masters can be clocked at a higher frequency than the flash clock, flash memory accesses that do not hit in the speculation buffer or cache usually require wait states. The number of wait states depends on both of the following:

1. the ratio of the core clock to the flash clock, and
2. the phase relationship of the core clock and flash clock at the time the read is requested.

The ratio of the core clock to the flash clock is equal to the value of PFB0CR[B0RWSC] + 1 for bank 0 and to the value of PFB1CR[B1RWSC] + 1 for bank 1.

For example, in a system with a 4:1 core-to-flash clock ratio, a read that does not hit in the speculation buffer or the cache can take between 4 and 7 core clock cycles to complete.

- The best-case scenario is a period of 4 core clock cycles because a read from the flash memory takes 1 flash clock, which translates to 4 core clocks.
- The worst-case scenario is a period of 7 core clock cycles, consisting of 4 cycles for the read operation and 3 cycles of delay to align the core and flash clocks.
    - A delay to align the core and flash clocks might occur because you can request a read cycle on any core clock edge, but that edge does not necessarily align with a flash clock edge where the read can start.
    - In this case, the read operation is delayed by a number of core clocks equal to the core-to-flash clock ratio minus one: 4 - 1 = 3. That is, 3 additional core clock cycles are required to synchronize the clocks before the read operation can start.

All wait states and synchronization delays are handled automatically by the Flash Memory Controller. No direct user configuration is required or even allowed to set up the flash wait states.

34.5.4 Speculative reads

The FMC has a single buffer that reads ahead to the next word in the flash memory if there is an idle cycle. Speculative prefetching is programmable for each bank for instruction and/or data accesses using the B0DPE and B0IPE fields of PFB0CR and the B1DPE and B1IPE fields of PFB1CR. Because many code accesses are sequential, using the speculative prefetch buffer improves performance in most cases.

When speculative reads are enabled, the FMC immediately requests the next sequential address after a read completes. By requesting the next word immediately, speculative reads can help to reduce or even eliminate wait states when accessing sequential code and/or data.

For example, consider the following scenario:

- Assume a system with a 4:1 core-to-flash clock ratio and with speculative reads enabled.
- The core requests four sequential longwords in back-to-back requests, meaning there are no core cycle delays except for stalls waiting for flash memory data to be returned.
- None of the data is already stored in the cache or speculation buffer.

In this scenario, the sequence of events for accessing the four longwords is as follows:

1. The first longword read requires 4 to 7 core clocks. See Wait states for more information.
2. Due to the 64-bit data bus of the flash memory, the second longword read takes only 1 core clock because the data is already available inside the FMC. While the data for the second longword is being returned to the core, the FMC also starts reading the third and fourth longwords from the flash memory.
3. Accessing the third longword requires 3 core clock cycles. The flash memory read itself takes 4 clocks, but the first clock overlaps with the second longword read.
4. Reading the fourth longword, like the second longword, takes only 1 clock due to the 64-bit flash memory data bus.

34.6 Initialization and application information

The FMC does not require user initialization. Flash acceleration features are enabled by default.

The FMC has no visibility into flash memory erase and program cycles because the Flash Memory module manages them directly. As a result, if an application is executing flash memory commands, the FMC's cache might need to be disabled and/or flushed to prevent the possibility of returning stale data. Use the PFB0CR[CINV_WAY] field to invalidate the cache in this manner.

Chapter 35
MCU: Flash Memory Module (FTFL)

35.1 Introduction

NOTE

> For the chip-specific implementation details of this module's instances, see the chip configuration information.

The flash memory module includes the following accessible memory regions:

- Program flash memory for vector space and code store
- For FlexNVM devices: FlexNVM for data store and additional code store
- For FlexNVM devices: FlexRAM for high-endurance data store or traditional RAM
- For program flash only devices: Programming acceleration RAM to speed flash programming

Flash memory is ideal for single-supply applications, permitting in-the-field erase and reprogramming operations without the need for any external high voltage power sources.

The flash memory module includes a memory controller that executes commands to modify flash memory contents. An erased bit reads '1' and a programmed bit reads '0'. The programming operation is unidirectional; it can only move bits from the '1' state (erased) to the '0' state (programmed). Only the erase operation restores bits from '0' to '1'; bits cannot be programmed from a '0' to a '1'.

CAUTION

> A flash memory location must be in the erased state before being programmed. Cumulative programming of bits (back-to-back program operations without an intervening erase) within a flash memory location is not allowed. Re-programming of existing 0s to 0 is not allowed as this overstresses the device.

The standard shipping condition for flash memory is erased with security disabled. Data loss over time may occur due to degradation of the erased ('1') states and/or programmed ('0') states. Therefore, it is recommended that each flash block or sector be re-erased immediately prior to factory programming to ensure that the full data retention capability is achieved.

35.1.1 Features

The flash memory module includes the following features.

NOTE

See the device's Chip Configuration details for the exact amount of flash memory available on your device.

35.1.1.1 Program Flash Memory Features

- Sector size of 2 KB
- Program flash protection scheme prevents accidental program or erase of stored data
- Automated, built-in, program and erase algorithms with verify
- Section programming for faster bulk programming times
- For devices containing only program flash memory: Read access to one logical program flash block is possible while programming or erasing data in the other logical program flash block
- For devices containing FlexNVM memory: Read access to program flash memory possible while programming or erasing data in the data flash memory or FlexRAM

35.1.1.2 FlexNVM Memory Features

When FlexNVM is partitioned for data flash memory (on devices that contain FlexNVM memory):

- Sector size of 2 KB
- Protection scheme prevents accidental program or erase of stored data
- Automated, built-in program and erase algorithms with verify

- Section programming for faster bulk programming times
- Read access to data flash memory possible while programming or erasing data in the program flash memory

35.1.1.3 Programming Acceleration RAM Features

- For devices with only program flash memory: RAM to support section programming

35.1.1.4 FlexRAM Features

For devices with FlexNVM memory:

- Memory that can be used as traditional RAM or as high-endurance EEPROM storage
- Up to 4 KB of FlexRAM configured for EEPROM or traditional RAM operations
- When configured for EEPROM:
    - Protection scheme prevents accidental program or erase of data written for EEPROM
    - Built-in hardware emulation scheme to automate EEPROM record maintenance functions
    - Programmable EEPROM data set size and FlexNVM partition code facilitating EEPROM memory endurance trade-offs
    - Supports FlexRAM aligned writes of 1, 2, or 4 bytes at a time
    - Read access to FlexRAM possible while programming or erasing data in the program or data flash memory
- When configured for traditional RAM:
    - Read and write access possible to the FlexRAM while programming or erasing data in the program or data flash memory

35.1.1.5 Other Flash Memory Module Features

- Internal high-voltage supply generator for flash memory program and erase operations

- Optional interrupt generation upon flash command completion
- Supports MCU security mechanisms which prevent unauthorized access to the flash memory contents

35.1.2 Block Diagram

The block diagram of the flash memory module is shown in the following figure.

For devices with FlexNVM feature:

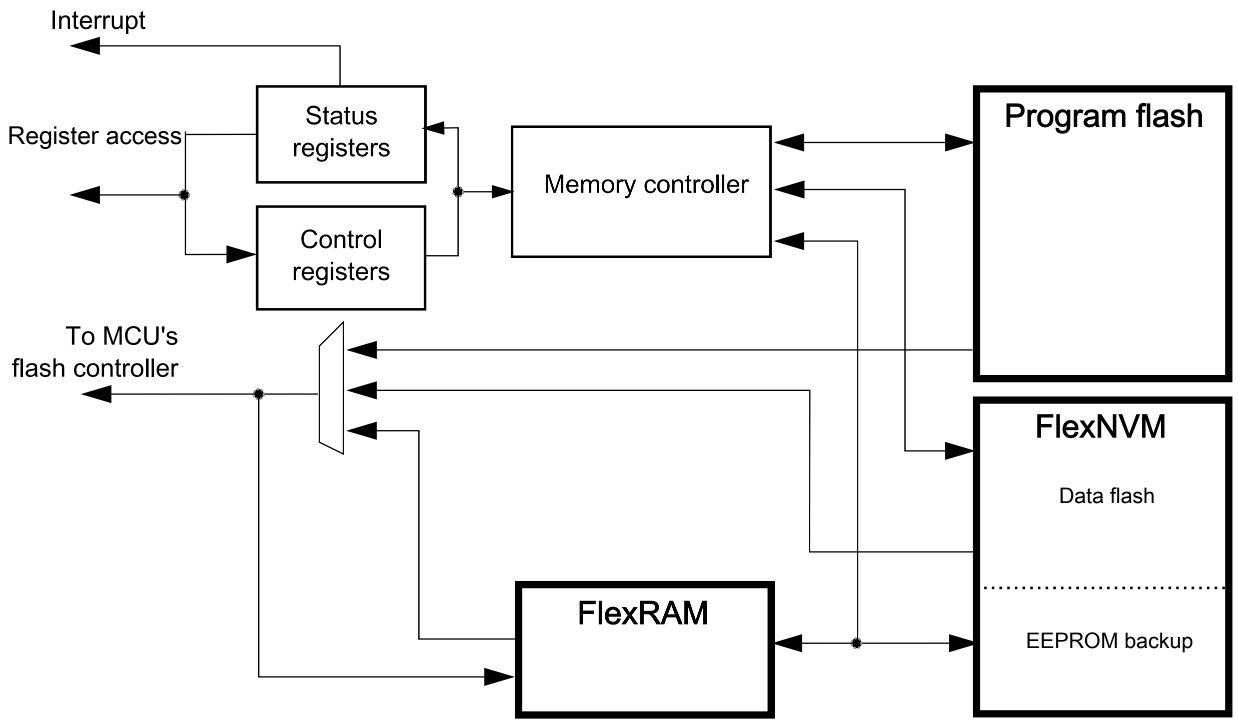


Figure 35-1. Flash Block Diagram

For devices that contain only program flash:

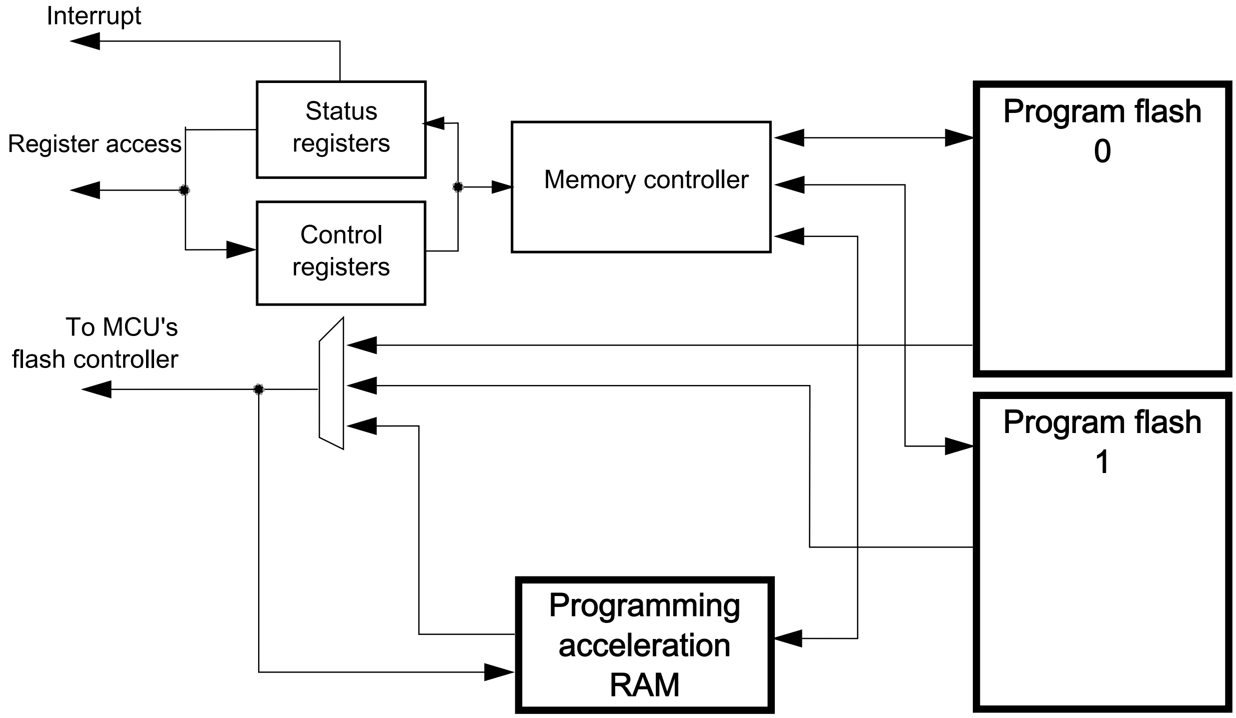


Figure 35-2. Flash Block Diagram

35.1.3 Glossary

Command write sequence — A series of MCU writes to the flash FCCOB register group that initiates and controls the execution of flash algorithms that are built into the flash memory module.

Data flash memory — Partitioned from the FlexNVM block, the data flash memory provides nonvolatile storage for user data, boot code, and additional code store.

Data flash sector — The data flash sector is the smallest portion of the data flash memory that can be erased.

EEPROM — Using a built-in filing system, the flash memory module emulates the characteristics of an EEPROM by effectively providing a high-endurance, byte-writeable (program and erase) NVM.

EEPROM backup data header — The EEPROM backup data header is comprised of a 32-bit field found in EEPROM backup data memory which contains information used by the EEPROM filing system to determine the status of a specific EEPROM backup flash sector.

EEPROM backup data record — The EEPROM backup data record is comprised of a 2-bit status field, a 14-bit address field, and a 16-bit data field found in EEPROM backup data memory which is used by the EEPROM filing system. If the status field indicates a record is valid, the data field is mirrored in the FlexRAM at a location determined by the address field.

EEPROM backup data memory — Partitioned from the FlexNVM block, EEPROM backup data memory provides nonvolatile storage for the EEPROM filing system representing data written to the FlexRAM requiring highest endurance.

EEPROM backup data sector — The EEPROM backup data sector contains one EEPROM backup data header and up to 255 EEPROM backup data records, which are used by the EEPROM filing system.

Endurance — The number of times that a flash memory location can be erased and reprogrammed.

FCCOB (Flash Common Command Object) — A group of flash registers that are used to pass command, address, data, and any associated parameters to the memory controller in the flash memory module.

Flash block — A macro within the flash memory module which provides the nonvolatile memory storage.

FlexMemory — Flash configuration that supports data flash, EEPROM, and FlexRAM.

FlexNVM Block — The FlexNVM block can be configured to be used as data flash memory, EEPROM backup flash memory, or a combination of both.

FlexRAM — The FlexRAM refers to a RAM, dedicated to the flash memory module, that can be configured to store EEPROM data or as traditional RAM. When configured for EEPROM, valid writes to the FlexRAM generate new EEPROM backup data records stored in the EEPROM backup flash memory.

Flash Memory Module — All flash blocks plus a flash management unit providing high-level control and an interface to MCU buses.

IFR — Nonvolatile information register found in each flash block, separate from the main memory array.

Longword — 32 bits of data with an aligned longword having byte-address[1:0] = 00.

NVM — Nonvolatile memory. A memory technology that maintains stored data during power-off. The flash array is an NVM using NOR-type flash memory technology.

NVM Normal Mode — An NVM mode that provides basic user access to flash memory module resources. The CPU or other bus masters initiate flash program and erase operations (or other flash commands) using writes to the FCCOB register group in the flash memory module.

Phrase — 64 bits of data with an aligned phrase having byte-address[2:0] = 000.

Program flash — The program flash memory provides nonvolatile storage for vectors and code store.

Program flash Sector — The smallest portion of the program flash memory (consecutive addresses) that can be erased.

Retention — The length of time that data can be kept in the NVM without experiencing errors upon readout. Since erased (1) states are subject to degradation just like programmed (0) states, the data retention limit may be reached from the last erase operation (not from the programming time).

RWW— Read-While-Write. The ability to simultaneously read from one memory resource while commanded operations are active in another memory resource.

Section Program Buffer — Lower half of the programming acceleration RAM or FlexRAM allocated for storing large amounts of data for programming via the Program Section command.

Secure — An MCU state conveyed to the flash memory module as described in the Chip Configuration details for this device. In the secure state, reading and changing NVM contents is restricted.

Word — 16 bits of data with an aligned word having byte-address[0] = 0.

35.2 External Signal Description

The flash memory module contains no signals that connect off-chip.

35.3 Memory Map and Registers

This section describes the memory map and registers for the flash memory module.

Data read from unimplemented memory space in the flash memory module is undefined. Writes to unimplemented or reserved memory space (registers) in the flash memory module are ignored.

35.3.1 Flash Configuration Field Description

The program flash memory contains a 16-byte flash configuration field that stores default protection settings (loaded on reset) and security information that allows the MCU to restrict access to the flash memory module.

Flash Configuration Field Offset Address	Size (Bytes)	Field Description
0x0_0400–0x0_0407	8	Backdoor Comparison Key. Refer to Verify Backdoor Access Key Command and Unsecuring the Chip Using Backdoor Key Access.
0x0_0408–0x0_040B	4	Program flash protection bytes. Refer to the description of the Program Flash Protection Registers (FPROT0-3).
0x0_040F	1	Program flash only devices: Reserved FlexNVM devices: Data flash protection byte. Refer to the description of the Data Flash Protection Register (FDPROT).
0x0_040E	1	Program flash only devices: Reserved FlexNVM devices: EEPROM protection byte. Refer to the description of the EEPROM Protection Register (FEPROT).
0x0_040D	1	Flash nonvolatile option byte. Refer to the description of the Flash Option Register (FOPT).
0x0_040C	1	Flash security byte. Refer to the description of the Flash Security Register (FSEC).

35.3.2 Program Flash IFR Map

The program flash IFR is nonvolatile information memory that can be read freely, but the user has no erase and limited program capabilities (see the Read Once, Program Once, and Read Resource commands in Read Once Command, Program Once Command and Read Resource Command).

The contents of the program flash IFR are summarized in the table found here and further described in the subsequent paragraphs.

The program flash IFR is located within the program flash 0 memory block for devices that only contain program flash.

Address Range	Size (Bytes)	Field Description
0x00 – 0xBF	192	Reserved
0xC0 – 0xFF	64	Program Once Field

35.3.2.1 Program Once Field

The Program Once Field in the program flash IFR provides 64 bytes of user data storage separate from the program flash main array. The user can program the Program Once Field one time only as there is no program flash IFR erase mechanism available to the user. The Program Once Field can be read any number of times. This section of the program flash IFR is accessed in 4-byte records using the Read Once and Program Once commands (see Read Once Command and Program Once Command).

35.3.3 Data Flash IFR Map

The following only applies to devices with FlexNVM.

The data flash IFR is a 256 byte nonvolatile information memory that can be read and erased, but the user has limited program capabilities in the data flash IFR (see the Program Partition command in Program Partition Command, the Erase All Blocks command in Erase All Blocks Command, and the Read Resource command in Read Resource Command). The contents of the data flash IFR are summarized in the following table and further described in the subsequent paragraphs.

Address Range	Size (Bytes)	Field Description
0x00 – 0xFB, 0xFE – 0xFF	254	Reserved
0xFD	1	EEPROM data set size
0xFC	1	FlexNVM partition code

35.3.3.1 EEPROM Data Set Size

The EEPROM data set size byte in the data flash IFR supplies information which determines the amount of FlexRAM used in each of the available EEPROM subsystems. To program the EEESPLIT and EEESIZE values, see the Program Partition command described in Program Partition Command.

Table 35-1. EEPROM Data Set Size

Data flash IFR: 0x00FD

Table continues on the next page...

Table 35-1. EEPROM Data Set Size (continued)

7	6	5	4	3	2	1	0
1	1	EEESPLIT		EEESIZE			
▨	= Unimplemented or Reserved						

Table 35-2. EEPROM Data Set Size Field Description

Field	Description
7-6 Reserved	This read-only bitfield is reserved and must always be written as one.
5-4 EEESPLIT	**EEPROM Split Factor** — Determines the relative sizes of the two EEPROM subsystems. ‘00’ = Subsystem A: EEESIZE*1/8, subsystem B: EEESIZE*7/8 ‘01’ = Subsystem A: EEESIZE*1/4, subsystem B: EEESIZE*3/4 ‘10’ = Subsystem A: EEESIZE*1/2, subsystem B: EEESIZE*1/2 ‘11’ = Subsystem A: EEESIZE*1/2, subsystem B: EEESIZE*1/2
3-0 EEESIZE	**EEPROM Size** — Encoding of the total available FlexRAM for EEPROM use. **NOTE:** EEESIZE must be 0 bytes (1111b) when the FlexNVM partition code (FlexNVM Partition Code) is set to 'No EEPROM'. '0000' = Reserved '0001' = Reserved '0010' = 4,096 Bytes '0011' = 2,048 Bytes '0100' = 1,024 Bytes '0101' = 512 Bytes '0110' = 256 Bytes '0111' = 128 Bytes '1000' = 64 Bytes '1001' = 32 Bytes '1010' = Reserved '1011' = Reserved '1100' = Reserved '1101' = Reserved '1110' = Reserved '1111' = 0 Bytes

35.3.3.2 FlexNVM Partition Code

The FlexNVM Partition Code byte in the data flash IFR supplies a code which specifies how to split the FlexNVM block between data flash memory and EEPROM backup memory supporting EEPROM functions. To program the DEPART value, see the Program Partition command described in Program Partition Command.

Table 35-3. FlexNVM Partition Code

Data Flash IFR: 0x00FC							
7	6	5	4	3	2	1	0
1	1	1	1	DEPART			
▨	= Unimplemented or Reserved						

Table 35-4. FlexNVM Partition Code Field Description

<table>
<tr><th>Field</th><th>Description</th></tr>
<tr><td>7-4
Reserved</td><td>This read-only bitfield is reserved and must always be written as one.</td></tr>
<tr><td>3-0
DEPART</td><td><b>FlexNVM Partition Code</b> — Encoding of the data flash / EEPROM backup split within the FlexNVM memory block. FlexNVM memory not partitioned for data flash will be used to store EEPROM records.
<table>
<tr><th>DEPART</th><th>Data flash (KB)</th><th>EEPROM backup (KB)</th></tr>
<tr><td>0000</td><td>64</td><td>0</td></tr>
<tr><td>0001</td><td>Reserved</td><td>Reserved</td></tr>
<tr><td>0010</td><td>Reserved</td><td>Reserved</td></tr>
<tr><td>0011</td><td>32</td><td>32</td></tr>
<tr><td>0100</td><td>0</td><td>64</td></tr>
<tr><td>0101</td><td>Reserved</td><td>Reserved</td></tr>
<tr><td>0110</td><td>Reserved</td><td>Reserved</td></tr>
<tr><td>0111</td><td>Reserved</td><td>Reserved</td></tr>
<tr><td>1000</td><td>0</td><td>64</td></tr>
<tr><td>1001</td><td>Reserved</td><td>Reserved</td></tr>
<tr><td>1010</td><td>Reserved</td><td>Reserved</td></tr>
<tr><td>1011</td><td>32</td><td>32</td></tr>
<tr><td>1100</td><td>64</td><td>0</td></tr>
<tr><td>1101</td><td>Reserved</td><td>Reserved</td></tr>
<tr><td>1110</td><td>Reserved</td><td>Reserved</td></tr>
<tr><td>1111</td><td>Reserved (defaults to 64)</td><td>Reserved (defaults to 0)</td></tr>
</table>
</td></tr>
</table>

35.3.4 Register Descriptions

The flash memory module contains a set of memory-mapped control and status registers.

NOTE

While a command is running (FSTAT[CCIF]=0), register writes are not accepted to any register except FCNFG and FSTAT. The no-write rule is relaxed during the start-up reset sequence, prior to the initial rise of CCIF. During this initialization period the user may write any register. All register writes are also disabled (except for registers FCNFG and FSTAT) whenever an erase suspend request is active (FCNFG[ERSSUSP]=1).

FTFL memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4002_0000	Flash Status Register (FTFL_FSTAT)	8	R/W	00h	35.3.4.1/ 743
4002_0001	Flash Configuration Register (FTFL_FCNFG)	8	R/W	See section	35.3.4.2/ 745
4002_0002	Flash Security Register (FTFL_FSEC)	8	R	Undefined	35.3.4.3/ 747
4002_0003	Flash Option Register (FTFL_FOPT)	8	R	Undefined	35.3.4.4/ 748
4002_0004	Flash Common Command Object Registers (FTFL_FCCOB3)	8	R/W	00h	35.3.4.5/ 749
4002_0005	Flash Common Command Object Registers (FTFL_FCCOB2)	8	R/W	00h	35.3.4.5/ 749
4002_0006	Flash Common Command Object Registers (FTFL_FCCOB1)	8	R/W	00h	35.3.4.5/ 749
4002_0007	Flash Common Command Object Registers (FTFL_FCCOB0)	8	R/W	00h	35.3.4.5/ 749
4002_0008	Flash Common Command Object Registers (FTFL_FCCOB7)	8	R/W	00h	35.3.4.5/ 749
4002_0009	Flash Common Command Object Registers (FTFL_FCCOB6)	8	R/W	00h	35.3.4.5/ 749
4002_000A	Flash Common Command Object Registers (FTFL_FCCOB5)	8	R/W	00h	35.3.4.5/ 749
4002_000B	Flash Common Command Object Registers (FTFL_FCCOB4)	8	R/W	00h	35.3.4.5/ 749
4002_000C	Flash Common Command Object Registers (FTFL_FCCOBB)	8	R/W	00h	35.3.4.5/ 749
4002_000D	Flash Common Command Object Registers (FTFL_FCCOBA)	8	R/W	00h	35.3.4.5/ 749

Table continues on the next page...

FTFL memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4002_000E	Flash Common Command Object Registers (FTFL_FCCOB9)	8	R/W	00h	35.3.4.5/ 749
4002_000F	Flash Common Command Object Registers (FTFL_FCCOB8)	8	R/W	00h	35.3.4.5/ 749
4002_0010	Program Flash Protection Registers (FTFL_FPROT3)	8	R/W	Undefined	35.3.4.6/ 750
4002_0011	Program Flash Protection Registers (FTFL_FPROT2)	8	R/W	Undefined	35.3.4.6/ 750
4002_0012	Program Flash Protection Registers (FTFL_FPROT1)	8	R/W	Undefined	35.3.4.6/ 750
4002_0013	Program Flash Protection Registers (FTFL_FPROT0)	8	R/W	Undefined	35.3.4.6/ 750
4002_0016	EEPROM Protection Register (FTFL_FEPROT)	8	R/W	Undefined	35.3.4.7/ 751
4002_0017	Data Flash Protection Register (FTFL_FDPROT)	8	R/W	Undefined	35.3.4.8/ 753

35.3.4.1 Flash Status Register (FTFL_FSTAT)

The FSTAT register reports the operational status of the flash memory module.

The CCIF, RDCOLERR, ACCERR, and FPVIOL bits are readable and writable. The MGSTAT0 bit is read only. The unassigned bits read 0 and are not writable.

NOTE

> When set, the Access Error (ACCERR) and Flash Protection Violation (FPVIOL) bits in this register prevent the launch of any more commands or writes to the FlexRAM (when EEERDY is set) until the flag is cleared (by writing a one to it).

Address: 4002_0000h base + 0h offset = 4002_0000h

Bit	7	6	5	4	3	2	1	0
Read	CCIF	RDCOLERR	ACCERR	FPVIOL	0			MGSTAT0
Write	w1c	w1c	w1c	w1c				
Reset	0	0	0	0	0	0	0	0

FTFL_FSTAT field descriptions

Field	Description
7 CCIF	Command Complete Interrupt Flag Indicates that a flash command or EEPROM file system operation has completed. The CCIF flag is cleared by writing a 1 to CCIF to launch a command, and CCIF stays low until command completion or

Table continues on the next page...

FTFL_FSTAT field descriptions (continued)

Field	Description
	command violation. The CCIF flag is also cleared by a successful write to FlexRAM while enabled for EEE, and CCIF stays low until the EEPROM file system has created the associated EEPROM data record. CCIF is reset to 0 but is set to 1 by the memory controller at the end of the reset initialization sequence. Depending on how quickly the read occurs after reset release, the user may or may not see the 0 hardware reset value. 0 Flash command or EEPROM file system operation in progress 1 Flash command or EEPROM file system operation has completed
6 RDCOLERR	Flash Read Collision Error Flag Indicates that the MCU attempted a read from a flash memory resource that was being manipulated by a flash command (CCIF=0). Any simultaneous access is detected as a collision error by the block arbitration logic. The read data in this case cannot be guaranteed. The RDCOLERR bit is cleared by writing a 1 to it. Writing a 0 to RDCOLERR has no effect. 0 No collision error detected 1 Collision error detected
5 ACCERR	Flash Access Error Flag Indicates an illegal access has occurred to a flash memory resource caused by a violation of the command write sequence or issuing an illegal flash command. While ACCERR is set, the CCIF flag cannot be cleared to launch a command. The ACCERR bit is cleared by writing a 1 to ACCERR while CCIF is set. Writing a 0 to the ACCERR bit has no effect. 0 No access error detected 1 Access error detected
4 FPVIOL	Flash Protection Violation Flag Indicates an attempt was made to program or erase an address in a protected area of program flash or data flash memory during a command write sequence or a write was attempted to a protected area of the FlexRAM while enabled for EEPROM. While FPVIOL is set, the CCIF flag cannot be cleared to launch a command. The FPVIOL bit is cleared by writing a 1 to FPVIOL while CCIF is set. Writing a 0 to the FPVIOL bit has no effect. 0 No protection violation detected 1 Protection violation detected
3–1 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
0 MGSTAT0	Memory Controller Command Completion Status Flag The MGSTAT0 status flag is set if an error is detected during execution of a flash command or during the flash reset sequence. As a status flag, this field cannot (and need not) be cleared by the user like the other error flags in this register. The value of the MGSTAT0 bit for "command-N" is valid only at the end of the "command-N" execution when CCIF=1 and before the next command has been launched. At some point during the execution of "command-N+1," the previous result is discarded and any previous error is cleared.

35.3.4.2 Flash Configuration Register (FTFL_FCNFG)

This register provides information on the current functional state of the flash memory module.

The erase control bits (ERSAREQ and ERSSUSP) have write restrictions. SWAP,PFLSH, RAMRDY, and EEERDY are read-only status bits . The unassigned bits read as noted and are not writable. The reset values for the SWAP, PFLASH, RAMRDY , and EEERDY bits are determined during the reset sequence.

Address: 4002_0000h base + 1h offset = 4002_0001h

Bit	7	6	5	4	3	2	1	0
Read	CCIE	RDCOLLIE	ERSAREQ	ERSSUSP	SWAP	PFLSH	RAMRDY	EEERDY
Write								
Reset	0	0	0	0	*	*	*	*

* Notes:

- SWAP field: Device specific value
- PFLSH field: Device specific value
- RAMRDY field: Device specific value
- EEERDY field: Device specific value

FTFL_FCNFG field descriptions

Field	Description
7 CCIE	Command Complete Interrupt Enable Controls interrupt generation when a flash command completes. 0 Command complete interrupt disabled 1 Command complete interrupt enabled. An interrupt request is generated whenever the FSTAT[CCIF] flag is set.
6 RDCOLLIE	Read Collision Error Interrupt Enable Controls interrupt generation when a flash memory read collision error occurs. 0 Read collision error interrupt disabled 1 Read collision error interrupt enabled. An interrupt request is generated whenever a flash memory read collision error is detected (see the description of FSTAT[RDCOLERR]).
5 ERSAREQ	Erase All Request Issues a request to the memory controller to execute the Erase All Blocks command and release security. ERSAREQ is not directly writable but is under indirect user control. Refer to the device's Chip Configuration details on how to request this command. ERSAREQ sets when an erase all request is triggered external to the flash memory module and CCIF is set (no command is currently being executed). ERSAREQ is cleared by the flash memory module when the operation completes.

Table continues on the next page...

FTFL_FCNFG field descriptions (continued)

Field	Description
	0 No request or request complete 1 Request to: 1. run the Erase All Blocks command, 2. verify the erased state, 3. program the security byte in the Flash Configuration Field to the unsecure state, and 4. release MCU security by setting the FSEC[SEC] field to the unsecure state.
4 ERSSUSP	Erase Suspend Allows the user to suspend (interrupt) the Erase Flash Sector command while it is executing. 0 No suspend requested 1 Suspend the current Erase Flash Sector command execution.
3 SWAP	Swap For program flash only configurations, the SWAP flag indicates which physical program flash block is located at relative address 0x0000. The state of the SWAP flag is set by the flash memory module during the reset sequence. See the Swap Control command section for information on swap management. 0 Physical program flash 0 is located at relative address 0x0000 1 If the PFLSH flag is set, physical program flash 1 is located at relative address 0x0000. If the PFLSH flag is not set, physical program flash 0 is located at relative address 0x0000
2 PFLSH	Flash memory configuration 0 For devices with FlexNVM: Flash memory module configured for FlexMemory that supports data flash and/or EEPROM. For devices with program flash only: Reserved 1 For devices with FlexNVM: Reserved. For devices with program flash only: Flash memory module configured for program flash only, without support for data flash and/or EEPROM
1 RAMRDY	RAM Ready This flag indicates the current status of the FlexRAM/programming acceleration RAM. For devices with FlexNVM: The state of the RAMRDY flag is normally controlled by the Set FlexRAM Function command. During the reset sequence, the RAMRDY flag is cleared if the FlexNVM block is partitioned for EEPROM and is set if the FlexNVM block is not partitioned for EEPROM. The RAMRDY flag is cleared if the Program Partition command is run to partition the FlexNVM block for EEPROM. The RAMRDY flag sets after completion of the Erase All Blocks command or execution of the erase-all operation triggered external to the flash memory module. For devices without FlexNVM: This bit should always be set. 0 For devices with FlexNVM: FlexRAM is not available for traditional RAM access. For devices without FlexNVM: Programming acceleration RAM is not available. 1 For devices with FlexNVM: FlexRAM is available as traditional RAM only; writes to the FlexRAM do not trigger EEPROM operations. For devices without FlexNVM: Programming acceleration RAM is available.
0 EEERDY	For devices with FlexNVM: This flag indicates if the EEPROM backup data has been copied to the FlexRAM and is therefore available for read access. During the reset sequence, the EEERDY flag will remain cleared while CCIF is clear and will only set if the FlexNVM block is partitioned for EEPROM. For devices without FlexNVM: This field is reserved. 0 For devices with FlexNVM: FlexRAM is not available for EEPROM operation. 1 For devices with FlexNVM: FlexRAM is available for EEPROM operations where:

Table continues on the next page...

FTFL_FCNFG field descriptions (continued)

Field	Description
	• reads from the FlexRAM return data previously written to the FlexRAM in EEPROM mode and • writes to the FlexRAM clear EEERDY and launch an EEPROM operation to store the written data in the FlexRAM and EEPROM backup.

35.3.4.3 Flash Security Register (FTFL_FSEC)

This read-only register holds all bits associated with the security of the MCU and flash memory module.

During the reset sequence, the register is loaded with the contents of the flash security byte in the Flash Configuration Field located in program flash memory. The flash basis for the values is signified by X in the reset value.

Address: 4002_0000h base + 2h offset = 4002_0002h

Bit	7	6	5	4	3	2	1	0
Read	KEYEN		MEEN		FSLACC		SEC	
Write								
Reset	x*	x*	x*	x*	x*	x*	x*	x*

* Notes:
- x = Undefined at reset.

FTFL_FSEC field descriptions

Field	Description
7–6 KEYEN	Backdoor Key Security Enable Enables or disables backdoor key access to the flash memory module. 00 Backdoor key access disabled 01 Backdoor key access disabled (preferred KEYEN state to disable backdoor key access) 10 Backdoor key access enabled 11 Backdoor key access disabled
5–4 MEEN	Mass Erase Enable Enables and disables mass erase capability of the flash memory module. When SEC is set to unsecure, the MEEN setting does not matter. 00 Mass erase is enabled 01 Mass erase is enabled 10 Mass erase is disabled 11 Mass erase is enabled
3–2 FSLACC	Factory Security Level Access Code

Table continues on the next page...

FTFL_FSEC field descriptions (continued)

Field	Description
	Enables or disables access to the flash memory contents during returned part failure analysis at NXP. When SEC is secure and FSLACC is denied, access to the program flash contents is denied and any failure analysis performed by NXP factory test must begin with a full erase to unsecure the part. When access is granted (SEC is unsecure, or SEC is secure and FSLACC is granted), NXP factory testing has visibility of the current flash contents. The state of the FSLACC bits is only relevant when SEC is set to secure. When SEC is set to unsecure, the FSLACC setting does not matter. 00 NXP factory access granted 01 NXP factory access denied 10 NXP factory access denied 11 NXP factory access granted
SEC	Flash Security Defines the security state of the MCU. In the secure state, the MCU limits access to flash memory module resources. The limitations are defined per device and are detailed in the Chip Configuration details. If the flash memory module is unsecured using backdoor key access, SEC is forced to 10b. 00 MCU security status is secure. 01 MCU security status is secure. 10 MCU security status is unsecure. (The standard shipping condition of the flash memory module is unsecure.) 11 MCU security status is secure.

35.3.4.4 Flash Option Register (FTFL_FOPT)

The flash option register allows the MCU to customize its operations by examining the state of these read-only bits, which are loaded from NVM at reset. The function of the bits is defined in the device's Chip Configuration details.

All bits in the register are read-only .

During the reset sequence, the register is loaded from the flash nonvolatile option byte in the Flash Configuration Field located in program flash memory. The flash basis for the values is signified by X in the reset value.

Address: 4002_0000h base + 3h offset = 4002_0003h

Bit	7	6	5	4	3	2	1	0
Read	OPT							
Write								
Reset	x*	x*	x*	x*	x*	x*	x*	x*

* Notes:
- x = Undefined at reset.

FTFL_FOPT field descriptions

Field	Description
OPT	Nonvolatile Option These bits are loaded from flash to this register at reset. Refer to the device's Chip Configuration details for the definition and use of these bits.

35.3.4.5 Flash Common Command Object Registers (FTFL_FCCOB*n*)

The FCCOB register group provides 12 bytes for command codes and parameters. The individual bytes within the set append a 0-B hex identifier to the FCCOB register name: FCCOB0, FCCOB1, ..., FCCOBB.

Address: 4002_0000h base + 4h offset + (1d × i), where i=0d to 11d

Bit	7	6	5	4	3	2	1	0
Read Write	CCOBn							
Reset	0	0	0	0	0	0	0	0

FTFL_FCCOB*n* field descriptions

<table>
<tr><th>Field</th><th>Description</th></tr>
<tr><td>CCOBn</td><td>
<p>The FCCOB register provides a command code and relevant parameters to the memory controller. The individual registers that compose the FCCOB data set can be written in any order, but you must provide all needed values, which vary from command to command. First, set up all required FCCOB fields and then initiate the command’s execution by writing a 1 to the FSTAT[CCIF] bit. This clears the CCIF bit, which locks all FCCOB parameter fields and they cannot be changed by the user until the command completes (CCIF returns to 1). No command buffering or queueing is provided; the next command can be loaded only after the current command completes.</p>
<p>Some commands return information to the FCCOB registers. Any values returned to FCCOB are available for reading after the FSTAT[CCIF] flag returns to 1 by the memory controller.</p>
<p>The following table shows a generic flash command format. The first FCCOB register, FCCOB0, always contains the command code. This 8-bit value defines the command to be executed. The command code is followed by the parameters required for this specific flash command, typically an address and/or data values.</p>
<p><b>NOTE:</b> The command parameter table is written in terms of FCCOB Number (which is equivalent to the byte number). This number is a reference to the FCCOB register name and is not the register address.</p>
<table>
<tr><th>FCCOB Number</th><th>Typical Command Parameter Contents [7:0]</th></tr>
<tr><td>0</td><td>FCMD (a code that defines the flash command)</td></tr>
<tr><td>1</td><td>Flash address [23:16]</td></tr>
<tr><td>2</td><td>Flash address [15:8]</td></tr>
<tr><td>3</td><td>Flash address [7:0]</td></tr>
<tr><td>4</td><td>Data Byte 0</td></tr>
</table>
</td></tr>
</table>

FTFL_FCCOB*n* field descriptions (continued)

Field	Description
	<table><tr><th>FCCOB Number</th><th>Typical Command Parameter Contents [7:0]</th></tr><tr><td>5</td><td>Data Byte 1</td></tr><tr><td>6</td><td>Data Byte 2</td></tr><tr><td>7</td><td>Data Byte 3</td></tr><tr><td>8</td><td>Data Byte 4</td></tr><tr><td>9</td><td>Data Byte 5</td></tr><tr><td>A</td><td>Data Byte 6</td></tr><tr><td>B</td><td>Data Byte 7</td></tr></table> **FCCOB Endianness and Multi-Byte Access :** The FCCOB register group uses a big endian addressing convention. For all command parameter fields larger than 1 byte, the most significant data resides in the lowest FCCOB register number. The FCCOB register group may be read and written as individual bytes, aligned words (2 bytes) or aligned longwords (4 bytes).

35.3.4.6 Program Flash Protection Registers (FTFL_FPROT*n*)

The FPROT registers define which program flash regions are protected from program and erase operations. Protected flash regions cannot have their content changed; that is, these regions cannot be programmed and cannot be erased by any flash command. Unprotected regions can be changed by program and erase operations.

The four FPROT registers allow 32 protectable regions. Each bit protects a 1/32 region of the program flash memory . The bitfields are defined in each register as follows:

Program flash protection register	Program flash protection bits
FPROT0	PROT[31:24]
FPROT1	PROT[23:16]
FPROT2	PROT[15:8]
FPROT3	PROT[7:0]

During the reset sequence, the FPROT registers are loaded with the contents of the program flash protection bytes in the Flash Configuration Field as indicated in the following table.

Program flash protection register	Flash Configuration Field offset address
FPROT0	0x000B
FPROT1	0x000A

Table continues on the next page...

Program flash protection register	Flash Configuration Field offset address
FPROT2	0x0009
FPROT3	0x0008

To change the program flash protection that is loaded during the reset sequence, unprotect the sector of program flash memory that contains the Flash Configuration Field. Then, reprogram the program flash protection byte.

Address: 4002_0000h base + 10h offset + (1d × i), where i=0d to 3d

Bit	7	6	5	4	3	2	1	0
Read Write	PROT							
Reset	x*	x*	x*	x*	x*	x*	x*	x*

* Notes:

- x = Undefined at reset.

FTFL_FPROT*n* field descriptions

Field	Description
PROT	Program Flash Region Protect Each program flash region can be protected from program and erase operations by setting the associated PROT bit. The protection can only be increased, meaning that currently unprotected memory can be protected, but currently protected memory cannot be unprotected. Since unprotected regions are marked with a 1 and protected regions use a 0, only writes changing 1s to 0s are accepted. This 1-to-0 transition check is performed on a bit-by-bit basis. Those FPROT bits with 1-to-0 transitions are accepted while all bits with 0-to-1 transitions are ignored. **Restriction:** The user must never write to any FPROT register while a command is running (CCIF=0). Trying to alter data in any protected area in the program flash memory results in a protection violation error and sets the FSTAT[FPVIOL] bit. A full block erase of a program flash block is not possible if it contains any protected region. Each bit in the 32-bit protection register represents 1/32 of the total program flash . 0 Program flash region is protected. 1 Program flash region is not protected

35.3.4.7 EEPROM Protection Register (FTFL_FEPROT)

For devices with FlexNVM: The FEPROT register defines which EEPROM regions of the FlexRAM are protected against program and erase operations. Protected EEPROM regions cannot have their content changed by writing to it. Unprotected regions can be changed by writing to the FlexRAM.

For devices with program flash only: This register is reserved and not used.

Address: 4002_0000h base + 16h offset = 4002_0016h

Bit	7	6	5	4	3	2	1	0
Read Write	EPROT							
Reset	x*	x*	x*	x*	x*	x*	x*	x*

* Notes:

- x = Undefined at reset.

FTFL_FEPROT field descriptions

Field	Description
EPROT	EEPROM Region Protect For devices with program flash only: Reserved For devices with FlexNVM: Individual EEPROM regions can be protected from alteration by setting the associated EPROT bit. The EPROT bits are not used when the FlexNVM Partition Code is set to data flash only. When the FlexNVM Partition Code is set to data flash and EEPROM or EEPROM only, each EPROT bit covers one-eighth of the configured EEPROM data (see the EEPROM Data Set Size parameter description). The protection can only be increased. This means that currently-unprotected memory can be protected, but currently-protected memory cannot be unprotected. Since unprotected regions are marked with a 1 and protected regions use a 0, only writes changing 1s to 0s are accepted. This 1-to-0 transition check is performed on a bit-by-bit basis. Those FEPROT bits with 1-to-0 transitions are accepted while all bits with 0-to-1 transitions are ignored. **In NVM Special mode :** All bits of the FEPROT register are writable without restriction. Unprotected areas can be protected and protected areas can be unprotected. **Restriction:** Never write to the FEPROT register while a command is running (CCIF=0). **Reset:** During the reset sequence, the FEPROT register is loaded with the contents of the FlexRAM protection byte in the Flash Configuration Field located in program flash. The flash basis for the reset values is signified by X in the register diagram. To change the EEPROM protection that will be loaded during the reset sequence, the sector of program flash that contains the Flash Configuration Field must be unprotected; then the EEPROM protection byte must be erased and reprogrammed. Trying to alter data by writing to any protected area in the EEPROM results in a protection violation error and sets the FPVIOL bit in the FSTAT register. 0 For devices with program flash only: Reserved. For devices with FlexNVM: EEPROM region is protected 1 For devices with program flash only: Reserved. For devices with FlexNVM: EEPROM region is not protected

35.3.4.8 Data Flash Protection Register (FTFL_FDPROT)

The FDPROT register defines which data flash regions are protected against program and erase operations. Protected Flash regions cannot have their content changed; that is, these regions cannot be programmed and cannot be erased by any flash command. Unprotected regions can be changed by both program and erase operations.

Address: 4002_0000h base + 17h offset = 4002_0017h

Bit	7	6	5	4	3	2	1	0
Read Write	DPROT							
Reset	x*	x*	x*	x*	x*	x*	x*	x*

* Notes:

- x = Undefined at reset.

FTFL_FDPROT field descriptions

Field	Description
DPROT	Data Flash Region Protect For devices with program flash only: Reserved. For devices with FlexNVM:Individual data flash regions can be protected from program and erase operations by setting the associated DPROT bit. Each DPROT bit protects one-eighth of the partitioned data flash memory space. The granularity of data flash protection cannot be less than the data flash sector size. If an unused DPROT bit is set, the Erase all Blocks command does not execute and the FSTAT[FPVIOL] flag is set. The protection can only be increased, meaning that currently unprotected memory can be protected but currently protected memory cannot be unprotected. Since unprotected regions are marked with a 1 and protected regions use a 0, only writes changing 1s to 0s are accepted. This 1-to-0 transition check is performed on a bit-by-bit basis. Those FDPROT bits with 1-to-0 transitions are accepted while all bits with 0-to-1 transitions are ignored. **Restriction:** The user must never write to the FDPROT register while a command is running (CCIF=0). **Reset:** During the reset sequence, the FDPROT register is loaded with the contents of the data flash protection byte in the Flash Configuration Field located in program flash memory. The flash basis for the reset values is signified by X in the register diagram. To change the data flash protection that will be loaded during the reset sequence, unprotect the sector of program flash that contains the Flash Configuration Field. Then, erase and reprogram the data flash protection byte. Trying to alter data with the program and erase commands in any protected area in the data flash memory results in a protection violation error and sets the FSTAT[FPVIOL] bit. A full block erase of the data flash memory (see the Erase Flash Block command description) is not possible if the data flash memory contains any protected region or if the FlexNVM block has been partitioned for EEPROM. 0 Data Flash region is protected 1 Data Flash region is not protected

35.4 Functional Description

The information found here describes functional details of the flash memory module.

35.4.1 Program Flash Memory Swap

For devices that only contain program flash memory: The user can configure the logical memory map of the program flash space such that either of the two physical program flash blocks can exist at relative address 0x0000. This swap feature enables the lower half of the logical program flash space to be operational while the upper half is being updated for future use.

The Swap Control command handles swapping the two logical P-Flash memory blocks within the memory map. See Swap Control Command for details.

35.4.2 Flash Protection

Individual regions within the flash memory can be protected from program and erase operations.

Protection is controlled by the following registers:

- FPROT*n* —
    - For 2^n program flash sizes, four registers typically protect 32 regions of the program flash memory as shown in the following figure

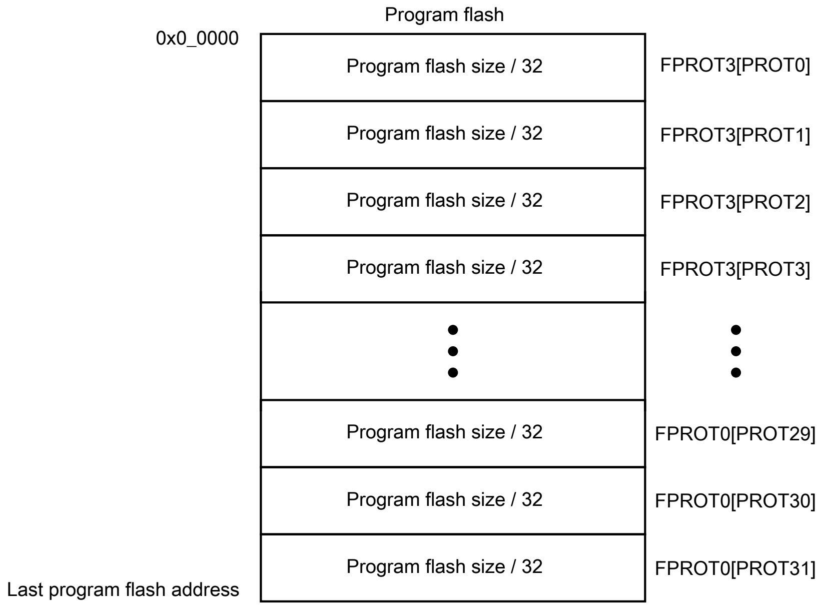


Figure 35-3. Program flash protection

- FDPROT —
  - protects eight regions of the data flash memory as shown in the following figure

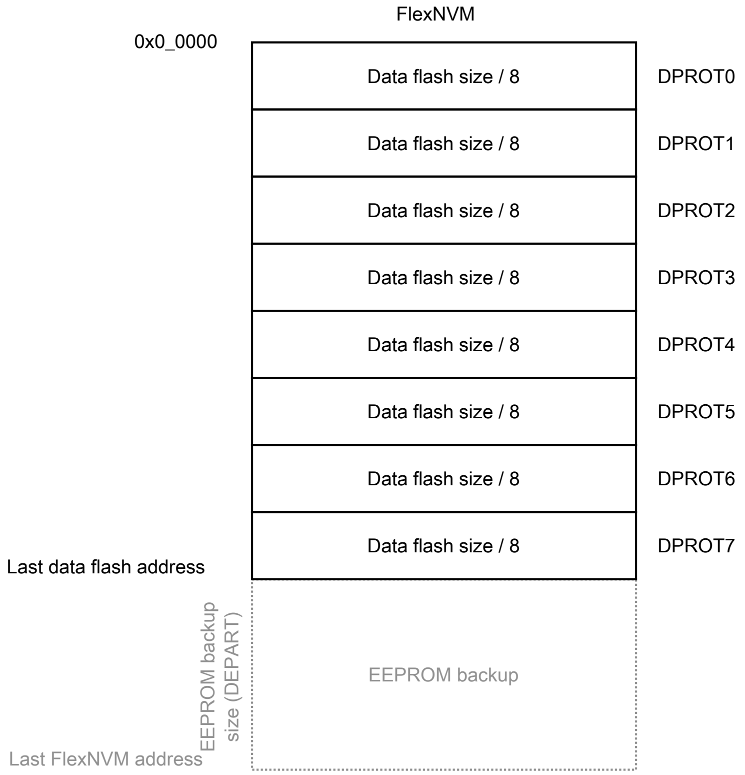


Figure 35-4. Data flash protection

- FEPROT — Protects eight regions of the EEPROM memory as shown in the following figure

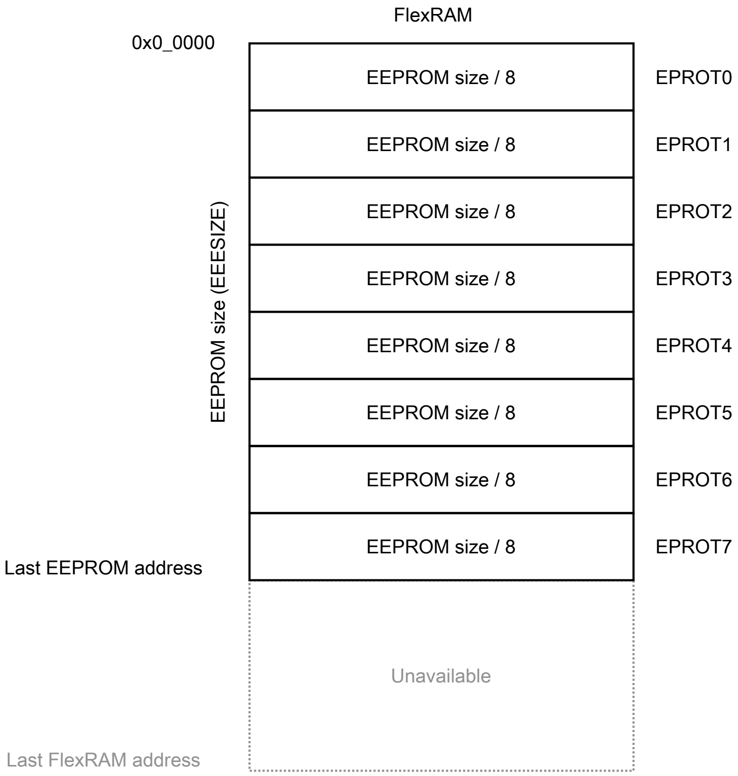


Figure 35-5. EEPROM protection

NOTE

Flash protection features are discussed further in AN4507: Using the Kinetis Security and Flash Protection Features . Not all features described in the application note are available on this device.

35.4.3 FlexNVM Description

This section describes the FlexNVM memory. This section does not apply for devices that contain only program flash memory.

35.4.3.1 FlexNVM Block Partitioning for FlexRAM

The user can configure the FlexNVM block as either:

- Basic data flash,
- EEPROM flash records to support the built-in EEPROM feature, or
- A combination of both.

The user's FlexNVM configuration choice is specified using the Program Partition command described in Program Partition Command.

CAUTION

While different partitions of the FlexNVM block are available, the intention is that a single partition choice is used throughout the entire lifetime of a given application. The FlexNVM partition code choices affect the endurance and data retention characteristics of the device.

35.4.3.2 EEPROM User Perspective

The EEPROM system is shown in the following figure.

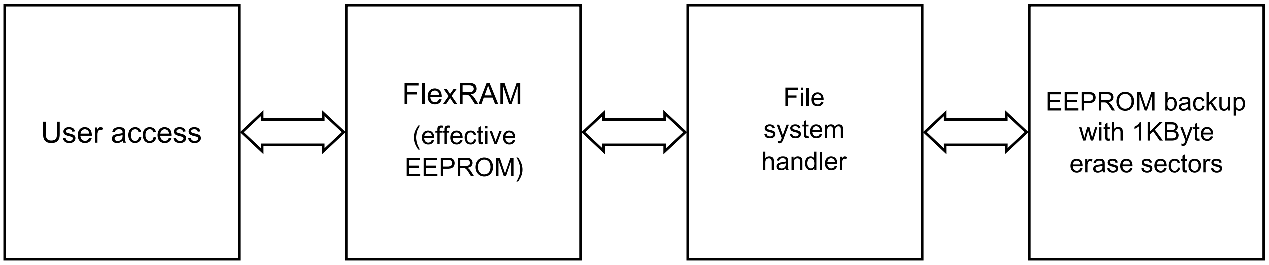


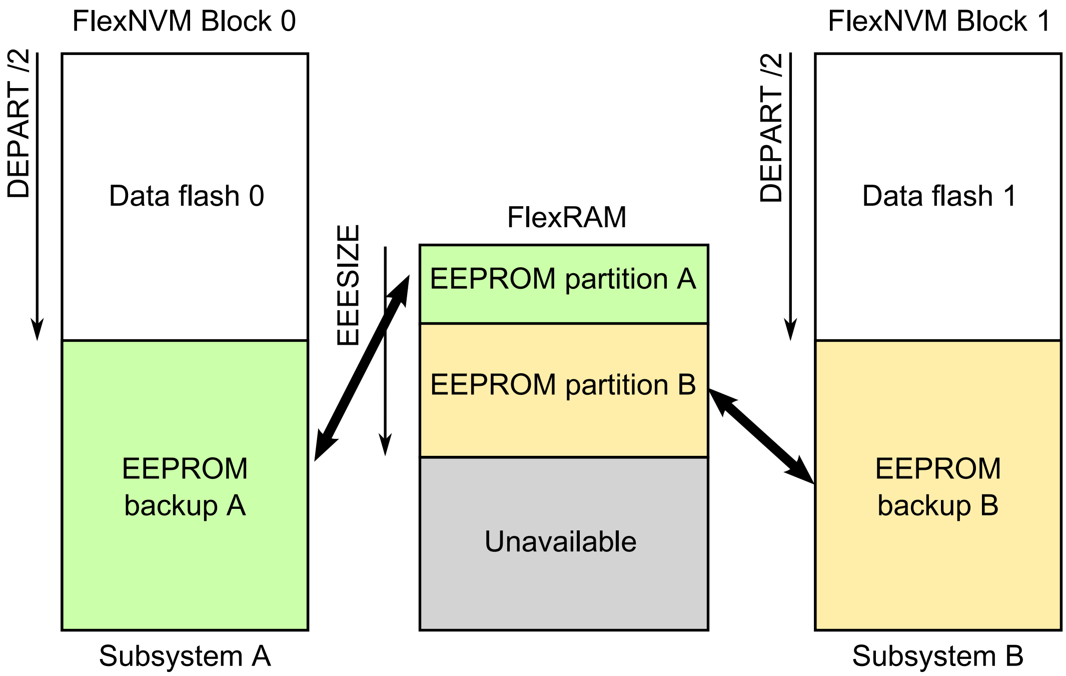
Figure 35-6. Top Level EEPROM Architecture

To handle varying customer requirements, the FlexRAM and FlexNVM blocks can be split into partitions as shown in the figure below.

1. **EEPROM partition** (EEESIZE) — The amount of FlexRAM used for EEPROM can be set from 0 Bytes (no EEPROM) to the maximum FlexRAM size (see Table 35-2). The remainder of the FlexRAM is not accessible while the FlexRAM is configured for EEPROM (see Set FlexRAM Function Command). The EEPROM partition grows upward from the bottom of the FlexRAM address space.
2. **Data flash partition** (DEPART) — The amount of FlexNVM memory used for data flash can be programmed from 0 bytes (all of the FlexNVM block is available for EEPROM backup) to the maximum size of the FlexNVM block (see Table 35-4).
3. **FlexNVM EEPROM partition** — The amount of FlexNVM memory used for EEPROM backup, which is equal to the FlexNVM block size minus the data flash memory partition size. The EEPROM backup size must be at least 16 times the EEPROM partition size in FlexRAM.
4. **EEPROM split factor** (EEESPLIT) — The FlexRAM partitioned for EEPROM can be divided into two subsystems, each backed by half of the partitioned EEPROM backup. One subsystem (A) is 1/8, 1/4, or 1/2 of the partitioned FlexRAM with the remainder belonging to the other subsystem (B).

The partition information (EEESIZE, DEPART, EEESPLIT) is stored in the data flash IFR and is programmed using the Program Partition command (see Program Partition Command). Typically, the Program Partition command is executed only once in the lifetime of the device.

Data flash memory is useful for applications that need to quickly store large amounts of data or store data that is static. The EEPROM partition in FlexRAM is useful for storing smaller amounts of data that will be changed often. The EEPROM partition in FlexRAM can be further sub-divided to provide subsystems, each backed by the same amount of EEPROM backup with subsystem A having higher endurance if the split factor is 1/8 or 1/4.



EEESPLIT = 1/8, 1/4, or 1/2
Size of EEPROM partition A = EEESIZE x EEESPLIT
Size of EEPROM partition B = EEESIZE x (1 - EEESPLIT)
Data flash 0 and 1 interleaved

Figure 35-7. FlexRAM to FlexNVM Memory Mapping with 2 Sub-systems

35.4.3.3 EEPROM Implementation Overview

Out of reset with the FSTAT[CCIF] bit clear, the partition settings (EEESIZE, DEPART, EEESPLIT) are read from the data flash IFR and the EEPROM file system is initialized accordingly. The EEPROM file system locates all valid EEPROM data records in EEPROM backup and copies the newest data to FlexRAM. The FSTAT[CCIF] and FCNFG[EEERDY] bits are set after data from all valid EEPROM data records is copied to the FlexRAM. After the CCIF bit is set, the FlexRAM is available for read or write access.

When configured for EEPROM use, writes to an unprotected location in FlexRAM invokes the EEPROM file system to program a new EEPROM data record in the EEPROM backup memory in a round-robin fashion. As needed, the EEPROM file system identifies the EEPROM backup sector that is being erased for future use and partially erases that EEPROM backup sector. After a write to the FlexRAM, the FlexRAM is not accessible until the FSTAT[CCIF] bit is set. The FCNFG[EEERDY] bit will also be set. If enabled, the interrupt associated with the FSTAT[CCIF] bit can be used to determine when the FlexRAM is available for read or write access.

After a sector in EEPROM backup is full of EEPROM data records, EEPROM data records from the sector holding the oldest data are gradually copied over to a previously-erased EEPROM backup sector. When the sector copy completes, the EEPROM backup sector holding the oldest data is tagged for erase.

35.4.3.4 Write endurance to FlexRAM for EEPROM

When the FlexNVM partition code is not set to full data flash, the EEPROM data set size can be set to any of several non-zero values.

The bytes not assigned to data flash via the FlexNVM partition code are used by the flash memory module to obtain an effective endurance increase for the EEPROM data. The built-in EEPROM record management system raises the number of program/erase cycles that can be attained prior to device wear-out by cycling the EEPROM data through a larger EEPROM NVM storage space.

While different partitions of the FlexNVM are available, the intention is that a single choice for the FlexNVM partition code and EEPROM data set size is used throughout the entire lifetime of a given application. The EEPROM endurance equation and graph shown below assume that only one configuration is ever used.

$$\text{Writes\_subsystem} = \frac{\text{EEPROM} - 2 \times \text{EEESPLIT} \times \text{EEESIZE}}{\text{EEESPLIT} \times \text{EEESIZE}} \times \text{Write\_efficiency} \times n_{\text{nvmcycd}}$$

where

- Writes_subsystem — minimum number of writes to each FlexRAM location for subsystem (each subsystem can have different endurance)
- EEPROM — allocated FlexNVM for each EEPROM subsystem based on DEPART; entered with the Program Partition command
- EEESPLIT — FlexRAM split factor for subsystem; entered with the Program Partition command
- EEESIZE — allocated FlexRAM based on DEPART; entered with the Program Partition command

- Write_efficiency —
  - 0.25 for 8-bit writes to FlexRAM
  - 0.50 for 16-bit or 32-bit writes to FlexRAM
- $n_{nvmcycd}$ — data flash cycling endurance (the following graph assumes 10,000 cycles)

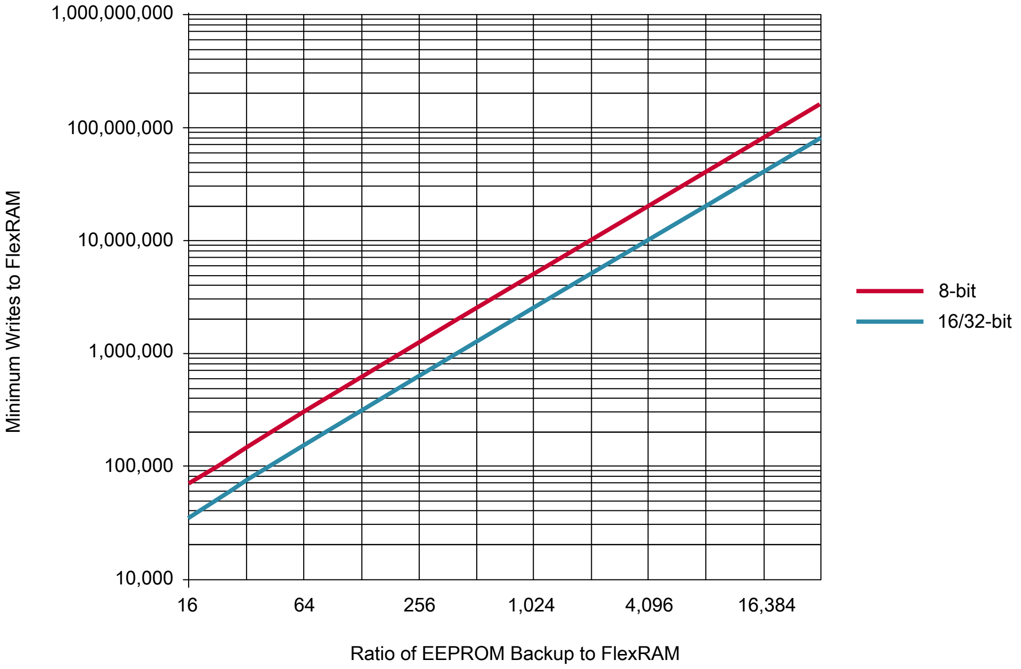


Figure 35-8. EEPROM backup writes to FlexRAM

35.4.4 Interrupts

The flash memory module can generate interrupt requests to the MCU upon the occurrence of various flash events.

These interrupt events and their associated status and control bits are shown in the following table.

Table 35-5. Flash Interrupt Sources

Flash Event	Readable Status Bit	Interrupt Enable Bit
Flash Command Complete	FSTAT[CCIF]	FCNFG[CCIE]
Flash Read Collision Error	FSTAT[RDCOLERR]	FCNFG[RDCOLLIE]

Note

Vector addresses and their relative interrupt priority are determined at the MCU level.

Some devices also generate a bus error response as a result of a Read Collision Error event. See the chip configuration information to determine if a bus error response is also supported.

35.4.5 Flash Operation in Low-Power Modes

35.4.5.1 Wait Mode

When the MCU enters wait mode, the flash memory module is not affected. The flash memory module can recover the MCU from wait via the command complete interrupt (see Interrupts).

35.4.5.2 Stop Mode

When the MCU requests stop mode, if a flash command is active (CCIF = 0) the command execution completes before the MCU is allowed to enter stop mode.

CAUTION

The MCU should never enter stop mode while any flash command is running (CCIF = 0).

NOTE

While the MCU is in very-low-power modes (VLPR, VLPW, VLPS), the flash memory module does not accept flash commands.

35.4.6 Flash Reads and Ignored Writes

The flash memory module requires only the flash address to execute a flash memory read.

The MCU must not read from the flash memory while commands are running (as evidenced by CCIF=0) on that block. Read data cannot be guaranteed from a flash block while any command is processing within that block. The block arbitration logic detects any simultaneous access and reports this as a read collision error (see the FSTAT[RDCOLERR] bit).

35.4.7 Read While Write (RWW)

The following simultaneous accesses are allowed for devices with FlexNVM:

- The user may read from the program flash memory while commands (typically program and erase operations) are active in the data flash and FlexRAM memory space.
- The MCU can fetch instructions from program flash during both data flash program and erase operations and while EEPROM backup data is maintained by the EEPROM commands.
- Conversely, the user may read from data flash and FlexRAM while program and erase commands are executing on the program flash.
- When configured as traditional RAM, writes to the FlexRAM are allowed during program and data flash operations.

Simultaneous data flash operations and FlexRAM writes, when FlexRAM is used for EEPROM, are not possible.

The following simultaneous accesses are allowed for devices with program flash only:

- The user may read from one logical program flash memory space while flash commands are active in the other logical program flash memory space.

Simultaneous operations are further discussed in Allowed Simultaneous Flash Operations.

35.4.8 Flash Program and Erase

All flash functions except read require the user to setup and launch a flash command through a series of peripheral bus writes.

The user cannot initiate any further flash commands until notified that the current command has completed. The flash command structure and operation are detailed in Flash Command Operations.

35.4.9 Flash Command Operations

Flash command operations are typically used to modify flash memory contents.

The next sections describe:

- The command write sequence used to set flash command parameters and launch execution
- A description of all flash commands available

35.4.9.1 Command Write Sequence

Flash commands are specified using a command write sequence illustrated in Figure 35-9. The flash memory module performs various checks on the command (FCCOB) content and continues with command execution if all requirements are fulfilled.

Before launching a command, the ACCERR and FPVIOL bits in the FSTAT register must be zero and the CCIF flag must read 1 to verify that any previous command has completed. If CCIF is zero, the previous command execution is still active, a new command write sequence cannot be started, and all writes to the FCCOB registers are ignored.

Attempts to launch a flash command in VLP mode will be ignored.

35.4.9.1.1 Load the FCCOB Registers

The user must load the FCCOB registers with all parameters required by the desired flash command. The individual registers that make up the FCCOB data set can be written in any order.

35.4.9.1.2 Launch the Command by Clearing CCIF

Once all relevant command parameters have been loaded, the user launches the command by clearing FSTAT[CCIF] by writing a '1' to it. FSTAT[CCIF] remains 0 until the flash command completes.

The FSTAT register contains a blocking mechanism that prevents a new command from launching (can't clear FSTAT[CCIF]) if the previous command resulted in an access error (FSTAT[ACCERR]=1) or a protection violation (FSTAT[FPVIOL]=1). In error scenarios, two writes to FSTAT are required to initiate the next command: the first write clears the error flags, the second write clears CCIF.

35.4.9.1.3 Command Execution and Error Reporting

The command processing has several steps:

1. The flash memory module reads the command code and performs a series of parameter checks and protection checks, if applicable, which are unique to each command.

   If the parameter check fails, the FSTAT[ACCERR] (access error) flag is set. FSTAT[ACCERR] reports invalid instruction codes and out-of bounds addresses. Usually, access errors suggest that the command was not set-up with valid parameters in the FCCOB register group.

   Program and erase commands also check the address to determine if the operation is requested to execute on protected areas. If the protection check fails, FSTAT[FPVIOL] (protection error) flag is set.

   Command processing never proceeds to execution when the parameter or protection step fails. Instead, command processing is terminated after setting FSTAT[CCIF].

2. If the parameter and protection checks pass, the command proceeds to execution. Run-time errors, such as failure to erase verify, may occur during the execution phase. Run-time errors are reported in FSTAT[MGSTAT0]. A command may have access errors, protection errors, and run-time errors, but the run-time errors are not seen until all access and protection errors have been corrected.
3. Command execution results, if applicable, are reported back to the user via the FCCOB and FSTAT registers.
4. The flash memory module sets FSTAT[CCIF] signifying that the command has completed.

The flow for a generic command write sequence is illustrated in the following figure.

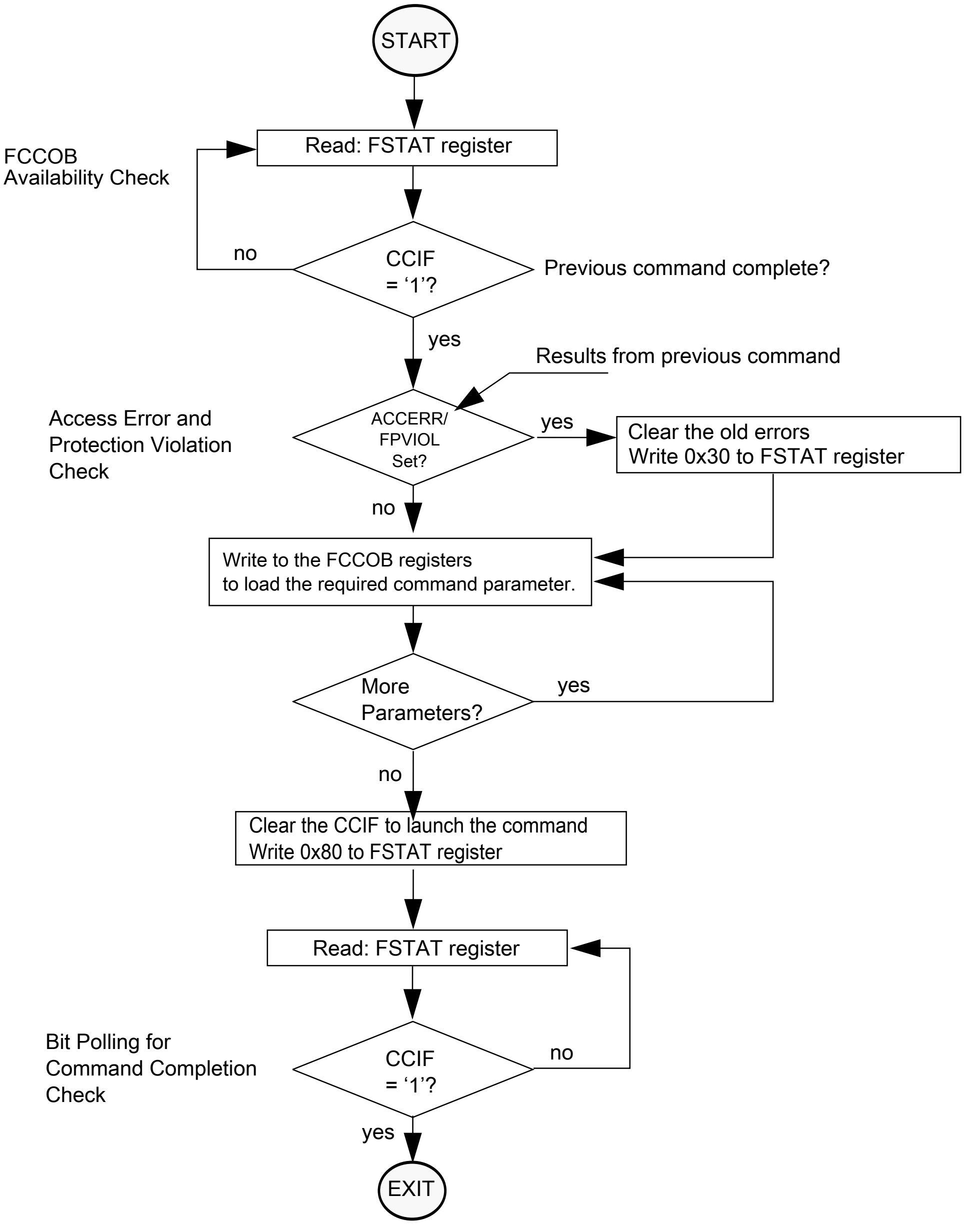


Figure 35-9. Generic flash command write sequence flowchart

35.4.9.2 Flash Commands

The following table summarizes the function of all flash commands. If the program flash, data flash, or FlexRAM column is marked with an 'X', the flash command is relevant to that particular memory resource.

FCMD	Command	Program flash 0	Program flash 1 (Devices with only program flash)	Data flash (Devices with FlexNVM)	FlexRAM (Devices with FlexNVM)	Function
0x00	Read 1s Block	×	×	×		Verify that a program flash or data flash block is erased. FlexNVM block must not be partitioned for EEPROM.
0x01	Read 1s Section	×	×	×		Verify that a given number of program flash or data flash locations from a starting address are erased.
0x02	Program Check	×	×	×		Tests previously-programmed locations at margin read levels.
0x03	Read Resource	IFR, ID	IFR	IFR		Read 4 bytes from program flash IFR, data flash IFR, or version ID.
0x06	Program Longword	×	×	×		Program 4 bytes in a program flash block or a data flash block.
0x08	Erase Flash Block	×	×	×		Erase a program flash block or data flash block. An erase of any flash block is only possible when unprotected. FlexNVM block must not be partitioned for EEPROM.
0x09	Erase Flash Sector	×	×	×		Erase all bytes in a program flash or data flash sector.

Table continues on the next page...

FCMD	Command	Program flash 0	Program flash 1 (Devices with only program flash)	Data flash (Devices with FlexNVM)	FlexRAM (Devices with FlexNVM)	Function
0x0B	Program Section	×	×	×	×	Program data from the Section Program Buffer to a program flash or data flash block.
0x40	Read 1s All Blocks	×	×	×		Verify that all program flash, data flash blocks, EEPROM backup data records, and data flash IFR are erased then release MCU security.
0x41	Read Once	IFR				Read 4 bytes of a dedicated 64 byte field in the program flash 0 IFR.
0x43	Program Once	IFR				One-time program of 4 bytes of a dedicated 64-byte field in the program flash 0 IFR.
0x44	Erase All Blocks	×	×	×	×	Erase all program flash blocks, program flash swap IFR, data flash blocks, FlexRAM, EEPROM backup data records, and data flash IFR. Then, verify-erase and release MCU security. **NOTE:** An erase is only possible when all memory

Table continues on the next page...

FCMD	Command	Program flash 0	Program flash 1 (Devices with only program flash)	Data flash (Devices with FlexNVM)	FlexRAM (Devices with FlexNVM)	Function
						locations are unprotected.
0x45	Verify Backdoor Access Key	×	×			Release MCU security after comparing a set of user-supplied security keys to those stored in the program flash.
0x46	Swap Control	×	×			Handles swap-related activities
0x80	Program Partition			IFR	×	Program the FlexNVM Partition Code and EEPROM Data Set Size into the data flash IFR. Format all EEPROM backup data sectors allocated for EEPROM. Initialize the FlexRAM.
0x81	Set FlexRAM Function			x	×	Switches FlexRAM function between RAM and EEPROM. When switching to EEPROM, FlexNVM is not available while valid data records are being copied from EEPROM backup to FlexRAM.

NOTE

FlexRAM, or Programming Acceleration RAM, is used during PGMSEC command.

35.4.9.3 Allowed Simultaneous Flash Operations

Only the operations marked 'OK' in the following table are permitted to run simultaneously on the program flash, data flash, and FlexRAM memories. Some operations cannot be executed simultaneously because certain hardware resources are shared by the memories. The priority has been placed on permitting program flash reads while program and erase operations execute on the FlexNVM and FlexRAM. This provides read (program flash) while write (FlexNVM, FlexRAM) functionality.

For devices containing FlexNVM:

Table 35-6. Allowed Simultaneous Memory Operations

		Program Flash			Data Flash			FlexRAM		
		Read	Program	Sector Erase	Read	Program	Sector Erase	Read	E-Write	R-Write
Program flash	Read	—				OK	OK		OK	
	Program		—		OK			OK		OK[3]
	Sector Erase			—	OK			OK		OK
Data flash	Read		OK	OK	—					
	Program	OK				—		OK		OK
	Sector Erase	OK					—	OK		OK
FlexRAM	Read		OK	OK		OK	OK	—		
	E-Write[1]	OK							—	
	R-Write[2]		OK	OK		OK	OK			—

1. When FlexRAM configured for EEPROM (writes are effectively multi-cycle operations).
2. When FlexRAM configured as traditional RAM (writes are single-cycle operations).
3. When FlexRAM configured as traditional RAM, writes to the RAM are ignored while the Program Section command is active (CCIF = 0).

For devices containing program flash only:

Table 35-7. Allowed Simultaneous Memory Operations

		Program Flash 0			Program Flash 1		
		Read	Program	Sector Erase	Read	Program	Sector Erase
Program flash 0	Read	—				OK	OK
	Program		—		OK		
	Sector Erase			—	OK		
Program flash 1	Read		OK	OK	—		
	Program	OK				—	
	Sector Erase	OK					—

35.4.10 Margin Read Commands

The Read-1s commands (Read 1s All Blocks, Read 1s Block, Read 1s Section) and the Program Check command have a margin choice parameter that allows the user to apply non-standard read reference levels to the program flash and data flash array reads performed by these commands. Using the preset 'user' and 'factory' margin levels, these commands perform their associated read operations at tighter tolerances than a 'normal' read. These non-standard read levels are applied only during the command execution. Basic flash array reads use the standard, un-margined, read reference level.

Only the 'normal' read level should be employed during normal flash usage. The non-standard, 'user' and 'factory' margin levels should be employed only in special cases. They can be used during special diagnostic routines to gain confidence that the device is not suffering from the end-of-life data loss customary of flash memory devices.

Erased ('1') and programmed ('0') bit states can degrade due to elapsed time and data cycling (number of times a bit is erased and re-programmed). The lifetime of the erased states is relative to the last erase operation. The lifetime of the programmed states is measured from the last program time.

The 'user' and 'factory' levels become, in effect, a minimum safety margin; i.e. if the reads pass at the tighter tolerances of the 'user' and 'factory' margins, then the 'normal' reads have at least this much safety margin before they experience data loss.

The 'user' margin is a small delta to the normal read reference level. 'User' margin levels can be employed to check that flash memory contents have adequate margin for normal level read operations. If unexpected read results are encountered when checking flash memory contents at the 'user' margin levels, loss of information might soon occur during 'normal' readout.

The 'factory' margin is a bigger deviation from the norm, a more stringent read criteria that should only be attempted immediately (or very soon) after completion of an erase or program command, early in the cycling life. 'Factory' margin levels can be used to check that flash memory contents have adequate margin for long-term data retention at the normal level setting. If unexpected results are encountered when checking flash memory contents at 'factory' margin levels, the flash memory contents should be erased and reprogrammed.

CAUTION

Factory margin levels must only be used during verify of the initial factory programming.

35.4.11 Flash Command Description

This section describes all flash commands that can be launched by a command write sequence.

The flash memory module sets the FSTAT[ACCERR] bit and aborts the command execution if any of the following illegal conditions occur:

- There is an unrecognized command code in the FCCOB FCMD field.
- There is an error in a FCCOB field for the specific commands. Refer to the error handling table provided for each command.

Ensure that FSTAT[ACCERR] and FSTAT[FPVIOL] are cleared prior to starting the command write sequence. As described in Launch the Command by Clearing CCIF, a new command cannot be launched while these error flags are set.

Do not attempt to read a flash block while the flash memory module is running a command (FSTAT[CCIF] = 0) on that same block. The flash memory module may return invalid data to the MCU with the collision error flag (FSTAT[RDCOLERR]) set.

When required by the command, address bit 23 selects between:

- program flash (=0)
- data flash (=1)

CAUTION

Flash data must be in the erased state before being programmed. Cumulative programming of bits (adding more zeros) is not allowed.

35.4.11.1 Read 1s Block Command

The Read 1s Block command checks to see if an entire program flash or data flash block has been erased to the specified margin level. The FCCOB flash address bits determine which flash block is erase-verified.

Table 35-8. Read 1s Block Command FCCOB Requirements

FCCOB Number	FCCOB Contents [7:0]
0	0x00 (RD1BLK)
1	Flash address [23:16] in the flash block to be verified

Table continues on the next page...

Table 35-8. Read 1s Block Command FCCOB Requirements (continued)

FCCOB Number	FCCOB Contents [7:0]
2	Flash address [15:8] in the flash block to be verified
3	Flash address [7:0][1] in the flash block to be verified
4	Read-1 Margin Choice

[1] Must be longword aligned (Flash address [1:0] = 00).

After clearing CCIF to launch the Read 1s Block command, the flash memory module sets the read margin for 1s according to Table 35-9 and then reads all locations within the selected program flash or data flash block.

When the data flash is targeted, DEPART must be set for no EEPROM, else the Read 1s Block command aborts setting the FSTAT[ACCERR] bit. If the flash memory module fails to read all 1s (i.e. the flash block is not fully erased), the FSTAT[MGSTAT0] bit is set. The CCIF flag sets after the Read 1s Block operation has completed.

Table 35-9. Margin Level Choices for Read 1s Block

Read Margin Choice	Margin Level Description
0x00	Use the 'normal' read level for 1s
0x01	Apply the 'User' margin to the normal read-1 level
0x02	Apply the 'Factory' margin to the normal read-1 level

Table 35-10. Read 1s Block Command Error Handling

Error Condition	Error Bit
Command not available in current mode/security	FSTAT[ACCERR]
An invalid margin choice is specified	FSTAT[ACCERR]
Program flash is selected and the address is out of program flash range	FSTAT[ACCERR]
Data flash is selected and the address is out of data flash range	FSTAT[ACCERR]
Data flash is selected with EEPROM enabled	FSTAT[ACCERR]
Flash address is not longword aligned	FSTAT[ACCERR]
Read-1s fails	FSTAT[MGSTAT0]

35.4.11.2 Read 1s Section Command

The Read 1s Section command checks if a section of program flash or data flash memory is erased to the specified read margin level. The Read 1s Section command defines the starting address and the number of phrases to be verified.

Table 35-11. Read 1s Section Command FCCOB Requirements

FCCOB Number	FCCOB Contents [7:0]
0	0x01 (RD1SEC)
1	Flash address [23:16] of the first phrase to be verified
2	Flash address [15:8] of the first phrase to be verified
3	Flash address [7:0][1] of the first phrase to be verified
4	Number of phrases to be verified [15:8]
5	Number of phrases to be verified [7:0]
6	Read-1 Margin Choice

[1] Must be phrase aligned (Flash address [2:0] = 000).

Upon clearing CCIF to launch the Read 1s Section command, the flash memory module sets the read margin for 1s according to Table 35-12 and then reads all locations within the specified section of flash memory. If the flash memory module fails to read all 1s (that is, the flash section is not erased), FSTAT[MGSTAT0] is set. FSTAT[CCIF] sets after the Read 1s Section operation completes.

Table 35-12. Margin Level Choices for Read 1s Section

Read Margin Choice	Margin Level Description
0x00	Use the 'normal' read level for 1s
0x01	Apply the 'User' margin to the normal read-1 level
0x02	Apply the 'Factory' margin to the normal read-1 level

Table 35-13. Read 1s Section Command Error Handling

Error condition	Error bit
Command not available in current mode/security	FSTAT[ACCERR]
An invalid margin code is supplied.	FSTAT[ACCERR]
An invalid flash address is supplied.	FSTAT[ACCERR]
Flash address is not phrase aligned.	FSTAT[ACCERR]
The requested section crosses a Flash block boundary.	FSTAT[ACCERR]
The requested number of phrases is 0.	FSTAT[ACCERR]
Read-1s fails.	FSTAT[MGSTAT0]

35.4.11.3 Program Check Command

The Program Check command tests a previously programmed program flash or data flash longword to see if it reads correctly at the specified margin level.

Table 35-14. Program Check Command FCCOB Requirements

FCCOB Number	FCCOB Contents [7:0]
0	0x02 (PGMCHK)
1	Flash address [23:16]
2	Flash address [15:8]
3	Flash address [7:0][1]
4	Margin Choice
8	Byte 0 expected data
9	Byte 1 expected data
A	Byte 2 expected data
B	Byte 3 expected data

[1] Must be longword aligned (Flash address [1:0] = 00).

Upon clearing CCIF to launch the Program Check command, the flash memory module sets the read margin for 1s according to Table 35-15, reads the specified longword, and compares the actual read data to the expected data provided by the FCCOB. If the comparison at margin-1 fails, FSTAT[MGSTAT0] is set.

The flash memory module then sets the read margin for 0s, re-reads, and compares again. If the comparison at margin-0 fails, FSTAT[MGSTAT0] is set. FSTAT[CCIF] is set after the Program Check operation completes.

The supplied address must be longword aligned (the lowest two bits of the byte address must be 00):

- Byte 3 data is written to the supplied byte address ('start'),
- Byte 2 data is programmed to byte address start+0b01,
- Byte 1 data is programmed to byte address start+0b10,
- Byte 0 data is programmed to byte address start+0b11.

NOTE

See the description of margin reads, Margin Read Commands

Table 35-15. Margin Level Choices for Program Check

Read Margin Choice	Margin Level Description
0x01	Read at 'User' margin-1 and 'User' margin-0
0x02	Read at 'Factory' margin-1 and 'Factory' margin-0

Table 35-16. Program Check Command Error Handling

Error Condition	Error Bit
Command not available in current mode/security	FSTAT[ACCERR]
An invalid flash address is supplied	FSTAT[ACCERR]

Table continues on the next page...

Table 35-16. Program Check Command Error Handling (continued)

Error Condition	Error Bit
Flash address is not longword aligned	FSTAT[ACCERR]
An invalid margin choice is supplied	FSTAT[ACCERR]
Either of the margin reads does not match the expected data	FSTAT[MGSTAT0]

35.4.11.4 Read Resource Command

The Read Resource command allows the user to read data from special-purpose memory resources located within the flash memory module. The special-purpose memory resources available include program flash IFR space, data flash IFR space, and the Version ID field. Each resource is assigned a select code as shown in Table 35-18.

Table 35-17. Read Resource Command FCCOB Requirements

FCCOB Number	FCCOB Contents [7:0]
0	0x03 (RDRSRC)
1	Flash address [23:16]
2	Flash address [15:8]
3	Flash address [7:0][1]
Returned Values	
4	Read Data [31:24]
5	Read Data [23:16]
6	Read Data [15:8]
7	Read Data [7:0]
User-provided values	
8	Resource Select Code (see Table 35-18)

[1] Must be longword aligned (Flash address [1:0] = 00).

Table 35-18. Read Resource Select Codes

Resource Select Code	Description	Resource Size	Local Address Range
0x00	Program Flash 0 IFR	256 Bytes	0x00_0000–0x00_00FF
0x00	Program Flash Swap IFR[1]	256 Bytes	0x02_0000 - 0x02_00FF (512 KB of program flash) 0x01_0000 - 0x01_00FF (256 KB of program flash) 0x00_8000 - 0x00_80FF (128 KB of program flash)

Table continues on the next page...

Table 35-18. Read Resource Select Codes (continued)

Resource Select Code	Description	Resource Size	Local Address Range
0x00	Data Flash 0 IFR[2]	256 Bytes	0x80_0000 - 0x80_00FF
0x01[3]	Version ID	8 Bytes	0x00_0000–0x00_0007

1. This is for devices with program flash only.
2. This is for devices with FlexNVM.
3. Located in program flash 0 reserved space.

After clearing CCIF to launch the Read Resource command, four consecutive bytes are read from the selected resource at the provided relative address and stored in the FCCOB register. The CCIF flag sets after the Read Resource operation completes. The Read Resource command exits with an access error if an invalid resource code is provided or if the address for the applicable area is out-of-range.

Table 35-19. Read Resource Command Error Handling

Error Condition	Error Bit
Command not available in current mode/security	FSTAT[ACCERR]
An invalid resource code is entered	FSTAT[ACCERR]
Flash address is out-of-range for the targeted resource.	FSTAT[ACCERR]
Flash address is not longword aligned	FSTAT[ACCERR]

35.4.11.5 Program Longword Command

The Program Longword command programs four previously-erased bytes in the program flash memory or in the data flash memory using an embedded algorithm.

CAUTION

A flash memory location must be in the erased state before being programmed. Cumulative programming of bits (back-to-back program operations without an intervening erase) within a flash memory location is not allowed. Re-programming of existing 0s to 0 is not allowed as this overstresses the device.

Table 35-20. Program Longword Command FCCOB Requirements

FCCOB Number	FCCOB Contents [7:0]
0	0x06 (PGM4)
1	Flash address [23:16]
2	Flash address [15:8]

Table continues on the next page...

Table 35-20. Program Longword Command FCCOB Requirements (continued)

FCCOB Number	FCCOB Contents [7:0]
3	Flash address [7:0][1]
4	Byte 0 program value
5	Byte 1 program value
6	Byte 2 program value
7	Byte 3 program value

[1] Must be longword aligned (Flash address [1:0] = 00).

Upon clearing CCIF to launch the Program Longword command, the flash memory module programs the data bytes into the flash using the supplied address. The swap indicator address in each program flash block is implicitly protected from programming. The targeted flash locations must be currently unprotected (see the description of the FPROT and FDPROT registers) to permit execution of the Program Longword operation.

The programming operation is unidirectional. It can only move NVM bits from the erased state ('1') to the programmed state ('0'). Erased bits that fail to program to the '0' state are flagged as errors in FSTAT[MGSTAT0]. The CCIF flag is set after the Program Longword operation completes.

The supplied address must be longword aligned (flash address [1:0] = 00):

- Byte 3 data is written to the supplied byte address ('start'),
- Byte 2 data is programmed to byte address start+0b01,
- Byte 1 data is programmed to byte address start+0b10, and
- Byte 0 data is programmed to byte address start+0b11.

Table 35-21. Program Longword Command Error Handling

Error Condition	Error Bit
Command not available in current mode/security	FSTAT[ACCERR]
An invalid flash address is supplied	FSTAT[ACCERR]
Flash address is not longword aligned	FSTAT[ACCERR]
Flash address points to a protected area	FSTAT[FPVIOL]
Any errors have been encountered during the verify operation	FSTAT[MGSTAT0]

35.4.11.6 Erase Flash Block Command

The Erase Flash Block operation erases all addresses in a single program flash or data flash block.

Table 35-22. Erase Flash Block Command FCCOB Requirements

FCCOB Number	FCCOB Contents [7:0]
0	0x08 (ERSBLK)
1	Flash address [23:16] in the flash block to be erased
2	Flash address [15:8] in the flash block to be erased
3	Flash address [7:0][1] in the flash block to be erased

[1] Must be longword aligned (Flash address [1:0] = 00).

Upon clearing CCIF to launch the Erase Flash Block command, the flash memory module erases the main array of the selected flash block and verifies that it is erased. When the data flash is targeted, DEPART must be set for no EEPROM (see Table 35-4) else the Erase Flash Block command aborts setting the FSTAT[ACCERR] bit. The Erase Flash Block command aborts and sets the FSTAT[FPVIOL] bit if any region within the block is protected (see the description of the FPROT and FDPROT registers). The swap indicator address in each program flash block is implicitly protected from block erase unless the swap system is in the UPDATE or UPDATE-ERASED state and the program flash block being erased is the non-active block. If the erase verify fails, FSTAT[MGSTAT0] is set. The CCIF flag will set after the Erase Flash Block operation has completed.

Table 35-23. Erase Flash Block Command Error Handling

Error Condition	Error Bit
Command not available in current mode/security	FSTAT[ACCERR]
Program flash is selected and the address is out of program flash range	FSTAT[ACCERR]
Data flash is selected and the address is out of data flash range	FSTAT[ACCERR]
Data flash is selected with EEPROM enabled	FSTAT[ACCERR]
Flash address is not longword aligned	FSTAT[ACCERR]
Any area of the selected flash block is protected	FSTAT[FPVIOL]
Any errors have been encountered during the verify operation[1]	FSTAT[MGSTAT0]

[1] User margin read may be run using the Read 1s Block command to verify all bits are erased.

35.4.11.7 Erase Flash Sector Command

The Erase Flash Sector operation erases all addresses in a flash sector.

Table 35-24. Erase Flash Sector Command FCCOB Requirements

FCCOB Number	FCCOB Contents [7:0]
0	0x09 (ERSSCR)
1	Flash address [23:16] in the flash sector to be erased

Table continues on the next page...

Table 35-24. Erase Flash Sector Command FCCOB Requirements (continued)

FCCOB Number	FCCOB Contents [7:0]
2	Flash address [15:8] in the flash sector to be erased
3	Flash address [7:0][1] in the flash sector to be erased

[1] Must be phrase aligned (flash address [2:0] = 000).

After clearing CCIF to launch the Erase Flash Sector command, the flash memory module erases the selected program flash or data flash sector and then verifies that it is erased. The Erase Flash Sector command aborts if the selected sector is protected (see the description of the FPROT and FDPROT registers). The swap indicator address in each program flash block is implicitly protected from sector erase unless the swap system is in the UPDATE or UPDATE-ERASED state and the program flash sector containing the swap indicator address being erased is the non-active block. If the erase-verify fails the FSTAT[MGSTAT0] bit is set. The CCIF flag is set after the Erase Flash Sector operation completes. The Erase Flash Sector command is suspendable (see the FCNFG[ERSSUSP] bit and Figure 35-10).

Table 35-25. Erase Flash Sector Command Error Handling

Error Condition	Error Bit
Command not available in current mode/security	FSTAT[ACCERR]
An invalid Flash address is supplied	FSTAT[ACCERR]
Flash address is not phrase aligned	FSTAT[ACCERR]
The selected program flash or data flash sector is protected	FSTAT[FPVIOL]
Any errors have been encountered during the verify operation[1]	FSTAT[MGSTAT0]

[1] User margin read may be run using the Read 1s Section command to verify all bits are erased.

35.4.11.7.1 Suspending an Erase Flash Sector Operation

To suspend an Erase Flash Sector operation set the FCNFG[ERSSUSP] bit when CCIF, ACCERR, and FPVIOL are clear and the CCOB command field holds the code for the Erase Flash Sector command. During the Erase Flash Sector operation (see Erase Flash Sector Command), the flash memory module samples the state of the ERSSUSP bit at convenient points. If the flash memory module detects that the ERSSUSP bit is set, the Erase Flash Sector operation is suspended and the flash memory module sets CCIF. While ERSSUSP is set, all writes to flash registers are ignored except for writes to the FSTAT and FCNFG registers.

If an Erase Flash Sector operation effectively completes before the flash memory module detects that a suspend request has been made, the flash memory module clears the ERSSUSP bit prior to setting CCIF. When an Erase Flash Sector operation has been successfully suspended, the flash memory module sets CCIF and leaves the ERSSUSP bit set. While CCIF is set, the ERSSUSP bit can only be cleared to prevent the withdrawal of a suspend request before the flash memory module has acknowledged it.

35.4.11.7.2 Resuming a Suspended Erase Flash Sector Operation

If the ERSSUSP bit is still set when CCIF is cleared to launch the next command, the previous Erase Flash Sector operation resumes. The flash memory module acknowledges the request to resume a suspended operation by clearing the ERSSUSP bit. A new suspend request can then be made by setting ERSSUSP. A single Erase Flash Sector operation can be suspended and resumed multiple times.

There is a minimum elapsed time limit of 4.3 msec between the request to resume the Erase Flash Sector operation (CCIF is cleared) and the request to suspend the operation again (ERSSUSP is set). This minimum time period is required to ensure that the Erase Flash Sector operation will eventually complete. If the minimum period is continually violated, i.e. the suspend requests come repeatedly and too quickly, no forward progress is made by the Erase Flash Sector algorithm. The resume/suspend sequence runs indefinitely without completing the erase.

35.4.11.7.3 Aborting a Suspended Erase Flash Sector Operation

The user may choose to abort a suspended Erase Flash Sector operation by clearing the ERSSUSP bit prior to clearing CCIF for the next command launch. When a suspended operation is aborted, the flash memory module starts the new command using the new FCCOB contents.

While FCNFG[ERSSUSP] is set, a write to the FlexRAM while FCNFG[EEERDY] is set clears ERSSUSP and aborts the suspended operation. The FlexRAM write operation is executed by the flash memory module.

Note

> Aborting the erase leaves the bitcells in an indeterminate, partially-erased state. Data in this sector is not reliable until a new erase command fully completes.

The following figure shows how to suspend and resume the Erase Flash Sector operation.

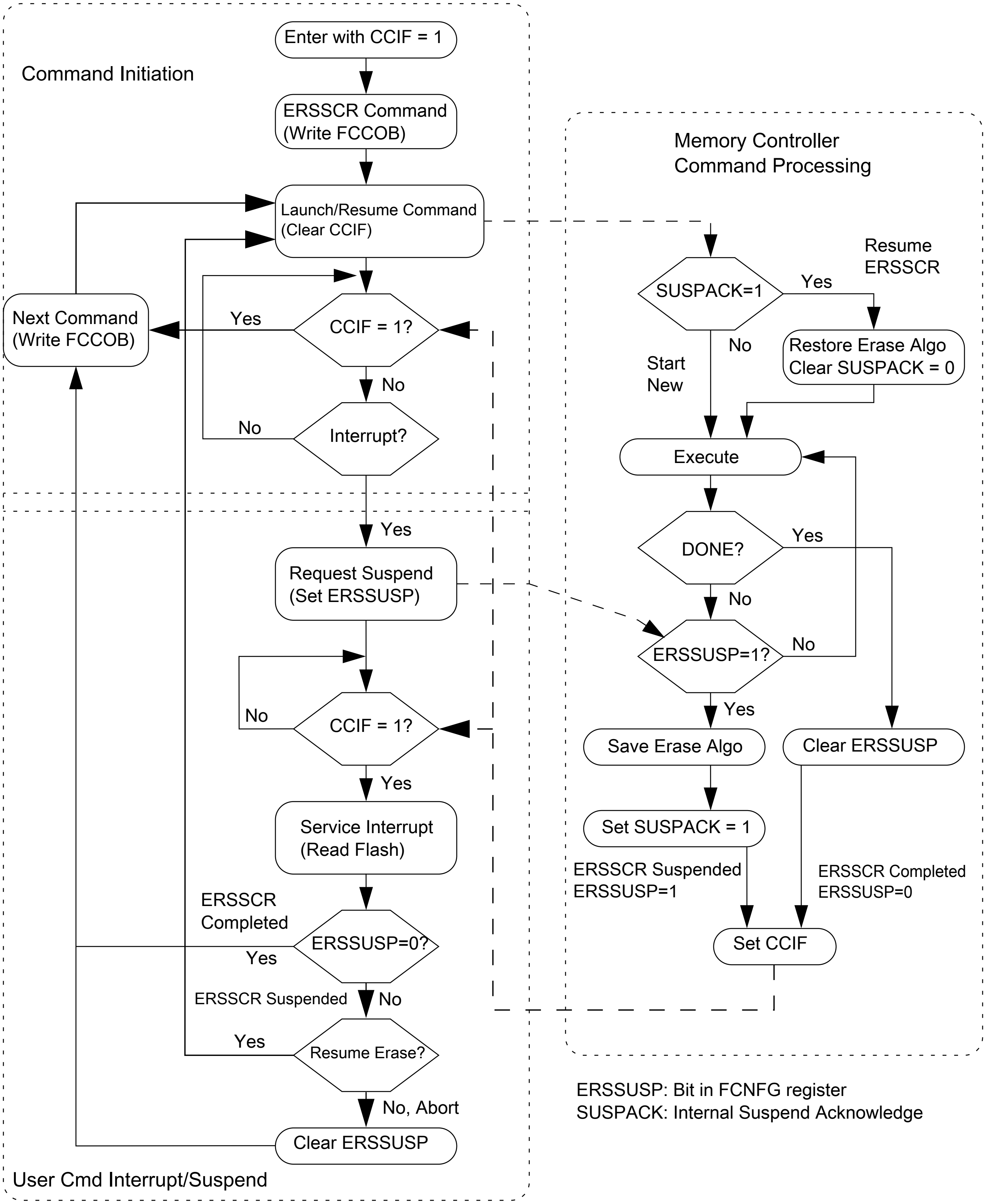

Enter with CCIF = 1
Command Initiation
ERSSCR Command (Write FCCOB)
Launch/Resume Command (Clear CCIF)
Next Command (Write FCCOB)
Yes
CCIF = 1?
No
No
Interrupt?
Yes
Request Suspend (Set ERSSUSP)
No
CCIF = 1?
Yes
Service Interrupt (Read Flash)
ERSSCR Completed
ERSSUSP=0?
Yes
ERSSCR Suspended
No
Yes
Resume Erase?
No, Abort
Clear ERSSUSP
User Cmd Interrupt/Suspend
Memory Controller Command Processing
SUSPACK=1
Yes
Resume ERSSCR
Restore Erase Algo Clear SUSPACK = 0
No
Start New
Execute
DONE?
Yes
No
ERSSUSP=1?
No
Yes
Save Erase Algo
Clear ERSSUSP
Set SUSPACK = 1
ERSSCR Suspended ERSSUSP=1
ERSSCR Completed ERSSUSP=0
Set CCIF
ERSSUSP: Bit in FCNFG register
SUSPACK: Internal Suspend Acknowledge

Figure 35-10. Suspend and Resume of Erase Flash Sector Operation

35.4.11.8 Program Section Command

The Program Section operation programs the data found in the section program buffer to previously erased locations in the flash memory using an embedded algorithm. Data is preloaded into the section program buffer by writing to the FlexRAM while it is set to function as traditional RAM or the programming acceleration RAM (see Flash Sector Programming).

The section program buffer is limited to the lower half of the RAM. Data written to the upper half of the RAM is ignored and may be overwritten during Program Section command execution.

CAUTION

> A flash memory location must be in the erased state before being programmed. Cumulative programming of bits (back-to-back program operations without an intervening erase) within a flash memory location is not allowed. Re-programming of existing 0s to 0 is not allowed as this overstresses the device.

Table 35-26. Program Section Command FCCOB Requirements

FCCOB Number	FCCOB Contents [7:0]
0	0x0B (PGMSEC)
1	Flash address [23:16]
2	Flash address [15:8]
3	Flash address [7:0][1]
4	Number of phrases to program [15:8]
5	Number of phrases to program [7:0]

[1] Must be phrase aligned (Flash address [2:0] = 000).

After clearing CCIF to launch the Program Section command, the flash memory module blocks access to the programming acceleration RAM (program flash only devices) or FlexRAM (FlexNVM devices) and programs the data residing in the section program buffer into the flash memory starting at the flash address provided.

The starting address must be unprotected (see the description of the FPROT and FDPROT registers) to permit execution of the Program Section operation. The swap indicator address in each program flash block is implicitly protected from programming. If the swap indicator address is encountered during the Program Section operation, it is bypassed without setting FPVIOL and the contents are not programmed. Programming, which is not allowed to cross a flash sector boundary, continues until all requested phrases have been programmed. The Program Section command also verifies that after programming, all bits requested to be programmed are programmed.

After the Program Section operation completes, the CCIF flag is set and normal access to the RAM is restored. The contents of the section program buffer may be changed by the Program Section operation.

Table 35-27. Program Section Command Error Handling

Error Condition	Error Bit
Command not available in current mode/security	FSTAT[ACCERR]
An invalid flash address is supplied	FSTAT[ACCERR]
Flash address is not phrase aligned	FSTAT[ACCERR]
The requested section crosses a program flash sector boundary	FSTAT[ACCERR]
The requested number of phrases is zero	FSTAT[ACCERR]
The space required to store data for the requested number of phrases is more than half the size of the programming acceleration RAM (program flash only devices) or FlexRAM (FlexNVM devices)	FSTAT[ACCERR]
The FlexRAM is not set to function as a traditional RAM, i.e. set if RAMRDY=0	FSTAT[ACCERR]
The flash address falls in a protected area	FSTAT[FPVIOL]
Any errors have been encountered during the verify operation	FSTAT[MGSTAT0]

35.4.11.8.1 Flash Sector Programming

The process of programming an entire flash sector using the Program Section command is as follows:

1. If required, for FlexNVM devices, execute the Set FlexRAM Function command to make the FlexRAM available as traditional RAM and initialize the FlexRAM to all ones.
2. Launch the Erase Flash Sector command to erase the flash sector to be programmed.
3. Beginning with the starting address of the programming acceleration RAM (program flash only devices) or FlexRAM (FlexNVM devices), sequentially write enough data to the RAM to fill an entire flash sector. This area of the RAM serves as the section program buffer.

   NOTE

   In step 1, the section program buffer was initialized to all ones, the erased state of the flash memory.

   The section program buffer can be written to while the operation launched in step 2 is executing, i.e. while CCIF = 0.
4. Execute the Program Section command to program the contents of the section program buffer into the selected flash sector.
5. If a flash sector is larger than half the RAM, repeat steps 3 and 4 until the sector is completely programmed.
6. To program additional flash sectors, repeat steps 2 through 4.

7. To restore EEPROM functionality for FlexNVM devices, execute the Set FlexRAM Function command to make the FlexRAM available as EEPROM.

35.4.11.9 Read 1s All Blocks Command

The Read 1s All Blocks command checks if the program flash blocks, data flash blocks, EEPROM backup records, and data flash IFR have been erased to the specified read margin level, if applicable, and releases security if the readout passes, i.e. all data reads as '1'.

Table 35-28. Read 1s All Blocks Command FCCOB Requirements

FCCOB Number	FCCOB Contents [7:0]
0	0x40 (RD1ALL)
1	Read-1 Margin Choice

After clearing CCIF to launch the Read 1s All Blocks command, the flash memory module :

- sets the read margin for 1s according to Table 35-29,
- checks the contents of the program flash, data flash, EEPROM backup records, and data flash IFR are in the erased state.

If the flash memory module confirms that these memory resources are erased, security is released by setting the FSEC[SEC] field to the unsecure state. The security byte in the flash configuration field (see Flash Configuration Field Description) remains unaffected by the Read 1s All Blocks command. If the read fails, i.e. all memory resources are not in the fully erased state, the FSTAT[MGSTAT0] bit is set.

The EEERDY and RAMRDY bits are clear during the Read 1s All Blocks operation and are restored at the end of the Read 1s All Blocks operation.

The CCIF flag sets after the Read 1s All Blocks operation has completed.

Table 35-29. Margin Level Choices for Read 1s All Blocks

Read Margin Choice	Margin Level Description
0x00	Use the 'normal' read level for 1s
0x01	Apply the 'User' margin to the normal read-1 level
0x02	Apply the 'Factory' margin to the normal read-1 level

Table 35-30. Read 1s All Blocks Command Error Handling

Error Condition	Error Bit
Command not available in current mode/security	FSTAT[ACCERR]
An invalid margin choice is specified	FSTAT[ACCERR]
Read-1s fails	FSTAT[MGSTAT0]

35.4.11.10 Read Once Command

The Read Once command provides read access to special 64-byte fields located in the program flash 0 IFR (see Program Flash IFR Map and Program Once Field). Access to the Program Once field is via 16 records (index values 0x00 - 0x0F), each 4 bytes long. These fields are programmed using the Program Once command described in Program Once Command.

Table 35-31. Read Once Command FCCOB Requirements

FCCOB Number	FCCOB Contents [7:0]
0	0x41 (RDONCE)
1	Program Once record index (0x00 - 0x0F)
2	Not used
3	Not used
Returned Values	
4	Program Once byte 0 value
5	Program Once byte 1 value
6	Program Once byte 2 value
7	Program Once byte 3 value

After clearing CCIF to launch the Read Once command, a 4-byte Program Once record is read and stored in the FCCOB register. The CCIF flag is set after the Read Once operation completes. Valid record index values for the Read Once command range from 0x00 - 0x0F. During execution of the Read Once command, any attempt to read addresses within the program flash block containing the selected record index returns invalid data. The Read Once command can be executed any number of times.

Table 35-32. Read Once Command Error Handling

Error Condition	Error Bit
Command not available in current mode/security	FSTAT[ACCERR]
An invalid record index is supplied	FSTAT[ACCERR]

35.4.11.11 Program Once Command

The Program Once command enables programming to special 64-byte fields in the program flash 0 IFR (see Program Flash IFR Map and Program Once Field). Access to the Program Once field is via 16 records (index values 0x00 - 0x0F), each 4 bytes long. These records can be read using the Read Once command (see Read Once Command) or using the Read Resource command (see Read Resource Command). These records can be programmed only once since the program flash 0 IFR cannot be erased.

Table 35-33. Program Once Command FCCOB Requirements

FCCOB Number	FCCOB Contents [7:0]
0	0x43 (PGMONCE)
1	Program Once record index (0x00 - 0x0F)
2	Not Used
3	Not Used
4	Program Once byte 0 value
5	Program Once byte 1 value
6	Program Once byte 2 value
7	Program Once byte 3 value

After clearing CCIF to launch the Program Once command, the flash memory module first verifies that the selected record is erased. If erased, then the selected record is programmed using the values provided. The Program Once command also verifies that the programmed values read back correctly. The CCIF flag is set after the Program Once operation has completed.

Any attempt to program one of these records when the existing value is not Fs (erased) is not allowed. Valid record index values for the Program Once command range from 0x00 - 0x0F. During execution of the Program Once command, any attempt to read addresses within the program flash block containing the selected record index returns invalid data.

Table 35-34. Program Once Command Error Handling

Error Condition	Error Bit
Command not available in current mode/security	FSTAT[ACCERR]
An invalid record index is supplied	FSTAT[ACCERR]
The requested record has already been programmed to a non-FFFF value[1]	FSTAT[ACCERR]
Any errors have been encountered during the verify operation	FSTAT[MGSTAT0]

[1] If a Program Once record is initially programmed to 0xFFFF_FFFF, the Program Once command is allowed to execute again on that same record.

35.4.11.12 Erase All Blocks Command

The Erase All Blocks operation erases all flash memory, initializes the FlexRAM, verifies all memory contents, and releases MCU security.

Table 35-35. Erase All Blocks Command FCCOB Requirements

FCCOB Number	FCCOB Contents [7:0]
0	0x44 (ERSALL)

After clearing CCIF to launch the Erase All Blocks command, the flash memory module erases all program flash memory, program flash swap IFR space, data flash memory, data flash IFR space, EEPROM backup memory, and FlexRAM, then verifies that all are erased.

If the flash memory module verifies that all flash memories and the FlexRAM were properly erased, security is released by setting the FSEC[SEC] field to the unsecure state and the FCNFG[RAMRDY] bit is set. The Erase All Blocks command aborts if any flash or FlexRAM region is protected. The swap indicator address in each program flash block is not implicitly protected from the Erase All Blocks operation. The security byte and all other contents of the flash configuration field (see Flash Configuration Field Description) are erased by the Erase All Blocks command. If the erase-verify fails, the FSTAT[MGSTAT0] bit is set. The CCIF flag is set after the Erase All Blocks operation completes.

Table 35-36. Erase All Blocks Command Error Handling

Error Condition	Error Bit
Command not available in current mode/security	FSTAT[ACCERR]
Any region of the program flash memory, data flash memory, or FlexRAM is protected	FSTAT[FPVIOL]
Any errors have been encountered during the verify operation[1]	FSTAT[MGSTAT0]

[1] User margin read may be run using the Read 1s All Blocks command to verify all bits are erased.

35.4.11.12.1 Triggering an Erase All External to the Flash Memory Module

The functionality of the Erase All Blocks command is also available in an uncommanded fashion outside of the flash memory. Refer to the device's Chip Configuration details for information on this functionality.

Before invoking the external erase all function, the FSTAT[ACCERR and PVIOL] flags must be cleared and the FCCOB0 register must not contain 0x44. When invoked, the erase-all function erases all program flash memory, program flash swap IFR space, data flash memory, data flash IFR space, EEPROM backup, and FlexRAM regardless of the protection settings or if the swap system has been initialized. If the post-erase verify passes, the routine then releases security by setting the FSEC[SEC] field register to the unsecure state and the FCNFG[RAMRDY] bit sets. The security byte in the Flash Configuration Field is also programmed to the unsecure state. The status of the erase-all request is reflected in the FCNFG[ERSAREQ] bit. The FCNFG[ERSAREQ] bit is cleared once the operation completes and the normal FSTAT error reporting is available, except FPVIOL, as described in Erase All Blocks Command.

CAUTION

> Since the IFR Swap Field in the program flash swap IFR containing the swap indicator address is erased during the Erase All Blocks command operation, the swap system becomes uninitialized. The Swap Control command must be run with the initialization code to set the swap indicator address and initialize the swap system.

35.4.11.13 Verify Backdoor Access Key Command

The Verify Backdoor Access Key command only executes if the mode and security conditions are satisfied (see Flash Commands by Mode). Execution of the Verify Backdoor Access Key command is further qualified by the FSEC[KEYEN] bits. The Verify Backdoor Access Key command releases security if user-supplied keys in the FCCOB match those stored in the Backdoor Comparison Key bytes of the Flash Configuration Field (see Flash Configuration Field Description). The column labelled Flash Configuration Field offset address shows the location of the matching byte in the Flash Configuration Field.

Table 35-37. Verify Backdoor Access Key Command FCCOB Requirements

FCCOB Number	FCCOB Contents [7:0]	Flash Configuration Field Offset Address
0	0x45 (VFYKEY)	
1-3	Not Used	
4	Key Byte 0	0x0_0003
5	Key Byte 1	0x0_0002
6	Key Byte 2	0x0_0001
7	Key Byte 3	0x0_0000
8	Key Byte 4	0x0_0007
9	Key Byte 5	0x0_0006
A	Key Byte 6	0x0_0005
B	Key Byte 7	0x0_0004

After clearing CCIF to launch the Verify Backdoor Access Key command, the flash memory module checks the FSEC[KEYEN] bits to verify that this command is enabled. If not enabled, the flash memory module sets the FSTAT[ACCERR] bit and terminates. If the command is enabled, the flash memory module compares the key provided in FCCOB to the backdoor comparison key in the Flash Configuration Field. If the backdoor keys match, the FSEC[SEC] field is changed to the unsecure state and security is released. If the backdoor keys do not match, security is not released and all future attempts to execute the Verify Backdoor Access Key command are immediately aborted and the FSTAT[ACCERR] bit is (again) set to 1 until a reset of the flash memory module module occurs. If the entire 8-byte key is all zeros or all ones, the Verify Backdoor Access Key command fails with an access error. The CCIF flag is set after the Verify Backdoor Access Key operation completes.

Table 35-38. Verify Backdoor Access Key Command Error Handling

Error Condition	Error Bit
Command not available in current mode/security	FSTAT[ACCERR]
The supplied key is all-0s or all-Fs	FSTAT[ACCERR]
An incorrect backdoor key is supplied	FSTAT[ACCERR]
Backdoor key access has not been enabled (see the description of the FSEC register)	FSTAT[ACCERR]
This command is launched and the backdoor key has mismatched since the last power down reset	FSTAT[ACCERR]

35.4.11.14 Swap Control Command

The Swap Control command handles specific activities associated with swapping the two logical program flash memory blocks within the memory map.

Table 35-39. Swap Control Command FCCOB Requirements

FCCOB Number	FCCOB Contents [7:0]
0	0x46 (SWAP)
1	Flash address [23:16]
2	Flash address [15:8]
3	Flash address [7:0] [1]
4	Swap Control Code: 0x01 - Initialize Swap System 0x02 - Set Swap in Update State 0x04 - Set Swap in Complete State 0x08 - Report Swap Status
Returned values	

Table continues on the next page...

Table 35-39. Swap Control Command FCCOB Requirements (continued)

FCCOB Number	FCCOB Contents [7:0]
5	Current Swap State: 0x00 - Uninitialized 0x01 - Ready 0x02 - Update 0x03 - Update-Erased 0x04 - Complete
6	Current Swap Block Status: 0x00 - Program flash block 0 at 0x0_0000 0x01 - Program flash block 1 at 0x0_0000
7	Next Swap Block Status (after any reset): 0x00 - Program flash block 0 at 0x0_0000 0X01 - Program flash block 1 at 0x0_0000

1. Must be phrase-aligned (Flash address [2:0] = 000).

Upon clearing CCIF to launch the Swap Control command, the flash memory module will handle swap-related activities based on the swap control code provided in FCCOB4 as follows:

- 0x01 (Initialize Swap System to UPDATE-ERASED State) - After verifying that the current swap state is UNINITIALIZED and that the flash address provided is in Program flash block 0 but not in the Flash Configuration Field, the flash address (shifted with bits[2:0] removed) will be programmed into the IFR Swap Field found in program flash swap IFR. After the swap indicator address has been programmed into the IFR Swap Field, the swap enable word will be programmed to 0x0000. After the swap enable word has been programmed, the swap indicator, located within the Program flash block 0 address provided, will be programmed to 0xFF00.
- 0x02 (Progress Swap to UPDATE State) - After verifying that the current swap state is READY and that the flash address provided matches the one stored in the IFR Swap Field, the swap indicator located within bits [15:0] of the flash address in the currently active program flash block will be programmed to 0xFF00.
- 0x04 (Progress Swap to COMPLETE State) - After verifying that the current swap state is UPDATE-ERASED and that the flash address provided matches the one stored in the IFR Swap Field, the swap indicator located within bits [15:0] of the flash address in the currently active program flash block will be programmed to 0x0000. Before executing with this swap control code, the user must erase the non-active swap indicator using the Erase Flash Block or Erase Flash Sector commands and update the application code or data as needed. The non-active swap indicator will

be checked at the erase verify level and if the check fails, the current swap state will be changed to UPDATE with FSTAT[ACCERR] set.

- 0x08 (Report Swap System Status) - After verifying that the flash address provided matches the one stored in the IFR Swap Field, the status of the swap system will be reported as follows:
  - FCCOB5 (Current Swap State) - indicates the current swap state based on the status of the swap enable word and the swap indicators. If the FSTAT[MGSTAT0] flag is set after command completion, the swap state returned was not successfully transitioned from and the appropriate swap command code must be attempted again. If the current swap state is UPDATE and the non-active swap indicator is 0xFFFF, the current swap state is changed to UPDATE-ERASED.
  - FCCOB6 (Current Swap Block Status) - indicates which program flash block is currently located at relative flash address 0x0_0000.
  - FCCOB7 (Next Swap Block Status) - indicates which program flash block will be located at relative flash address 0x0_0000 after the next reset of the flash memory module.

NOTE

It is recommended that the user execute the Swap Control command to report swap status (code 0x08) after any reset to determine if issues with the swap system were detected during the swap state determination procedure.

NOTE

It is recommended that the user write 0xFF to FCCOB5, FCCOB6, and FCCOB7 since the Swap Control command will not always return the swap state and status fields when an access error is detected.

The swap indicators are implicitly protected from being programmed during Program Longword or Program Section command operations and are implicitly unprotected during Swap Control command operations. The swap indicators are implicitly protected from being erased during Erase Flash Block and Erase Flash Sector command operations unless the swap indicator being erased is in the non-active program flash block and the swap system is in the UPDATE or UPDATE-ERASED state. Once the swap system has been initialized, the Erase All Blocks command can be used to uninitialize the swap system.

Table 35-40. Swap Control Command Error Handling

Error Condition	Swap Control Code	Error Bit
Command not available in current mode/security[1]	All	FSTAT[ACCERR]
Flash address is not in program flash block 0	All	FSTAT[ACCERR]
Flash address is in the Flash Configuration Field	All	FSTAT[ACCERR]
Flash address is not phrase aligned	All	FSTAT[ACCERR]
Flash address does not match the swap indicator address in the IFR	2, 4	FSTAT[ACCERR]
Swap initialize requested when swap system is not in the uninitialized state	1	FSTAT[ACCERR]
Swap update requested when swap system is not in the ready state	2	FSTAT[ACCERR]
Swap complete requested when swap system is not in the update-erased state	4	FSTAT[ACCERR]
An undefined swap control code is provided	-	FSTAT[ACCERR]
Any errors have been encountered during the swap determination and program-verify operations	1, 2, 4	FSTAT[MGSTAT0]
Any brownouts were detected during the swap determination procedure	8	FSTAT[MGSTAT0]

[1] Returned fields will not be updated, i.e. no swap state or status reporting

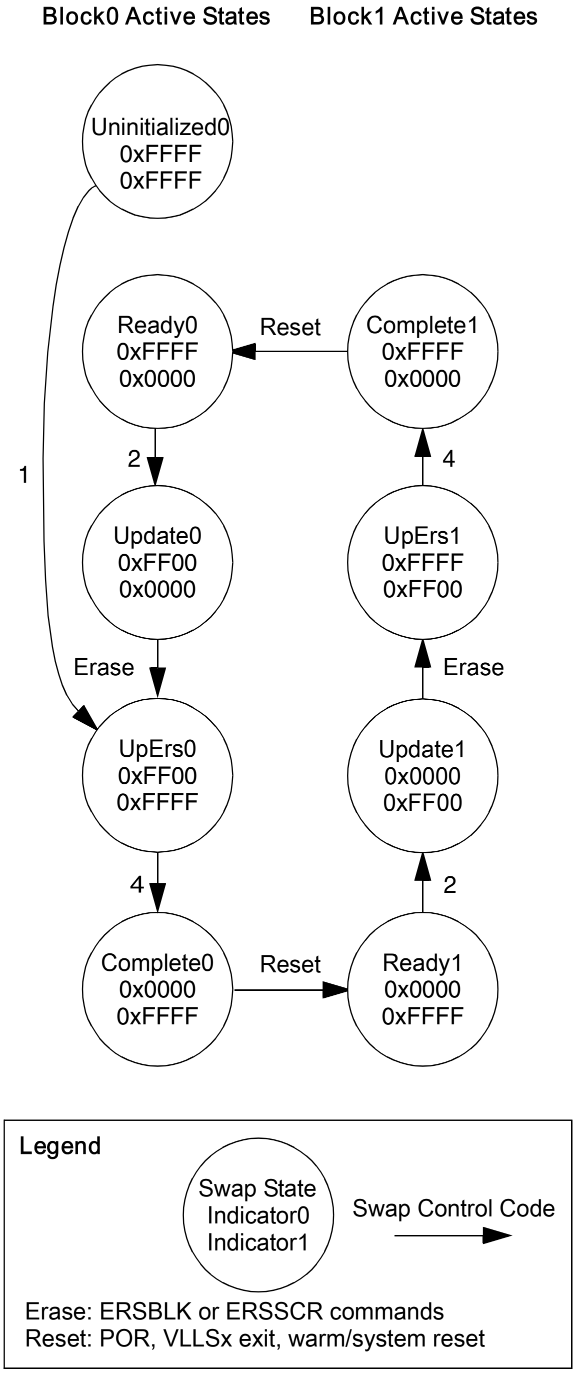


Figure 35-11. Valid Swap State Sequencing

Table 35-41. Swap State Report Mapping

Case	Swap Enable Field	Swap Indicator 0[1]	Swap Indicator 1[1]	Swap State[2]	State Code	MGST AT0	Active Block
1	0xFFFF	-	-	Uninitialized	0	0	0
2	0x0000	0xFF00	0x0000	Update	2	0	0
3	0x0000	0xFF00-	0xFFFF	Update-Erased	3	0	0
4	0x0000	0x0000	0xFFFF	Complete	4	0	0
5	0x0000	0x0000	0xFFFF	Ready	1	0	1
6	0x0000	0x0000	0xFF00	Update	2	0	1
7	0x0000	0xFFFF	0xFF00	Update-Erased	3	0	1
8	0x0000	0xFFFF[3]	0x0000	Complete[4]	4	0	1
9	0x0000	0xFFFF	0x0000	Ready[5]	1	0	0
10	0xXXXX	-	-	Uninitialized	0	1	0
11	0x0000	0xFFFF	0xFFFF	Uninitialized	0	1	0
12	0x0000	0xFFXX	0xFFFF	Ready	1	1	0
13	0x0000	0xFFXX	0x0000	Ready	1	1	0
14	0x0000	0xXXXX	0x0000	Ready	1	1	0
15[6]	0x0000	0xFFFF	0xFFXX	Ready	1	1	1
16	0x0000	0x0000	0xFFXX	Ready	1	1	1
17[6]	0x0000	0x0000	0xXXXX	Ready	1	1	1
18	0x0000	0xFF00	0xFFFF	Update	2	1	0
19	0x0000	0xFF00	0xXXXX	Update	2	1	0
20	0x0000	0xFF(00)	0xFFXX	Update	2	1	0
21[6]	0x0000	0x0000	0x0000	Update	2	1	0
22[6]	0x0000	0xXXXX	0xXXXX	Update	2	1	0
23	0x0000	0xFFFF[7]	0xFF00	Update	2	1	1
24	0x0000	0xXXXX	0xFF00	Update	2	1	1
25	0x0000	0xFFXX	0xFF(00)	Update	2	1	1
26	0x0000	0xXX00	0xFFFF	Update-Erased	3	1	0
27	0x0000	0xXXXX	0xFFFF	Update-Erased	3	1	0
28	0x0000	0xFFFF	0xXX00	Update-Erased	3	1	1
29	0x0000	0xFFFF	0xXXXX	Update-Erased	3	1	1

[1] 0xXXXX, 0xFFXX, 0xXX00 indicates a non-valid value was read; 0xFF(00) indicates more 0's than other indicator (if same number of 0's, then swap system defaults to block 0 active)

[2] Cases 10-29 due to brownout (abort) detected during program or erase steps related to swap

[3] Must read 0xFFFF with erase verify level before transition to Complete allowed

[4] No reset since successful Swap Complete execution

[5] Reset after successful Swap Complete execution

[6] Not a valid case

[7] Fails to read 0xFFFF at erase verify level

35.4.11.14.1 Swap State Determination

During the reset sequence, the state of the swap system is determined by evaluating the IFR Swap Field in the program flash swap IFR and the swap indicators located in each of the program flash blocks at the swap indicator address stored in the IFR Swap Field.

Table 35-42. Program Flash Swap IFR Fields

Address Range	Size (Bytes)	Field Description
0x00 – 0x01	2	Swap Indicator Address
0x02 – 0x03	2	Swap Enable Word
0x04 – 0xFF	252	Reserved

35.4.11.15 Program Partition Command

The Program Partition command prepares the FlexNVM block for use as data flash, EEPROM backup, or a combination of both and initializes the FlexRAM. The Program Partition command must not be launched from flash memory, since flash memory resources are not accessible during Program Partition command execution.

CAUTION

> While different partitions of the FlexNVM are available, the intention is that a single partition choice is used throughout the entire lifetime of a given application. The FlexNVM Partition Code choices affect the endurance and data retention characteristics of the device.

Table 35-43. Program Partition Command FCCOB Requirements

FCCOB Number	FCCOB Contents [7:0]
0	0x80 (PGMPART)
1	Not Used
2	Not Used
3	Not Used
4	EEPROM Data Set Size Code[1]
5	FlexNVM Partition Code[2]

1. See Table 2 and EEPROM Data Set Size
1. See Table 2 and EEPROM Data Set Size
1. See Table 2 and EEPROM Data Set Size
2. See Table 3
2. See Table 3

Table 35-44. Valid EEPROM Data Set Size Codes

EEPROM Data Set Size Code (FCCOB4)[1]		EEPROM Data Set Size (Bytes)
FCCOB4[EEESPLIT]	FCCOB4[EEESIZE]	Subsystem A + B
11	0xF	0[2]
00	0x9	4 + 28
01	0x9	8 + 24
10	0x9	16 + 16
11	0x9	16 + 16
00	0x8	8 + 56
01	0x8	16 + 48
10	0x8	32 + 32
11	0x8	32 + 32
00	0x7	16 + 112
01	0x7	32 + 96
10	0x7	64 + 64
11	0x7	64 + 64
00	0x6	32 + 224
01	0x6	64 + 192
10	0x6	128 + 128
11	0x6	128 + 128
00	0x5	64 + 448
01	0x5	128 + 384
10	0x5	256 + 256
11	0x5	256 + 256
00	0x4	128 + 896
01	0x4	256 + 768
10	0x4	512 + 512
11	0x4	512 + 512
00	0x3	256 + 1,792
01	0x3	512 + 1,536
10	0x3	1,024 + 1,024
11	0x3	1,024 + 1,024
00	0x2	512 + 3,584
01	0x2	1,024 + 3,072
10	0x2	2,048 + 2,048
11	0x2	2,048 + 2,048

1. FCCOB4[7:6] = 00
2. EEPROM Data Set Size must be set to 0 bytes when the FlexNVM Partition Code is set for no EEPROM.

Table 35-45. Valid FlexNVM Partition Codes

FlexNVM Partition Code (FCCOB5[DEPART])[1]	Data flash Size (KB)	EEPROM backup Size (KB)
0000	64	0
0011	32	32
0100	0	64
1000	0	64
1011	32	32
1100	64	0

[1] FCCOB5[7:4] = 0000

After clearing CCIF to launch the Program Partition command, the flash memory module first verifies that the EEPROM Data Set Size Code and FlexNVM Partition Code in the data flash IFR are erased. If erased, the Program Partition command erases the contents of the FlexNVM memory. If the FlexNVM is to be partitioned for EEPROM backup, the allocated EEPROM backup sectors are formatted for EEPROM use. Finally, the partition codes are programmed into the data flash IFR using the values provided. The Program Partition command also verifies that the partition codes read back correctly after programming. If the FlexNVM is partitioned for EEPROM backup, the EEERDY flag will set with RAMRDY clear. If the FlexNVM is not partitioned for EEPROM backup, the RAMRDY flag will set with EEERDY clear. The CCIF flag is set after the Program Partition operation completes.

Prior to launching the Program Partition command, the data flash IFR must be in an erased state, which can be accomplished by executing the Erase All Blocks command or by an external request (see Erase All Blocks Command). The EEPROM Data Set Size Code and FlexNVM Partition Code are read using the Read Resource command (see Read Resource Command).

Table 35-46. Program Partition Command Error Handling

Error Condition	Error Bit
Command not available in current mode/security	FSTAT[ACCERR]
The EEPROM data size and FlexNVM partition code bytes are not initially 0xFFFF	FSTAT[ACCERR]
Invalid EEPROM Data Set Size Code is entered (see Table 35-44 for valid codes)	FSTAT[ACCERR]
Invalid FlexNVM Partition Code is entered (see Table 35-45 for valid codes)	FSTAT[ACCERR]
FlexNVM Partition Code = full data flash (no EEPROM) and EEPROM Data Set Size Code allocates FlexRAM for EEPROM	FSTAT[ACCERR]
FlexNVM Partition Code allocates space for EEPROM backup, but EEPROM Data Set Size Code allocates no FlexRAM for EEPROM	FSTAT[ACCERR]
FCCOB4[7:6] != 00	FSTAT[ACCERR]
FCCOB5[7:4] != 0000	FSTAT[ACCERR]
Any errors have been encountered during the verify operation	FSTAT[MGSTAT0]

35.4.11.16 Set FlexRAM Function Command

The Set FlexRAM Function command changes the function of the FlexRAM:

- When not partitioned for EEPROM, the FlexRAM is typically used as traditional RAM.
- When partitioned for EEPROM, the FlexRAM is typically used to store EEPROM data.

Table 35-47. Set FlexRAM Function Command FCCOB Requirements

FCCOB Number	FCCOB Contents [7:0]
0	0x81 (SETRAM)
1	FlexRAM Function Control Code (see Table 35-48)

Table 35-48. FlexRAM Function Control

FlexRAM Function Control Code	Action
0xFF	Make FlexRAM available as RAM: • Clear the FCNFG[EEERDY] and FCNFG[RAMRDY] flags • Write a background of ones to all FlexRAM locations • Set the FCNFG[RAMRDY] flag
0x00	Make FlexRAM available for EEPROM: • Clear the FCNFG[EEERDY] and FCNFG[RAMRDY] flags • Write a background of ones to all FlexRAM locations • Copy-down existing EEPROM data to FlexRAM • Set the FCNFG[EEERDY] flag

After clearing CCIF to launch the Set FlexRAM Function command, the flash memory module sets the function of the FlexRAM based on the FlexRAM Function Control Code.

When making the FlexRAM available as traditional RAM, the flash memory module clears the FCNFG[EEERDY] and FCNFG[RAMRDY] flags, overwrites the contents of the entire FlexRAM with a background pattern of all ones, and sets the FCNFG[RAMRDY] flag. The state of the FEPROT register does not prevent the FlexRAM from being overwritten. When the FlexRAM is set to function as a RAM, normal read and write accesses to the FlexRAM are available. When large sections of flash memory need to be programmed, e.g. during factory programming, the FlexRAM can be used as the Section Program Buffer for the Program Section command (see Program Section Command).

When making the FlexRAM available for EEPROM, the flash memory module clears the FCNFG[EEERDY] and FCNFG[RAMRDY] flags, overwrites the contents of the FlexRAM allocated for EEPROM with a background pattern of all ones, and copies the existing EEPROM data from the EEPROM backup record space to the FlexRAM. After completion of the EEPROM copy-down, the FCNFG[EEERDY] flag is set. When the FlexRAM is set to function as EEPROM, normal read and write access to the FlexRAM is available, but writes to the FlexRAM also invoke EEPROM activity. The CCIF flag is set after the Set FlexRAM Function operation completes.

Table 35-49. Set FlexRAM Function Command Error Handling

Error Condition	Error Bit
Command not available in current mode/security	FSTAT[ACCERR]
FlexRAM Function Control Code is not defined	FSTAT[ACCERR]
FlexRAM Function Control Code is set to make the FlexRAM available for EEPROM, but FlexNVM is not partitioned for EEPROM	FSTAT[ACCERR]

35.4.12 Security

The flash memory module provides security information to the MCU based on contents of the FSEC security register.

The MCU then limits access to flash memory resources as defined in the device's Chip Configuration details. During reset, the flash memory module initializes the FSEC register using data read from the security byte of the Flash Configuration Field (see Flash Configuration Field Description).

The following fields are available in the FSEC register. The settings are described in the Flash Security Register (FTFL_FSEC) details.

Flash security features are discussed further in AN4507: Using the Kinetis Security and Flash Protection Features . Note that not all features described in the application note are available on this device.

Table 35-50. FSEC register fields

FSEC field	Description
KEYEN	Backdoor Key Access
MEEN	Mass Erase Capability
FSLACC	Factory Security Level Access
SEC	MCU security

35.4.12.1 Changing the Security State

The security state out of reset can be permanently changed by programming the security byte of the flash configuration field. This assumes that you are starting from a mode where the necessary program flash erase and program commands are available and that the region of the program flash containing the flash configuration field is unprotected. If the flash security byte is successfully programmed, its new value takes affect after the next chip reset.

35.4.12.1.1 Unsecuring the Chip Using Backdoor Key Access

The chip can be unsecured by using the backdoor key access feature, which requires knowledge of the contents of the 8-byte backdoor key value stored in the Flash Configuration Field (see Flash Configuration Field Description). If the FSEC[KEYEN] bits are in the enabled state, the Verify Backdoor Access Key command (see Verify Backdoor Access Key Command) can be run; it allows the user to present prospective keys for comparison to the stored keys. If the keys match, the FSEC[SEC] bits are changed to unsecure the chip. The entire 8-byte key cannot be all 0s or all 1s; that is, 0000_0000_0000_0000h and FFFF_FFFF_FFFF_FFFFh are not accepted by the Verify Backdoor Access Key command as valid comparison values. While the Verify Backdoor Access Key command is active, program flash memory is not available for read access and returns invalid data.

The user code stored in the program flash memory must have a method of receiving the backdoor keys from an external stimulus. This external stimulus would typically be through one of the on-chip serial ports.

If the KEYEN bits are in the enabled state, the chip can be unsecured by the following backdoor key access sequence:

1. Follow the command sequence for the Verify Backdoor Access Key command as explained in Verify Backdoor Access Key Command
2. If the Verify Backdoor Access Key command is successful, the chip is unsecured and the FSEC[SEC] bits are forced to the unsecure state

An illegal key provided to the Verify Backdoor Access Key command prohibits further use of the Verify Backdoor Access Key command. A reset of the chip is the only method to re-enable the Verify Backdoor Access Key command when a comparison fails.

After the backdoor keys have been correctly matched, the chip is unsecured by changing the FSEC[SEC] bits. A successful execution of the Verify Backdoor Access Key command changes the security in the FSEC register only. It does not alter the security byte or the keys stored in the Flash Configuration Field (Flash Configuration Field Description). After the next reset of the chip, the security state of the flash memory module reverts back to the flash security byte in the Flash Configuration Field. The Verify Backdoor Access Key command sequence has no effect on the program and erase protections defined in the program flash protection registers.

If the backdoor keys successfully match, the unsecured chip has full control of the contents of the Flash Configuration Field. The chip may erase the sector containing the Flash Configuration Field and reprogram the flash security byte to the unsecure state and change the backdoor keys to any desired value.

35.4.13 Reset Sequence

On each system reset the flash memory module executes a sequence which establishes initial values for the flash block configuration parameters, FPROT, FDPROT, FEPROT, FOPT, and FSEC registers and the FCNFG[SWAP, PFLSH, RAMRDY, EEERDY] bits.

FSTAT[CCIF] is cleared throughout the reset sequence. The flash memory module holds off CPU access during the reset sequence. Flash reads are possible when the hold is removed. Completion of the reset sequence is marked by setting CCIF which enables flash user commands.

If a reset occurs while any flash command is in progress, that command is immediately aborted. The state of the word being programmed or the sector/block being erased is not guaranteed. Commands and operations do not automatically resume after exiting reset.

Chapter 36
MCU: EzPort

36.1 Overview

NOTE

For the chip-specific implementation details of this module's instances, see the chip configuration information.

The EzPort module is a serial flash programming interface that enables In-System Programming (ISP) of flash memory contents in a 32-bit general-purpose microcontroller. Memory contents can be read/erased/programmed from an external source, in a format that is compatible with many standalone flash memory chips, without requiring the removal of the microcontroller from the system board.

36.1.1 Block diagram

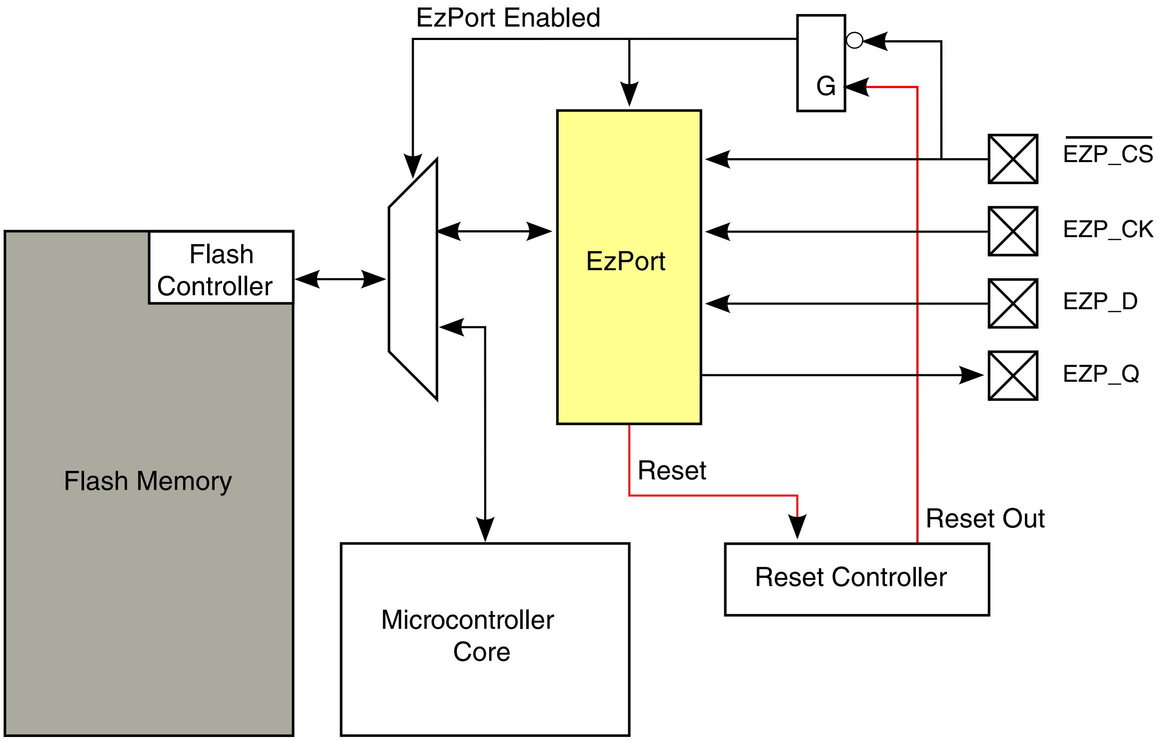


Figure 36-1. EzPort block diagram

36.1.2 Features

EzPort includes the following features:

- Serial interface that is compatible with a subset of the SPI format.
- Ability to read, erase, and program flash memory.
- Ability to reset the microcontroller, allowing it to boot from the flash memory after the memory has been configured.

36.1.3 Modes of operation

The EzPort can operate in one of two modes, enabled or disabled.

- Enabled — When enabled, the EzPort steals access to the flash memory, preventing access from other cores or peripherals. The rest of the microcontroller is disabled to avoid conflicts. The flash is configured for NVM Special mode.
- Disabled — When the EzPort is disabled, the rest of the microcontroller can access flash memory as normal.

The EzPort provides a simple interface to connect an external device to the flash memory on board a 32 bit microcontroller. The interface itself is compatible with the SPI interface, with the EzPort operating as a slave, running in either of the two following modes. The data is transmitted with the most significant bit first.

- CPOL = 0, CPHA = 0
- CPOL = 1, CPHA = 1

Commands are issued by the external device to erase, program, or read the contents of the flash memory. The serial data out from the EzPort is tri-stated unless data is being driven. This allows the signal to be shared among several different EzPort (or compatible) devices in parallel, as long as they have different chip-selects.

36.2 External signal descriptions

After the table of EzPort external signals, subsequent sections explain the signals in more detail.

Table 36-1. EzPort external signals

JTAG (cJTAG) Signal	External Signal	Name	I/O
TCK (TCKC)	EZP_CK	EzPort Clock	Input
TMS (TMSC)	$\overline{\text{EZP\_CS}}$	EzPort Chip Select	Input
TDI (TDIC)	EZP_D	EzPort Serial Data In	Input
TDO (TDOC)	EZP_Q	EzPort Serial Data Out	Output

36.2.1 EzPort Clock (EZP_CK)

EZP_CK is the serial clock for data transfers. The serial data in (EZP_D) and chip select ($\overline{\text{EZP\_CS}}$) are registered on the rising edge of EZP_CK, while serial data out (EZP_Q) is driven on the falling edge of EZP_CK.

The maximum frequency of the EzPort clock is half the system clock frequency for all commands, except when executing the Read Data or Read FlexRAM commands. When executing the Read Data or Read FlexRAM commands, the EzPort clock has a maximum frequency of 1/8 the system clock frequency.

36.2.2 EzPort Chip Select ($\overline{\text{EZP\_CS}}$)

$\overline{\text{EZP\_CS}}$ is the chip select for signaling the start and end of serial transfers. While $\overline{\text{EZP\_CS}}$ is asserted, if the microcontroller's reset out signal is negated, then EzPort is enabled out of reset; otherwise EzPort is disabled. After EzPort is enabled, asserting $\overline{\text{EZP\_CS}}$ starts a serial data transfer, which continues until $\overline{\text{EZP\_CS}}$ is negated again. The negation of $\overline{\text{EZP\_CS}}$ indicates that the current command has finished and resets the EzPort state machine, so that EzPort is ready to receive the next command.

36.2.3 EzPort Serial Data In (EZP_D)

EZP_D is the serial data in for data transfers. EZP_D is registered on the rising edge of EZP_CK. All commands, addresses, and data are shifted in most significant bit first. When the EzPort is driving output data on EZP_Q, the data shifted in EZP_D is ignored.

36.2.4 EzPort Serial Data Out (EZP_Q)

EZP_Q is the serial data out for data transfers. EZP_Q is driven on the falling edge of EZP_CK. It is tri-stated unless $\overline{\text{EZP\_CS}}$ is asserted and the EzPort is driving data out. All data is shifted out most significant bit first.

36.3 Command definition

The EzPort receives commands from an external device and translates the commands into flash memory accesses. The following table lists the supported commands.

Table 36-2. EzPort commands

Command	Description	Code	Address Bytes	Data Bytes	Accepted when secure?
WREN	Write Enable	0x06	0	0	Yes
WRDI	Write Disable	0x04	0	0	Yes
RDSR	Read Status Register	0x05	0	1	Yes

Table continues on the next page...

Table 36-2. EzPort commands (continued)

Command	Description	Code	Address Bytes	Data Bytes	Accepted when secure?
READ	Flash Read Data	0x03	3	1+	No
FAST_READ	Flash Read Data at High Speed	0x0B	3[1]	1+	No
SP	Flash Section Program	0x02	3	48 - SECTION[4]	No
SE	Flash Sector Erase	0xD8	3[3]	0	No
BE	Flash Bulk Erase	0xC7	0	0	Yes[5]
RESET	Reset Chip	0xB9	0	0	Yes
WRFCCOB	Write FCCOB Registers	0xBA	0	12	Yes[6]
FAST_RDFCCOB	Read FCCOB registers at high speed	0xBB	0	1 - 12[2]	No
WRFLEXRAM	Write FlexRAM	0xBC	3[1]	4	No
RDFLEXRAM	Read FlexRAM	0xBD	3[1]	1+	No
FAST_RDFLEXRAM	Read FlexRAM at high speed	0xBE	3[1]	1+[2]	No

1. Address must be 32-bit aligned (two LSBs must be zero).
2. One byte of dummy data must be shifted in before valid data is shifted out.
3. Address must be 32-bit aligned (two LSBs must be zero).64-bit aligned (three LSBs must be zero).
4. Please see the Flash Memory chapter for a definition of section size. Total number of data bytes programmed must be a multiple of 48.
5. Bulk Erase is accepted when security is set and only when the BEDIS status field is not set .
6. The flash will be in NVM Special mode, restricting the type of commands that can be executed through WRITE_FCCOB when security is enabled.

36.3.1 Command descriptions

This section describes the module commands.

36.3.1.1 Write Enable

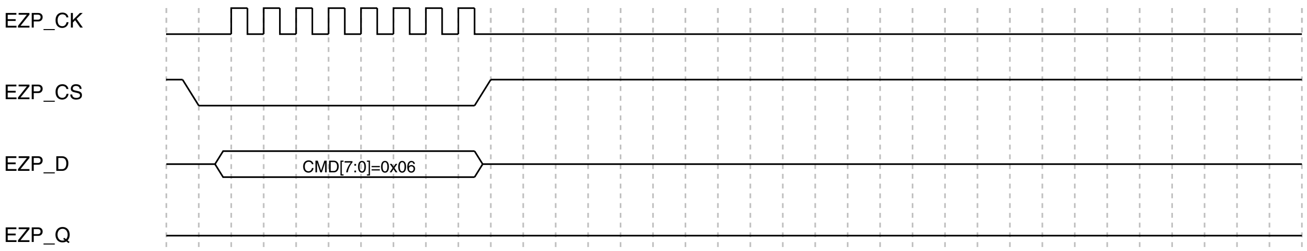


Figure 36-2. Write Enable command sequence

The Write Enable (WREN) command sets the write enable register field in the EzPort status register. The write enable field must be set for a write command (SP, SE, BE, WRFCCOB, or WRFLEXRAM) to be accepted. The write enable register field clears on reset, on a Write Disable command, and at the completion of write command. This command must not be used if a write is already in progress.

36.3.1.2 Write Disable

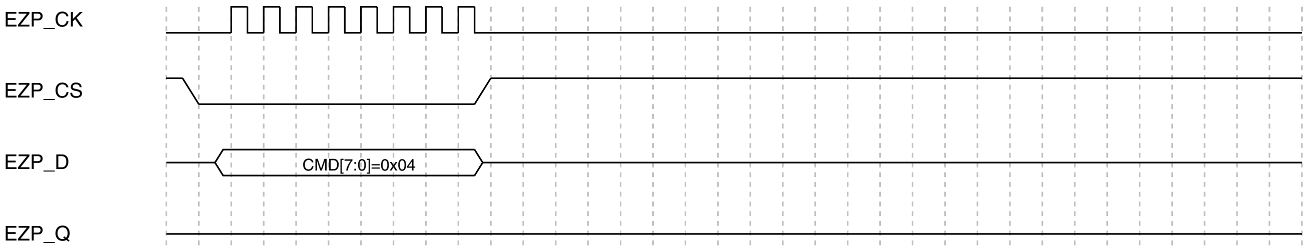


Figure 36-3. Write Disable command sequence

The Write Disable (WRDI) command clears the write enable register field in the status register. This command must not be used if a write is already in progress.

36.3.1.3 Read Status Register

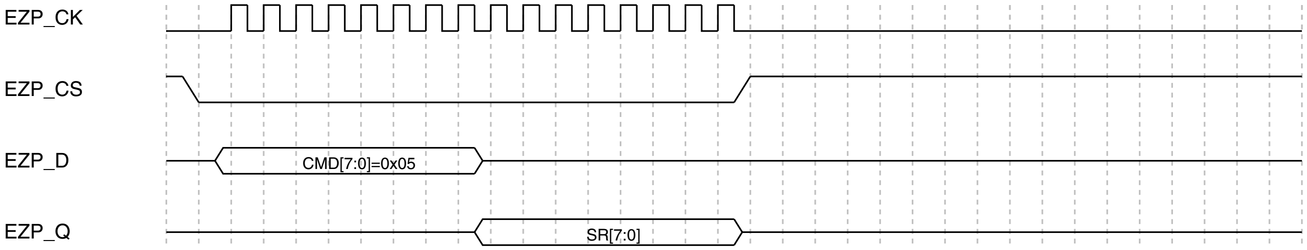


Figure 36-4. Read Status Register command sequence

The Read Status Register (RDSR) command returns the contents of the EzPort status register.

Table 36-3. EzPort status register

	7	6	5	4	3	2	1	0
R	FS	WEF			FLEXRAM	BEDIS	WEN	WIP

Table continues on the next page...

Table 36-3. EzPort status register (continued)

W								
Reset:	0/1[1]	0	0	0	0/1[2]	0/1[3]	0	1[4]

1. Reset value reflects the status of flash security out of reset.
2. Reset value reflects FlexNVM flash partitioning. If FlexNVM flash has been paritioned for EEPROM, this field is set immediately after reset. Note that FLEXRAM is cleared after the EzPort initialization sequence completes, as indicated by clearing of WIP.
3. Reset value reflects whether bulk erase is enabled or disabled out of reset.
4. Initial value of WIP is 1, but the value clears to 0 after EzPort initialization is complete.

Table 36-4. EzPort status register field description

Field	Description
0 WIP	Write in progress. Sets after a write command (SP, SE, BE, WRFCCOB, or WRFLEXRAM) is accepted and clears after the flash memory has completed all operations associated with the write command, as indicated by the Command Complete Interrupt Flag (CCIF) inside the flash. This field is also asserted on reset and cleared when EzPort initialization is complete. Only the Read Status Register (RDSR) command is accepted while a write is in progress. 0 = Write is not in progress. Accept any command. 1 = Write is in progress. Only accept RDSR command.
1 WEN	Write enable Enables the write comman that follows. It is a control field that must be set before a write command (SP, SE, BE, WRFCCOB, or WRFLEXRAM) is accepted. Is set by the Write Enable (WREN) command and cleared by reset or a Write Disable (WRDI) command. This field also clears when the flash memory has completed all operations associated with the command. 0 = Disables the following write command. 1 = Enables the following write command.
2 BEDIS	Bulk erase disable Indicates whether bulk erase (BE) is disabled when flash is secure. 0 = BE is enabled. 1 = BE is disabled if FS is also set. Attempts to issue a BE command will result in the WEF flag being set.
3 FLEXRAM	For devices with FlexRAM: FlexRAM mode Indicates the current mode of the FlexRAM. Valid only when WIP is cleared. 0 = FlexRAM is in RAM mode. RD/WRFLEXRAM command can be used to read/write data in FlexRAM. 1 = FlexRAM is in EEPROM mode. SP command is not accepted. RD/WRFLEXRAM command can be used to read/write data in the FlexRAM.
6 WEF	Write error flag Indicates whether there has been an error while executing a write command (SP, SE, BE, WRFCCOB, or WRFLEXRAM). The WEF flag will set if Flash Access Error Flag (ACCERR), Flash Protection Violation (FPVIOL), or Memory Controller Command Completion Status (MGSTAT0) inside the flash memory is set at the completion of the write command. See the flash memory chapter for further description of these flags and their sources. The WEF flag clears after a Read Status Register (RDSR) command. 0 = No error on previous write command. 1 = Error on previous write command.
7	Flash security

Table 36-4. EzPort status register field description

Field	Description
FS	Indicates whether the flash is secure. See Table 36-2 for the list of commands that will be accepted when flash is secure. Flash security can be disabled by performing a BE command. 0 = Flash is not secure. 1 = Flash is secure.

36.3.1.4 Read Data

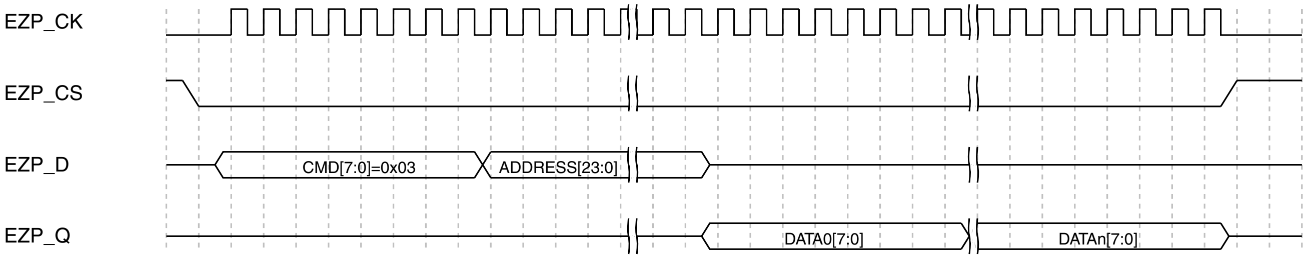


Figure 36-5. Read command sequence

The Read Data (READ) command returns data from the flash memory or FlexNVM, depending on the initial address specified in the command word. The initial address must be 32-bit aligned with the two LSBs being zero.

Data continues being returned for as long as the EzPort chip select ($\overline{\text{EZP\_CS}}$) is asserted, with the address automatically incrementing. In this way, the entire contents of flash can be returned by one command. Attempts to read from an address which does not fall within the valid address range for the flash memory regions returns unknown data. See Flash memory map for EzPort access.

For this command to return the correct data, the EzPort clock (EZP_CK) must run at the internal system clock divided by eight or slower. This command is not accepted if the WEF, WIP, or FS field in the EzPort status register is set.

36.3.1.5 Read Data at High Speed

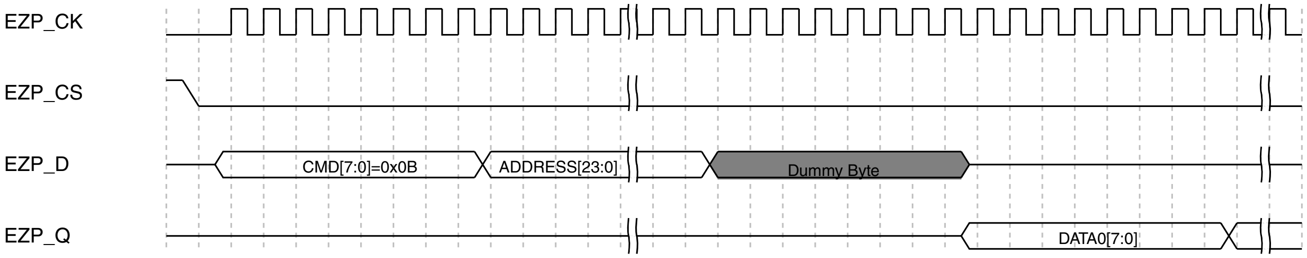


Figure 36-6. Read Data at High Speed command sequence

The Read Data at High Speed (FAST_READ) command is identical to the READ command, except for the inclusion of a dummy byte following the address bytes and before the first data byte is returned.

This command can be run with an EzPort clock (EZP_CK) frequency of half the internal system clock frequency of the microcontroller or slower. This command is not accepted if the WEF, WIP, or FS field in the EzPort status register is set.

36.3.1.6 Section Program

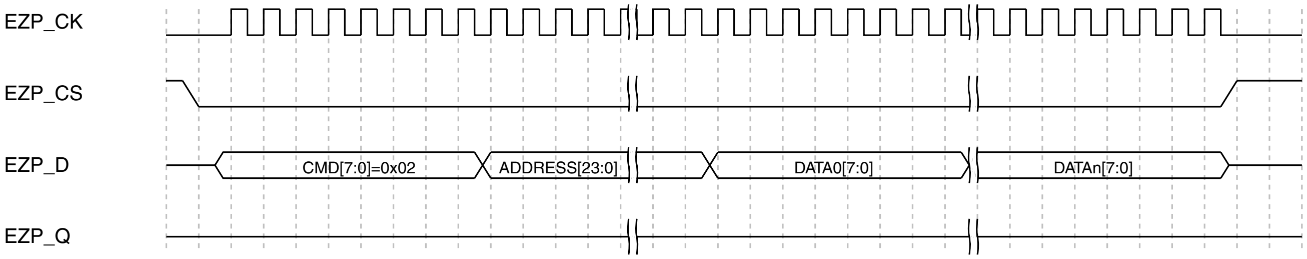


Figure 36-7. Section Program command sequence

The Section Program (SP) command programs up to one section of flash memory that has previously been erased. Please see the Flash Memory chapter for a definition of section size. The starting address of the memory to program is sent after the command word and must be a 32-bit aligned address with the two LSBs being zero.64-bit aligned address with the three LSBs being zero.

As data is shifted in, the EzPort buffers the data in FlexRAMSystem RAM before executing an SP command within the flashsequentially moving the data into flash using the Program Longword command. For this reason, the number of bytes to be programmed must be a multiple of 48 and up to one flash section can be programmed at a time. For more details, see the Flash Memory Module.

Attempts to program more than one section, across a sector boundary or from an initial address which does not fall within the valid address range for the flash causes the WEF flag to set. See Flash memory map for EzPort access.

For devices with FlexRAM: This command requires the FlexRAM to be configured for traditional RAM operation. By default, after entering EzPort mode, the FlexRAM is configured for traditional RAM operation. If the user reconfigures FlexRAM for EEPROM operation, then the user should use the WRFCCOB command to configure FlexRAM back to traditional RAM operation before issuing an SP command. See the Flash Memory chapter for details on how the FlexRAM function is modified.

This command is not accepted if the WEF, WIP, FLEXRAM, or FS field is set or if the WEN field is not set in the EzPort status register.

36.3.1.7 Sector Erase

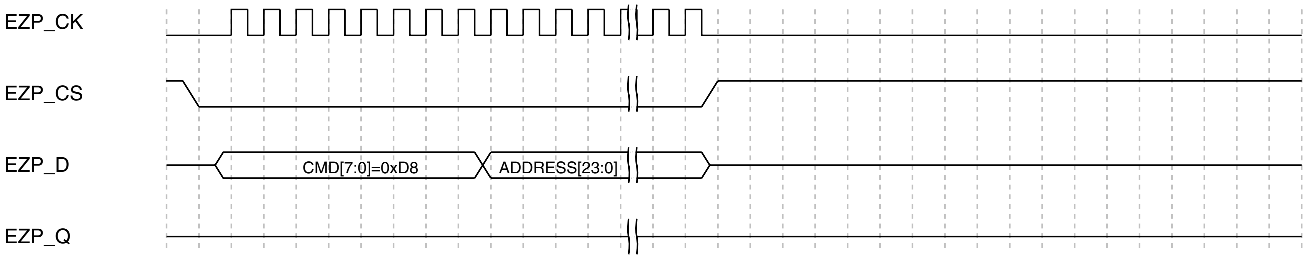


Figure 36-8. Sector Erase command sequence

The Sector Erase (SE) command erases the contents of one sector of flash memory. The three byte address sent after the command byte can be any address within the sector to erase, but must be a 64-bit aligned address (the three LSBs must be zero). Attempts to erase from an initial address which does not fall within the valid address range (see Flash memory map for EzPort access) for the flash results in the WEF flag being set.

This command is not accepted if the WEF, WIP or FS field is set or if the WEN field is not set in the EzPort status register.

36.3.1.8 Bulk Erase

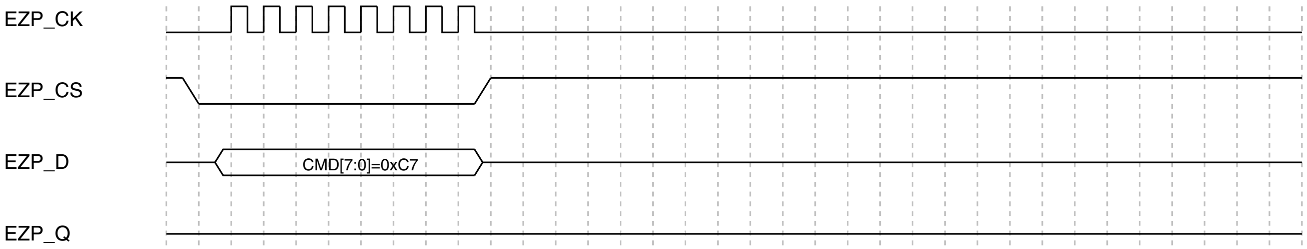


Figure 36-9. Bulk Erase command sequence

The Bulk Erase (BE) command erases the entire contents of flash memory, ignoring any protected sectors or flash security. Flash security is disabled upon successful completion of the BE command.

Attempts to issue a BE command while the BEDIS and FS fields are set results in the WEF flag being set in the EzPort status register. Also, this command is not accepted if the WEF or WIP field is set or if the WEN field is not set in the EzPort status register.

36.3.1.9 EzPort Reset Chip

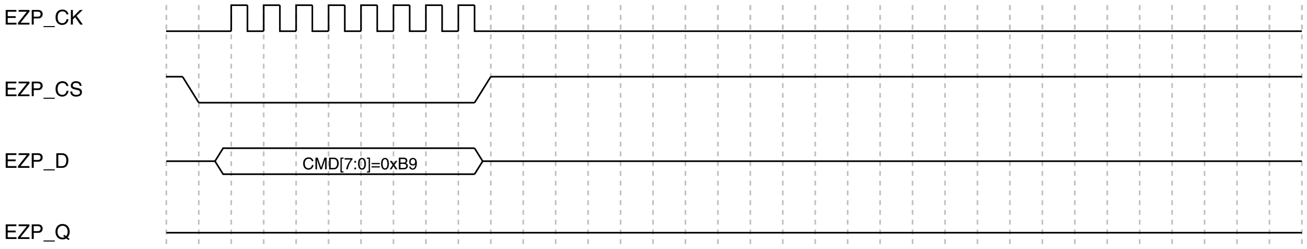


Figure 36-10. Reset Chip command sequence

The Reset Chip (RESET) command forces the chip into the reset state. If the EzPort chip select ($\overline{\text{EZP\_CS}}$) pin is asserted at the end of the reset period, EzPort is enabled; otherwise, it is disabled. This command allows the chip to boot up from flash memory after being programmed by an external source.

This command is not accepted if the WIP field is set in the EzPort status register.

36.3.1.10 Write FCCOB Registers

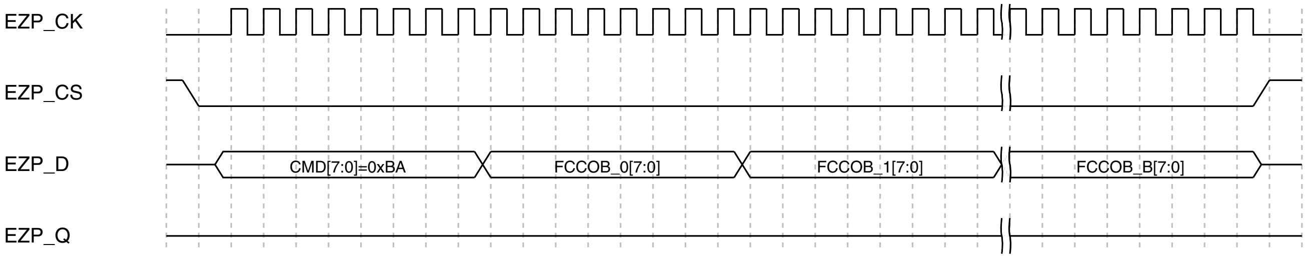


Figure 36-11. Write FCCOB Registers command sequence

The Write FCCOB Registers (WRFCCOB) command allows the user to write to the flash common command object registers and execute any command allowed by the flash.

NOTE

> When security is enabled, the flash is configured in NVM Special mode, restricting the commands that can be executed by the flash.

After receiving 12 bytes of data, EzPort writes the data to the FCCOB 0-B registers in the flash and then automatically launches the command within the flash. If greater or less than 12 bytes of data is received, this command has unexpected results and may result in the WEF flag being set.

This command is not accepted if the WEF or WIP field is set or if the WEN field is not set in the EzPort status register.

36.3.1.11 Read FCCOB Registers at High Speed

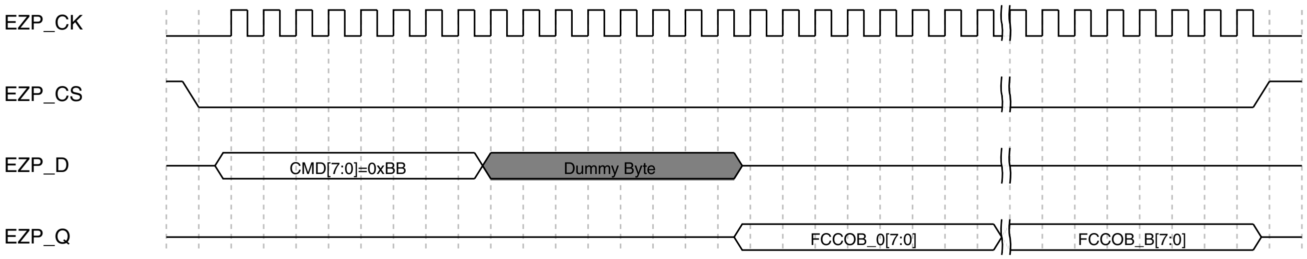


Figure 36-12. Read FCCOB Registers at High Speed command sequence

The Read FCCOB Registers at High Speed (FAST_RDFCCOB) command allows the user to read the contents of the flash common command object registers. After receiving the command, EzPort waits for one dummy byte of data before returning FCCOB register data starting at FCCOB 0 and ending with FCCOB B.

This command can be run with an EzPort clock (EZP_CK) frequency half the internal system clock frequency of the microcontroller or slower. Attempts to read greater than 12 bytes of data returns unknown data. This command is not accepted if the WEF, WIP, or FS fields in the EzPort status register are 1.

36.3.1.12 Write FlexRAM

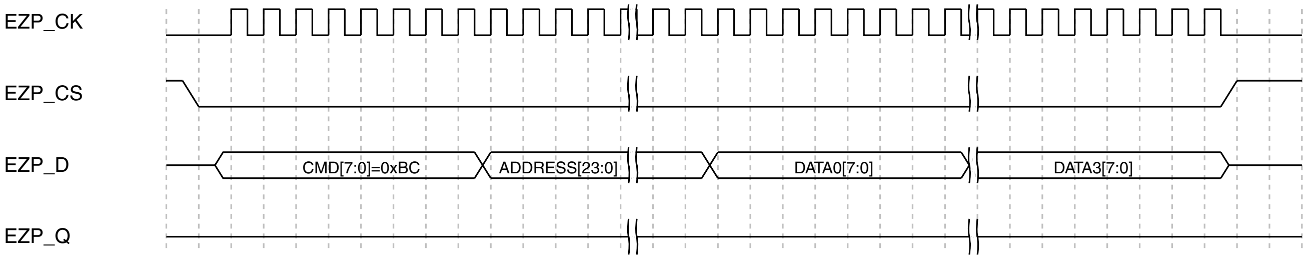


Figure 36-13. Write FlexRAM command sequence

This command is only applicable for devices with FlexRAM.

The Write FlexRAM (WRFLEXRAM) command allows the user to write four bytes of data to the FlexRAM. If the FlexRAM is configured for EEPROM operation, the WRFLEXRAM command can effectively be used to create data records in the EEPROM flash memory.

By default, after entering EzPort mode, the FlexRAM is configured for traditional RAM operation and functions as direct RAM. The user can alter the FlexRAM configuration by using WRFCCOB to execute a Set FlexRAM or Program Partition command within the flash.

The address of the FlexRAM location to be written is sent after the command word and must be a 32-bit aligned address (the two LSBs must be zero). Attempts to write an address which does not fall within the valid address range for the FlexRAM results in the value of the WEF flag being 1. See Flash memory map for EzPort access for more information.

After receiving four bytes of data, EzPort writes the data to the FlexRAM. If greater or less than four bytes of data is received, this command has unexpected results and may result in the value of the WEF flag being 1.

This command is not accepted if the WEF, WIP or FS fields are 1 or if the WEN field is 0 in the EzPort status register.

36.3.1.13 Read FlexRAM

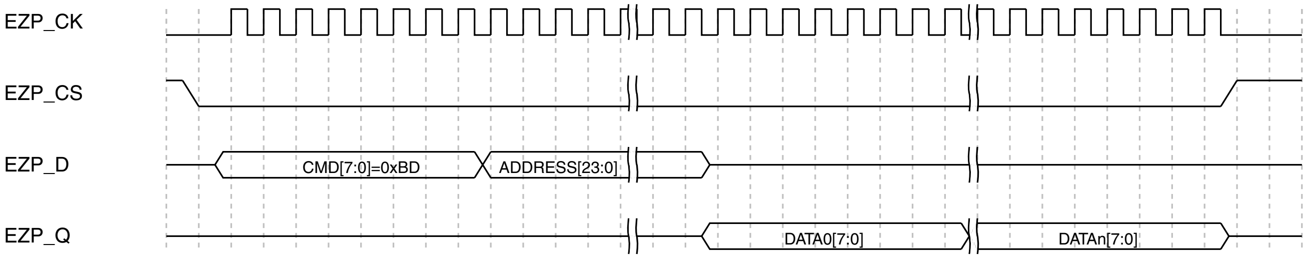


Figure 36-14. Read FlexRAM command sequence

This command is only applicable for devices with FlexRAM.

The Read FlexRAM (RDFLEXRAM) command returns data from the FlexRAM. If the FlexRAM is configured for EEPROM operation, the RDFLEXRAM command can effectively be used to read data from EEPROM flash memory.

Data continues being returned for as long as the EzPort chip select (EZP_CS) is asserted, with the address automatically incrementing. In this way, the entire contents of FlexRAM can be returned by one command.

The initial address must be 32-bit aligned (the two LSBs must be zero). Attempts to read from an address which does not fall within the valid address range for the FlexRAM returns unknown data. See Flash memory map for EzPort access for more information.

For this command to return the correct data, the EzPort clock (EZP_CK) must run at the internal system clock divided by eight or slower. This command is not accepted if the WEF, WIP, or FS fields in the EzPort status register are set.

36.3.1.14 Read FlexRAM at High Speed

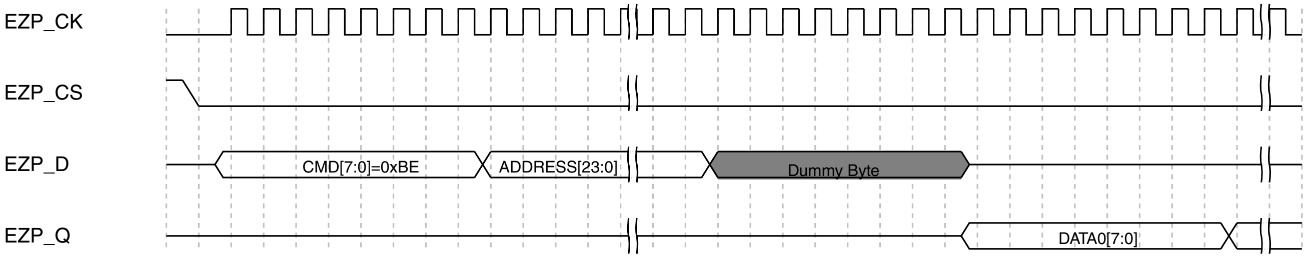


Figure 36-15. Read FlexRAM at High Speed command sequence

This command is only applicable for devices with FlexRAM.

The Read FlexRAM at High Speed (FAST_RDFLEXRAM) command is identical to the RDFLEXRAM command, except for the inclusion of a dummy byte following the address bytes and before the first data byte is returned.

This command can be run with an EzPort clock (EZP_CK) frequency up to and including half the internal system clock frequency of the microcontroller. This command is not accepted if the WEF, WIP, or FS fields in the EzPort status register are set.

36.4 Flash memory map for EzPort access

The following table shows the flash memory map for access through EzPort.

NOTE

The flash block address map for access through EzPort may not conform to the system memory map. Changes are made to allow the EzPort address width to remain 24 bits.

Table 36-5. Flash Memory Map for EzPort Access

Valid start address	Size	Flash block	Valid commands
0x0000_0000	See device's chip configuration details	Flash	READ, FAST_READ, SP, SE, BE
0x0000_0000	See device's chip configuration details	FlexRAM	RDFLEXRAM, FAST_RDFLEXRAM, WRFLEXRAM, BE

Chapter 37
MCU: Cyclic Redundancy Check (CRC)

37.1 Introduction

NOTE

> For the chip-specific implementation details of this module's instances, see the chip configuration information.

The cyclic redundancy check (CRC) module generates 16/32-bit CRC code for error detection.

The CRC module provides a programmable polynomial and other parameters required to implement a 16-bit or 32-bit CRC standard.

The 16/32-bit code is calculated for 32 bits of data at a time.

37.1.1 Features

Features of the CRC module include:

- Hardware CRC generator circuit using a 16-bit or 32-bit programmable shift register
- Programmable initial seed value and polynomial
- Option to transpose input data or output data (the CRC result) bitwise or bytewise. This option is required for certain CRC standards. A bytewise transpose operation is not possible when accessing the CRC data register via 8-bit accesses. In this case, the user's software must perform the bytewise transpose function.
- Option for inversion of final CRC result
- 32-bit CPU register programming interface

37.1.2 Block diagram

The following is a block diagram of the CRC.

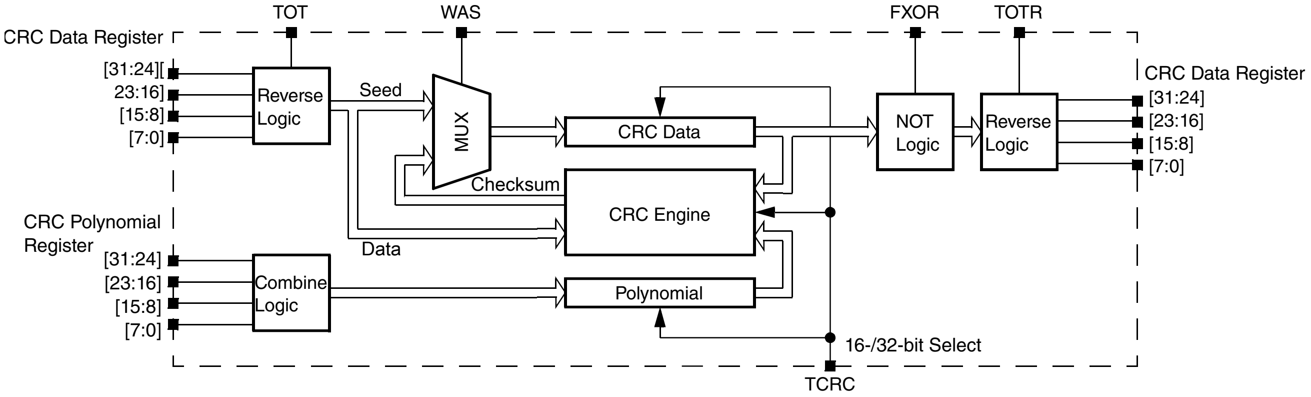


Figure 37-1. Programmable cyclic redundancy check (CRC) block diagram

37.1.3 Modes of operation

Various MCU modes affect the CRC module's functionality.

37.1.3.1 Run mode

This is the basic mode of operation.

37.1.3.2 Low-power modes (Wait or Stop)

Any CRC calculation in progress stops when the MCU enters a low-power mode that disables the module clock. It resumes after the clock is enabled or via the system reset for exiting the low-power mode. Clock gating for this module is dependent on the MCU.

37.2 Memory map and register descriptions

CRC memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4003_2000	CRC Data register (CRC_DATA)	32	R/W	FFFF_FFFFh	37.2.1/821
4003_2004	CRC Polynomial register (CRC_GPOLY)	32	R/W	0000_1021h	37.2.2/822
4003_2008	CRC Control register (CRC_CTRL)	32	R/W	0000_0000h	37.2.3/822

37.2.1 CRC Data register (CRC_DATA)

The CRC Data register contains the value of the seed, data, and checksum. When CTRL[WAS] is set, any write to the data register is regarded as the seed value. When CTRL[WAS] is cleared, any write to the data register is regarded as data for general CRC computation.

In 16-bit CRC mode, the HU and HL fields are not used for programming the seed value, and reads of these fields return an indeterminate value. In 32-bit CRC mode, all fields are used for programming the seed value.

When programming data values, the values can be written 8 bits, 16 bits, or 32 bits at a time, provided all bytes are contiguous; with MSB of data value written first.

After all data values are written, the CRC result can be read from this data register. In 16-bit CRC mode, the CRC result is available in the LU and LL fields. In 32-bit CRC mode, all fields contain the result. Reads of this register at any time return the intermediate CRC value, provided the CRC module is configured.

Address: 4003_2000h base + 0h offset = 4003_2000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	HU								HL								LU								LL							
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

CRC_DATA field descriptions

Field	Description
31–24 HU	CRC High Upper Byte In 16-bit CRC mode (CTRL[TCRC] is 0), this field is not used for programming a seed value. In 32-bit CRC mode (CTRL[TCRC] is 1), values written to this field are part of the seed value when CTRL[WAS] is 1. When CTRL[WAS] is 0, data written to this field is used for CRC checksum generation in both 16-bit and 32-bit CRC modes.
23–16 HL	CRC High Lower Byte In 16-bit CRC mode (CTRL[TCRC] is 0), this field is not used for programming a seed value. In 32-bit CRC mode (CTRL[TCRC] is 1), values written to this field are part of the seed value when CTRL[WAS] is 1. When CTRL[WAS] is 0, data written to this field is used for CRC checksum generation in both 16-bit and 32-bit CRC modes.
15–8 LU	CRC Low Upper Byte When CTRL[WAS] is 1, values written to this field are part of the seed value. When CTRL[WAS] is 0, data written to this field is used for CRC checksum generation.
LL	CRC Low Lower Byte When CTRL[WAS] is 1, values written to this field are part of the seed value. When CTRL[WAS] is 0, data written to this field is used for CRC checksum generation.

37.2.2 CRC Polynomial register (CRC_GPOLY)

This register contains the value of the polynomial for the CRC calculation. The HIGH field contains the upper 16 bits of the CRC polynomial, which are used only in 32-bit CRC mode. Writes to the HIGH field are ignored in 16-bit CRC mode. The LOW field contains the lower 16 bits of the CRC polynomial, which are used in both 16- and 32-bit CRC modes.

Address: 4003_2000h base + 4h offset = 4003_2004h

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

CRC_GPOLY field descriptions

Field	Description
31–16 HIGH	High Polynominal Half-word Writable and readable in 32-bit CRC mode (CTRL[TCRC] is 1). This field is not writable in 16-bit CRC mode (CTRL[TCRC] is 0).
LOW	Low Polynominal Half-word Writable and readable in both 32-bit and 16-bit CRC modes.

37.2.3 CRC Control register (CRC_CTRL)

This register controls the configuration and working of the CRC module. Appropriate bits must be set before starting a new CRC calculation. A new CRC calculation is initialized by asserting CTRL[WAS] and then writing the seed into the CRC data register.

Address: 4003_2000h base + 8h offset = 4003_2008h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	TOT		TOTR		0	FXOR	WAS	TCRC	0							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CRC_CTRL field descriptions

Field	Description
31–30 TOT	Type Of Transpose For Writes Defines the transpose configuration of the data written to the CRC data register. See the description of the transpose feature for the available transpose options. 00 No transposition. 01 Bits in bytes are transposed; bytes are not transposed. 10 Both bits in bytes and bytes are transposed. 11 Only bytes are transposed; no bits in a byte are transposed.
29–28 TOTR	Type Of Transpose For Read Identifies the transpose configuration of the value read from the CRC Data register. See the description of the transpose feature for the available transpose options. 00 No transposition. 01 Bits in bytes are transposed; bytes are not transposed. 10 Both bits in bytes and bytes are transposed. 11 Only bytes are transposed; no bits in a byte are transposed.
27 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
26 FXOR	Complement Read Of CRC Data Register Some CRC protocols require the final checksum to be XORed with 0xFFFFFFFF or 0xFFFF. Asserting this bit enables on the fly complementing of read data. 0 No XOR on reading. 1 Invert or complement the read value of the CRC Data register.
25 WAS	Write CRC Data Register As Seed When asserted, a value written to the CRC data register is considered a seed value. When deasserted, a value written to the CRC data register is taken as data for CRC computation. 0 Writes to the CRC data register are data values. 1 Writes to the CRC data register are seed values.
24 TCRC	Width of CRC protocol. 0 16-bit CRC protocol. 1 32-bit CRC protocol.
Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

37.3 Functional description

37.3.1 CRC initialization/reinitialization

To enable the CRC calculation, the user must program CRC_CTRL[WAS], CRC_GPOLY,necessary parameters for transposition and CRC result inversion in the applicable registers. Asserting CRC_CTRL[WAS] enables the programming of the seed value into the CRC_DATA register.

After a completed CRC calculation, the module can be reinitialized for a new CRC computation by reasserting CRC_CTRL[WAS] and programming a new, or previously used, seed value. All other parameters must be set before programming the seed value and subsequent data values.

37.3.2 CRC calculations

In 16-bit and 32-bit CRC modes, data values can be programmed 8 bits, 16 bits, or 32 bits at a time, provided all bytes are contiguous. Noncontiguous bytes can lead to an incorrect CRC computation.

37.3.2.1 16-bit CRC

To compute a 16-bit CRC:

1. Clear CRC_CTRL[TCRC] to enable 16-bit CRC mode.
2. Program the transpose and complement options in the CTRL register as required for the CRC calculation. See Transpose feature and CRC result complement for details.
3. Write a 16-bit polynomial to the CRC_GPOLY[LOW] field. The CRC_GPOLY[HIGH] field is not usable in 16-bit CRC mode.
4. Set CRC_CTRL[WAS] to program the seed value.
5. Write a 16-bit seed to CRC_DATA[LU:LL]. CRC_DATA[HU:HL] are not used.
6. Clear CRC_CTRL[WAS] to start writing data values.
7. Write data values into CRC_DATA[HU:HL:LU:LL]. A CRC is computed on every data value write, and the intermediate CRC result is stored back into CRC_DATA[LU:LL].
8. When all values have been written, read the final CRC result from CRC_DATA[LU:LL].

Transpose and complement operations are performed on the fly while reading or writing values. See Transpose feature and CRC result complement for details.

37.3.2.2 32-bit CRC

To compute a 32-bit CRC:

1. Set CRC_CTRL[TCRC] to enable 32-bit CRC mode.
2. Program the transpose and complement options in the CTRL register as required for the CRC calculation. See Transpose feature and CRC result complement for details.
3. Write a 32-bit polynomial to CRC_GPOLY[HIGH:LOW].
4. Set CRC_CTRL[WAS] to program the seed value.
5. Write a 32-bit seed to CRC_DATA[HU:HL:LU:LL].
6. Clear CRC_CTRL[WAS] to start writing data values.
7. Write data values into CRC_DATA[HU:HL:LU:LL]. A CRC is computed on every data value write, and the intermediate CRC result is stored back into CRC_DATA[HU:HL:LU:LL].
8. When all values have been written, read the final CRC result from CRC_DATA[HU:HL:LU:LL]. The CRC is calculated bytewise, and two clocks are required to complete one CRC calculation.

Transpose and complement operations are performed on the fly while reading or writing values. See Transpose feature and CRC result complement for details.

37.3.3 Transpose feature

By default, the transpose feature is not enabled. However, some CRC standards require the input data and/or the final checksum to be transposed. The user software has the option to configure each transpose operation separately, as desired by the CRC standard. The data is transposed on the fly while being read or written.

Some protocols use little endian format for the data stream to calculate a CRC. In this case, the transpose feature usefully flips the bits. This transpose option is one of the types supported by the CRC module.

37.3.3.1 Types of transpose

The CRC module provides several types of transpose functions to flip the bits and/or bytes, for both writing input data and reading the CRC result, separately using the CTRL[TOT] or CTRL[TOTR] fields, according to the CRC calculation being used.

The following types of transpose functions are available for writing to and reading from the CRC data register:

1. CTRL[TOT] or CTRL[TOTR] is 00.

No transposition occurs.

2. CTRL[TOT] or CTRL[TOTR] is 01

   Bits in a byte are transposed, while bytes are not transposed.

   reg[31:0] becomes {reg[24:31], reg[16:23], reg[8:15], reg[0:7]}

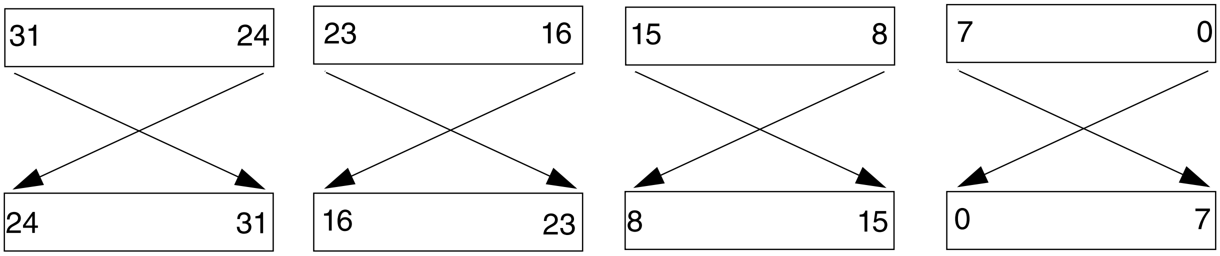


Figure 37-2. Transpose type 01

3. CTRL[TOT] or CTRL[TOTR] is 10.

   Both bits in bytes and bytes are transposed.

   reg[31:0] becomes = {reg[0:7], reg[8:15],reg[16:23], reg[24:31]}

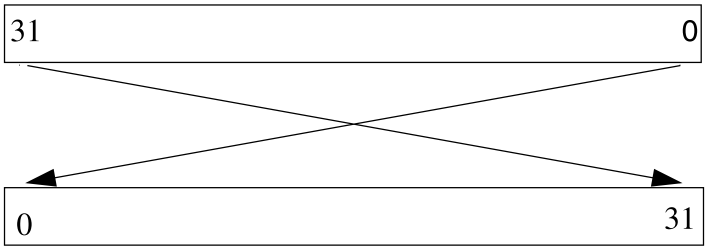


Figure 37-3. Transpose type 10

4. CTRL[TOT] or CTRL[TOTR] is 11.

   Bytes are transposed, but bits are not transposed.

   reg[31:0] becomes {reg[7:0], reg[15:8], reg[23:16], reg[31:24]}

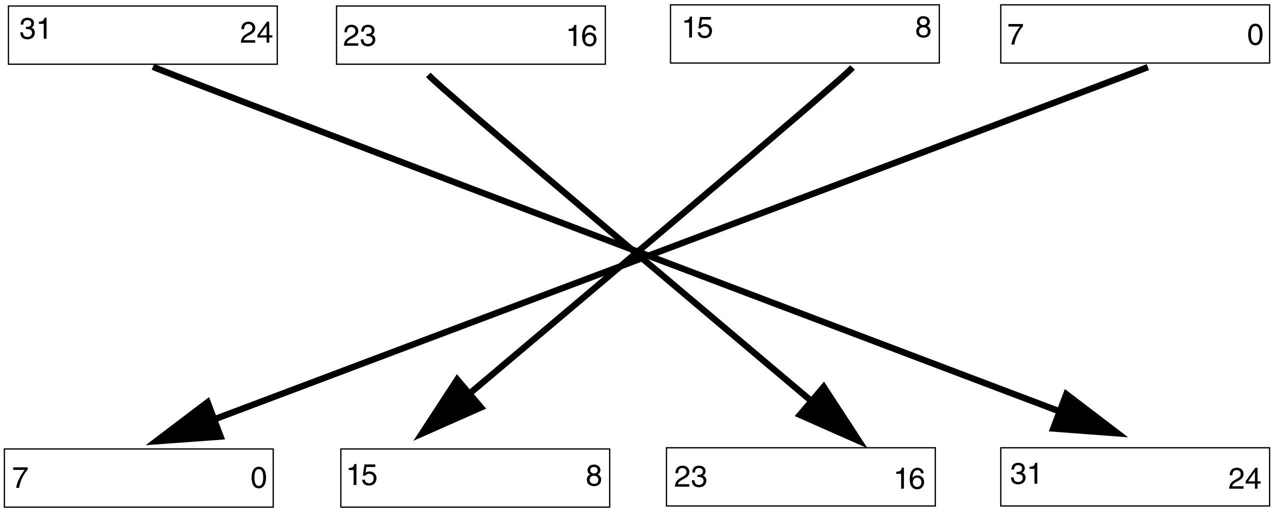


Figure 37-4. Transpose type 11

NOTE

- For 8-bit and 16-bit write accesses to the CRC data register, the data is transposed with zeros on the unused byte or bytes (taking 32 bits as a whole), but the CRC is calculated on the valid byte(s) only.
- When reading the CRC data register for a 16-bit CRC result and using transpose options 10 and 11, the resulting value after transposition resides in the CRC[HU:HL] fields. The user software must account for this situation when reading the 16-bit CRC result, so reading 32 bits is preferred.

37.3.4 CRC result complement

When CTRL[FXOR] is set, the checksum is complemented. The CRC result complement function outputs the complement of the checksum value stored in the CRC data register every time the CRC data register is read. When CTRL[FXOR] is cleared, reading the CRC data register accesses the raw checksum value.

Chapter 38
MCU: Memory-Mapped Cryptographic Acceleration Unit (MMCAU)

38.1 Introduction

NOTE

> For the chip-specific implementation details of this module's instances, see the chip configuration information.

The memory-mapped Cryptographic Acceleration Unit (CAU) is a coprocessor that is connected to the processor's Private Peripheral Bus (PPB). It supports the hardware implementation of a set of specialized operations to improve the throughput of software-based security encryption/decryption operations and message digest functions.

The CAU supports acceleration of the DES, 3DES, AES, MD5, SHA-1, and SHA-256 algorithms. NXP provides an optimized C-function library that provides the appropriate software building blocks to implement higher-level security functions.

38.2 CAU Block Diagram

A simplified block diagram is given below that illustrates the CAU and a table to show its parts.

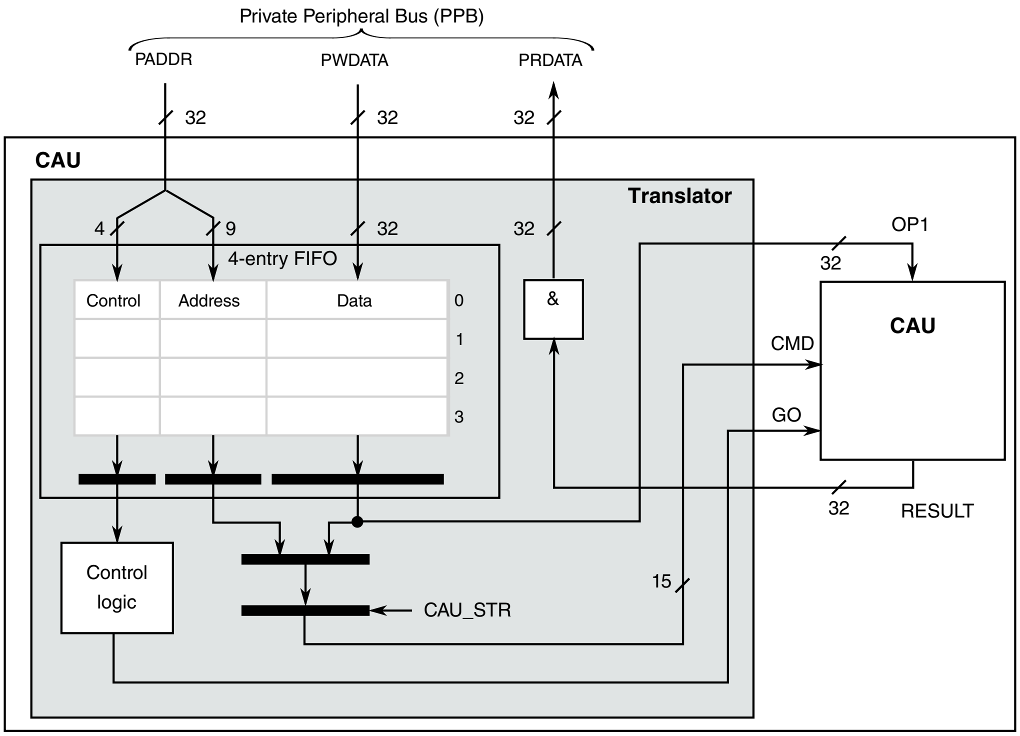


Figure 38-1. CAU block diagram

Table 38-1. CAU parts table

Item	Description
Translator submodule	Provides the bridge between the PPB interface and the CAU module. Passes memory-mapped commands and data on the PPB to/from the CAU
4-entry FIFO	Contains commands and input operands and the associated control captured from the PPB and sent to the CAU
CAU	3-terminal block with a command and optional input operand and a result bus. More details in following figure.

The following figure shows the CAU block in more detail.

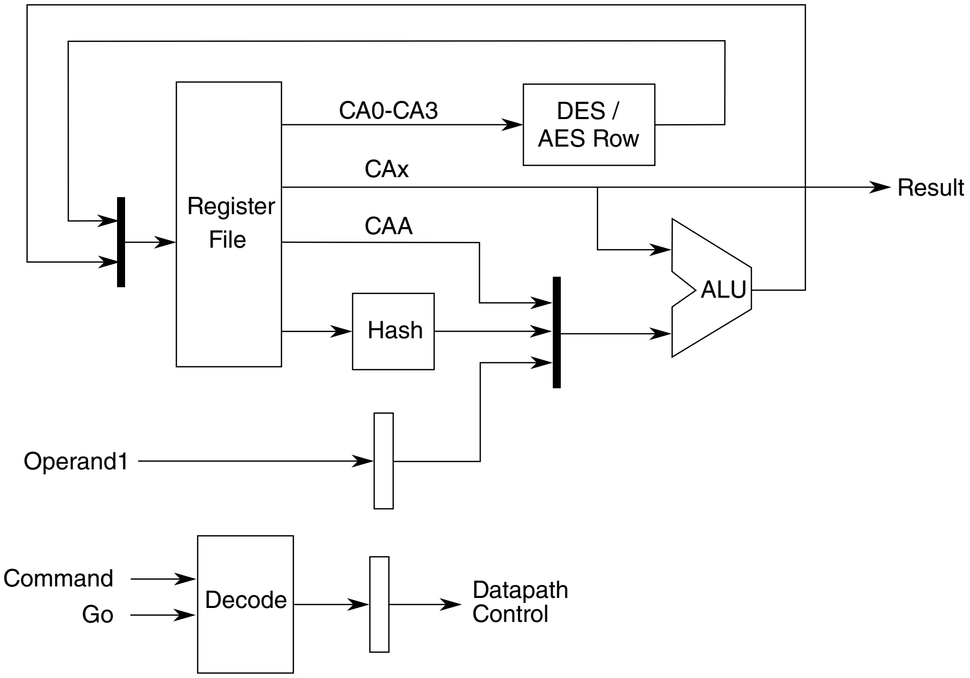


Figure 38-2. Top-level CAU block diagram

38.3 Overview

As the name suggests, the CAU provides a mechanism for memory-mapped register reads and writes to be transformed into specific commands and operands sent to the CAU coprocessor.

The CAU translator module performs the following functions:

- All the required functions affecting the transmission of commands to the CAU module.
- If needed, stalling the PPB transactions based on the state of the 4-entry command/ data FIFO.
- Some basic integrity checks on PPB operations.

The set of implemented algorithms provides excellent support for network security standards, such as SSL and IPsec. Additionally, using the CAU efficiently permits the implementation of any higher level functions or modes of operation, such as HMAC, CBC, and so on based on the supported algorithms.

The cryptographic algorithms are implemented partially in software with only functions critical to increasing performance implemented in hardware. The CAU allows for efficient, fine-grained partitioning of functions between hardware and software:

- Implement the innermost security kernel functions using the coprocessor instructions.
- Implement higher level functions in software by using the standard processor instructions.

This partitioning of functions is key to minimizing size of the CAU while maintaining a high level of throughput. Using software for some functions also simplifies the CAU design. The CAU implements a set of coprocessor commands that operate on a register file of 32-bit registers.

38.4 Features

The CAU includes the following distinctive features:

- Supports DES, 3DES, AES, MD5, SHA-1, and SHA-256 algorithms
- Simple, flexible programming model
- Ability to send up to three commands in one data write operation

38.5 Memory map/Register definition

The CAU contains multiple registers used by each of the supported algorithms. The following table shows registers that are applicable to each supported algorithm and indicates the corresponding letter designations for each algorithm.

For more information on these letter designations, see the supported algorithm specifications.

Code	Register	DES	AES	MD5	SHA-1	SHA-256
0	CAU Status Register (CASR)	—	—	—	—	—
1	CAU Accumulator (CAA)	—	—	a	T	T
2	General-Purpose Register 0 (CA0)	C	W0	—	A	A
3	General-Purpose Register 1 (CA1)	D	W1	b	B	B

Table continues on the next page...

Code	Register	DES	AES	MD5	SHA-1	SHA-256
4	General-Purpose Register 2 (CA2)	L	W2	c	C	C
5	General-Purpose Register 3 (CA3)	R	W3	d	D	D
6	General-Purpose Register 4 (CA4)	—	—	—	E	E
7	General-Purpose Register 5 (CA5)	—	—	—	W	F
8	General-Purpose Register 6 (CA6)	—	—	—	—	G
9	General-Purpose Register 7 (CA7)	—	—	—	—	H
10	General-Purpose Register 8 (CA8)	—	—	—	—	W/T_1

The CAU supports only 32-bit operations and register accesses. All registers support read, write, and ALU operations. However, only bits 1–0 of the CASR are writeable. Bits 31–2 of the CASR must be written as 0 for compatibility with future versions of the CAU.

The codes listed in this section are used in the memory-mapped commands. For more details on this, see CAU programming model.

NOTE

In the following table, the "address" or "offset" refers to the command code value for the CAU registers.

CAU memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
E008_1000	Status Register (CAU_CASR)	32	R/W	2000_0000h	38.5.1/834
E008_1001	Accumulator (CAU_CAA)	32	R/W	0000_0000h	38.5.2/835

Table continues on the next page...

CAU memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
E008_1002	General Purpose Register (CAU_CA0)	32	R/W	0000_0000h	38.5.3/835
E008_1003	General Purpose Register (CAU_CA1)	32	R/W	0000_0000h	38.5.3/835
E008_1004	General Purpose Register (CAU_CA2)	32	R/W	0000_0000h	38.5.3/835
E008_1005	General Purpose Register (CAU_CA3)	32	R/W	0000_0000h	38.5.3/835
E008_1006	General Purpose Register (CAU_CA4)	32	R/W	0000_0000h	38.5.3/835
E008_1007	General Purpose Register (CAU_CA5)	32	R/W	0000_0000h	38.5.3/835
E008_1008	General Purpose Register (CAU_CA6)	32	R/W	0000_0000h	38.5.3/835
E008_1009	General Purpose Register (CAU_CA7)	32	R/W	0000_0000h	38.5.3/835
E008_100A	General Purpose Register (CAU_CA8)	32	R/W	0000_0000h	38.5.3/835

38.5.1 Status Register (CAU_CASR)

CASR contains the status and configuration for the CAU.

Address: E008_1000h base + 0h offset = E008_1000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	VER				0											
W																
Reset	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0														DPE	IC
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CAU_CASR field descriptions

Field	Description
31–28 VER	CAU Version Indicates CAU version. 0x1 Initial CAU version. 0x2 Second version, added support for SHA-256 algorithm (This is the value on this device).
27–2 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
1 DPE	DES Parity Error Indicates whether the DES parity error is detected. 0 No error detected. 1 DES key parity error detected.

Table continues on the next page...

CAU_CASR field descriptions (continued)

Field	Description
0 IC	Illegal Command Indicates an illegal instruction has been executed. 0 No illegal commands issued. 1 Illegal command issued.

38.5.2 Accumulator (CAU_CAA)

Commands use the CAU accumulator for storage of results and as an operand for the cryptographic algorithms.

Address: E008_1000h base + 1h offset = E008_1001h

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

CAU_CAA field descriptions

Field	Description
ACC	Accumulator Stores results of various CAU commands.

38.5.3 General Purpose Register (CAU_CA*n*)

The General Purpose Register is used in the CAU commands for storage of results and as operands for various cryptographic algorithms.

Address: E008_1000h base + 2h offset + (1d × i), where i=0d to 8d

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

CAU_CA*n* field descriptions

Field	Description
CAn	General Purpose Registers Used by the CAU commands. Some cryptographic operations work with specific registers.

38.6 Functional description

This section discusses the programming model and operation of the CAU.

38.6.1 CAU programming model

The 4-entry FIFO is indirectly mapped into a 4-KB address space associated with the CAU located on this device. This address space is effectively split into two equal regions:

- one used to directly write commands for CAU load operations
- the other used to send commands and input operands for CAU loads

Data writes on the PPB are loaded into this FIFO and automatically converted into CAU load operands by the CAU translator. Data reads on the PPB are converted into CAU store register operations where the result is returned to the processor as the read data value.

The CAU requires a 15-bit command, and optionally, a 32-bit input operand, for each CAU load, PPB write. The 15-bit command includes the 9-bit opcode and other bits statically formed by the CAU translator logic controlling the CAU.

The following figure shows the 4-KB address space and the mapping of the CAU commands in this space.

NOTE

- Although the indirect store/load portion of the address space in the figure below shows only the indirect load/store commands, direct load commands can also be used in this space. However, it is more efficient to use the direct load portion of the address space.
- Accesses to the reserved space in the direct load space are terminated with an error, while accesses to the reserved space in the indirect load/store space are detected as an illegal CAU command. See CAU integrity checks for details.

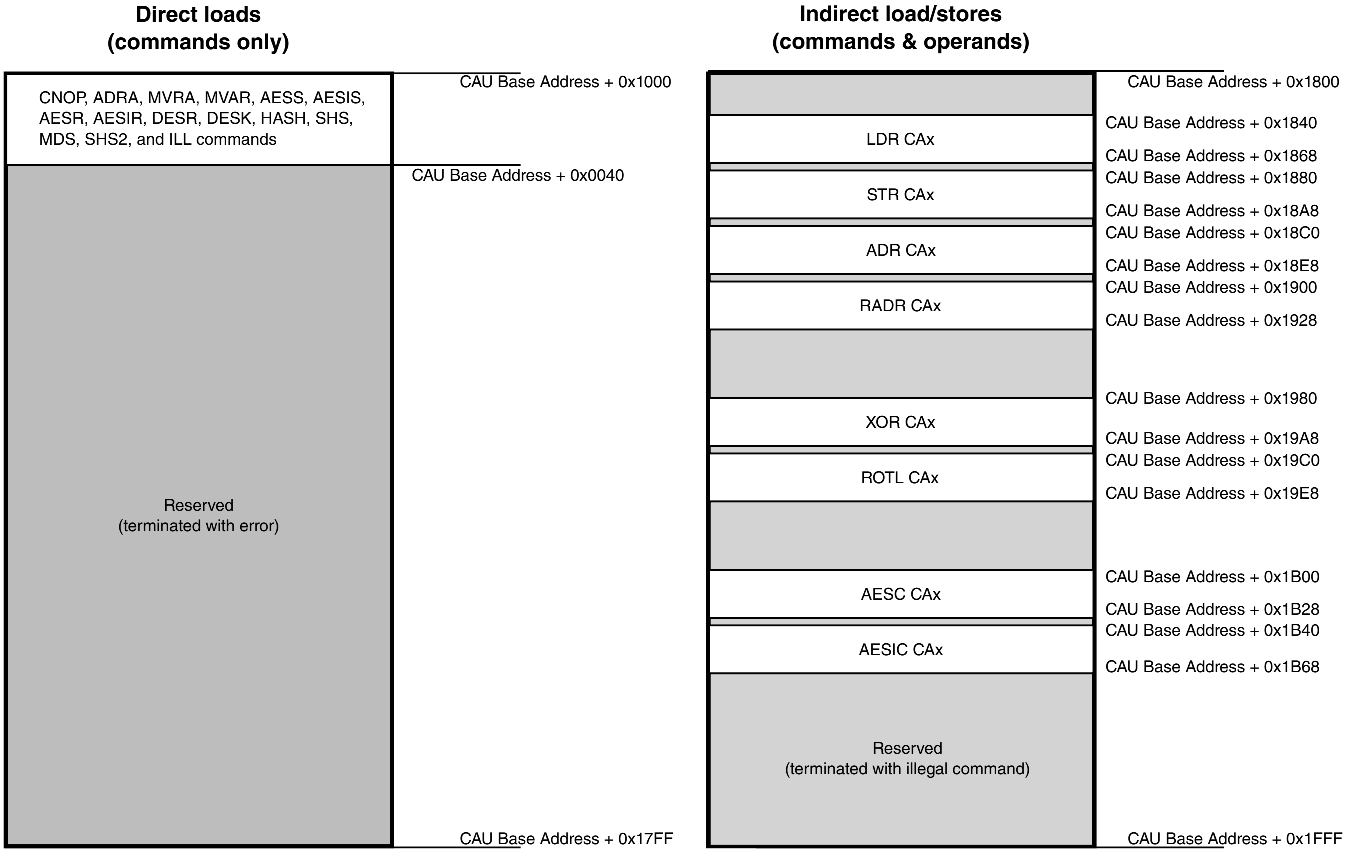

Direct loads
(commands only)
Indirect load/stores
(commands & operands)
CNOP, ADRA, MVRA, MVAR, AESS, AESIS, AESR, AESIR, DESR, DESK, HASH, SHS, MDS, SHS2, and ILL commands
CAU Base Address + 0x1000
CAU Base Address + 0x0040
Reserved
(terminated with error)
CAU Base Address + 0x17FF
CAU Base Address + 0x1800
CAU Base Address + 0x1840
LDR CAx
CAU Base Address + 0x1868
CAU Base Address + 0x1880
STR CAx
CAU Base Address + 0x18A8
CAU Base Address + 0x18C0
ADR CAx
CAU Base Address + 0x18E8
CAU Base Address + 0x1900
RADR CAx
CAU Base Address + 0x1928
CAU Base Address + 0x1980
XOR CAx
CAU Base Address + 0x19A8
CAU Base Address + 0x19C0
ROTL CAx
CAU Base Address + 0x19E8
CAU Base Address + 0x1B00
AESC CAx
CAU Base Address + 0x1B28
CAU Base Address + 0x1B40
AESIC CAx
CAU Base Address + 0x1B68
Reserved
(terminated with illegal command)
CAU Base Address + 0x1FFF

Figure 38-3. CAU memory map

38.6.1.1 Direct loads

The CAU supports writing multiple commands in each 32-bit direct write operation. Each 9-bit opcode also includes a valid bit. Therefore, one, two, or three commands can be transmitted in a single 32-bit PPB write. The following figure illustrates the accepted formats for the 32-bit CAU write data value:

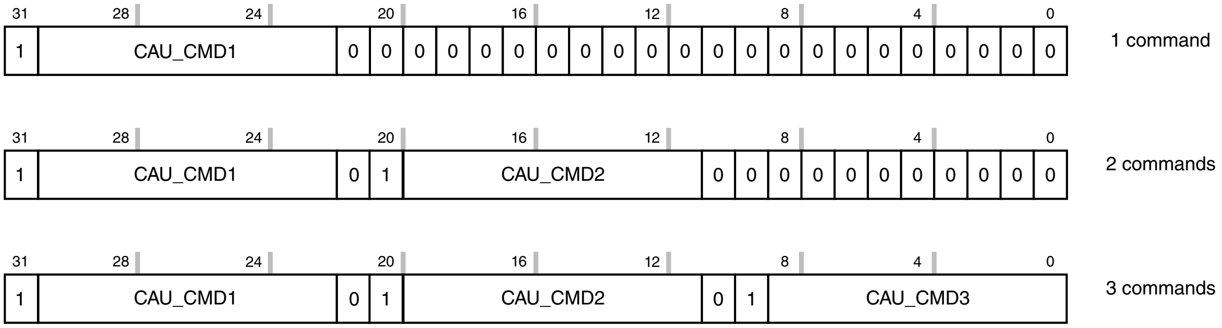

31 28 24 20 16 12 8 4 0
1 | CAU_CMD1 | 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1 command
31 28 24 20 16 12 8 4 0
1 | CAU_CMD1 | 0 | 1 | CAU_CMD2 | 0 0 0 0 0 0 0 0 0 0 0
2 commands
31 28 24 20 16 12 8 4 0
1 | CAU_CMD1 | 0 | 1 | CAU_CMD2 | 0 | 1 | CAU_CMD3
3 commands

Figure 38-4. Direct loads

38.6.1.2 Indirect loads

For CAU load operations requiring a 32-bit input operand, the address contains the 9-bit opcode to be passed to the CAU while the data is the 32-bit operand. Specifically, the CAU address and data for these indirect writes is shown in the figure below.

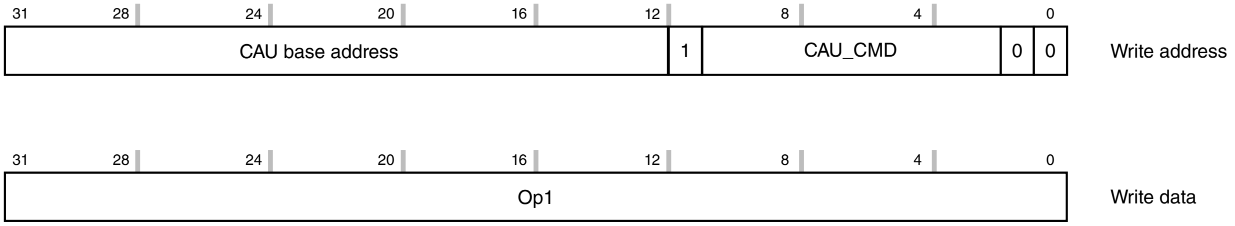


Figure 38-5. Indirect loads

38.6.1.3 Indirect stores

For CAU store operations, a PPB read is performed with the appropriate CAU store register opcode embedded in the address. This appears as another indirect command. The detail of Indirect stores is shown in the figure below.

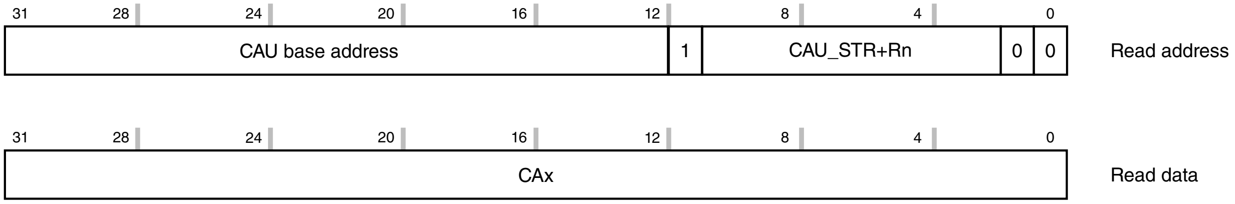


Figure 38-6. Indirect store

38.6.2 CAU integrity checks

If an illegal operation or access is attempted, the PPB bus cycle is terminated with an error response and the operation is aborted and not sent to the CAU.

The CAU performs a series of address and data integrity checks as described in the following sections. The results of these checks are logically summed together and, if appropriate, a PPB error termination is generated.

38.6.2.1 Address integrity checks

The CAU address checking includes the following. See Figure 38-3 for the CAU memory map details.

- Any CAU reference using a non-0-modulo-4 byte address (addr[1:0] ≠ 00) generates an error termination.
- For CAU writes:
  - Only the first 64 bytes of the 2-KB direct write address space can be referenced. Attempting to access regions beyond the first 64 bytes terminates with an error.
  - The second 2-KB space defines the indirect address-as-command region and any reference in this space is allowed by the CAU.

NOTE

The CAU contains error logic to detect any illegal command sent to it. Accordingly, there are address values in this upper 2-KB region of the address space that are passed to the CAU, and then detected as illegal commands. If the CAU detects an illegal command, it sets the CASR[IC] flag and performs no operation.

- For CAU reads:
  - Any attempted read from the first 2-KB region of the address space (an attempted direct read) is illegal and produces an error termination.
  - Within the second 2-KB region of the address space, i.e., addr[11] = 1, only a 64-byte space is treated as a legal CAU store operation. The allowable addresses are defined as:

    addr[11:0] = 1000_10xx_xx_00

    where the 4-bit xxxx value specifies the CAU register number. The CAU supports a subset of the allowable register numbers, 0x0 - 0xA. Attempting a store of a reserved register produces an undefined result.

38.6.2.2 Data integrity checks

Direct writes can send 1, 2, or 3 commands to the CAU in a single 32-bit transfer. As shown in Figure 38-4, the commands include a valid bit located at bits 31, 20, and 9 of the write data where:

- Bit 31 is the valid bit for the first command
- Bit 20 is the valid bit for the second command
- Bit 9 is the valid bit for the third command

The direct write data check validates the combination of these three valid bits. The following table presents the three legal states associated with these bits:

Value of bits 31, 20, and 9	Number of commands included
100	1
110	2
111	3

All other combinations of bits 31, 20, and 9 are illegal and generate an error termination.

38.6.3 CAU commands

The CAU supports the commands shown in the following table. All other encodings are reserved. The CASR[IC] bit is set if an undefined command is issued. A specific illegal command (ILL) is defined to allow software self-checking. Reserved commands must not be issued so as to ensure compatibility with future implementations.

The CMD field specifies the 9-bit CAU opcode for the operation command.

See Assembler equate values for a set of assembly constants used in the command descriptions here. If supported by the assembler, macros can also be created for each instruction. The value CA*x* should be interpreted as any CAU register (CASR, CAA, and CA*n*).

Table 38-2. CAU commands

Type	Command name	Description	CMD									Operation
			8	7	6	5	4	3	2	1	0	
Direct load	CNOP	No Operation	0x000									—
Indirect load	LDR	Load Reg	0x01					CAx				Op1 → CAx
Indirect store	STR	Store Reg	0x02					CAx				CAx → Result
Indirect load	ADR	Add	0x03					CAx				CAx + Op1 → CAx
Indirect load	RADR	Reverse and Add	0x04					CAx				CAx + ByteRev(Op1) → CAx
Direct load	ADRA	Add Reg to Acc	0x05					CAx				CAx + CAA → CAA
Indirect load	XOR	Exclusive Or	0x06					CAx				CAx ^ Op1 → CAx
Indirect load	ROTL	Rotate Left	0x07					CAx				(CAx <<< (Op1 % 32)) | (CAx >>> (32 - (Op1 % 32))) → CAx
Direct load	MVRA	Move Reg to Acc	0x08					CAx				CAx → CAA
Direct load	MVAR	Move Acc to Reg	0x09					CAx				CAA → CAx
Direct load	AESS	AES Sub Bytes	0x0A					CAx				SubBytes(CAx) → CAx
Direct load	AESIS	AES Inv Sub Bytes	0x0B					CAx				InvSubBytes(CAx) → CAx
Indirect load	AESC	AES Column Op	0x0C					CAx				MixColumns(CAx)^Op1→ CAx
Indirect load	AESIC	AES Inv Column Op	0x0D					CAx				InvMixColumns(CAx^Op1) → CAx

Table continues on the next page...

Table 38-2. CAU commands (continued)

Type	Command name	Description	CMD									Operation
			8	7	6	5	4	3	2	1	0	
Direct load	AESR	AES Shift Rows	0x0E0									ShiftRows(CA0-CA3) → CA0-CA3
Direct load	AESIR	AES Inv Shift Rows	0x0F0									InvShiftRows(CA0-CA3)→ CA0-CA3
Direct load	DESR	DES Round	0x10					IP	FP	KS[1:0]		DES Round(CA0-CA3) →CA0-CA3
Direct load	DESK	DES Key Setup	0x11					0	0	CP	DC	DES Key Op(CA0-CA1)→ CA0-CA1 Key Parity Error & CP → CASR[1]
Direct load	HASH	Hash Function	0x12					0	HF[2:0]			Hash Func(CA1-CA3)+CAA→ CAA
Direct load	SHS	Secure Hash Shift	0x130									CAA <<< 5→ CAA, CAA→CA0, CA0→CA1, CA1 <<< 30 → CA2, CA2→CA3, CA3→CA4
Direct load	MDS	Message Digest Shift	0x140									CA3→CAA, CAA→CA1, CA1→CA2, CA2→CA3,
Direct load	SHS2	Secure Hash Shift 2	0x150									CAA→CA0, CA0→CA1, CA1 → CA2, CA2→CA3, CA3 + CA8 →CA4, CA4 → CA5, CA5 → CA6, CA6 → CA7
Direct load	ILL	Illegal Command	0x1F0									0x1→CASR[IC]

38.6.3.1 Coprocessor No Operation (CNOP)

The CNOP command is the coprocessor no-op. It is issued by the CAU and consumes a location in the CAU FIFO, but has no effect on any CAU register.

38.6.3.2 Load Register (LDR)

The LDR command loads CAx with the source data specified by the write data.

38.6.3.3 Store Register (STR)

The STR command returns the value of CAx specified in the read address to the destination specified as read data.

38.6.3.4 Add to Register (ADR)

The ADR command adds the source operand specified by the write data to CAx and stores the result in CAx.

38.6.3.5 Reverse and Add to Register (RADR)

The RADR command performs a byte reverse on the source operand specified by the write data, adds that value to CAx, and stores the result in CAx. The table below shows an example.

Table 38-3. RADR command example

Operand	CAx before	CAx after
0x0102_0304	0xA0B0_C0D0	0xA4B3_C2D1

38.6.3.6 Add Register to Accumulator (ADRA)

The ADRA command adds CAx to CAA and stores the result in CAA.

38.6.3.7 Exclusive Or (XOR)

The XOR command performs an exclusive-or of the source operand specified by the write data with CAx and stores the result in CAx.

38.6.3.8 Rotate Left (ROTL)

ROTL rotates the CAx bits to the left with the result stored back to CAx. The number of bits to rotate is the value specified by the write data modulo 32.

38.6.3.9 Move Register to Accumulator (MVRA)

The MVRA command moves the value from the source register CAx to the destination register CAA.

38.6.3.10 Move Accumulator to Register (MVAR)

The MVAR command moves the value from source register CAA to the destination register CAx.

38.6.3.11 AES Substitution (AESS)

The AESS command performs the AES byte substitution operation on CAx and stores the result back to CAx.

38.6.3.12 AES Inverse Substitution (AESIS)

The AESIS command performs the AES inverse byte substitution operation on CAx and stores the result back to CAx.

38.6.3.13 AES Column Operation (AESC)

The AESC command performs the AES column operation on the contents of CAx. It then performs an exclusive-or of that result with the source operand specified by the write data and stores the result in CAx.

38.6.3.14 AES Inverse Column Operation (AESIC)

The AESIC command performs an exclusive-or operation of the source operand specified by the write data on the contents of CAx followed by the AES inverse mix column operation on that result and stores the result back in CAx.

38.6.3.15 AES Shift Rows (AESR)

The AESR command performs the AES shift rows operation on registers CA0, CA1, CA2, and CA3. The table below shows an example.

Table 38-4. AESR command example

Register	Before	After
CA0	0x0102_0304	0x0106_0B00
CA1	0x0506_0708	0x050A_0F04
CA2	0x090A_0B0C	0x090E_0308
CA3	0x0D0E_0F00	0x0D02_070C

Where:

- row 1 = CA0[31:24], CA1[31:24], CA2[31:24], CA3[31:24]
- row 2 = CA0[23:16], CA1[23:16], CA2[23:16], CA3[23:16]
- row 3 = CA0[15:8], CA1[15:8], CA2[15:8], CA3[15:8]
- row 4 = CA0[7:0], CA1[7:0], CA2[7:0], CA3[7:0]

38.6.3.16 AES Inverse Shift Rows (AESIR)

The AESIR command performs the AES inverse shift rows operation on registers CA0, CA1, CA2, and CA3. The table below shows an example.

Table 38-5. AESIR command example

Register	Before	After
CA0	0x0106_0B00	0x0102_0304
CA1	0x050A_0F04	0x0506_0708
CA2	0x090E_0308	0x090A_0B0C
CA3	0x0D02_070C	0x0D0E_0F00

Where:

- row 1 = CA0[31:24], CA1[31:24], CA2[31:24], CA3[31:24]
- row 2 = CA0[23:16], CA1[23:16], CA2[23:16], CA3[23:16]
- row 3 = CA0[15:8], CA1[15:8], CA2[15:8], CA3[15:8]
- row 4 = CA0[7:0], CA1[7:0], CA2[7:0], CA3[7:0]

38.6.3.17 DES Round (DESR)

The DESR command performs a round of the DES algorithm and a key schedule update with the following source and destination designations: CA0=C, CA1=D, CA2=L, CA3=R. If the IP bit is set, DES initial permutation performs on CA2 and CA3 before the round operation. If the FP bit is set, DES final permutation, that is, inverse initial permutation, performs on CA2 and CA3 after the round operation. The round operation uses the source values from registers CA0 and CA1 for the key addition operation. The KSx field specifies the shift for the key schedule operation to update the values in CA0 and CA1. The following table defines the specific shift function performed based on the KSx field.

Table 38-6. Key shift function codes

KSx code	KSx define	Shift function
0	KSL1	Left 1
1	KSL2	Left 2
2	KSR1	Right 1
3	KSR2	Right 2

38.6.3.18 DES Key setup (DESK)

The DESK command performs the initial key transformation, permuted choice 1, defined by the DES algorithm on CA0 and CA1 with CA0 containing bits 1–32 of the key and CA1 containing bits 33–64 of the key[1] . If the DC bit is set, no shift operation performs and the values C_0 and D_0 store back to CA0 and CA1, respectively. The DC bit must be set for decrypt operations. If the DC bit is not set, a left shift by one also occurs and the values C_1 and D_1 store back to CA0 and CA1, respectively. The DC bit should be cleared for encrypt operations. If the CP bit is set and a key parity error is detected, CASR[DPE] bit is set; otherwise, it is cleared.

38.6.3.19 Hash Function (HASH)

The HASH command performs a hashing operation on a set of registers and adds that result to the value in CAA and stores the result in CAA. The specific hash function performed is based on the HFx field as defined in the table below.

This table uses the following terms:

- ROTR^n(CAx): rotate CAx register right *n* times
- SHR^n(CAx): shift CAx register right *n* times

Table 38-7. Hash Function codes

HFx code	HFx define	Hash Function	Hash logic
0	HFF	MD5 F()	$(CA1 \ \& \ CA2) \vert (\overline{CA1} \ \& \ CA3)$
1	HFG	MD5 G()	$(CA1 \ \& \ CA3) \vert (CA2 \ \& \ \overline{CA3})$

Table continues on the next page...

[1] The DES algorithm numbers the most significant bit of a block as bit 1 and the least significant as bit 64.

Table 38-7. Hash Function codes (continued)

HFx code	HFx define	Hash Function	Hash logic
2	HFH	MD5 H(), SHA Parity()	CA1 ^ CA2 ^ CA3
3	HFI	MD5 I()	CA2 ^ (CA1 | $\overline{\text{CA3}}$)
4	HFC	SHA Ch()	(CA1 & CA2) ^ ($\overline{\text{CA1}}$ & CA3)
5	HFM	SHA Maj()	(CA1 & CA2) ^ (CA1 & CA3) ^ (CA2 & CA3)
6	HF2C	SHA-256 Ch()	(CA4 & CA5) ^ ($\overline{\text{CA1}}$ & CA6)
7	HF2M	SHA-256 Maj()	(CA0 & CA1) ^ (CA0 & CA2) ^ (CA1 & CA2)
8	HF2S	SHA-256 Sigma 0	ROTR^{2}(CA0) ^ ROTR^{13}(CA0) ^ ROTR^{22}(CA0)
9	HF2T	SHA-256 Sigma 1	ROTR^{6}(CA4) ^ ROTR^{11}(CA4) ^ ROTR^{25}(CA4)
A	HF2U	SHA-256 Sigma 0	ROTR^{7}(CA8) ^ ROTR^{18}(CA8) ^ SHR^{3}(CA8)
B	HF2V	SHA-256 Sigma 1	ROTR^{17}(CA8) ^ ROTR^{19}(CA8) ^ SHR^{10}(CA8)

38.6.3.20 Secure Hash Shift (SHS)

The SHS command does a set of parallel register-to-register move and shift operations for implementing SHA-1. The following source and destination assignments are made:

Register	Value prior to command	Value after command executes
CA4	CA4	CA3
CA3	CA3	CA2
CA2	CA2	CA1<<<30
CA1	CA1	CA0
CA0	CA0	CAA
CAA	CAA	CAA<<<5

38.6.3.21 Message Digest Shift (MDS)

The MDS command does a set of parallel register-to-register move operations for implementing MD5. The following source and destination assignments are made:

Register	Value prior to command	Value after command executes
CA3	CA3	CA2
CA2	CA2	CA1
CA1	CA1	CAA
CAA	CAA	CA3

38.6.3.22 Secure Hash Shift 2 (SHS2)

The SHS2 command does an addition and a set of register to register moves in parallel for implementing SHA-256. The following source and destination assignments are made:

Register	Value prior to command	Value after command executes
CA7	CA7	CA6
CA6	CA6	CA5
CA5	CA5	CA4
CA4	CA4	CA3+CA8
CA3	CA3	CA2
CA2	CA2	CA1
CA1	CA1	CA0
CA0	CA0	CAA

38.6.3.23 Illegal command (ILL)

The ILL command is a specific illegal command that sets CASR[IC]. All other illegal commands are reserved for use in future implementations.

38.7 Application/initialization information

This section discusses how to initialize and use the CAU.

38.7.1 Code example

A code fragment is shown below as an example of how the CAU is used. This example shows the round function of the AES algorithm. Core registers are defined as follows:

- R1 points to the key schedule
- R3 contains 3 direct CAU commands
- R8 contains 2 direct CAU commands
- R9 contains an indirect CAU command
- FP points to the CAU indirect command address space
- IP points to the CAU direct command space

```
    movw    fp, #:lower16:MMCAU_PPB_INDIRECT              @ fp -> MMCAU_PPB_INDIRECT
    movt    fp, #:upper16:MMCAU_PPB_INDIRECT
    movw    ip, #:lower16:MMCAU_PPB_DIRECT                @ ip -> MMCAU_PPB_DIRECT
    movt    ip, #:upper16:MMCAU_PPB_DIRECT
```

```
# r3 = mmcau_3_cmds(AESS+CA0,AESS+CA1,AESS+CA2)
    movw    r3, #:lower16:(0x80100200+(AESS+CA0)<<22+(AESS+CA1)<<11+AESS+CA2)
    movt    r3, #:upper16:(0x80100200+(AESS+CA0)<<22+(AESS+CA1)<<11+AESS+CA2)

# r8 = mmcau_2_cmds(AESS+CA3,AESR)
    movw    r8, #:lower16:(0x80100000+(AESS+CA3)<<22+(AESR)<<11)
    movt    r8, #:upper16:(0x80100000+(AESS+CA3)<<22+(AESR)<<11)

    add     r9, fp, $((AESC+CA0)<<2)                            @ r9 = mmcau_cmd(AESC+CA0)

    str     r3, [ip]                                        @ sub bytes w0, w1, w2
    str     r8, [ip]                                        @ sub bytes w3, shift rows
    ldmia   r1!, {r4-r7}                                    @ get next 4 keys; r1++
    stmia   r9, {r4-r7}                                     @ mix columns, add keys
```

38.7.2 Assembler equate values

The following equates ease programming of the CAU.

```
; CAU Registers (CAx)
     .set CASR,0x0
     .set CAA,0x1
     .set CA0,0x2
     .set CA1,0x3
     .set CA2,0x4
     .set CA3,0x5
     .set CA4,0x6
     .set CA5,0x7
     .set CA6,0x8
     .set CA7,0x9
     .set CA8,0xA
; CAU Commands
     .set CNOP,0x000
     .set LDR,0x010
     .set STR,0x020
     .set ADR,0x030
     .set RADR,0x040
     .set ADRA,0x050
     .set XOR,0x060
     .set ROTL,0x070
     .set MVRA,0x080
     .set MVAR,0x090
     .set AESS,0x0A0
     .set AESIS,0x0B0
     .set AESC,0x0C0
     .set AESIC,0x0D0
     .set AESR,0x0E0
     .set AESIR,0x0F0
     .set DESR,0x100
     .set DESK,0x110
     .set HASH,0x120
     .set SHS,0x130
     .set MDS,0x140
     .set SHS2,0x150
     .set ILL,0x1F0
; DESR  Fields
     .set IP,0x08         ; initial permutation
     .set FP,0x04         ; final permutation
     .set KSL1,0x00       ; key schedule left 1 bit
     .set KSL2,0x01       ; key schedule left 2 bits
     .set KSR1,0x02       ; key schedule right 1 bit
     .set KSR2,0x03       ; key schedule right 2 bits
; DESK Field
     .set DC,0x01         ; decrypt key schedule
     .set CP,0x02          ; check parity
; HASH Functions Codes
     .set HFF,0x0          ; MD5 F() CA1&CA2 | ~CA1&CA3
     .set HFG,0x1          ; MD5 G() CA1&CA3 | CA2&~CA3
     .set HFH,0x2          ; MD5 H(), SHA Parity() CA1^CA2^CA3
     .set HFI,0x3          ; MD5 I()  CA2^(CA1|~CA3)
     .set HFC,0x4          ; SHA Ch() CA1&CA2 ^ ~CA1&CA3
     .set HFM,0x5          ; SHA Maj() CA1&CA2 ^ CA1&CA3 ^ CA2&CA3
     .set HF2C,0x6         ; SHA-256 Ch() CA4&CA5 ^ ~CA4&CA6
     .set HF2M,0x7         ; SHA-256 Maj() CA0&CA1 ^ CA0&CA2 ^ CA1&CA2
     .set HF2S,0x8         ; SHA-256 Sigma 0 ROTR2(CA0)^ROTR13(CA0)^ROTR22(CA0)
     .set HF2T,0x9         ; SHA-256 Sigma 1 ROTR6(CA4)^ROTR11(CA4)^ROTR25(CA4)
     .set HF2U,0xA         ; SHA-256 sigma 0 ROTR7(CA8)^ROTR18(CA8)^SHR3(CA8)
     .set HF2V,0xB         ; SHA-256 sigma 1 ROTR17(CA8)^ROTR19(CA8)^SHR10(CA8)
```

Chapter 39
MCU: Random Number Generator Accelerator (RNGA)

39.1 Introduction

NOTE

> For the chip-specific implementation details of this module's instances, see the chip configuration information.

This chapter describes the random-number-generator accelerator RNGA, including a programming model, functional description, and application information. Throughout this chapter, the terms "RNG" and "RNGA" are meant to be synonymous.

39.1.1 Overview

RNGA is a digital integrated circuit capable of generating 32-bit random numbers. The random bits are generated using shift registers with clocks derived from two free-running, independent ring oscillators. The configuration of the shift registers ensures statistically good data, that is, data that looks random. The oscillators, with their unknown frequencies and independent phases, provide the means of generating the required entropy needed to create random data. The random words generated by RNGA are loaded into an output register (OR). RNGA is designed to generate an error interrupt (if not masked), if OR is read and does not contain valid random data. OR contains valid random data if the LVL field in the status register (SR) is 1.

It is important to note there is no known cryptographic proof showing this is a secure method of generating random data. In fact, there may be an attack against this random number generator if its output is used directly in a cryptographic application. The attack is based on the linearity of the internal shift registers. Therefore, it is highly recommended that this random data produced by this module be used as an entropy source to provide an input seed to a NIST-approved pseudo-random-number generator based on DES or SHA-1 and defined in *NIST FIPS PUB 186-2 Appendix 3* and *NIST FIPS PUB SP 800-90*.

The requirement is to maximize the entropy of this input seed. In order to do this, when data is extracted from RNGA as quickly as the hardware allows, there are about one or two bits of added entropy per 32-bit word. Any single bit of that word contains that entropy. Therefore, when used as an entropy source, a random number should be generated for each bit of entropy required, and the least significant bit (any bit would be equivalent) of each word retained. The remainder of each random number should then be discarded. Used this way, even with full knowledge of the internal state of RNGA and all prior random numbers, an attacker is not able to predict the values of the extracted bits.

Other sources of entropy can be used along with RNGA to generate the seed to the pseudorandom algorithm. The more random sources combined to create the seed, the better. The following is a list of sources that can be easily combined with the output of this module:

- Current time using highest precision possible
- Real-time system inputs that can be characterized as "random"
- Other entropy supplied directly by the user

39.2 Modes of operation

RNGA supports the following modes of operation.

Table 39-1. Modes of operation supported by RNGA

Mode	Description
Normal	The ring-oscillator clocks are active; RNGA generates entropy (randomness) from the clocks and stores it in shift registers.
Sleep	The ring-oscillator clocks are inactive; RNGA does not generate entropy.

39.2.1 Entering Normal mode

To enter Normal mode, write 0 to CR[SLP].

39.2.2 Entering Sleep mode

To enter Sleep mode, write 1 to CR[SLP].

39.3 Memory map and register definition

This section describes the RNGA registers.

RNG memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4002_9000	RNGA Control Register (RNG_CR)	32	R/W	0000_0000h	39.3.1/853
4002_9004	RNGA Status Register (RNG_SR)	32	R	0001_0000h	39.3.2/855
4002_9008	RNGA Entropy Register (RNG_ER)	32	W (always reads 0)	0000_0000h	39.3.3/857
4002_900C	RNGA Output Register (RNG_OR)	32	R	0000_0000h	39.3.4/857

39.3.1 RNGA Control Register (RNG_CR)

Controls the operation of RNGA.

Address: 4002_9000h base + 0h offset = 4002_9000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0											SLP	0	INTM	HA	GO
W													CLRI			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

RNG_CR field descriptions

Field	Description
31–5 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
4 SLP	Sleep Specifies whether RNGA is in Sleep or Normal mode. **NOTE:** You can also enter Sleep mode by asserting the DOZE signal. 0 Normal mode 1 Sleep (low-power) mode
3 CLRI	Clear Interrupt Clears the interrupt by resetting the error-interrupt indicator (SR[ERRI]). 0 Do not clear the interrupt. 1 Clear the interrupt. When you write 1 to this field, RNGA then resets the error-interrupt indicator (SR[ERRI]). This bit always reads as 0.
2 INTM	Interrupt Mask Masks the triggering of an error interrupt to the interrupt controller when an OR underflow condition occurs. An OR underflow condition occurs when you read OR[RANDOUT] and SR[OREG_LVL]=0. See the Output Register (OR) description. 0 Not masked 1 Masked
1 HA	High Assurance Enables notification of security violations (via SR[SECV]). A security violation occurs when you read OR[RANDOUT] and SR[OREG_LVL]=0. **NOTE:** This field is sticky. After enabling notification of security violations, you must reset RNGA to disable them again. 0 Disabled 1 Enabled
0 GO	Go Specifies whether random-data generation and loading (into OR[RANDOUT]) is enabled. **NOTE:** This field is sticky. You must reset RNGA to stop RNGA from loading OR[RANDOUT] with data. 0 Disabled 1 Enabled

39.3.2 RNGA Status Register (RNG_SR)

Indicates the status of RNGA. This register is read-only.

Address: 4002_9000h base + 4h offset = 4002_9004h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0								OREG_SIZE							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	OREG_LVL								0			SLP	ERRI	ORU	LRS	SECV
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

RNG_SR field descriptions

Field	Description
31–24 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
23–16 OREG_SIZE	Output Register Size Indicates the size of the Output (OR) register in terms of the number of 32-bit random-data words it can hold. 1 One word (this value is fixed)
15–8 OREG_LVL	Output Register Level Indicates the number of random-data words that are in OR[RANDOUT], which indicates whether OR[RANDOUT] is valid. **NOTE:** If you read OR[RANDOUT] when SR[OREG_LVL] is not 0, then the contents of a random number contained in OR[RANDOUT] are returned, and RNGA writes 0 to both OR[RANDOUT] and SR[OREG_LVL]. 0 No words (empty) 1 One word (valid)

Table continues on the next page...

RNG_SR field descriptions (continued)

Field	Description
7–5 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
4 SLP	Sleep Specifies whether RNGA is in Sleep or Normal mode. **NOTE:** You can also enter Sleep mode by asserting the DOZE signal. 0 Normal mode 1 Sleep (low-power) mode
3 ERRI	Error Interrupt Indicates whether an OR underflow condition has occurred since you last cleared the error interrupt (CR[CLRI]) or RNGA was reset, regardless of whether the error interrupt is masked (CR[INTM]). An OR underflow condition occurs when you read OR[RANDOUT] and SR[OREG_LVL]=0. **NOTE:** After you reset the error-interrupt indicator (via CR[CLRI]), RNGA writes 0 to this field. 0 No underflow 1 Underflow
2 ORU	Output Register Underflow Indicates whether an OR underflow condition has occurred since you last read this register (SR) or RNGA was reset, regardless of whether the error interrupt is masked (CR[INTM]). An OR underflow condition occurs when you read OR[RANDOUT] and SR[OREG_LVL]=0. **NOTE:** After you read this register, RNGA writes 0 to this field. 0 No underflow 1 Underflow
1 LRS	Last Read Status Indicates whether the most recent read of OR[RANDOUT] caused an OR underflow condition, regardless of whether the error interrupt is masked (CR[INTM]). An OR underflow condition occurs when you read OR[RANDOUT] and SR[OREG_LVL]=0. **NOTE:** After you read this register, RNGA writes 0 to this field. 0 No underflow 1 Underflow
0 SECV	Security Violation Used only when high assurance is enabled (CR[HA]). Indicates that a security violation has occurred. **NOTE:** This field is sticky. To clear SR[SECV], you must reset RNGA. 0 No security violation 1 Security violation

39.3.3 RNGA Entropy Register (RNG_ER)

Specifies an entropy value that RNGA uses in addition to its ring oscillators to seed its pseudorandom algorithm. This is a write-only register; reads return all zeros.

Address: 4002_9000h base + 8h offset = 4002_9008h

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

RNG_ER field descriptions

Field	Description
EXT_ENT	External Entropy Specifies an entropy value that RNGA uses in addition to its ring oscillators to seed its pseudorandom algorithm. **NOTE:** Specifying a value for this field is optional but recommended. You can write to this field at any time during operation.

39.3.4 RNGA Output Register (RNG_OR)

Stores a random-data word generated by RNGA.

Address: 4002_9000h base + Ch offset = 4002_900Ch

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

RNG_OR field descriptions

Field	Description
RANDOUT	Random Output Stores a random-data word generated by RNGA. This is a read-only field. **NOTE:** Before reading RANDOUT, be sure it is valid (SR[OREG_LVL]=1).

RNG_OR field descriptions (continued)

Field	Description	
	0	Invalid data (if you read this field when it is 0 and SR[OREG_LVL] is 0, RNGA then writes 1 to SR[ERRI], SR[ORU], and SR[LRS]; when the error interrupt is not masked (CR[INTM]=0), RNGA also asserts an error interrupt request to the interrupt controller).
	All other values	Valid data (if you read this field when SR[OREG_LVL] is not 0, RNGA returns RANDOUT, and then writes 0 to this field and to SR[OREG_LVL]).

39.4 Functional description

This is a block diagram of RNGA.

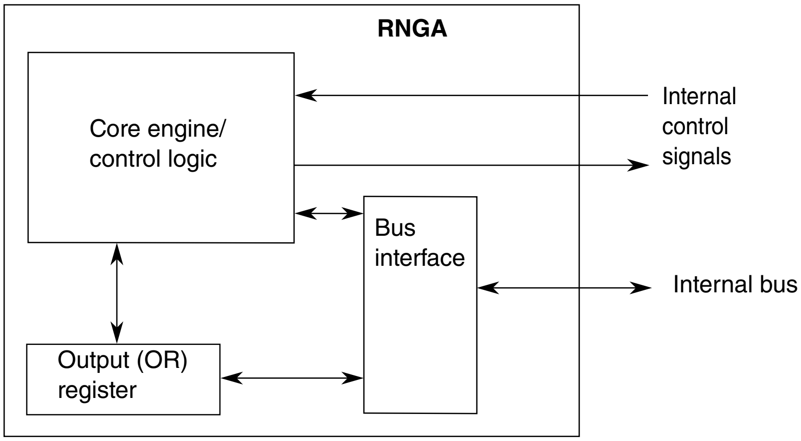


Figure 39-1. RNGA block diagram

39.4.1 Output (OR) register

The Output (OR) register provides temporary storage for random data generated by the core engine / control logic. The Status (SR) register allows the user to monitor the presence of valid random data in OR through SR[OREG_LVL].

If the OR is read while containing valid random data (as signaled by SR[OREG_LVL] = 1), the valid data is returned, then OR and SR[OREG_LVL] are both cleared. If the user reads from OR when it is empty, RNGA returns all zeros and, if the interrupt is enabled, RNGA drives a request to the interrupt controller. Polling SR[OREG_LVL] is very important to make sure random values are present before reading from OR.

39.4.2 Core engine / control logic

This block contains RNGA's control logic as well as its core engine used to generate random data.

39.4.2.1 Control logic

The control logic contains the address decoder, all addressable registers, and control state machines for RNGA. This block is responsible for communication with both the peripheral interface and the Output (OR) register interface. The block also controls the core engine to generate random data. The general functionality of the block is as follows:

After reset, RNGA operates in Normal mode as follows:

1. The core engine generates entropy and stores it in the shift registers.
2. After you enable random-data generation by loading CR[GO], every 256 clock cycles the core engine generates a new random-data word. If SR[OREG_LVL] = 0, then the control block loads the new random data into OR and set SR[OREG_LVL] = 1; else the new data is discarded.

39.4.2.2 Core engine

The core engine block contains the logic used to generate random data. The logic within the core engine contains the internal shift registers as well as the logic used to generate the two oscillator-based clocks. The control logic determines how the shift registers are configured as well as when the oscillator clocks are turned on.

39.5 Initialization/application information

The intended general operation of RNGA is as follows:

1. Reset/initialize.
2. Write 1 to CR[INTM], CR[HA], and CR[GO].
3. Poll SR[OREG_LVL] until it is not 0.
4. When SR[OREG_LVL] is not 0, read the available random data from OR[RANDOUT].
5. Repeat steps 3 and 4 as needed.

For application information, see Overview.

Chapter 40
MCU: Analog-to-Digital Converter (ADC)

40.1 Chip-specific ADC information

40.1.1 Voltage Reference Selection

The ADC can only be configured to accept the voltage reference pair VREFH and VREFL. VREFH is internally tied to VDDA, and VREFL is internally tied to VSSA. The voltage reference selection bits SC2[REFSEL] must be set to the default 00. Alternate reference options, such as V_{ALTH} and V_{ALTL}, are not available.

40.2 Introduction

NOTE

> For the chip-specific implementation details of this module's instances, see the chip configuration information.

The 16-bit analog-to-digital converter (ADC) is a successive approximation ADC designed for operation within an integrated microcontroller system-on-chip.

NOTE

> For the chip specific modes of operation, see the power management information of the device.

40.2.1 Features

Following are the features of the ADC module.

- Linear successive approximation algorithm with up to 16-bit resolution
- Up to four pairs of differential and 24 single-ended external analog inputs
- Output modes:
  - differential 16-bit, 13-bit, 11-bit, and 9-bit modes
  - single-ended 16-bit, 12-bit, 10-bit, and 8-bit modes
- Output format in 2's complement 16-bit sign extended for differential modes
- Output in right-justified unsigned format for single-ended
- Single or continuous conversion, that is, automatic return to idle after single conversion
- Configurable sample time and conversion speed/power
- Conversion complete/hardware average complete flag and interrupt
- Input clock selectable from up to four sources
- Operation in low-power modes for lower noise
- Asynchronous clock source for lower noise operation with option to output the clock
- Selectable hardware conversion trigger with hardware channel select
- Automatic compare with interrupt for less-than, greater-than or equal-to, within range, or out-of-range, programmable value
- Temperature sensor
- Hardware average function
- Selectable voltage reference: external or alternate
- Self-Calibration mode

40.2.2 Block diagram

The following figure is the ADC module block diagram.

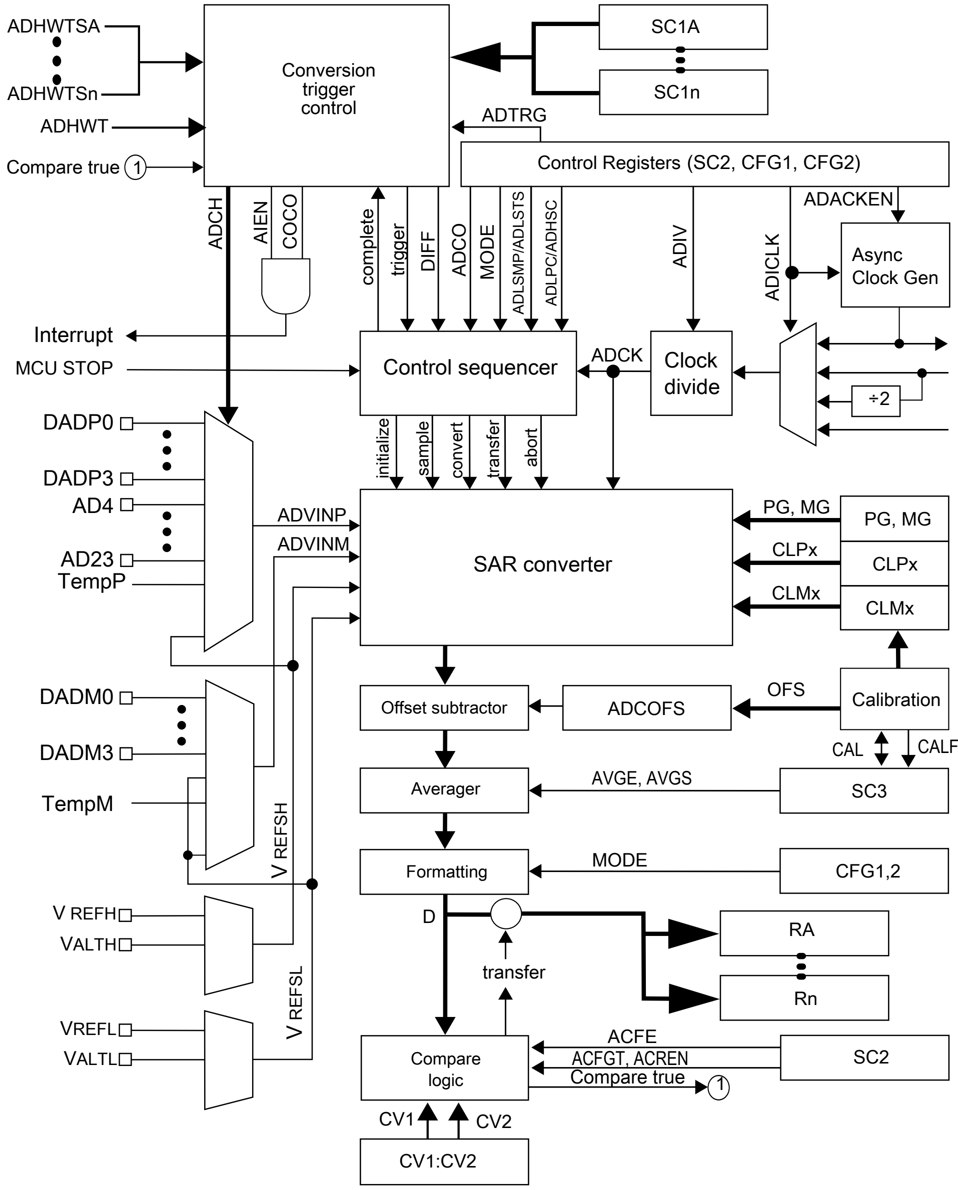


Figure 40-1. ADC block diagram

40.3 ADC signal descriptions

The ADC module supports up to 4 pairs of differential inputs and up to 24 single-ended inputs.

Each differential pair requires two inputs, DADPx and DADMx. The ADC also requires four supply/reference/ground connections.

NOTE

For the number of channels supported on this device as well as information regarding other chip-specific inputs into the ADC block, see the chip-specific ADC configuration information.

Table 40-1. ADC signal descriptions

Signal	Description	I/O
DADP3–DADP0	Differential Analog Channel Inputs	I
DADM3–DADM0	Differential Analog Channel Inputs	I
AD*n*	Single-Ended Analog Channel Inputs	I
V_{REFSH}	Voltage Reference Select High	I
V_{REFSL}	Voltage Reference Select Low	I
V_{DDA}	Analog Power Supply	I
V_{SSA}	Analog Ground	I

40.3.1 Analog Power (V_{DDA})

The ADC analog portion uses V_{DDA} as its power connection. In some packages, V_{DDA} is connected internally to V_{DD}. If externally available, connect the V_{DDA} pin to the same voltage potential as V_{DD}. External filtering may be necessary to ensure clean V_{DDA} for good results.

40.3.2 Analog Ground (V_{SSA})

The ADC analog portion uses V_{SSA} as its ground connection. In some packages, V_{SSA} is connected internally to V_{SS}. If externally available, connect the V_{SSA} pin to the same voltage potential as V_{SS}.

40.3.3 Voltage Reference Select

V_{REFSH} and V_{REFSL} are the high and low reference voltages for the ADC module.

The ADC can be configured to accept one of two voltage reference pairs for V_{REFSH} and V_{REFSL}. Each pair contains a positive reference that must be between the minimum Ref Voltage High and V_{DDA}, and a ground reference that must be at the same potential as V_{SSA}. The two pairs are external (V_{REFH} and V_{REFL}) and alternate (V_{ALTH} and V_{ALTL}). These voltage references are selected using SC2[REFSEL]. The alternate V_{ALTH} and

V_{ALTL} voltage reference pair may select additional external pins or internal sources depending on MCU configuration. See the chip configuration information on the Voltage References specific to this MCU.

In some packages, V_{REFH} is connected in the package to V_{DDA} and V_{REFL} to V_{SSA}. If externally available, the positive reference(s) may be connected to the same potential as V_{DDA} or may be driven by an external source to a level between the minimum Ref Voltage High and the V_{DDA} potential. V_{REFH} must never exceed V_{DDA}. Connect the ground references to the same voltage potential as V_{SSA}.

40.3.4 Analog Channel Inputs (ADx)

The ADC module supports up to 24 single-ended analog inputs. A single-ended input is selected for conversion through the SC1[ADCH] channel select bits when SC1n[DIFF] is low.

40.3.5 Differential Analog Channel Inputs (DADx)

The ADC module supports up to four differential analog channel inputs. Each differential analog input is a pair of external pins, DADPx and DADMx, referenced to each other to provide the most accurate analog to digital readings. A differential input is selected for conversion through SC1[ADCH] when SC1n[DIFF] is high. All DADPx inputs may be used as single-ended inputs if SC1n[DIFF] is low. In certain MCU configurations, some DADMx inputs may also be used as single-ended inputs if SC1n[DIFF] is low. For ADC connections specific to this device, see the chip-specific ADC information.

40.4 Memory map and register definitions

This section describes the ADC registers.

ADC memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4003_B000	ADC Status and Control Registers 1 (ADC0_SC1A)	32	R/W	0000_001Fh	40.4.1/866
4003_B004	ADC Status and Control Registers 1 (ADC0_SC1B)	32	R/W	0000_001Fh	40.4.1/866
4003_B008	ADC Configuration Register 1 (ADC0_CFG1)	32	R/W	0000_0000h	40.4.2/870
4003_B00C	ADC Configuration Register 2 (ADC0_CFG2)	32	R/W	0000_0000h	40.4.3/871
4003_B010	ADC Data Result Register (ADC0_RA)	32	R	0000_0000h	40.4.4/872

Table continues on the next page...

ADC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4003_B014	ADC Data Result Register (ADC0_RB)	32	R	0000_0000h	40.4.4/872
4003_B018	Compare Value Registers (ADC0_CV1)	32	R/W	0000_0000h	40.4.5/874
4003_B01C	Compare Value Registers (ADC0_CV2)	32	R/W	0000_0000h	40.4.5/874
4003_B020	Status and Control Register 2 (ADC0_SC2)	32	R/W	0000_0000h	40.4.6/875
4003_B024	Status and Control Register 3 (ADC0_SC3)	32	R/W	0000_0000h	40.4.7/877
4003_B028	ADC Offset Correction Register (ADC0_OFS)	32	R/W	0000_0004h	40.4.8/878
4003_B02C	ADC Plus-Side Gain Register (ADC0_PG)	32	R/W	0000_8200h	40.4.9/879
4003_B030	ADC Minus-Side Gain Register (ADC0_MG)	32	R/W	0000_8200h	40.4.10/ 879
4003_B034	ADC Plus-Side General Calibration Value Register (ADC0_CLPD)	32	R/W	0000_000Ah	40.4.11/ 880
4003_B038	ADC Plus-Side General Calibration Value Register (ADC0_CLPS)	32	R/W	0000_0020h	40.4.12/ 881
4003_B03C	ADC Plus-Side General Calibration Value Register (ADC0_CLP4)	32	R/W	0000_0200h	40.4.13/ 881
4003_B040	ADC Plus-Side General Calibration Value Register (ADC0_CLP3)	32	R/W	0000_0100h	40.4.14/ 882
4003_B044	ADC Plus-Side General Calibration Value Register (ADC0_CLP2)	32	R/W	0000_0080h	40.4.15/ 882
4003_B048	ADC Plus-Side General Calibration Value Register (ADC0_CLP1)	32	R/W	0000_0040h	40.4.16/ 883
4003_B04C	ADC Plus-Side General Calibration Value Register (ADC0_CLP0)	32	R/W	0000_0020h	40.4.17/ 883
4003_B054	ADC Minus-Side General Calibration Value Register (ADC0_CLMD)	32	R/W	0000_000Ah	40.4.18/ 884
4003_B058	ADC Minus-Side General Calibration Value Register (ADC0_CLMS)	32	R/W	0000_0020h	40.4.19/ 884
4003_B05C	ADC Minus-Side General Calibration Value Register (ADC0_CLM4)	32	R/W	0000_0200h	40.4.20/ 885
4003_B060	ADC Minus-Side General Calibration Value Register (ADC0_CLM3)	32	R/W	0000_0100h	40.4.21/ 885
4003_B064	ADC Minus-Side General Calibration Value Register (ADC0_CLM2)	32	R/W	0000_0080h	40.4.22/ 886
4003_B068	ADC Minus-Side General Calibration Value Register (ADC0_CLM1)	32	R/W	0000_0040h	40.4.23/ 886
4003_B06C	ADC Minus-Side General Calibration Value Register (ADC0_CLM0)	32	R/W	0000_0020h	40.4.24/ 887

40.4.1 ADC Status and Control Registers 1 (ADCx_SC1*n*)

SC1A is used for both software and hardware trigger modes of operation.

To allow sequential conversions of the ADC to be triggered by internal peripherals, the ADC can have more than one status and control register: one for each conversion. The SC1B–SC1n registers indicate potentially multiple SC1 registers for use only in hardware trigger mode. See the chip configuration information about the number of SC1n registers specific to this device. The SC1n registers have identical fields, and are used in a "ping-pong" approach to control ADC operation.

At any one point in time, only one of the SC1n registers is actively controlling ADC conversions. Updating SC1A while SC1n is actively controlling a conversion is allowed, and vice-versa for any of the SC1n registers specific to this MCU.

Writing SC1A while SC1A is actively controlling a conversion aborts the current conversion. In Software Trigger mode, when SC2[ADTRG]=0, writes to SC1A subsequently initiate a new conversion, if SC1[ADCH] contains a value other than all 1s (module disabled).

Writing any of the SC1n registers while that specific SC1n register is actively controlling a conversion aborts the current conversion. None of the SC1B-SC1n registers are used for software trigger operation and therefore writes to the SC1B–SC1n registers do not initiate a new conversion.

Address: 4003_B000h base + 0h offset + (4d × i), where i=0d to 1d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								COCO	AIEN	DIFF	ADCH				
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1

ADCx_SC1n field descriptions

Field	Description
31–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7 COCO	Conversion Complete Flag This is a read-only field that is set each time a conversion is completed when the compare function is disabled, or SC2[ACFE]=0 and the hardware average function is disabled, or SC3[AVGE]=0. When the compare function is enabled, or SC2[ACFE]=1, COCO is set upon completion of a conversion only if the compare result is true. When the hardware average function is enabled, or SC3[AVGE]=1, COCO is set upon completion of the selected number of conversions (determined by AVGS). COCO in SC1A is also set at the completion of a calibration sequence. COCO is cleared when the respective SC1n register is written or when the respective Rn register is read. 0 Conversion is not completed. 1 Conversion is completed.
6 AIEN	Interrupt Enable Enables conversion complete interrupts. When COCO becomes set while the respective AIEN is high, an interrupt is asserted. 0 Conversion complete interrupt is disabled. 1 Conversion complete interrupt is enabled.
5 DIFF	Differential Mode Enable Configures the ADC to operate in differential mode. When enabled, this mode automatically selects from the differential channels, and changes the conversion algorithm and the number of cycles to complete a conversion. 0 Single-ended conversions and input channels are selected. 1 Differential conversions and input channels are selected.
ADCH	Input channel select Selects one of the input channels. The input channel decode depends on the value of DIFF. DAD0-DAD3 are associated with the input pin pairs DADPx and DADMx. **NOTE:** Some of the input channel options in the bitfield-setting descriptions might not be available for your device. For the actual ADC channel assignments for your device, see the Chip Configuration details. The successive approximation converter subsystem is turned off when the channel select bits are all set, that is, ADCH = 11111. This feature allows explicit disabling of the ADC and isolation of the input channel from all sources. Terminating continuous conversions this way prevents an additional single conversion from being performed. It is not necessary to set ADCH to all 1s to place the ADC in a low-power state when continuous conversions are not enabled because the module automatically enters a low-power state when a conversion completes. 00000 When DIFF=0, DADP0 is selected as input; when DIFF=1, DAD0 is selected as input. 00001 When DIFF=0, DADP1 is selected as input; when DIFF=1, DAD1 is selected as input. 00010 When DIFF=0, DADP2 is selected as input; when DIFF=1, DAD2 is selected as input. 00011 When DIFF=0, DADP3 is selected as input; when DIFF=1, DAD3 is selected as input. 00100 When DIFF=0, AD4 is selected as input; when DIFF=1, it is reserved. 00101 When DIFF=0, AD5 is selected as input; when DIFF=1, it is reserved. 00110 When DIFF=0, AD6 is selected as input; when DIFF=1, it is reserved. 00111 When DIFF=0, AD7 is selected as input; when DIFF=1, it is reserved.

Table continues on the next page...

ADCx_SC1n field descriptions (continued)

Field	Description	
	01000	When DIFF=0, AD8 is selected as input; when DIFF=1, it is reserved.
	01001	When DIFF=0, AD9 is selected as input; when DIFF=1, it is reserved.
	01010	When DIFF=0, AD10 is selected as input; when DIFF=1, it is reserved.
	01011	When DIFF=0, AD11 is selected as input; when DIFF=1, it is reserved.
	01100	When DIFF=0, AD12 is selected as input; when DIFF=1, it is reserved.
	01101	When DIFF=0, AD13 is selected as input; when DIFF=1, it is reserved.
	01110	When DIFF=0, AD14 is selected as input; when DIFF=1, it is reserved.
	01111	When DIFF=0, AD15 is selected as input; when DIFF=1, it is reserved.
	10000	When DIFF=0, AD16 is selected as input; when DIFF=1, it is reserved.
	10001	When DIFF=0, AD17 is selected as input; when DIFF=1, it is reserved.
	10010	When DIFF=0, AD18 is selected as input; when DIFF=1, it is reserved.
	10011	When DIFF=0, AD19 is selected as input; when DIFF=1, it is reserved.
	10100	When DIFF=0, AD20 is selected as input; when DIFF=1, it is reserved.
	10101	When DIFF=0, AD21 is selected as input; when DIFF=1, it is reserved.
	10110	When DIFF=0, AD22 is selected as input; when DIFF=1, it is reserved.
	10111	When DIFF=0, AD23 is selected as input; when DIFF=1, it is reserved.
	11000	Reserved.
	11001	Reserved.
	11010	When DIFF=0, Temp Sensor (single-ended) is selected as input; when DIFF=1, Temp Sensor (differential) is selected as input.
	11011	When DIFF=0,Bandgap (single-ended) is selected as input; when DIFF=1, Bandgap (differential) is selected as input.
	11100	Reserved.
	11101	When DIFF=0,V_{REFSH} is selected as input; when DIFF=1, -V_{REFSH} (differential) is selected as input. Voltage reference selected is determined by SC2[REFSEL].
	11110	When DIFF=0,V_{REFSL} is selected as input; when DIFF=1, it is reserved. Voltage reference selected is determined by SC2[REFSEL].
	11111	Module is disabled.

40.4.2 ADC Configuration Register 1 (ADCx_CFG1)

The configuration Register 1 (CFG1) selects the mode of operation, clock source, clock divide, and configuration for low power or long sample time.

Address: 4003_B000h base + 8h offset = 4003_B008h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								ADLPC	ADIV		ADLSMP	MODE		ADICLK	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ADCx_CFG1 field descriptions

Field	Description
31–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7 ADLPC	Low-Power Configuration Controls the power configuration of the successive approximation converter. This optimizes power consumption when higher sample rates are not required. 0 Normal power configuration. 1 Low-power configuration. The power is reduced at the expense of maximum clock speed.
6–5 ADIV	Clock Divide Select Selects the divide ratio used by the ADC to generate the internal clock ADCK. 00 The divide ratio is 1 and the clock rate is input clock. 01 The divide ratio is 2 and the clock rate is (input clock)/2. 10 The divide ratio is 4 and the clock rate is (input clock)/4. 11 The divide ratio is 8 and the clock rate is (input clock)/8.
4 ADLSMP	Sample Time Configuration Selects between different sample times based on the conversion mode selected. This field adjusts the sample period to allow higher impedance inputs to be accurately sampled or to maximize conversion speed for lower impedance inputs. Longer sample times can also be used to lower overall power consumption if continuous conversions are enabled and high conversion rates are not required. When ADLSMP=1, the long sample time select bits, (ADLSTS[1:0]), can select the extent of the long sample time.

Table continues on the next page...

ADCx_CFG1 field descriptions (continued)

Field	Description
	0 Short sample time. 1 Long sample time.
3–2 MODE	Conversion mode selection Selects the ADC resolution mode. 00 When DIFF=0:It is single-ended 8-bit conversion; when DIFF=1, it is differential 9-bit conversion with 2's complement output. 01 When DIFF=0:It is single-ended 12-bit conversion ; when DIFF=1, it is differential 13-bit conversion with 2's complement output. 10 When DIFF=0:It is single-ended 10-bit conversion. ; when DIFF=1, it is differential 11-bit conversion with 2's complement output 11 When DIFF=0:It is single-ended 16-bit conversion..; when DIFF=1, it is differential 16-bit conversion with 2's complement output
ADICLK	Input Clock Select Selects the input clock source to generate the internal clock, ADCK. Note that when the ADACK clock source is selected, it is not required to be active prior to conversion start. When it is selected and it is not active prior to a conversion start, when CFG2[ADACKEN]=0, the asynchronous clock is activated at the start of a conversion and deactivated when conversions are terminated. In this case, there is an associated clock startup delay each time the clock source is re-activated. 00 Bus clock 01 Bus clock divided by 2(BUSCLK/2) 10 Alternate clock (ALTCLK) 11 Asynchronous clock (ADACK)

40.4.3 ADC Configuration Register 2 (ADCx_CFG2)

Configuration Register 2 (CFG2) selects the special high-speed configuration for very high speed conversions and selects the long sample time duration during long sample mode.

Address: 4003_B000h base + Ch offset = 4003_B00Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								0			MUXSEL	ADACKEN	ADHSC	ADLSTS	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ADCx_CFG2 field descriptions

Field	Description
31–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7–5 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
4 MUXSEL	ADC Mux Select Changes the ADC mux setting to select between alternate sets of ADC channels. 0 ADxxa channels are selected. 1 ADxxb channels are selected.
3 ADACKEN	Asynchronous Clock Output Enable Enables the asynchronous clock source and the clock source output regardless of the conversion and status of CFG1[ADICLK]. Based on MCU configuration, the asynchronous clock may be used by other modules. See chip configuration information. Setting this field allows the clock to be used even while the ADC is idle or operating from a different clock source. Also, latency of initiating a single or first-continuous conversion with the asynchronous clock selected is reduced because the ADACK clock is already operational. 0 Asynchronous clock output disabled; Asynchronous clock is enabled only if selected by ADICLK and a conversion is active. 1 Asynchronous clock and clock output is enabled regardless of the state of the ADC.
2 ADHSC	High-Speed Configuration Configures the ADC for very high-speed operation. The conversion sequence is altered with 2 ADCK cycles added to the conversion time to allow higher speed conversion clocks. 0 Normal conversion sequence selected. 1 High-speed conversion sequence selected with 2 additional ADCK cycles to total conversion time.
ADLSTS	Long Sample Time Select Selects between the extended sample times when long sample time is selected, that is, when CFG1[ADLSMP]=1. This allows higher impedance inputs to be accurately sampled or to maximize conversion speed for lower impedance inputs. Longer sample times can also be used to lower overall power consumption when continuous conversions are enabled if high conversion rates are not required. 00 Default longest sample time; 20 extra ADCK cycles; 24 ADCK cycles total. 01 12 extra ADCK cycles; 16 ADCK cycles total sample time. 10 6 extra ADCK cycles; 10 ADCK cycles total sample time. 11 2 extra ADCK cycles; 6 ADCK cycles total sample time.

40.4.4 ADC Data Result Register (ADC*x*_R*n*)

The data result registers (Rn) contain the result of an ADC conversion of the channel selected by the corresponding status and channel control register (SC1A:SC1n). For every status and channel control register, there is a corresponding data result register.

Unused bits in R n are cleared in unsigned right-aligned modes and carry the sign bit (MSB) in sign-extended 2's complement modes. For example, when configured for 10-bit single-ended mode, D[15:10] are cleared. When configured for 11-bit differential mode, D[15:10] carry the sign bit, that is, bit 10 extended through bit 15.

The following table describes the behavior of the data result registers in the different modes of operation.

Table 40-2. Data result register description

Conversion mode	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	Format
16-bit differential	S	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	Signed 2's complement
16-bit single-ended	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	Unsigned right justified
13-bit differential	S	S	S	S	D	D	D	D	D	D	D	D	D	D	D	D	Sign-extended 2's complement
12-bit single-ended	0	0	0	0	D	D	D	D	D	D	D	D	D	D	D	D	Unsigned right-justified
11-bit differential	S	S	S	S	S	S	D	D	D	D	D	D	D	D	D	D	Sign-extended 2's complement
10-bit single-ended	0	0	0	0	0	0	D	D	D	D	D	D	D	D	D	D	Unsigned right-justified
9-bit differential	S	S	S	S	S	S	S	S	D	D	D	D	D	D	D	D	Sign-extended 2's complement
8-bit single-ended	0	0	0	0	0	0	0	0	D	D	D	D	D	D	D	D	Unsigned right-justified

NOTE

S: Sign bit or sign bit extension;

D: Data, which is 2's complement data if indicated

Address: 4003_B000h base + 10h offset + (4d × i), where i=0d to 1d

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

ADCx_R*n* field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
D	Data result

40.4.5 Compare Value Registers (ADCx_CV*n*)

The Compare Value Registers (CV1 and CV2) contain a compare value used to compare the conversion result when the compare function is enabled, that is, SC2[ACFE]=1. This register is formatted in the same way as the Rn registers in different modes of operation for both bit position definition and value format using unsigned or sign-extended 2's complement. Therefore, the compare function uses only the CVn fields that are related to the ADC mode of operation.

The compare value 2 register (CV2) is used only when the compare range function is enabled, that is, SC2[ACREN]=1.

Address: 4003_B000h base + 18h offset + (4d × i), where i=0d to 1d

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

ADCx_CV*n* field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
CV	Compare Value.

40.4.6 Status and Control Register 2 (ADCx_SC2)

The status and control register 2 (SC2) contains the conversion active, hardware/software trigger select, compare function, and voltage reference select of the ADC module.

Address: 4003_B000h base + 20h offset = 4003_B020h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								ADACT	ADTRG	ACFE	ACFGT	ACREN	DMAEN	REFSEL	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ADCx_SC2 field descriptions

Field	Description
31–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7 ADACT	Conversion Active Indicates that a conversion or hardware averaging is in progress. ADACT is set when a conversion is initiated and cleared when a conversion is completed or aborted. 0 Conversion not in progress. 1 Conversion in progress.
6 ADTRG	Conversion Trigger Select Selects the type of trigger used for initiating a conversion. Two types of trigger are selectable:

Table continues on the next page...

ADCx_SC2 field descriptions (continued)

Field	Description
	- Software trigger: When software trigger is selected, a conversion is initiated following a write to SC1A. - Hardware trigger: When hardware trigger is selected, a conversion is initiated following the assertion of the ADHWT input after a pulse of the ADHWTSn input. 0 Software trigger selected. 1 Hardware trigger selected.
5 ACFE	Compare Function Enable Enables the compare function. 0 Compare function disabled. 1 Compare function enabled.
4 ACFGT	Compare Function Greater Than Enable Configures the compare function to check the conversion result relative to the CV1 and CV2 based upon the value of ACREN. ACFE must be set for ACFGT to have any effect. 0 Configures less than threshold, outside range not inclusive and inside range not inclusive; functionality based on the values placed in CV1 and CV2. 1 Configures greater than or equal to threshold, outside and inside ranges inclusive; functionality based on the values placed in CV1 and CV2.
3 ACREN	Compare Function Range Enable Configures the compare function to check if the conversion result of the input being monitored is either between or outside the range formed by CV1 and CV2 determined by the value of ACFGT. ACFE must be set for ACFGT to have any effect. 0 Range function disabled. Only CV1 is compared. 1 Range function enabled. Both CV1 and CV2 are compared.
2 DMAEN	DMA Enable 0 DMA is disabled. 1 DMA is enabled and will assert the ADC DMA request during an ADC conversion complete event noted when any of the SC1n[COCO] flags is asserted.
REFSEL	Voltage Reference Selection Selects the voltage reference source used for conversions. 00 Default voltage reference pin pair, that is, external pins V_{REFH} and V_{REFL} 01 Alternate reference pair, that is, V_{ALTH} and V_{ALTL} . This pair may be additional external pins or internal sources depending on the MCU configuration. See the chip configuration information for details specific to this MCU 10 Reserved 11 Reserved

40.4.7 Status and Control Register 3 (ADCx_SC3)

The Status and Control Register 3 (SC3) controls the calibration, continuous convert, and hardware averaging functions of the ADC module.

Address: 4003_B000h base + 24h offset = 4003_B024h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								CAL	CALF	0		ADCO	AVGE	AVGS	
W										w1c						
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ADCx_SC3 field descriptions

Field	Description
31–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7 CAL	Calibration Begins the calibration sequence when set. This field stays set while the calibration is in progress and is cleared when the calibration sequence is completed. CALF must be checked to determine the result of the calibration sequence. Once started, the calibration routine cannot be interrupted by writes to the ADC registers or the results will be invalid and CALF will set. Setting CAL will abort any current conversion.
6 CALF	Calibration Failed Flag Displays the result of the calibration sequence. The calibration sequence will fail if SC2[ADTRG] = 1, any ADC register is written, or any stop mode is entered before the calibration sequence completes. Writing 1 to CALF clears it. 0 Calibration completed normally. 1 Calibration failed. ADC accuracy specifications are not guaranteed.

Table continues on the next page...

ADCx_SC3 field descriptions (continued)

Field	Description
5–4 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
3 ADCO	Continuous Conversion Enable Enables continuous conversions. 0 One conversion or one set of conversions if the hardware average function is enabled, that is, AVGE=1, after initiating a conversion. 1 Continuous conversions or sets of conversions if the hardware average function is enabled, that is, AVGE=1, after initiating a conversion.
2 AVGE	Hardware Average Enable Enables the hardware average function of the ADC. 0 Hardware average function disabled. 1 Hardware average function enabled.
AVGS	Hardware Average Select Determines how many ADC conversions will be averaged to create the ADC average result. 00 4 samples averaged. 01 8 samples averaged. 10 16 samples averaged. 11 32 samples averaged.

40.4.8 ADC Offset Correction Register (ADCx_OFS)

The ADC Offset Correction Register (OFS) contains the user-selected or calibration-generated offset error correction value. This register is a 2's complement, left-justified, 16-bit value . The value in OFS is subtracted from the conversion and the result is transferred into the result registers, Rn. If the result is greater than the maximum or less than the minimum result value, it is forced to the appropriate limit for the current mode of operation.

For more information regarding the calibration procedure, please refer to the Calibration function section.

Address: 4003_B000h base + 28h offset = 4003_B028h

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

ADCx_OFS field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
OFS	Offset Error Correction Value

40.4.9 ADC Plus-Side Gain Register (ADCx_PG)

The Plus-Side Gain Register (PG) contains the gain error correction for the plus-side input in differential mode or the overall conversion in single-ended mode. PG, a 16-bit real number in binary format, is the gain adjustment factor, with the radix point fixed between PG[15] and PG[14]. This register must be written by the user with the value described in the calibration procedure. Otherwise, the gain error specifications may not be met.

For more information regarding the calibration procedure, please refer to the Calibration function section.

Address: 4003_B000h base + 2Ch offset = 4003_B02Ch

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

ADCx_PG field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
PG	Plus-Side Gain

40.4.10 ADC Minus-Side Gain Register (ADCx_MG)

The Minus-Side Gain Register (MG) contains the gain error correction for the minus-side input in differential mode. This register is ignored in single-ended mode. MG, a 16-bit real number in binary format, is the gain adjustment factor, with the radix point fixed between MG[15] and MG[14]. This register must be written by the user with the value described in the calibration procedure. Otherwise, the gain error specifications may not be met.

For more information regarding the calibration procedure, please refer to the Calibration function section.

Address: 4003_B000h base + 30h offset = 4003_B030h

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

ADCx_MG field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
MG	Minus-Side Gain

40.4.11 ADC Plus-Side General Calibration Value Register (ADCx_CLPD)

The Plus-Side General Calibration Value Registers (CLPx) contain calibration information that is generated by the calibration function. These registers contain seven calibration values of varying widths: CLP0[5:0], CLP1[6:0], CLP2[7:0], CLP3[8:0], CLP4[9:0], CLPS[5:0], and CLPD[5:0]. CLPx are automatically set when the self-calibration sequence is done, that is, CAL is cleared. If these registers are written by the user after calibration, the linearity error specifications may not be met.

For more information regarding the calibration procedure, please refer to the Calibration function section.

Address: 4003_B000h base + 34h offset = 4003_B034h

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

ADCx_CLPD field descriptions

Field	Description
31–6 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
CLPD	Calibration Value Calibration Value

40.4.12 ADC Plus-Side General Calibration Value Register (ADCx_CLPS)

For more information, see CLPD register description.

Address: 4003_B000h base + 38h offset = 4003_B038h

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

ADCx_CLPS field descriptions

Field	Description
31–6 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
CLPS	Calibration Value Calibration Value

40.4.13 ADC Plus-Side General Calibration Value Register (ADCx_CLP4)

For more information, see CLPD register description.

Address: 4003_B000h base + 3Ch offset = 4003_B03Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																						CLP4									
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0

ADCx_CLP4 field descriptions

Field	Description
31–10 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
CLP4	Calibration Value Calibration Value

40.4.14 ADC Plus-Side General Calibration Value Register (ADCx_CLP3)

For more information, see CLPD register description.

Address: 4003_B000h base + 40h offset = 4003_B040h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																							CLP3								
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0

ADCx_CLP3 field descriptions

Field	Description
31–9 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
CLP3	Calibration Value Calibration Value

40.4.15 ADC Plus-Side General Calibration Value Register (ADCx_CLP2)

For more information, see CLPD register description.

Address: 4003_B000h base + 44h offset = 4003_B044h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																								CLP2							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0

ADCx_CLP2 field descriptions

Field	Description
31–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
CLP2	Calibration Value Calibration Value

40.4.16 ADC Plus-Side General Calibration Value Register (ADCx_CLP1)

For more information, see CLPD register description.

Address: 4003_B000h base + 48h offset = 4003_B048h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																									CLP1						
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0

ADCx_CLP1 field descriptions

Field	Description
31–7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
CLP1	Calibration Value Calibration Value

40.4.17 ADC Plus-Side General Calibration Value Register (ADCx_CLP0)

For more information, see CLPD register description.

Address: 4003_B000h base + 4Ch offset = 4003_B04Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																										CLP0					
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0

ADCx_CLP0 field descriptions

Field	Description
31–6 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
CLP0	Calibration Value Calibration Value

40.4.18 ADC Minus-Side General Calibration Value Register (ADCx_CLMD)

The Minus-Side General Calibration Value (CLMx) registers contain calibration information that is generated by the calibration function. These registers contain seven calibration values of varying widths: CLM0[5:0], CLM1[6:0], CLM2[7:0], CLM3[8:0], CLM4[9:0], CLMS[5:0], and CLMD[5:0]. CLMx are automatically set when the self-calibration sequence is done, that is, CAL is cleared. If these registers are written by the user after calibration, the linearity error specifications may not be met.

For more information regarding the calibration procedure, please refer to the Calibration function section.

Address: 4003_B000h base + 54h offset = 4003_B054h

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

ADCx_CLMD field descriptions

Field	Description
31–6 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
CLMD	Calibration Value Calibration Value

40.4.19 ADC Minus-Side General Calibration Value Register (ADCx_CLMS)

For more information, see CLMD register description.

Address: 4003_B000h base + 58h offset = 4003_B058h

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

ADCx_CLMS field descriptions

Field	Description
31–6 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
CLMS	Calibration Value Calibration Value

40.4.20 ADC Minus-Side General Calibration Value Register (ADCx_CLM4)

For more information, see CLMD register description.

Address: 4003_B000h base + 5Ch offset = 4003_B05Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																						CLM4									
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0

ADCx_CLM4 field descriptions

Field	Description
31–10 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
CLM4	Calibration Value Calibration Value

40.4.21 ADC Minus-Side General Calibration Value Register (ADCx_CLM3)

For more information, see CLMD register description.

Address: 4003_B000h base + 60h offset = 4003_B060h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																							CLM3								
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0

ADCx_CLM3 field descriptions

Field	Description
31–9 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

Table continues on the next page...

ADCx_CLM3 field descriptions (continued)

Field	Description
CLM3	Calibration Value Calibration Value

40.4.22 ADC Minus-Side General Calibration Value Register (ADCx_CLM2)

For more information, see CLMD register description.

Address: 4003_B000h base + 64h offset = 4003_B064h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																								CLM2							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0

ADCx_CLM2 field descriptions

Field	Description
31–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
CLM2	Calibration Value Calibration Value

40.4.23 ADC Minus-Side General Calibration Value Register (ADCx_CLM1)

For more information, see CLMD register description.

Address: 4003_B000h base + 68h offset = 4003_B068h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																									CLM1						
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0

ADCx_CLM1 field descriptions

Field	Description
31–7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
CLM1	Calibration Value Calibration Value

40.4.24 ADC Minus-Side General Calibration Value Register (ADCx_CLM0)

For more information, see CLMD register description.

Address: 4003_B000h base + 6Ch offset = 4003_B06Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																										CLM0					
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0

ADCx_CLM0 field descriptions

Field	Description
31–6 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
CLM0	Calibration Value Calibration Value

40.5 Functional description

The ADC module is disabled during reset, in Low-Power Stop mode, or when SC1n[ADCH] are all high; see the power management information for details. The module is idle when a conversion has completed and another conversion has not been initiated. When it is idle and the asynchronous clock output enable is disabled, or CFG2[ADACKEN]= 0, the module is in its lowest power state. The ADC can perform an analog-to-digital conversion on any of the software selectable channels. All modes perform conversion by a successive approximation algorithm.

To meet accuracy specifications, the ADC module must be calibrated using the on-chip calibration function.

See Calibration function for details on how to perform calibration.

When the conversion is completed, the result is placed in the Rn data registers. The respective SC1n[COCO] is then set and an interrupt is generated if the respective conversion complete interrupt has been enabled, or, when SC1n[AIEN]=1.

The ADC module has the capability of automatically comparing the result of a conversion with the contents of the CV1 and CV2 registers. The compare function is enabled by setting SC2[ACFE] and operates in any of the conversion modes and configurations.

The ADC module has the capability of automatically averaging the result of multiple conversions. The hardware average function is enabled by setting SC3[AVGE] and operates in any of the conversion modes and configurations.

NOTE

> For the chip specific modes of operation, see the power management information of this MCU.

40.5.1 Clock select and divide control

One of four clock sources can be selected as the clock source for the ADC module.

This clock source is then divided by a configurable value to generate the input clock ADCK, to the module. The clock is selected from one of the following sources by means of CFG1[ADICLK].

- Bus clock. This is the default selection following reset.
- Bus clock divided by two. For higher bus clock rates, this allows a maximum divide-by-16 of the bus clock using CFG1[ADIV].
- ALTCLK: As defined for this MCU. See the chip configuration information. Conversions are possible using ALTCLK as the input clock source while the MCU is in Normal Stop mode.
- Asynchronous clock (ADACK): This clock is generated from a clock source within the ADC module. When the ADACK clock source is selected, it is not required to be active prior to conversion start. When it is selected and it is not active prior to a conversion start CFG2[ADACKEN]=0, ADACK is activated at the start of a conversion and deactivated when conversions are terminated. In this case, there is an associated clock startup delay each time the clock source is re-activated. To avoid the conversion time variability and latency associated with the ADACK clock startup, set CFG2[ADACKEN]=1 and wait the worst-case startup time of 5 µs prior to initiating any conversions using the ADACK clock source. Conversions are possible using ADACK as the input clock source while the MCU is in Normal Stop mode. See Power Control for more information.

Whichever clock is selected, its frequency must fall within the specified frequency range for ADCK. If the available clocks are too slow, the ADC may not perform according to specifications. If the available clocks are too fast, the clock must be divided to the appropriate frequency. This divider is specified by CFG1[ADIV] and can be divide-by 1, 2, 4, or 8.

40.5.2 Voltage reference selection

The ADC can be configured to accept one of the two voltage reference pairs as the reference voltage (V_{REFSH} and V_{REFSL}) used for conversions.

Each pair contains a positive reference that must be between the minimum Ref Voltage High and V_{DDA}, and a ground reference that must be at the same potential as V_{SSA}. The two pairs are external (V_{REFH} and V_{REFL}) and alternate (V_{ALTH} and V_{ALTL}). These voltage references are selected using SC2[REFSEL]. The alternate (V_{ALTH} and V_{ALTL}) voltage reference pair may select additional external pins or internal sources depending on MCU configuration. See the chip configuration information on the voltage references specific to this MCU.

40.5.3 Hardware trigger and channel selects

The ADC module has a selectable asynchronous hardware conversion trigger, ADHWT, that is enabled when SC2[ADTRG] is set and a hardware trigger select event, ADHWTSn, has occurred.

This source is not available on all MCUs. See the chip-specific ADC information for information on the ADHWT source and the ADHWTSn configurations specific to this MCU.

When an ADHWT source is available and hardware trigger is enabled, that is SC2[ADTRG]=1, a conversion is initiated on the rising-edge of ADHWT after a hardware trigger select event, that is, ADHWTSn, has occurred. If a conversion is in progress when a rising-edge of a trigger occurs, the rising-edge is ignored. In continuous convert configuration, only the initial rising-edge to launch continuous conversions is observed, and until conversion is aborted, the ADC continues to do conversions on the same SCn register that initiated the conversion. The hardware trigger function operates in conjunction with any of the conversion modes and configurations.

The hardware trigger select event, ADHWTSn, must be set prior to the receipt of the ADHWT signal. If these conditions are not met, the converter may ignore the trigger or use the incorrect configuration. If a hardware trigger select event is asserted during a conversion, it must stay asserted until the end of current conversion and remain set until the receipt of the ADHWT signal to trigger a new conversion. The channel and status fields selected for the conversion depend on the active trigger select signal:

- ADHWTSA active selects SC1A.
- ADHWTSn active selects SC1n.

Note

> Asserting more than one hardware trigger select signal (ADHWTSn) at the same time results in unknown results. To avoid this, select only one hardware trigger select signal (ADHWTSn) prior to the next intended conversion.

When the conversion is completed, the result is placed in the Rn registers associated with the ADHWTSn received. For example:

- ADHWTSA active selects RA register
- ADHWTSn active selects Rn register

The conversion complete flag associated with the ADHWTSn received, that is, SC1n[COCO], is then set and an interrupt is generated if the respective conversion complete interrupt has been enabled, that is, SC1[AIEN]=1.

40.5.4 Conversion control

Conversions can be performed as determined by CFG1[MODE] and SC1n[DIFF] as shown in the description of CFG1[MODE].

Conversions can be initiated by a software or hardware trigger.

In addition, the ADC module can be configured for:

- Low-power operation
- Long sample time
- Continuous conversion
- Hardware average
- Automatic compare of the conversion result to a software determined compare value

40.5.4.1 Initiating conversions

A conversion is initiated:

- Following a write to SC1A, with SC1n[ADCH] not all 1's, if software triggered operation is selected, that is, when SC2[ADTRG]=0.
- Following a hardware trigger, or ADHWT event, if hardware triggered operation is selected, that is, SC2[ADTRG]=1, and a hardware trigger select event, ADHWTSn, has occurred. The channel and status fields selected depend on the active trigger select signal:
  - ADHWTSA active selects SC1A.

- ADHWTSn active selects SC1n.
- if neither is active, the off condition is selected

Note

> Selecting more than one ADHWTSn prior to a conversion completion will result in unknown results. To avoid this, select only one ADHWTSn prior to a conversion completion.

- Following the transfer of the result to the data registers when continuous conversion is enabled, that is, when SC3[ADCO] = 1.

If continuous conversions are enabled, a new conversion is automatically initiated after the completion of the current conversion. In software triggered operation, that is, when SC2[ADTRG] = 0, continuous conversions begin after SC1A is written and continue until aborted. In hardware triggered operation, that is, when SC2[ADTRG] = 1 and one ADHWTSn event has occurred, continuous conversions begin after a hardware trigger event and continue until aborted.

If hardware averaging is enabled, a new conversion is automatically initiated after the completion of the current conversion until the correct number of conversions are completed. In software triggered operation, conversions begin after SC1A is written. In hardware triggered operation, conversions begin after a hardware trigger. If continuous conversions are also enabled, a new set of conversions to be averaged are initiated following the last of the selected number of conversions.

40.5.4.2 Completing conversions

A conversion is completed when the result of the conversion is transferred into the data result registers, Rn. If the compare functions are disabled, this is indicated by setting of SC1n[COCO]. If hardware averaging is enabled, the respective SC1n[COCO] sets only if the last of the selected number of conversions is completed. If the compare function is enabled, the respective SC1n[COCO] sets and conversion result data is transferred only if the compare condition is true. If both hardware averaging and compare functions are enabled, then the respective SC1n[COCO] sets only if the last of the selected number of conversions is completed and the compare condition is true. An interrupt is generated if the respective SC1n[AIEN] is high at the time that the respective SC1n[COCO] is set.

40.5.4.3 Aborting conversions

Any conversion in progress is aborted when:

- Writing to SC1A while it is actively controlling a conversion, aborts the current conversion. In Software Trigger mode, when SC2[ADTRG]=0, a write to SC1A initiates a new conversion if SC1A[ADCH] is equal to a value other than all 1s. Writing to any of the SC1B–SC1n registers while that specific SC1B–SC1n register is actively controlling a conversion aborts the current conversion. The SC1(B-n) registers are not used for software trigger operation and therefore writes to the SC1(B-n) registers do not initiate a new conversion.
- A write to any ADC register besides the SC1A-SC1n registers occurs. This indicates that a change in mode of operation has occurred and the current conversion is therefore invalid.
- The MCU is reset or enters Low-Power Stop modes.
- The MCU enters Normal Stop mode with ADACK or Alternate Clock Sources not enabled.

When a conversion is aborted, the contents of the data registers, Rn, are not altered. The data registers continue to be the values transferred after the completion of the last successful conversion. If the conversion was aborted by a reset or Low-Power Stop modes, RA and Rn return to their reset states.

40.5.4.4 Power control

The ADC module remains in its idle state until a conversion is initiated. If ADACK is selected as the conversion clock source, but the asynchronous clock output is disabled, that is CFG2[ADACKEN]=0, the ADACK clock generator also remains in its idle state (disabled) until a conversion is initiated. If the asynchronous clock output is enabled, that is, CFG2[ADACKEN]=1, it remains active regardless of the state of the ADC or the MCU power mode.

Power consumption when the ADC is active can be reduced by setting CFG1[ADLPC]. This results in a lower maximum value for f_{ADCK}.

40.5.4.5 Sample time and total conversion time

For short sample, that is, when CFG1[ADLSMP]=0, there is a 2-cycle adder for first conversion over the base sample time of four ADCK cycles. For high-speed conversions, that is, when CFG2[ADHSC]=1, there is an additional 2-cycle adder on any conversion. The table below summarizes sample times for the possible ADC configurations.

ADC configuration			Sample time (ADCK cycles)	
CFG1[ADLSMP]	CFG2[ADLSTS]	CFG2[ADHSC]	First or Single	Subsequent
0	X	0	6	4
1	00	0	24	
1	01	0	16	
1	10	0	10	
1	11	0	6	
0	X	1	8	6
1	00	1	26	
1	01	1	18	
1	10	1	12	
1	11	1	8	

The total conversion time depends upon:

- The sample time as determined by CFG1[ADLSMP] and CFG2[ADLSTS]
- The MCU bus frequency
- The conversion mode, as determined by CFG1[MODE] and SC1n[DIFF]
- The high-speed configuration, that is, CFG2[ADHSC]
- The frequency of the conversion clock, that is, f_{ADCK}.

CFG2[ADHSC] is used to configure a higher clock input frequency. This will allow faster overall conversion times. To meet internal ADC timing requirements, CFG2[ADHSC] adds additional ADCK cycles. Conversions with CFG2[ADHSC]=1 take two more ADCK cycles. CFG2[ADHSC] must be used when the ADCLK exceeds the limit for CFG2[ADHSC]=0.

After the module becomes active, sampling of the input begins.

1. CFG1[ADLSMP] and CFG2[ADLSTS] select between sample times based on the conversion mode that is selected.
2. When sampling is completed, the converter is isolated from the input channel and a successive approximation algorithm is applied to determine the digital value of the analog signal.
3. The result of the conversion is transferred to Rn upon completion of the conversion algorithm.

If the bus frequency is less than f_{ADCK}, precise sample time for continuous conversions cannot be guaranteed when short sample is enabled, that is, when CFG1[ADLSMP]=0.

The maximum total conversion time is determined by the clock source chosen and the divide ratio selected. The clock source is selectable by CFG1[ADICLK], and the divide ratio is specified by CFG1[ADIV].

The maximum total conversion time for all configurations is summarized in the equation below. See the following tables for the variables referenced in the equation.

$$\mathrm{ConversionTime} = \mathrm{SFCAdder} + \mathrm{AverageNum} \times \left(\mathrm{BCT} + \mathrm{LSTAdder} + \mathrm{HSCAdder}\right)$$

Equation 1. Conversion time equation

Table 40-3. Single or first continuous time adder (SFCAdder)

CFG1[ADLSMP]	CFG2[ADACKEN]	CFG1[ADICLK]	Single or first continuous time adder (SFCAdder)
1	x	0x, 10	3 ADCK cycles + 5 bus clock cycles
1	1	11	3 ADCK cycles + 5 bus clock cycles[1]
1	0	11	5 µs + 3 ADCK cycles + 5 bus clock cycles
0	x	0x, 10	5 ADCK cycles + 5 bus clock cycles
0	1	11	5 ADCK cycles + 5 bus clock cycles[1]
0	0	11	5 µs + 5 ADCK cycles + 5 bus clock cycles

[1] To achieve this time, CFG2[ADACKEN] must be 1 for at least 5 µs prior to the conversion is initiated.

Table 40-4. Average number factor (AverageNum)

SC3[AVGE]	SC3[AVGS]	Average number factor (AverageNum)
0	xx	1
1	00	4
1	01	8
1	10	16
1	11	32

Table 40-5. Base conversion time (BCT)

Mode	Base conversion time (BCT)
8b single-ended	17 ADCK cycles
9b differential	27 ADCK cycles
10b single-ended	20 ADCK cycles
11b differential	30 ADCK cycles
12b single-ended	20 ADCK cycles
13b differential	30 ADCK cycles

Table continues on the next page...

Table 40-5. Base conversion time (BCT) (continued)

Mode	Base conversion time (BCT)
16b single-ended	25 ADCK cycles
16b differential	34 ADCK cycles

Table 40-6. Long sample time adder (LSTAdder)

CFG1[ADLSMP]	CFG2[ADLSTS]	Long sample time adder (LSTAdder)
0	xx	0 ADCK cycles
1	00	20 ADCK cycles
1	01	12 ADCK cycles
1	10	6 ADCK cycles
1	11	2 ADCK cycles

Table 40-7. High-speed conversion time adder (HSCAdder)

CFG2[ADHSC]	High-speed conversion time adder (HSCAdder)
0	0 ADCK cycles
1	2 ADCK cycles

Note

The ADCK frequency must be between f_{ADCK} minimum and f_{ADCK} maximum to meet ADC specifications.

40.5.4.6 Conversion time examples

The following examples use the Equation 1 on page 894, and the information provided in Table 40-3 through Table 40-7.

40.5.4.6.1 Typical conversion time configuration

A typical configuration for ADC conversion is:

- 10-bit mode, with the bus clock selected as the input clock source
- The input clock divide-by-1 ratio selected
- Bus frequency of 8 MHz
- Long sample time disabled
- High-speed conversion disabled

The conversion time for a single conversion is calculated by using the Equation 1 on page 894, and the information provided in Table 40-3 through Table 40-7. The table below lists the variables of Equation 1 on page 894.

Table 40-8. Typical conversion time

Variable	Time
SFCAdder	5 ADCK cycles + 5 bus clock cycles
AverageNum	1
BCT	20 ADCK cycles
LSTAdder	0
HSCAdder	0

The resulting conversion time is generated using the parameters listed in the preceding table. Therefore, for a bus clock and an ADCK frequency equal to 8 MHz, the resulting conversion time is 3.75 µs.

40.5.4.6.2 Long conversion time configuration

A configuration for long ADC conversion is:

- 16-bit differential mode with the bus clock selected as the input clock source
- The input clock divide-by-8 ratio selected
- Bus frequency of 8 MHz
- Long sample time enabled
- Configured for longest adder
- High-speed conversion disabled
- Average enabled for 32 conversions

The conversion time for this conversion is calculated by using the Equation 1 on page 894, and the information provided in Table 40-3 through Table 40-7. The following table lists the variables of the Equation 1 on page 894.

Table 40-9. Typical conversion time

Variable	Time
SFCAdder	3 ADCK cycles + 5 bus clock cycles
AverageNum	32
BCT	34 ADCK cycles
LSTAdder	20 ADCK cycles
HSCAdder	0

The resulting conversion time is generated using the parameters listed in the preceding table. Therefore, for bus clock equal to 8 MHz and ADCK equal to 1 MHz, the resulting conversion time is 57.625 µs, that is, AverageNum. This results in a total conversion time of 1.844 ms.

40.5.4.6.3 Short conversion time configuration

A configuration for short ADC conversion is:

- 8-bit Single-Ended mode with the bus clock selected as the input clock source
- The input clock divide-by-1 ratio selected
- Bus frequency of 20 MHz
- Long sample time disabled
- High-speed conversion enabled

The conversion time for this conversion is calculated by using the Equation 1 on page 894, and the information provided in Table 40-3 through Table 40-7. The table below lists the variables of Equation 1 on page 894.

Table 40-10. Typical conversion time

Variable	Time
SFCAdder	5 ADCK cycles + 5 bus clock cycles
AverageNum	1
BCT	17 ADCK cycles
LSTAdder	0 ADCK cycles
HSCAdder	2

The resulting conversion time is generated using the parameters listed in in the preceding table. Therefore, for bus clock and ADCK frequency equal to 20 MHz, the resulting conversion time is 1.45 µs.

40.5.4.7 Hardware average function

The hardware average function can be enabled by setting SC3[AVGE]=1 to perform a hardware average of multiple conversions. The number of conversions is determined by the AVGS[1:0] bits, which can select 4, 8, 16, or 32 conversions to be averaged. While the hardware average function is in progress, SC2[ADACT] will be set.

After the selected input is sampled and converted, the result is placed in an accumulator from which an average is calculated once the selected number of conversions have been completed. When hardware averaging is selected, the completion of a single conversion will not set SC1n[COCO].

If the compare function is either disabled or evaluates true, after the selected number of conversions are completed, the average conversion result is transferred into the data result registers, Rn, and SC1n[COCO] is set. An ADC interrupt is generated upon the setting of SC1n[COCO] if the respective ADC interrupt is enabled, that is, SC1n[AIEN]=1.

Note

> The hardware average function can perform conversions on a channel while the MCU is in Wait or Normal Stop modes. The ADC interrupt wakes the MCU when the hardware average is completed if SC1n[AIEN] is set.

40.5.5 Automatic compare function

The compare function can be configured to check whether the result is less than or greater-than-or-equal-to a single compare value, or, if the result falls within or outside a range determined by two compare values.

The compare mode is determined by SC2[ACFGT], SC2[ACREN], and the values in the compare value registers, CV1 and CV2. After the input is sampled and converted, the compare values in CV1 and CV2 are used as described in the following table. There are six Compare modes as shown in the following table.

Table 40-11. Compare modes

SC2[ACFGT]	SC2[ACREN]	ADCCV1 relative to ADCCV2	Function	Compare mode description
0	0	—	Less than threshold	Compare true if the result is less than the CV1 registers.
1	0	—	Greater than or equal to threshold	Compare true if the result is greater than or equal to CV1 registers.
0	1	Less than or equal	Outside range, not inclusive	Compare true if the result is less than CV1 **Or** the result is greater than CV2.
0	1	Greater than	Inside range, not inclusive	Compare true if the result is less than CV1 **And** the result is greater than CV2.
1	1	Less than or equal	Inside range, inclusive	Compare true if the result is greater than or equal to CV1 **And** the result is less than or equal to CV2.
1	1	Greater than	Outside range, inclusive	Compare true if the result is greater than or equal to CV1 **Or** the result is less than or equal to CV2.

With SC2[ACREN] =1, and if the value of CV1 is less than or equal to the value of CV2, then setting SC2[ACFGT] will select a trigger-if-inside-compare-range inclusive-of-endpoints function. Clearing SC2[ACFGT] will select a trigger-if-outside-compare-range, not-inclusive-of-endpoints function.

If CV1 is greater than CV2, setting SC2[ACFGT] will select a trigger-if-outside-compare-range, inclusive-of-endpoints function. Clearing SC2[ACFGT] will select a trigger-if-inside-compare-range, not-inclusive-of-endpoints function.

If the condition selected evaluates true, SC1n[COCO] is set.

Upon completion of a conversion while the compare function is enabled, if the compare condition is not true, SC1n[COCO] is not set and the conversion result data will not be transferred to the result register, Rn. If the hardware averaging function is enabled, the compare function compares the averaged result to the compare values. The same compare function definitions apply. An ADC interrupt is generated when SC1n[COCO] is set and the respective ADC interrupt is enabled, that is, SC1n[AIEN]=1.

Note

> The compare function can monitor the voltage on a channel while the MCU is in Wait or Normal Stop modes. The ADC interrupt wakes the MCU when the compare condition is met.

40.5.6 Calibration function

The ADC contains a self-calibration function that is required to achieve the specified accuracy.

Calibration must be run, or valid calibration values written, after any reset and before a conversion is initiated. The calibration function sets the offset calibration value, the minus-side calibration values, and the plus-side calibration values. The offset calibration value is automatically stored in the ADC offset correction register (OFS), and the plus-side and minus-side calibration values are automatically stored in the ADC plus-side and minus-side calibration registers, CLPx and CLMx. The user must configure the ADC correctly prior to calibration, and must generate the plus-side and minus-side gain calibration results and store them in the ADC plus-side gain register (PG) after the calibration function completes.

Prior to calibration, the user must configure the ADC's clock source and frequency, low power configuration, voltage reference selection, sample time, and high speed configuration according to the application's clock source availability and needs. If the application uses the ADC in a wide variety of configurations, the configuration for which the highest accuracy is required should be selected, or multiple calibrations can be done for the different configurations. For best calibration results:

- Set hardware averaging to maximum, that is, SC3[AVGE]=1 and SC3[AVGS]=11 for an average of 32
- Set ADC clock frequency f_{ADCK} less than or equal to 4 MHz
- $V_{REFH}=V_{DDA}$
- Calibrate at nominal voltage and temperature

The input channel, conversion mode continuous function, compare function, resolution mode, and differential/single-ended mode are all ignored during the calibration function.

To initiate calibration, the user sets SC3[CAL] and the calibration will automatically begin if the SC2[ADTRG] is 0. If SC2[ADTRG] is 1, SC3[CAL] will not get set and SC3[CALF] will be set. While calibration is active, no ADC register can be written and no stop mode may be entered, or the calibration routine will be aborted causing SC3[CAL] to clear and SC3[CALF] to set. At the end of a calibration sequence, SC1n[COCO] will be set. SC1n[AIEN] can be used to allow an interrupt to occur at the end of a calibration sequence. At the end of the calibration routine, if SC3[CALF] is not set, the automatic calibration routine is completed successfully.

To complete calibration, the user must generate the gain calibration values using the following procedure:

1. Initialize or clear a 16-bit variable in RAM.
2. Add the plus-side calibration results CLP0, CLP1, CLP2, CLP3, CLP4, and CLPS to the variable.
3. Divide the variable by two.
4. Set the MSB of the variable.
5. The previous two steps can be achieved by setting the carry bit, rotating to the right through the carry bit on the high byte and again on the low byte.
6. Store the value in the plus-side gain calibration register PG.
7. Repeat the procedure for the minus-side gain calibration value.

When calibration is complete, the user may reconfigure and use the ADC as desired. A second calibration may also be performed, if desired, by clearing and again setting SC3[CAL].

Overall, the calibration routine may take as many as 14k ADCK cycles and 100 bus cycles, depending on the results and the clock source chosen. For an 8 MHz clock source, this length amounts to about 1.7 ms. To reduce this latency, the calibration values, which are offset, plus-side and minus-side gain, and plus-side and minus-side calibration values, may be stored in flash memory after an initial calibration and recovered prior to the first ADC conversion. This method can reduce the calibration latency to 20 register store operations on all subsequent power, reset, or Low-Power Stop mode recoveries.

Further information on the calibration procedure can be found in the Calibration section of AN3949: ADC16 Calibration Procedure and Programmable Delay Block Synchronization.

40.5.7 User-defined offset function

OFS contains the user-selected or calibration-generated offset error correction value.

This register is a 2's complement, left-justified. The value in OFS is subtracted from the conversion and the result is transferred into the result registers, Rn. If the result is greater than the maximum or less than the minimum result value, it is forced to the appropriate limit for the current mode of operation.

The formatting of the OFS is different from the data result register, Rn, to preserve the resolution of the calibration value regardless of the conversion mode selected. Lower order bits are ignored in lower resolution modes. For example, in 8-bit single-ended mode, OFS[14:7] are subtracted from D[7:0]; OFS[15] indicates the sign (negative numbers are effectively added to the result) and OFS[6:0] are ignored. The same bits are used in 9-bit differential mode because OFS[15] indicates the sign bit, which maps to D[8]. For 16-bit differential mode, OFS[15:0] are directly subtracted from the conversion result data D[15:0]. In 16-bit single-ended mode, there is no field in the OFS corresponding to the least significant result D[0], so odd values, such as -1 or +1, cannot be subtracted from the result.

OFS is automatically set according to calibration requirements once the self-calibration sequence is done, that is, SC3[CAL] is cleared. The user may write to OFS to override the calibration result if desired. If the OFS is written by the user to a value that is different from the calibration value, the ADC error specifications may not be met. Storing the value generated by the calibration function in memory before overwriting with a user-specified value is recommended.

Note

> There is an effective limit to the values of offset that can be set by the user. If the magnitude of the offset is too high, the results of the conversions will cap off at the limits.

The offset calibration function may be employed by the user to remove application offsets or DC bias values. OFS may be written with a number in 2's complement format and this offset will be subtracted from the result, or hardware averaged value. To add an offset, store the negative offset in 2's complement format and the effect will be an addition. An offset correction that results in an out-of-range value will be forced to the minimum or maximum value. The minimum value for single-ended conversions is 0x0000; for a differential conversion it is 0x8000.

To preserve accuracy, the calibrated offset value initially stored in OFS must be added to the user-defined offset. For applications that may change the offset repeatedly during operation, store the initial offset calibration value in flash so it can be recovered and added to any user offset adjustment value and the sum stored in OFS.

40.5.8 Temperature sensor

The ADC module includes a temperature sensor whose output is connected to one of the ADC analog channel inputs.

The following equation provides an approximate transfer function of the temperature sensor.

$$\text{Temp} = 25 - \left(\left(V_{\text{TEMP}} - V_{\text{TEMP25}}\right) \div \text{m}\right)$$

Equation 2. Approximate transfer function of the temperature sensor

where:

- V_{TEMP} is the voltage of the temperature sensor channel at the ambient temperature.
- V_{TEMP25} is the voltage of the temperature sensor channel at 25 °C.
- m is referred as temperature sensor slope in the device data sheet. It is the hot or cold voltage versus temperature slope in V/°C.

For temperature calculations, use the V_{TEMP25} and temperature sensor slope values from the ADC Electricals table.

In application code, the user reads the temperature sensor channel, calculates V_{TEMP}, and compares to V_{TEMP25}. If V_{TEMP} is greater than V_{TEMP25} the cold slope value is applied in the preceding equation. If V_{TEMP} is less than V_{TEMP25}, the hot slope value is applied in the preceding equation. ADC Electricals table may only specify one temperature sensor slope value. In that case, the user could use the same slope for the calculation across the operational temperature range.

For more information on using the temperature sensor, see the application note titled *Temperature Sensor for the HCS08 Microcontroller Family* (document AN3031).

40.5.9 MCU wait mode operation

Wait mode is a lower-power consumption Standby mode from which recovery is fast because the clock sources remain active.

If a conversion is in progress when the MCU enters Wait mode, it continues until completion. Conversions can be initiated while the MCU is in Wait mode by means of the hardware trigger or if continuous conversions are enabled.

The bus clock, bus clock divided by two; and ADACK are available as conversion clock sources while in Wait mode. The use of ALTCLK as the conversion clock source in Wait is dependent on the definition of ALTCLK for this MCU. See the Chip Configuration information on ALTCLK specific to this MCU.

If the compare and hardware averaging functions are disabled, a conversion complete event sets SC1n[COCO] and generates an ADC interrupt to wake the MCU from Wait mode if the respective ADC interrupt is enabled, that is, when SC1n[AIEN]=1. If the hardware averaging function is enabled, SC1n[COCO] will set, and generate an interrupt if enabled, when the selected number of conversions are completed. If the compare function is enabled, SC1n[COCO] will set, and generate an interrupt if enabled, only if the compare conditions are met. If a single conversion is selected and the compare trigger is not met, the ADC will return to its idle state and cannot wake the MCU from Wait mode unless a new conversion is initiated by the hardware trigger.

40.5.10 MCU Normal Stop mode operation

Stop mode is a low-power consumption Standby mode during which most or all clock sources on the MCU are disabled.

40.5.10.1 Normal Stop mode with ADACK disabled

If the asynchronous clock, ADACK, is not selected as the conversion clock, executing a stop instruction aborts the current conversion and places the ADC in its Idle state. The contents of the ADC registers, including Rn, are unaffected by Normal Stop mode. After exiting from Normal Stop mode, a software or hardware trigger is required to resume conversions.

40.5.10.2 Normal Stop mode with ADACK enabled

If ADACK is selected as the conversion clock, the ADC continues operation during Normal Stop mode. See the chip-specific ADC information for configuration information for this device.

If a conversion is in progress when the MCU enters Normal Stop mode, it continues until completion. Conversions can be initiated while the MCU is in Normal Stop mode by means of the hardware trigger or if continuous conversions are enabled.

If the compare and hardware averaging functions are disabled, a conversion complete event sets SC1n[COCO] and generates an ADC interrupt to wake the MCU from Normal Stop mode if the respective ADC interrupt is enabled, that is, when SC1n[AIEN]=1. The result register, Rn, will contain the data from the first completed conversion that occurred during Normal Stop mode. If the hardware averaging function is enabled, SC1n[COCO] will set, and generate an interrupt if enabled, when the selected number of conversions are completed. If the compare function is enabled, SC1n[COCO] will set, and generate an interrupt if enabled, only if the compare conditions are met. If a single conversion is selected and the compare is not true, the ADC will return to its idle state and cannot wake the MCU from Normal Stop mode unless a new conversion is initiated by another hardware trigger.

40.6 Initialization information

This section gives an example that provides some basic direction on how to initialize and configure the ADC module.

The user can configure the module for 16-bit, 12-bit, 10-bit, or 8-bit single-ended resolution or 16-bit, 13-bit, 11-bit, or 9-bit differential resolution, single or continuous conversion, and a polled or interrupt approach, among many other options. For information used in this example, refer to Table 40-6, Table 40-7, and Table 40-8.

Note

> Hexadecimal values are designated by a preceding 0x, binary values designated by a preceding %, and decimal values have no preceding character.

40.6.1 ADC module initialization example

40.6.1.1 Initialization sequence

Before the ADC module can be used to complete conversions, an initialization procedure must be performed. A typical sequence is:

1. Calibrate the ADC by following the calibration instructions in Calibration function.
2. Update CFG to select the input clock source and the divide ratio used to generate ADCK. This register is also used for selecting sample time and low-power configuration.
3. Update SC2 to select the conversion trigger, hardware or software, and compare function options, if enabled.
4. Update SC3 to select whether conversions will be continuous or completed only once (ADCO) and whether to perform hardware averaging.
5. Update SC1:SC1n registers to select whether conversions will be single-ended or differential and to enable or disable conversion complete interrupts. Also, select the input channel which can be used to perform conversions.

40.6.1.2 Pseudo-code example

In this example, the ADC module is set up with interrupts enabled to perform a single 10-bit conversion at low-power with a long sample time on input channel 1, where ADCK is derived from the bus clock divided by 1.

CFG1 = 0x98 (%10011000)

```
        Bit 7   ADLPC   1      Configures for low power, lowers maximum clock speed.
        Bit 6:5 ADIV    00     Sets the ADCK to the input clock ÷ 1.
        Bit 4   ADLSMP  1      Configures for long sample time.
        Bit 3:2   MODE   10    Selects the single-ended 10-bit conversion, differential 11-
bit conversion.
        Bit 1:0   ADICLK   00    Selects the bus clock.
```

SC2 = 0x00 (%00000000)

```
       Bit 7   ADACT   0       Flag indicates if a conversion is in progress.
       Bit 6   ADTRG   0       Software trigger selected.
       Bit 5   ACFE    0       Compare function disabled.
       Bit 4   ACFGT   0       Not used in this example.
       Bit 3   ACREN   0      Compare range disabled.
 Bit 2 DMAEN 0 DMA request disabled.
       Bit 1:0 REFSEL  00     Selects default voltage reference pin pair (External pins
VREFH and VREFL).
```

SC1A = 0x41 (%01000001)

```
       Bit 7   COCO   0       Read-only flag which is set when a conversion completes.
       Bit 6   AIEN   1       Conversion complete interrupt enabled.
 Bit 5 DIFF 0 Single-ended conversion selected.
       Bit 4:0 ADCH   00001   Input channel 1 selected as ADC input channel.
```

RA = 0xxx

```
       Holds results of conversion.
```

CV = 0xxx

```
       Holds compare value when compare function enabled.
```

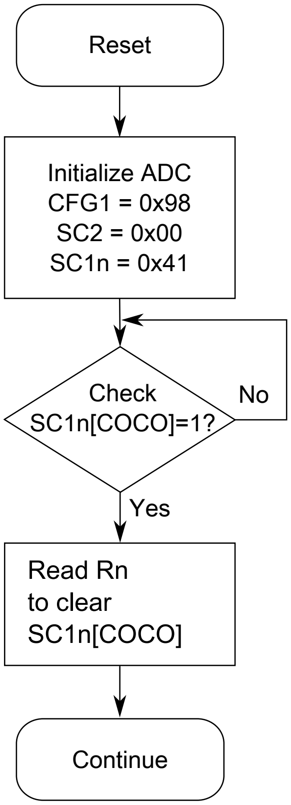


Figure 40-2. Initialization flowchart example

40.7 Application information

The ADC has been designed to be integrated into a microcontroller for use in embedded control applications requiring an ADC.

For guidance on selecting optimum external component values and converter parameters see AN4373: Cookbook for SAR ADC Measurements.

40.7.1 External pins and routing

40.7.1.1 Analog supply pins

Depending on the device, the analog power and ground supplies, V_{DDA} and V_{SSA}, of the ADC module are available as:

- V_{DDA} and V_{SSA} available as separate pins—When available on a separate pin, both V_{DDA} and V_{SSA} must be connected to the same voltage potential as their corresponding MCU digital supply, V_{DD} and V_{SS}, and must be routed carefully for maximum noise immunity and bypass capacitors placed as near as possible to the package.
- V_{SSA} is shared on the same pin as the MCU digital V_{SS}.
- V_{SSA} and V_{DDA} are shared with the MCU digital supply pins—In these cases, there are separate pads for the analog supplies bonded to the same pin as the corresponding digital supply so that some degree of isolation between the supplies is maintained.

If separate power supplies are used for analog and digital power, the ground connection between these supplies must be at the V_{SSA} pin. This must be the only ground connection between these supplies, if possible. V_{SSA} makes a good single point ground location.

40.7.1.2 Analog voltage reference pins

In addition to the analog supplies, the ADC module has connections for two reference voltage inputs used by the converter:

- V_{REFSH} is the high reference voltage for the converter.
- V_{REFSL} is the low reference voltage for the converter.

The ADC can be configured to accept one of two voltage reference pairs for V_{REFSH} and V_{REFSL}. Each pair contains a positive reference and a ground reference. The two pairs are external, V_{REFH} and V_{REFL} and alternate, V_{ALTH} and V_{ALTL}. These voltage references are selected using SC2[REFSEL]. The alternate voltage reference pair, V_{ALTH} and V_{ALTL}, may select additional external pins or internal sources based on MCU configuration. See the chip configuration information on the voltage references specific to this MCU.

In some packages, the external or alternate pairs are connected in the package to V_{DDA} and V_{SSA}, respectively. One of these positive references may be shared on the same pin as V_{DDA} on some devices. One of these ground references may be shared on the same pin as V_{SSA} on some devices.

If externally available, the positive reference may be connected to the same potential as V_{DDA} or may be driven by an external source to a level between the minimum Ref Voltage High and the V_{DDA} potential. The positive reference must never exceed V_{DDA}. If externally available, the ground reference must be connected to the same voltage potential as V_{SSA}. The voltage reference pairs must be routed carefully for maximum noise immunity and bypass capacitors placed as near as possible to the package.

AC current in the form of current spikes required to supply charge to the capacitor array at each successive approximation step is drawn through the V_{REFH} and V_{REFL} loop. The best external component to meet this current demand is a 0.1 µF capacitor with good high-frequency characteristics. This capacitor is connected between V_{REFH} and V_{REFL} and must be placed as near as possible to the package pins. Resistance in the path is not recommended because the current causes a voltage drop that could result in conversion errors. Inductance in this path must be minimum, that is, parasitic only.

40.7.1.3 Analog input pins

The external analog inputs are typically shared with digital I/O pins on MCU devices.

Empirical data shows that capacitors on the analog inputs improve performance in the presence of noise or when the source impedance is high. Use of 0.01 µF capacitors with good high-frequency characteristics is sufficient. These capacitors are not necessary in all cases, but when used, they must be placed as near as possible to the package pins and be referenced to V_{SSA}.

For proper conversion, the input voltage must fall between V_{REFH} and V_{REFL}. If the input is equal to or exceeds V_{REFH}, the converter circuit converts the signal to 0xFFF, which is full scale 12-bit representation, 0x3FF, which is full scale 10-bit representation, or 0xFF, which is full scale 8-bit representation. If the input is equal to or less than V_{REFL}, the converter circuit converts it to 0x000. Input voltages between V_{REFH} and V_{REFL} are straight-line linear conversions. There is a brief current associated with V_{REFL} when the sampling capacitor is charging.

For minimal loss of accuracy due to current injection, pins adjacent to the analog input pins must not be transitioning during conversions.

40.7.2 Sources of error

40.7.2.1 Sampling error

For proper conversions, the input must be sampled long enough to achieve the proper accuracy.

$$RAS + RADIN = SC / (FMAX * NUMTAU * CADIN)$$

Figure 40-3. Sampling equation

Where:

RAS = External analog source resistance

SC = Number of ADCK cycles used during sample window

CADIN = Internal ADC input capacitance

NUMTAU = -ln(LSBERR / 2^N)

LSBERR = value of acceptable sampling error in LSBs

N = 8 in 8-bit mode, 10 in 10-bit mode, 12 in 12-bit mode or 16 in 16-bit mode

Higher source resistances or higher-accuracy sampling is possible by setting CFG1[ADLSMP] and changing CFG2[ADLSTS] to increase the sample window, or decreasing ADCK frequency to increase sample time.

40.7.2.2 Pin leakage error

Leakage on the I/O pins can cause conversion error if the external analog source resistance, R_{AS}, is high. If this error cannot be tolerated by the application, keep R_{AS} lower than $V_{REFH} / (4 \times I_{LEAK} \times 2^N)$ for less than 1/4 LSB leakage error, where N = 8 in 8-bit mode, 10 in 10-bit mode, 12 in 12-bit mode, or 16 in 16-bit mode.

40.7.2.3 Noise-induced errors

System noise that occurs during the sample or conversion process can affect the accuracy of the conversion. The ADC accuracy numbers are guaranteed as specified only if the following conditions are met:

- There is a 0.1 µF low-ESR capacitor from V_{REFH} to V_{REFL}.
- There is a 0.1 µF low-ESR capacitor from V_{DDA} to V_{SSA}.
- If inductive isolation is used from the primary supply, an additional 1 µF capacitor is placed from V_{DDA} to V_{SSA}.
- V_{SSA}, and V_{REFL}, if connected, is connected to V_{SS} at a quiet point in the ground plane.
- Operate the MCU in Wait or Normal Stop mode before initiating (hardware-triggered conversions) or immediately after initiating (hardware- or software-triggered conversions) the ADC conversion.
  - For software triggered conversions, immediately follow the write to SC1 with a Wait instruction or Stop instruction.
  - For Normal Stop mode operation, select ADACK as the clock source. Operation in Normal Stop reduces V_{DD} noise but increases effective conversion time due to stop recovery.
- There is no I/O switching, input or output, on the MCU during the conversion.

There are some situations where external system activity causes radiated or conducted noise emissions or excessive V_{DD} noise is coupled into the ADC. In these situations, or when the MCU cannot be placed in Wait or Normal Stop mode, or I/O activity cannot be halted, the following actions may reduce the effect of noise on the accuracy:

- Place a 0.01 µF capacitor (C_{AS}) on the selected input channel to V_{REFL} or V_{SSA}. This improves noise issues, but affects the sample rate based on the external analog source resistance.
- Average the result by converting the analog input many times in succession and dividing the sum of the results. Four samples are required to eliminate the effect of a 1 LSB, one-time error.
- Reduce the effect of synchronous noise by operating off the asynchronous clock, that is, ADACK, and averaging. Noise that is synchronous to ADCK cannot be averaged out.

40.7.2.4 Code width and quantization error

The ADC quantizes the ideal straight-line transfer function into 65536 steps in the 16-bit mode.. Each step ideally has the same height, that is, 1 code, and width. The width is defined as the delta between the transition points to one code and the next. The ideal code width for an N-bit converter, where N can be 16, 12, 10, or 8, defined as 1 LSB, is:

$$1\text{LSB}=\left(V_{REFH}\right)/2^{N}$$

Equation 3. Ideal code width for an N-bit converter

There is an inherent quantization error due to the digitization of the result. For 8-bit, 10-bit, or 12-bit conversions, the code transitions when the voltage is at the midpoint between the points where the straight line transfer function is exactly represented by the actual transfer function. Therefore, the quantization error will be ± 1/2 LSB in 8-bit, 10-bit, or 12-bit modes. As a consequence, however, the code width of the first (0x000) conversion is only 1/2 LSB and the code width of the last (0xFF or 0x3FF) is 1.5 LSB.

For 16-bit conversions, the code transitions only after the full code width is present, so the quantization error is -1 LSB to 0 LSB and the code width of each step is 1 LSB.

40.7.2.5 Linearity errors

The ADC may also exhibit non-linearity of several forms. Every effort has been made to reduce these errors, but the system designers must be aware of these errors because they affect overall accuracy:

- Zero-scale error (E_{ZS}), sometimes called offset: This error is defined as the difference between the actual code width of the first conversion and the ideal code width. This is 1/2 LSB in 8-bit, 10-bit, or 12-bit modes and 1 LSB in 16-bit mode. If the first conversion is 0x001, the difference between the actual 0x001 code width and its ideal (1 LSB) is used.
- Full-scale error (E_{FS}): This error is defined as the difference between the actual code width of the last conversion and the ideal code width. This is 1.5 LSB in 8-bit, 10-bit, or 12-bit modes and 1 LSB in 16-bit mode. If the last conversion is 0x3FE, the difference between the actual 0x3FE code width and its ideal (1 LSB) is used.
- Differential non-linearity (DNL): This error is defined as the worst-case difference between the actual code width and the ideal code width for all conversions.

- Integral non-linearity (INL): This error is defined as the highest-value or absolute value that the running sum of DNL achieves. More simply, this is the worst-case difference of the actual transition voltage to a given code and its corresponding ideal transition voltage, for all codes.
- Total unadjusted error (TUE): This error is defined as the difference between the actual transfer function and the ideal straight-line transfer function and includes all forms of error.

40.7.2.6 Code jitter, non-monotonicity, and missing codes

Analog-to-digital converters are susceptible to three special forms of error:

- Code jitter: Code jitter occurs when a given input voltage converts to one of the two values when sampled repeatedly. Ideally, when the input voltage is infinitesimally smaller than the transition voltage, the converter yields the lower code, and vice-versa. However, even small amounts of system noise can cause the converter to be indeterminate, between two codes, for a range of input voltages around the transition voltage.

  This error may be reduced by repeatedly sampling the input and averaging the result. Additionally, the techniques discussed in Noise-induced errors reduces this error.
- Non-monotonicity: Non-monotonicity occurs when, except for code jitter, the converter converts to a lower code for a higher input voltage.
- Missing codes: Missing codes are those values never converted for any input value.

In 8-bit or 10-bit mode, the ADC is guaranteed to be monotonic and have no missing codes.

Chapter 41
MCU: Comparator (CMP)

41.1 Introduction

NOTE

> For the chip-specific implementation details of this module's instances, see the chip configuration information.

The comparator (CMP) module provides a circuit for comparing two analog input voltages. The comparator circuit is designed to operate across the full range of the supply voltage, known as rail-to-rail operation.

The Analog MUX (ANMUX) provides a circuit for selecting an analog input signal from eight channels. One signal is provided by the 6-bit digital-to-analog converter (DAC). The mux circuit is designed to operate across the full range of the supply voltage.

The 6-bit DAC is 64-tap resistor ladder network which provides a selectable voltage reference for applications where voltage reference is needed. The 64-tap resistor ladder network divides the supply reference V_{in} into 64 voltage levels. A 6-bit digital signal input selects the output voltage level, which varies from V_{in} to $V_{in}/64$. V_{in} can be selected from two voltage sources, V_{in1} and V_{in2}. The 6-bit DAC from a comparator is available as an on-chip internal signal only and is not available externally to a pin.

41.1.1 CMP features

The CMP has the following features:

- Operational over the entire supply range
- Inputs may range from rail to rail
- Programmable hysteresis control

- Selectable interrupt on rising-edge, falling-edge, or both rising or falling edges of the comparator output
- Selectable inversion on comparator output
- Capability to produce a wide range of outputs such as:
  - Sampled
  - Windowed, which is ideal for certain PWM zero-crossing-detection applications
  - Digitally filtered:
    - Filter can be bypassed
    - Can be clocked via external SAMPLE signal or scaled bus clock
- External hysteresis can be used at the same time that the output filter is used for internal functions
- Two software selectable performance levels:
  - Shorter propagation delay at the expense of higher power
  - Low power, with longer propagation delay
- DMA transfer support
  - A comparison event can be selected to trigger a DMA transfer
- Functional in all modes of operation
- The window and filter functions are not available in the following modes:
  - Stop
  - VLPS
  - LLS
  - VLLSx

41.1.2 6-bit DAC key features

The 6-bit DAC has the following features:

- 6-bit resolution
- Selectable supply reference source
- Power Down mode to conserve power when not in use
- Option to route the output to internal comparator input

41.1.3 ANMUX key features

The ANMUX has the following features:

- Two 8-to-1 channel mux
- Operational over the entire supply range

41.1.4 CMP, DAC and ANMUX diagram

The following figure shows the block diagram for the High-Speed Comparator, DAC, and ANMUX modules.

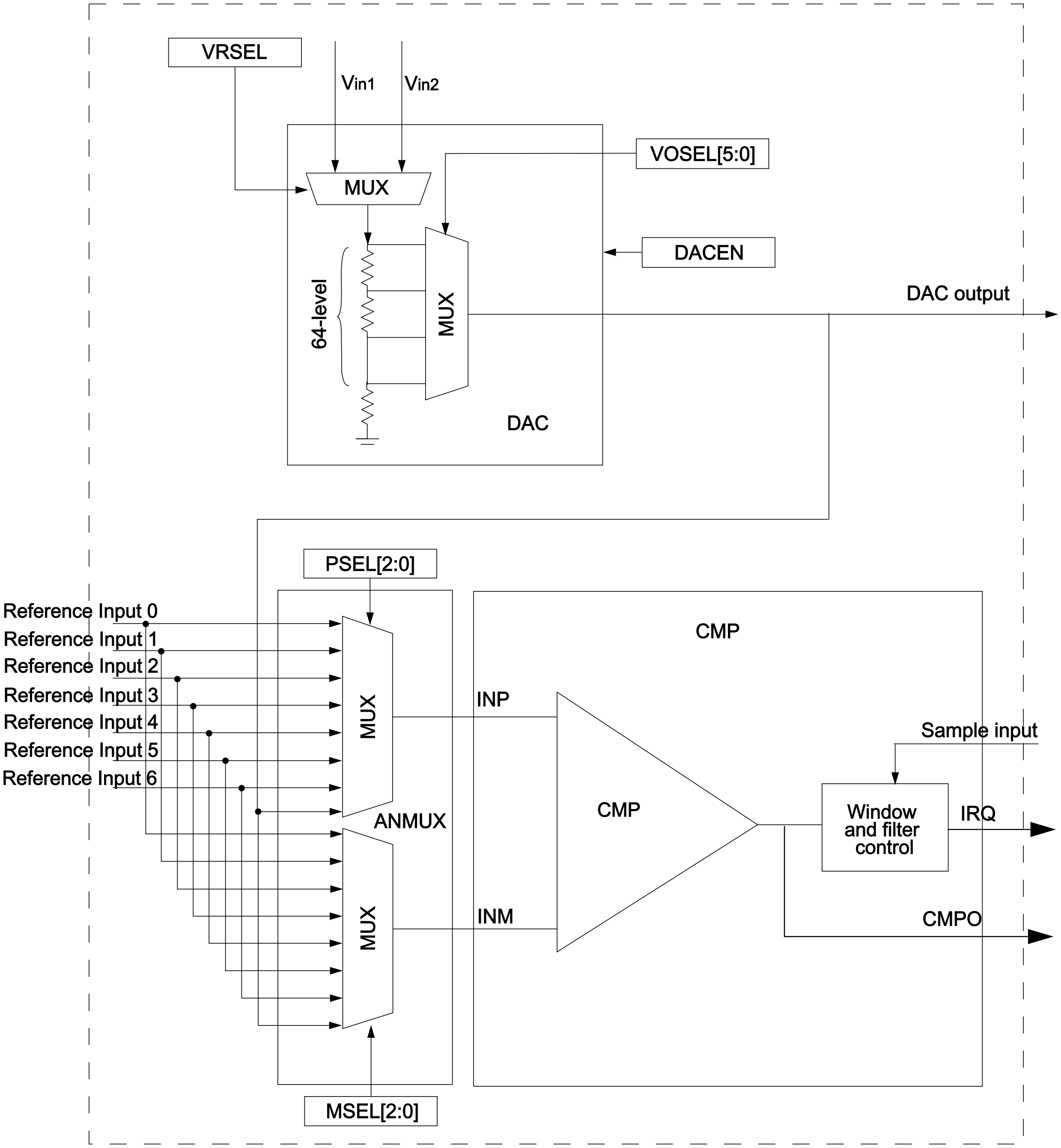


Figure 41-1. CMP, DAC and ANMUX block diagram

41.1.5 CMP block diagram

The following figure shows the block diagram for the CMP module.

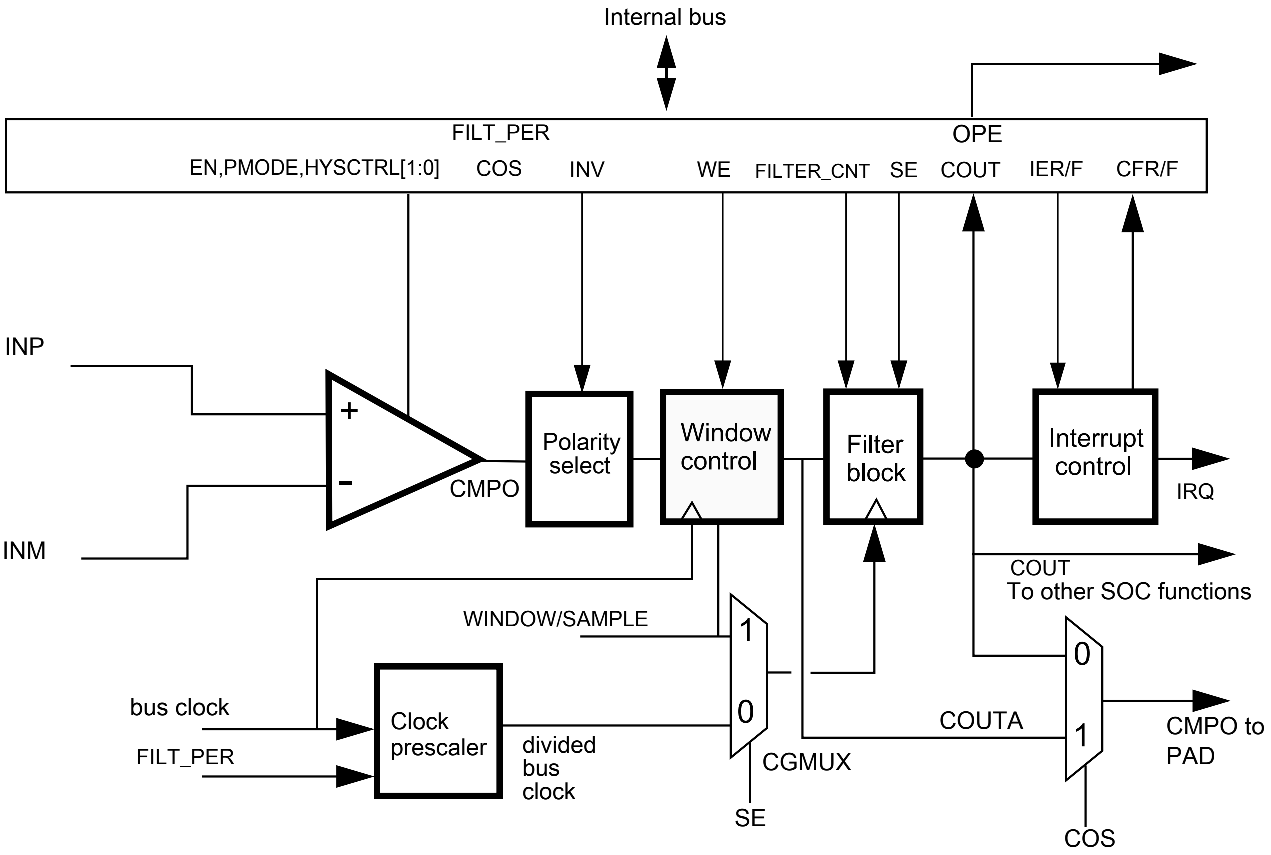


Figure 41-2. Comparator module block diagram

In the CMP block diagram:

- The Window Control block is bypassed when CR1[WE] = 0
- If CR1[WE] = 1, the comparator output will be sampled on every bus clock when WINDOW=1 to generate COUTA. Sampling does NOT occur when WINDOW = 0.
- The Filter block is bypassed when not in use.
- The Filter block acts as a simple sampler if the filter is bypassed and CR0[FILTER_CNT] is set to 0x01.
- The Filter block filters based on multiple samples when the filter is bypassed and CR0[FILTER_CNT] is set greater than 0x01.
  - If CR1[SE] = 1, the external SAMPLE input is used as sampling clock
  - If CR1[SE] = 0, the divided bus clock is used as sampling clock

- If enabled, the Filter block will incur up to one bus clock additional latency penalty on COUT due to the fact that COUT, which is crossing clock domain boundaries, must be resynchronized to the bus clock.
- CR1[WE] and CR1[SE] are mutually exclusive.

41.2 Memory map/register definitions

CMP memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4007_3000	CMP Control Register 0 (CMP0_CR0)	8	R/W	00h	41.2.1/918
4007_3001	CMP Control Register 1 (CMP0_CR1)	8	R/W	00h	41.2.2/919
4007_3002	CMP Filter Period Register (CMP0_FPR)	8	R/W	00h	41.2.3/921
4007_3003	CMP Status and Control Register (CMP0_SCR)	8	R/W	00h	41.2.4/921
4007_3004	DAC Control Register (CMP0_DACCR)	8	R/W	00h	41.2.5/922
4007_3005	MUX Control Register (CMP0_MUXCR)	8	R/W	00h	41.2.6/923
4007_3008	CMP Control Register 0 (CMP1_CR0)	8	R/W	00h	41.2.1/918
4007_3009	CMP Control Register 1 (CMP1_CR1)	8	R/W	00h	41.2.2/919
4007_300A	CMP Filter Period Register (CMP1_FPR)	8	R/W	00h	41.2.3/921
4007_300B	CMP Status and Control Register (CMP1_SCR)	8	R/W	00h	41.2.4/921
4007_300C	DAC Control Register (CMP1_DACCR)	8	R/W	00h	41.2.5/922
4007_300D	MUX Control Register (CMP1_MUXCR)	8	R/W	00h	41.2.6/923

41.2.1 CMP Control Register 0 (CMPx_CR0)

Address: Base address + 0h offset

Bit	7	6	5	4	3	2	1	0
Read	0	FILTER_CNT			0	0	HYSTCTR	
Write								
Reset	0	0	0	0	0	0	0	0

CMPx_CR0 field descriptions

Field	Description
7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
6–4 FILTER_CNT	Filter Sample Count

Table continues on the next page...

CMPx_CR0 field descriptions (continued)

Field	Description
	Represents the number of consecutive samples that must agree prior to the comparator ouput filter accepting a new output state. For information regarding filter programming and latency, see the Functional description. 000 Filter is disabled. If SE = 1, then COUT is a logic 0. This is not a legal state, and is not recommended. If SE = 0, COUT = COUTA. 001 One sample must agree. The comparator output is simply sampled. 010 2 consecutive samples must agree. 011 3 consecutive samples must agree. 100 4 consecutive samples must agree. 101 5 consecutive samples must agree. 110 6 consecutive samples must agree. 111 7 consecutive samples must agree.
3 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
2 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
HYSTCTR	Comparator hard block hysteresis control Defines the programmable hysteresis level. The hysteresis values associated with each level are device-specific. See the Data Sheet of the device for the exact values. 00 Level 0 01 Level 1 10 Level 2 11 Level 3

41.2.2 CMP Control Register 1 (CMPx_CR1)

Address: Base address + 1h offset

Bit	7	6	5	4	3	2	1	0
Read	SE	WE	0	PMODE	INV	COS	OPE	EN
Write			▨					
Reset	0	0	0	0	0	0	0	0

CMPx_CR1 field descriptions

Field	Description
7 SE	Sample Enable At any given time, either SE or WE can be set. If a write to this register attempts to set both, then SE is set and WE is cleared. However, avoid writing 1s to both field locations because this "11" case is reserved and may change in future implementations. 0 Sampling mode is not selected. 1 Sampling mode is selected.

Table continues on the next page...

CMPx_CR1 field descriptions (continued)

Field	Description
6 WE	Windowing Enable At any given time, either SE or WE can be set. If a write to this register attempts to set both, then SE is set and WE is cleared. However, avoid writing 1s to both field locations because this "11" case is reserved and may change in future implementations. 0 Windowing mode is not selected. 1 Windowing mode is selected.
5 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
4 PMODE	Power Mode Select See the electrical specifications table in the device Data Sheet for details. 0 Low-Speed (LS) Comparison mode selected. In this mode, CMP has slower output propagation delay and lower current consumption. 1 High-Speed (HS) Comparison mode selected. In this mode, CMP has faster output propagation delay and higher current consumption.
3 INV	Comparator INVERT Allows selection of the polarity of the analog comparator function. It is also driven to the COUT output, on both the device pin and as SCR[COUT], when OPE=0. 0 Does not invert the comparator output. 1 Inverts the comparator output.
2 COS	Comparator Output Select 0 Set the filtered comparator output (CMPO) to equal COUT. 1 Set the unfiltered comparator output (CMPO) to equal COUTA.
1 OPE	Comparator Output Pin Enable 0 CMPO is not available on the associated CMPO output pin. If the comparator does not own the pin, this field has no effect. 1 CMPO is available on the associated CMPO output pin. The comparator output (CMPO) is driven out on the associated CMPO output pin if the comparator owns the pin. If the comparator does not own the field, this bit has no effect.
0 EN	Comparator Module Enable Enables the Analog Comparator module. When the module is not enabled, it remains in the off state, and consumes no power. When the user selects the same input from analog mux to the positive and negative port, the comparator is disabled automatically. 0 Analog Comparator is disabled. 1 Analog Comparator is enabled.

41.2.3 CMP Filter Period Register (CMPx_FPR)

Address: Base address + 2h offset

Bit	7	6	5	4	3	2	1	0
Read Write	FILT_PER							
Reset	0	0	0	0	0	0	0	0

CMPx_FPR field descriptions

Field	Description
FILT_PER	Filter Sample Period Specifies the sampling period, in bus clock cycles, of the comparator output filter, when CR1[SE]=0. Setting FILT_PER to 0x0 disables the filter. Filter programming and latency details appear in the Functional description. This field has no effect when CR1[SE]=1. In that case, the external SAMPLE signal is used to determine the sampling period.

41.2.4 CMP Status and Control Register (CMPx_SCR)

Address: Base address + 3h offset

Bit	7	6	5	4	3	2	1	0
Read	0	DMAEN	0	IER	IEF	CFR	CFF	COUT
Write						w1c	w1c	
Reset	0	0	0	0	0	0	0	0

CMPx_SCR field descriptions

Field	Description
7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
6 DMAEN	DMA Enable Control Enables the DMA transfer triggered from the CMP module. When this field is set, a DMA request is asserted when CFR or CFF is set. 0 DMA is disabled. 1 DMA is enabled.
5 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
4 IER	Comparator Interrupt Enable Rising Enables the CFR interrupt from the CMP. When this field is set, an interrupt will be asserted when CFR is set.

Table continues on the next page...

CMPx_SCR field descriptions (continued)

Field	Description
	0 Interrupt is disabled. 1 Interrupt is enabled.
3 IEF	Comparator Interrupt Enable Falling Enables the CFF interrupt from the CMP. When this field is set, an interrupt will be asserted when CFF is set. 0 Interrupt is disabled. 1 Interrupt is enabled.
2 CFR	Analog Comparator Flag Rising Detects a rising-edge on COUT, when set, during normal operation. CFR is cleared by writing 1 to it. During Stop modes, CFR is level sensitive . 0 Rising-edge on COUT has not been detected. 1 Rising-edge on COUT has occurred.
1 CFF	Analog Comparator Flag Falling Detects a falling-edge on COUT, when set, during normal operation. CFF is cleared by writing 1 to it. During Stop modes, CFF is level sensitive . 0 Falling-edge on COUT has not been detected. 1 Falling-edge on COUT has occurred.
0 COUT	Analog Comparator Output Returns the current value of the Analog Comparator output, when read. The field is reset to 0 and will read as CR1[INV] when the Analog Comparator module is disabled, that is, when CR1[EN] = 0. Writes to this field are ignored.

41.2.5 DAC Control Register (CMPx_DACCR)

Address: Base address + 4h offset

Bit	7	6	5	4	3	2	1	0
Read Write	DACEN	VRSEL	VOSEL					
Reset	0	0	0	0	0	0	0	0

CMPx_DACCR field descriptions

Field	Description
7 DACEN	DAC Enable Enables the DAC. When the DAC is disabled, it is powered down to conserve power. 0 DAC is disabled. 1 DAC is enabled.
6 VRSEL	Supply Voltage Reference Source Select

Table continues on the next page...

CMPx_DACCR field descriptions (continued)

Field	Description
	0 V_{in1} is selected as resistor ladder network supply reference. 1 V_{in2} is selected as resistor ladder network supply reference.
VOSEL	DAC Output Voltage Select Selects an output voltage from one of 64 distinct levels. DACO = (V_{in} /64) * (VOSEL[5:0] + 1) , so the DACO range is from V_{in} /64 to V_{in} .

41.2.6 MUX Control Register (CMPx_MUXCR)

Address: Base address + 5h offset

Bit	7	6	5	4	3	2	1	0
Read	PSTM	0	PSEL			MSEL		
Write		▨						
Reset	0	0	0	0	0	0	0	0

CMPx_MUXCR field descriptions

Field	Description
7 PSTM	Pass Through Mode Enable This bit is used to enable to MUX pass through mode. Pass through mode is always available but for some devices this feature must be always disabled due to the lack of package pins. 0 Pass Through Mode is disabled. 1 Pass Through Mode is enabled.
6 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
5–3 PSEL	Plus Input Mux Control Determines which input is selected for the plus input of the comparator. For INx inputs, see CMP, DAC, and ANMUX block diagrams. **NOTE:** When an inappropriate operation selects the same input for both muxes, the comparator automatically shuts down to prevent itself from becoming a noise generator. 000 IN0 001 IN1 010 IN2 011 IN3 100 IN4 101 IN5 110 IN6 111 IN7
MSEL	Minus Input Mux Control Determines which input is selected for the minus input of the comparator. For INx inputs, see CMP, DAC, and ANMUX block diagrams.

Table continues on the next page...

CMPx_MUXCR field descriptions (continued)

Field	Description
	NOTE: When an inappropriate operation selects the same input for both muxes, the comparator automatically shuts down to prevent itself from becoming a noise generator. 000 IN0 001 IN1 010 IN2 011 IN3 100 IN4 101 IN5 110 IN6 111 IN7

41.3 Functional description

The CMP module can be used to compare two analog input voltages applied to INP and INM.

CMPO is high when the non-inverting input is greater than the inverting input, and is low when the non-inverting input is less than the inverting input. This signal can be selectively inverted by setting CR1[INV] = 1.

SCR[IER] and SCR[IEF] are used to select the condition which will cause the CMP module to assert an interrupt to the processor. SCR[CFF] is set on a falling-edge and SCR[CFR] is set on rising-edge of the comparator output. The optionally filtered CMPO can be read directly through SCR[COUT].

41.3.1 CMP functional modes

There are the following main sub-blocks to the CMP module:

- The comparator itself
- The window function
- The filter function

The filter, CR0[FILTER_CNT], can be clocked from an internal or external clock source. The filter is programmable with respect to the number of samples that must agree before a change in the output is registered. In the simplest case, only one sample must agree. In this case, the filter acts as a simple sampler.

The external sample input is enabled using CR1[SE]. When set, the output of the comparator is sampled only on rising edges of the sample input.

The "windowing mode" is enabled by setting CR1[WE]. When set, the comparator output is sampled only when WINDOW=1. This feature can be used to ignore the comparator output during time periods in which the input voltages are not valid. This is especially useful when implementing zero-crossing-detection for certain PWM applications.

The comparator filter and sampling features can be combined as shown in the following table. Individual modes are discussed below.

Table 41-1. Comparator sample/filter controls

Mode #	CR1[EN]	CR1[WE]	CR1[SE]	CR0[FILTER_CNT]	FPR[FILT_PER]	Operation
1	0	X	X	X	X	**Disabled** See the Disabled mode (# 1).
2A	1	0	0	0x00	X	**Continuous Mode** See the Continuous mode (#s 2A & 2B).
2B	1	0	0	X	0x00	
3A	1	0	1	0x01	X	**Sampled, Non-Filtered mode** See the Sampled, Non-Filtered mode (#s 3A & 3B).
3B	1	0	0	0x01	> 0x00	
4A	1	0	1	> 0x01	X	**Sampled, Filtered mode** See the Sampled, Filtered mode (#s 4A & 4B).
4B	1	0	0	> 0x01	> 0x00	
5A	1	1	0	0x00	X	**Windowed mode** Comparator output is sampled on every rising bus clock edge when SAMPLE=1 to generate COUTA. See the Windowed mode (#s 5A & 5B).
5B	1	1	0	X	0x00	
6	1	1	0	0x01	0x01–0xFF	**Windowed/Resampled mode** Comparator output is sampled on every rising bus clock edge when SAMPLE=1 to generate COUTA, which is then resampled on an interval determined by FILT_PER to generate COUT. See the Windowed/Resampled mode (# 6).
7	1	1	0	> 0x01	0x01–0xFF	**Windowed/Filtered mode** Comparator output is sampled on every rising bus clock edge when SAMPLE=1 to generate COUTA, which is then resampled and filtered to generate COUT. See the Windowed/Filtered mode (#7).
All other combinations of CR1[EN], CR1[WE], CR1[SE], CR0[FILTER_CNT], and FPR[FILT_PER] are illegal.						

For cases where a comparator is used to drive a fault input, for example, for a motor-control module such as FTM, it must be configured to operate in Continuous mode so that an external fault can immediately pass through the comparator to the target fault circuitry.

Note

> Filtering and sampling settings must be changed only after setting CR1[SE]=0 and CR0[FILTER_CNT]=0x00. This resets the filter to a known state.

41.3.1.1 Disabled mode (# 1)

In Disabled mode, the analog comparator is non-functional and consumes no power. CMPO is 0 in this mode.

41.3.1.2 Continuous mode (#s 2A & 2B)

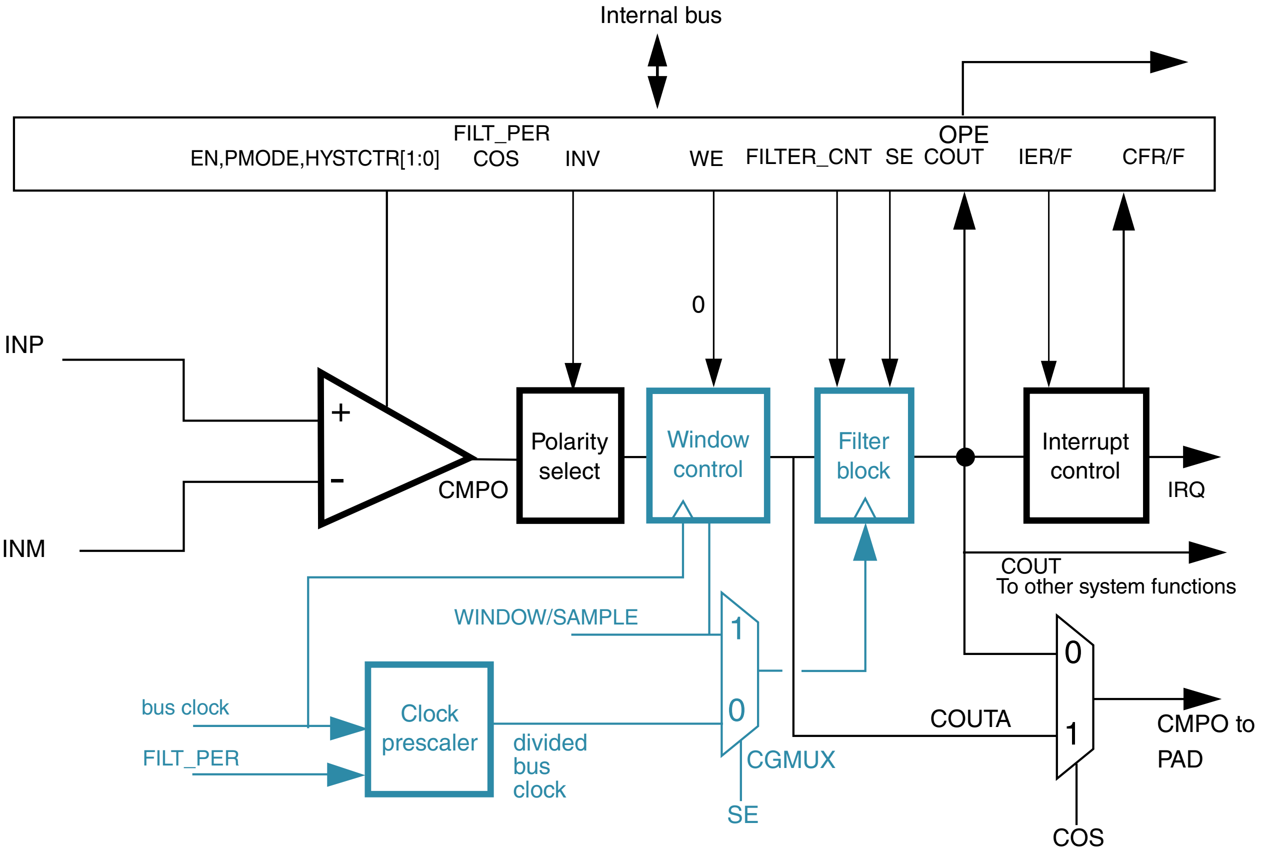


Figure 41-3. Comparator operation in Continuous mode

NOTE

See the chip configuration section for the source of sample/ window input.

The analog comparator block is powered and active. CMPO may be optionally inverted, but is not subject to external sampling or filtering. Both window control and filter blocks are completely bypassed. SCR[COUT] is updated continuously. The path from comparator input pins to output pin is operating in combinational unclocked mode. COUT and COUTA are identical.

For control configurations which result in disabling the filter block, see the Filter Block Bypass Logic diagram.

41.3.1.3 Sampled, Non-Filtered mode (#s 3A & 3B)

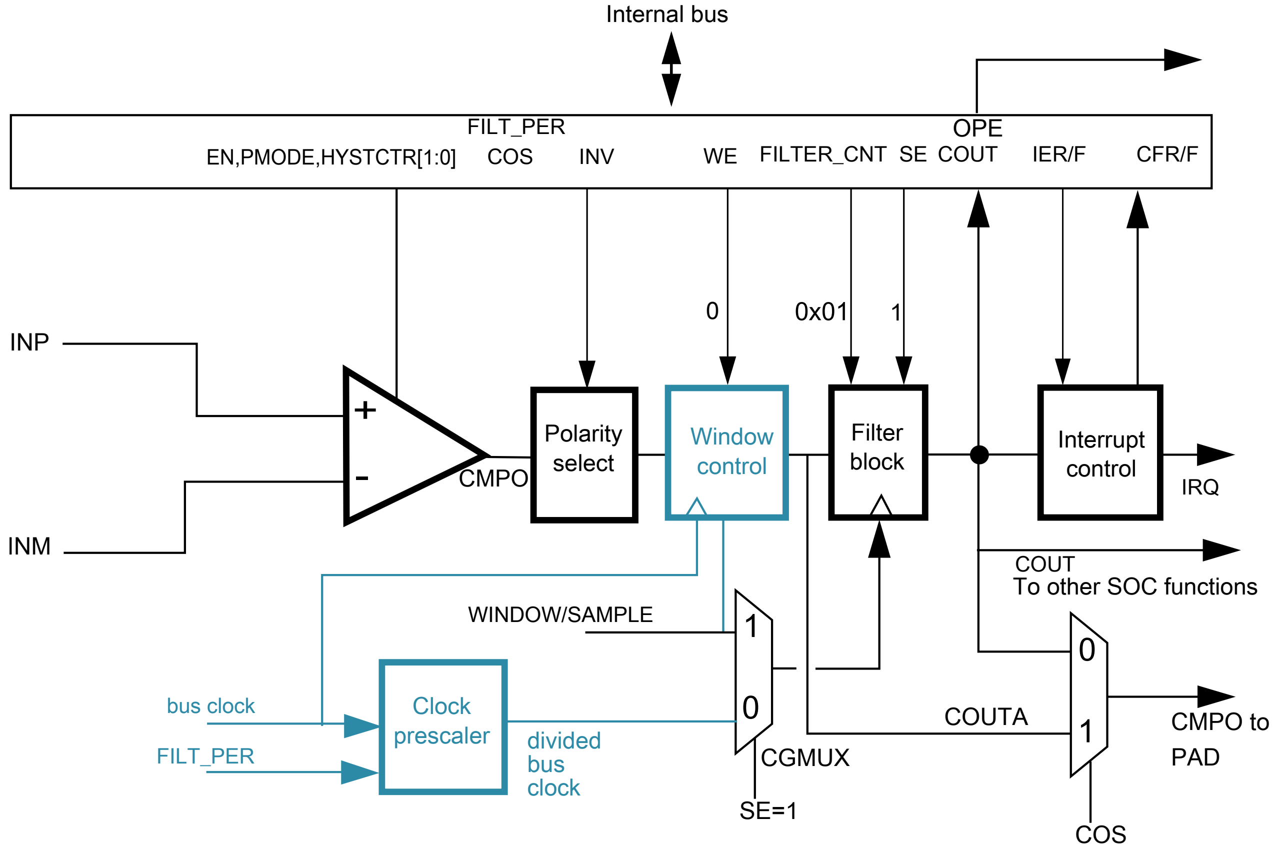


Figure 41-4. Sampled, Non-Filtered (# 3A): sampling point externally driven

In Sampled, Non-Filtered mode, the analog comparator block is powered and active. The path from analog inputs to COUTA is combinational unclocked. Windowing control is completely bypassed. COUTA is sampled whenever a rising-edge is detected on the filter block clock input.

The only difference in operation between Sampled, Non-Filtered (# 3A) and Sampled, Non-Filtered (# 3B) is in how the clock to the filter block is derived. In #3A, the clock to filter block is externally derived while in #3B, the clock to filter block is internally derived.

The comparator filter has no other function than sample/hold of the comparator output in this mode (# 3B).

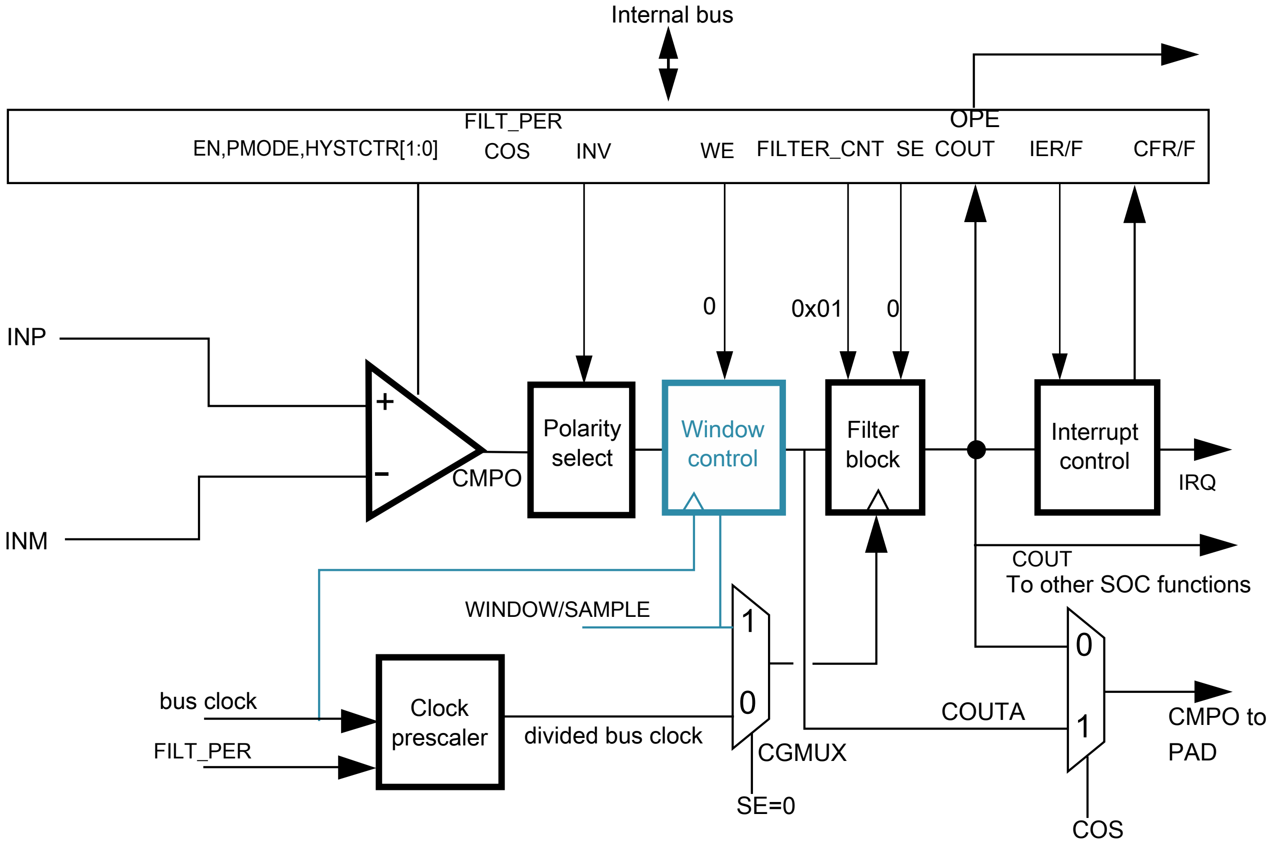


Figure 41-5. Sampled, Non-Filtered (# 3B): sampling interval internally derived

41.3.1.4 Sampled, Filtered mode (#s 4A & 4B)

In Sampled, Filtered mode, the analog comparator block is powered and active. The path from analog inputs to COUTA is combinational unclocked. Windowing control is completely bypassed. COUTA is sampled whenever a rising edge is detected on the filter block clock input.

The only difference in operation between Sampled, Non-Filtered (# 3A) and Sampled, Filtered (# 4A) is that, now, CR0[FILTER_CNT]>1, which activates filter operation.

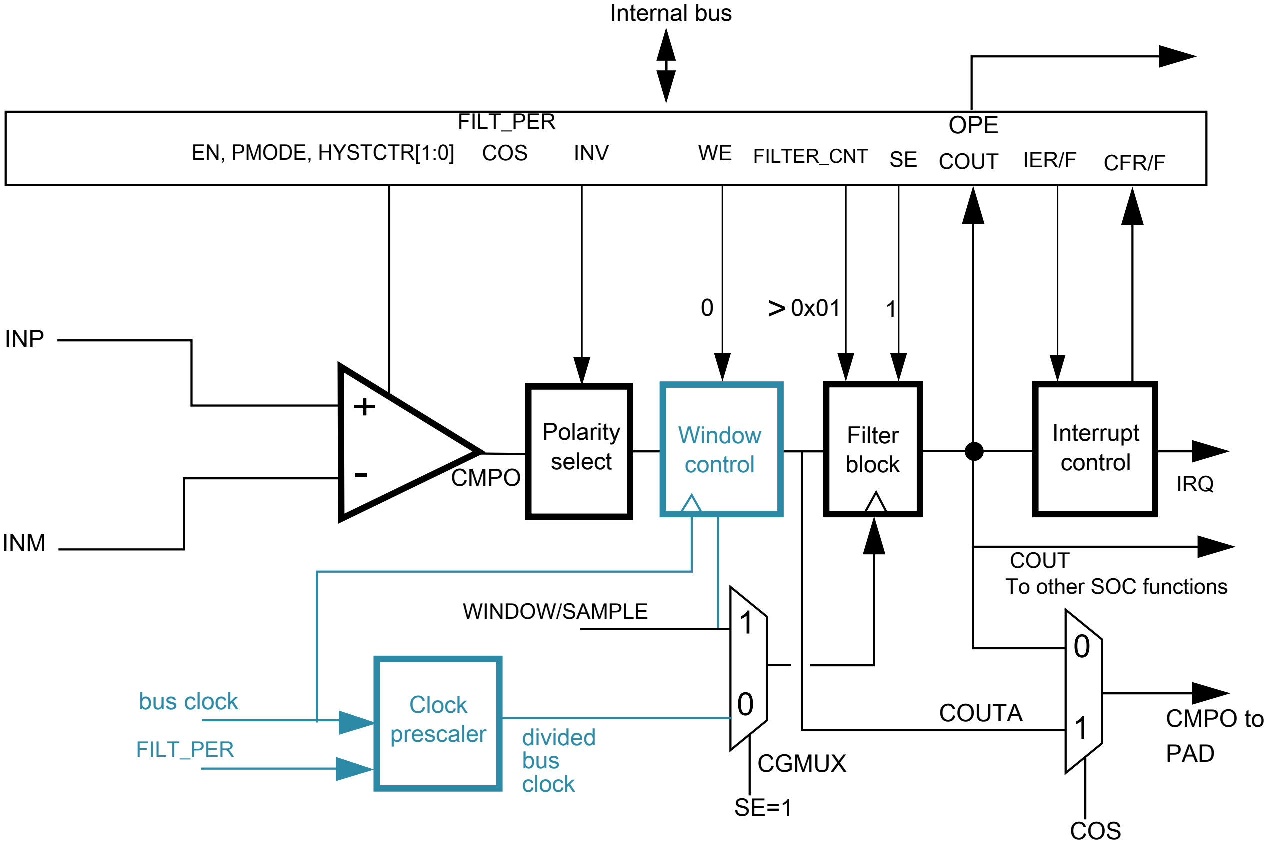


Figure 41-6. Sampled, Filtered (# 4A): sampling point externally driven

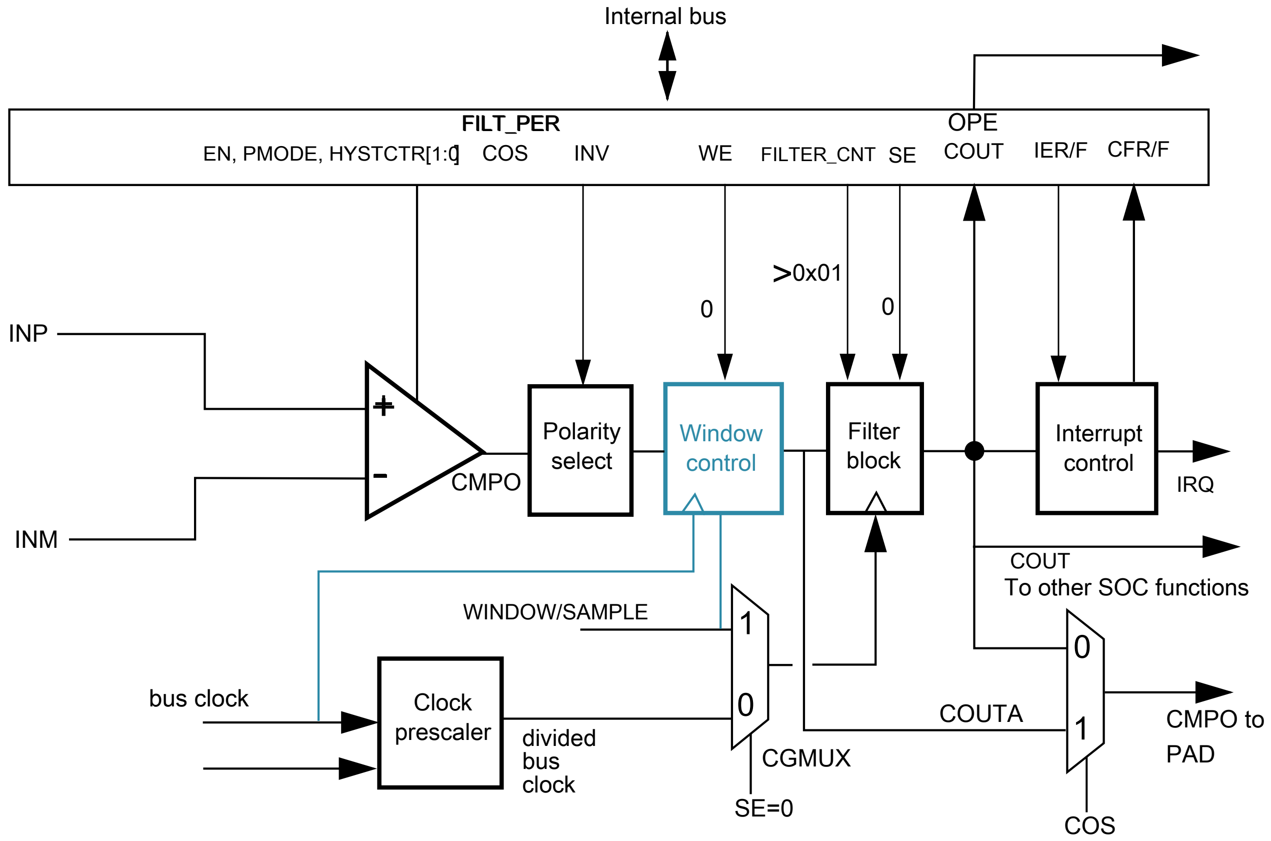


Figure 41-7. Sampled, Filtered (# 4B): sampling point internally derived

The only difference in operation between Sampled, Non-Filtered (# 3B) and Sampled, Filtered (# 4B) is that now, CR0[FILTER_CNT]>1, which activates filter operation.

41.3.1.5 Windowed mode (#s 5A & 5B)

The following figure illustrates comparator operation in the Windowed mode, ignoring latency of the analog comparator, polarity select, and window control block. It also assumes that the polarity select is set to non-inverting state.

NOTE

The analog comparator output is passed to COUTA only when the WINDOW signal is high.

In actual operation, COUTA may lag the analog inputs by up to one bus clock cycle plus the combinational path delay through the comparator and polarity select logic.

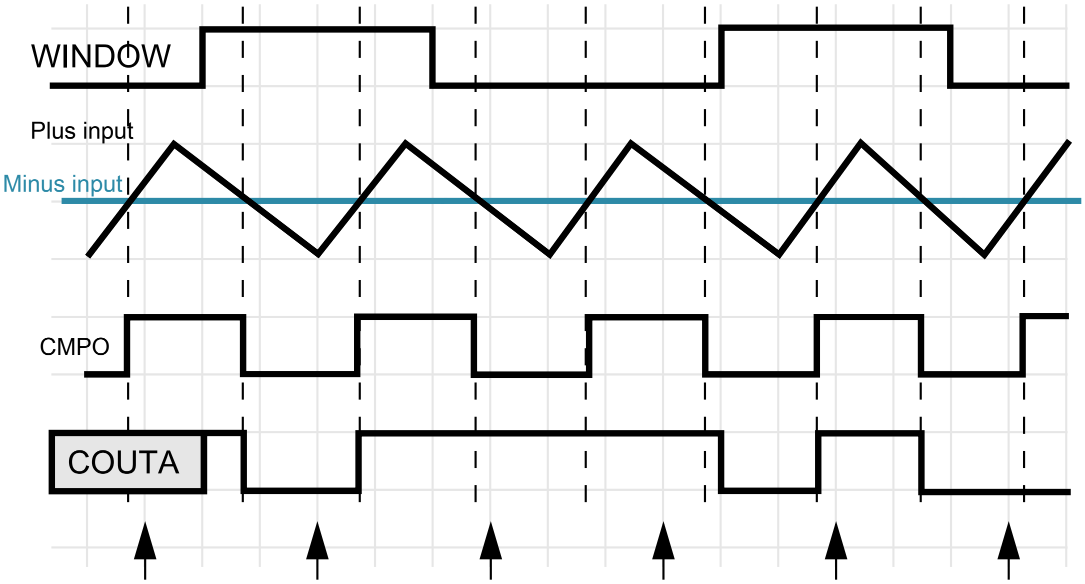


Figure 41-8. Windowed mode operation

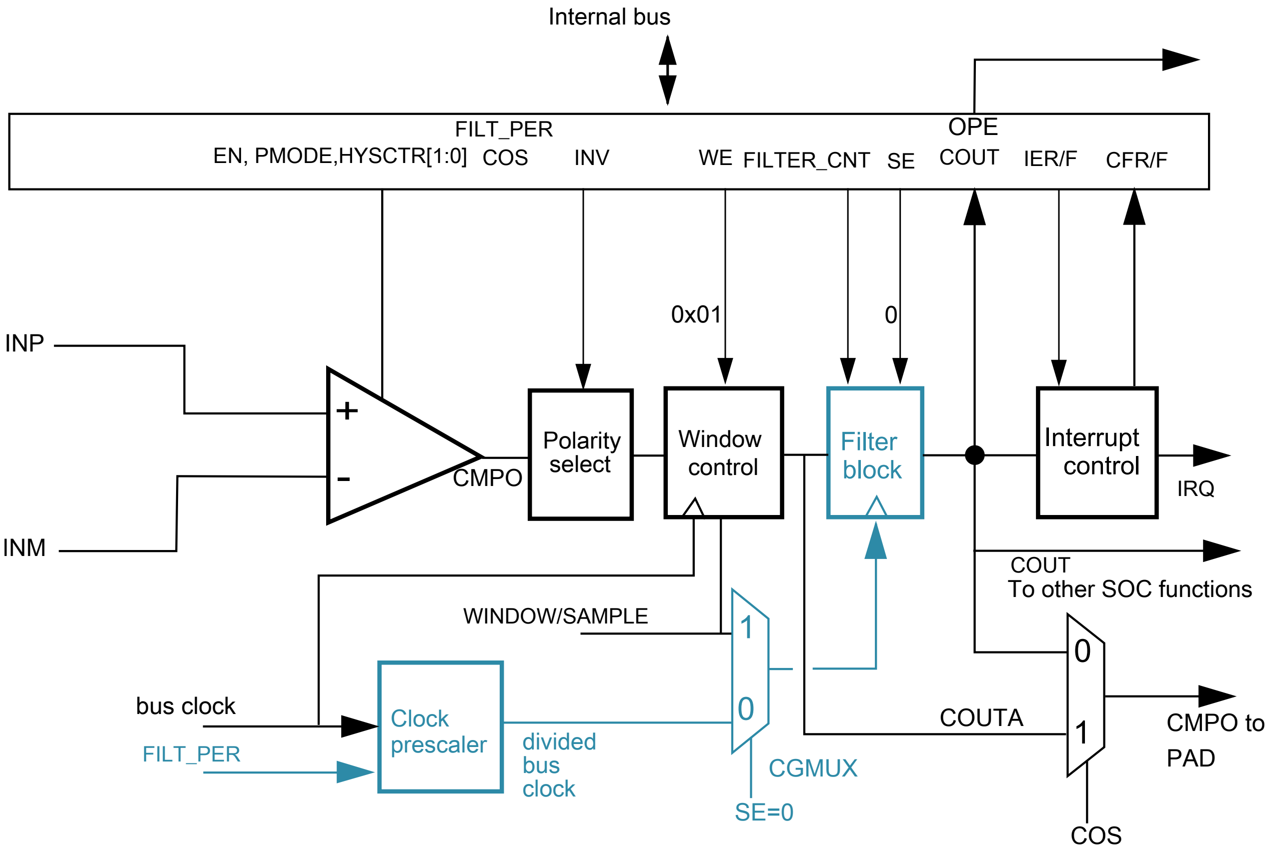


Figure 41-9. Windowed mode

For control configurations which result in disabling the filter block, see Filter Block Bypass Logic diagram.

When any windowed mode is active, COUTA is clocked by the bus clock whenever WINDOW = 1. The last latched value is held when WINDOW = 0.

41.3.1.6 Windowed/Resampled mode (# 6)

The following figure uses the same input stimulus shown in Figure 41-8, and adds resampling of COUTA to generate COUT. Samples are taken at the time points indicated by the arrows in the figure. Again, prop delays and latency are ignored for the sake of clarity.

This example was generated solely to demonstrate operation of the comparator in windowed/resampled mode, and does not reflect any specific application. Depending upon the sampling rate and window placement, COUT may not see zero-crossing events detected by the analog comparator. Sampling period and/or window placement must be carefully considered for a given application.

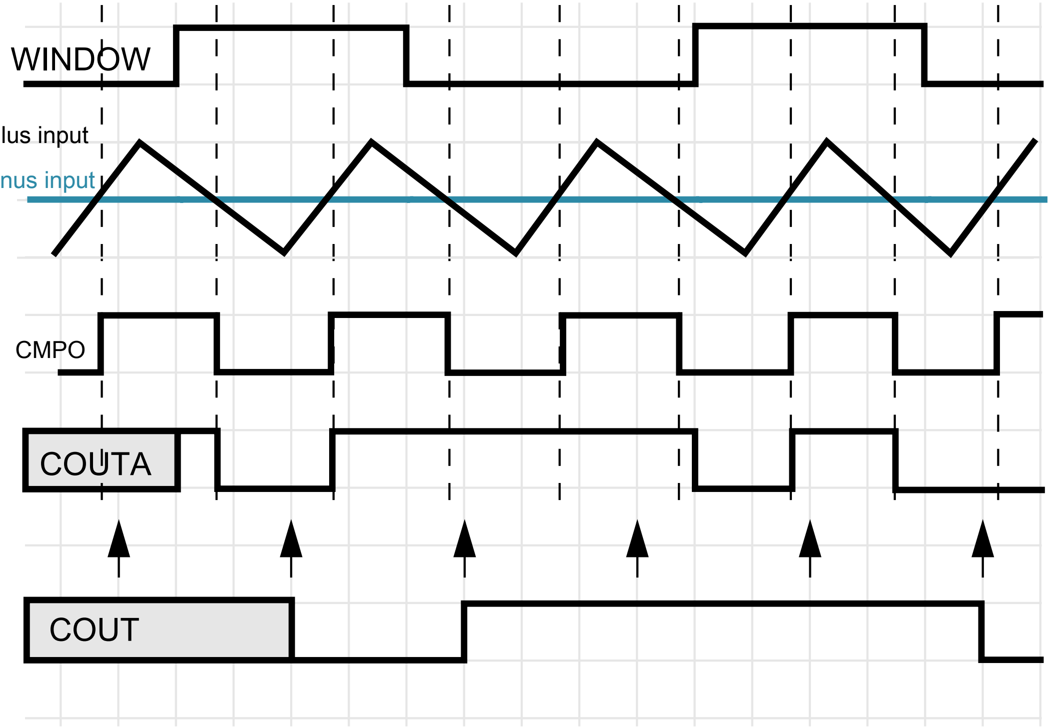


Figure 41-10. Windowed/resampled mode operation

This mode of operation results in an unfiltered string of comparator samples where the interval between the samples is determined by FPR[FILT_PER] and the bus clock rate. Configuration for this mode is virtually identical to that for the Windowed/Filtered Mode shown in the next section. The only difference is that the value of CR0[FILTER_CNT] must be 1.

41.3.1.7 Windowed/Filtered mode (#7)

This is the most complex mode of operation for the comparator block, as it uses both windowing and filtering features. It also has the highest latency of any of the modes. This can be approximated: up to 1 bus clock synchronization in the window function + ((CR0[FILTER_CNT] * FPR[FILT_PER]) + 1) * bus clock for the filter function.

When any windowed mode is active, COUTA is clocked by the bus clock whenever WINDOW = 1. The last latched value is held when WINDOW = 0.

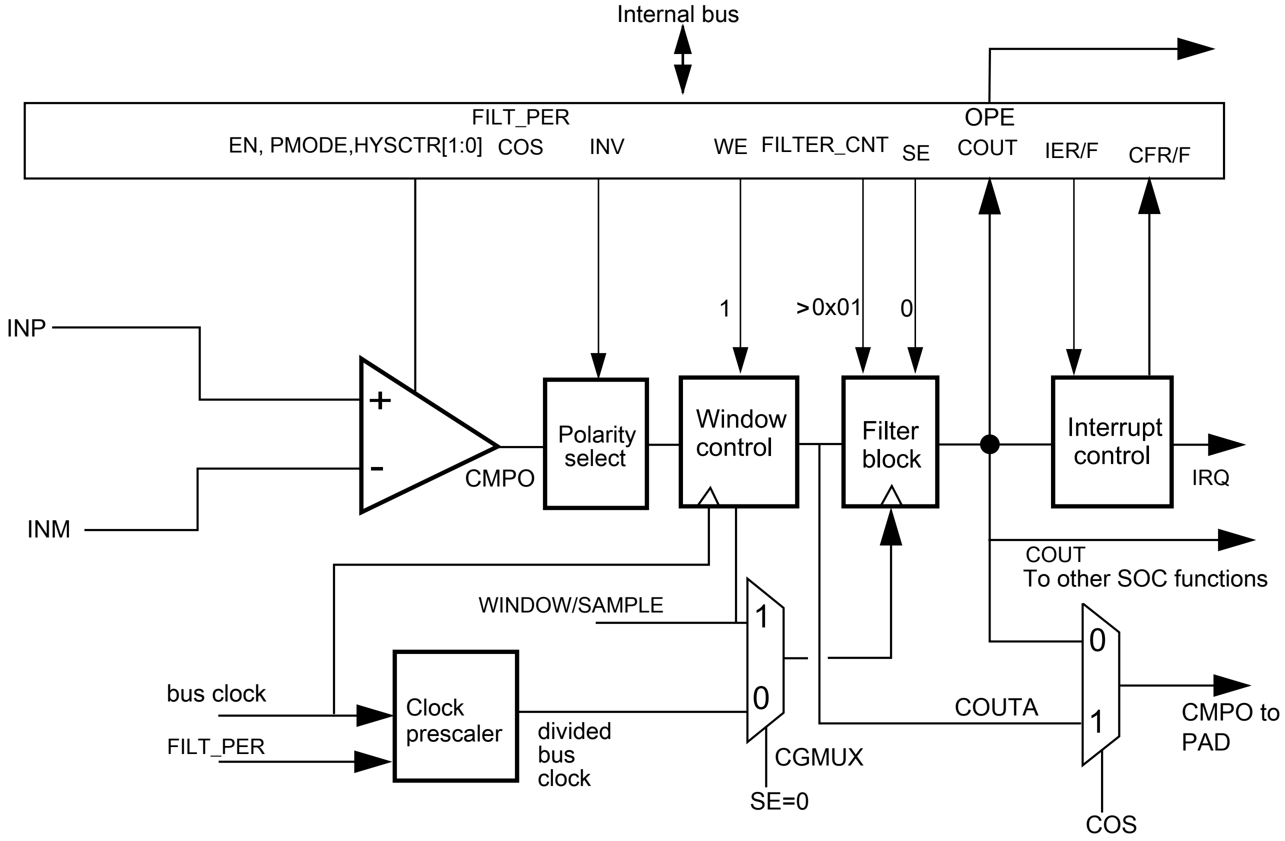


Figure 41-11. Windowed/Filtered mode

41.3.2 Power modes

41.3.2.1 Wait mode operation

During Wait and VLPW modes, the CMP, if enabled, continues to operate normally and a CMP interrupt can wake the MCU.

41.3.2.2 Stop mode operation

Depending on clock restrictions related to the MCU core or core peripherals, the MCU is brought out of stop when a compare event occurs and the corresponding interrupt is enabled. Similarly, if CR1[OPE] is enabled, the comparator output operates as in the normal operating mode and comparator output is placed onto the external pin. In Stop modes, the comparator can be operational in both:

- High-Speed (HS) Comparison mode when CR1[PMODE] = 1
- Low-Speed (LS) Comparison mode when CR1[PMODE] = 0

It is recommended to use the LS mode to minimize power consumption.

If stop is exited with a reset, all comparator registers are put into their reset state.

41.3.2.3 Low-Leakage mode operation

When the chip is in Low-Leakage modes:

- The CMP module is partially functional and is limited to Low-Speed mode, regardless of CR1[PMODE] setting
- Windowed, Sampled, and Filtered modes are not supported
- The CMP output pin is latched and does not reflect the compare output state.

The positive- and negative-input voltage can be supplied from external pins or the DAC output. The MCU can be brought out of the Low-Leakage mode if a compare event occurs and the CMP interrupt is enabled. After wakeup from low-leakage modes, the CMP module is in the reset state except for SCR[CFF] and SCR[CFR].

41.3.3 Startup and operation

A typical startup sequence is listed here.

- The time required to stabilize COUT will be the power-on delay of the comparators plus the largest propagation delay from a selected analog source through the analog comparator, windowing function and filter. See the Data Sheets for power-on delays of the comparators. The windowing function has a maximum of one bus clock period delay. The filter delay is specified in the Low-pass filter.
- During operation, the propagation delay of the selected data paths must always be considered. It may take many bus clock cycles for COUT and SCR[CFR]/SCR[CFF]

to reflect an input change or a configuration change to one of the components involved in the data path.

- When programmed for filtering modes, COUT will initially be equal to 0, until sufficient clock cycles have elapsed to fill all stages of the filter. This occurs even if COUTA is at a logic 1.

41.3.4 Low-pass filter

The low-pass filter operates on the unfiltered and unsynchronized and optionally inverted comparator output COUTA and generates the filtered and synchronized output COUT.

Both COUTA and COUT can be configured as module outputs and are used for different purposes within the system.

Synchronization and edge detection are always used to determine status register bit values. They also apply to COUT for all sampling and windowed modes. Filtering can be performed using an internal timebase defined by FPR[FILT_PER], or using an external SAMPLE input to determine sample time.

The need for digital filtering and the amount of filtering is dependent on user requirements. Filtering can become more useful in the absence of an external hysteresis circuit. Without external hysteresis, high-frequency oscillations can be generated at COUTA when the selected INM and INP input voltages differ by less than the offset voltage of the differential comparator.

41.3.4.1 Enabling filter modes

Filter modes can be enabled by:

- Setting CR0[FILTER_CNT] > 0x01 and
- Setting FPR[FILT_PER] to a nonzero value or setting CR1[SE]=1

If using the divided bus clock to drive the filter, it will take samples of COUTA every FPR[FILT_PER] bus clock cycles.

The filter output will be at logic 0 when first initalized, and will subsequently change when all the consecutive CR0[FILTER_CNT] samples agree that the output value has changed. In other words, SCR[COUT] will be 0 for some initial period, even when COUTA is at logic 1.

Setting both CR1[SE] and FPR[FILT_PER] to 0 disables the filter and eliminates switching current associated with the filtering process.

Note

> Always switch to this setting prior to making any changes in filter parameters. This resets the filter to a known state. Switching CR0[FILTER_CNT] on the fly without this intermediate step can result in unexpected behavior.

If CR1[SE]=1, the filter takes samples of COUTA on each positive transition of the sample input. The output state of the filter changes when all the consecutive CR0[FILTER_CNT] samples agree that the output value has changed.

41.3.4.2 Latency issues

The value of FPR[FILT_PER] or SAMPLE period must be set such that the sampling period is just longer than the period of the expected noise. This way a noise spike will corrupt only one sample. The value of CR0[FILTER_CNT] must be chosen to reduce the probability of noisy samples causing an incorrect transition to be recognized. The probability of an incorrect transition is defined as the probability of an incorrect sample raised to the power of CR0[FILTER_CNT].

The values of FPR[FILT_PER] or SAMPLE period and CR0[FILTER_CNT] must also be traded off against the desire for minimal latency in recognizing actual comparator output transitions. The probability of detecting an actual output change within the nominal latency is the probability of a correct sample raised to the power of CR0[FILTER_CNT].

The following table summarizes maximum latency values for the various modes of operation *in the absence of noise*. Filtering latency is restarted each time an actual output transition is masked by noise.

Table 41-2. Comparator sample/filter maximum latencies

Mode #	CR1[EN]	CR1[WE]	CR1[SE]	CR0[FILTER_CNT]	FPR[FILT_PER]	Operation	Maximum latency[1]
1	0	X	X	X	X	Disabled	N/A
2A	1	0	0	0x00	X	Continuous Mode	T_{PD}
2B	1	0	0	X	0x00		
3A	1	0	1	0x01	X	Sampled, Non-Filtered mode	T_{PD} + T_{SAMPLE} + T_{per}
3B	1	0	0	0x01	> 0x00		T_{PD} + (FPR[FILT_PER] * T_{per}) + T_{per}
4A	1	0	1	> 0x01	X	Sampled, Filtered mode	T_{PD} + (CR0[FILTER_CNT] * T_{SAMPLE}) + T_{per}
4B	1	0	0	> 0x01	> 0x00		T_{PD} + (CR0[FILTER_CNT] * FPR[FILT_PER] x T_{per}) + T_{per}

Table continues on the next page...

Table 41-2. Comparator sample/filter maximum latencies (continued)

Mode #	CR1[EN]	CR1[WE]	CR1[SE]	CR0[FILTER_CNT]	FPR[FILT_PER]	Operation	Maximum latency[1]
5A	1	1	0	0x00	X	Windowed mode	$T_{PD} + T_{per}$
5B	1	1	0	X	0x00		$T_{PD} + T_{per}$
6	1	1	0	0x01	0x01 - 0xFF	Windowed / Resampled mode	T_{PD} + (FPR[FILT_PER] * T_{per}) + $2T_{per}$
7	1	1	0	> 0x01	0x01 - 0xFF	Windowed / Filtered mode	T_{PD} + (CR0[FILTER_CNT] * FPR[FILT_PER] x T_{per}) + $2T_{per}$

[1] T_{PD} represents the intrinsic delay of the analog component plus the polarity select logic. T_{SAMPLE} is the clock period of the external sample clock. T_{per} is the period of the bus clock.

41.4 CMP interrupts

The CMP module is capable of generating an interrupt on either the rising- or falling-edge of the comparator output, or both.

The following table gives the conditions in which the interrupt request is asserted and deasserted.

When	Then
SCR[IER] and SCR[CFR] are set	The interrupt request is asserted
SCR[IEF] and SCR[CFF] are set	The interrupt request is asserted
SCR[IER] and SCR[CFR] are cleared for a rising-edge interrupt	The interrupt request is deasserted
SCR[IEF] and SCR[CFF] are cleared for a falling-edge interrupt	The interrupt request is deasserted

41.5 DMA support

Normally, the CMP generates a CPU interrupt if there is a change on the COUT. When DMA support is enabled by setting SCR[DMAEN] and the interrupt is enabled by setting SCR[IER], SCR[IEF], or both, the corresponding change on COUT forces a DMA transfer request rather than a CPU interrupt instead. When the DMA has completed the transfer, it sends a transfer completing indicator that deasserts the DMA transfer request and clears the flag to allow a subsequent change on comparator output to occur and force another DMA request.

The comparator can remain functional in STOP modes.

When DMA support is enabled by setting SCR[DMAEN] and the interrupt is enabled by setting SCR[IER], SCR[IEF], or both, the corresponding change on COUT forces a DMA transfer request to wake up the system from STOP modes. After the data transfer has finished, system will go back to STOP modes. Refer to DMA chapters in the device reference manual for the asynchronous DMA function for details.

41.6 Digital-to-analog converter

The figure found here shows the block diagram of the DAC module.

It contains a 64-tap resistor ladder network and a 64-to-1 multiplexer, which selects an output voltage from one of 64 distinct levels that outputs from DACO. It is controlled through the DAC Control Register (DACCR). Its supply reference source can be selected from two sources V_{in1} and V_{in2}. The module can be powered down or disabled when not in use. When in Disabled mode, DACO is connected to the analog ground.

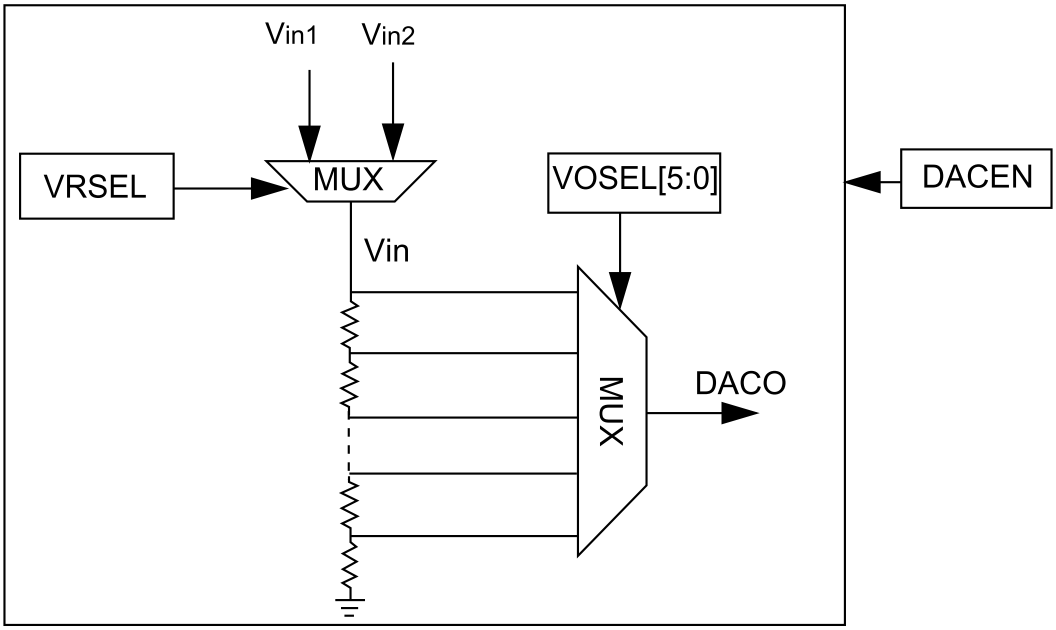


Figure 41-12. 6-bit DAC block diagram

41.7 DAC functional description

This section provides DAC functional description information.

41.7.1 Voltage reference source select

- V_{in1} connects to the primary voltage source as supply reference of 64 tap resistor ladder
- V_{in2} connects to an alternate voltage source

41.8 DAC resets

This module has a single reset input, corresponding to the chip-wide peripheral reset.

41.9 DAC clocks

This module has a single clock input, the bus clock.

41.10 DAC interrupts

This module has no interrupts.

Chapter 42
MCU: Programmable Delay Block (PDB)

42.1 Introduction

NOTE

For the chip-specific implementation details of this module's instances, see the chip configuration information.

The Programmable Delay Block (PDB) provides controllable delays from either an internal or an external trigger, or a programmable interval tick, to the hardware trigger inputs of ADCs and/or generates the interval triggers to DACs, so that the precise timing between ADC conversions and/or DAC updates can be achieved. The PDB can optionally provide pulse outputs (Pulse-Out's) that are used as the sample window in the CMP block.

42.1.1 Features

- Up to 15 trigger input sources and one software trigger source
- Up to 8 configurable PDB channels for ADC hardware trigger
  - One PDB channel is associated with one ADC
  - One trigger output for ADC hardware trigger and up to 8 pre-trigger outputs for ADC trigger select per PDB channel
  - Trigger outputs can be enabled or disabled independently
  - One 16-bit delay register per pre-trigger output
  - Optional bypass of the delay registers of the pre-trigger outputs
  - Operation in One-Shot or Continuous modes

  - Optional back-to-back mode operation, which enables the ADC conversions complete to trigger the next PDB channel
  - One programmable delay interrupt
  - One sequence error interrupt
  - One channel flag and one sequence error flag per pre-trigger
  - DMA support
- Up to 8 pulse outputs (pulse-out's)
  - Pulse-out's can be enabled or disabled independently
  - Programmable pulse width

NOTE

> The number of PDB input and output triggers are chip-specific. See the chip-specific PDB information for details.

42.1.2 Implementation

In this section, the following letters refer to the number of output triggers:

- N—Total available number of PDB channels.
- n—PDB channel number, valid from 0 to *N*-1.
- M—Total available pre-trigger per PDB channel.
- m—Pre-trigger number, valid from 0 to *M*-1.
- X—Total number of DAC interval triggers.
- x—DAC interval trigger output number, valid from 0 to *X*-1.
- Y—Total number of Pulse-Out's.
- y—Pulse-Out number, valid value is from 0 to *Y*-1.

NOTE

> The number of module output triggers to core is chip-specific. For module to core output triggers implementation, see the chip configuration information.

42.1.3 Back-to-back acknowledgment connections

PDB back-to-back operation acknowledgment connections are chip-specific. For implementation, see the chip configuration information.

42.1.4 DAC External Trigger Input Connections

The implementation of DAC external trigger inputs is chip-specific. See the chip configuration information for details.

42.1.5 Block diagram

This diagram illustrates the major components of the PDB.

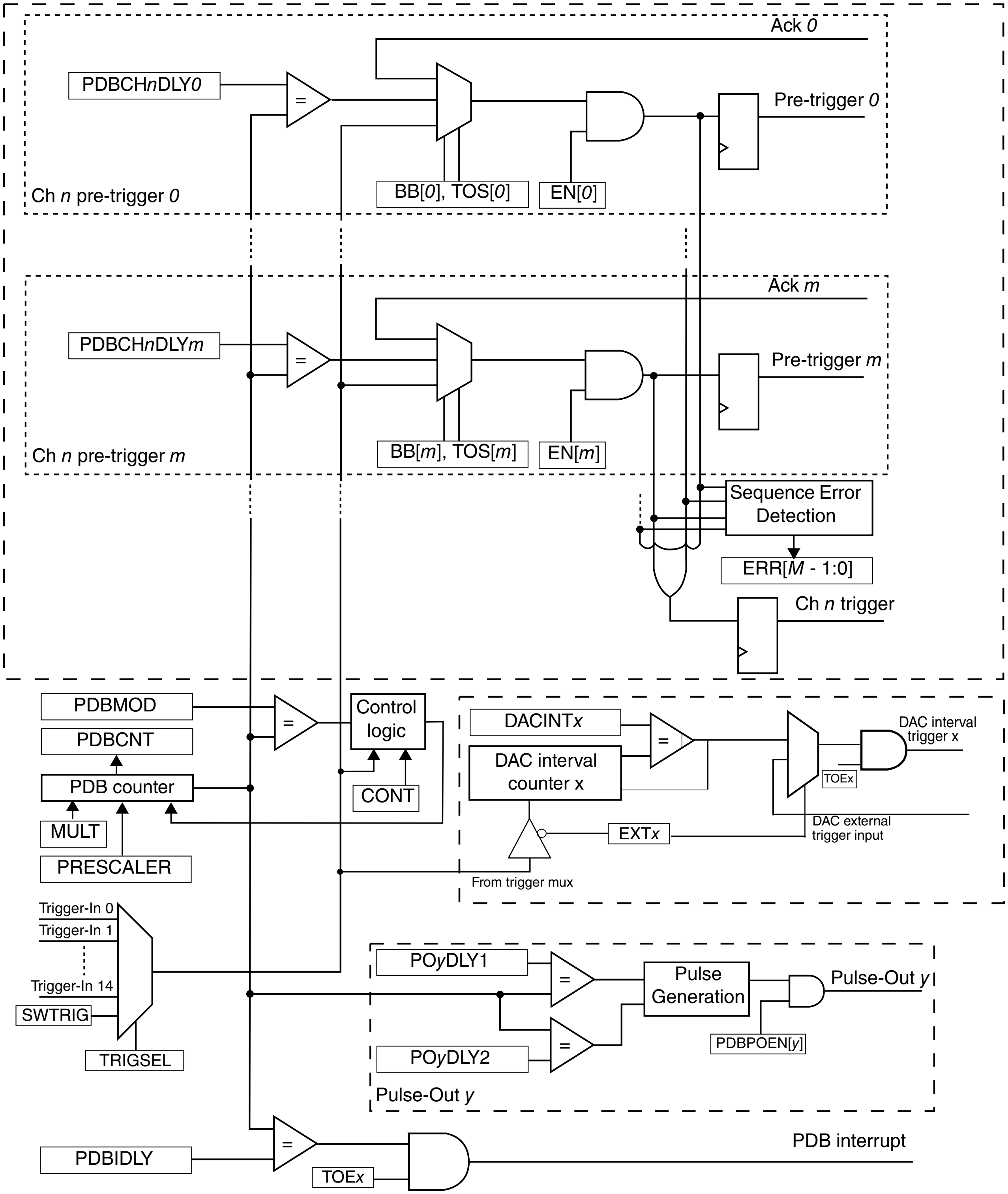


Figure 42-1. PDB block diagram

In this diagram, only one PDB channel *n*, one DAC interval trigger *x*, and one Pulse-Out *y* are shown. The PDB-enabled control logic and the sequence error interrupt logic are not shown.

42.1.6 Modes of operation

PDB ADC trigger operates in the following modes:

- Disabled—Counter is off, all pre-trigger and trigger outputs are low if PDB is not in back-to-back operation of Bypass mode.
- Debug—Counter is paused when processor is in Debug mode, and the counter for the DAC trigger is also paused in Debug mode.
- Enabled One-Shot—Counter is enabled and restarted at count zero upon receiving a positive edge on the selected trigger input source or software trigger is selected and SC[SWTRIG] is written with 1. In each PDB channel, an enabled pre-trigger asserts once per trigger input event. The trigger output asserts whenever any of the pre-triggers is asserted.
- Enabled Continuous—Counter is enabled and restarted at count zero. The counter is rolled over to zero again when the count reaches the value specified in the modulus register, and the counting is restarted. This enables a continuous stream of pre-triggers/trigger outputs as a result of a single trigger input event.
- Enabled Bypassed—The pre-trigger and trigger outputs assert immediately after a positive edge on the selected trigger input source or software trigger is selected and SC[SWTRIG] is written with 1, that is the delay registers are bypassed. It is possible to bypass any one or more of the delay registers; therefore, this mode can be used in conjunction with One-Shot or Continuous mode.

42.2 PDB signal descriptions

This table shows the detailed description of the external signal.

Table 42-1. PDB signal descriptions

Signal	Description	I/O
EXTRG	External Trigger Input Source If the PDB is enabled and external trigger input source is selected, a positive edge on the EXTRG signal resets and starts the counter.	I

42.3 Memory map and register definition

PDB memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4003_6000	Status and Control register (PDB0_SC)	32	R/W	0000_0000h	42.3.1/947
4003_6004	Modulus register (PDB0_MOD)	32	R/W	0000_FFFFh	42.3.2/950
4003_6008	Counter register (PDB0_CNT)	32	R	0000_0000h	42.3.3/950
4003_600C	Interrupt Delay register (PDB0_IDLY)	32	R/W	0000_FFFFh	42.3.4/951
4003_6010	Channel n Control register 1 (PDB0_CH0C1)	32	R/W	0000_0000h	42.3.5/951
4003_6014	Channel n Status register (PDB0_CH0S)	32	R/W	0000_0000h	42.3.6/952
4003_6018	Channel n Delay 0 register (PDB0_CH0DLY0)	32	R/W	0000_0000h	42.3.7/953
4003_601C	Channel n Delay 1 register (PDB0_CH0DLY1)	32	R/W	0000_0000h	42.3.8/954
4003_6038	Channel n Control register 1 (PDB0_CH1C1)	32	R/W	0000_0000h	42.3.5/951
4003_603C	Channel n Status register (PDB0_CH1S)	32	R/W	0000_0000h	42.3.6/952
4003_6040	Channel n Delay 0 register (PDB0_CH1DLY0)	32	R/W	0000_0000h	42.3.7/953
4003_6044	Channel n Delay 1 register (PDB0_CH1DLY1)	32	R/W	0000_0000h	42.3.8/954
4003_6150	DAC Interval Trigger n Control register (PDB0_DACINTC0)	32	R/W	0000_0000h	42.3.9/954
4003_6154	DAC Interval n register (PDB0_DACINT0)	32	R/W	0000_0000h	42.3.10/ 955
4003_6190	Pulse-Out n Enable register (PDB0_POEN)	32	R/W	0000_0000h	42.3.11/ 956
4003_6194	Pulse-Out n Delay register (PDB0_PO0DLY)	32	R/W	0000_0000h	42.3.12/ 956
4003_6198	Pulse-Out n Delay register (PDB0_PO1DLY)	32	R/W	0000_0000h	42.3.12/ 956

42.3.1 Status and Control register (PDBx_SC)

Address: 4003_6000h base + 0h offset = 4003_6000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0												LDMOD		PDBEIE	0
W																SWTRIG
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	DMAEN	PRESCALER			TRGSEL				PDBEN	PDBIF	PDBIE	0	MULT		CONT	LDOK
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PDBx_SC field descriptions

Field	Description
31–20 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
19–18 LDMOD	Load Mode Select Selects the mode to load the MOD, IDLY, CH*n*DLY*m*, INT*x*, and PO*y*DLY registers, after 1 is written to LDOK. 00 The internal registers are loaded with the values from their buffers immediately after 1 is written to LDOK. 01 The internal registers are loaded with the values from their buffers when the PDB counter reaches the MOD register value after 1 is written to LDOK. 10 The internal registers are loaded with the values from their buffers when a trigger input event is detected after 1 is written to LDOK. 11 The internal registers are loaded with the values from their buffers when either the PDB counter reaches the MOD register value or a trigger input event is detected, after 1 is written to LDOK.
17 PDBEIE	PDB Sequence Error Interrupt Enable Enables the PDB sequence error interrupt. When this field is set, any of the PDB channel sequence error flags generates a PDB sequence error interrupt.

Table continues on the next page...

PDBx_SC field descriptions (continued)

Field	Description
	0 PDB sequence error interrupt disabled. 1 PDB sequence error interrupt enabled.
16 SWTRIG	Software Trigger When PDB is enabled and the software trigger is selected as the trigger input source, writing 1 to this field resets and restarts the counter. Writing 0 to this field has no effect. Reading this field results 0.
15 DMAEN	DMA Enable When DMA is enabled, the PDBIF flag generates a DMA request instead of an interrupt. 0 DMA disabled. 1 DMA enabled.
14–12 PRESCALER	Prescaler Divider Select 000 Counting uses the peripheral clock divided by multiplication factor selected by MULT. 001 Counting uses the peripheral clock divided by twice of the multiplication factor selected by MULT. 010 Counting uses the peripheral clock divided by four times of the multiplication factor selected by MULT. 011 Counting uses the peripheral clock divided by eight times of the multiplication factor selected by MULT. 100 Counting uses the peripheral clock divided by 16 times of the multiplication factor selected by MULT. 101 Counting uses the peripheral clock divided by 32 times of the multiplication factor selected by MULT. 110 Counting uses the peripheral clock divided by 64 times of the multiplication factor selected by MULT. 111 Counting uses the peripheral clock divided by 128 times of the multiplication factor selected by MULT.
11–8 TRGSEL	Trigger Input Source Select Selects the trigger input source for the PDB. The trigger input source can be internal or external (EXTRG pin), or the software trigger. Refer to chip configuration details for the actual PDB input trigger connections. 0000 Trigger-In 0 is selected. 0001 Trigger-In 1 is selected. 0010 Trigger-In 2 is selected. 0011 Trigger-In 3 is selected. 0100 Trigger-In 4 is selected. 0101 Trigger-In 5 is selected. 0110 Trigger-In 6 is selected. 0111 Trigger-In 7 is selected. 1000 Trigger-In 8 is selected. 1001 Trigger-In 9 is selected. 1010 Trigger-In 10 is selected. 1011 Trigger-In 11 is selected. 1100 Trigger-In 12 is selected. 1101 Trigger-In 13 is selected. 1110 Trigger-In 14 is selected. 1111 Software trigger is selected.

Table continues on the next page...

PDBx_SC field descriptions (continued)

Field	Description
7 PDBEN	PDB Enable 0 PDB disabled. Counter is off. 1 PDB enabled.
6 PDBIF	PDB Interrupt Flag This field is set when the counter value is equal to the IDLY register. Writing zero clears this field.
5 PDBIE	PDB Interrupt Enable Enables the PDB interrupt. When this field is set and DMAEN is cleared, PDBIF generates a PDB interrupt. 0 PDB interrupt disabled. 1 PDB interrupt enabled.
4 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
3–2 MULT	Multiplication Factor Select for Prescaler Selects the multiplication factor of the prescaler divider for the counter clock. 00 Multiplication factor is 1. 01 Multiplication factor is 10. 10 Multiplication factor is 20. 11 Multiplication factor is 40.
1 CONT	Continuous Mode Enable Enables the PDB operation in Continuous mode. 0 PDB operation in One-Shot mode 1 PDB operation in Continuous mode
0 LDOK	Load OK Writing 1 to LDOK bit updates the MOD, IDLY, CHnDLYm, DACINTx, and POyDLY registers with the values previously written to their internal buffers (and stored there). The new values of MOD, IDLY, CHnDLYm, DACINTx, and POyDLY registers will take effect according to the setting of the LDMOD field (Load Mode Select). After 1 is written to the LDOK field, the values in the internal buffers of these registers are not effective, and new values cannot be written to the internal buffers until the existing values in the internal buffers are loaded into their corresponding registers. • LDOK can be written only when PDBEN is set, or LDOK can be written at the same time when PDBEN is written to 1. • LDOK is automatically cleared when the values in the internal buffers are loaded into the registers or when PDBEN bit (PDB Enable) is cleared. • Writing 0 to LDOK has no effect.

42.3.2 Modulus register (PDBx_MOD)

Note: This register is internally buffered, and any values written to the register are written to its internal buffer instead; in other words, the internal device bus does not write directly to this register. The value in this register's internal buffer is loaded into this register only after "1" is written to the SC[LDOK] bit.

Address: 4003_6000h base + 4h offset = 4003_6004h

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

PDBx_MOD field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
MOD	PDB Modulus Specifies the period of the counter. When the counter reaches this value, it will be reset back to zero. If the PDB is in Continuous mode, the count begins anew. Reading this field returns the value of the internal register that is effective for the current cycle of PDB.

42.3.3 Counter register (PDBx_CNT)

Address: 4003_6000h base + 8h offset = 4003_6008h

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

PDBx_CNT field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
CNT	PDB Counter Contains the current value of the counter.

42.3.4 Interrupt Delay register (PDBx_IDLY)

Note: This register is internally buffered, and any values written to the register are written to its internal buffer instead; in other words, the internal device bus does not write directly to this register. The value in this register's internal buffer is loaded into this register only after "1" is written to the SC[LDOK] bit.

Address: 4003_6000h base + Ch offset = 4003_600Ch

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

PDBx_IDLY field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
IDLY	PDB Interrupt Delay Specifies the delay value to schedule the PDB interrupt. It can be used to schedule an independent interrupt at some point in the PDB cycle. If enabled, a PDB interrupt is generated, when the counter is equal to the IDLY. Reading this field returns the value of internal register that is effective for the current cycle of the PDB.

42.3.5 Channel n Control register 1 (PDBx_CH*n*C1)

Each PDB channel has one control register, CHnC1. The bits in this register control the functionality of each PDB channel operation.

Address: 4003_6000h base + 10h offset + (40d × i), where i=0d to 1d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								BB								TOS								EN							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PDBx_CH*n*C1 field descriptions

Field	Description
31–24 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
23–16 BB	PDB Channel Pre-Trigger Back-to-Back Operation Enable

Table continues on the next page...

PDBx_CH*n*C1 field descriptions (continued)

Field	Description
	These bits enable the PDB ADC pre-trigger operation as back-to-back mode. Only lower M pre-trigger bits are implemented in this MCU. Back-to-back operation enables the ADC conversions complete to trigger the next PDB channel pre-trigger and trigger output, so that the ADC conversions can be triggered on next set of configuration and results registers. Application code must only enable the back-to-back operation of the PDB pre-triggers at the leading of the back-to-back connection chain. 0 PDB channel's corresponding pre-trigger back-to-back operation disabled. 1 PDB channel's corresponding pre-trigger back-to-back operation enabled.
15–8 TOS	PDB Channel Pre-Trigger Output Select Selects the PDB ADC pre-trigger outputs. Only lower M pre-trigger fields are implemented in this MCU. 0 PDB channel's corresponding pre-trigger is in bypassed mode. The pre-trigger asserts one peripheral clock cycle after a rising edge is detected on selected trigger input source or software trigger is selected and SWTRIG is written with 1. 1 PDB channel's corresponding pre-trigger asserts when the counter reaches the channel delay register and one peripheral clock cycle after a rising edge is detected on selected trigger input source or software trigger is selected and SETRIG is written with 1.
EN	PDB Channel Pre-Trigger Enable These bits enable the PDB ADC pre-trigger outputs. Only lower M pre-trigger bits are implemented in this MCU. 0 PDB channel's corresponding pre-trigger disabled. 1 PDB channel's corresponding pre-trigger enabled.

42.3.6 Channel n Status register (PDBx_CH*n*S)

Address: 4003_6000h base + 14h offset + (40d × i), where i=0d to 1d

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

PDBx_CH*n*S field descriptions

Field	Description
31–24 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
23–16 CF	PDB Channel Flags The CF[*m*] bit is set when the PDB counter matches the CH*n*DLY*m*. Write 0 to clear these bits.
15–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
ERR	PDB Channel Sequence Error Flags Only the lower M bits are implemented in this MCU.

Table continues on the next page...

PDBx_CH*n*S field descriptions (continued)

Field	Description
	0 Sequence error not detected on PDB channel's corresponding pre-trigger. 1 Sequence error detected on PDB channel's corresponding pre-trigger. ADCn block can be triggered for a conversion by one pre-trigger from PDB channel *n*. When one conversion, which is triggered by one of the pre-triggers from PDB channel *n*, is in progress, new trigger from PDB channel's corresponding pre-trigger m cannot be accepted by ADCn, and ERR[m] is set. Writing 0's to clear the sequence error flags.

42.3.7 Channel n Delay 0 register (PDBx_CH*n*DLY0)

Note: This register is internally buffered, and any values written to the register are written to its internal buffer instead; in other words, the internal device bus does not write directly to this register. The value in this register's internal buffer is loaded into this register only after "1" is written to the SC[LDOK] bit.

Address: 4003_6000h base + 18h offset + (40d × i), where i=0d to 1d

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

PDBx_CH*n*DLY0 field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
DLY	PDB Channel Delay Specifies the delay value for the channel's corresponding pre-trigger. The pre-trigger asserts when the counter is equal to DLY. Reading this field returns the value of internal register that is effective for the current PDB cycle.

42.3.8 Channel n Delay 1 register (PDBx_CH*n*DLY1)

Note: This register is internally buffered, and any values written to the register are written to its internal buffer instead; in other words, the internal device bus does not write directly to this register. The value in this register's internal buffer is loaded into this register only after "1" is written to the SC[LDOK] bit.

Address: 4003_6000h base + 1Ch offset + (40d × i), where i=0d to 1d

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

PDBx_CH*n*DLY1 field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
DLY	PDB Channel Delay These bits specify the delay value for the channel's corresponding pre-trigger. The pre-trigger asserts when the counter is equal to DLY. Reading these bits returns the value of internal register that is effective for the current PDB cycle.

42.3.9 DAC Interval Trigger n Control register (PDBx_DACINTC*n*)

Address: 4003_6000h base + 150h offset + (8d × i), where i=0d to 0d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0														EXT	TOE
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PDBx_DACINTC*n* field descriptions

Field	Description
31–2 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
1 EXT	DAC External Trigger Input Enable

Table continues on the next page...

PDBx_DACINTC*n* field descriptions (continued)

Field	Description
	This bit enables the external trigger for DAC interval counter. 0 DAC external trigger input disabled. DAC interval counter is reset and started counting when a rising edge is detected on selected trigger input source or software trigger is selected and SWTRIG is written with 1. 1 DAC external trigger input enabled. DAC interval counter is bypassed and DAC external trigger input triggers the DAC interval trigger.
0 TOE	DAC Interval Trigger Enable Enables the DAC interval trigger. 0 DAC interval trigger disabled. 1 DAC interval trigger enabled.

42.3.10 DAC Interval n register (PDBx_DACINT*n*)

Note: This register is internally buffered, and any values written to the register are written to its internal buffer instead; in other words, the internal device bus does not write directly to this register. The value in this register's internal buffer is loaded into this register only after "1" is written to the SC[LDOK] bit.

Address: 4003_6000h base + 154h offset + (8d × i), where i=0d to 0d

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

PDBx_DACINT*n* field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
INT	DAC Interval These bits specify the interval value for DAC interval trigger. DAC interval trigger triggers DAC[1:0] update when the DAC interval counter is equal to the DACINT. Reading these bits returns the value of internal register that is effective for the current PDB cycle.

42.3.11 Pulse-Out n Enable register (PDBx_POEN)

Address: 4003_6000h base + 190h offset = 4003_6190h

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

PDBx_POEN field descriptions

Field	Description
31–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
POEN	PDB Pulse-Out Enable Enables the pulse output. Only lower Y bits are implemented in this MCU. 0 PDB Pulse-Out disabled 1 PDB Pulse-Out enabled

42.3.12 Pulse-Out n Delay register (PDBx_PO*n*DLY)

Note: This register is internally buffered, and any values written to the register are written to its internal buffer instead; in other words, the internal device bus does not write directly to this register. The value in this register's internal buffer is loaded into this register only after "1" is written to the SC[LDOK] bit.

Address: 4003_6000h base + 194h offset + (4d × i), where i=0d to 1d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DLY1																DLY2															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PDBx_PO*n*DLY field descriptions

Field	Description
31–16 DLY1	PDB Pulse-Out Delay 1 Specifies the delay 1 value for the PDB Pulse-Out. Pulse-Out goes high when the PDB counter is equal to the DLY1. Reading this field returns the value of internal register that is effective for the current PDB cycle.
DLY2	PDB Pulse-Out Delay 2 Specifies the delay 2 value for the PDB Pulse-Out. Pulse-Out goes low when the PDB counter is equal to the DLY2. Reading this field returns the value of internal register that is effective for the current PDB cycle.

42.4 Functional description

42.4.1 PDB pre-trigger and trigger outputs

The PDB contains a counter whose output is compared to several different digital values. If the PDB is enabled, then a trigger input event will reset the counter and make it start to count. A trigger input event is defined as a rising edge being detected on a selected trigger input source, or if a software trigger is selected and SC[SWTRIG] is written with 1. For each channel, a delay *m* determines the time between assertion of the trigger input event to the time at which changes in the pre-trigger *m* output signal are started. The time is defined as:

- Trigger input event to pre-trigger *m* = (prescaler × multiplication factor × delay *m*) + 2 peripheral clock cycles
- Add 1 additional peripheral clock cycle to determine the time when the channel trigger output changes.

Each channel is associated with 1 ADC block. PDB channel n pre-trigger outputs 0 to *M*; each pre-trigger output is connected to ADC hardware trigger select and hardware trigger inputs. The pre-triggers are used to precondition the ADC block before the actual trigger occurs. When the ADC receives the rising edge of the trigger, the ADC will start the conversion according to the precondition determined by the pre-triggers. The ADC contains *M* sets of configuration and result registers, allowing it to alternate conversions between *M* different analog sources (like a ping-pong game). The pre-trigger outputs are used to specify which signal will be sampled next. When a pre-trigger *m* is asserted, the ADC conversion is triggered with set *m* of the configuration and result registers.

The waveforms shown in the following diagram show the pre-trigger and trigger outputs of PDB channel *n*. The delays can be independently set using the CH*n*DLY*m* registers, and the pre-triggers can be enabled or disabled in CH*n*C1[EN[*m*]].

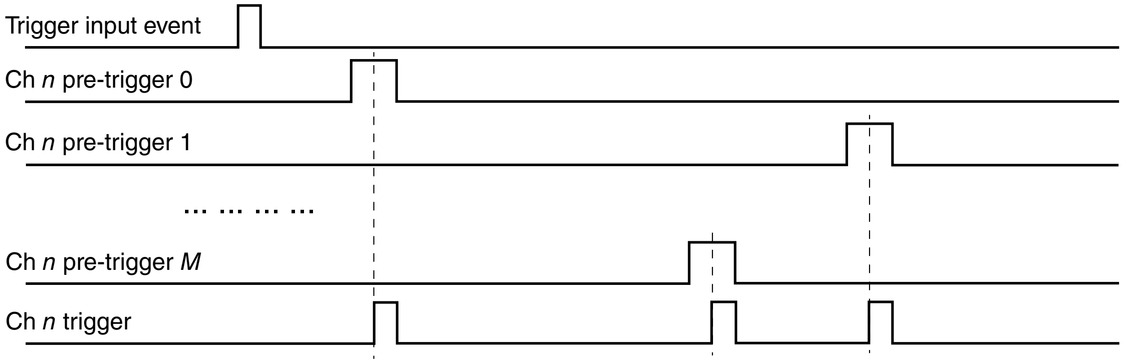


Figure 42-2. Pre-trigger and trigger outputs

The delay in CH*n*DLY*m* register can be optionally bypassed, if CH*n*C1[TOS[*m*]] is cleared. In this case, when the trigger input event occurs, the pre-trigger *m* is asserted after 2 peripheral clock cycles.

The PDB can be configured for back-to-back operation. Back-to-back operation enables the ADC conversion completions to trigger the next PDB channel pre-trigger and trigger outputs, so that the ADC conversions can be triggered on the next set of configuration and results registers. When back-to-back operation is enabled by setting CH*n*C1[BB[*m*]], then the delay *m* is ignored and the pre-trigger *m* is asserted 2 peripheral cycles after the acknowledgment *m* is received. The acknowledgment connections in this MCU are described in Back-to-back acknowledgment connections.

When a pre-trigger from a PDB channel *n* is asserted, the associated lock of the pre-trigger becomes active. The associated lock is released by the rising edge of the corresponding ADC*n*SC1[COCO]; the ADC*n*SC1[COCO] should be cleared after the conversion result is read, so that the next rising edge of ADC*n*SC1[COCO] can be generated to clear the lock later. The lock becomes inactive when:

- the rising edge of corresponding ADC*n*SC1[COCO] occurs,
- or the corresponding PDB pre-trigger is disabled,
- or the PDB is disabled

The channel *n* trigger output is suppressed when any of the locks of the pre-triggers in channel *n* is active. If a new pre-trigger *m* asserts when there is active lock in the PDB channel *n*, then a register flag bit CH*n*S[ERR[*m*]] (associated with the pre-trigger *m*) is set. If SC[PDBEIE] is set, then the sequence error interrupt is generated. A sequence error typically happens because the delay *m* is set too short and the pre-trigger *m* asserts before the previously triggered ADC conversion finishes.

When the PDB counter reaches the value set in IDLY register, the SC[PDBIF] flag is set. A PDB interrupt can be generated if SC[PDBIE] is set and SC[DMAEN] is cleared. If SC[DMAEN] is set, then the PDB requests a DMA transfer when the SC[PDBIF] flag is set.

The modulus value in the MOD register is used to reset the counter back to zero at the end of the count. If SC[CONT] is set, then the counter will then resume a new count; otherwise, the counter operation will stop until the next trigger input event occurs.

42.4.2 PDB trigger input source selection

The PDB has up to 15 trigger input sources, namely Trigger-In 0 to Trigger-In 14. They are connected to on-chip or off-chip event sources. The PDB can be triggered by software through SC[SWTRIG].

For the trigger input sources implemented in this MCU, see chip configuration information.

42.4.3 Pulse-Out's

PDB can generate pulse outputs of configurable width.

- When the PDB counter reaches the value set in PO*y*DLY[DLY1], then the Pulse-Out goes high.
- When the PDB counter reaches PO*y*DLY[DLY2], then it goes low.

PO*y*DLY[DLY2] can be set either greater or less than PO*y*DLY[DLY1].

ADC pre-trigger/trigger outputs and Pulse-Out generation have the same time base, because they both share the PDB counter. The pulse-out connections implemented in this MCU are described in the device's chip configuration details.

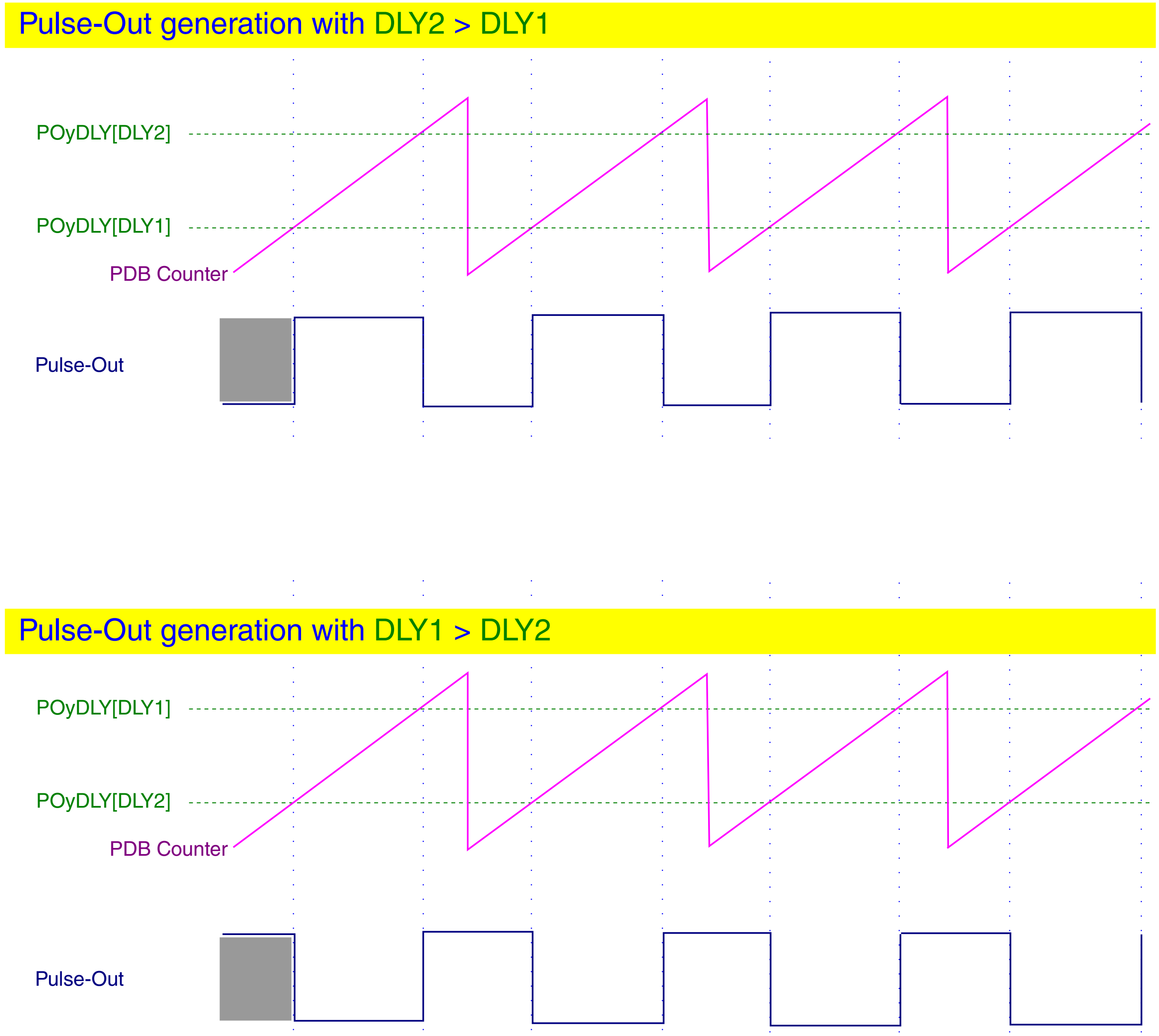


Figure 42-3. How Pulse Out is generated

42.4.4 Updating the delay registers

The following registers control the timing of the PDB operation; and in some of the applications, they may need to become effective at the same time.

- PDB Modulus register (MOD)
- PDB Interrupt Delay register (IDLY)
- PDB Channel *n* Delay *m* register (CH*n*DLY*m*)

- DAC Interval *x* register (DACINT*x*)
- PDB Pulse-Out *y* Delay register (PO*y*DLY)

The internal registers of them are buffered and any values written to them are written first to their buffers. The circumstances that cause their internal registers to be updated with the values from the buffers are summarized as shown in the table below.

Table 42-2. Circumstances of update to the delay registers

SC[LDMOD]	Update to the delay registers
00	The internal registers are loaded with the values from their buffers immediately after 1 is written to SC[LDOK].
01	The PDB counter reaches the MOD register value after 1 is written to SC[LDOK].
10	A trigger input event is detected after 1 is written to SC[LDOK].
11	Either the PDB counter reaches the MOD register value, or a trigger input event is detected, after 1 is written to SC[LDOK].

After 1 is written to SC[LDOK], the buffers cannot be written until the values in buffers are loaded into their internal registers. SC[LDOK] is self-cleared when the internal registers are loaded, so the application code can read it to determine the updates to the internal registers.

The following diagrams show the cases of the internal registers being updated with SC[LDMOD] is 00 and x1.

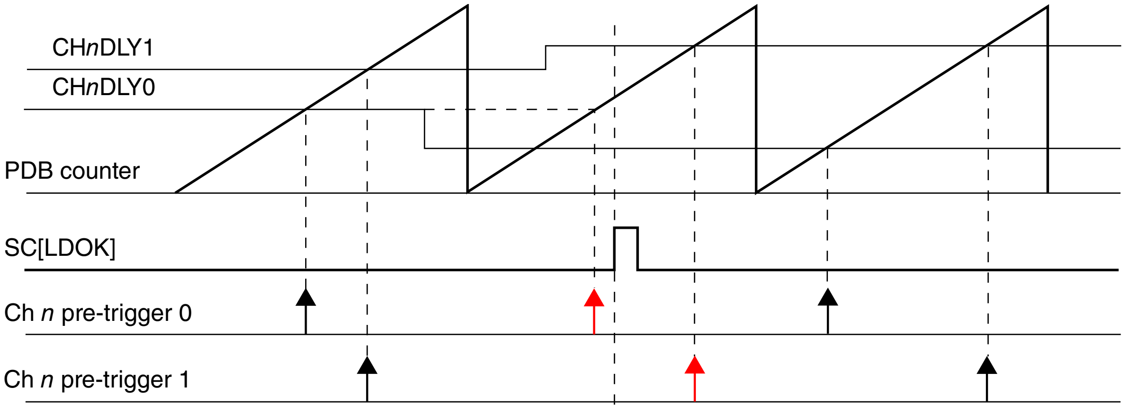


Figure 42-4. Registers update with SC[LDMOD] = 00

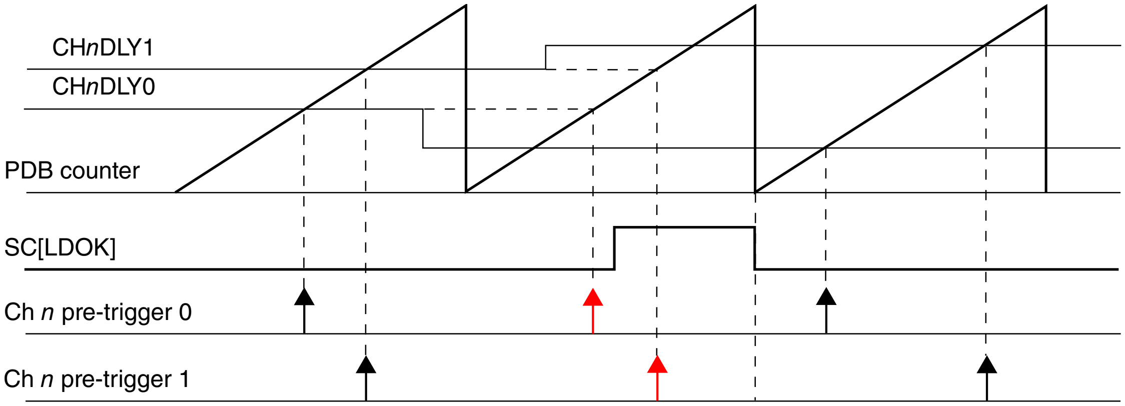


Figure 42-5. Registers update with SC[LDMOD] = x1

42.4.5 Interrupts

PDB can generate two interrupts: PDB interrupt and PDB sequence error interrupt. The following table summarizes the interrupts.

Table 42-3. PDB interrupt summary

Interrupt	Flags	Enable bit
PDB Interrupt	SC[PDBIF]	SC[PDBIE] = 1 and SC[DMAEN] = 0
PDB Sequence Error Interrupt	CH*n*S[ERR*m*]	SC[PDBEIE] = 1

42.4.6 DMA

If SC[DMAEN] is set, PDB can generate a DMA transfer request when SC[PDBIF] is set. When DMA is enabled, the PDB interrupt is not issued.

42.5 Application information

42.5.1 Impact of using the prescaler and multiplication factor on timing resolution

Use of prescaler and multiplication factor greater than 1 limits the count/delay accuracy in terms of peripheral clock cycles (to the modulus of the prescaler X multiplication factor). If the multiplication factor is set to 1 and the prescaler is set to 2 then the only values of total peripheral clocks that can be detected are even values; if prescaler is set to 4 then the only values of total peripheral clocks that can be decoded as detected are mod(4) and so forth. If the applications need a really long delay value and use a prescaler set to 128, then the resolution would be limited to 128 peripheral clock cycles.

Therefore, use the lowest possible prescaler and multiplication factor for a given application.

Chapter 43
MCU: FlexTimer Module (FTM)

43.1 Introduction

NOTE

> For the chip-specific implementation details of this module's instances, see the chip configuration information.

The FlexTimer module (FTM) is a two-to-eight channel timer that supports input capture, output compare, and the generation of PWM signals to control electric motor and power management applications. The FTM time reference is a 16-bit counter that can be used as an unsigned or signed counter.

NOTE

> The number of channels supported can vary for each instance of the FTM module on a chip. See the chip-specific FTM information to see how many channels are supported for each module instance. For example, if a module instance supports only six channels, references to channel numbers 6 and 7 do not apply for that instance.

43.1.1 FlexTimer philosophy

The FlexTimer is built upon a simple timer, the HCS08 Timer PWM Module – TPM, used for many years on Freescale's 8-bit microcontrollers. The FlexTimer extends the functionality to meet the demands of motor control, digital lighting solutions, and power conversion, while providing low cost and backwards compatibility with the TPM module.

Several key enhancements are made:

- Signed up counter
- Deadtime insertion hardware
- Fault control inputs

- Enhanced triggering functionality
- Initialization and polarity control

All of the features common with the TPM have fully backwards compatible register assignments. The FlexTimer can also use code on the same core platform without change to perform the same functions.

Motor control and power conversion features have been added through a dedicated set of registers and defaults turn off all new features. The new features, such as hardware deadtime insertion, polarity, fault control, and output forcing and masking, greatly reduce loading on the execution software and are usually each controlled by a group of registers.

FlexTimer input triggers can be from comparators, ADC, or other submodules to initiate timer functions automatically. These triggers can be linked in a variety of ways during integration of the sub modules so please note the options available for used FlexTimer configuration.

More than one FlexTimers may be synchronized to provide a larger timer with their counters incrementing in unison, assuming the initialization, the input clocks, the initial and final counting values are the same in each FlexTimer.

All main user access registers are buffered to ease the load on the executing software. A number of trigger options exist to determine which registers are updated with this user defined data.

43.1.2 Features

The FTM features include:

- FTM source clock is selectable.
  - The source clock can be the system clock or an external clock
  - Selecting external clock connects FTM clock to a chip level input pin therefore allowing to synchronize the FTM counter with an off chip clock source
- Prescaler divide-by 1, 2, 4, 8, 16, 32, 64, or 128
- 16-bit counter
  - It can be a free-running counter or a counter with initial and final value
  - The counting can be up or up-down
- Each channel can be configured for input capture, output compare, or edge-aligned PWM mode
- In Input Capture mode:

    - The capture can occur on rising edges, falling edges or both edges
    - An input filter can be selected for some channels
- In Output Compare mode the output signal can be set, cleared, or toggled on match
- All channels can be configured for center-aligned PWM mode
- Each pair of channels can be combined to generate a PWM signal with independent control of both edges of PWM signal
- The FTM channels can operate as pairs with equal outputs, pairs with complementary outputs, or independent channels with independent outputs
- The deadtime insertion is available for each complementary pair
- Generation of match triggers
- Initialization trigger
- Software control of PWM outputs
- Up to 4 fault inputs for global fault control
- The polarity of each channel is configurable
- The generation of an interrupt per channel
- The generation of an interrupt when the counter overflows
- The generation of an interrupt when the fault condition is detected
- Synchronized loading of write buffered FTM registers
- Write protection for critical registers
- Backwards compatible with TPM
- Testing of input captures for a stuck at zero and one conditions
- Dual edge capture for pulse and period width measurement
- Quadrature decoder with input filters, relative position counting, and interrupt on position count or capture of position count on external event

43.1.3 Modes of operation

When the chip is in an active BDM mode, the FTM temporarily suspends all counting until the chip returns to normal user operating mode. During Stop mode, all FTM input clocks are stopped, so the FTM is effectively disabled until clocks resume. During Wait mode, the FTM continues to operate normally. If the FTM does not need to produce a real time reference or provide the interrupt sources needed to wake the chip from Wait mode, the power can then be saved by disabling FTM functions before entering Wait mode.

43.1.4 Block diagram

The FTM uses one input/output (I/O) pin per channel, CHn (FTM channel (n)) where n is the channel number (0–7).

The following figure shows the FTM structure. The central component of the FTM is the 16-bit counter with programmable initial and final values and its counting can be up or up-down.

NOTE

The number of channels supported can vary for each instance of the FTM module on a chip. See the chip-specific FTM information to see how many channels are supported for each module instance. For example, if a module instance supports only six channels, references to channel numbers 6 and 7 do not apply for that instance.

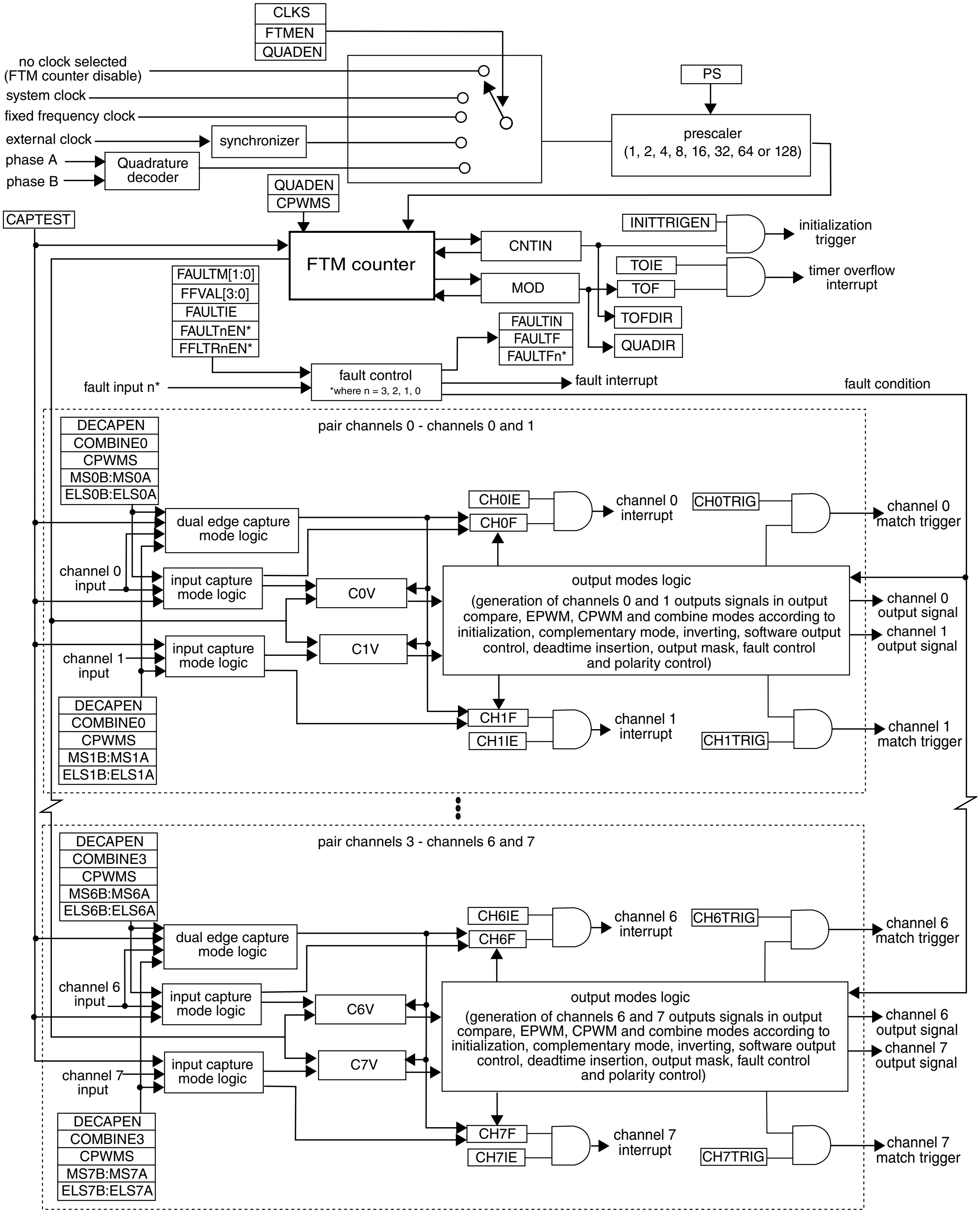


Figure 43-1. FTM block diagram

43.2 FTM signal descriptions

Table 43-1 shows the user-accessible signals for the FTM.

Table 43-1. FTM signal descriptions

Signal	Description	I/O	Function
EXTCLK	External clock. FTM external clock can be selected to drive the FTM counter.	I	The external clock input signal is used as the FTM counter clock if selected by CLKS[1:0] bits in the SC register. This clock signal must not exceed 1/4 of system clock frequency. The FTM counter prescaler selection and settings are also used when an external clock is selected.
CHn	FTM channel (n), where n can be 7-0	I/O	Each FTM channel can be configured to operate either as input or output. The direction associated with each channel, input or output, is selected according to the mode assigned for that channel.
FAULTj	Fault input (j), where j can be 3-0	I	The fault input signals are used to control the CHn channel output state. If a fault is detected, the FAULTj signal is asserted and the channel output is put in a safe state. The behavior of the fault logic is defined by the FAULTM[1:0] control bits in the MODE register and FAULTEN bit in the COMBINEm register. Note that each FAULTj input may affect all channels selectively since FAULTM[1:0] and FAULTEN control bits are defined for each pair of channels. Because there are several FAULTj inputs, maximum of 4 for the FTM module, each one of these inputs is activated by the FAULTjEN bit in the FLTCTRL register.
PHA	Quadrature decoder phase A input. Input pin associated with quadrature decoder phase A.	I	The quadrature decoder phase A input is used as the Quadrature Decoder mode is selected. The phase A input signal is one of the signals that control the FTM counter increment or decrement in the Quadrature Decoder mode.
PHB	Quadrature decoder phase B input. Input pin associated with quadrature decoder phase B.	I	The quadrature decoder phase B input is used as the Quadrature Decoder mode is selected. The phase B input signal is one of the signals that control the FTM counter increment or decrement in the Quadrature Decoder mode.

43.3 Memory map and register definition

43.3.1 Memory map

This section presents a high-level summary of the FTM registers and how they are mapped.

The registers and bits of an unavailable function in the FTM remain in the memory map and in the reset value, but they have no active function.

Note

Do not write in the region from the CNTIN register through the PWMLOAD register when FTMEN = 0.

NOTE

The number of channels supported can vary for each instance of the FTM module on a chip. See the chip-specific FTM information to see how many channels are supported for each module instance. For example, if a module instance supports only six channels, references to channel numbers 6 and 7 do not apply for that instance.

43.3.2 Register descriptions

Accesses to reserved addresses result in transfer errors. Registers for absent channels are considered reserved.

FTM memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4003_8000	Status And Control (FTM0_SC)	32	R/W	0000_0000h	43.3.3/976
4003_8004	Counter (FTM0_CNT)	32	R/W	0000_0000h	43.3.4/977
4003_8008	Modulo (FTM0_MOD)	32	R/W	0000_0000h	43.3.5/978
4003_800C	Channel (n) Status And Control (FTM0_C0SC)	32	R/W	0000_0000h	43.3.6/979
4003_8010	Channel (n) Value (FTM0_C0V)	32	R/W	0000_0000h	43.3.7/981
4003_8014	Channel (n) Status And Control (FTM0_C1SC)	32	R/W	0000_0000h	43.3.6/979
4003_8018	Channel (n) Value (FTM0_C1V)	32	R/W	0000_0000h	43.3.7/981
4003_801C	Channel (n) Status And Control (FTM0_C2SC)	32	R/W	0000_0000h	43.3.6/979
4003_8020	Channel (n) Value (FTM0_C2V)	32	R/W	0000_0000h	43.3.7/981
4003_8024	Channel (n) Status And Control (FTM0_C3SC)	32	R/W	0000_0000h	43.3.6/979
4003_8028	Channel (n) Value (FTM0_C3V)	32	R/W	0000_0000h	43.3.7/981
4003_802C	Channel (n) Status And Control (FTM0_C4SC)	32	R/W	0000_0000h	43.3.6/979
4003_8030	Channel (n) Value (FTM0_C4V)	32	R/W	0000_0000h	43.3.7/981
4003_8034	Channel (n) Status And Control (FTM0_C5SC)	32	R/W	0000_0000h	43.3.6/979
4003_8038	Channel (n) Value (FTM0_C5V)	32	R/W	0000_0000h	43.3.7/981
4003_803C	Channel (n) Status And Control (FTM0_C6SC)	32	R/W	0000_0000h	43.3.6/979
4003_8040	Channel (n) Value (FTM0_C6V)	32	R/W	0000_0000h	43.3.7/981
4003_8044	Channel (n) Status And Control (FTM0_C7SC)	32	R/W	0000_0000h	43.3.6/979
4003_8048	Channel (n) Value (FTM0_C7V)	32	R/W	0000_0000h	43.3.7/981
4003_804C	Counter Initial Value (FTM0_CNTIN)	32	R/W	0000_0000h	43.3.8/982

Table continues on the next page...

FTM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4003_8050	Capture And Compare Status (FTM0_STATUS)	32	R/W	0000_0000h	43.3.9/982
4003_8054	Features Mode Selection (FTM0_MODE)	32	R/W	0000_0004h	43.3.10/ 984
4003_8058	Synchronization (FTM0_SYNC)	32	R/W	0000_0000h	43.3.11/ 986
4003_805C	Initial State For Channels Output (FTM0_OUTINIT)	32	R/W	0000_0000h	43.3.12/ 989
4003_8060	Output Mask (FTM0_OUTMASK)	32	R/W	0000_0000h	43.3.13/ 990
4003_8064	Function For Linked Channels (FTM0_COMBINE)	32	R/W	0000_0000h	43.3.14/ 992
4003_8068	Deadtime Insertion Control (FTM0_DEADTIME)	32	R/W	0000_0000h	43.3.15/ 997
4003_806C	FTM External Trigger (FTM0_EXTTRIG)	32	R/W	0000_0000h	43.3.16/ 998
4003_8070	Channels Polarity (FTM0_POL)	32	R/W	0000_0000h	43.3.17/ 1000
4003_8074	Fault Mode Status (FTM0_FMS)	32	R/W	0000_0000h	43.3.18/ 1002
4003_8078	Input Capture Filter Control (FTM0_FILTER)	32	R/W	0000_0000h	43.3.19/ 1004
4003_807C	Fault Control (FTM0_FLTCTRL)	32	R/W	0000_0000h	43.3.20/ 1005
4003_8080	Quadrature Decoder Control And Status (FTM0_QDCTRL)	32	R/W	0000_0000h	43.3.21/ 1007
4003_8084	Configuration (FTM0_CONF)	32	R/W	0000_0000h	43.3.22/ 1009
4003_8088	FTM Fault Input Polarity (FTM0_FLTPOL)	32	R/W	0000_0000h	43.3.23/ 1010
4003_808C	Synchronization Configuration (FTM0_SYNCONF)	32	R/W	0000_0000h	43.3.24/ 1012
4003_8090	FTM Inverting Control (FTM0_INVCTRL)	32	R/W	0000_0000h	43.3.25/ 1014
4003_8094	FTM Software Output Control (FTM0_SWOCTRL)	32	R/W	0000_0000h	43.3.26/ 1015
4003_8098	FTM PWM Load (FTM0_PWMLOAD)	32	R/W	0000_0000h	43.3.27/ 1017
4003_9000	Status And Control (FTM1_SC)	32	R/W	0000_0000h	43.3.3/976
4003_9004	Counter (FTM1_CNT)	32	R/W	0000_0000h	43.3.4/977
4003_9008	Modulo (FTM1_MOD)	32	R/W	0000_0000h	43.3.5/978
4003_900C	Channel (n) Status And Control (FTM1_C0SC)	32	R/W	0000_0000h	43.3.6/979
4003_9010	Channel (n) Value (FTM1_C0V)	32	R/W	0000_0000h	43.3.7/981
4003_9014	Channel (n) Status And Control (FTM1_C1SC)	32	R/W	0000_0000h	43.3.6/979

Table continues on the next page...

FTM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4003_9018	Channel (n) Value (FTM1_C1V)	32	R/W	0000_0000h	43.3.7/981
4003_901C	Channel (n) Status And Control (FTM1_C2SC)	32	R/W	0000_0000h	43.3.6/979
4003_9020	Channel (n) Value (FTM1_C2V)	32	R/W	0000_0000h	43.3.7/981
4003_9024	Channel (n) Status And Control (FTM1_C3SC)	32	R/W	0000_0000h	43.3.6/979
4003_9028	Channel (n) Value (FTM1_C3V)	32	R/W	0000_0000h	43.3.7/981
4003_902C	Channel (n) Status And Control (FTM1_C4SC)	32	R/W	0000_0000h	43.3.6/979
4003_9030	Channel (n) Value (FTM1_C4V)	32	R/W	0000_0000h	43.3.7/981
4003_9034	Channel (n) Status And Control (FTM1_C5SC)	32	R/W	0000_0000h	43.3.6/979
4003_9038	Channel (n) Value (FTM1_C5V)	32	R/W	0000_0000h	43.3.7/981
4003_903C	Channel (n) Status And Control (FTM1_C6SC)	32	R/W	0000_0000h	43.3.6/979
4003_9040	Channel (n) Value (FTM1_C6V)	32	R/W	0000_0000h	43.3.7/981
4003_9044	Channel (n) Status And Control (FTM1_C7SC)	32	R/W	0000_0000h	43.3.6/979
4003_9048	Channel (n) Value (FTM1_C7V)	32	R/W	0000_0000h	43.3.7/981
4003_904C	Counter Initial Value (FTM1_CNTIN)	32	R/W	0000_0000h	43.3.8/982
4003_9050	Capture And Compare Status (FTM1_STATUS)	32	R/W	0000_0000h	43.3.9/982
4003_9054	Features Mode Selection (FTM1_MODE)	32	R/W	0000_0004h	43.3.10/ 984
4003_9058	Synchronization (FTM1_SYNC)	32	R/W	0000_0000h	43.3.11/ 986
4003_905C	Initial State For Channels Output (FTM1_OUTINIT)	32	R/W	0000_0000h	43.3.12/ 989
4003_9060	Output Mask (FTM1_OUTMASK)	32	R/W	0000_0000h	43.3.13/ 990
4003_9064	Function For Linked Channels (FTM1_COMBINE)	32	R/W	0000_0000h	43.3.14/ 992
4003_9068	Deadtime Insertion Control (FTM1_DEADTIME)	32	R/W	0000_0000h	43.3.15/ 997
4003_906C	FTM External Trigger (FTM1_EXTTRIG)	32	R/W	0000_0000h	43.3.16/ 998
4003_9070	Channels Polarity (FTM1_POL)	32	R/W	0000_0000h	43.3.17/ 1000
4003_9074	Fault Mode Status (FTM1_FMS)	32	R/W	0000_0000h	43.3.18/ 1002
4003_9078	Input Capture Filter Control (FTM1_FILTER)	32	R/W	0000_0000h	43.3.19/ 1004
4003_907C	Fault Control (FTM1_FLTCTRL)	32	R/W	0000_0000h	43.3.20/ 1005
4003_9080	Quadrature Decoder Control And Status (FTM1_QDCTRL)	32	R/W	0000_0000h	43.3.21/ 1007
4003_9084	Configuration (FTM1_CONF)	32	R/W	0000_0000h	43.3.22/ 1009

Table continues on the next page...

FTM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4003_9088	FTM Fault Input Polarity (FTM1_FLTPOL)	32	R/W	0000_0000h	43.3.23/ 1010
4003_908C	Synchronization Configuration (FTM1_SYNCONF)	32	R/W	0000_0000h	43.3.24/ 1012
4003_9090	FTM Inverting Control (FTM1_INVCTRL)	32	R/W	0000_0000h	43.3.25/ 1014
4003_9094	FTM Software Output Control (FTM1_SWOCTRL)	32	R/W	0000_0000h	43.3.26/ 1015
4003_9098	FTM PWM Load (FTM1_PWMLOAD)	32	R/W	0000_0000h	43.3.27/ 1017
4003_A000	Status And Control (FTM2_SC)	32	R/W	0000_0000h	43.3.3/976
4003_A004	Counter (FTM2_CNT)	32	R/W	0000_0000h	43.3.4/977
4003_A008	Modulo (FTM2_MOD)	32	R/W	0000_0000h	43.3.5/978
4003_A00C	Channel (n) Status And Control (FTM2_C0SC)	32	R/W	0000_0000h	43.3.6/979
4003_A010	Channel (n) Value (FTM2_C0V)	32	R/W	0000_0000h	43.3.7/981
4003_A014	Channel (n) Status And Control (FTM2_C1SC)	32	R/W	0000_0000h	43.3.6/979
4003_A018	Channel (n) Value (FTM2_C1V)	32	R/W	0000_0000h	43.3.7/981
4003_A01C	Channel (n) Status And Control (FTM2_C2SC)	32	R/W	0000_0000h	43.3.6/979
4003_A020	Channel (n) Value (FTM2_C2V)	32	R/W	0000_0000h	43.3.7/981
4003_A024	Channel (n) Status And Control (FTM2_C3SC)	32	R/W	0000_0000h	43.3.6/979
4003_A028	Channel (n) Value (FTM2_C3V)	32	R/W	0000_0000h	43.3.7/981
4003_A02C	Channel (n) Status And Control (FTM2_C4SC)	32	R/W	0000_0000h	43.3.6/979
4003_A030	Channel (n) Value (FTM2_C4V)	32	R/W	0000_0000h	43.3.7/981
4003_A034	Channel (n) Status And Control (FTM2_C5SC)	32	R/W	0000_0000h	43.3.6/979
4003_A038	Channel (n) Value (FTM2_C5V)	32	R/W	0000_0000h	43.3.7/981
4003_A03C	Channel (n) Status And Control (FTM2_C6SC)	32	R/W	0000_0000h	43.3.6/979
4003_A040	Channel (n) Value (FTM2_C6V)	32	R/W	0000_0000h	43.3.7/981
4003_A044	Channel (n) Status And Control (FTM2_C7SC)	32	R/W	0000_0000h	43.3.6/979
4003_A048	Channel (n) Value (FTM2_C7V)	32	R/W	0000_0000h	43.3.7/981
4003_A04C	Counter Initial Value (FTM2_CNTIN)	32	R/W	0000_0000h	43.3.8/982
4003_A050	Capture And Compare Status (FTM2_STATUS)	32	R/W	0000_0000h	43.3.9/982
4003_A054	Features Mode Selection (FTM2_MODE)	32	R/W	0000_0004h	43.3.10/ 984
4003_A058	Synchronization (FTM2_SYNC)	32	R/W	0000_0000h	43.3.11/ 986
4003_A05C	Initial State For Channels Output (FTM2_OUTINIT)	32	R/W	0000_0000h	43.3.12/ 989
4003_A060	Output Mask (FTM2_OUTMASK)	32	R/W	0000_0000h	43.3.13/ 990
4003_A064	Function For Linked Channels (FTM2_COMBINE)	32	R/W	0000_0000h	43.3.14/ 992

Table continues on the next page...

FTM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4003_A068	Deadtime Insertion Control (FTM2_DEADTIME)	32	R/W	0000_0000h	43.3.15/ 997
4003_A06C	FTM External Trigger (FTM2_EXTTRIG)	32	R/W	0000_0000h	43.3.16/ 998
4003_A070	Channels Polarity (FTM2_POL)	32	R/W	0000_0000h	43.3.17/ 1000
4003_A074	Fault Mode Status (FTM2_FMS)	32	R/W	0000_0000h	43.3.18/ 1002
4003_A078	Input Capture Filter Control (FTM2_FILTER)	32	R/W	0000_0000h	43.3.19/ 1004
4003_A07C	Fault Control (FTM2_FLTCTRL)	32	R/W	0000_0000h	43.3.20/ 1005
4003_A080	Quadrature Decoder Control And Status (FTM2_QDCTRL)	32	R/W	0000_0000h	43.3.21/ 1007
4003_A084	Configuration (FTM2_CONF)	32	R/W	0000_0000h	43.3.22/ 1009
4003_A088	FTM Fault Input Polarity (FTM2_FLTPOL)	32	R/W	0000_0000h	43.3.23/ 1010
4003_A08C	Synchronization Configuration (FTM2_SYNCONF)	32	R/W	0000_0000h	43.3.24/ 1012
4003_A090	FTM Inverting Control (FTM2_INVCTRL)	32	R/W	0000_0000h	43.3.25/ 1014
4003_A094	FTM Software Output Control (FTM2_SWOCTRL)	32	R/W	0000_0000h	43.3.26/ 1015
4003_A098	FTM PWM Load (FTM2_PWMLOAD)	32	R/W	0000_0000h	43.3.27/ 1017

43.3.3 Status And Control (FTMx_SC)

SC contains the overflow status flag and control bits used to configure the interrupt enable, FTM configuration, clock source, and prescaler factor. These controls relate to all channels within this module.

Address: Base address + 0h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								TOF	TOIE	CPWMS	CLKS		PS		
W									0							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

FTMx_SC field descriptions

Field	Description
31–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7 TOF	Timer Overflow Flag Set by hardware when the FTM counter passes the value in the MOD register. The TOF bit is cleared by reading the SC register while TOF is set and then writing a 0 to TOF bit. Writing a 1 to TOF has no effect. If another FTM overflow occurs between the read and write operations, the write operation has no effect; therefore, TOF remains set indicating an overflow has occurred. In this case, a TOF interrupt request is not lost due to the clearing sequence for a previous TOF. 0 FTM counter has not overflowed. 1 FTM counter has overflowed.

Table continues on the next page...

FTMx_SC field descriptions (continued)

Field	Description
6 TOIE	Timer Overflow Interrupt Enable Enables FTM overflow interrupts. 0 Disable TOF interrupts. Use software polling. 1 Enable TOF interrupts. An interrupt is generated when TOF equals one.
5 CPWMS	Center-Aligned PWM Select Selects CPWM mode. This mode configures the FTM to operate in Up-Down Counting mode. This field is write protected. It can be written only when MODE[WPDIS] = 1. 0 FTM counter operates in Up Counting mode. 1 FTM counter operates in Up-Down Counting mode.
4–3 CLKS	Clock Source Selection Selects FTM counter clock sources. This field is write protected. It can be written only when MODE[WPDIS] = 1. 00 No clock selected. This in effect disables the FTM counter. 01 System clock 11 External clock
PS	Prescale Factor Selection Selects one of 8 division factors for the clock source selected by CLKS. The new prescaler factor affects the clock source on the next system clock cycle after the new value is updated into the register bits. This field is write protected. It can be written only when MODE[WPDIS] = 1. 000 Divide by 1 001 Divide by 2 010 Divide by 4 011 Divide by 8 100 Divide by 16 101 Divide by 32 110 Divide by 64 111 Divide by 128

43.3.4 Counter (FTMx_CNT)

The CNT register contains the FTM counter value.

Reset clears the CNT register. Writing any value to COUNT updates the counter with its initial value, CNTIN.

When BDM is active, the FTM counter is frozen. This is the value that you may read.

Address: Base address + 4h offset

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

FTMx_CNT field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
COUNT	Counter Value

43.3.5 Modulo (FTMx_MOD)

The Modulo register contains the modulo value for the FTM counter. After the FTM counter reaches the modulo value, the overflow flag (TOF) becomes set at the next clock, and the next value of FTM counter depends on the selected counting method; see Counter.

Writing to the MOD register latches the value into a buffer. The MOD register is updated with the value of its write buffer according to Registers updated from write buffers.

If FTMEN = 0, this write coherency mechanism may be manually reset by writing to the SC register whether BDM is active or not.

Initialize the FTM counter, by writing to CNT, before writing to the MOD register to avoid confusion about when the first counter overflow will occur.

Address: Base address + 8h offset

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

FTMx_MOD field descriptions

Field	Description
31–16 Reserved	This field is reserved.
MOD	Modulo Value

43.3.6 Channel (n) Status And Control (FTMx_C*n*SC)

CnSC contains the channel-interrupt-status flag and control bits used to configure the interrupt enable, channel configuration, and pin function.

Table 43-2. Mode, edge, and level selection

DECAPEN	COMBINE	CPWMS	MSnB:MSnA	ELSnB:ELSnA	Mode	Configuration
X	X	X	XX	00	Pin not used for FTM—revert the channel pin to general purpose I/O or other peripheral control	
0	0	0	00	01	Input Capture	Capture on Rising Edge Only
				10		Capture on Falling Edge Only
				11		Capture on Rising or Falling Edge
			01	01	Output Compare	Toggle Output on match
				10		Clear Output on match
				11		Set Output on match
			1X	10	Edge-Aligned PWM	High-true pulses (clear Output on match)
				X1		Low-true pulses (set Output on match)
		1	XX	10	Center-Aligned PWM	High-true pulses (clear Output on match-up)
				X1		Low-true pulses (set Output on match-up)
	1	0	XX	10	Combine PWM	High-true pulses (set on channel (n) match, and clear on channel (n+1) match)
				X1		Low-true pulses (clear on channel (n) match, and set on channel (n +1) match)

Table continues on the next page...

Table 43-2. Mode, edge, and level selection (continued)

DECAPEN	COMBINE	CPWMS	MSnB:MSnA	ELSnB:ELSnA	Mode	Configuration
1	0	0	X0	See the following table (Table 43-3).	Dual Edge Capture	One-Shot Capture mode
			X1			Continuous Capture mode

Table 43-3. Dual Edge Capture mode — edge polarity selection

ELSnB	ELSnA	Channel Port Enable	Detected Edges
0	0	Disabled	No edge
0	1	Enabled	Rising edge
1	0	Enabled	Falling edge
1	1	Enabled	Rising and falling edges

Address: Base address + Ch offset + (8d × i), where i=0d to 7d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								CHF	CHIE	MSB	MSA	ELSB	ELSA	0	DMA
W									0							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

FTMx_CnSC field descriptions

Field	Description
31–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7 CHF	Channel Flag Set by hardware when an event occurs on the channel. CHF is cleared by reading the CSC register while CHnF is set and then writing a 0 to the CHF bit. Writing a 1 to CHF has no effect. If another event occurs between the read and write operations, the write operation has no effect; therefore, CHF remains set indicating an event has occurred. In this case a CHF interrupt request is not lost due to the clearing sequence for a previous CHF. 0 No channel event has occurred. 1 A channel event has occurred.
6 CHIE	Channel Interrupt Enable Enables channel interrupts. 0 Disable channel interrupts. Use software polling. 1 Enable channel interrupts.

Table continues on the next page...

FTMx_*Cn*SC field descriptions (continued)

Field	Description
5 MSB	Channel Mode Select Used for further selections in the channel logic. Its functionality is dependent on the channel mode. See Table 43-2 . This field is write protected. It can be written only when MODE[WPDIS] = 1.
4 MSA	Channel Mode Select Used for further selections in the channel logic. Its functionality is dependent on the channel mode. See Table 43-2 . This field is write protected. It can be written only when MODE[WPDIS] = 1.
3 ELSB	Edge or Level Select The functionality of ELSB and ELSA depends on the channel mode. See Table 43-2 . This field is write protected. It can be written only when MODE[WPDIS] = 1.
2 ELSA	Edge or Level Select The functionality of ELSB and ELSA depends on the channel mode. See Table 43-2 . This field is write protected. It can be written only when MODE[WPDIS] = 1.
1 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
0 DMA	DMA Enable Enables DMA transfers for the channel. 0 Disable DMA transfers. 1 Enable DMA transfers.

43.3.7 Channel (n) Value (FTMx_*Cn*V)

These registers contain the captured FTM counter value for the input modes or the match value for the output modes.

In Input Capture, Capture Test, and Dual Edge Capture modes, any write to a CnV register is ignored.

In output modes, writing to a CnV register latches the value into a buffer. A CnV register is updated with the value of its write buffer according to Registers updated from write buffers.

If FTMEN = 0, this write coherency mechanism may be manually reset by writing to the CnSC register whether BDM mode is active or not.

Address: Base address + 10h offset + (8d × i), where i=0d to 7d

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

FTMx_C*n*V field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
VAL	Channel Value Captured FTM counter value of the input modes or the match value for the output modes

43.3.8 Counter Initial Value (FTMx_CNTIN)

The Counter Initial Value register contains the initial value for the FTM counter.

Writing to the CNTIN register latches the value into a buffer. The CNTIN register is updated with the value of its write buffer according to Registers updated from write buffers.

When the FTM clock is initially selected, by writing a non-zero value to the CLKS bits, the FTM counter starts with the value 0x0000. To avoid this behavior, before the first write to select the FTM clock, write the new value to the the CNTIN register and then initialize the FTM counter by writing any value to the CNT register.

Address: Base address + 4Ch offset

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

FTMx_CNTIN field descriptions

Field	Description
31–16 Reserved	This field is reserved.
INIT	Initial Value Of The FTM Counter

43.3.9 Capture And Compare Status (FTMx_STATUS)

The STATUS register contains a copy of the status flag CHnF bit in CnSC for each FTM channel for software convenience.

Each CHnF bit in STATUS is a mirror of CHnF bit in CnSC. All CHnF bits can be checked using only one read of STATUS. All CHnF bits can be cleared by reading STATUS followed by writing 0x00 to STATUS.

Hardware sets the individual channel flags when an event occurs on the channel. CHnF is cleared by reading STATUS while CHnF is set and then writing a 0 to the CHnF bit. Writing a 1 to CHnF has no effect.

If another event occurs between the read and write operations, the write operation has no effect; therefore, CHnF remains set indicating an event has occurred. In this case, a CHnF interrupt request is not lost due to the clearing sequence for a previous CHnF.

Address: Base address + 50h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								CH7F	CH6F	CH5F	CH4F	CH3F	CH2F	CH1F	CH0F
W									0	0	0	0	0	0	0	0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

FTMx_STATUS field descriptions

Field	Description
31–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7 CH7F	Channel 7 Flag See the register description. 0 No channel event has occurred. 1 A channel event has occurred.
6 CH6F	Channel 6 Flag

Table continues on the next page...

FTMx_STATUS field descriptions (continued)

Field	Description
	See the register description. 0 No channel event has occurred. 1 A channel event has occurred.
5 CH5F	Channel 5 Flag See the register description. 0 No channel event has occurred. 1 A channel event has occurred.
4 CH4F	Channel 4 Flag See the register description. 0 No channel event has occurred. 1 A channel event has occurred.
3 CH3F	Channel 3 Flag See the register description. 0 No channel event has occurred. 1 A channel event has occurred.
2 CH2F	Channel 2 Flag See the register description. 0 No channel event has occurred. 1 A channel event has occurred.
1 CH1F	Channel 1 Flag See the register description. 0 No channel event has occurred. 1 A channel event has occurred.
0 CH0F	Channel 0 Flag See the register description. 0 No channel event has occurred. 1 A channel event has occurred.

43.3.10 Features Mode Selection (FTMx_MODE)

This register contains the global enable bit for FTM-specific features and the control bits used to configure:

- Fault control mode and interrupt
- Capture Test mode
- PWM synchronization

- Write protection
- Channel output initialization

These controls relate to all channels within this module.

Address: Base address + 54h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								FAULTIE	FAULTM		CAPTEST	PWMSYNC	WPDIS	INIT	FTMEN
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0

FTMx_MODE field descriptions

Field	Description
31–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7 FAULTIE	Fault Interrupt Enable Enables the generation of an interrupt when a fault is detected by FTM and the FTM fault control is enabled. 0 Fault control interrupt is disabled. 1 Fault control interrupt is enabled.
6–5 FAULTM	Fault Control Mode Defines the FTM fault control mode. This field is write protected. It can be written only when MODE[WPDIS] = 1. 00 Fault control is disabled for all channels. 01 Fault control is enabled for even channels only (channels 0, 2, 4, and 6), and the selected mode is the manual fault clearing. 10 Fault control is enabled for all channels, and the selected mode is the manual fault clearing. 11 Fault control is enabled for all channels, and the selected mode is the automatic fault clearing.
4 CAPTEST	Capture Test Mode Enable Enables the capture test mode. This field is write protected. It can be written only when MODE[WPDIS] = 1. 0 Capture test mode is disabled. 1 Capture test mode is enabled.
3 PWMSYNC	PWM Synchronization Mode

Table continues on the next page...

FTMx_MODE field descriptions (continued)

Field	Description
	Selects which triggers can be used by MOD, CnV, OUTMASK, and FTM counter synchronization. See PWM synchronization. The PWMSYNC bit configures the synchronization when SYNCMODE is 0. 0 No restrictions. Software and hardware triggers can be used by MOD, CnV, OUTMASK, and FTM counter synchronization. 1 Software trigger can only be used by MOD and CnV synchronization, and hardware triggers can only be used by OUTMASK and FTM counter synchronization.
2 WPDIS	Write Protection Disable When write protection is enabled (WPDIS = 0), write protected bits cannot be written. When write protection is disabled (WPDIS = 1), write protected bits can be written. The WPDIS bit is the negation of the WPEN bit. WPDIS is cleared when 1 is written to WPEN. WPDIS is set when WPEN bit is read as a 1 and then 1 is written to WPDIS. Writing 0 to WPDIS has no effect. 0 Write protection is enabled. 1 Write protection is disabled.
1 INIT	Initialize The Channels Output When a 1 is written to INIT bit the channels output is initialized according to the state of their corresponding bit in the OUTINIT register. Writing a 0 to INIT bit has no effect. The INIT bit is always read as 0.
0 FTMEN	FTM Enable This field is write protected. It can be written only when MODE[WPDIS] = 1. 0 TPM compatibility. Free running counter and synchronization compatible with TPM. 1 Free running counter and synchronization are different from TPM behavior.

43.3.11 Synchronization (FTMx_SYNC)

This register configures the PWM synchronization.

A synchronization event can perform the synchronized update of MOD, CV, and OUTMASK registers with the value of their write buffer and the FTM counter initialization.

NOTE

The software trigger, SWSYNC bit, and hardware triggers TRIG0, TRIG1, and TRIG2 bits have a potential conflict if used together when SYNCMODE = 0. Use only hardware or software triggers but not both at the same time, otherwise unpredictable behavior is likely to happen.

The selection of the loading point, CNTMAX and CNTMIN bits, is intended to provide the update of MOD, CNTIN, and CnV registers across all enabled channels simultaneously. The use of the loading point selection together with SYNCMODE = 0 and hardware trigger selection, TRIG0, TRIG1, or TRIG2 bits, is likely to result in unpredictable behavior.

The synchronization event selection also depends on the PWMSYNC (MODE register) and SYNCMODE (SYNCONF register) bits. See PWM synchronization.

Address: Base address + 58h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								SWSYNC	TRIG2	TRIG1	TRIG0	SYNCHOM	REINIT	CNTMAX	CNTMIN
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

FTMx_SYNC field descriptions

Field	Description
31–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7 SWSYNC	PWM Synchronization Software Trigger Selects the software trigger as the PWM synchronization trigger. The software trigger happens when a 1 is written to SWSYNC bit. 0 Software trigger is not selected. 1 Software trigger is selected.
6 TRIG2	PWM Synchronization Hardware Trigger 2 Enables hardware trigger 2 to the PWM synchronization. Hardware trigger 2 happens when a rising edge is detected at the trigger 2 input signal. 0 Trigger is disabled. 1 Trigger is enabled.
5 TRIG1	PWM Synchronization Hardware Trigger 1 Enables hardware trigger 1 to the PWM synchronization. Hardware trigger 1 happens when a rising edge is detected at the trigger 1 input signal.

Table continues on the next page...

FTMx_SYNC field descriptions (continued)

Field	Description
	0 Trigger is disabled. 1 Trigger is enabled.
4 TRIG0	PWM Synchronization Hardware Trigger 0 Enables hardware trigger 0 to the PWM synchronization. Hardware trigger 0 occurs when a rising edge is detected at the trigger 0 input signal. 0 Trigger is disabled. 1 Trigger is enabled.
3 SYNCHOM	Output Mask Synchronization Selects when the OUTMASK register is updated with the value of its buffer. 0 OUTMASK register is updated with the value of its buffer in all rising edges of the system clock. 1 OUTMASK register is updated with the value of its buffer only by the PWM synchronization.
2 REINIT	FTM Counter Reinitialization By Synchronization (FTM counter synchronization) Determines if the FTM counter is reinitialized when the selected trigger for the synchronization is detected. The REINIT bit configures the synchronization when SYNCMODE is zero. 0 FTM counter continues to count normally. 1 FTM counter is updated with its initial value when the selected trigger is detected.
1 CNTMAX	Maximum Loading Point Enable Selects the maximum loading point to PWM synchronization. See Boundary cycle and loading points. If CNTMAX is 1, the selected loading point is when the FTM counter reaches its maximum value (MOD register). 0 The maximum loading point is disabled. 1 The maximum loading point is enabled.
0 CNTMIN	Minimum Loading Point Enable Selects the minimum loading point to PWM synchronization. See Boundary cycle and loading points. If CNTMIN is one, the selected loading point is when the FTM counter reaches its minimum value (CNTIN register). 0 The minimum loading point is disabled. 1 The minimum loading point is enabled.

43.3.12 Initial State For Channels Output (FTMx_OUTINIT)

Address: Base address + 5Ch offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								CH7OI	CH6OI	CH5OI	CH4OI	CH3OI	CH2OI	CH1OI	CH0OI
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

FTMx_OUTINIT field descriptions

Field	Description
31–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7 CH7OI	Channel 7 Output Initialization Value Selects the value that is forced into the channel output when the initialization occurs. 0 The initialization value is 0. 1 The initialization value is 1.
6 CH6OI	Channel 6 Output Initialization Value Selects the value that is forced into the channel output when the initialization occurs. 0 The initialization value is 0. 1 The initialization value is 1.
5 CH5OI	Channel 5 Output Initialization Value Selects the value that is forced into the channel output when the initialization occurs. 0 The initialization value is 0. 1 The initialization value is 1.
4 CH4OI	Channel 4 Output Initialization Value Selects the value that is forced into the channel output when the initialization occurs. 0 The initialization value is 0. 1 The initialization value is 1.
3 CH3OI	Channel 3 Output Initialization Value Selects the value that is forced into the channel output when the initialization occurs.

Table continues on the next page...

FTMx_OUTINIT field descriptions (continued)

Field	Description
	0 The initialization value is 0. 1 The initialization value is 1.
2 CH2OI	Channel 2 Output Initialization Value Selects the value that is forced into the channel output when the initialization occurs. 0 The initialization value is 0. 1 The initialization value is 1.
1 CH1OI	Channel 1 Output Initialization Value Selects the value that is forced into the channel output when the initialization occurs. 0 The initialization value is 0. 1 The initialization value is 1.
0 CH0OI	Channel 0 Output Initialization Value Selects the value that is forced into the channel output when the initialization occurs. 0 The initialization value is 0. 1 The initialization value is 1.

43.3.13 Output Mask (FTMx_OUTMASK)

This register provides a mask for each FTM channel. The mask of a channel determines if its output responds, that is, it is masked or not, when a match occurs. This feature is used for BLDC control where the PWM signal is presented to an electric motor at specific times to provide electronic commutation.

Any write to the OUTMASK register, stores the value in its write buffer. The register is updated with the value of its write buffer according to PWM synchronization.

Address: Base address + 60h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								CH7OM	CH6OM	CH5OM	CH4OM	CH3OM	CH2OM	CH1OM	CH0OM
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

FTMx_OUTMASK field descriptions

Field	Description
31–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7 CH7OM	Channel 7 Output Mask Defines if the channel output is masked or unmasked. 0 Channel output is not masked. It continues to operate normally. 1 Channel output is masked. It is forced to its inactive state.
6 CH6OM	Channel 6 Output Mask Defines if the channel output is masked or unmasked. 0 Channel output is not masked. It continues to operate normally. 1 Channel output is masked. It is forced to its inactive state.
5 CH5OM	Channel 5 Output Mask Defines if the channel output is masked or unmasked. 0 Channel output is not masked. It continues to operate normally. 1 Channel output is masked. It is forced to its inactive state.
4 CH4OM	Channel 4 Output Mask Defines if the channel output is masked or unmasked. 0 Channel output is not masked. It continues to operate normally. 1 Channel output is masked. It is forced to its inactive state.
3 CH3OM	Channel 3 Output Mask Defines if the channel output is masked or unmasked. 0 Channel output is not masked. It continues to operate normally. 1 Channel output is masked. It is forced to its inactive state.
2 CH2OM	Channel 2 Output Mask Defines if the channel output is masked or unmasked. 0 Channel output is not masked. It continues to operate normally. 1 Channel output is masked. It is forced to its inactive state.
1 CH1OM	Channel 1 Output Mask Defines if the channel output is masked or unmasked. 0 Channel output is not masked. It continues to operate normally. 1 Channel output is masked. It is forced to its inactive state.
0 CH0OM	Channel 0 Output Mask Defines if the channel output is masked or unmasked. 0 Channel output is not masked. It continues to operate normally. 1 Channel output is masked. It is forced to its inactive state.

43.3.14 Function For Linked Channels (FTMx_COMBINE)

This register contains the control bits used to configure the fault control, synchronization, deadtime insertion, Dual Edge Capture mode, Complementary, and Combine mode for each pair of channels (n) and (n+1), where n equals 0, 2, 4, and 6.

Address: Base address + 64h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	FAULTEN3	SYNCEN3	DTEN3	DECAP3	DECAPEN3	COMP3	COMBINE3	0	FAULTEN2	SYNCEN2	DTEN2	DECAP2	DECAPEN2	COMP2	COMBINE2
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	FAULTEN1	SYNCEN1	DTEN1	DECAP1	DECAPEN1	COMP1	COMBINE1	0	FAULTEN0	SYNCEN0	DTEN0	DECAP0	DECAPEN0	COMP0	COMBINE0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

FTMx_COMBINE field descriptions

Field	Description
31 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
30 FAULTEN3	Fault Control Enable For n = 6 Enables the fault control in channels (n) and (n+1). This field is write protected. It can be written only when MODE[WPDIS] = 1. 0 The fault control in this pair of channels is disabled. 1 The fault control in this pair of channels is enabled.
29 SYNCEN3	Synchronization Enable For n = 6 Enables PWM synchronization of registers C(n)V and C(n+1)V. 0 The PWM synchronization in this pair of channels is disabled. 1 The PWM synchronization in this pair of channels is enabled.
28 DTEN3	Deadtime Enable For n = 6 Enables the deadtime insertion in the channels (n) and (n+1). This field is write protected. It can be written only when MODE[WPDIS] = 1.

Table continues on the next page...

FTMx_COMBINE field descriptions (continued)

Field	Description
	0 The deadtime insertion in this pair of channels is disabled. 1 The deadtime insertion in this pair of channels is enabled.
27 DECAP3	Dual Edge Capture Mode Captures For n = 6 Enables the capture of the FTM counter value according to the channel (n) input event and the configuration of the dual edge capture bits. This field applies only when DECAPEN = 1. DECAP bit is cleared automatically by hardware if dual edge capture – one-shot mode is selected and when the capture of channel (n+1) event is made. 0 The dual edge captures are inactive. 1 The dual edge captures are active.
26 DECAPEN3	Dual Edge Capture Mode Enable For n = 6 Enables the Dual Edge Capture mode in the channels (n) and (n+1). This bit reconfigures the function of MSnA, ELSnB:ELSnA and ELS(n+1)B:ELS(n+1)A bits in Dual Edge Capture mode according to Table 43-2. This field is write protected. It can be written only when MODE[WPDIS] = 1. 0 The Dual Edge Capture mode in this pair of channels is disabled. 1 The Dual Edge Capture mode in this pair of channels is enabled.
25 COMP3	Complement Of Channel (n) for n = 6 Enables Complementary mode for the combined channels. In Complementary mode the channel (n+1) output is the inverse of the channel (n) output. This field is write protected. It can be written only when MODE[WPDIS] = 1. 0 The channel (n+1) output is the same as the channel (n) output. 1 The channel (n+1) output is the complement of the channel (n) output.
24 COMBINE3	Combine Channels For n = 6 Enables the combine feature for channels (n) and (n+1). This field is write protected. It can be written only when MODE[WPDIS] = 1. 0 Channels (n) and (n+1) are independent. 1 Channels (n) and (n+1) are combined.
23 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
22 FAULTEN2	Fault Control Enable For n = 4 Enables the fault control in channels (n) and (n+1). This field is write protected. It can be written only when MODE[WPDIS] = 1. 0 The fault control in this pair of channels is disabled. 1 The fault control in this pair of channels is enabled.
21 SYNCEN2	Synchronization Enable For n = 4 Enables PWM synchronization of registers C(n)V and C(n+1)V.

Table continues on the next page...

FTMx_COMBINE field descriptions (continued)

Field	Description
	0 The PWM synchronization in this pair of channels is disabled. 1 The PWM synchronization in this pair of channels is enabled.
20 DTEN2	Deadtime Enable For n = 4 Enables the deadtime insertion in the channels (n) and (n+1). This field is write protected. It can be written only when MODE[WPDIS] = 1. 0 The deadtime insertion in this pair of channels is disabled. 1 The deadtime insertion in this pair of channels is enabled.
19 DECAP2	Dual Edge Capture Mode Captures For n = 4 Enables the capture of the FTM counter value according to the channel (n) input event and the configuration of the dual edge capture bits. This field applies only when DECAPEN = 1. DECAP bit is cleared automatically by hardware if dual edge capture – one-shot mode is selected and when the capture of channel (n+1) event is made. 0 The dual edge captures are inactive. 1 The dual edge captures are active.
18 DECAPEN2	Dual Edge Capture Mode Enable For n = 4 Enables the Dual Edge Capture mode in the channels (n) and (n+1). This bit reconfigures the function of MSnA, ELSnB:ELSnA and ELS(n+1)B:ELS(n+1)A bits in Dual Edge Capture mode according to Table 43-2. This field is write protected. It can be written only when MODE[WPDIS] = 1. 0 The Dual Edge Capture mode in this pair of channels is disabled. 1 The Dual Edge Capture mode in this pair of channels is enabled.
17 COMP2	Complement Of Channel (n) For n = 4 Enables Complementary mode for the combined channels. In Complementary mode the channel (n+1) output is the inverse of the channel (n) output. This field is write protected. It can be written only when MODE[WPDIS] = 1. 0 The channel (n+1) output is the same as the channel (n) output. 1 The channel (n+1) output is the complement of the channel (n) output.
16 COMBINE2	Combine Channels For n = 4 Enables the combine feature for channels (n) and (n+1). This field is write protected. It can be written only when MODE[WPDIS] = 1. 0 Channels (n) and (n+1) are independent. 1 Channels (n) and (n+1) are combined.
15 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
14 FAULTEN1	Fault Control Enable For n = 2 Enables the fault control in channels (n) and (n+1). This field is write protected. It can be written only when MODE[WPDIS] = 1.

Table continues on the next page...

FTMx_COMBINE field descriptions (continued)

Field	Description
	0 The fault control in this pair of channels is disabled. 1 The fault control in this pair of channels is enabled.
13 SYNCEN1	Synchronization Enable For n = 2 Enables PWM synchronization of registers C(n)V and C(n+1)V. 0 The PWM synchronization in this pair of channels is disabled. 1 The PWM synchronization in this pair of channels is enabled.
12 DTEN1	Deadtime Enable For n = 2 Enables the deadtime insertion in the channels (n) and (n+1). This field is write protected. It can be written only when MODE[WPDIS] = 1. 0 The deadtime insertion in this pair of channels is disabled. 1 The deadtime insertion in this pair of channels is enabled.
11 DECAP1	Dual Edge Capture Mode Captures For n = 2 Enables the capture of the FTM counter value according to the channel (n) input event and the configuration of the dual edge capture bits. This field applies only when DECAPEN = 1. DECAP bit is cleared automatically by hardware if Dual Edge Capture – One-Shot mode is selected and when the capture of channel (n+1) event is made. 0 The dual edge captures are inactive. 1 The dual edge captures are active.
10 DECAPEN1	Dual Edge Capture Mode Enable For n = 2 Enables the Dual Edge Capture mode in the channels (n) and (n+1). This bit reconfigures the function of MSnA, ELSnB:ELSnA and ELS(n+1)B:ELS(n+1)A bits in Dual Edge Capture mode according to Table 43-2. This field is write protected. It can be written only when MODE[WPDIS] = 1. 0 The Dual Edge Capture mode in this pair of channels is disabled. 1 The Dual Edge Capture mode in this pair of channels is enabled.
9 COMP1	Complement Of Channel (n) For n = 2 Enables Complementary mode for the combined channels. In Complementary mode the channel (n+1) output is the inverse of the channel (n) output. This field is write protected. It can be written only when MODE[WPDIS] = 1. 0 The channel (n+1) output is the same as the channel (n) output. 1 The channel (n+1) output is the complement of the channel (n) output.
8 COMBINE1	Combine Channels For n = 2 Enables the combine feature for channels (n) and (n+1). This field is write protected. It can be written only when MODE[WPDIS] = 1. 0 Channels (n) and (n+1) are independent. 1 Channels (n) and (n+1) are combined.

Table continues on the next page...

FTMx_COMBINE field descriptions (continued)

Field	Description
7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
6 FAULTEN0	Fault Control Enable For n = 0 Enables the fault control in channels (n) and (n+1). This field is write protected. It can be written only when MODE[WPDIS] = 1. 0 The fault control in this pair of channels is disabled. 1 The fault control in this pair of channels is enabled.
5 SYNCEN0	Synchronization Enable For n = 0 Enables PWM synchronization of registers C(n)V and C(n+1)V. 0 The PWM synchronization in this pair of channels is disabled. 1 The PWM synchronization in this pair of channels is enabled.
4 DTEN0	Deadtime Enable For n = 0 Enables the deadtime insertion in the channels (n) and (n+1). This field is write protected. It can be written only when MODE[WPDIS] = 1. 0 The deadtime insertion in this pair of channels is disabled. 1 The deadtime insertion in this pair of channels is enabled.
3 DECAP0	Dual Edge Capture Mode Captures For n = 0 Enables the capture of the FTM counter value according to the channel (n) input event and the configuration of the dual edge capture bits. This field applies only when DECAPEN = 1. DECAP bit is cleared automatically by hardware if dual edge capture – one-shot mode is selected and when the capture of channel (n+1) event is made. 0 The dual edge captures are inactive. 1 The dual edge captures are active.
2 DECAPEN0	Dual Edge Capture Mode Enable For n = 0 Enables the Dual Edge Capture mode in the channels (n) and (n+1). This bit reconfigures the function of MSnA, ELSnB:ELSnA and ELS(n+1)B:ELS(n+1)A bits in Dual Edge Capture mode according to Table 43-2. This field is write protected. It can be written only when MODE[WPDIS] = 1. 0 The Dual Edge Capture mode in this pair of channels is disabled. 1 The Dual Edge Capture mode in this pair of channels is enabled.
1 COMP0	Complement Of Channel (n) For n = 0 Enables Complementary mode for the combined channels. In Complementary mode the channel (n+1) output is the inverse of the channel (n) output. This field is write protected. It can be written only when MODE[WPDIS] = 1. 0 The channel (n+1) output is the same as the channel (n) output. 1 The channel (n+1) output is the complement of the channel (n) output.

Table continues on the next page...

FTMx_COMBINE field descriptions (continued)

Field	Description
0 COMBINE0	Combine Channels For n = 0 Enables the combine feature for channels (n) and (n+1). This field is write protected. It can be written only when MODE[WPDIS] = 1. 0 Channels (n) and (n+1) are independent. 1 Channels (n) and (n+1) are combined.

43.3.15 Deadtime Insertion Control (FTMx_DEADTIME)

This register selects the deadtime prescaler factor and deadtime value. All FTM channels use this clock prescaler and this deadtime value for the deadtime insertion.

Address: Base address + 68h offset

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

FTMx_DEADTIME field descriptions

Field	Description
31–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7–6 DTPS	Deadtime Prescaler Value Selects the division factor of the system clock. This prescaled clock is used by the deadtime counter. This field is write protected. It can be written only when MODE[WPDIS] = 1. 0x Divide the system clock by 1. 10 Divide the system clock by 4. 11 Divide the system clock by 16.
DTVAL	Deadtime Value Selects the deadtime insertion value for the deadtime counter. The deadtime counter is clocked by a scaled version of the system clock. See the description of DTPS. Deadtime insert value = (DTPS × DTVAL). DTVAL selects the number of deadtime counts inserted as follows: When DTVAL is 0, no counts are inserted. When DTVAL is 1, 1 count is inserted. When DTVAL is 2, 2 counts are inserted. This pattern continues up to a possible 63 counts. This field is write protected. It can be written only when MODE[WPDIS] = 1.

43.3.16 FTM External Trigger (FTMx_EXTTRIG)

This register:

- Indicates when a channel trigger was generated
- Enables the generation of a trigger when the FTM counter is equal to its initial value
- Selects which channels are used in the generation of the channel triggers

Several channels can be selected to generate multiple triggers in one PWM period. See Channel trigger output and Initialization trigger.

Channels 6 and 7 are not used to generate channel triggers.

Address: Base address + 6Ch offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved								TRIGF	INITTRIGEN	CH1TRIG	CH0TRIG	CH5TRIG	CH4TRIG	CH3TRIG	CH2TRIG
W									0							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

FTMx_EXTTRIG field descriptions

Field	Description
31–8 Reserved	This field is reserved.
7 TRIGF	Channel Trigger Flag Set by hardware when a channel trigger is generated. Clear TRIGF by reading EXTTRIG while TRIGF is set and then writing a 0 to TRIGF. Writing a 1 to TRIGF has no effect. If another channel trigger is generated before the clearing sequence is completed, the sequence is reset so TRIGF remains set after the clear sequence is completed for the earlier TRIGF. 0 No channel trigger was generated. 1 A channel trigger was generated.
6 INITTRIGEN	Initialization Trigger Enable Enables the generation of the trigger when the FTM counter is equal to the CNTIN register. 0 The generation of initialization trigger is disabled. 1 The generation of initialization trigger is enabled.
5 CH1TRIG	Channel 1 Trigger Enable Enables the generation of the channel trigger when the FTM counter is equal to the CnV register. 0 The generation of the channel trigger is disabled. 1 The generation of the channel trigger is enabled.
4 CH0TRIG	Channel 0 Trigger Enable Enables the generation of the channel trigger when the FTM counter is equal to the CnV register. 0 The generation of the channel trigger is disabled. 1 The generation of the channel trigger is enabled.
3 CH5TRIG	Channel 5 Trigger Enable Enables the generation of the channel trigger when the FTM counter is equal to the CnV register. 0 The generation of the channel trigger is disabled. 1 The generation of the channel trigger is enabled.
2 CH4TRIG	Channel 4 Trigger Enable Enables the generation of the channel trigger when the FTM counter is equal to the CnV register. 0 The generation of the channel trigger is disabled. 1 The generation of the channel trigger is enabled.
1 CH3TRIG	Channel 3 Trigger Enable Enables the generation of the channel trigger when the FTM counter is equal to the CnV register. 0 The generation of the channel trigger is disabled. 1 The generation of the channel trigger is enabled.
0 CH2TRIG	Channel 2 Trigger Enable Enables the generation of the channel trigger when the FTM counter is equal to the CnV register.

Table continues on the next page...

FTMx_EXTTRIG field descriptions (continued)

Field	Description
	0 The generation of the channel trigger is disabled. 1 The generation of the channel trigger is enabled.

43.3.17 Channels Polarity (FTMx_POL)

This register defines the output polarity of the FTM channels.

NOTE

> The safe value that is driven in a channel output when the fault control is enabled and a fault condition is detected is the inactive state of the channel. That is, the safe value of a channel is the value of its POL bit.

Address: Base address + 70h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved								POL7	POL6	POL5	POL4	POL3	POL2	POL1	POL0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

FTMx_POL field descriptions

Field	Description
31–8 Reserved	This field is reserved.
7 POL7	Channel 7 Polarity Defines the polarity of the channel output. This field is write protected. It can be written only when MODE[WPDIS] = 1. 0 The channel polarity is active high. 1 The channel polarity is active low.
6 POL6	Channel 6 Polarity Defines the polarity of the channel output. This field is write protected. It can be written only when MODE[WPDIS] = 1. 0 The channel polarity is active high. 1 The channel polarity is active low.
5 POL5	Channel 5 Polarity Defines the polarity of the channel output. This field is write protected. It can be written only when MODE[WPDIS] = 1.

Table continues on the next page...

FTMx_POL field descriptions (continued)

Field	Description
	0 The channel polarity is active high. 1 The channel polarity is active low.
4 POL4	Channel 4 Polarity Defines the polarity of the channel output. This field is write protected. It can be written only when MODE[WPDIS] = 1. 0 The channel polarity is active high. 1 The channel polarity is active low.
3 POL3	Channel 3 Polarity Defines the polarity of the channel output. This field is write protected. It can be written only when MODE[WPDIS] = 1. 0 The channel polarity is active high. 1 The channel polarity is active low.
2 POL2	Channel 2 Polarity Defines the polarity of the channel output. This field is write protected. It can be written only when MODE[WPDIS] = 1. 0 The channel polarity is active high. 1 The channel polarity is active low.
1 POL1	Channel 1 Polarity Defines the polarity of the channel output. This field is write protected. It can be written only when MODE[WPDIS] = 1. 0 The channel polarity is active high. 1 The channel polarity is active low.
0 POL0	Channel 0 Polarity Defines the polarity of the channel output. This field is write protected. It can be written only when MODE[WPDIS] = 1. 0 The channel polarity is active high. 1 The channel polarity is active low.

43.3.18 Fault Mode Status (FTMx_FMS)

This register contains the fault detection flags, write protection enable bit, and the logic OR of the enabled fault inputs.

Address: Base address + 74h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								FAULTF	WPEN	FAULTIN	0	FAULTF3	FAULTF2	FAULTF1	FAULTF0
W									0				0	0	0	0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

FTMx_FMS field descriptions

Field	Description
31–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7 FAULTF	Fault Detection Flag Represents the logic OR of the individual FAULTFj bits where j = 3, 2, 1, 0. Clear FAULTF by reading the FMS register while FAULTF is set and then writing a 0 to FAULTF while there is no existing fault condition at the enabled fault inputs. Writing a 1 to FAULTF has no effect. If another fault condition is detected in an enabled fault input before the clearing sequence is completed, the sequence is reset so FAULTF remains set after the clearing sequence is completed for the earlier fault condition. FAULTF is also cleared when FAULTFj bits are cleared individually. 0 No fault condition was detected. 1 A fault condition was detected.

Table continues on the next page...

FTMx_FMS field descriptions (continued)

Field	Description
6 WPEN	Write Protection Enable The WPEN bit is the negation of the WPDIS bit. WPEN is set when 1 is written to it. WPEN is cleared when WPEN bit is read as a 1 and then 1 is written to WPDIS. Writing 0 to WPEN has no effect. 0 Write protection is disabled. Write protected bits can be written. 1 Write protection is enabled. Write protected bits cannot be written.
5 FAULTIN	Fault Inputs Represents the logic OR of the enabled fault inputs after their filter (if their filter is enabled) when fault control is enabled. 0 The logic OR of the enabled fault inputs is 0. 1 The logic OR of the enabled fault inputs is 1.
4 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
3 FAULTF3	Fault Detection Flag 3 Set by hardware when fault control is enabled, the corresponding fault input is enabled and a fault condition is detected at the fault input. Clear FAULTF3 by reading the FMS register while FAULTF3 is set and then writing a 0 to FAULTF3 while there is no existing fault condition at the corresponding fault input. Writing a 1 to FAULTF3 has no effect. FAULTF3 bit is also cleared when FAULTF bit is cleared. If another fault condition is detected at the corresponding fault input before the clearing sequence is completed, the sequence is reset so FAULTF3 remains set after the clearing sequence is completed for the earlier fault condition. 0 No fault condition was detected at the fault input. 1 A fault condition was detected at the fault input.
2 FAULTF2	Fault Detection Flag 2 Set by hardware when fault control is enabled, the corresponding fault input is enabled and a fault condition is detected at the fault input. Clear FAULTF2 by reading the FMS register while FAULTF2 is set and then writing a 0 to FAULTF2 while there is no existing fault condition at the corresponding fault input. Writing a 1 to FAULTF2 has no effect. FAULTF2 bit is also cleared when FAULTF bit is cleared. If another fault condition is detected at the corresponding fault input before the clearing sequence is completed, the sequence is reset so FAULTF2 remains set after the clearing sequence is completed for the earlier fault condition. 0 No fault condition was detected at the fault input. 1 A fault condition was detected at the fault input.
1 FAULTF1	Fault Detection Flag 1 Set by hardware when fault control is enabled, the corresponding fault input is enabled and a fault condition is detected at the fault input. Clear FAULTF1 by reading the FMS register while FAULTF1 is set and then writing a 0 to FAULTF1 while there is no existing fault condition at the corresponding fault input. Writing a 1 to FAULTF1 has no effect. FAULTF1 bit is also cleared when FAULTF bit is cleared.

Table continues on the next page...

FTMx_FMS field descriptions (continued)

Field	Description
	If another fault condition is detected at the corresponding fault input before the clearing sequence is completed, the sequence is reset so FAULTF1 remains set after the clearing sequence is completed for the earlier fault condition. 0 No fault condition was detected at the fault input. 1 A fault condition was detected at the fault input.
0 FAULTF0	Fault Detection Flag 0 Set by hardware when fault control is enabled, the corresponding fault input is enabled and a fault condition is detected at the fault input. Clear FAULTF0 by reading the FMS register while FAULTF0 is set and then writing a 0 to FAULTF0 while there is no existing fault condition at the corresponding fault input. Writing a 1 to FAULTF0 has no effect. FAULTF0 bit is also cleared when FAULTF bit is cleared. If another fault condition is detected at the corresponding fault input before the clearing sequence is completed, the sequence is reset so FAULTF0 remains set after the clearing sequence is completed for the earlier fault condition. 0 No fault condition was detected at the fault input. 1 A fault condition was detected at the fault input.

43.3.19 Input Capture Filter Control (FTMx_FILTER)

This register selects the filter value for the inputs of channels.

Channels 4, 5, 6 and 7 do not have an input filter.

NOTE

Writing to the FILTER register has immediate effect and must be done only when the channels 0, 1, 2, and 3 are not in input modes. Failure to do this could result in a missing valid signal.

Address: Base address + 78h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved																CH3FVAL				CH2FVAL				CH1FVAL				CH0FVAL			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

FTMx_FILTER field descriptions

Field	Description
31–16 Reserved	This field is reserved.
15–12 CH3FVAL	Channel 3 Input Filter Selects the filter value for the channel input. The filter is disabled when the value is zero.

Table continues on the next page...

FTMx_FILTER field descriptions (continued)

Field	Description
11–8 CH2FVAL	Channel 2 Input Filter Selects the filter value for the channel input. The filter is disabled when the value is zero.
7–4 CH1FVAL	Channel 1 Input Filter Selects the filter value for the channel input. The filter is disabled when the value is zero.
CH0FVAL	Channel 0 Input Filter Selects the filter value for the channel input. The filter is disabled when the value is zero.

43.3.20 Fault Control (FTMx_FLTCTRL)

This register selects the filter value for the fault inputs, enables the fault inputs and the fault inputs filter.

Address: Base address + 7Ch offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0				FFVAL				FFLTR3EN	FFLTR2EN	FFLTR1EN	FFLTR0EN	FAULT3EN	FAULT2EN	FAULT1EN	FAULT0EN
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

FTMx_FLTCTRL field descriptions

Field	Description
31–12 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
11–8 FFVAL	Fault Input Filter Selects the filter value for the fault inputs. The fault filter is disabled when the value is zero.

Table continues on the next page...

FTMx_FLTCTRL field descriptions (continued)

Field	Description
	NOTE: Writing to this field has immediate effect and must be done only when the fault control or all fault inputs are disabled. Failure to do this could result in a missing fault detection.
7 FFLTR3EN	Fault Input 3 Filter Enable Enables the filter for the fault input. This field is write protected. It can be written only when MODE[WPDIS] = 1. 0 Fault input filter is disabled. 1 Fault input filter is enabled.
6 FFLTR2EN	Fault Input 2 Filter Enable Enables the filter for the fault input. This field is write protected. It can be written only when MODE[WPDIS] = 1. 0 Fault input filter is disabled. 1 Fault input filter is enabled.
5 FFLTR1EN	Fault Input 1 Filter Enable Enables the filter for the fault input. This field is write protected. It can be written only when MODE[WPDIS] = 1. 0 Fault input filter is disabled. 1 Fault input filter is enabled.
4 FFLTR0EN	Fault Input 0 Filter Enable Enables the filter for the fault input. This field is write protected. It can be written only when MODE[WPDIS] = 1. 0 Fault input filter is disabled. 1 Fault input filter is enabled.
3 FAULT3EN	Fault Input 3 Enable Enables the fault input. This field is write protected. It can be written only when MODE[WPDIS] = 1. 0 Fault input is disabled. 1 Fault input is enabled.
2 FAULT2EN	Fault Input 2 Enable Enables the fault input. This field is write protected. It can be written only when MODE[WPDIS] = 1. 0 Fault input is disabled. 1 Fault input is enabled.
1 FAULT1EN	Fault Input 1 Enable Enables the fault input. This field is write protected. It can be written only when MODE[WPDIS] = 1.

Table continues on the next page...

FTMx_FLTCTRL field descriptions (continued)

Field	Description
	0 Fault input is disabled. 1 Fault input is enabled.
0 FAULT0EN	Fault Input 0 Enable Enables the fault input. This field is write protected. It can be written only when MODE[WPDIS] = 1. 0 Fault input is disabled. 1 Fault input is enabled.

43.3.21 Quadrature Decoder Control And Status (FTMx_QDCTRL)

This register has the control and status bits for the Quadrature Decoder mode.

Address: Base address + 80h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								PHAFLTREN	PHBFLTREN	PHAPOL	PHBPOL	QUADMODE	QUADIR	TOFDIR	QUADEN
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

FTMx_QDCTRL field descriptions

Field	Description
31–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7 PHAFLTREN	Phase A Input Filter Enable Enables the filter for the quadrature decoder phase A input. The filter value for the phase A input is defined by the CH0FVAL field of FILTER. The phase A filter is also disabled when CH0FVAL is zero. 0 Phase A input filter is disabled. 1 Phase A input filter is enabled.
6 PHBFLTREN	Phase B Input Filter Enable Enables the filter for the quadrature decoder phase B input. The filter value for the phase B input is defined by the CH1FVAL field of FILTER. The phase B filter is also disabled when CH1FVAL is zero. 0 Phase B input filter is disabled. 1 Phase B input filter is enabled.
5 PHAPOL	Phase A Input Polarity Selects the polarity for the quadrature decoder phase A input. 0 Normal polarity. Phase A input signal is not inverted before identifying the rising and falling edges of this signal. 1 Inverted polarity. Phase A input signal is inverted before identifying the rising and falling edges of this signal.
4 PHBPOL	Phase B Input Polarity Selects the polarity for the quadrature decoder phase B input. 0 Normal polarity. Phase B input signal is not inverted before identifying the rising and falling edges of this signal. 1 Inverted polarity. Phase B input signal is inverted before identifying the rising and falling edges of this signal.
3 QUADMODE	Quadrature Decoder Mode Selects the encoding mode used in the Quadrature Decoder mode. 0 Phase A and phase B encoding mode. 1 Count and direction encoding mode.
2 QUADIR	FTM Counter Direction In Quadrature Decoder Mode Indicates the counting direction. 0 Counting direction is decreasing (FTM counter decrement). 1 Counting direction is increasing (FTM counter increment).
1 TOFDIR	Timer Overflow Direction In Quadrature Decoder Mode Indicates if the TOF bit was set on the top or the bottom of counting. 0 TOF bit was set on the bottom of counting. There was an FTM counter decrement and FTM counter changes from its minimum value (CNTIN register) to its maximum value (MOD register). 1 TOF bit was set on the top of counting. There was an FTM counter increment and FTM counter changes from its maximum value (MOD register) to its minimum value (CNTIN register).

Table continues on the next page...

FTMx_QDCTRL field descriptions (continued)

Field	Description
0 QUADEN	Quadrature Decoder Mode Enable Enables the Quadrature Decoder mode. In this mode, the phase A and B input signals control the FTM counter direction. The Quadrature Decoder mode has precedence over the other modes. See Table 43-2. This field is write protected. It can be written only when MODE[WPDIS] = 1. 0 Quadrature Decoder mode is disabled. 1 Quadrature Decoder mode is enabled.

43.3.22 Configuration (FTMx_CONF)

This register selects the number of times that the FTM counter overflow should occur before the TOF bit to be set, the FTM behavior in BDM modes, the use of an external global time base, and the global time base signal generation.

Address: Base address + 84h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0					GTBEOUT	GTBEEN	0	BDMMODE		0	NUMTOF				
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

FTMx_CONF field descriptions

Field	Description
31–11 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
10 GTBEOUT	Global Time Base Output Enables the global time base signal generation to other FTMs. 0 A global time base signal generation is disabled. 1 A global time base signal generation is enabled.
9 GTBEEN	Global Time Base Enable

Table continues on the next page...

FTMx_CONF field descriptions (continued)

Field	Description
	Configures the FTM to use an external global time base signal that is generated by another FTM. 0 Use of an external global time base is disabled. 1 Use of an external global time base is enabled.
8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7–6 BDMMODE	BDM Mode Selects the FTM behavior in BDM mode. See BDM mode.
5 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
NUMTOF	TOF Frequency Selects the ratio between the number of counter overflows to the number of times the TOF bit is set. NUMTOF = 0: The TOF bit is set for each counter overflow. NUMTOF = 1: The TOF bit is set for the first counter overflow but not for the next overflow. NUMTOF = 2: The TOF bit is set for the first counter overflow but not for the next 2 overflows. NUMTOF = 3: The TOF bit is set for the first counter overflow but not for the next 3 overflows. This pattern continues up to a maximum of 31.

43.3.23 FTM Fault Input Polarity (FTMx_FLTPOL)

This register defines the fault inputs polarity.

Address: Base address + 88h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0												FLT3POL	FLT2POL	FLT1POL	FLT0POL
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

FTMx_FLTPOL field descriptions

Field	Description
31–4 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

Table continues on the next page...

FTMx_FLTPOL field descriptions (continued)

Field	Description
3 FLT3POL	Fault Input 3 Polarity Defines the polarity of the fault input. This field is write protected. It can be written only when MODE[WPDIS] = 1. 0 The fault input polarity is active high. A 1 at the fault input indicates a fault. 1 The fault input polarity is active low. A 0 at the fault input indicates a fault.
2 FLT2POL	Fault Input 2 Polarity Defines the polarity of the fault input. This field is write protected. It can be written only when MODE[WPDIS] = 1. 0 The fault input polarity is active high. A 1 at the fault input indicates a fault. 1 The fault input polarity is active low. A 0 at the fault input indicates a fault.
1 FLT1POL	Fault Input 1 Polarity Defines the polarity of the fault input. This field is write protected. It can be written only when MODE[WPDIS] = 1. 0 The fault input polarity is active high. A 1 at the fault input indicates a fault. 1 The fault input polarity is active low. A 0 at the fault input indicates a fault.
0 FLT0POL	Fault Input 0 Polarity Defines the polarity of the fault input. This field is write protected. It can be written only when MODE[WPDIS] = 1. 0 The fault input polarity is active high. A 1 at the fault input indicates a fault. 1 The fault input polarity is active low. A 0 at the fault input indicates a fault.

43.3.24 Synchronization Configuration (FTMx_SYNCONF)

This register selects the PWM synchronization configuration, SWOCTRL, INVCTRL and CNTIN registers synchronization, if FTM clears the TRIGj bit, where j = 0, 1, 2, when the hardware trigger j is detected.

Address: Base address + 8Ch offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0											HWSOC	HWINVC	HWOM	HWWRBUF	HWRSTCNT
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0			SWSOC	SWINVC	SWOM	SWWRBUF	SWRSTCNT	SYNCMODE	0	SWOC	INVC	0	CNTINC	0	HWTRIGMODE
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

FTMx_SYNCONF field descriptions

Field	Description
31–21 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
20 HWSOC	Software output control synchronization is activated by a hardware trigger. 0 A hardware trigger does not activate the SWOCTRL register synchronization. 1 A hardware trigger activates the SWOCTRL register synchronization.
19 HWINVC	Inverting control synchronization is activated by a hardware trigger. 0 A hardware trigger does not activate the INVCTRL register synchronization. 1 A hardware trigger activates the INVCTRL register synchronization.
18 HWOM	Output mask synchronization is activated by a hardware trigger. 0 A hardware trigger does not activate the OUTMASK register synchronization. 1 A hardware trigger activates the OUTMASK register synchronization.
17 HWWRBUF	MOD, CNTIN, and CV registers synchronization is activated by a hardware trigger. 0 A hardware trigger does not activate MOD, CNTIN, and CV registers synchronization. 1 A hardware trigger activates MOD, CNTIN, and CV registers synchronization.
16 HWRSTCNT	FTM counter synchronization is activated by a hardware trigger. 0 A hardware trigger does not activate the FTM counter synchronization. 1 A hardware trigger activates the FTM counter synchronization.

Table continues on the next page...

FTMx_SYNCONF field descriptions (continued)

Field	Description
15–13 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
12 SWSOC	Software output control synchronization is activated by the software trigger. 0 The software trigger does not activate the SWOCTRL register synchronization. 1 The software trigger activates the SWOCTRL register synchronization.
11 SWINVC	Inverting control synchronization is activated by the software trigger. 0 The software trigger does not activate the INVCTRL register synchronization. 1 The software trigger activates the INVCTRL register synchronization.
10 SWOM	Output mask synchronization is activated by the software trigger. 0 The software trigger does not activate the OUTMASK register synchronization. 1 The software trigger activates the OUTMASK register synchronization.
9 SWWRBUF	MOD, CNTIN, and CV registers synchronization is activated by the software trigger. 0 The software trigger does not activate MOD, CNTIN, and CV registers synchronization. 1 The software trigger activates MOD, CNTIN, and CV registers synchronization.
8 SWRSTCNT	FTM counter synchronization is activated by the software trigger. 0 The software trigger does not activate the FTM counter synchronization. 1 The software trigger activates the FTM counter synchronization.
7 SYNCMODE	Synchronization Mode Selects the PWM Synchronization mode. 0 Legacy PWM synchronization is selected. 1 Enhanced PWM synchronization is selected.
6 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
5 SWOC	SWOCTRL Register Synchronization 0 SWOCTRL register is updated with its buffer value at all rising edges of system clock. 1 SWOCTRL register is updated with its buffer value by the PWM synchronization.
4 INVC	INVCTRL Register Synchronization 0 INVCTRL register is updated with its buffer value at all rising edges of system clock. 1 INVCTRL register is updated with its buffer value by the PWM synchronization.
3 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
2 CNTINC	CNTIN Register Synchronization 0 CNTIN register is updated with its buffer value at all rising edges of system clock. 1 CNTIN register is updated with its buffer value by the PWM synchronization.
1 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
0 HWTRIGMODE	Hardware Trigger Mode

Table continues on the next page...

FTMx_SYNCONF field descriptions (continued)

Field	Description
	0 FTM clears the TRIGj bit when the hardware trigger j is detected, where j = 0, 1,2. 1 FTM does not clear the TRIGj bit when the hardware trigger j is detected, where j = 0, 1,2.

43.3.25 FTM Inverting Control (FTMx_INVCTRL)

This register controls when the channel (n) output becomes the channel (n+1) output, and channel (n+1) output becomes the channel (n) output. Each INVmEN bit enables the inverting operation for the corresponding pair channels m.

This register has a write buffer. The INVmEN bit is updated by the INVCTRL register synchronization.

Address: Base address + 90h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0												INV3EN	INV2EN	INV1EN	INV0EN
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

FTMx_INVCTRL field descriptions

Field	Description
31–4 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
3 INV3EN	Pair Channels 3 Inverting Enable 0 Inverting is disabled. 1 Inverting is enabled.
2 INV2EN	Pair Channels 2 Inverting Enable 0 Inverting is disabled. 1 Inverting is enabled.
1 INV1EN	Pair Channels 1 Inverting Enable 0 Inverting is disabled. 1 Inverting is enabled.

Table continues on the next page...

FTMx_INVCTRL field descriptions (continued)

Field	Description
0 INV0EN	Pair Channels 0 Inverting Enable 0 Inverting is disabled. 1 Inverting is enabled.

43.3.26 FTM Software Output Control (FTMx_SWOCTRL)

This register enables software control of channel (n) output and defines the value forced to the channel (n) output:

- The CHnOC bits enable the control of the corresponding channel (n) output by software.
- The CHnOCV bits select the value that is forced at the corresponding channel (n) output.

This register has a write buffer. The fields are updated by the SWOCTRL register synchronization.

Address: Base address + 94h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CH7OCV	CH6OCV	CH5OCV	CH4OCV	CH3OCV	CH2OCV	CH1OCV	CH0OCV	CH7OC	CH6OC	CH5OC	CH4OC	CH3OC	CH2OC	CH1OC	CH0OC
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

FTMx_SWOCTRL field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
15 CH7OCV	Channel 7 Software Output Control Value 0 The software output control forces 0 to the channel output. 1 The software output control forces 1 to the channel output.
14 CH6OCV	Channel 6 Software Output Control Value

Table continues on the next page...

FTMx_SWOCTRL field descriptions (continued)

Field	Description
	0 The software output control forces 0 to the channel output. 1 The software output control forces 1 to the channel output.
13 CH5OCV	Channel 5 Software Output Control Value 0 The software output control forces 0 to the channel output. 1 The software output control forces 1 to the channel output.
12 CH4OCV	Channel 4 Software Output Control Value 0 The software output control forces 0 to the channel output. 1 The software output control forces 1 to the channel output.
11 CH3OCV	Channel 3 Software Output Control Value 0 The software output control forces 0 to the channel output. 1 The software output control forces 1 to the channel output.
10 CH2OCV	Channel 2 Software Output Control Value 0 The software output control forces 0 to the channel output. 1 The software output control forces 1 to the channel output.
9 CH1OCV	Channel 1 Software Output Control Value 0 The software output control forces 0 to the channel output. 1 The software output control forces 1 to the channel output.
8 CH0OCV	Channel 0 Software Output Control Value 0 The software output control forces 0 to the channel output. 1 The software output control forces 1 to the channel output.
7 CH7OC	Channel 7 Software Output Control Enable 0 The channel output is not affected by software output control. 1 The channel output is affected by software output control.
6 CH6OC	Channel 6 Software Output Control Enable 0 The channel output is not affected by software output control. 1 The channel output is affected by software output control.
5 CH5OC	Channel 5 Software Output Control Enable 0 The channel output is not affected by software output control. 1 The channel output is affected by software output control.
4 CH4OC	Channel 4 Software Output Control Enable 0 The channel output is not affected by software output control. 1 The channel output is affected by software output control.
3 CH3OC	Channel 3 Software Output Control Enable 0 The channel output is not affected by software output control. 1 The channel output is affected by software output control.
2 CH2OC	Channel 2 Software Output Control Enable

Table continues on the next page...

FTMx_SWOCTRL field descriptions (continued)

Field	Description
	0 The channel output is not affected by software output control. 1 The channel output is affected by software output control.
1 CH1OC	Channel 1 Software Output Control Enable 0 The channel output is not affected by software output control. 1 The channel output is affected by software output control.
0 CH0OC	Channel 0 Software Output Control Enable 0 The channel output is not affected by software output control. 1 The channel output is affected by software output control.

43.3.27 FTM PWM Load (FTMx_PWMLOAD)

Enables the loading of the MOD, CNTIN, C(n)V, and C(n+1)V registers with the values of their write buffers when the FTM counter changes from the MOD register value to its next value or when a channel (j) match occurs. A match occurs for the channel (j) when FTM counter = C(j)V.

Address: Base address + 98h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0						LDOK	0	CH7SEL	CH6SEL	CH5SEL	CH4SEL	CH3SEL	CH2SEL	CH1SEL	CH0SEL
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

FTMx_PWMLOAD field descriptions

Field	Description
31–10 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
9 LDOK	Load Enable Enables the loading of the MOD, CNTIN, and CV registers with the values of their write buffers. 0 Loading updated values is disabled. 1 Loading updated values is enabled.

Table continues on the next page...

FTMx_PWMLOAD field descriptions (continued)

Field	Description
8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7 CH7SEL	Channel 7 Select 0 Do not include the channel in the matching process. 1 Include the channel in the matching process.
6 CH6SEL	Channel 6 Select 0 Do not include the channel in the matching process. 1 Include the channel in the matching process.
5 CH5SEL	Channel 5 Select 0 Do not include the channel in the matching process. 1 Include the channel in the matching process.
4 CH4SEL	Channel 4 Select 0 Do not include the channel in the matching process. 1 Include the channel in the matching process.
3 CH3SEL	Channel 3 Select 0 Do not include the channel in the matching process. 1 Include the channel in the matching process.
2 CH2SEL	Channel 2 Select 0 Do not include the channel in the matching process. 1 Include the channel in the matching process.
1 CH1SEL	Channel 1 Select 0 Do not include the channel in the matching process. 1 Include the channel in the matching process.
0 CH0SEL	Channel 0 Select 0 Do not include the channel in the matching process. 1 Include the channel in the matching process.

43.4 Functional description

The notation used in this document to represent the counters and the generation of the signals is shown in the following figure.

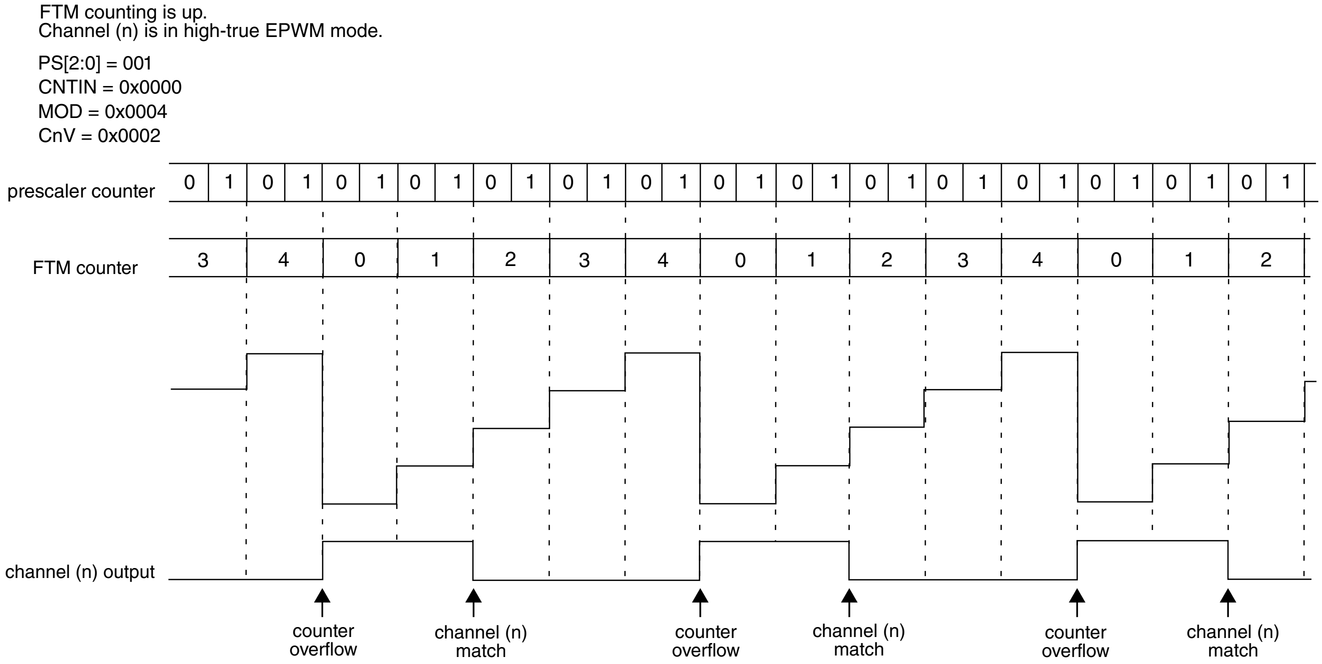


Figure 43-2. Notation used

43.4.1 Clock source

The FTM has only one clock domain: the system clock.

43.4.1.1 Counter clock source

The CLKS[1:0] bits in the SC register selects clock sources for the FTM counter or disables the FTM counter. After any chip reset, CLKS[1:0] = 0:0 so no clock source is selected.

The CLKS[1:0] bits may be read or written at any time. Disabling the FTM counter by writing 0:0 to the CLKS[1:0] bits does not affect the FTM counter value or other registers.

The external clock passes through a synchronizer clocked by the system clock to assure that counter transitions are properly aligned to system clock transitions.Therefore, to meet Nyquist criteria considering also jitter, the frequency of the external clock source must not exceed 1/4 of the system clock frequency.

43.4.2 Prescaler

The selected counter clock source passes through a prescaler that is a 7-bit counter. The value of the prescaler is selected by the PS[2:0] bits. The following figure shows an example of the prescaler counter and FTM counter.

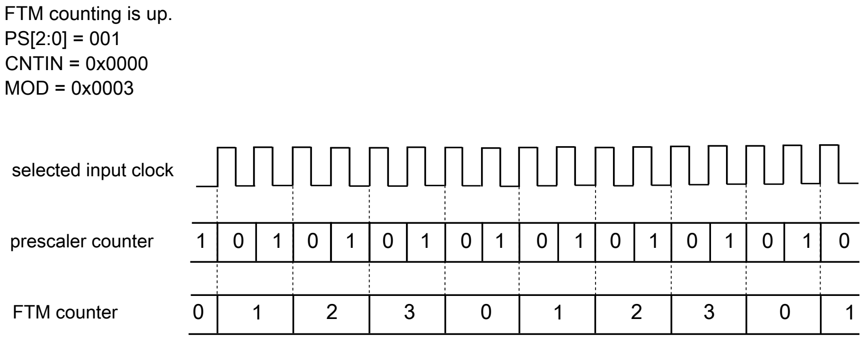


Figure 43-3. Example of the prescaler counter

43.4.3 Counter

The FTM has a 16-bit counter that is used by the channels either for input or output modes. The FTM counter clock is the selected clock divided by the prescaler.

The FTM counter has these modes of operation:

- Up counting
- Up-down counting
- Quadrature Decoder mode

43.4.3.1 Up counting

Up counting is selected when:

- QUADEN = 0, and
- CPWMS = 0

CNTIN defines the starting value of the count and MOD defines the final value of the count, see the following figure. The value of CNTIN is loaded into the FTM counter, and the counter increments until the value of MOD is reached, at which point the counter is reloaded with the value of CNTIN.

The FTM period when using up counting is (MOD – CNTIN + 0x0001) × period of the FTM counter clock.

The TOF bit is set when the FTM counter changes from MOD to CNTIN.

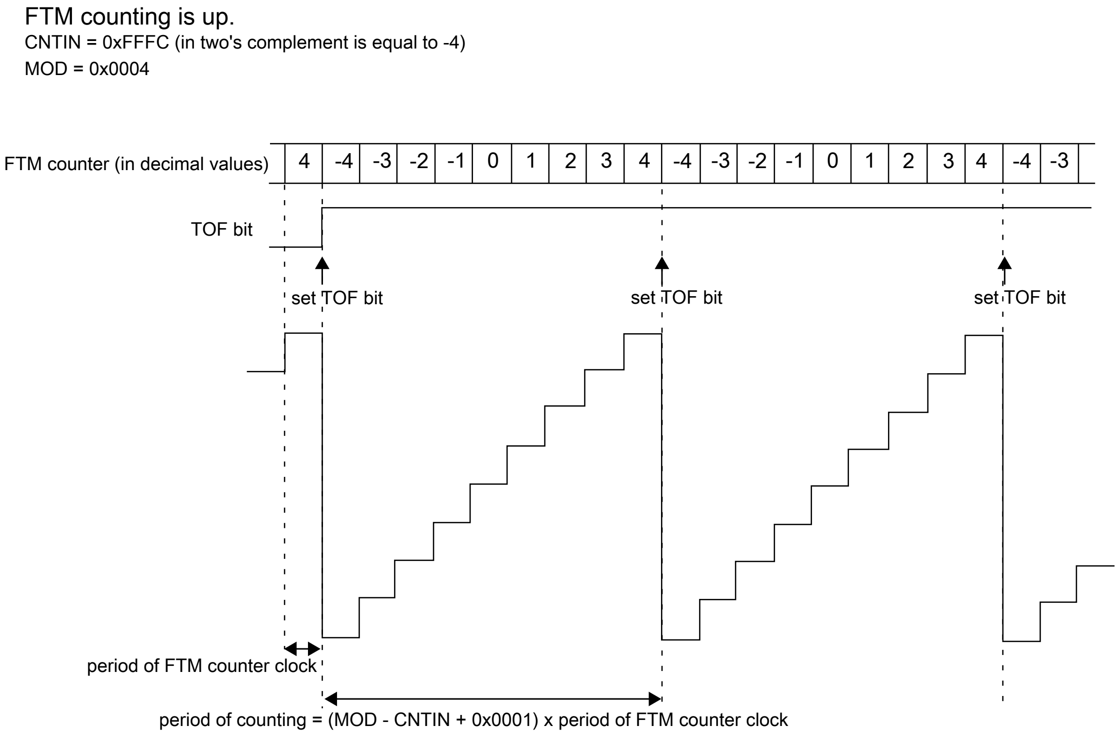

FTM counting is up.
CNTIN = 0xFFFC (in two's complement is equal to -4)
MOD = 0x0004
FTM counter (in decimal values)
4 -4 -3 -2 -1 0 1 2 3 4 -4 -3 -2 -1 0 1 2 3 4 -4 -3
TOF bit
set TOF bit
set TOF bit
set TOF bit
period of FTM counter clock
period of counting = (MOD - CNTIN + 0x0001) x period of FTM counter clock

Figure 43-4. Example of FTM up and signed counting

Table 43-4. FTM counting based on CNTIN value

When	Then
CNTIN = 0x0000	The FTM counting is equivalent to TPM up counting, that is, up and unsigned counting. See the following figure.
CNTIN[15] = 1	The initial value of the FTM counter is a negative number in two's complement, so the FTM counting is up and signed.
CNTIN[15] = 0 and CNTIN ≠ 0x0000	The initial value of the FTM counter is a positive number, so the FTM counting is up and unsigned.

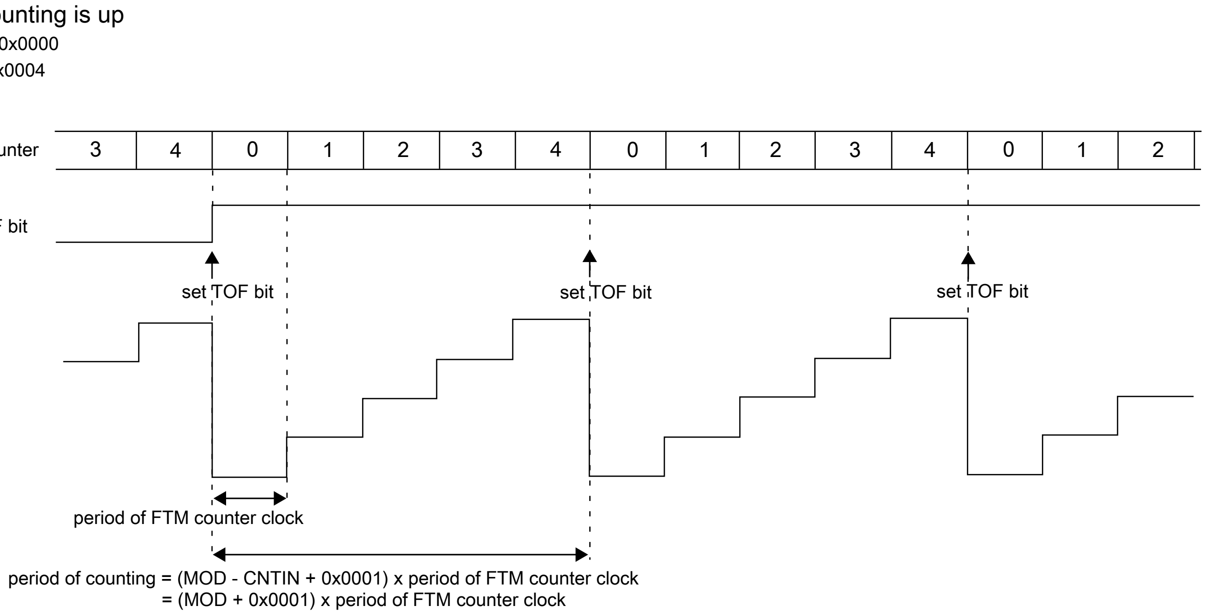

FTM counting is up
CNTIN = 0x0000
MOD = 0x0004
FTM counter
3
4
0
1
2
3
4
0
1
2
3
4
0
1
2
TOF bit
set TOF bit
set TOF bit
set TOF bit
period of FTM counter clock
period of counting = (MOD - CNTIN + 0x0001) x period of FTM counter clock
= (MOD + 0x0001) x period of FTM counter clock

Figure 43-5. Example of FTM up counting with CNTIN = 0x0000

Note

- FTM operation is only valid when the value of the CNTIN register is less than the value of the MOD register, either in the unsigned counting or signed counting. It is the responsibility of the software to ensure that the values in the CNTIN and MOD registers meet this requirement. Any values of CNTIN and MOD that do not satisfy this criteria can result in unpredictable behavior.
- MOD = CNTIN is a redundant condition. In this case, the FTM counter is always equal to MOD and the TOF bit is set in each rising edge of the FTM counter clock.
- When MOD = 0x0000, CNTIN = 0x0000, for example after reset, and FTMEN = 1, the FTM counter remains stopped at 0x0000 until a non-zero value is written into the MOD or CNTIN registers.
- Setting CNTIN to be greater than the value of MOD is not recommended as this unusual setting may make the FTM operation difficult to comprehend. However, there is no restriction on this configuration, and an example is shown in the following figure.

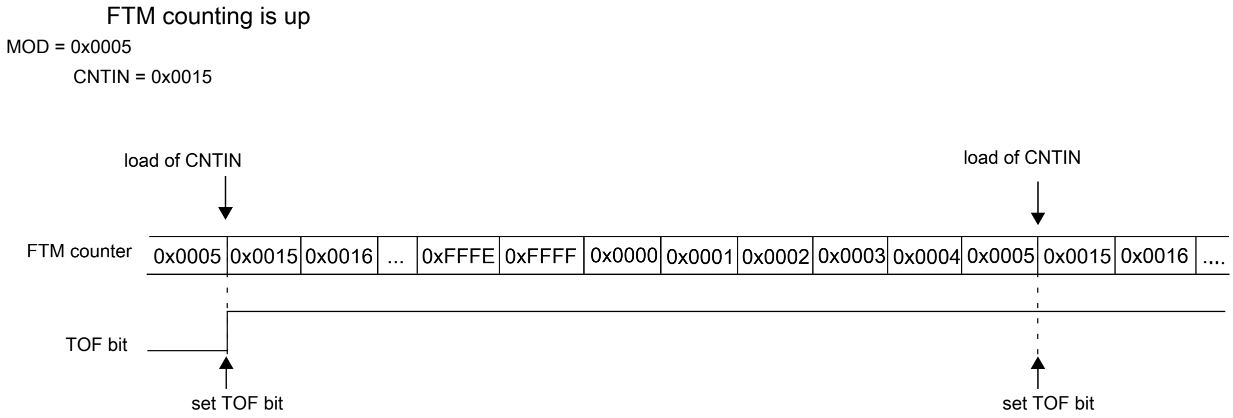


Figure 43-6. Example of up counting when the value of CNTIN is greater than the value of MOD

43.4.3.2 Up-down counting

Up-down counting is selected when:

- QUADEN = 0, and
- CPWMS = 1

CNTIN defines the starting value of the count and MOD defines the final value of the count. The value of CNTIN is loaded into the FTM counter, and the counter increments until the value of MOD is reached, at which point the counter is decremented until it returns to the value of CNTIN and the up-down counting restarts.

The FTM period when using up-down counting is 2 × (MOD – CNTIN) × period of the FTM counter clock.

The TOF bit is set when the FTM counter changes from MOD to (MOD – 1).

If (CNTIN = 0x0000), the FTM counting is equivalent to TPM up-down counting, that is, up-down and unsigned counting. See the following figure.

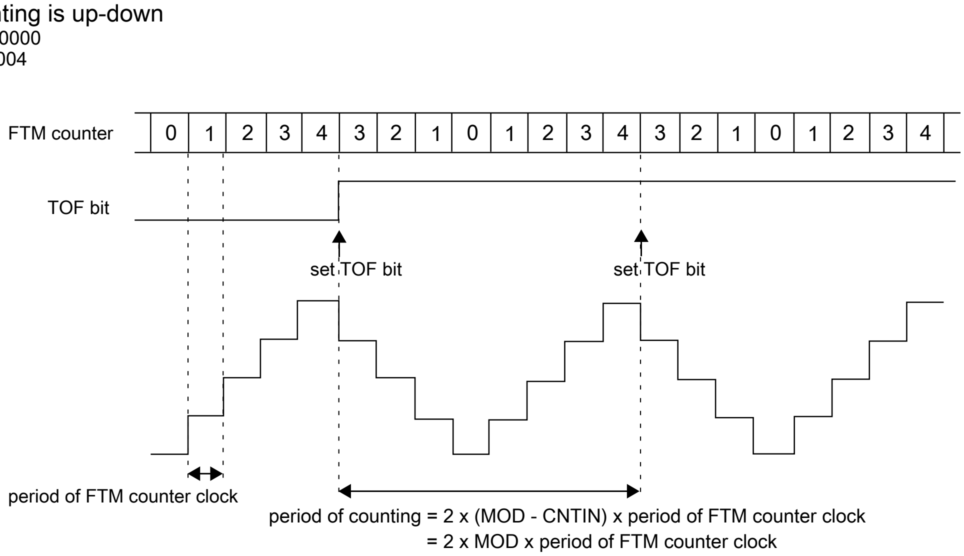

FTM counting is up-down
CNTIN = 0x0000
MOD = 0x0004
FTM counter
0 1 2 3 4 3 2 1 0 1 2 3 4 3 2 1 0 1 2 3 4
TOF bit
set TOF bit
set TOF bit
period of FTM counter clock
period of counting = 2 x (MOD - CNTIN) x period of FTM counter clock
= 2 x MOD x period of FTM counter clock

Figure 43-7. Example of up-down counting when CNTIN = 0x0000

Note

When CNTIN is different from zero in the up-down counting, a valid CPWM signal is generated:

- if CnV > CNTIN, or
- if CnV = 0 or if CnV[15] = 1. In this case, 0% CPWM is generated.

43.4.3.3 Free running counter

If (FTMEN = 0) and (MOD = 0x0000 or MOD = 0xFFFF), the FTM counter is a free running counter. In this case, the FTM counter runs free from 0x0000 through 0xFFFF and the TOF bit is set when the FTM counter changes from 0xFFFF to 0x0000. See the following figure.

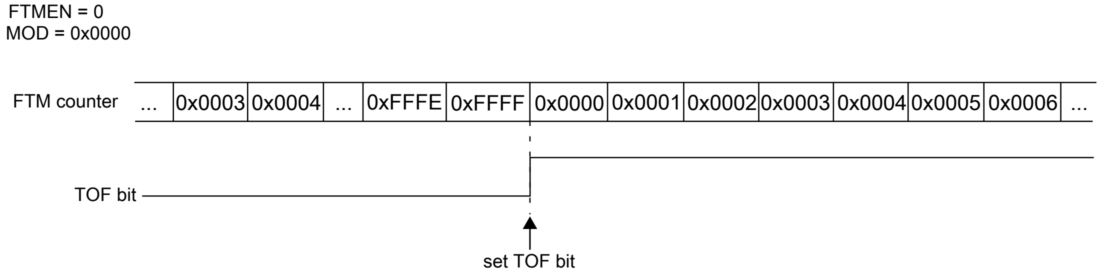

FTMEN = 0
MOD = 0x0000
FTM counter
... 0x0003 0x0004 ... 0xFFFE 0xFFFF 0x0000 0x0001 0x0002 0x0003 0x0004 0x0005 0x0006 ...
TOF bit
set TOF bit

Figure 43-8. Example when the FTM counter is free running

The FTM counter is also a free running counter when:

- FTMEN = 1
- QUADEN = 0
- CPWMS = 0
- CNTIN = 0x0000, and
- MOD = 0xFFFF

43.4.3.4 Counter reset

Any one of the following cases resets the FTM counter to the value in the CNTIN register and the channels output to its initial value, except for channels in Output Compare mode.

- Any write to CNT.
- FTM counter synchronization.

43.4.3.5 When the TOF bit is set

The NUMTOF[4:0] bits define the number of times that the FTM counter overflow should occur before the TOF bit to be set. If NUMTOF[4:0] = 0x00, then the TOF bit is set at each FTM counter overflow.

Initialize the FTM counter, by writing to CNT, after writing to the NUMTOF[4:0] bits to avoid confusion about when the first counter overflow will occur.

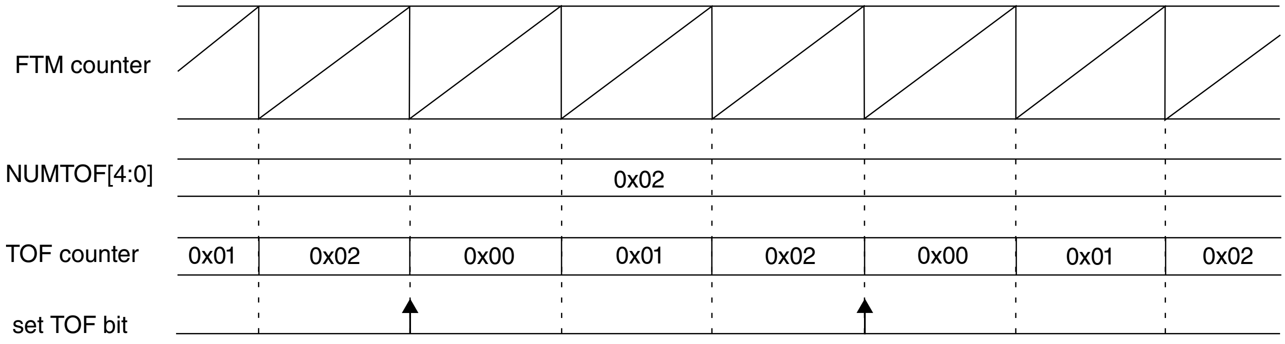


Figure 43-9. Periodic TOF when NUMTOF = 0x02

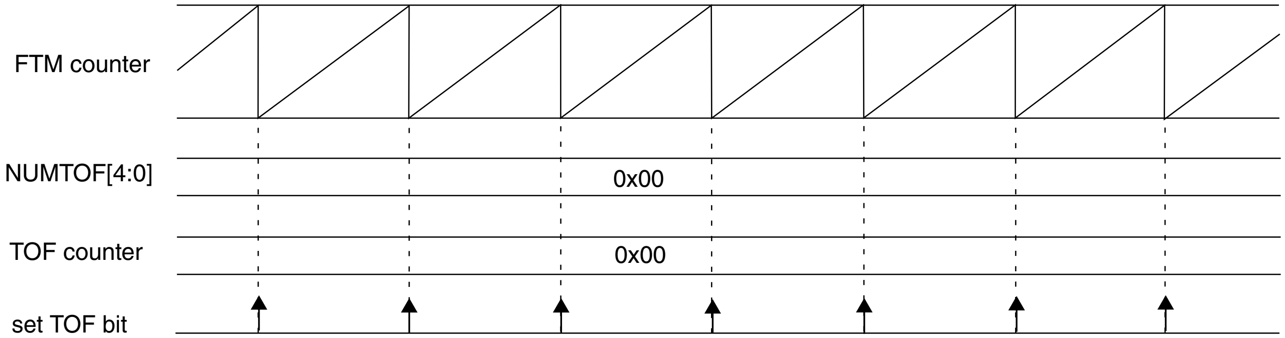


Figure 43-10. Periodic TOF when NUMTOF = 0x00

43.4.4 Input Capture mode

The Input Capture mode is selected when:

- DECAPEN = 0
- COMBINE = 0
- CPWMS = 0
- MSnB:MSnA = 0:0, and
- ELSnB:ELSnA ≠ 0:0

When a selected edge occurs on the channel input, the current value of the FTM counter is captured into the CnV register, at the same time the CHnF bit is set and the channel interrupt is generated if enabled by CHnIE = 1. See the following figure.

When a channel is configured for input capture, the FTMxCHn pin is an edge-sensitive input. ELSnB:ELSnA control bits determine which edge, falling or rising, triggers input-capture event. Note that the maximum frequency for the channel input signal to be detected correctly is system clock divided by 4, which is required to meet Nyquist criteria for signal sampling.

Writes to the CnV register is ignored in Input Capture mode.

While in BDM, the input capture function works as configured. When a selected edge event occurs, the FTM counter value, which is frozen because of BDM, is captured into the CnV register and the CHnF bit is set.

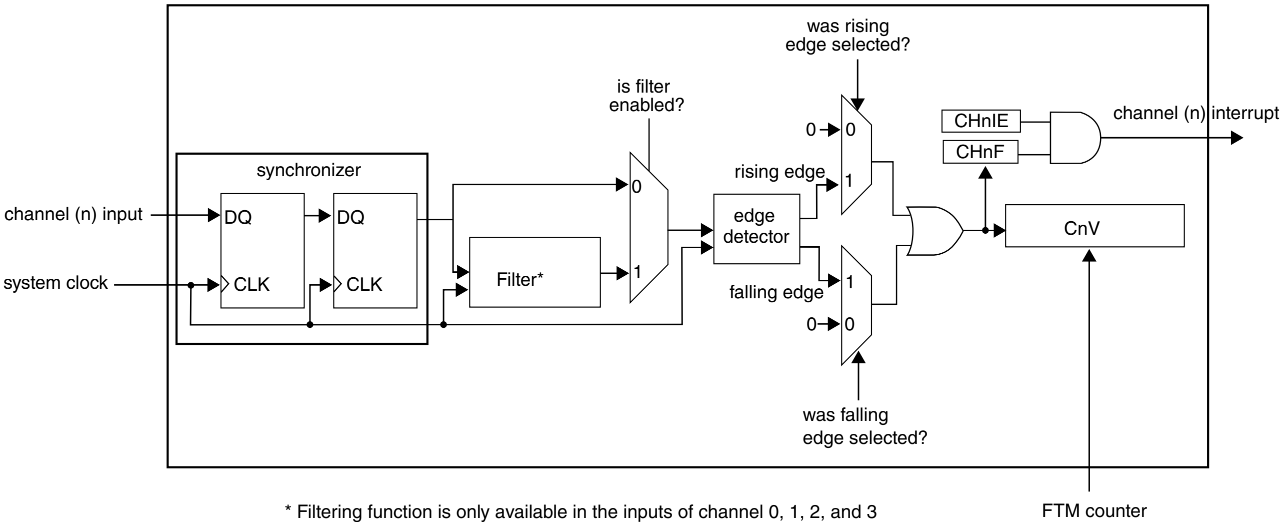


Figure 43-11. Input Capture mode

If the channel input does not have a filter enabled, then the input signal is always delayed 3 rising edges of the system clock, that is, two rising edges to the synchronizer plus one more rising edge to the edge detector. In other words, the CHnF bit is set on the third rising edge of the system clock after a valid edge occurs on the channel input.

43.4.4.1 Filter for Input Capture mode

The filter function is only available on channels 0, 1, 2, and 3.

First, the input signal is synchronized by the system clock. Following synchronization, the input signal enters the filter block. See the following figure.

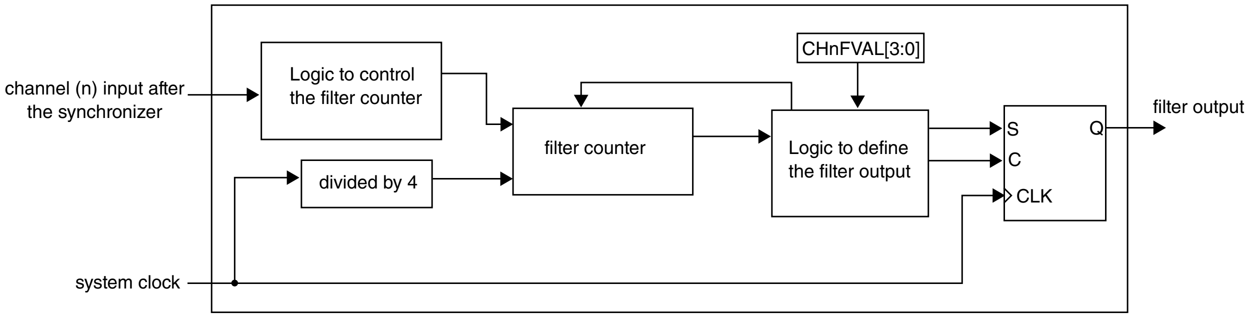


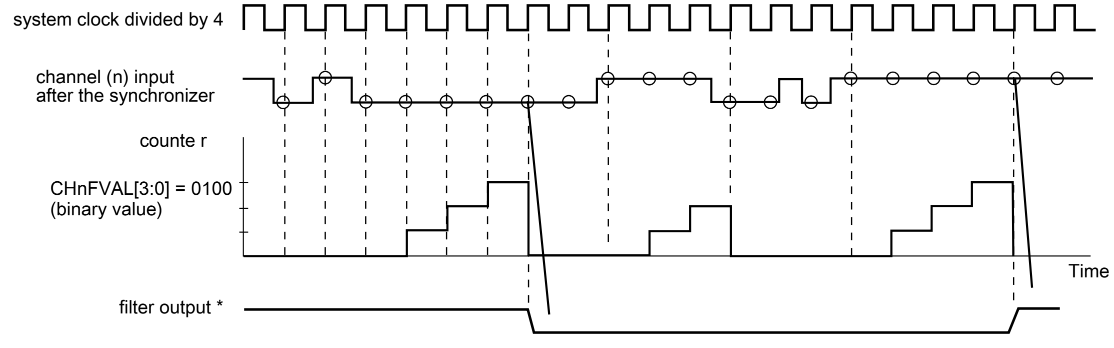
Figure 43-12. Channel input filter

When there is a state change in the input signal, the counter is reset and starts counting up. As long as the new state is stable on the input, the counter continues to increment. When the counter is equal to CHnFVAL[3:0], the state change of the input signal is validated. It is then transmitted as a pulse edge to the edge detector.

If the opposite edge appears on the input signal before it can be validated, the counter is reset. At the next input transition, the counter starts counting again. Any pulse that is shorter than the minimum value selected by CHnFVAL[3:0] (× 4 system clocks) is regarded as a glitch and is not passed on to the edge detector. A timing diagram of the input filter is shown in the following figure.

The filter function is disabled when CHnFVAL[3:0] bits are zero. In this case, the input signal is delayed 3 rising edges of the system clock. If (CHnFVAL[3:0] ≠ 0000), then the input signal is delayed by the minimum pulse width (CHnFVAL[3:0] × 4 system clocks) plus a further 4 rising edges of the system clock: two rising edges to the synchronizer, one rising edge to the filter output, plus one more to the edge detector. In other words, CHnF is set (4 + 4 × CHnFVAL[3:0]) system clock periods after a valid edge occurs on the channel input.

The clock for the counter in the channel input filter is the system clock divided by 4.



* Note: Filter output is delayed one system clock of filter counter logic output.

Figure 43-13. Channel input filter example

The figure below shows an example of input capture with filter enabled and the delay added by each part of the input capture logic. Note that the input signal is delayed only by the synchronizer and edge dector logic if the filter is disabled.

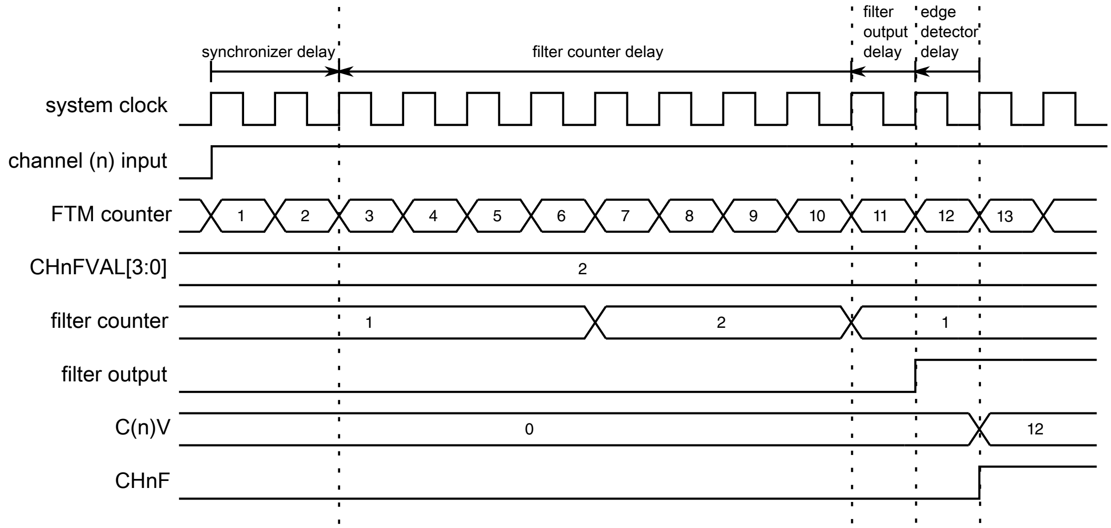


Figure 43-14. Input capture example

43.4.5 Output Compare mode

The Output Compare mode is selected when:

- DECAPEN = 0
- COMBINE = 0
- CPWMS = 0, and
- MSnB:MSnA = 0:1

In Output Compare mode, the FTM can generate timed pulses with programmable position, polarity, duration, and frequency. When the counter matches the value in the CnV register of an output compare channel, the channel (n) output can be set, cleared, or toggled.

When a channel is initially configured to Toggle mode, the previous value of the channel output is held until the first output compare event occurs.

The CHnF bit is set and the channel (n) interrupt is generated if CHnIE = 1 at the channel (n) match (FTM counter = CnV).

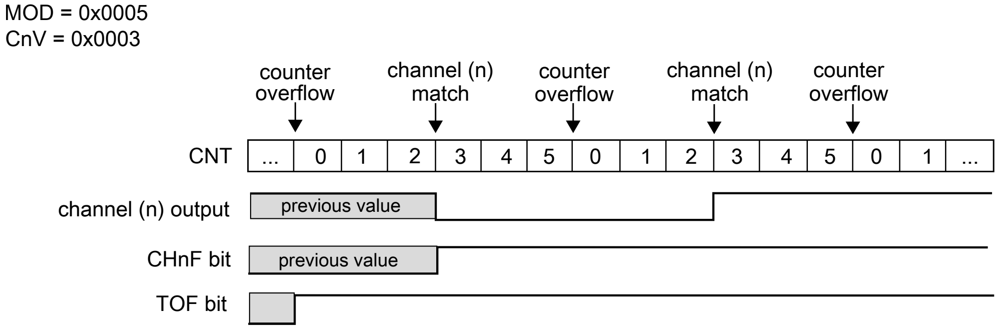


Figure 43-15. Example of the Output Compare mode when the match toggles the channel output

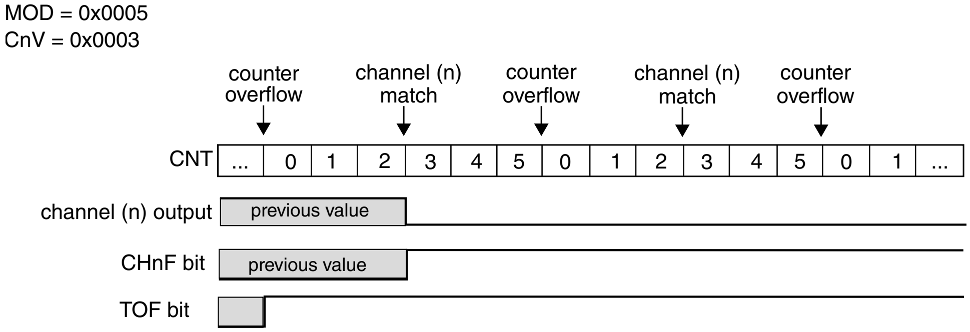


Figure 43-16. Example of the Output Compare mode when the match clears the channel output

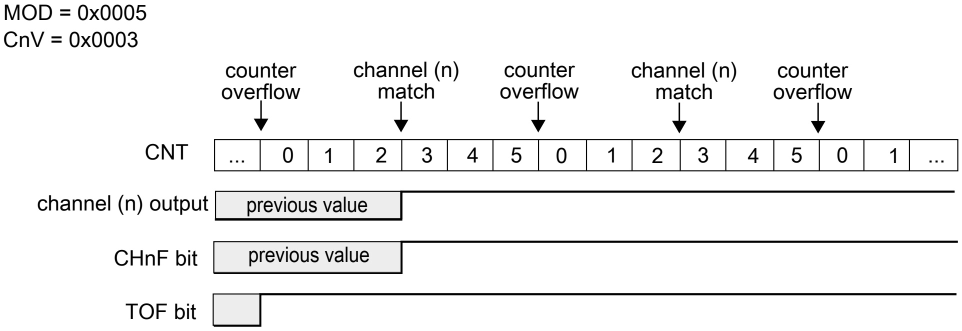


Figure 43-17. Example of the Output Compare mode when the match sets the channel output

If (ELSnB:ELSnA = 0:0) when the counter reaches the value in the CnV register, the CHnF bit is set and the channel (n) interrupt is generated if CHnIE = 1, however the channel (n) output is not modified and controlled by FTM.

43.4.6 Edge-Aligned PWM (EPWM) mode

The Edge-Aligned mode is selected when:

- QUADEN = 0

- DECAPEN = 0
- COMBINE = 0
- CPWMS = 0, and
- MSnB = 1

The EPWM period is determined by (MOD − CNTIN + 0x0001) and the pulse width (duty cycle) is determined by (CnV − CNTIN).

The CHnF bit is set and the channel (n) interrupt is generated if CHnIE = 1 at the channel (n) match (FTM counter = CnV), that is, at the end of the pulse width.

This type of PWM signal is called edge-aligned because the leading edges of all PWM signals are aligned with the beginning of the period, which is the same for all channels within an FTM.

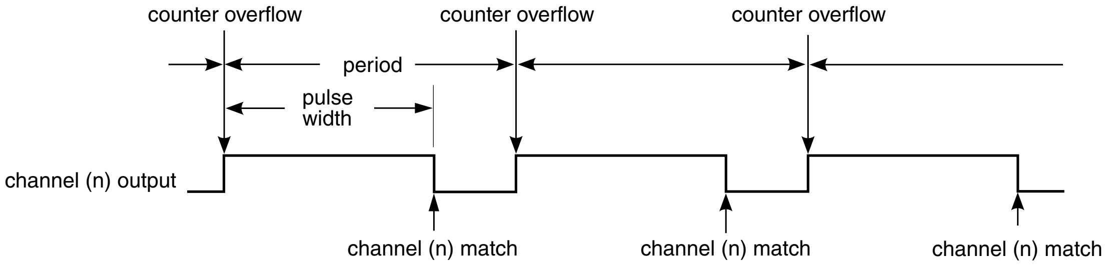


Figure 43-18. EPWM period and pulse width with ELSnB:ELSnA = 1:0

If (ELSnB:ELSnA = 0:0) when the counter reaches the value in the CnV register, the CHnF bit is set and the channel (n) interrupt is generated if CHnIE = 1, however the channel (n) output is not controlled by FTM.

If (ELSnB:ELSnA = 1:0), then the channel (n) output is forced high at the counter overflow when the CNTIN register value is loaded into the FTM counter, and it is forced low at the channel (n) match (FTM counter = CnV). See the following figure.

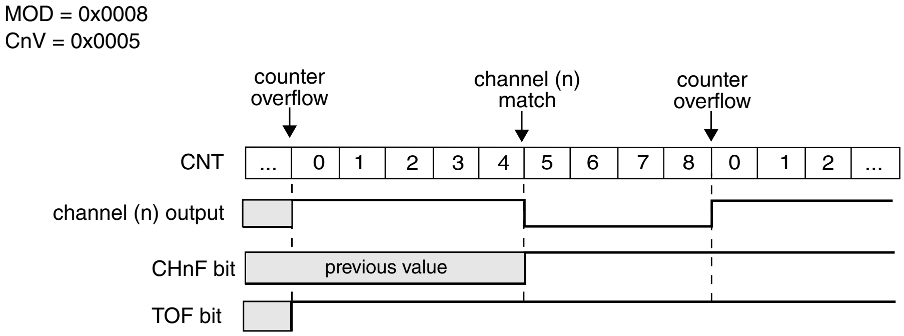


Figure 43-19. EPWM signal with ELSnB:ELSnA = 1:0

If (ELSnB:ELSnA = X:1), then the channel (n) output is forced low at the counter overflow when the CNTIN register value is loaded into the FTM counter, and it is forced high at the channel (n) match (FTM counter = CnV). See the following figure.

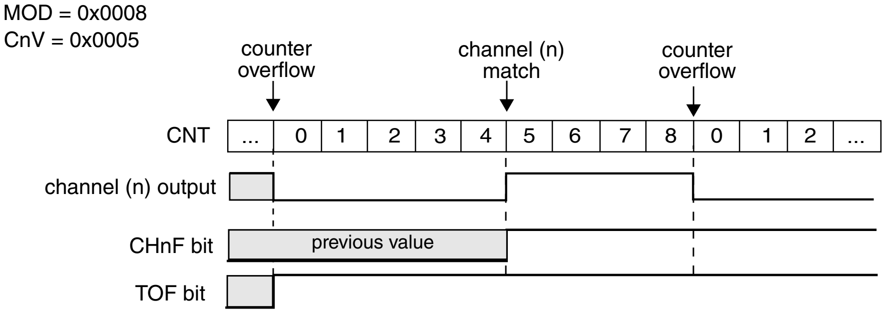


Figure 43-20. EPWM signal with ELSnB:ELSnA = X:1

If (CnV = 0x0000), then the channel (n) output is a 0% duty cycle EPWM signal and CHnF bit is not set even when there is the channel (n) match.

If (CnV > MOD), then the channel (n) output is a 100% duty cycle EPWM signal and CHnF bit is not set.Therefore, MOD must be less than 0xFFFF in order to get a 100% duty cycle EPWM signal.

Note

When CNTIN is different from zero the following EPWM signals can be generated:

- 0% EPWM signal if CnV = CNTIN,
- EPWM signal between 0% and 100% if CNTIN < CnV <= MOD,
- 100% EPWM signal when CNTIN > CnV or CnV > MOD.

43.4.7 Center-Aligned PWM (CPWM) mode

The Center-Aligned mode is selected when:

- QUADEN = 0
- DECAPEN = 0
- COMBINE = 0, and
- CPWMS = 1

The CPWM pulse width (duty cycle) is determined by 2 × (CnV − CNTIN) and the period is determined by 2 × (MOD − CNTIN). See the following figure. MOD must be kept in the range of 0x0001 to 0x7FFF because values outside this range can produce ambiguous results.

In the CPWM mode, the FTM counter counts up until it reaches MOD and then counts down until it reaches CNTIN.

The CHnF bit is set and channel (n) interrupt is generated (if CHnIE = 1) at the channel (n) match (FTM counter = CnV) when the FTM counting is down (at the begin of the pulse width) and when the FTM counting is up (at the end of the pulse width).

This type of PWM signal is called center-aligned because the pulse width centers for all channels are aligned with the value of CNTIN.

The other channel modes are not compatible with the up-down counter (CPWMS = 1). Therefore, all FTM channels must be used in CPWM mode when (CPWMS = 1).

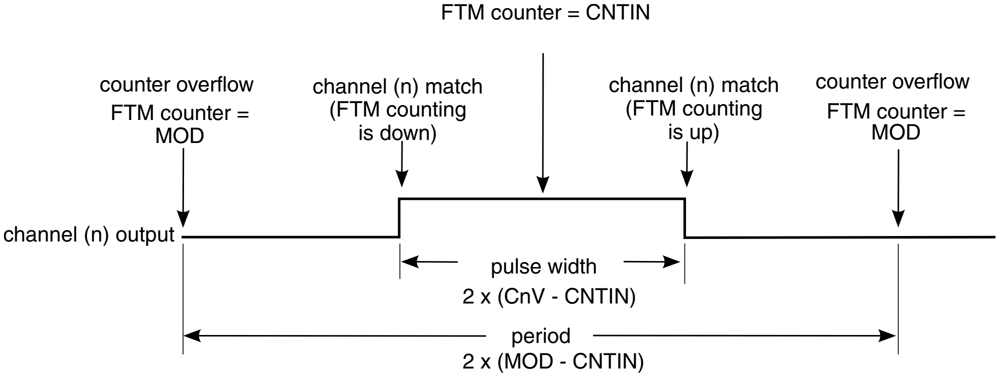


Figure 43-21. CPWM period and pulse width with ELSnB:ELSnA = 1:0

If (ELSnB:ELSnA = 0:0) when the FTM counter reaches the value in the CnV register, the CHnF bit is set and the channel (n) interrupt is generated (if CHnIE = 1), however the channel (n) output is not controlled by FTM.

If (ELSnB:ELSnA = 1:0), then the channel (n) output is forced high at the channel (n) match (FTM counter = CnV) when counting down, and it is forced low at the channel (n) match when counting up. See the following figure.

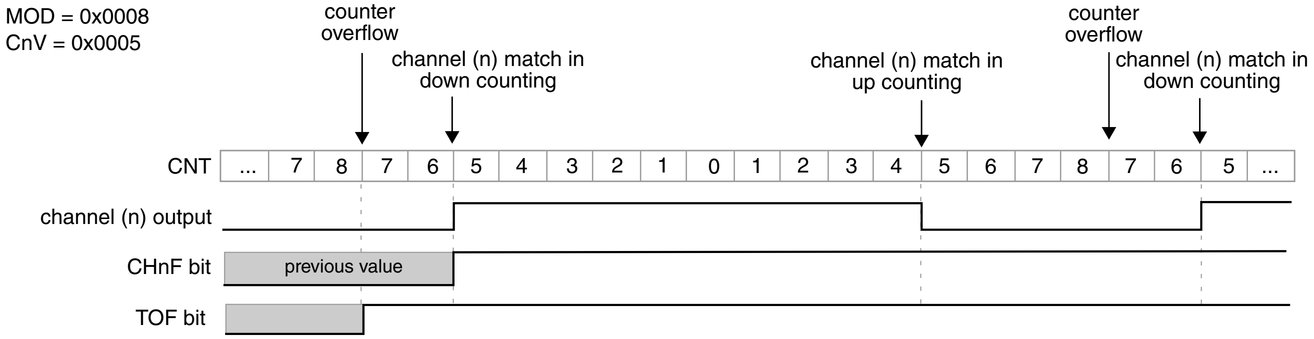


Figure 43-22. CPWM signal with ELSnB:ELSnA = 1:0

If (ELSnB:ELSnA = X:1), then the channel (n) output is forced low at the channel (n) match (FTM counter = CnV) when counting down, and it is forced high at the channel (n) match when counting up. See the following figure.

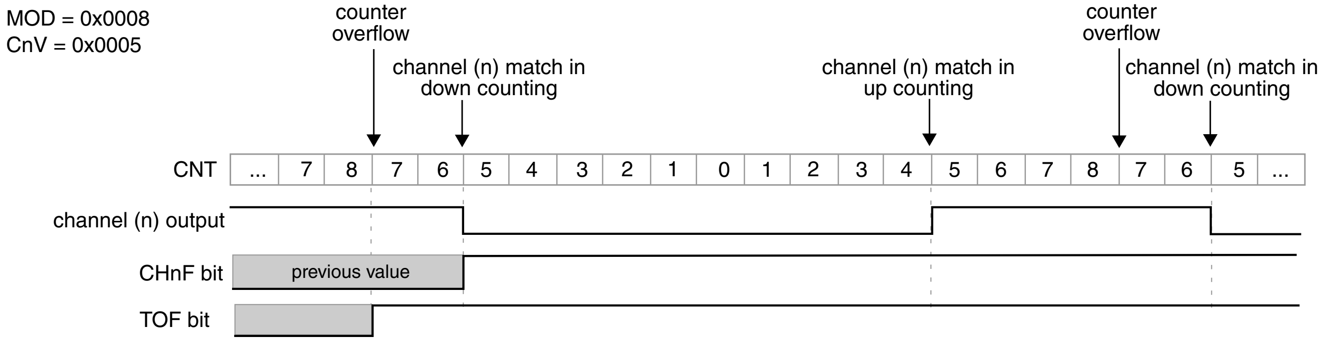


Figure 43-23. CPWM signal with ELSnB:ELSnA = X:1

If (CnV = 0x0000) or CnV is a negative value, that is (CnV[15] = 1), then the channel (n) output is a 0% duty cycle CPWM signal and CHnF bit is not set even when there is the channel (n) match.

If CnV is a positive value, that is (CnV[15] = 0), (CnV ≥ MOD), and (MOD ≠ 0x0000), then the channel (n) output is a 100% duty cycle CPWM signal and CHnF bit is not set even when there is the channel (n) match. This implies that the usable range of periods set by MOD is 0x0001 through 0x7FFE, 0x7FFF if you do not need to generate a 100% duty cycle CPWM signal. This is not a significant limitation because the resulting period is much longer than required for normal applications.

The CPWM mode must not be used when the FTM counter is a free running counter.

43.4.8 Combine mode

The Combine mode is selected when:

- QUADEN = 0
- DECAPEN = 0
- COMBINE = 1, and
- CPWMS = 0

In Combine mode, an even channel (n) and adjacent odd channel (n+1) are combined to generate a PWM signal in the channel (n) output.

In the Combine mode, the PWM period is determined by (MOD − CNTIN + 0x0001) and the PWM pulse width (duty cycle) is determined by (|C(n+1)V − C(n)V|).

The CHnF bit is set and the channel (n) interrupt is generated (if CHnIE = 1) at the channel (n) match (FTM counter = C(n)V). The CH(n+1)F bit is set and the channel (n +1) interrupt is generated, if CH(n+1)IE = 1, at the channel (n+1) match (FTM counter = C(n+1)V).

If (ELSnB:ELSnA = 1:0), then the channel (n) output is forced low at the beginning of the period (FTM counter = CNTIN) and at the channel (n+1) match (FTM counter = C(n +1)V). It is forced high at the channel (n) match (FTM counter = C(n)V). See the following figure.

If (ELSnB:ELSnA = X:1), then the channel (n) output is forced high at the beginning of the period (FTM counter = CNTIN) and at the channel (n+1) match (FTM counter = C(n +1)V). It is forced low at the channel (n) match (FTM counter = C(n)V). See the following figure.

In Combine mode, the ELS(n+1)B and ELS(n+1)A bits are not used in the generation of the channels (n) and (n+1) output. However, if (ELSnB:ELSnA = 0:0) then the channel (n) output is not controlled by FTM, and if (ELS(n+1)B:ELS(n+1)A = 0:0) then the channel (n+1) output is not controlled by FTM.

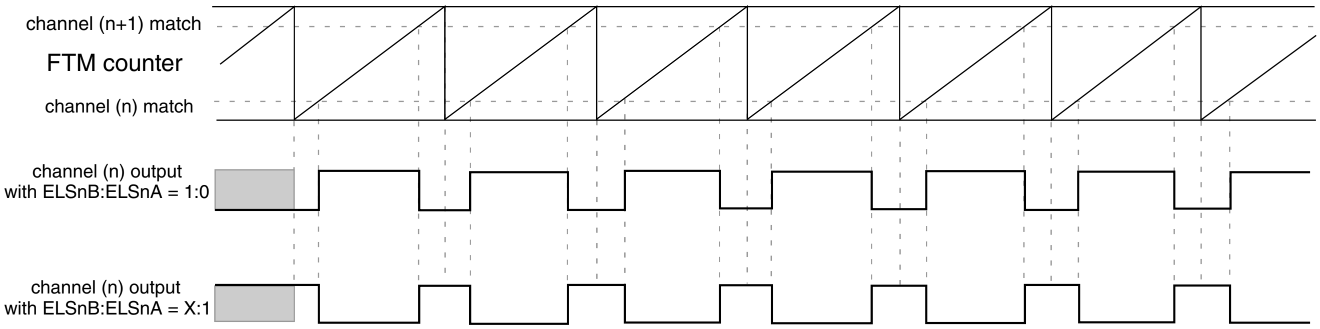


Figure 43-24. Combine mode

The following figures illustrate the PWM signals generation using Combine mode.

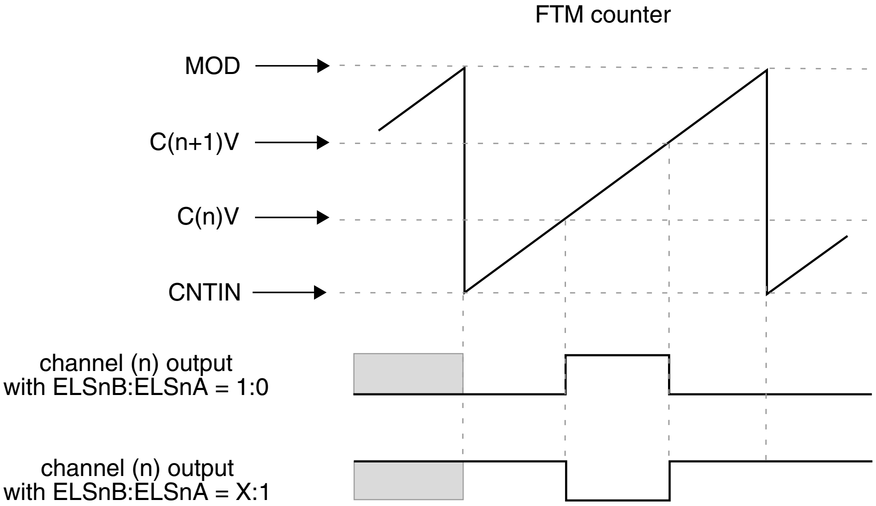


Figure 43-25. Channel (n) output if (CNTIN < C(n)V < MOD) and (CNTIN < C(n+1)V < MOD) and (C(n)V < C(n+1)V)

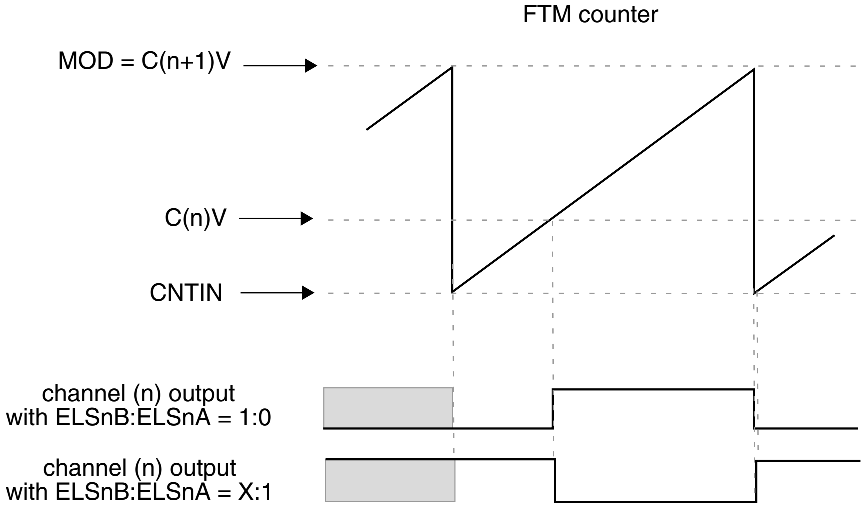


Figure 43-26. Channel (n) output if (CNTIN < C(n)V < MOD) and (C(n+1)V = MOD)

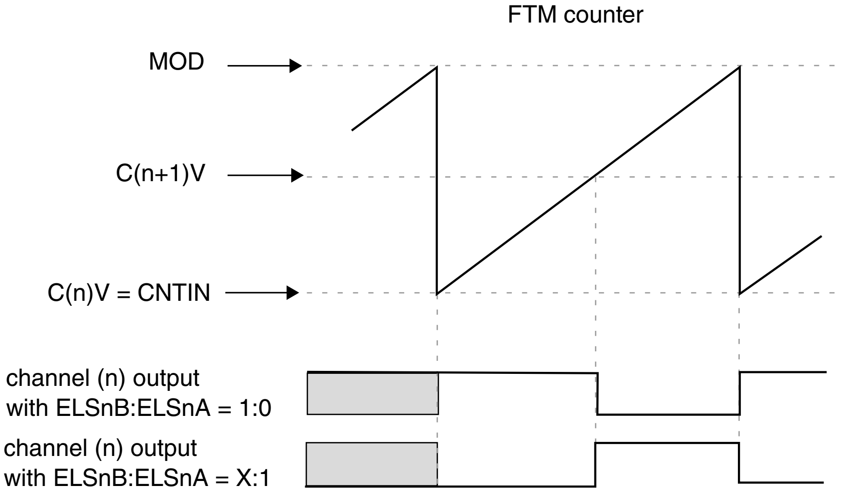


Figure 43-27. Channel (n) output if (C(n)V = CNTIN) and (CNTIN < C(n+1)V < MOD)

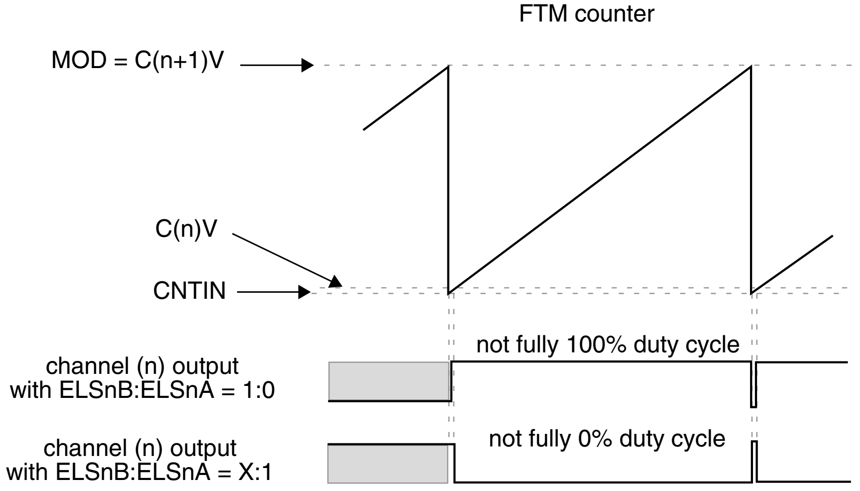


Figure 43-28. Channel (n) output if (CNTIN < C(n)V < MOD) and (C(n)V is Almost Equal to CNTIN) and (C(n+1)V = MOD)

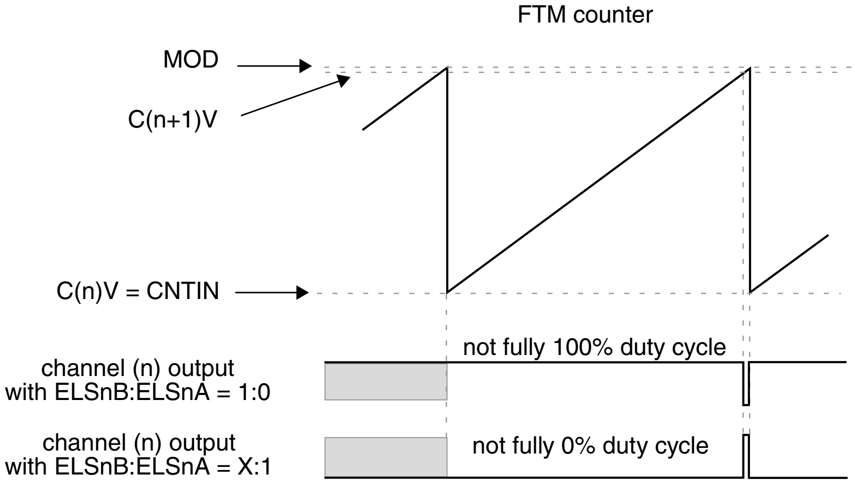


Figure 43-29. Channel (n) output if (C(n)V = CNTIN) and (CNTIN < C(n+1)V < MOD) and (C(n+1)V is Almost Equal to MOD)

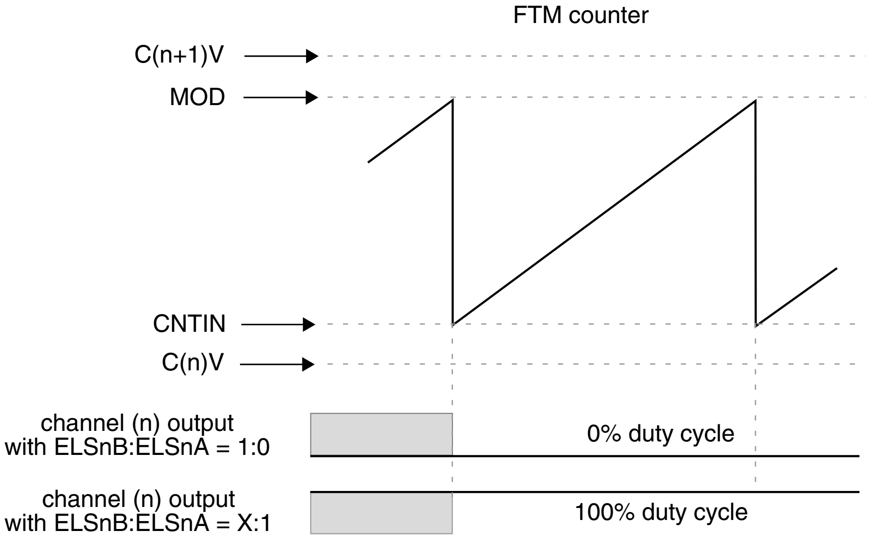


Figure 43-30. Channel (n) output if C(n)V and C(n+1)V are not between CNTIN and MOD

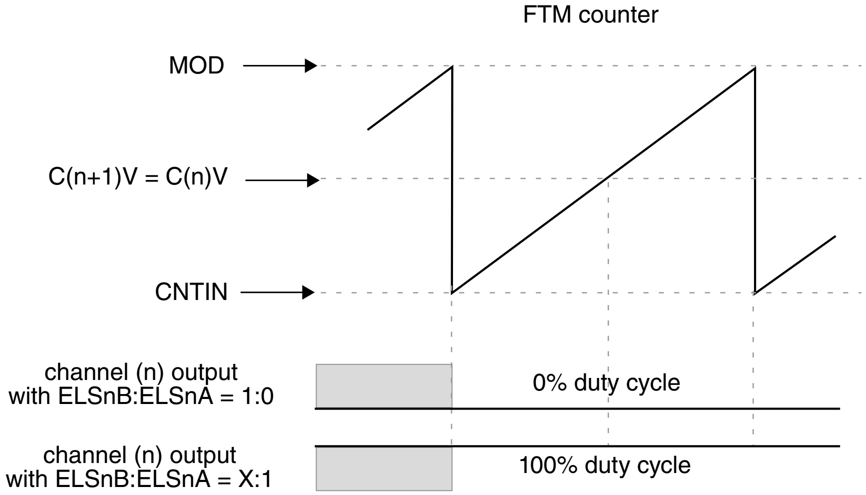

FTM counter
MOD
C(n+1)V = C(n)V
CNTIN
channel (n) output with ELSnB:ELSnA = 1:0
0% duty cycle
channel (n) output with ELSnB:ELSnA = X:1
100% duty cycle

Figure 43-31. Channel (n) output if (CNTIN < C(n)V < MOD) and (CNTIN < C(n+1)V < MOD) and (C(n)V = C(n+1)V)

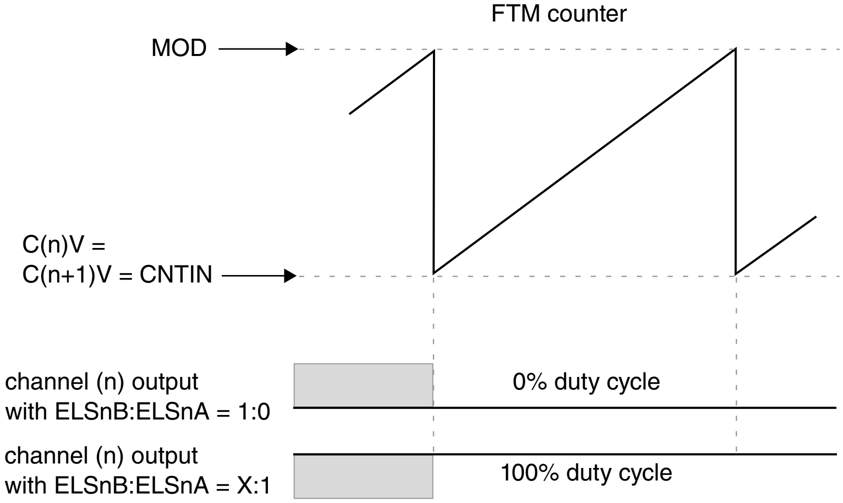

FTM counter
MOD
C(n)V = C(n+1)V = CNTIN
channel (n) output with ELSnB:ELSnA = 1:0
0% duty cycle
channel (n) output with ELSnB:ELSnA = X:1
100% duty cycle

Figure 43-32. Channel (n) output if (C(n)V = C(n+1)V = CNTIN)

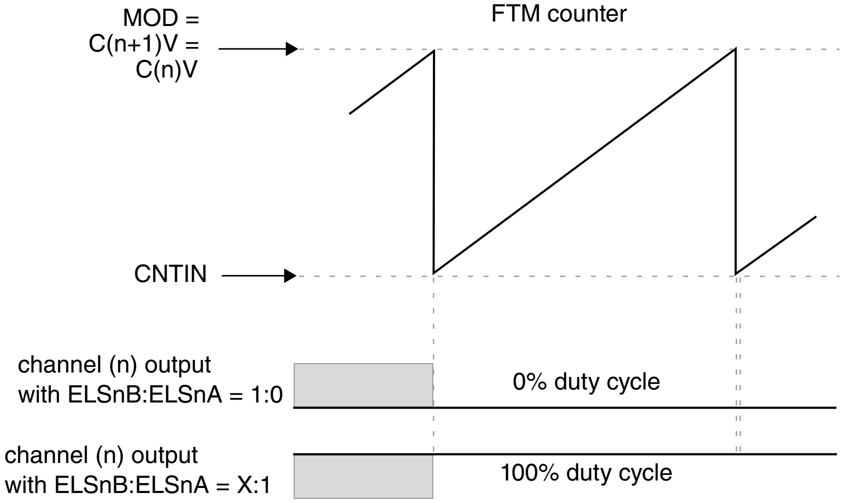

MOD = C(n+1)V = C(n)V
FTM counter
CNTIN
channel (n) output with ELSnB:ELSnA = 1:0
0% duty cycle
channel (n) output with ELSnB:ELSnA = X:1
100% duty cycle

Figure 43-33. Channel (n) output if (C(n)V = C(n+1)V = MOD)

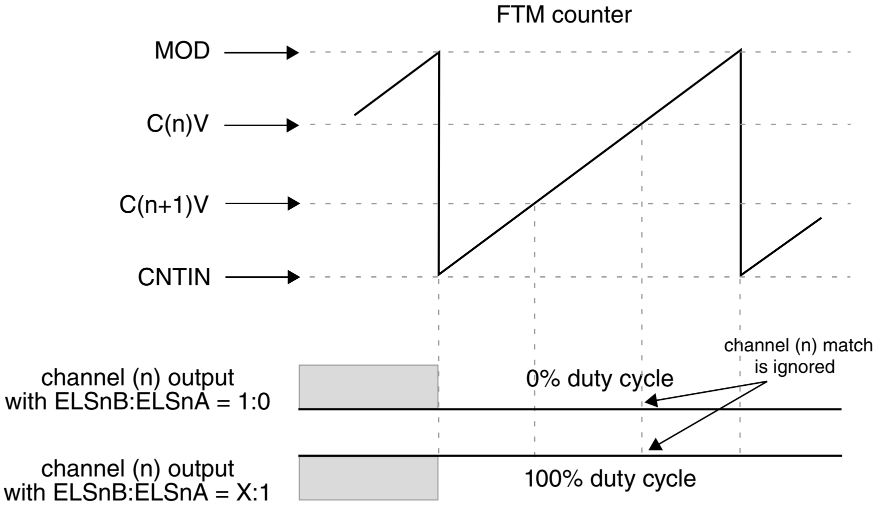

FTM counter
MOD
C(n)V
C(n+1)V
CNTIN
channel (n) match is ignored
channel (n) output with ELSnB:ELSnA = 1:0
0% duty cycle
channel (n) output with ELSnB:ELSnA = X:1
100% duty cycle

Figure 43-34. Channel (n) output if (CNTIN < C(n)V < MOD) and (CNTIN < C(n+1)V < MOD) and (C(n)V > C(n+1)V)

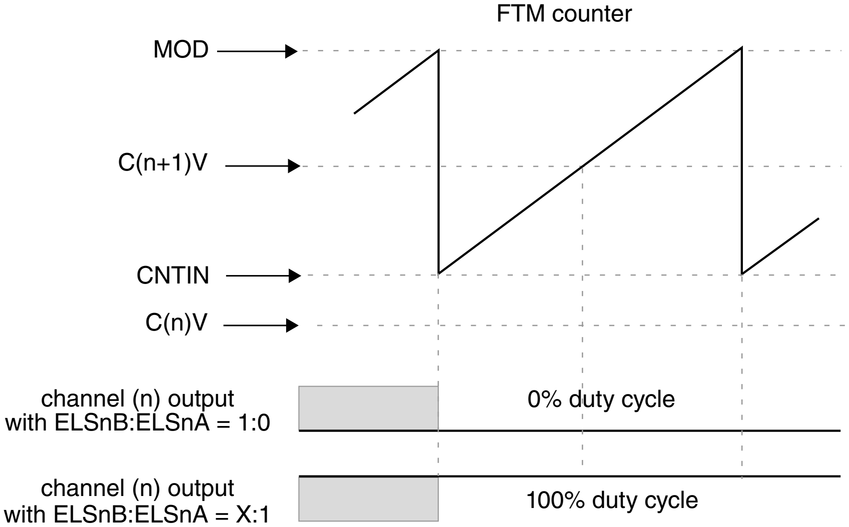

FTM counter
MOD
C(n+1)V
CNTIN
C(n)V
channel (n) output with ELSnB:ELSnA = 1:0
0% duty cycle
channel (n) output with ELSnB:ELSnA = X:1
100% duty cycle

Figure 43-35. Channel (n) output if (C(n)V < CNTIN) and (CNTIN < C(n+1)V < MOD)

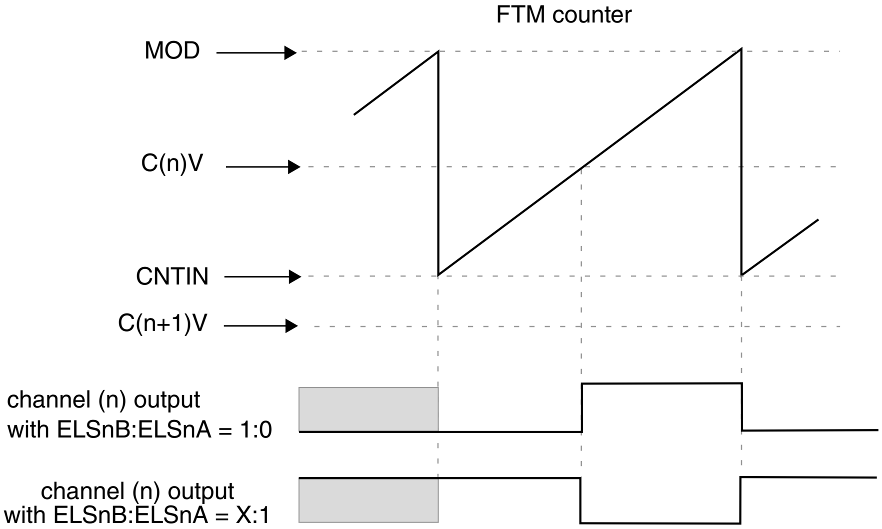


Figure 43-36. Channel (n) output if (C(n+1)V < CNTIN) and (CNTIN < C(n)V < MOD)

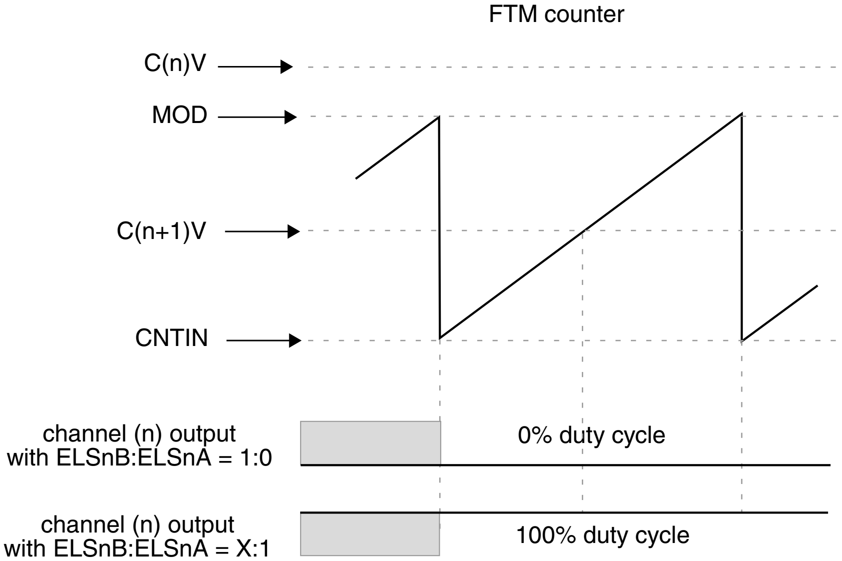


Figure 43-37. Channel (n) output if (C(n)V > MOD) and (CNTIN < C(n+1)V < MOD)

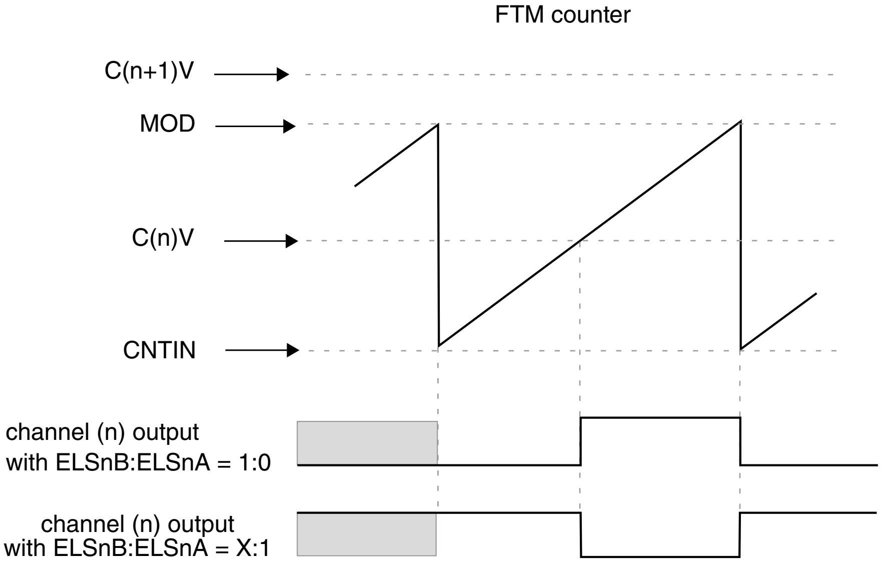


Figure 43-38. Channel (n) output if (C(n+1)V > MOD) and (CNTIN < C(n)V < MOD)

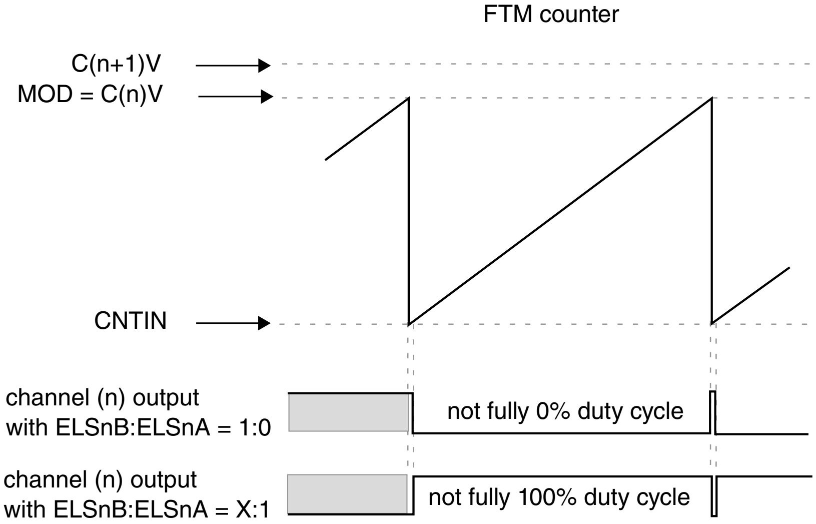


Figure 43-39. Channel (n) output if (C(n+1)V > MOD) and (CNTIN < C(n)V = MOD)

43.4.8.1 Asymmetrical PWM

In Combine mode, the control of the PWM signal first edge, when the channel (n) match occurs, that is, FTM counter = C(n)V, is independent of the control of the PWM signal second edge, when the channel (n+1) match occurs, that is, FTM counter = C(n+1)V. So, Combine mode allows the generation of asymmetrical PWM signals.

43.4.9 Complementary mode

The Complementary mode is selected when:

- QUADEN = 0
- DECAPEN = 0
- COMP = 1

In Complementary mode, the channel (n+1) output is the inverse of the channel (n) output.

So, the channel (n+1) output is the same as the channel (n) output when:

- QUADEN = 0
- DECAPEN = 0
- COMP = 0

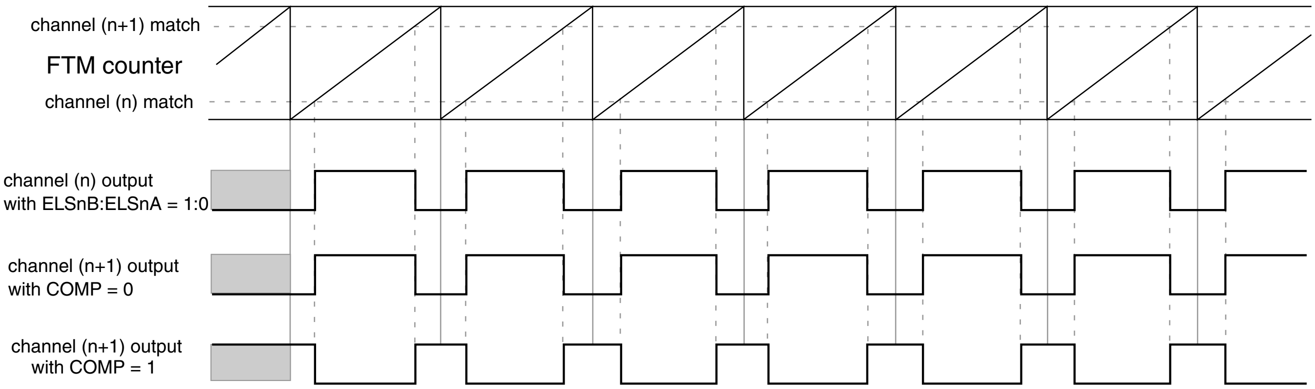


Figure 43-40. Channel (n+1) output in Complementary mode with (ELSnB:ELSnA = 1:0)

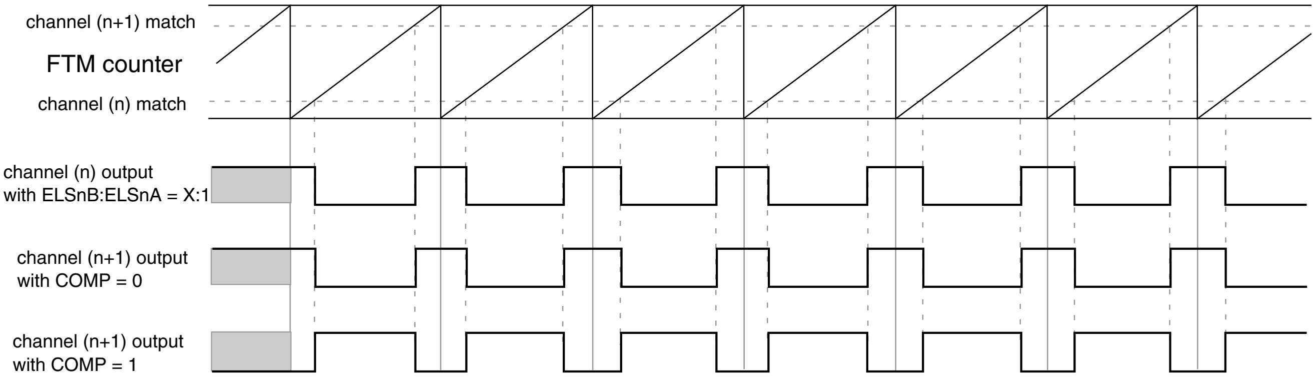


Figure 43-41. Channel (n+1) output in Complementary mode with (ELSnB:ELSnA = X:1)

NOTE

The complementary mode is not available in Output Compare mode.

43.4.10 Registers updated from write buffers

43.4.10.1 CNTIN register update

The following table describes when CNTIN register is updated:

Table 43-5. CNTIN register update

When	Then CNTIN register is updated
CLKS[1:0] = 0:0	When CNTIN register is written, independent of FTMEN bit.
• FTMEN = 0, or • CNTINC = 0	At the next system clock after CNTIN was written.
• FTMEN = 1, • SYNCMODE = 1, and • CNTINC = 1	By the CNTIN register synchronization.

43.4.10.2 MOD register update

The following table describes when MOD register is updated:

Table 43-6. MOD register update

When	Then MOD register is updated
CLKS[1:0] = 0:0	When MOD register is written, independent of FTMEN bit.
• CLKS[1:0] ≠ 0:0, and • FTMEN = 0	According to the CPWMS bit, that is: • If the selected mode is not CPWM then MOD register is updated after MOD register was written and the FTM counter changes from MOD to CNTIN. If the FTM counter is at free-running counter mode then this update occurs when the FTM counter changes from 0xFFFF to 0x0000. • If the selected mode is CPWM then MOD register is updated after MOD register was written and the FTM counter changes from MOD to (MOD – 0x0001).
• CLKS[1:0] ≠ 0:0, and • FTMEN = 1	By the MOD register synchronization.

43.4.10.3 CnV register update

The following table describes when CnV register is updated:

Table 43-7. CnV register update

When	Then CnV register is updated
CLKS[1:0] = 0:0	When CnV register is written, independent of FTMEN bit.
• CLKS[1:0] ≠ 0:0, and • FTMEN = 0	According to the selected mode, that is:

Table continues on the next page...

Table 43-7. CnV register update (continued)

When	Then CnV register is updated
	• If the selected mode is Output Compare, then CnV register is updated on the next FTM counter change, end of the prescaler counting, after CnV register was written. • If the selected mode is EPWM, then CnV register is updated after CnV register was written and the FTM counter changes from MOD to CNTIN. If the FTM counter is at free-running counter mode then this update occurs when the FTM counter changes from 0xFFFF to 0x0000. • If the selected mode is CPWM, then CnV register is updated after CnV register was written and the FTM counter changes from MOD to (MOD – 0x0001).
• CLKS[1:0] ≠ 0:0, and • FTMEN = 1	According to the selected mode, that is: • If the selected mode is output compare then CnV register is updated according to the SYNCEN bit. If (SYNCEN = 0) then CnV register is updated after CnV register was written at the next change of the FTM counter, the end of the prescaler counting. If (SYNCEN = 1) then CnV register is updated by the C(n)V and C(n+1)V register synchronization. • If the selected mode is not output compare and (SYNCEN = 1) then CnV register is updated by the C(n)V and C(n+1)V register synchronization.

43.4.11 PWM synchronization

The PWM synchronization provides an opportunity to update the MOD, CNTIN, CnV, OUTMASK, INVCTRL and SWOCTRL registers with their buffered value and force the FTM counter to the CNTIN register value.

Note

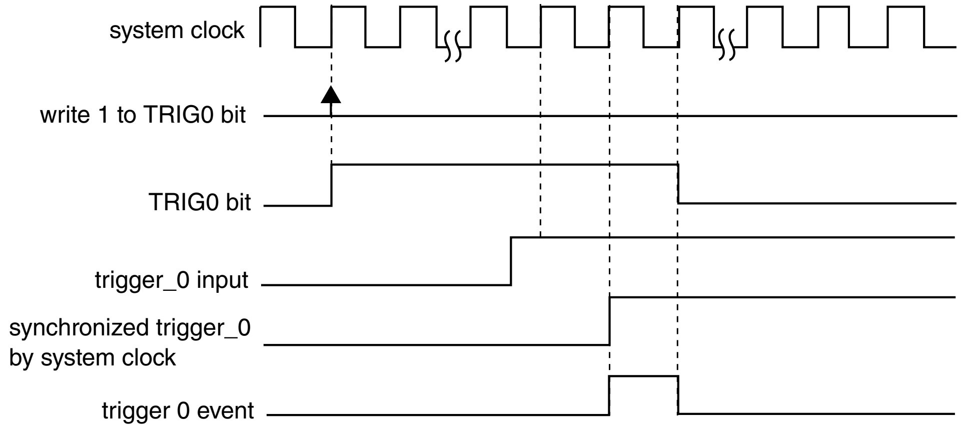
> The legacy PWM synchronization (SYNCMODE = 0) is a subset of the enhanced PWM synchronization (SYNCMODE = 1). Thus, only the enhanced PWM synchronization must be used.

43.4.11.1 Hardware trigger

Three hardware trigger signal inputs of the FTM module are enabled when TRIGn = 1, where n = 0, 1 or 2 corresponding to each one of the input signals, respectively. The hardware trigger input n is synchronized by the system clock. The PWM synchronization with hardware trigger is initiated when a rising edge is detected at the enabled hardware trigger inputs.

If (HWTRIGMODE = 0) then the TRIGn bit is cleared when 0 is written to it or when the trigger n event is detected.

In this case, if two or more hardware triggers are enabled (for example, TRIG0 and TRIG1 = 1) and only trigger 1 event occurs, then only TRIG1 bit is cleared. If a trigger n event occurs together with a write setting TRIGn bit, then the synchronization is initiated, but TRIGn bit remains set due to the write operation.



Note
All hardware trigger inputs have the same behavior.

Figure 43-42. Hardware trigger event with HWTRIGMODE = 0

If HWTRIGMODE = 1, then the TRIGn bit is only cleared when 0 is written to it.

NOTE

The HWTRIGMODE bit must be 1 only with enhanced PWM synchronization (SYNCMODE = 1).

43.4.11.2 Software trigger

A software trigger event occurs when 1 is written to the SYNC[SWSYNC] bit. The SWSYNC bit is cleared when 0 is written to it or when the PWM synchronization, initiated by the software event, is completed.

If another software trigger event occurs (by writing another 1 to the SWSYNC bit) at the same time the PWM synchronization initiated by the previous software trigger event is ending, a new PWM synchronization is started and the SWSYNC bit remains equal to 1.

If SYNCMODE = 0 then the SWSYNC bit is also cleared by FTM according to PWMSYNC and REINIT bits. In this case if (PWMSYNC = 1) or (PWMSYNC = 0 and REINIT = 0) then SWSYNC bit is cleared at the next selected loading point after that the software trigger event occurred; see Boundary cycle and loading points and the following figure. If (PWMSYNC = 0) and (REINIT = 1) then SWSYNC bit is cleared when the software trigger event occurs.

If SYNCMODE = 1 then the SWSYNC bit is also cleared by FTM according to the SWRSTCNT bit. If SWRSTCNT = 0 then SWSYNC bit is cleared at the next selected loading point after that the software trigger event occurred; see the following figure. If SWRSTCNT = 1 then SWSYNC bit is cleared when the software trigger event occurs.

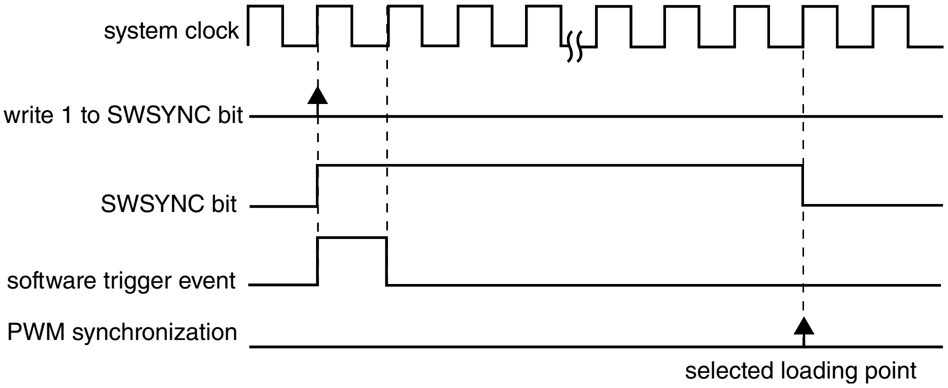


Figure 43-43. Software trigger event

43.4.11.3 Boundary cycle and loading points

The boundary cycle definition is important for the loading points for the registers MOD, CNTIN, and C(n)V.

In Up counting mode, the boundary cycle is defined as when the counter wraps to its initial value (CNTIN). If in Up-down counting mode, then the boundary cycle is defined as when the counter turns from down to up counting and when from up to down counting.

The following figure shows the boundary cycles and the loading points for the registers. In the Up Counting mode, the loading points are enabled if one of CNTMIN or CTMAX bits are 1. In the Up-Down Counting mode, the loading points are selected by CNTMIN and CNTMAX bits, as indicated in the figure. These loading points are safe places for register updates thus allowing a smooth transitions in PWM waveform generation.

For both counting modes, if neither CNTMIN nor CNTMAX are 1, then the boundary cycles are not used as loading points for registers updates. See the register synchronization descriptions in the following sections for details.

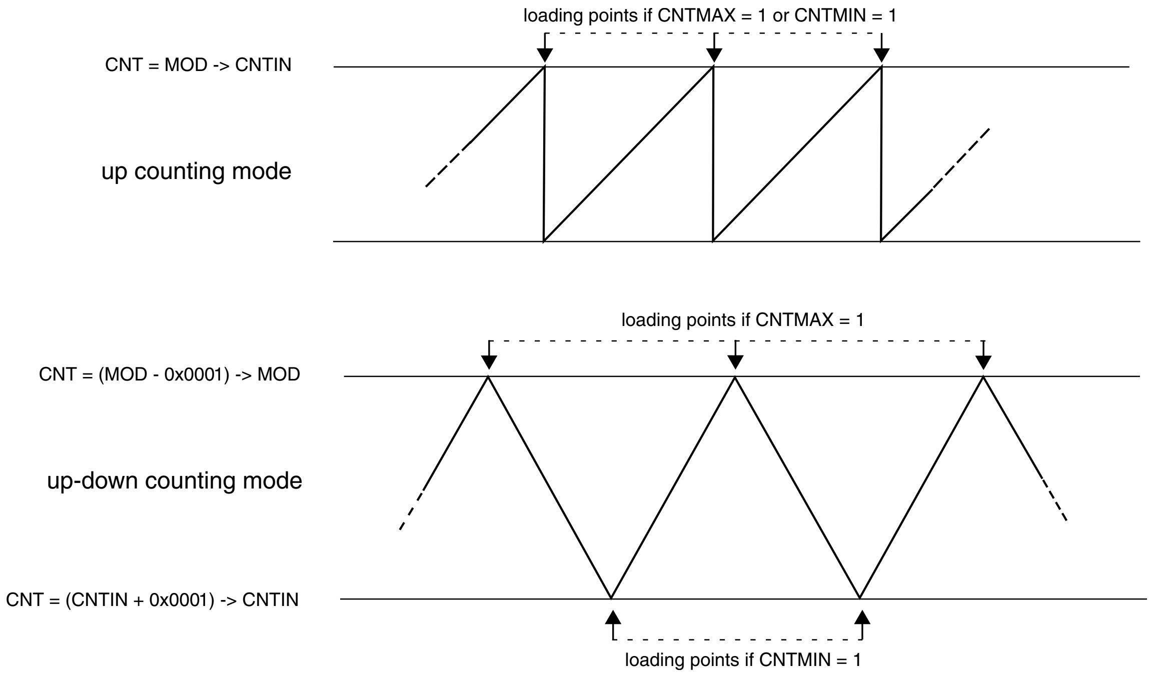

loading points if CNTMAX = 1 or CNTMIN = 1
CNT = MOD -> CNTIN
up counting mode
loading points if CNTMAX = 1
CNT = (MOD - 0x0001) -> MOD
up-down counting mode
CNT = (CNTIN + 0x0001) -> CNTIN
loading points if CNTMIN = 1

Figure 43-44. Boundary cycles and loading points

43.4.11.4 MOD register synchronization

The MOD register synchronization updates the MOD register with its buffer value. This synchronization is enabled if (FTMEN = 1).

The MOD register synchronization can be done by either the enhanced PWM synchronization (SYNCMODE = 1) or the legacy PWM synchronization (SYNCMODE = 0). However, it is expected that the MOD register be synchronized only by the enhanced PWM synchronization.

In the case of enhanced PWM synchronization, the MOD register synchronization depends on SWWRBUF, SWRSTCNT, HWWRBUF, and HWRSTCNT bits according to this flowchart:

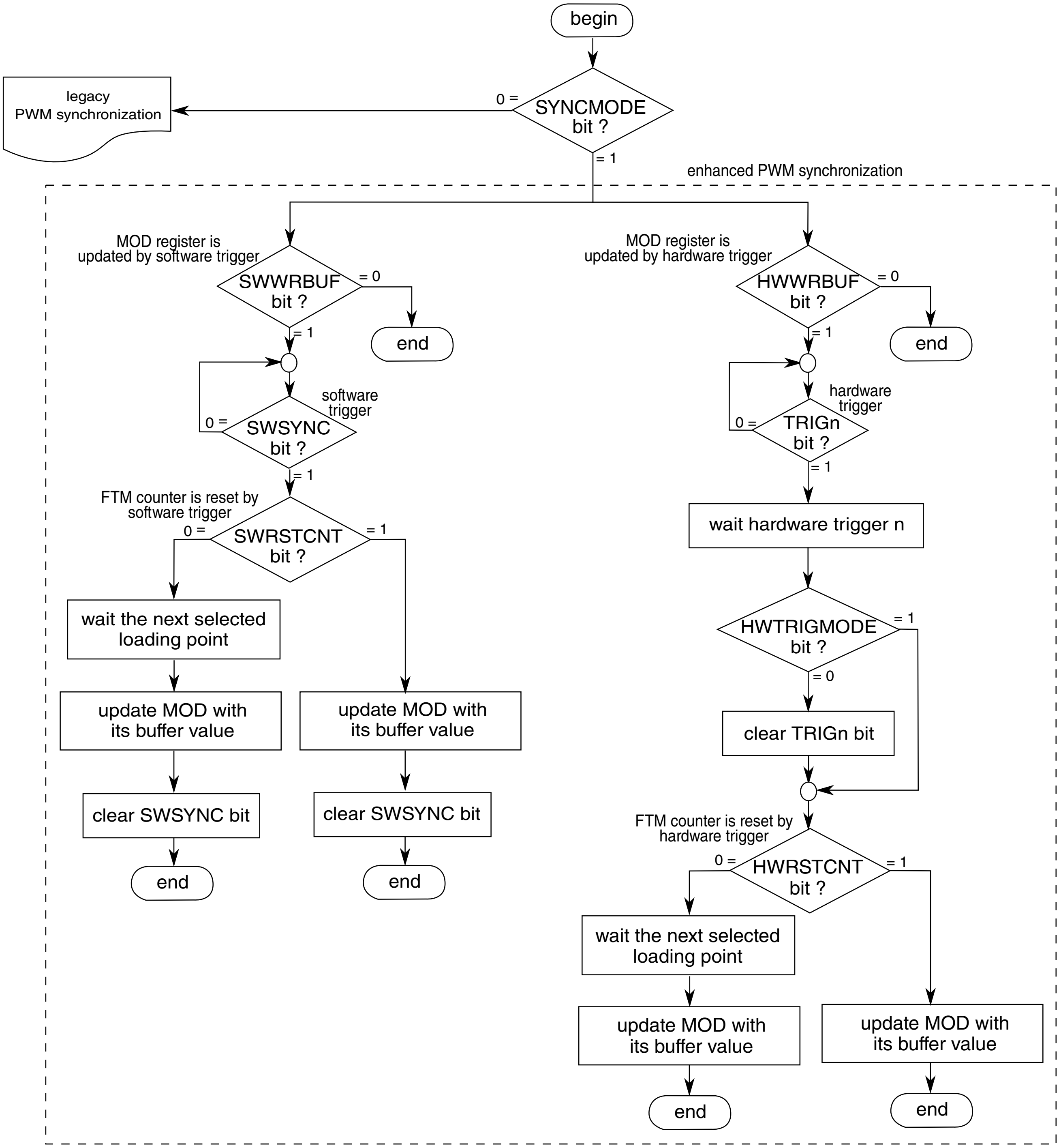

begin
legacy PWM synchronization
0 =
SYNCMODE bit ?
= 1
enhanced PWM synchronization
MOD register is updated by software trigger
SWWRBUF bit ?
= 0
end
= 1
software trigger
0 =
SWSYNC bit ?
= 1
FTM counter is reset by software trigger
0 =
SWRSTCNT bit ?
= 1
wait the next selected loading point
update MOD with its buffer value
clear SWSYNC bit
end
update MOD with its buffer value
clear SWSYNC bit
end
MOD register is updated by hardware trigger
HWWRBUF bit ?
= 0
end
= 1
hardware trigger
0 =
TRIGn bit ?
= 1
wait hardware trigger n
HWTRIGMODE bit ?
= 1
= 0
clear TRIGn bit
FTM counter is reset by hardware trigger
0 =
HWRSTCNT bit ?
= 1
wait the next selected loading point
update MOD with its buffer value
end
update MOD with its buffer value
end

Figure 43-45. MOD register synchronization flowchart

In the case of legacy PWM synchronization, the MOD register synchronization depends on PWMSYNC and REINIT bits according to the following description.

If (SYNCMODE = 0), (PWMSYNC = 0), and (REINIT = 0), then this synchronization is made on the next selected loading point after an enabled trigger event takes place. If the trigger event was a software trigger, then the SWSYNC bit is cleared on the next selected loading point. If the trigger event was a hardware trigger, then the trigger enable bit (TRIGn) is cleared according to Hardware trigger. Examples with software and hardware triggers follow.

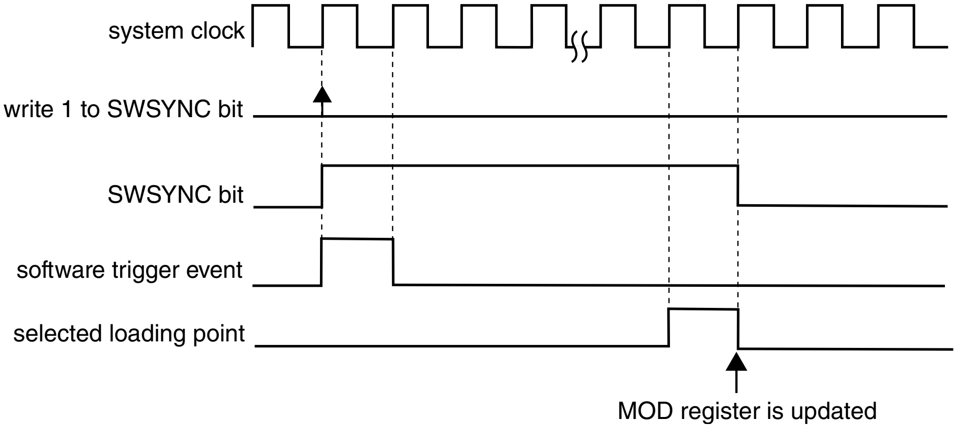


Figure 43-46. MOD synchronization with (SYNCMODE = 0), (PWMSYNC = 0), (REINIT = 0), and software trigger was used

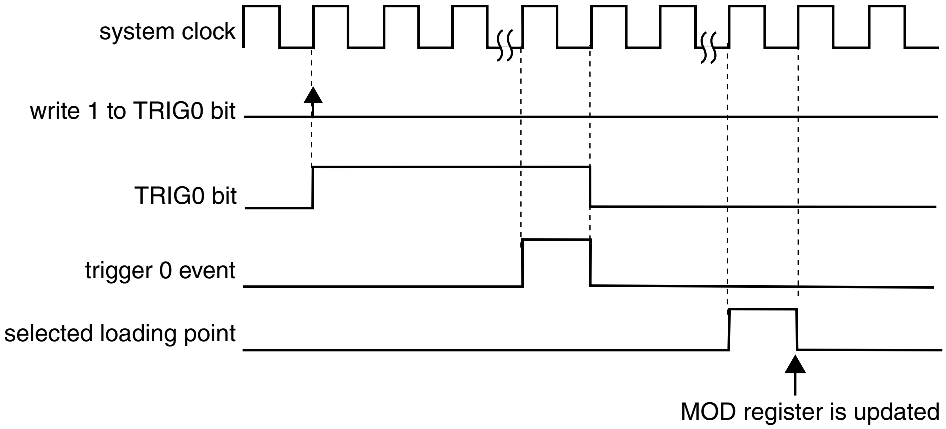


Figure 43-47. MOD synchronization with (SYNCMODE = 0), (HWTRIGMODE = 0), (PWMSYNC = 0), (REINIT = 0), and a hardware trigger was used

If (SYNCMODE = 0), (PWMSYNC = 0), and (REINIT = 1), then this synchronization is made on the next enabled trigger event. If the trigger event was a software trigger, then the SWSYNC bit is cleared according to the following example. If the trigger event was a hardware trigger, then the TRIGn bit is cleared according to Hardware trigger. Examples with software and hardware triggers follow.

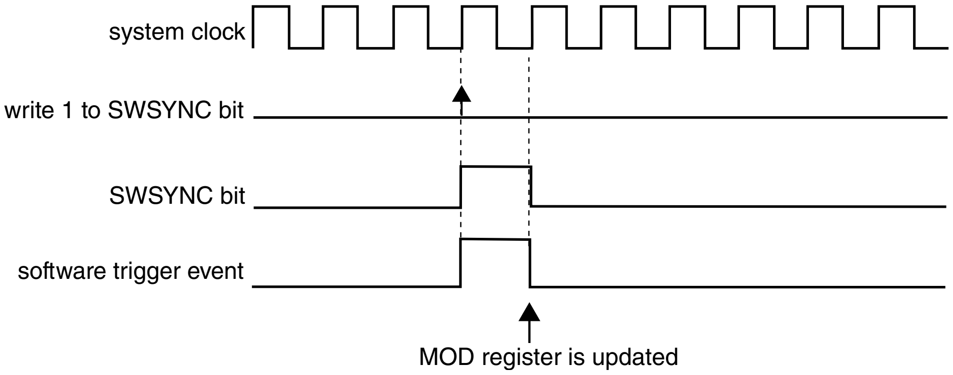


Figure 43-48. MOD synchronization with (SYNCMODE = 0), (PWMSYNC = 0), (REINIT = 1), and software trigger was used

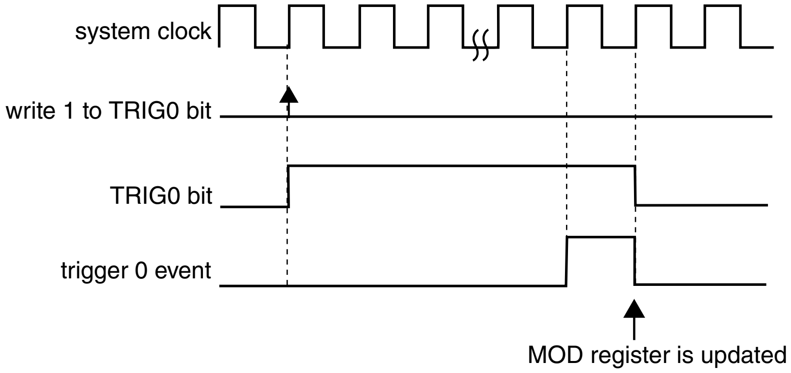


Figure 43-49. MOD synchronization with (SYNCMODE = 0), (HWTRIGMODE = 0), (PWMSYNC = 0), (REINIT = 1), and a hardware trigger was used

If (SYNCMODE = 0) and (PWMSYNC = 1), then this synchronization is made on the next selected loading point after the software trigger event takes place. The SWSYNC bit is cleared on the next selected loading point:

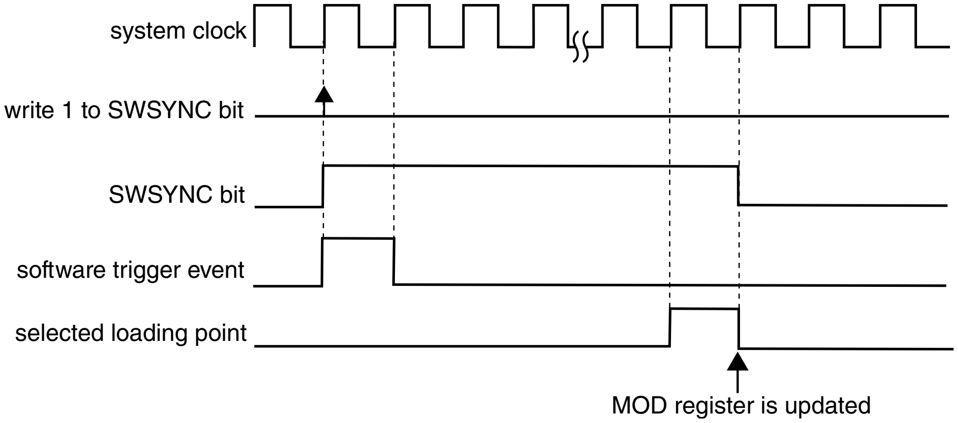


Figure 43-50. MOD synchronization with (SYNCMODE = 0) and (PWMSYNC = 1)

43.4.11.5 CNTIN register synchronization

The CNTIN register synchronization updates the CNTIN register with its buffer value.

This synchronization is enabled if (FTMEN = 1), (SYNCMODE = 1), and (CNTINC = 1). The CNTIN register synchronization can be done only by the enhanced PWM synchronization (SYNCMODE = 1). The synchronization mechanism is the same as the MOD register synchronization done by the enhanced PWM synchronization; see MOD register synchronization.

43.4.11.6 C(n)V and C(n+1)V register synchronization

The C(n)V and C(n+1)V registers synchronization updates the C(n)V and C(n+1)V registers with their buffer values.

This synchronization is enabled if (FTMEN = 1) and (SYNCEN = 1). The synchronization mechanism is the same as the MOD register synchronization. However, it is expected that the C(n)V and C(n+1)V registers be synchronized only by the enhanced PWM synchronization (SYNCMODE = 1).

43.4.11.7 OUTMASK register synchronization

The OUTMASK register synchronization updates the OUTMASK register with its buffer value.

The OUTMASK register can be updated at each rising edge of system clock (SYNCHOM = 0), by the enhanced PWM synchronization (SYNCHOM = 1 and SYNCMODE = 1) or by the legacy PWM synchronization (SYNCHOM = 1 and SYNCMODE = 0). However, it is expected that the OUTMASK register be synchronized only by the enhanced PWM synchronization.

In the case of enhanced PWM synchronization, the OUTMASK register synchronization depends on SWOM and HWOM bits. See the following flowchart:

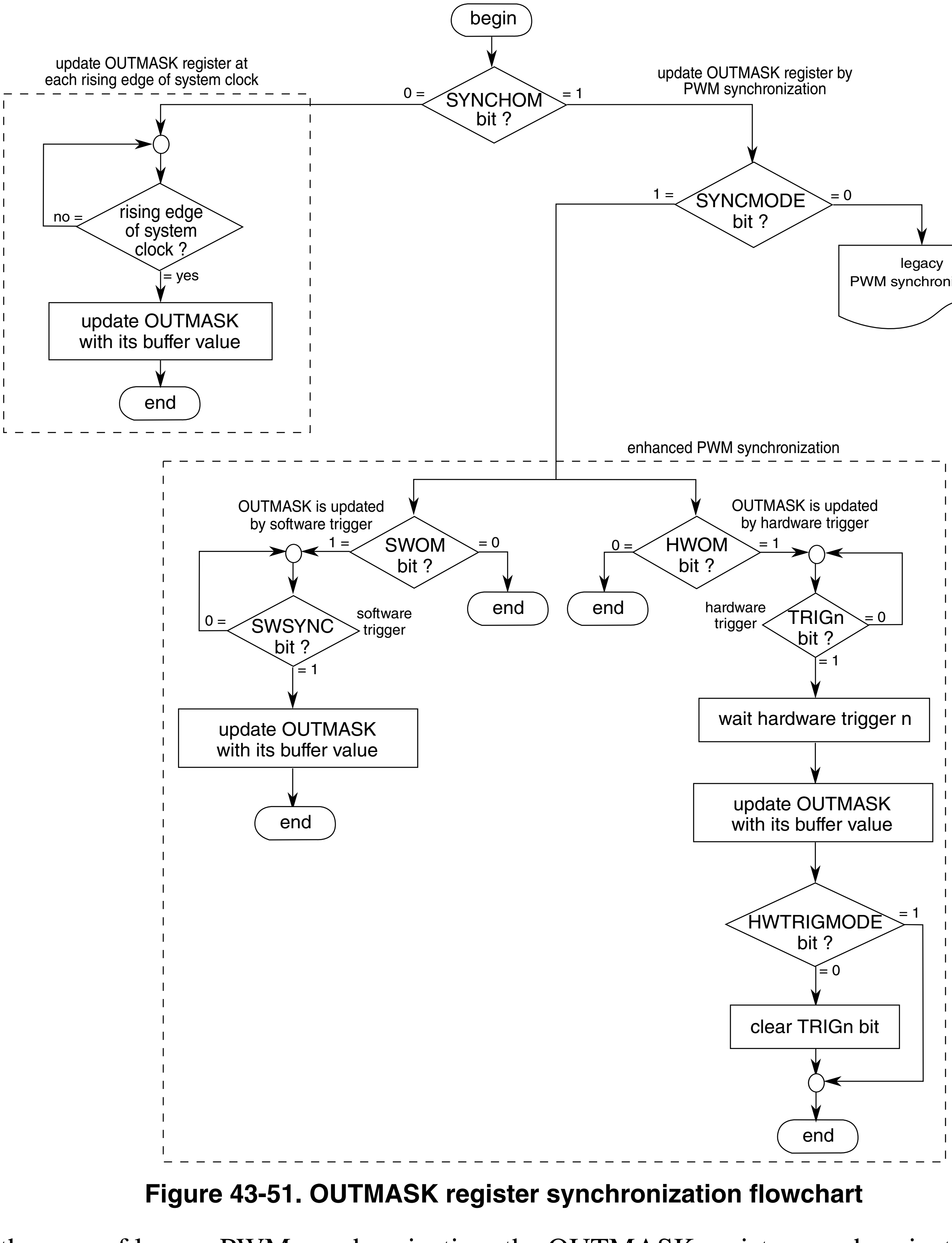


Figure 43-51. OUTMASK register synchronization flowchart

In the case of legacy PWM synchronization, the OUTMASK register synchronization depends on PWMSYNC bit according to the following description.

If (SYNCMODE = 0), (SYNCHOM = 1), and (PWMSYNC = 0), then this synchronization is done on the next enabled trigger event. If the trigger event was a software trigger, then the SWSYNC bit is cleared on the next selected loading point. If the trigger event was a hardware trigger, then the TRIGn bit is cleared according to Hardware trigger. Examples with software and hardware triggers follow.

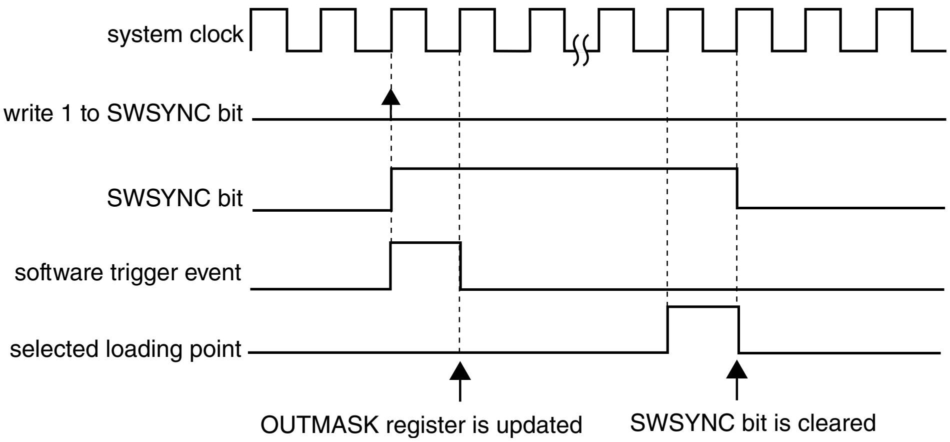


Figure 43-52. OUTMASK synchronization with (SYNCMODE = 0), (SYNCHOM = 1), (PWMSYNC = 0) and software trigger was used

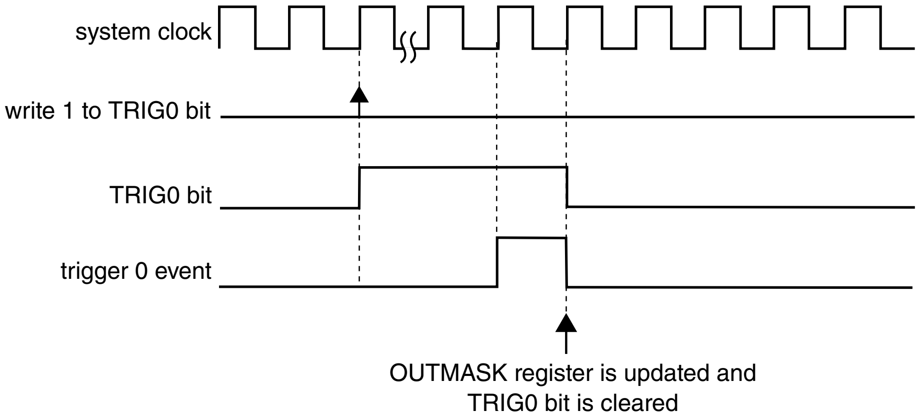


Figure 43-53. OUTMASK synchronization with (SYNCMODE = 0), (HWTRIGMODE = 0), (SYNCHOM = 1), (PWMSYNC = 0), and a hardware trigger was used

If (SYNCMODE = 0), (SYNCHOM = 1), and (PWMSYNC = 1), then this synchronization is made on the next enabled hardware trigger. The TRIGn bit is cleared according to Hardware trigger. An example with a hardware trigger follows.

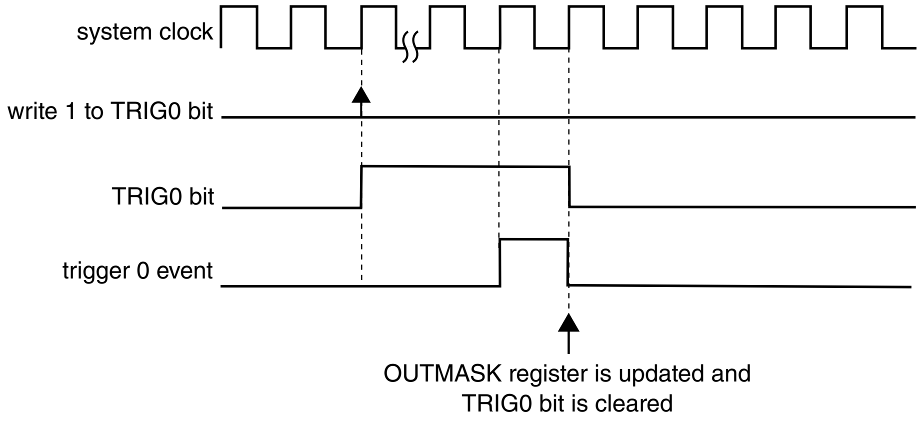


Figure 43-54. OUTMASK synchronization with (SYNCMODE = 0), (HWTRIGMODE = 0), (SYNCHOM = 1), (PWMSYNC = 1), and a hardware trigger was used

43.4.11.8 INVCTRL register synchronization

The INVCTRL register synchronization updates the INVCTRL register with its buffer value.

The INVCTRL register can be updated at each rising edge of system clock (INVC = 0) or by the enhanced PWM synchronization (INVC = 1 and SYNCMODE = 1) according to the following flowchart.

In the case of enhanced PWM synchronization, the INVCTRL register synchronization depends on SWINVC and HWINVC bits.

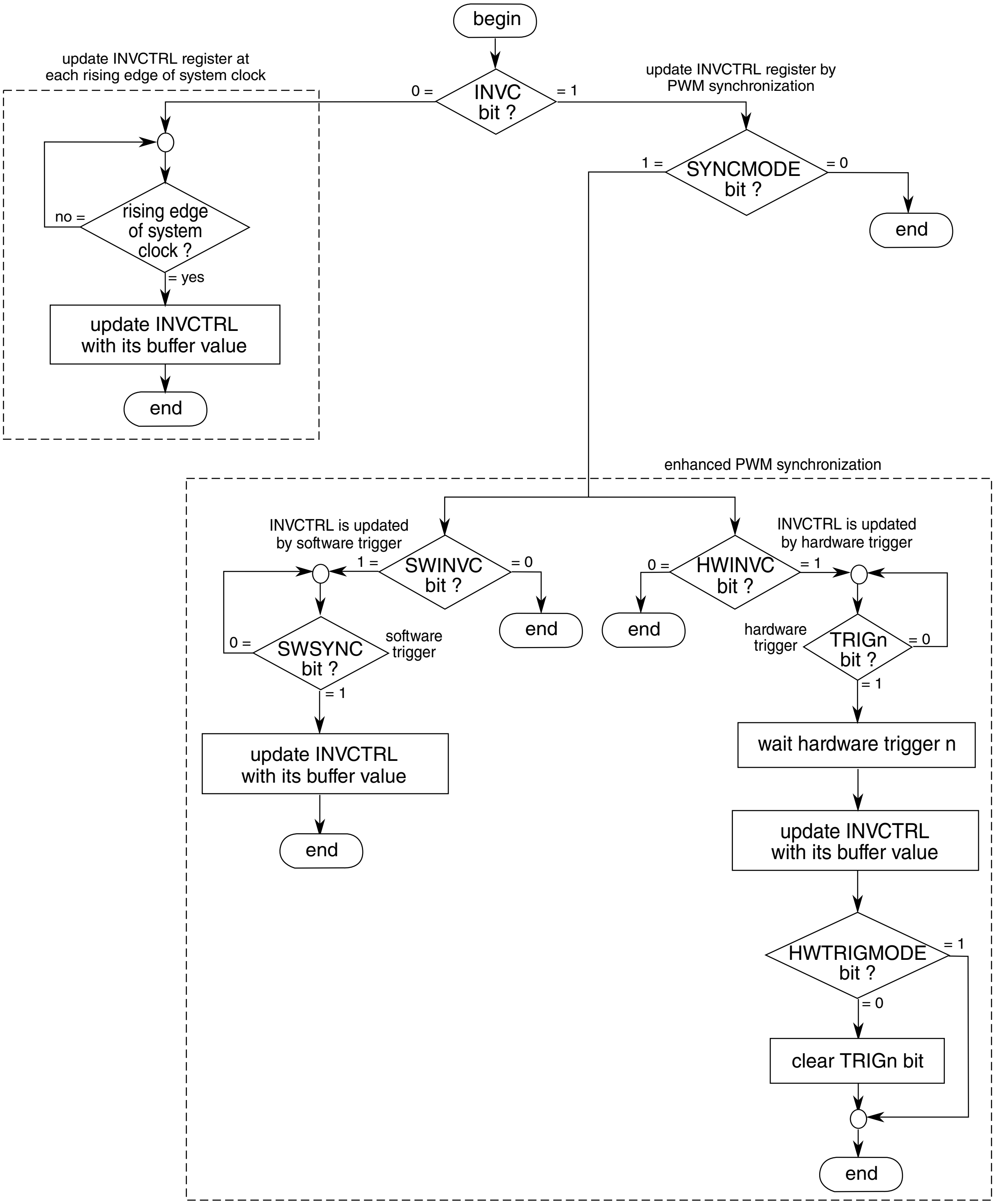

begin
update INVCTRL register at each rising edge of system clock
0 =
INVC bit ?
= 1
update INVCTRL register by PWM synchronization
1 =
SYNCMODE bit ?
= 0
end
no =
rising edge of system clock ?
= yes
update INVCTRL with its buffer value
end
enhanced PWM synchronization
INVCTRL is updated by software trigger
1 =
SWINVC bit ?
= 0
end
0 =
SWSYNC bit ?
software trigger
= 1
update INVCTRL with its buffer value
end
INVCTRL is updated by hardware trigger
0 =
HWINVC bit ?
= 1
end
hardware trigger
TRIGn bit ?
= 0
= 1
wait hardware trigger n
update INVCTRL with its buffer value
HWTRIGMODE bit ?
= 1
= 0
clear TRIGn bit
end

Figure 43-55. INVCTRL register synchronization flowchart

43.4.11.9 SWOCTRL register synchronization

The SWOCTRL register synchronization updates the SWOCTRL register with its buffer value.

The SWOCTRL register can be updated at each rising edge of system clock (SWOC = 0) or by the enhanced PWM synchronization (SWOC = 1 and SYNCMODE = 1) according to the following flowchart.

In the case of enhanced PWM synchronization, the SWOCTRL register synchronization depends on SWSOC and HWSOC bits.

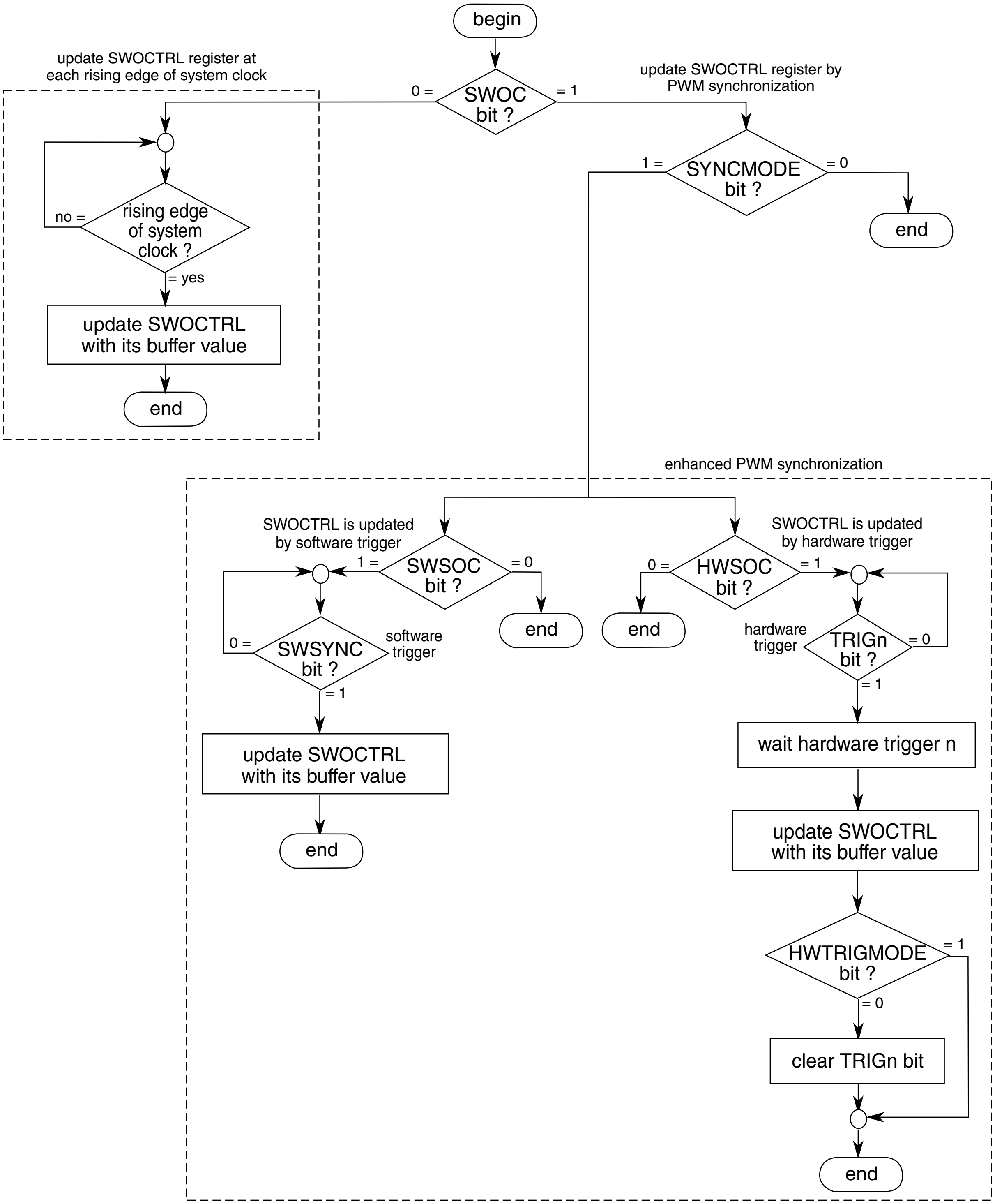

begin
update SWOCTRL register at each rising edge of system clock
0 =
SWOC bit ?
= 1
update SWOCTRL register by PWM synchronization
1 =
SYNCMODE bit ?
= 0
end
no =
rising edge of system clock ?
= yes
update SWOCTRL with its buffer value
end
enhanced PWM synchronization
SWOCTRL is updated by software trigger
1 =
SWSOC bit ?
= 0
end
0 =
SWSYNC bit ?
software trigger
= 1
update SWOCTRL with its buffer value
end
0 =
HWSOC bit ?
= 1
end
SWOCTRL is updated by hardware trigger
hardware trigger
TRIGn bit ?
= 0
= 1
wait hardware trigger n
update SWOCTRL with its buffer value
HWTRIGMODE bit ?
= 1
= 0
clear TRIGn bit
end

Figure 43-56. SWOCTRL register synchronization flowchart

43.4.11.10 FTM counter synchronization

The FTM counter synchronization is a mechanism that allows the FTM to restart the PWM generation at a certain point in the PWM period. The channels outputs are forced to their initial value, except for channels in Output Compare mode, and the FTM counter is forced to its initial counting value defined by CNTIN register.

The following figure shows the FTM counter synchronization. Note that after the synchronization event occurs, the channel (n) is set to its initial value and the channel (n +1) is not set to its initial value due to a specific timing of this figure in which the deadtime insertion prevents this channel output from transitioning to 1. If no deadtime insertion is selected, then the channel (n+1) transitions to logical value 1 immediately after the synchronization event occurs.

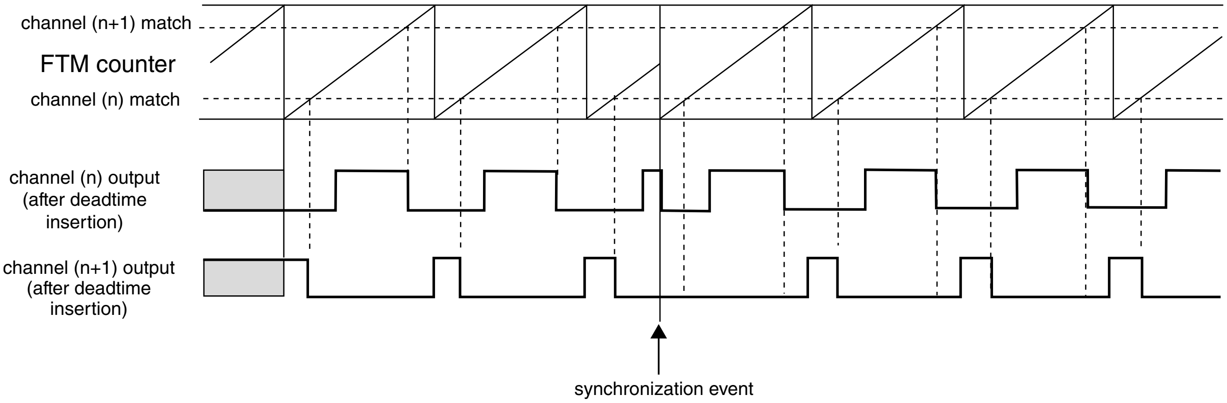

channel (n+1) match
FTM counter
channel (n) match
channel (n) output (after deadtime insertion)
channel (n+1) output (after deadtime insertion)
synchronization event

Figure 43-57. FTM counter synchronization

The FTM counter synchronization can be done by either the enhanced PWM synchronization (SYNCMODE = 1) or the legacy PWM synchronization (SYNCMODE = 0). However, the FTM counter must be synchronized only by the enhanced PWM synchronization.

In the case of enhanced PWM synchronization, the FTM counter synchronization depends on SWRSTCNT and HWRSTCNT bits according to the following flowchart.

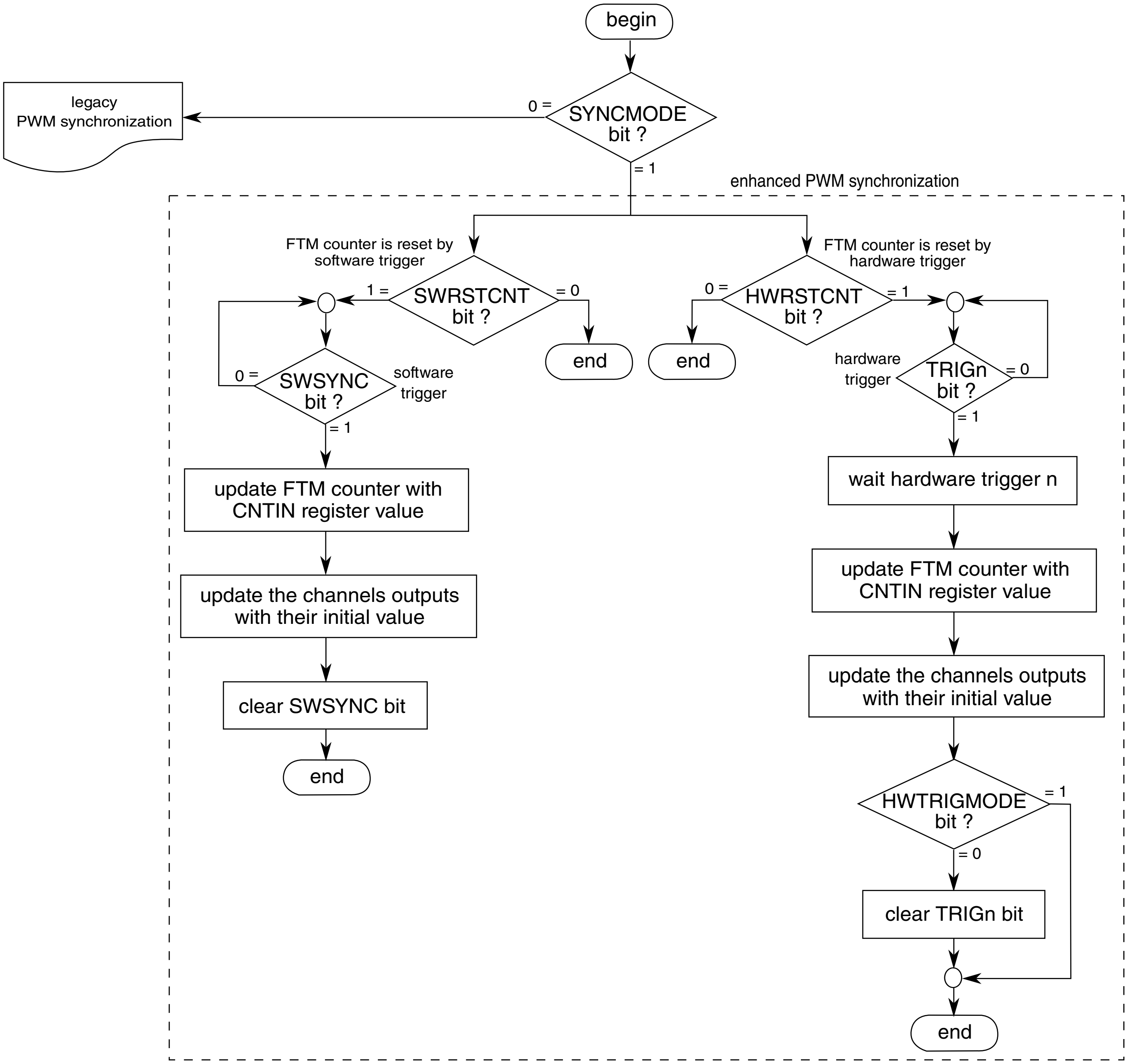

begin
SYNCMODE bit ?
0 =
legacy PWM synchronization
= 1
enhanced PWM synchronization
FTM counter is reset by software trigger
SWRSTCNT bit ?
1 =
= 0
end
SWSYNC bit ?
0 =
software trigger
= 1
update FTM counter with CNTIN register value
update the channels outputs with their initial value
clear SWSYNC bit
end
FTM counter is reset by hardware trigger
HWRSTCNT bit ?
0 =
= 1
end
hardware trigger
TRIGn bit ?
= 0
= 1
wait hardware trigger n
update FTM counter with CNTIN register value
update the channels outputs with their initial value
HWTRIGMODE bit ?
= 1
= 0
clear TRIGn bit
end

Figure 43-58. FTM counter synchronization flowchart

In the case of legacy PWM synchronization, the FTM counter synchronization depends on REINIT and PWMSYNC bits according to the following description.

If (SYNCMODE = 0), (REINIT = 1), and (PWMSYNC = 0) then this synchronization is made on the next enabled trigger event. If the trigger event was a software trigger then the SWSYNC bit is cleared according to the following example. If the trigger event was a hardware trigger then the TRIGn bit is cleared according to Hardware trigger. Examples with software and hardware triggers follow.

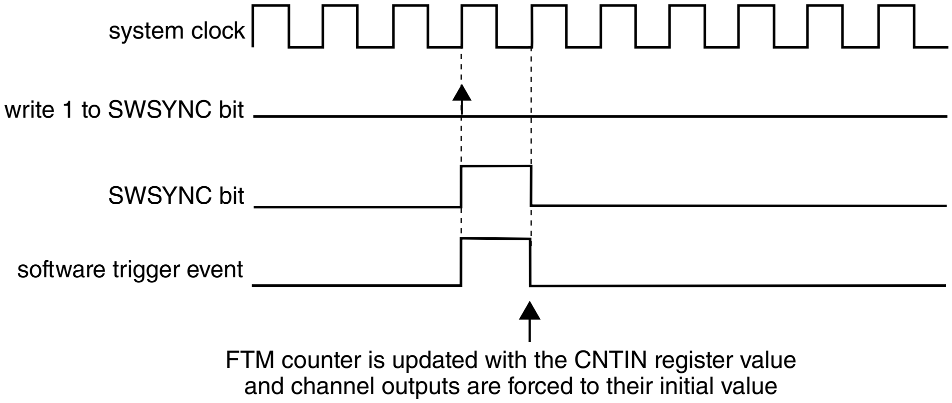

system clock
write 1 to SWSYNC bit
SWSYNC bit
software trigger event
FTM counter is updated with the CNTIN register value
and channel outputs are forced to their initial value

Figure 43-59. FTM counter synchronization with (SYNCMODE = 0), (REINIT = 1), (PWMSYNC = 0), and software trigger was used

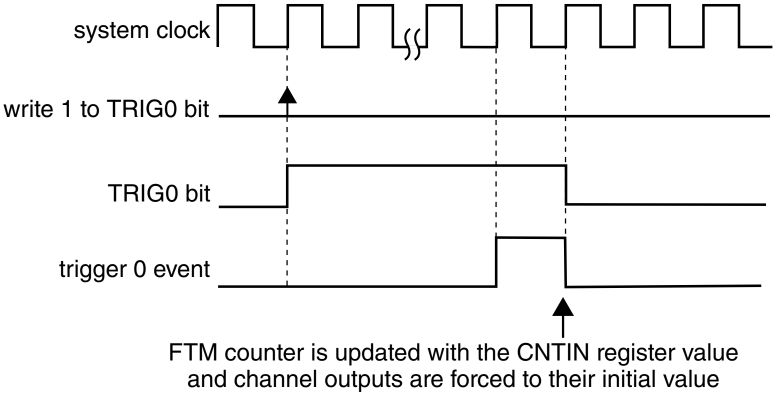

system clock
write 1 to TRIG0 bit
TRIG0 bit
trigger 0 event
FTM counter is updated with the CNTIN register value
and channel outputs are forced to their initial value

Figure 43-60. FTM counter synchronization with (SYNCMODE = 0), (HWTRIGMODE = 0), (REINIT = 1), (PWMSYNC = 0), and a hardware trigger was used

If (SYNCMODE = 0), (REINIT = 1), and (PWMSYNC = 1) then this synchronization is made on the next enabled hardware trigger. The TRIGn bit is cleared according to Hardware trigger.

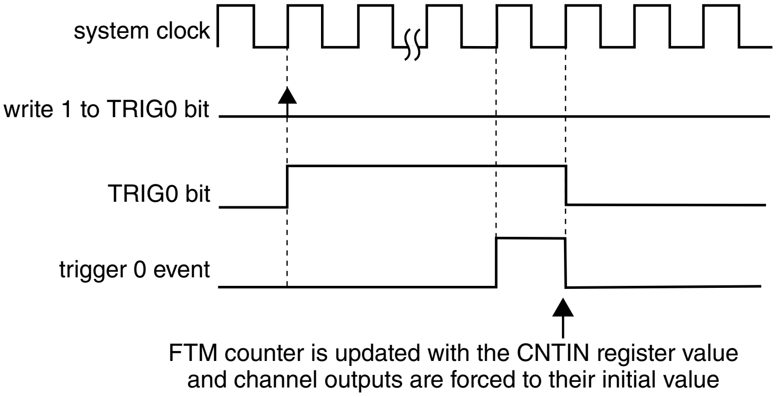

system clock
write 1 to TRIG0 bit
TRIG0 bit
trigger 0 event
FTM counter is updated with the CNTIN register value
and channel outputs are forced to their initial value

Figure 43-61. FTM counter synchronization with (SYNCMODE = 0), (HWTRIGMODE = 0), (REINIT = 1), (PWMSYNC = 1), and a hardware trigger was used

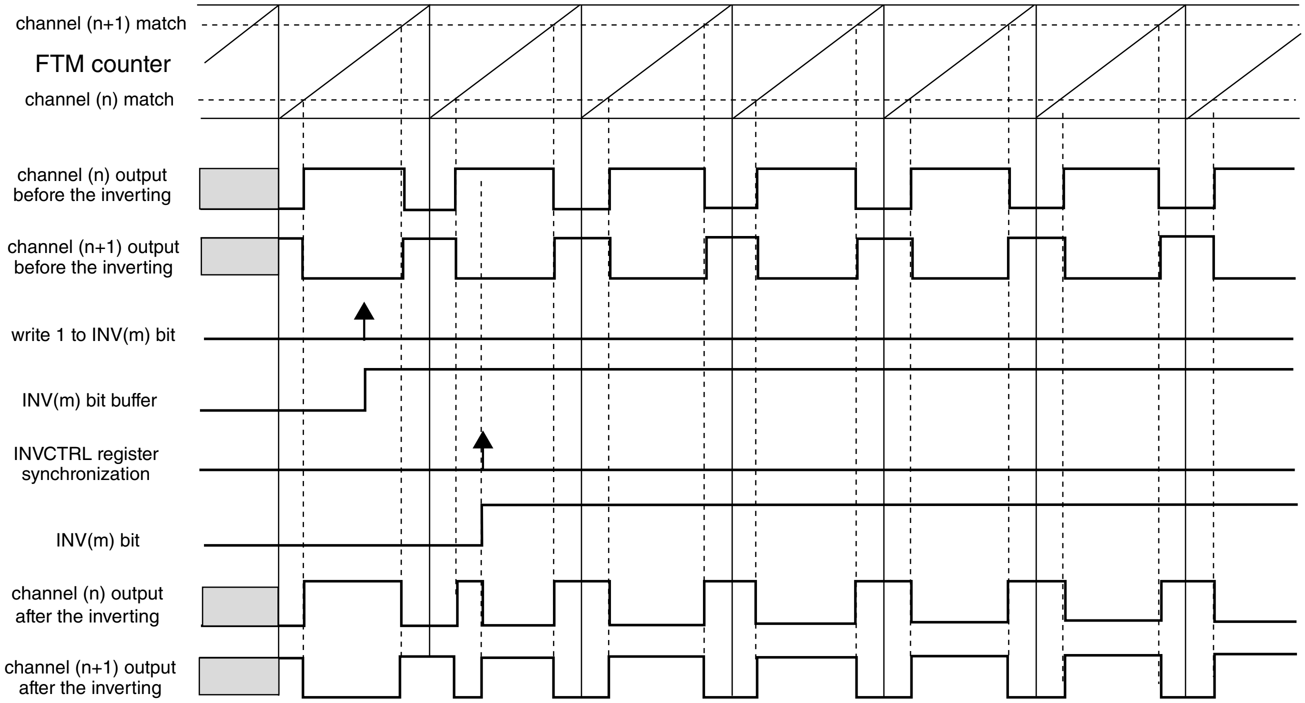
43.4.12 Inverting

The invert functionality swaps the signals between channel (n) and channel (n+1) outputs. The inverting operation is selected when:

- QUADEN = 0
- DECAPEN = 0
- COMP = 1, and
- INVm = 1 (where m represents a channel pair)

The INVm bit in INVCTRL register is updated with its buffer value according to INVCTRL register synchronization

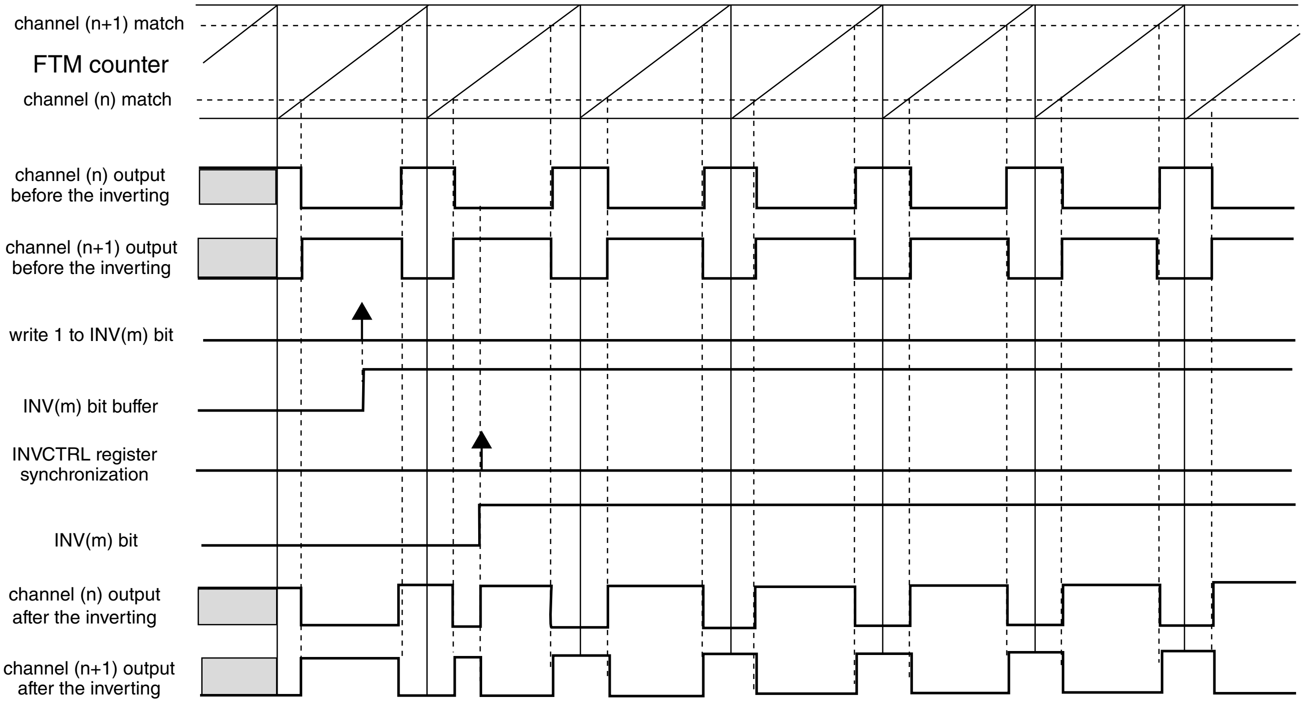
In High-True (ELSnB:ELSnA = 1:0) Combine mode, the channel (n) output is forced low at the beginning of the period (FTM counter = CNTIN), forced high at the channel (n) match and forced low at the channel (n+1) match. If the inverting is selected, the channel (n) output behavior is changed to force high at the beginning of the PWM period, force low at the channel (n) match and force high at the channel (n+1) match. See the following figure.


channel (n+1) match
FTM counter
channel (n) match
channel (n) output before the inverting
channel (n+1) output before the inverting
write 1 to INV(m) bit
INV(m) bit buffer
INVCTRL register synchronization
INV(m) bit
channel (n) output after the inverting
channel (n+1) output after the inverting

NOTE
INV(m) bit selects the inverting to the pair channels (n) and (n+1).

Figure 43-62. Channels (n) and (n+1) outputs after the inverting in High-True (ELSnB:ELSnA = 1:0) Combine mode

Note that the ELSnB:ELSnA bits value should be considered because they define the active state of the channels outputs. In Low-True (ELSnB:ELSnA = X:1) Combine mode, the channel (n) output is forced high at the beginning of the period, forced low at the channel (n) match and forced high at the channel (n+1) match. When inverting is selected, the channels (n) and (n+1) present waveforms as shown in the following figure.


channel (n+1) match
FTM counter
channel (n) match
channel (n) output before the inverting
channel (n+1) output before the inverting
write 1 to INV(m) bit
INV(m) bit buffer
INVCTRL register synchronization
INV(m) bit
channel (n) output after the inverting
channel (n+1) output after the inverting

NOTE
INV(m) bit selects the inverting to the pair channels (n) and (n+1).

Figure 43-63. Channels (n) and (n+1) outputs after the inverting in Low-True (ELSnB:ELSnA = X:1) Combine mode

Note

The inverting feature is not available in Output Compare mode.

43.4.13 Software output control

The software output control forces the channel output according to software defined values at a specific time in the PWM generation.

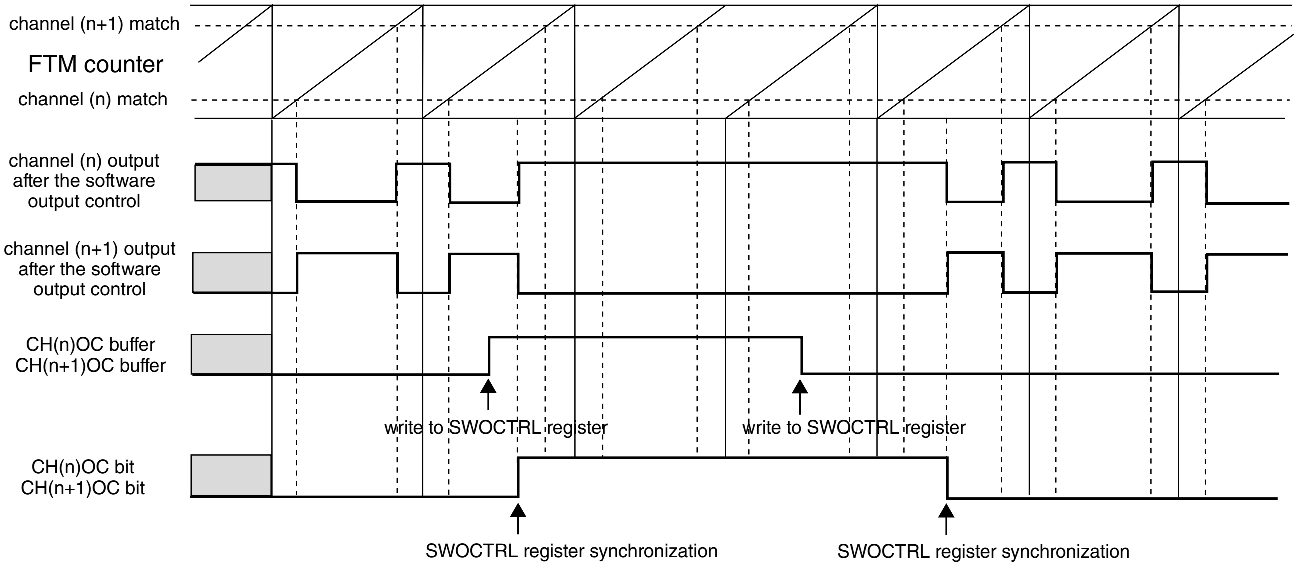
The software output control is selected when:

- QUADEN = 0
- DECAPEN = 0, and
- CHnOC = 1

The CHnOC bit enables the software output control for a specific channel output and the CHnOCV selects the value that is forced to this channel output.

Both CHnOC and CHnOCV bits in SWOCTRL register are buffered and updated with their buffer value according to SWOCTRL register synchronization.

The following figure shows the channels (n) and (n+1) outputs signals when the software output control is used. In this case the channels (n) and (n+1) are set to Combine and Complementary mode.


channel (n+1) match
FTM counter
channel (n) match
channel (n) output after the software output control
channel (n+1) output after the software output control
CH(n)OC buffer
CH(n+1)OC buffer
write to SWOCTRL register
write to SWOCTRL register
CH(n)OC bit
CH(n+1)OC bit
SWOCTRL register synchronization
SWOCTRL register synchronization

NOTE
CH(n)OCV = 1 and CH(n+1)OCV = 0.

Figure 43-64. Example of software output control in Combine and Complementary mode

Software output control forces the following values on channels (n) and (n+1) when the COMP bit is zero.

Table 43-8. Software ouput control behavior when (COMP = 0)

CH(n)OC	CH(n+1)OC	CH(n)OCV	CH(n+1)OCV	Channel (n) Output	Channel (n+1) Output
0	0	X	X	is not modified by SWOC	is not modified by SWOC
1	1	0	0	is forced to zero	is forced to zero
1	1	0	1	is forced to zero	is forced to one
1	1	1	0	is forced to one	is forced to zero
1	1	1	1	is forced to one	is forced to one

Software output control forces the following values on channels (n) and (n+1) when the COMP bit is one.

Table 43-9. Software ouput control behavior when (COMP = 1)

CH(n)OC	CH(n+1)OC	CH(n)OCV	CH(n+1)OCV	Channel (n) Output	Channel (n+1) Output
0	0	X	X	is not modified by SWOC	is not modified by SWOC
1	1	0	0	is forced to zero	is forced to zero
1	1	0	1	is forced to zero	is forced to one
1	1	1	0	is forced to one	is forced to zero
1	1	1	1	is forced to one	is forced to zero

Note

- The CH(n)OC and CH(n+1)OC bits should be equal.
- The COMP bit must not be modified when software output control is enabled, that is, CH(n)OC = 1 and/or CH(n +1)OC = 1.
- Software output control has the same behavior with disabled or enabled FTM counter (see the CLKS field description in the Status and Control register).

43.4.14 Deadtime insertion

The deadtime insertion is enabled when (DTEN = 1) and (DTVAL[5:0] is non- zero).

DEADTIME register defines the deadtime delay that can be used for all FTM channels. The DTPS[1:0] bits define the prescaler for the system clock and the DTVAL[5:0] bits define the deadtime modulo, that is, the number of the deadtime prescaler clocks.

The deadtime delay insertion ensures that no two complementary signals (channels (n) and (n+1)) drive the active state at the same time.

If POL(n) = 0, POL(n+1) = 0, and the deadtime is enabled, then when the channel (n) match (FTM counter = C(n)V) occurs, the channel (n) output remains at the low value until the end of the deadtime delay when the channel (n) output is set. Similarly, when the channel (n+1) match (FTM counter = C(n+1)V) occurs, the channel (n+1) output remains at the low value until the end of the deadtime delay when the channel (n+1) output is set. See the following figures.

If POL(n) = 1, POL(n+1) = 1, and the deadtime is enabled, then when the channel (n) match (FTM counter = C(n)V) occurs, the channel (n) output remains at the high value until the end of the deadtime delay when the channel (n) output is cleared. Similarly, when the channel (n+1) match (FTM counter = C(n+1)V) occurs, the channel (n+1) output remains at the high value until the end of the deadtime delay when the channel (n +1) output is cleared.

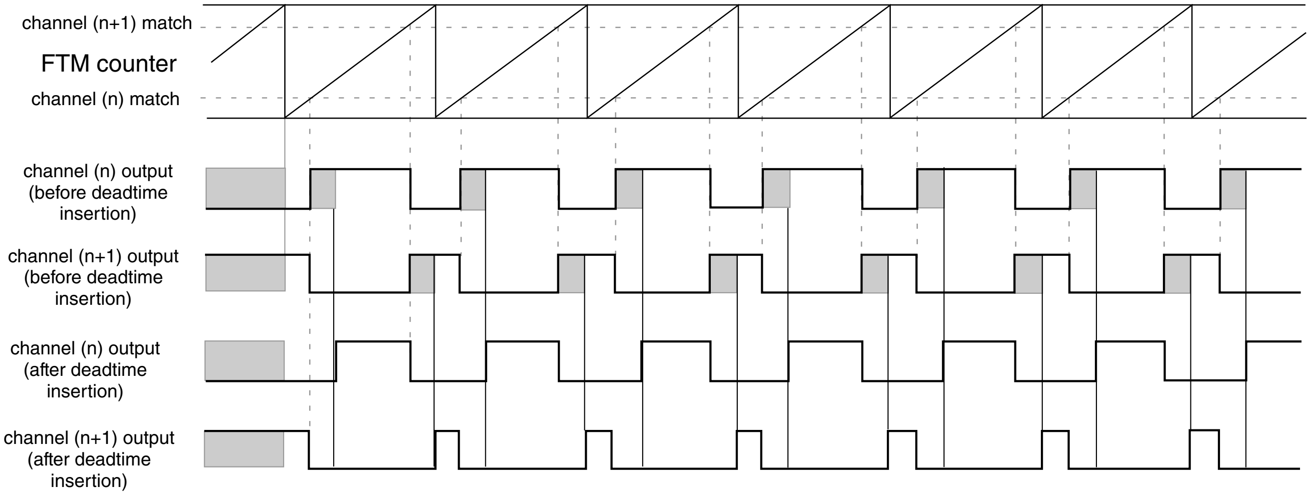

channel (n+1) match
FTM counter
channel (n) match
channel (n) output (before deadtime insertion)
channel (n+1) output (before deadtime insertion)
channel (n) output (after deadtime insertion)
channel (n+1) output (after deadtime insertion)

Figure 43-65. Deadtime insertion with ELSnB:ELSnA = 1:0, POL(n) = 0, and POL(n+1) = 0

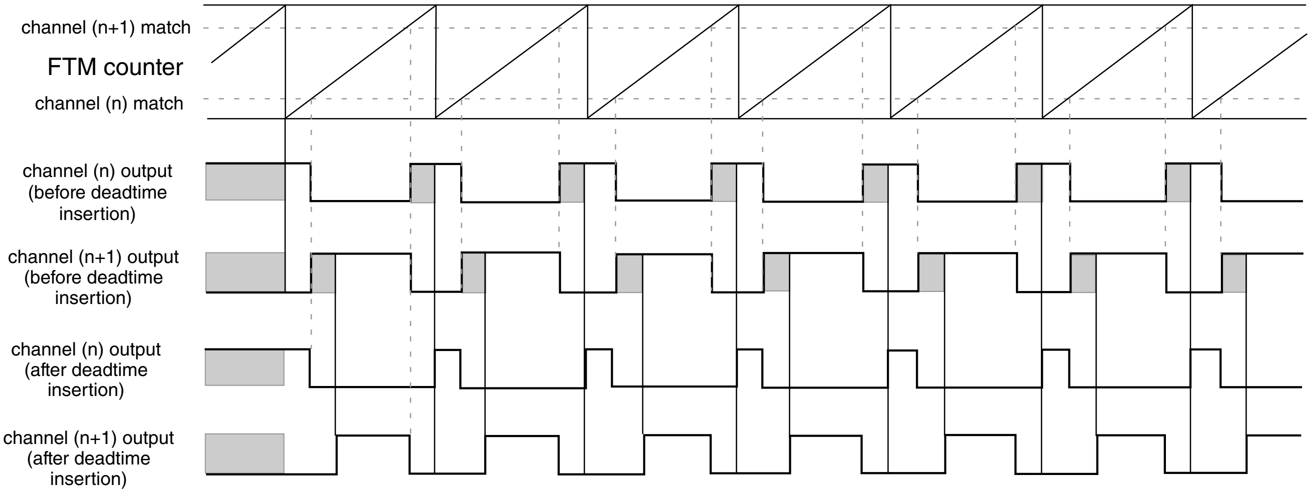

channel (n+1) match
FTM counter
channel (n) match
channel (n) output (before deadtime insertion)
channel (n+1) output (before deadtime insertion)
channel (n) output (after deadtime insertion)
channel (n+1) output (after deadtime insertion)

Figure 43-66. Deadtime insertion with ELSnB:ELSnA = X:1, POL(n) = 0, and POL(n+1) = 0

NOTE

- The deadtime feature must be used only in Complementary mode.
- The deadtime feature is not available in Output Compare mode.

43.4.14.1 Deadtime insertion corner cases

If (PS[2:0] is cleared), (DTPS[1:0] = 0:0 or DTPS[1:0] = 0:1):

- and the deadtime delay is greater than or equal to the channel (n) duty cycle ((C(n +1)V – C(n)V) × system clock), then the channel (n) output is always the inactive value (POL(n) bit value).
- and the deadtime delay is greater than or equal to the channel (n+1) duty cycle ((MOD – CNTIN + 1 – (C(n+1)V – C(n)V)) × system clock), then the channel (n+1) output is always the inactive value (POL(n+1) bit value).

Although, in most cases the deadtime delay is not comparable to channels (n) and (n+1) duty cycle, the following figures show examples where the deadtime delay is comparable to the duty cycle.

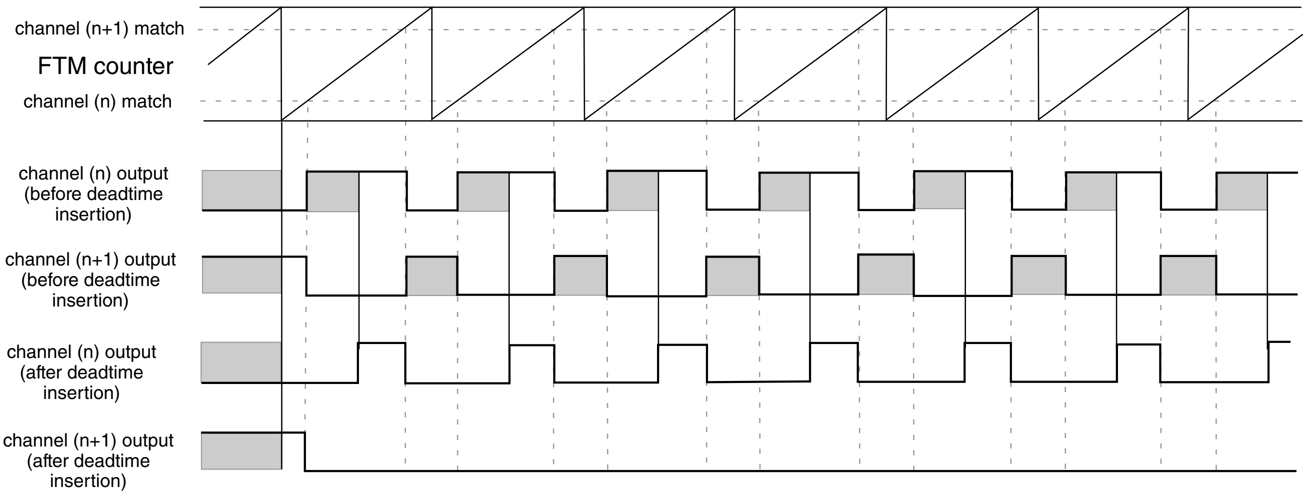

channel (n+1) match
FTM counter
channel (n) match
channel (n) output (before deadtime insertion)
channel (n+1) output (before deadtime insertion)
channel (n) output (after deadtime insertion)
channel (n+1) output (after deadtime insertion)

Figure 43-67. Example of the deadtime insertion (ELSnB:ELSnA = 1:0, POL(n) = 0, and POL(n+1) = 0) when the deadtime delay is comparable to channel (n+1) duty cycle

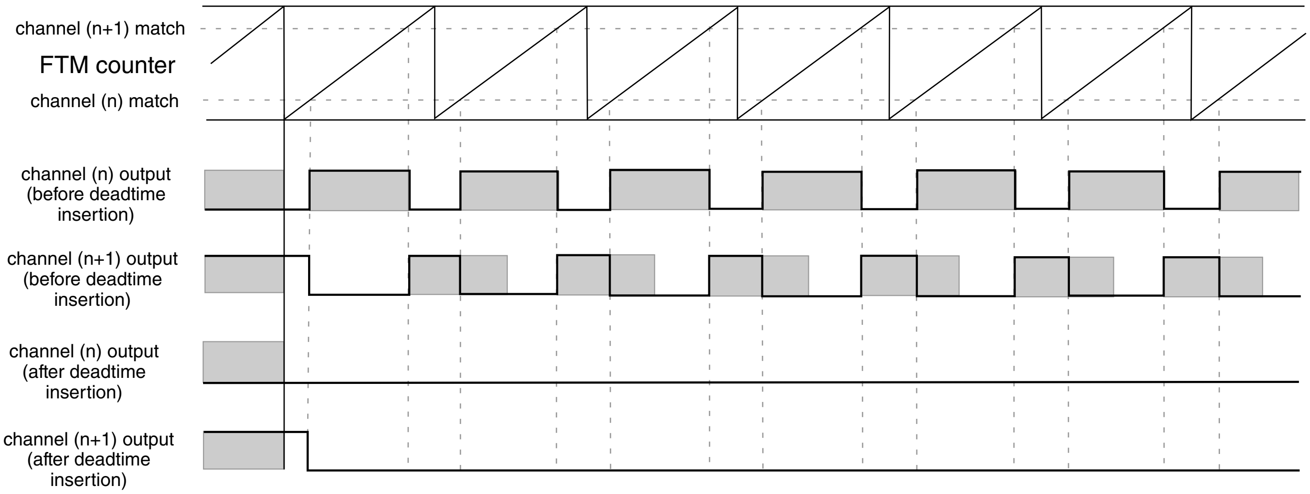

channel (n+1) match
FTM counter
channel (n) match
channel (n) output (before deadtime insertion)
channel (n+1) output (before deadtime insertion)
channel (n) output (after deadtime insertion)
channel (n+1) output (after deadtime insertion)

Figure 43-68. Example of the deadtime insertion (ELSnB:ELSnA = 1:0, POL(n) = 0, and POL(n+1) = 0) when the deadtime delay is comparable to channels (n) and (n+1) duty cycle

43.4.15 Output mask

The output mask can be used to force channels output to their inactive state through software. For example: to control a BLDC motor.

Any write to the OUTMASK register updates its write buffer. The OUTMASK register is updated with its buffer value by PWM synchronization; see OUTMASK register synchronization.

If CHnOM = 1, then the channel (n) output is forced to its inactive state (POLn bit value). If CHnOM = 0, then the channel (n) output is unaffected by the output mask. See the following figure.

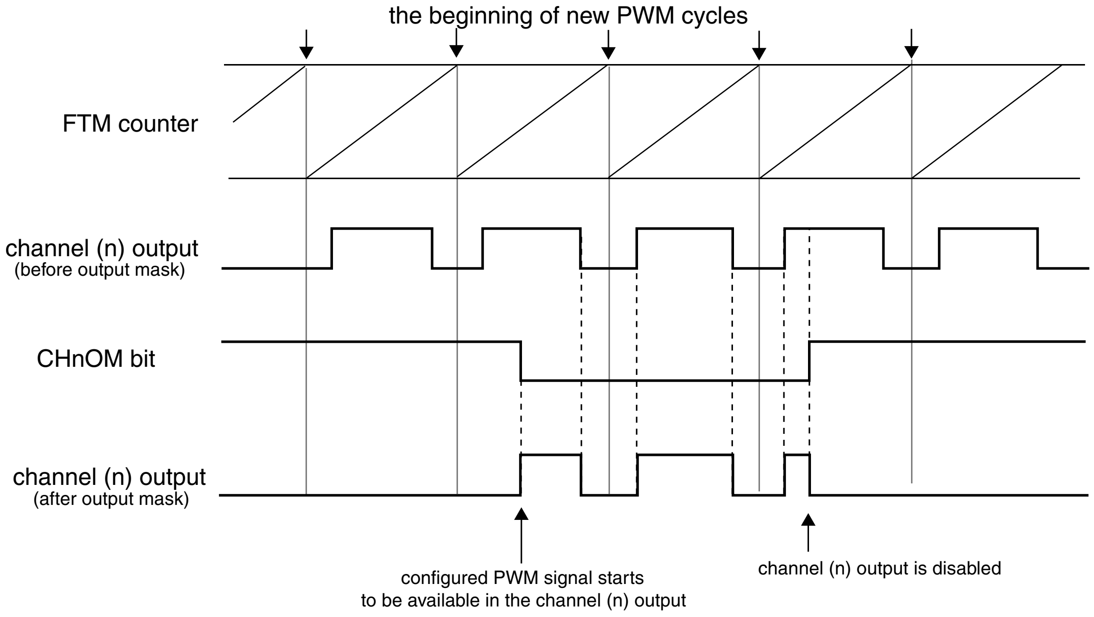

the beginning of new PWM cycles
FTM counter
channel (n) output
(before output mask)
CHnOM bit
channel (n) output
(after output mask)
configured PWM signal starts
to be available in the channel (n) output
channel (n) output is disabled

Figure 43-69. Output mask with POLn = 0

The following table shows the output mask result before the polarity control.

Table 43-10. Output mask result for channel (n) before the polarity control

CHnOM	Output Mask Input	Output Mask Result
0	inactive state	inactive state
	active state	active state
1	inactive state	inactive state
	active state	

43.4.16 Fault control

The fault control is enabled if (FAULTM[1:0] ≠ 0:0).

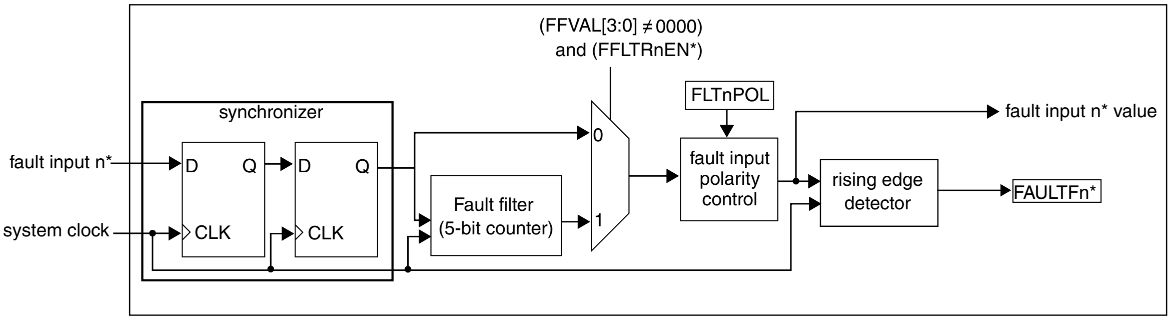
FTM can have up to four fault inputs. FAULTnEN bit (where n = 0, 1, 2, 3) enables the fault input n and FFLTRnEN bit enables the fault input n filter. FFVAL[3:0] bits select the value of the enabled filter in each enabled fault input.

First, each fault input signal is synchronized by the system clock; see the synchronizer block in the following figure. Following synchronization, the fault input n signal enters the filter block. When there is a state change in the fault input n signal, the 5-bit counter is reset and starts counting up. As long as the new state is stable on the fault input n, the counter continues to increment. If the 5-bit counter overflows, that is, the counter exceeds the value of the FFVAL[3:0] bits, the new fault input n value is validated. It is then transmitted as a pulse edge to the edge detector.

If the opposite edge appears on the fault input n signal before validation (counter overflow), the counter is reset. At the next input transition, the counter starts counting again. Any pulse that is shorter than the minimum value selected by FFVAL[3:0] bits (× system clock) is regarded as a glitch and is not passed on to the edge detector.

The fault input n filter is disabled when the FFVAL[3:0] bits are zero or when FAULTnEN = 0. In this case, the fault input n signal is delayed 2 rising edges of the system clock and the FAULTFn bit is set on 3th rising edge of the system clock after a rising edge occurs on the fault input n.

If FFVAL[3:0] ≠ 0000 and FAULTnEN = 1, then the fault input n signal is delayed (3 + FFVAL[3:0]) rising edges of the system clock, that is, the FAULTFn bit is set (4 + FFVAL[3:0]) rising edges of the system clock after a rising edge occurs on the fault input n.



* where n = 3, 2, 1, 0

Figure 43-70. Fault input n control block diagram

If the fault control and fault input n are enabled and a rising edge at the fault input n signal is detected, a fault condition has occurred and the FAULTFn bit is set. The FAULTF bit is the logic OR of FAULTFn[3:0] bits. See the following figure.

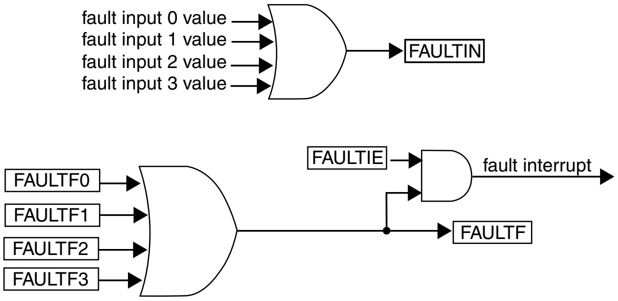


Figure 43-71. FAULTF and FAULTIN bits and fault interrupt

If the fault control is enabled (FAULTM[1:0] ≠ 0:0), a fault condition has occurred and (FAULTEN = 1), then outputs are forced to their safe values:

- Channel (n) output takes the value of POL(n)
- Channel (n+1) takes the value of POL(n+1)

The fault interrupt is generated when (FAULTF = 1) and (FAULTIE = 1). This interrupt request remains set until:

- Software clears the FAULTF bit by reading FAULTF bit as 1 and writing 0 to it
- Software clears the FAULTIE bit
- A reset occurs

43.4.16.1 Automatic fault clearing

If the automatic fault clearing is selected (FAULTM[1:0] = 1:1), then the channels output disabled by fault control is again enabled when the fault input signal (FAULTIN) returns to zero and a new PWM cycle begins. See the following figure.

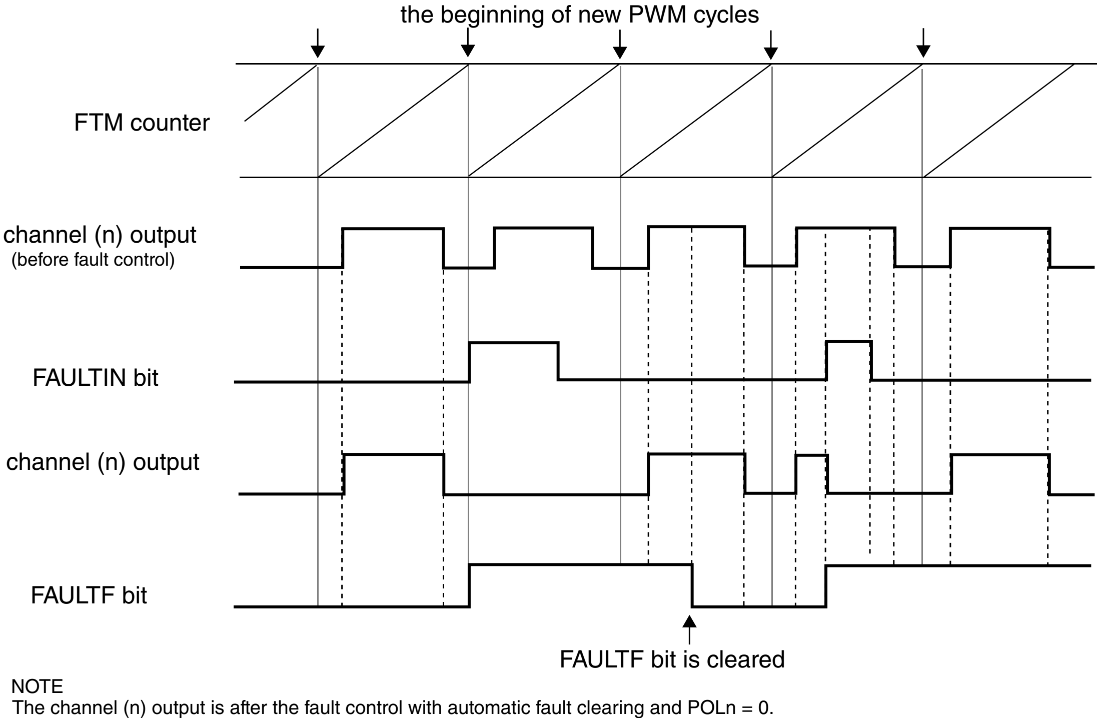

the beginning of new PWM cycles
FTM counter
channel (n) output
(before fault control)
FAULTIN bit
channel (n) output
FAULTF bit
FAULTF bit is cleared
NOTE
The channel (n) output is after the fault control with automatic fault clearing and POLn = 0.

Figure 43-72. Fault control with automatic fault clearing

43.4.16.2 Manual fault clearing

If the manual fault clearing is selected (FAULTM[1:0] = 0:1 or 1:0), then the channels output disabled by fault control is again enabled when the FAULTF bit is cleared and a new PWM cycle begins. See the following figure.

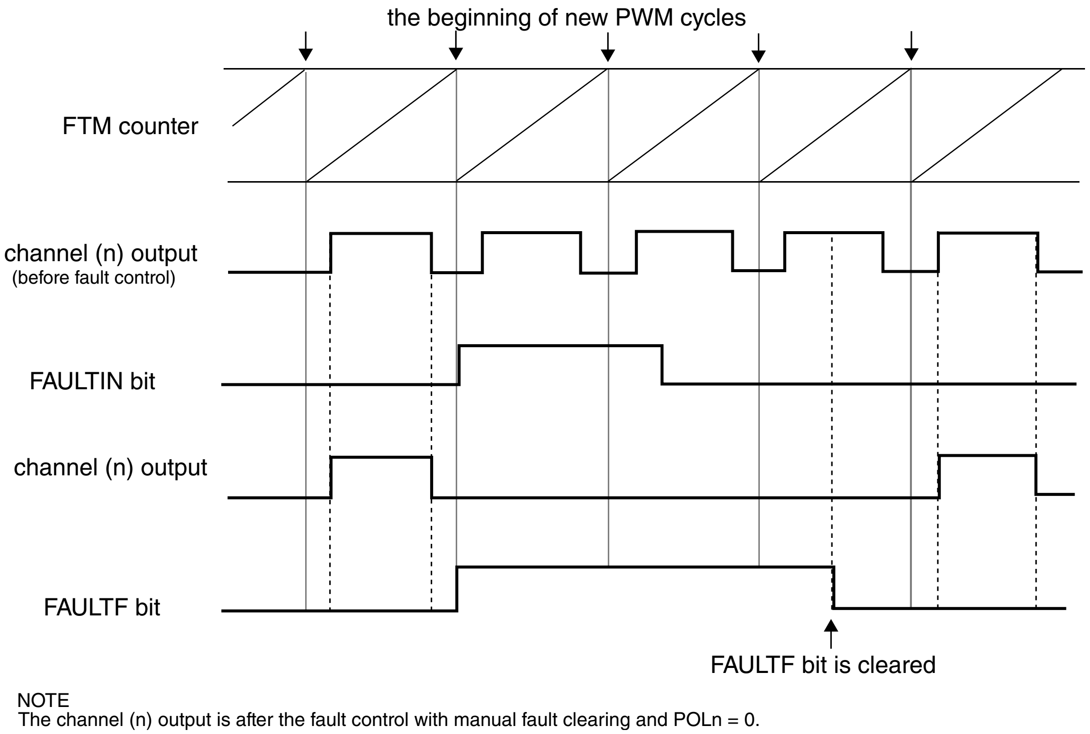

the beginning of new PWM cycles
FTM counter
channel (n) output
(before fault control)
FAULTIN bit
channel (n) output
FAULTF bit
FAULTF bit is cleared
NOTE
The channel (n) output is after the fault control with manual fault clearing and POLn = 0.

Figure 43-73. Fault control with manual fault clearing

43.4.16.3 Fault inputs polarity control

The FLTjPOL bit selects the fault input j polarity, where j = 0, 1, 2, 3:

- If FLTjPOL = 0, the fault j input polarity is high, so the logical one at the fault input j indicates a fault.
- If FLTjPOL = 1, the fault j input polarity is low, so the logical zero at the fault input j indicates a fault.

43.4.17 Polarity control

The POLn bit selects the channel (n) output polarity:

- If POLn = 0, the channel (n) output polarity is high, so the logical one is the active state and the logical zero is the inactive state.
- If POLn = 1, the channel (n) output polarity is low, so the logical zero is the active state and the logical one is the inactive state.

43.4.18 Initialization

The initialization forces the CHnOI bit value to the channel (n) output when a one is written to the INIT bit.

The initialization depends on COMP and DTEN bits. The following table shows the values that channels (n) and (n+1) are forced by initialization when the COMP and DTEN bits are zero.

Table 43-11. Initialization behavior when (COMP = 0 and DTEN = 0)

CH(n)OI	CH(n+1)OI	Channel (n) Output	Channel (n+1) Output
0	0	is forced to zero	is forced to zero
0	1	is forced to zero	is forced to one
1	0	is forced to one	is forced to zero
1	1	is forced to one	is forced to one

The following table shows the values that channels (n) and (n+1) are forced by initialization when (COMP = 1) or (DTEN = 1).

Table 43-12. Initialization behavior when (COMP = 1 or DTEN = 1)

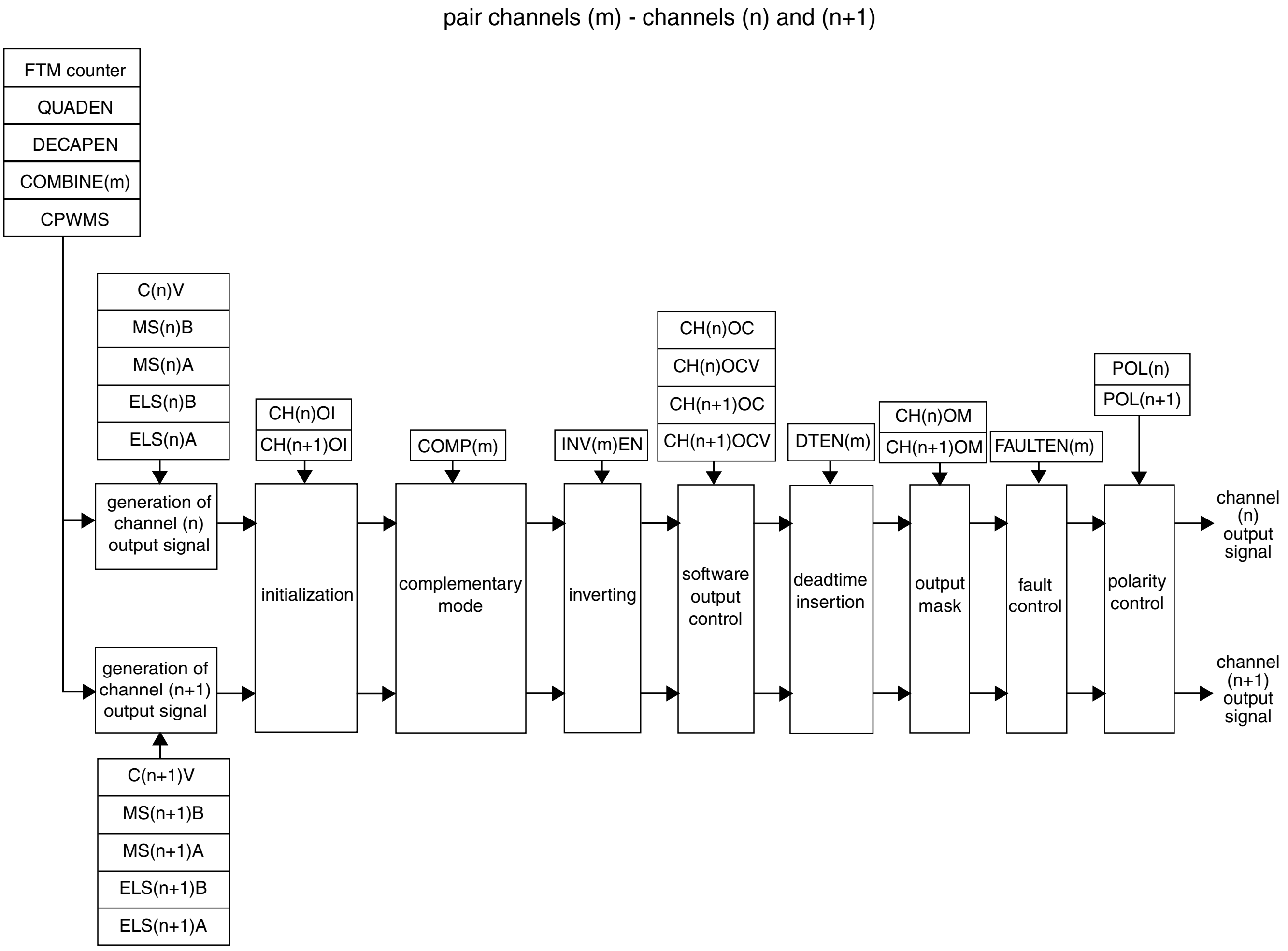
CH(n)OI	CH(n+1)OI	Channel (n) Output	Channel (n+1) Output
0	X	is forced to zero	is forced to one
1	X	is forced to one	is forced to zero

Note

The initialization feature must be used only with disabled FTM counter. See the description of the CLKS field in the Status and Control register.

43.4.19 Features priority

The following figure shows the priority of the features used at the generation of channels (n) and (n+1) outputs signals.



NOTE

The channels (n) and (n+1) are in output compare, EPWM, CPWM or combine modes.

Figure 43-74. Priority of the features used at the generation of channels (n) and (n+1) outputs signals

Note

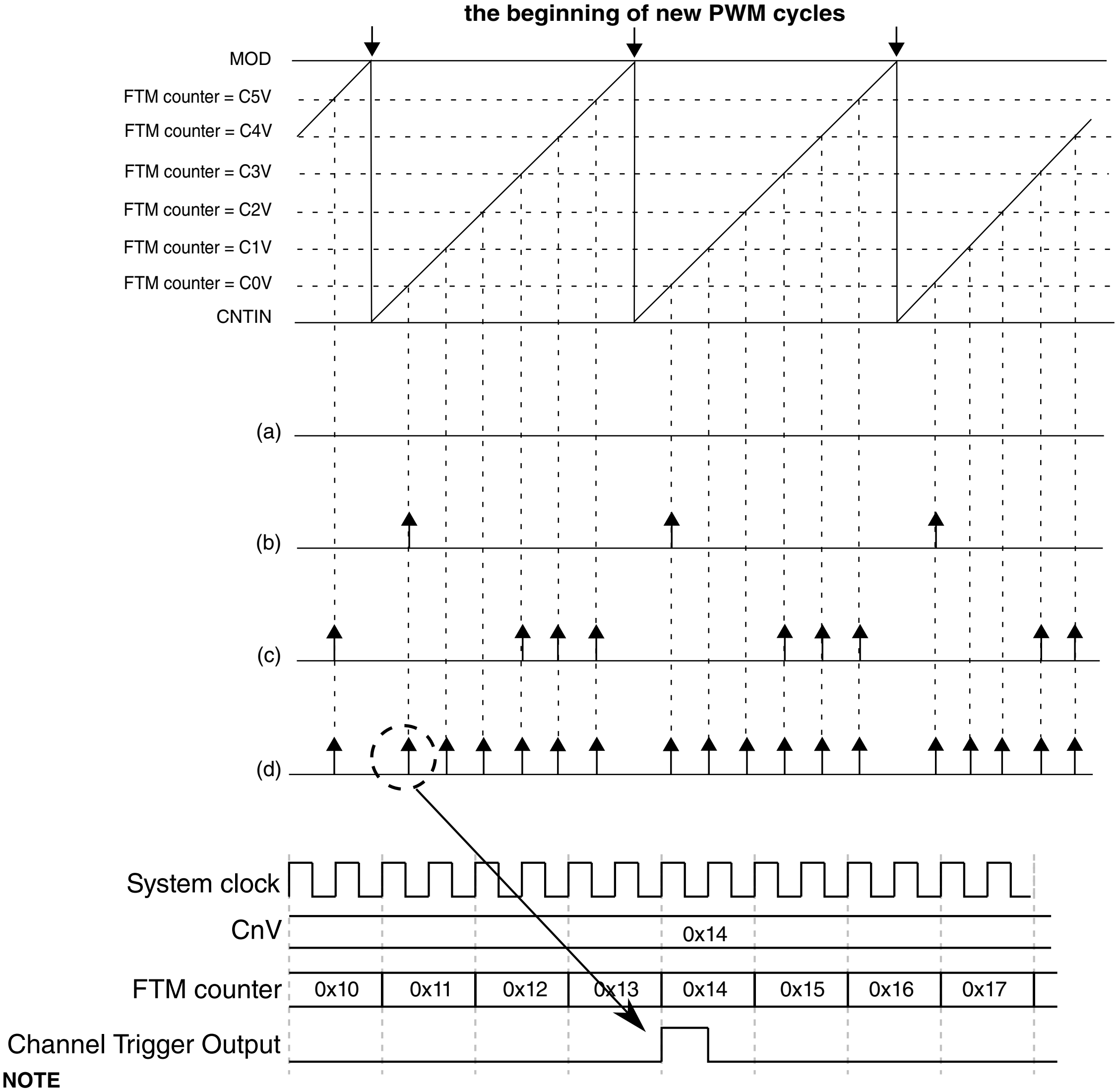
The Initialization feature must not be used with Inverting and Software output control features.

43.4.20 Channel trigger output

If CH(j)TRIG bit of the FTM External Trigger (FTM_EXTTRIG) register is set, where j = 0, 1, 2, 3, 4, or 5, then the FTM generates a trigger when the channel (j) match occurs (FTM counter = C(j)V).

The channel trigger output provides a trigger signal which has one FTM clock period width and is used for on-chip modules.

The FTM is able to generate multiple triggers in one PWM period. Because each trigger is generated for a specific channel, several channels are required to implement this functionality. This behavior is described in the following figure.



NOTE

(a) CH0TRIG = 0, CH1TRIG = 0, CH2TRIG = 0, CH3TRIG = 0, CH4TRIG = 0, CH5TRIG = 0
(b) CH0TRIG = 1, CH1TRIG = 0, CH2TRIG = 0, CH3TRIG = 0, CH4TRIG = 0, CH5TRIG = 0
(c) CH0TRIG = 0, CH1TRIG = 0, CH2TRIG = 0, CH3TRIG = 1, CH4TRIG = 1, CH5TRIG = 1
(d) CH0TRIG = 1, CH1TRIG = 1, CH2TRIG = 1, CH3TRIG = 1, CH4TRIG = 1, CH5TRIG = 1

Figure 43-75. Channel match trigger

43.4.21 Initialization trigger

If INITTRIGEN = 1, then the FTM generates a trigger when the FTM counter is updated with the CNTIN register value in the following cases.

- The FTM counter is automatically updated with the CNTIN register value by the selected counting mode.
- When there is a write to CNT register.
- When there is the FTM counter synchronization.
- If (CNT = CNTIN), (CLKS[1:0] = 0:0), and a value different from zero is written to CLKS[1:0] bits.

The following figures show these cases.

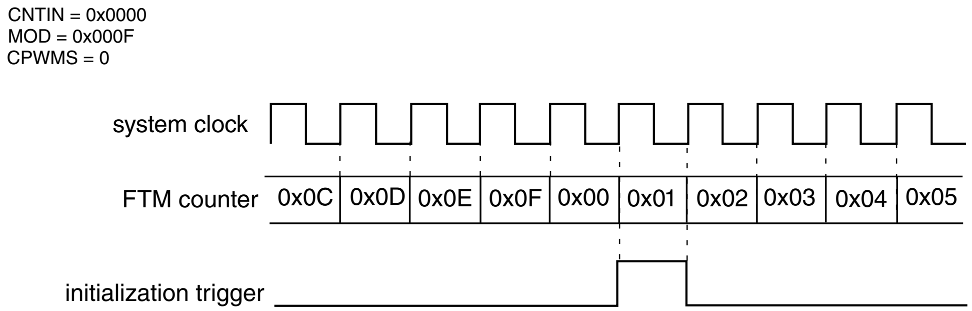


Figure 43-76. Initialization trigger is generated when the FTM counting achieves the CNTIN register value

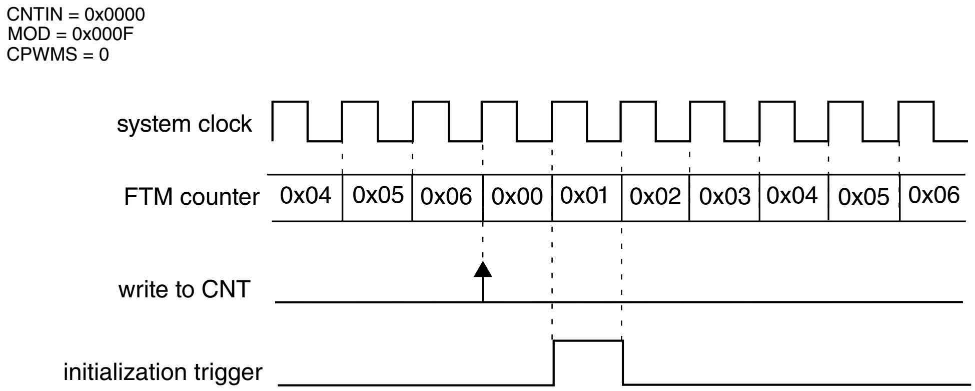


Figure 43-77. Initialization trigger is generated when there is a write to CNT register

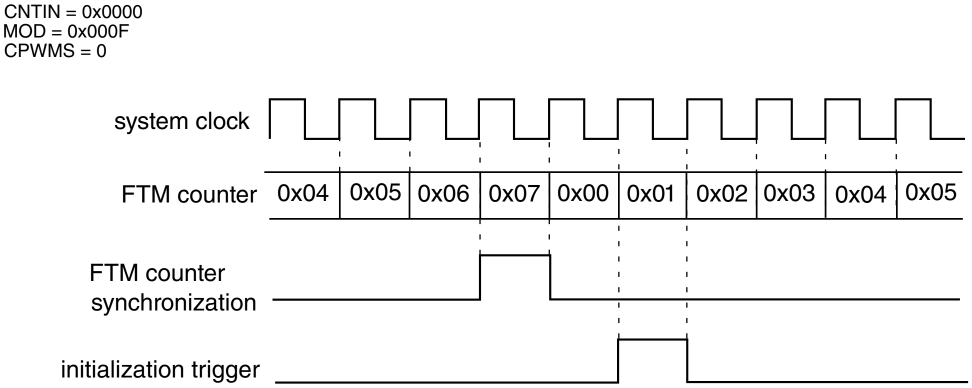


Figure 43-78. Initialization trigger is generated when there is the FTM counter synchronization

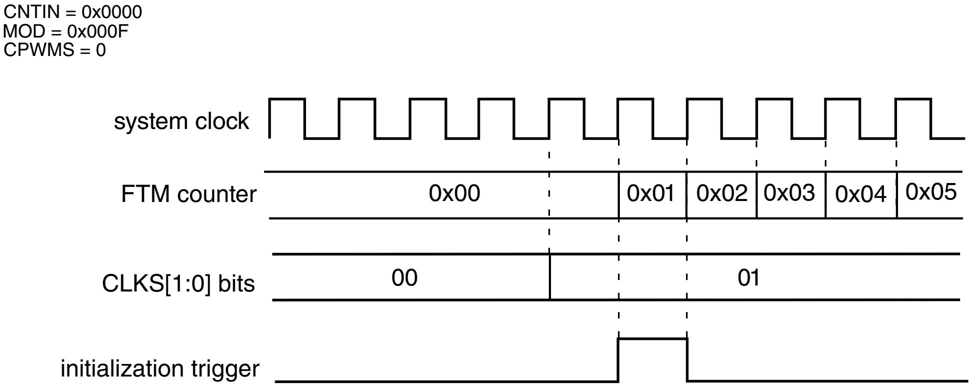


Figure 43-79. Initialization trigger is generated if (CNT = CNTIN), (CLKS[1:0] = 0:0), and a value different from zero is written to CLKS[1:0] bits

The initialization trigger output provides a trigger signal that is used for on-chip modules.

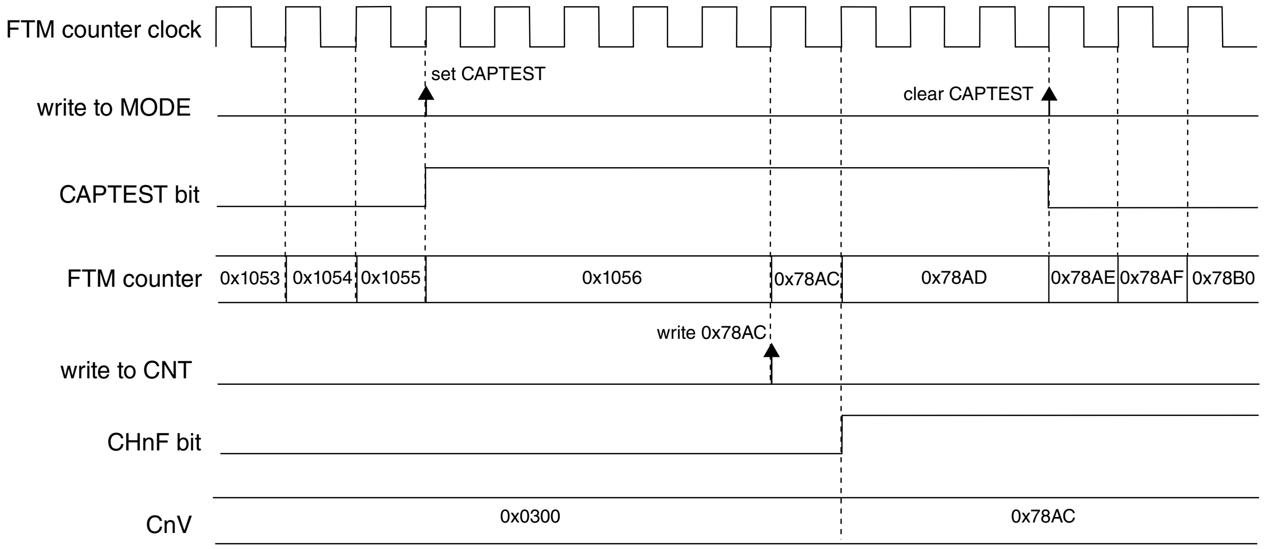
43.4.22 Capture Test mode

The Capture Test mode allows to test the CnV registers, the FTM counter and the interconnection logic between the FTM counter and CnV registers.

In this test mode, all channels must be configured for Input Capture mode and FTM counter must be configured to the Up counting.

When the Capture Test mode is enabled (CAPTEST = 1), the FTM counter is frozen and any write to CNT register updates directly the FTM counter; see the following figure. After it was written, all CnV registers are updated with the written value to CNT register and CHnF bits are set. Therefore, the FTM counter is updated with its next value according to its configuration. Its next value depends on CNTIN, MOD, and the written value to FTM counter.

The next reads of CnV registers return the written value to the FTM counter and the next reads of CNT register return FTM counter next value.



NOTE
- FTM counter configuration: (FTMEN = 1), (QUADEN = 0), (CAPTEST = 1), (CPWMS = 0), (CNTIN = 0x0000), and (MOD = 0xFFFF)
- FTM channel n configuration: input capture mode - (DECAPEN = 0), (COMBINE = 0), and (MSnB:MSnA = 0:0)

Figure 43-80. Capture Test mode

43.4.23 DMA

The channel generates a DMA transfer request according to DMA and CHnIE bits. See the following table.

Table 43-13. Channel DMA transfer request

DMA	CHnIE	Channel DMA Transfer Request	Channel Interrupt
0	0	The channel DMA transfer request is not generated.	The channel interrupt is not generated.
0	1	The channel DMA transfer request is not generated.	The channel interrupt is generated if (CHnF = 1).
1	0	The channel DMA transfer request is not generated.	The channel interrupt is not generated.
1	1	The channel DMA transfer request is generated if (CHnF = 1).	The channel interrupt is not generated.

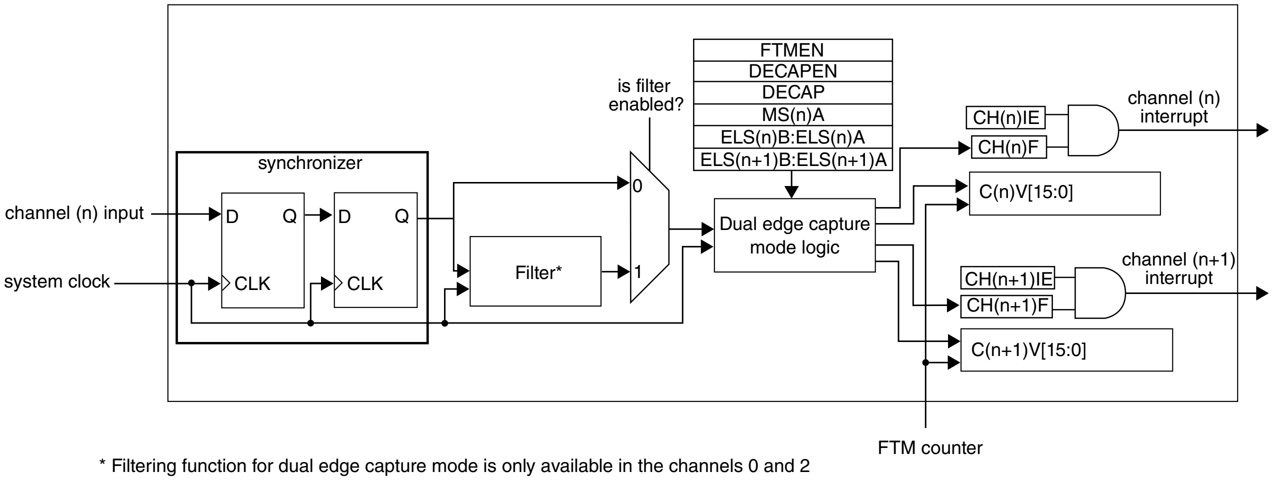
If DMA = 1, the CHnF bit is cleared either by channel DMA transfer done or reading CnSC while CHnF is set and then writing a zero to CHnF bit according to CHnIE bit. See the following table.

Table 43-14. Clear CHnF bit when DMA = 1

CHnIE	How CHnF Bit Can Be Cleared
0	CHnF bit is cleared either when the channel DMA transfer is done or by reading CnSC while CHnF is set and then writing a 0 to CHnF bit.
1	CHnF bit is cleared when the channel DMA transfer is done.

43.4.24 Dual Edge Capture mode

The Dual Edge Capture mode is selected if DECAPEN = 1. This mode allows to measure a pulse width or period of the signal on the input of channel (n) of a channel pair. The channel (n) filter can be active in this mode when n is 0 or 2.



* Filtering function for dual edge capture mode is only available in the channels 0 and 2

Figure 43-81. Dual Edge Capture mode block diagram

The MS(n)A bit defines if the Dual Edge Capture mode is one-shot or continuous.

The ELS(n)B:ELS(n)A bits select the edge that is captured by channel (n), and ELS(n +1)B:ELS(n+1)A bits select the edge that is captured by channel (n+1). If both ELS(n)B:ELS(n)A and ELS(n+1)B:ELS(n+1)A bits select the same edge, then it is the period measurement. If these bits select different edges, then it is a pulse width measurement.

In the Dual Edge Capture mode, only channel (n) input is used and channel (n+1) input is ignored.

If the selected edge by channel (n) bits is detected at channel (n) input, then CH(n)F bit is set and the channel (n) interrupt is generated (if CH(n)IE = 1). If the selected edge by channel (n+1) bits is detected at channel (n) input and (CH(n)F = 1), then CH(n+1)F bit is set and the channel (n+1) interrupt is generated (if CH(n+1)IE = 1).

The C(n)V register stores the value of FTM counter when the selected edge by channel (n) is detected at channel (n) input. The C(n+1)V register stores the value of FTM counter when the selected edge by channel (n+1) is detected at channel (n) input.

In this mode, a coherency mechanism ensures coherent data when the C(n)V and C(n+1)V registers are read. The only requirement is that C(n)V must be read before C(n+1)V.

Note

- The CH(n)F, CH(n)IE, MS(n)A, ELS(n)B, and ELS(n)A bits are channel (n) bits.
- The CH(n+1)F, CH(n+1)IE, MS(n+1)A, ELS(n+1)B, and ELS(n+1)A bits are channel (n+1) bits.
- The Dual Edge Capture mode must be used with ELS(n)B:ELS(n)A = 0:1 or 1:0, ELS(n+1)B:ELS(n+1)A = 0:1 or 1:0 and the FTM counter in Free running counter.

43.4.24.1 One-Shot Capture mode

The One-Shot Capture mode is selected when (DECAPEN = 1), and (MS(n)A = 0). In this capture mode, only one pair of edges at the channel (n) input is captured. The ELS(n)B:ELS(n)A bits select the first edge to be captured, and ELS(n+1)B:ELS(n+1)A bits select the second edge to be captured.

The edge captures are enabled while DECAP bit is set. For each new measurement in One-Shot Capture mode, first the CH(n)F and CH(n+1) bits must be cleared, and then the DECAP bit must be set.

In this mode, the DECAP bit is automatically cleared by FTM when the edge selected by channel (n+1) is captured. Therefore, while DECAP bit is set, the one-shot capture is in process. When this bit is cleared, both edges were captured and the captured values are ready for reading in the C(n)V and C(n+1)V registers.

Similarly, when the CH(n+1)F bit is set, both edges were captured and the captured values are ready for reading in the C(n)V and C(n+1)V registers.

43.4.24.2 Continuous Capture mode

The Continuous Capture mode is selected when (DECAPEN = 1), and (MS(n)A = 1). In this capture mode, the edges at the channel (n) input are captured continuously. The ELS(n)B:ELS(n)A bits select the initial edge to be captured, and ELS(n+1)B:ELS(n+1)A bits select the final edge to be captured.

The edge captures are enabled while DECAP bit is set. For the initial use, first the CH(n)F and CH(n+1)F bits must be cleared, and then DECAP bit must be set to start the continuous measurements.

When the CH(n+1)F bit is set, both edges were captured and the captured values are ready for reading in the C(n)V and C(n+1)V registers. The latest captured values are always available in these registers even after the DECAP bit is cleared.

In this mode, it is possible to clear only the CH(n+1)F bit. Therefore, when the CH(n+1)F bit is set again, the latest captured values are available in C(n)V and C(n+1)V registers.

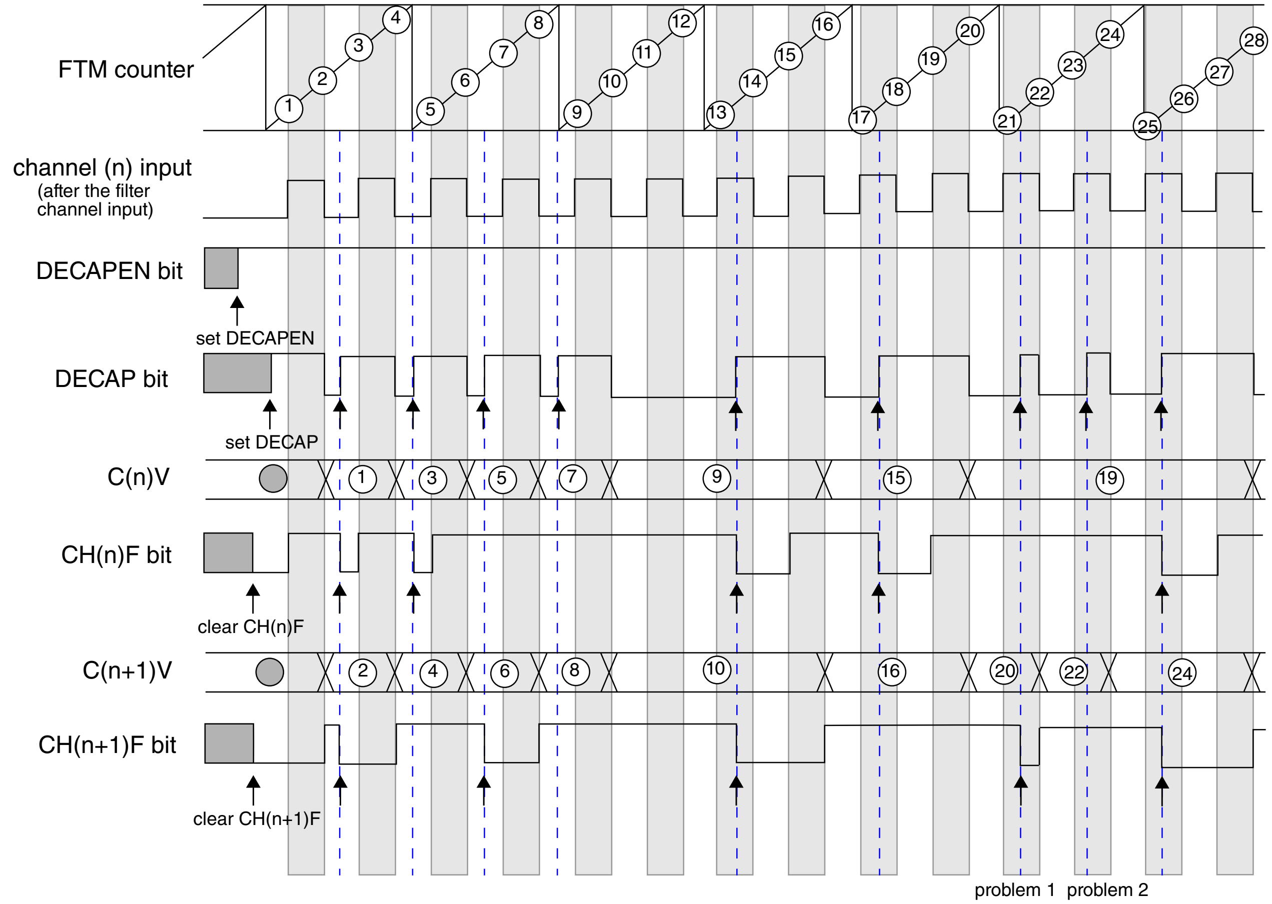
For a new sequence of the measurements in the Dual Edge Capture – Continuous mode, clear the CH(n)F and CH(n+1)F bits to start new measurements.

43.4.24.3 Pulse width measurement

If the channel (n) is configured to capture rising edges (ELS(n)B:ELS(n)A = 0:1) and the channel (n+1) to capture falling edges (ELS(n+1)B:ELS(n+1)A = 1:0), then the positive polarity pulse width is measured. If the channel (n) is configured to capture falling edges (ELS(n)B:ELS(n)A = 1:0) and the channel (n+1) to capture rising edges (ELS(n +1)B:ELS(n+1)A = 0:1), then the negative polarity pulse width is measured.

The pulse width measurement can be made in One-Shot Capture mode or Continuous Capture mode.

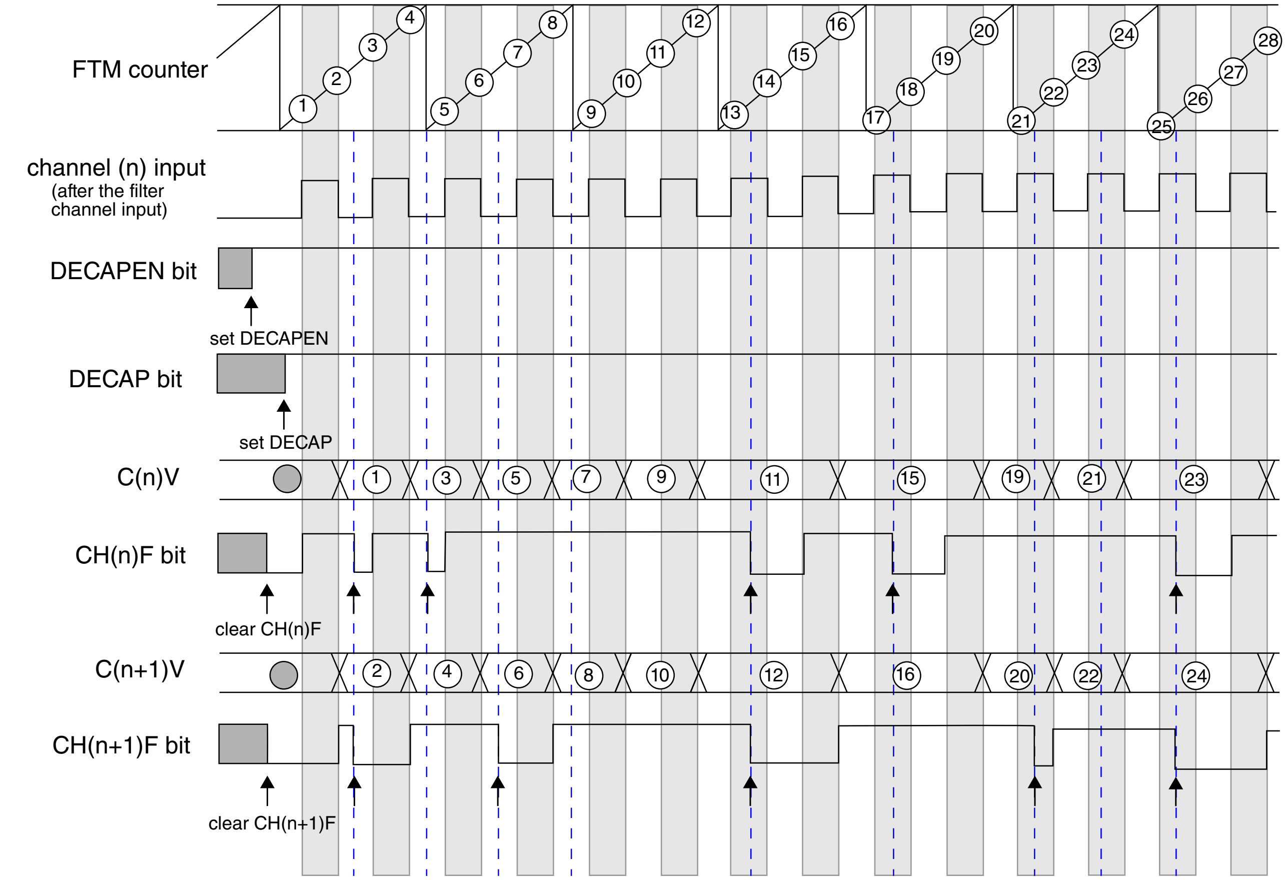
The following figure shows an example of the Dual Edge Capture – One-Shot mode used to measure the positive polarity pulse width. The DECAPEN bit selects the Dual Edge Capture mode, so it remains set. The DECAP bit is set to enable the measurement of next positive polarity pulse width. The CH(n)F bit is set when the first edge of this pulse is detected, that is, the edge selected by ELS(n)B:ELS(n)A bits. The CH(n+1)F bit is set and DECAP bit is cleared when the second edge of this pulse is detected, that is, the edge selected by ELS(n+1)B:ELS(n+1)A bits. Both DECAP and CH(n+1)F bits indicate when two edges of the pulse were captured and the C(n)V and C(n+1)V registers are ready for reading.



Note
- The commands set DECAPEN, set DECAP, clear CH(n)F, and clear CH(n+1)F are made by the user.
- Problem 1: channel (n) input = 1, set DECAP, not clear CH(n)F, and clear CH(n+1)F.
- Problem 2: channel (n) input = 1, set DECAP, not clear CH(n)F, and not clear CH(n+1)F.

Figure 43-82. Dual Edge Capture – One-Shot mode for positive polarity pulse width measurement

The following figure shows an example of the Dual Edge Capture – Continuous mode used to measure the positive polarity pulse width. The DECAPEN bit selects the Dual Edge Capture mode, so it remains set. While the DECAP bit is set the configured measurements are made. The CH(n)F bit is set when the first edge of the positive polarity pulse is detected, that is, the edge selected by ELS(n)B:ELS(n)A bits. The CH(n+1)F bit is set when the second edge of this pulse is detected, that is, the edge selected by ELS(n+1)B:ELS(n+1)A bits. The CH(n+1)F bit indicates when two edges of the pulse were captured and the C(n)V and C(n+1)V registers are ready for reading.



Note
- The commands set DECAPEN, set DECAP, clear CH(n)F, and clear CH(n+1)F are made by the user.

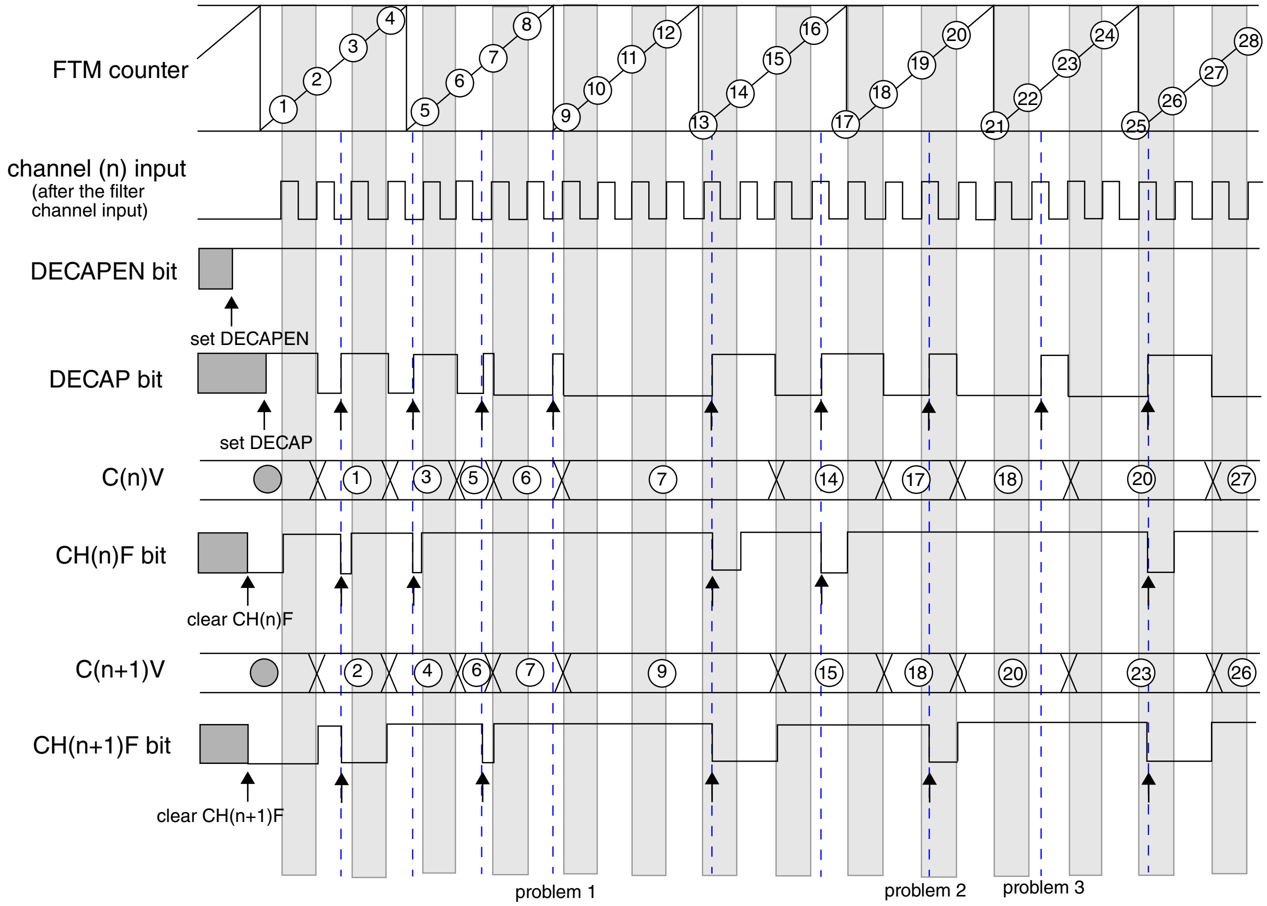
Figure 43-83. Dual Edge Capture – Continuous mode for positive polarity pulse width measurement

43.4.24.4 Period measurement

If the channels (n) and (n+1) are configured to capture consecutive edges of the same polarity, then the period of the channel (n) input signal is measured. If both channels (n) and (n+1) are configured to capture rising edges (ELS(n)B:ELS(n)A = 0:1 and ELS(n +1)B:ELS(n+1)A = 0:1), then the period between two consecutive rising edges is measured. If both channels (n) and (n+1) are configured to capture falling edges (ELS(n)B:ELS(n)A = 1:0 and ELS(n+1)B:ELS(n+1)A = 1:0), then the period between two consecutive falling edges is measured.

The period measurement can be made in One-Shot Capture mode or Continuous Capture mode.

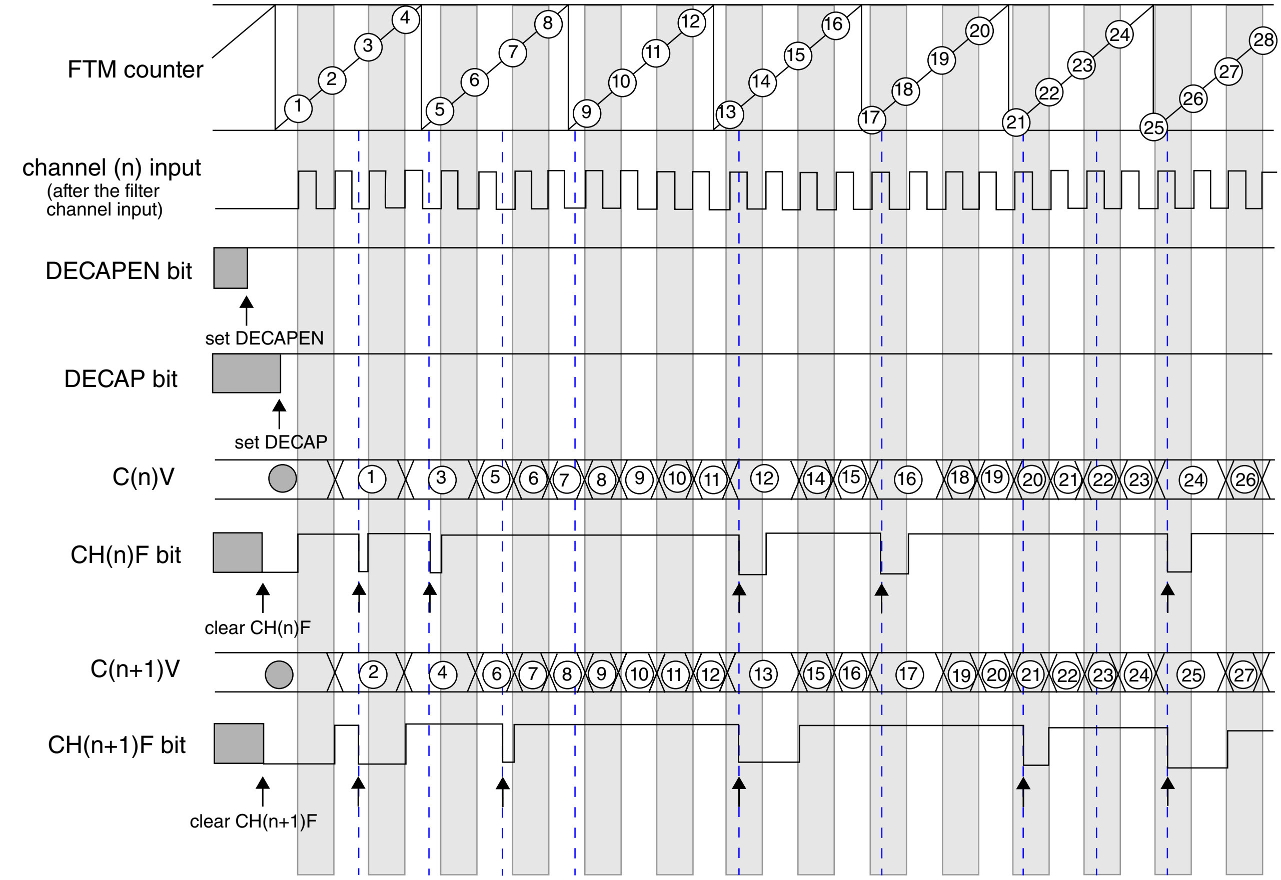
The following figure shows an example of the Dual Edge Capture – One-Shot mode used to measure the period between two consecutive rising edges. The DECAPEN bit selects the Dual Edge Capture mode, so it remains set. The DECAP bit is set to enable the measurement of next period. The CH(n)F bit is set when the first rising edge is detected, that is, the edge selected by ELS(n)B:ELS(n)A bits. The CH(n+1)F bit is set and DECAP bit is cleared when the second rising edge is detected, that is, the edge selected by ELS(n +1)B:ELS(n+1)A bits. Both DECAP and CH(n+1)F bits indicate when two selected edges were captured and the C(n)V and C(n+1)V registers are ready for reading.



Note
- The commands set DECAPEN, set DECAP, clear CH(n)F, and clear CH(n+1)F are made by the user.
- Problem 1: channel (n) input = 0, set DECAP, not clear CH(n)F, and not clear CH(n+1)F.
- Problem 2: channel (n) input = 1, set DECAP, not clear CH(n)F, and clear CH(n+1)F.
- Problem 3: channel (n) input = 1, set DECAP, not clear CH(n)F, and not clear CH(n+1)F.

Figure 43-84. Dual Edge Capture – One-Shot mode to measure of the period between two consecutive rising edges

The following figure shows an example of the Dual Edge Capture – Continuous mode used to measure the period between two consecutive rising edges. The DECAPEN bit selects the Dual Edge Capture mode, so it remains set. While the DECAP bit is set the configured measurements are made. The CH(n)F bit is set when the first rising edge is detected, that is, the edge selected by ELS(n)B:ELS(n)A bits. The CH(n+1)F bit is set when the second rising edge is detected, that is, the edge selected by ELS(n+1)B:ELS(n+1)A bits. The CH(n+1)F bit indicates when two edges of the period were captured and the C(n)V and C(n+1)V registers are ready for reading.



Note
- The commands set DECAPEN, set DECAP, clear CH(n)F, and clear CH(n+1)F are made by the user.

Figure 43-85. Dual Edge Capture – Continuous mode to measure of the period between two consecutive rising edges

43.4.24.5 Read coherency mechanism

The Dual Edge Capture mode implements a read coherency mechanism between the FTM counter value captured in C(n)V and C(n+1)V registers. The read coherency mechanism is illustrated in the following figure. In this example, the channels (n) and (n+1) are in Dual Edge Capture – Continuous mode for positive polarity pulse width measurement. Thus, the channel (n) is configured to capture the FTM counter value when there is a rising edge at channel (n) input signal, and channel (n+1) to capture the FTM counter value when there is a falling edge at channel (n) input signal.

When a rising edge occurs in the channel (n) input signal, the FTM counter value is captured into channel (n) capture buffer. The channel (n) capture buffer value is transferred to C(n)V register when a falling edge occurs in the channel (n) input signal. C(n)V register has the FTM counter value when the previous rising edge occurred, and the channel (n) capture buffer has the FTM counter value when the last rising edge occurred.

When a falling edge occurs in the channel (n) input signal, the FTM counter value is captured into channel (n+1) capture buffer. The channel (n+1) capture buffer value is transferred to C(n+1)V register when the C(n)V register is read.

In the following figure, the read of C(n)V returns the FTM counter value when the event 1 occurred and the read of C(n+1)V returns the FTM counter value when the event 2 occurred.

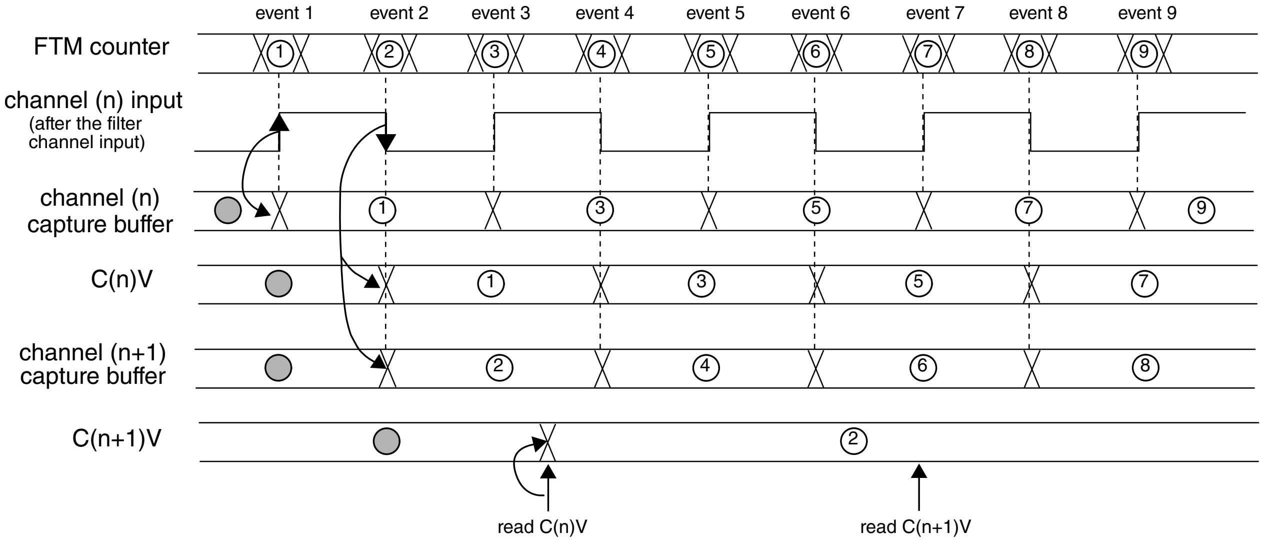


Figure 43-86. Dual Edge Capture mode read coherency mechanism

C(n)V register must be read prior to C(n+1)V register in dual edge capture one-shot and continuous modes for the read coherency mechanism works properly.

43.4.25 Quadrature Decoder mode

The Quadrature Decoder mode is selected if (QUADEN = 1). The Quadrature Decoder mode uses the input signals phase A and B to control the FTM counter increment and decrement. The following figure shows the quadrature decoder block diagram.

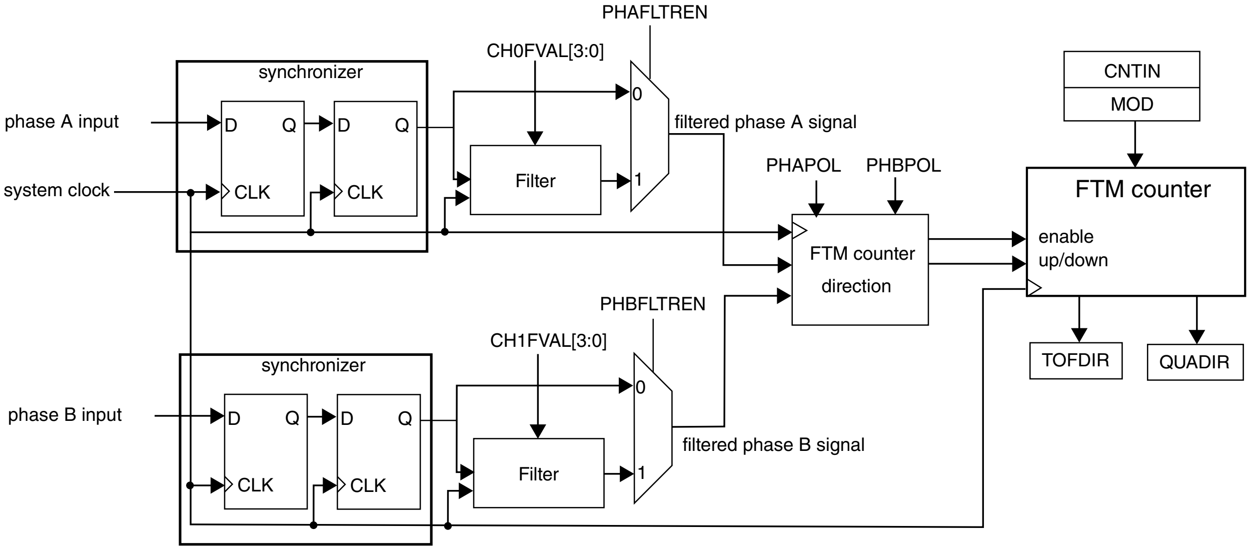


Figure 43-87. Quadrature Decoder block diagram

Each one of input signals phase A and B has a filter that is equivalent to the filter used in the channels input; Filter for Input Capture mode. The phase A input filter is enabled by PHAFLTREN bit and this filter's value is defined by CH0FVAL[3:0] bits (CH(n)FVAL[3:0] bits in FILTER0 register). The phase B input filter is enabled by PHBFLTREN bit and this filter's value is defined by CH1FVAL[3:0] bits (CH(n +1)FVAL[3:0] bits in FILTER0 register).

Except for CH0FVAL[3:0] and CH1FVAL[3:0] bits, no channel logic is used in Quadrature Decoder mode.

Note

Notice that the FTM counter is clocked by the phase A and B input signals when quadrature decoder mode is selected. Therefore it is expected that the Quadrature Decoder be used only with the FTM channels in input capture or output compare modes.

Note

An edge at phase A must not occur together an edge at phase B and vice-versa.

The PHAPOL bit selects the polarity of the phase A input, and the PHBPOL bit selects the polarity of the phase B input.

The QUADMODE selects the encoding mode used in the Quadrature Decoder mode. If QUADMODE = 1, then the count and direction encoding mode is enabled; see the following figure. In this mode, the phase B input value indicates the counting direction, and the phase A input defines the counting rate. The FTM counter is updated when there is a rising edge at phase A input signal.

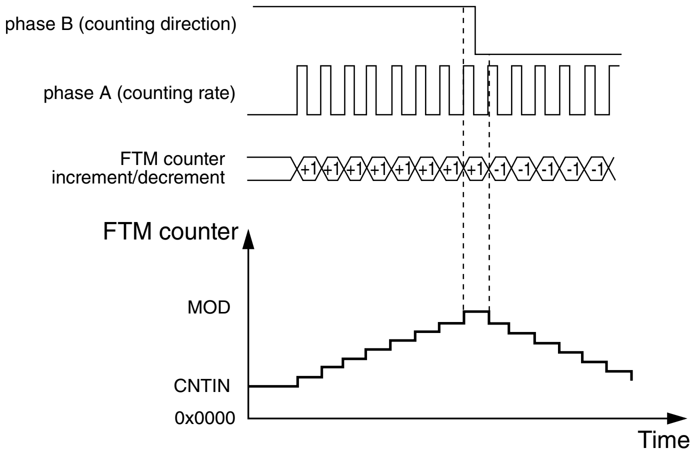


Figure 43-88. Quadrature Decoder – Count and Direction Encoding mode

If QUADMODE = 0, then the Phase A and Phase B Encoding mode is enabled; see the following figure. In this mode, the relationship between phase A and B signals indicates the counting direction, and phase A and B signals define the counting rate. The FTM counter is updated when there is an edge either at the phase A or phase B signals.

If PHAPOL = 0 and PHBPOL = 0, then the FTM counter increment happens when:

- there is a rising edge at phase A signal and phase B signal is at logic zero;
- there is a rising edge at phase B signal and phase A signal is at logic one;
- there is a falling edge at phase B signal and phase A signal is at logic zero;
- there is a falling edge at phase A signal and phase B signal is at logic one;

and the FTM counter decrement happens when:

- there is a falling edge at phase A signal and phase B signal is at logic zero;
- there is a falling edge at phase B signal and phase A signal is at logic one;
- there is a rising edge at phase B signal and phase A signal is at logic zero;
- there is a rising edge at phase A signal and phase B signal is at logic one.

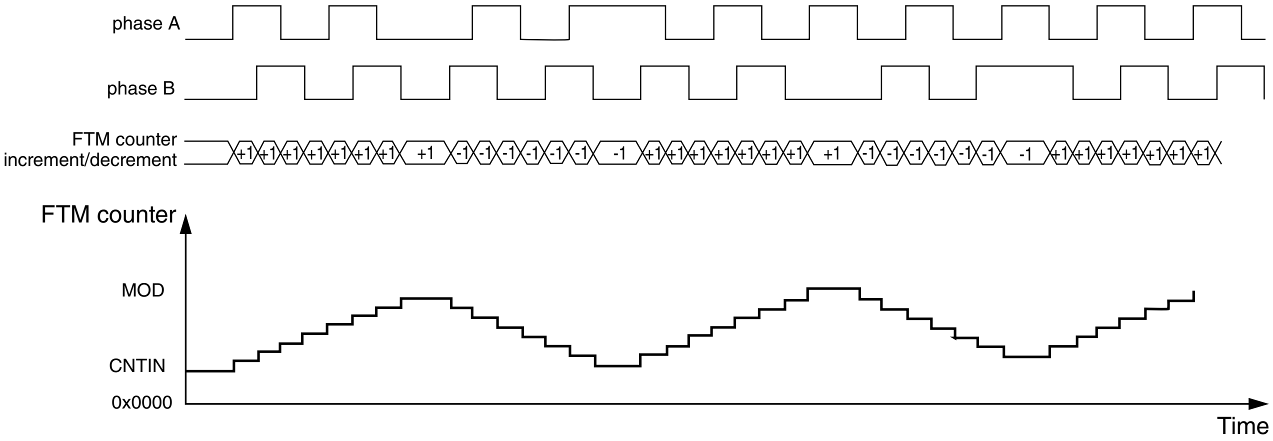


Figure 43-89. Quadrature Decoder – Phase A and Phase B Encoding mode

The following figure shows the FTM counter overflow in up counting. In this case, when the FTM counter changes from MOD to CNTIN, TOF and TOFDIR bits are set. TOF bit indicates the FTM counter overflow occurred. TOFDIR indicates the counting was up when the FTM counter overflow occurred.

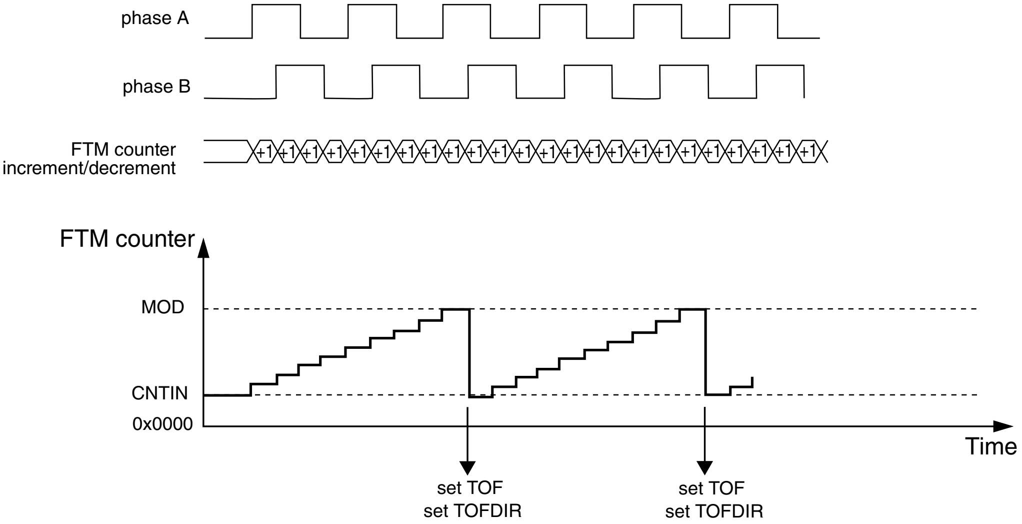


Figure 43-90. FTM Counter overflow in up counting for Quadrature Decoder mode

The following figure shows the FTM counter overflow in down counting. In this case, when the FTM counter changes from CNTIN to MOD, TOF bit is set and TOFDIR bit is cleared. TOF bit indicates the FTM counter overflow occurred. TOFDIR indicates the counting was down when the FTM counter overflow occurred.

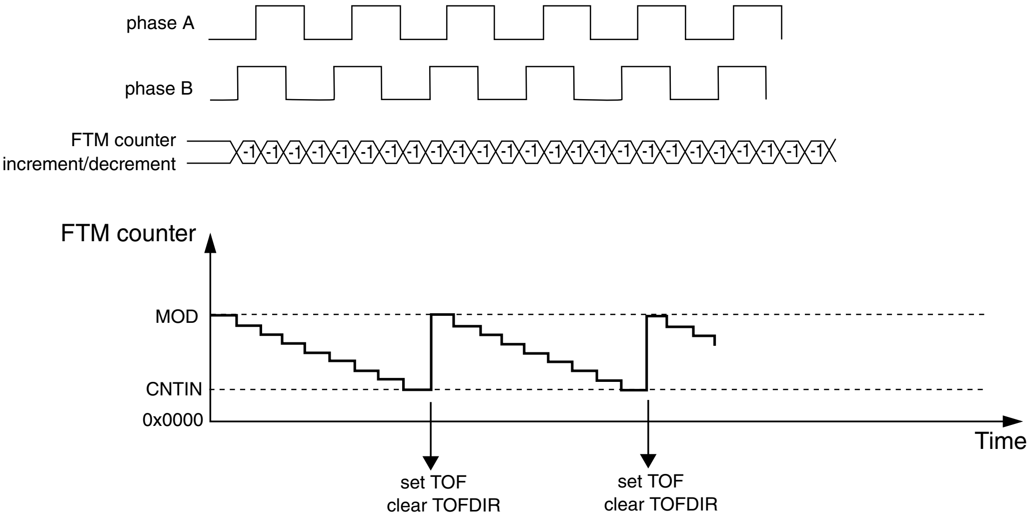


Figure 43-91. FTM counter overflow in down counting for Quadrature Decoder mode

43.4.25.1 Quadrature Decoder boundary conditions

The following figures show the FTM counter responding to motor jittering typical in motor position control applications.

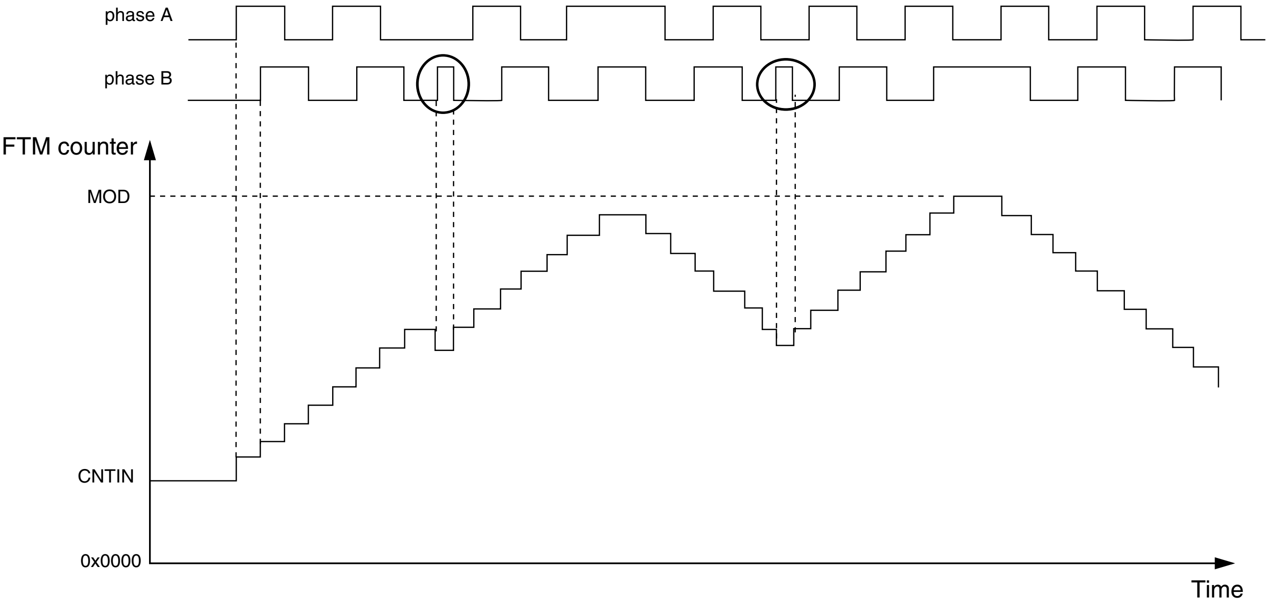


Figure 43-92. Motor position jittering in a mid count value

The following figure shows motor jittering produced by the phase B and A pulses respectively:

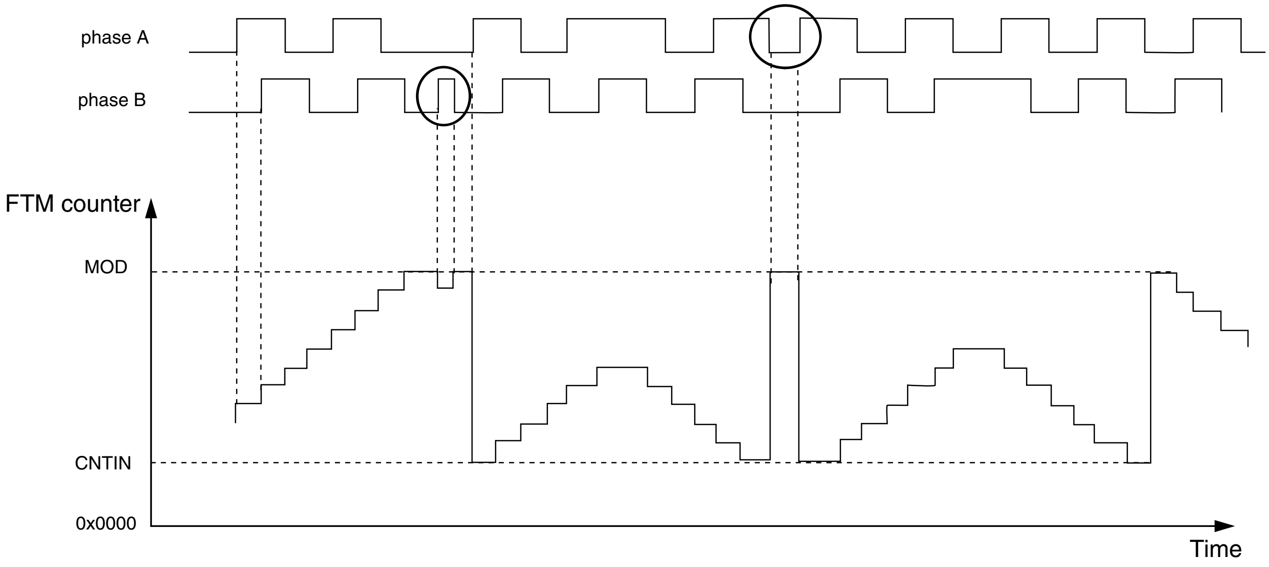


Figure 43-93. Motor position jittering near maximum and minimum count value

The first highlighted transition causes a jitter on the FTM counter value near the maximum count value (MOD). The second indicated transition occurs on phase A and causes the FTM counter transition between the maximum and minimum count values which are defined by MOD and CNTIN registers.

The appropriate settings of the phase A and phase B input filters are important to avoid glitches that may cause oscillation on the FTM counter value. The preceding figures show examples of oscillations that can be caused by poor input filter setup. Thus, it is important to guarantee a minimum pulse width to avoid these oscillations.

43.4.26 Intermediate load

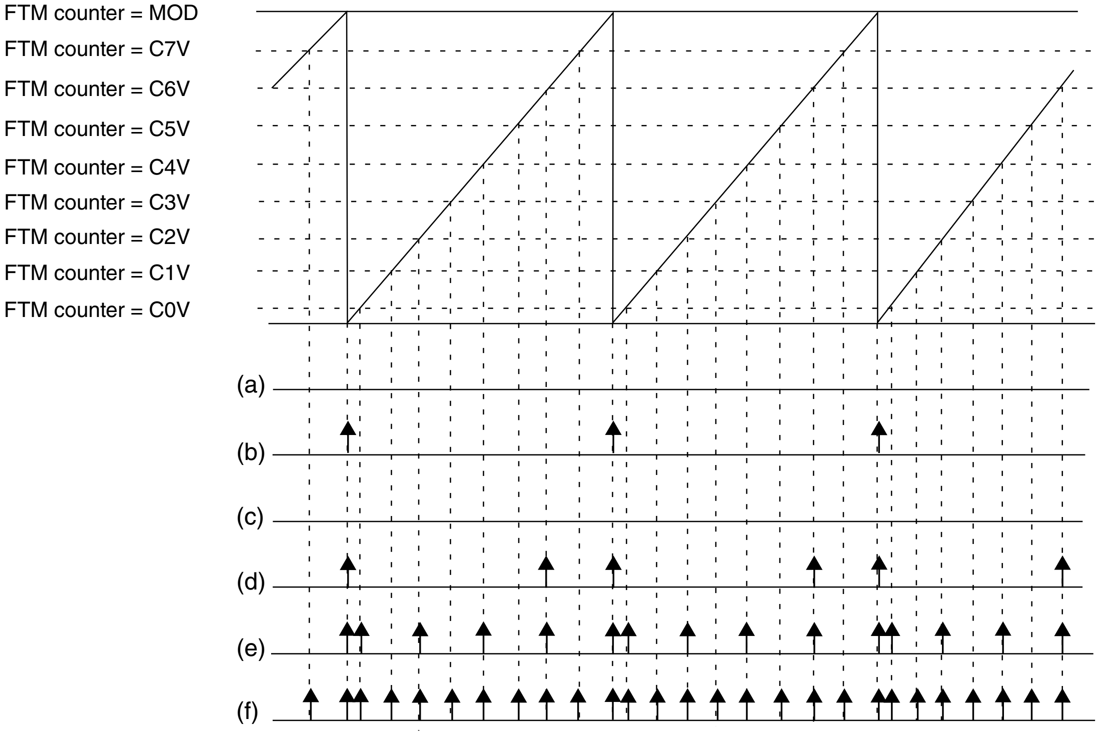
The PWMLOAD register allows to update the MOD, CNTIN, and C(n)V registers with the content of the register buffer at a defined load point. In this case, it is not required to use the PWM synchronization.

There are multiple possible loading points for intermediate load:

Table 43-15. When possible loading points are enabled

Loading point	Enabled
When the FTM counter wraps from MOD value to CNTIN value	Always
At the channel (j) match (FTM counter = C(j)V)	When CHjSEL = 1

The following figure shows some examples of enabled loading points.



NOTE

(a) LDOK = 0, CH0SEL = 0, CH1SEL = 0, CH2SEL = 0, CH3SEL = 0, CH4SEL = 0, CH5SEL = 0, CH6SEL = 0, CH7SEL = 0
(b) LDOK = 1, CH0SEL = 0, CH1SEL = 0, CH2SEL = 0, CH3SEL = 0, CH4SEL = 0, CH5SEL = 0, CH6SEL = 0, CH7SEL = 0
(c) LDOK = 0, CH0SEL = 0, CH1SEL = 0, CH2SEL = 0, CH3SEL = 1, CH4SEL = 0, CH5SEL = 0, CH6SEL = 0, CH7SEL = 0
(d) LDOK = 1, CH0SEL = 0, CH1SEL = 0, CH2SEL = 0, CH3SEL = 0, CH4SEL = 0, CH5SEL = 0, CH6SEL = 1, CH7SEL = 0
(e) LDOK = 1, CH0SEL = 1, CH1SEL = 0, CH2SEL = 1, CH3SEL = 0, CH4SEL = 1, CH5SEL = 0, CH6SEL = 1, CH7SEL = 0
(f) LDOK = 1, CH0SEL = 1, CH1SEL = 1, CH2SEL = 1, CH3SEL = 1, CH4SEL = 1, CH5SEL = 1, CH6SEL = 1, CH7SEL = 1

Figure 43-94. Loading points for intermediate load

After enabling the loading points, the LDOK bit must be set for the load to occur. In this case, the load occurs at the next enabled loading point according to the following conditions:

Table 43-16. Conditions for loads occurring at the next enabled loading point

When a new value was written	Then
To the MOD register	The MOD register is updated with its write buffer value.
To the CNTIN register and CNTINC = 1	The CNTIN register is updated with its write buffer value.
To the C(n)V register and SYNCENm = 1 – where m indicates the pair channels (n) and (n+1)	The C(n)V register is updated with its write buffer value.
To the C(n+1)V register and SYNCENm = 1 – where m indicates the pair channels (n) and (n+1)	The C(n+1)V register is updated with its write buffer value.

NOTE

- If ELSjB and ELSjA bits are different from zero, then the channel (j) output signal is generated according to the configured output mode. If ELSjB and ELSjA bits are zero, then the generated signal is not available on channel (j) output.

- If CHjIE = 1, then the channel (j) interrupt is generated when the channel (j) match occurs.
- At the intermediate load neither the channels outputs nor the FTM counter are changed. Software must set the intermediate load at a safe point in time.

43.4.27 Global time base (GTB)

The global time base (GTB) is a FTM function that allows the synchronization of multiple FTM modules on a chip. The following figure shows an example of the GTB feature used to synchronize two FTM modules. In this case, the FTM A and B channels can behave as if just one FTM module was used, that is, a global time base.

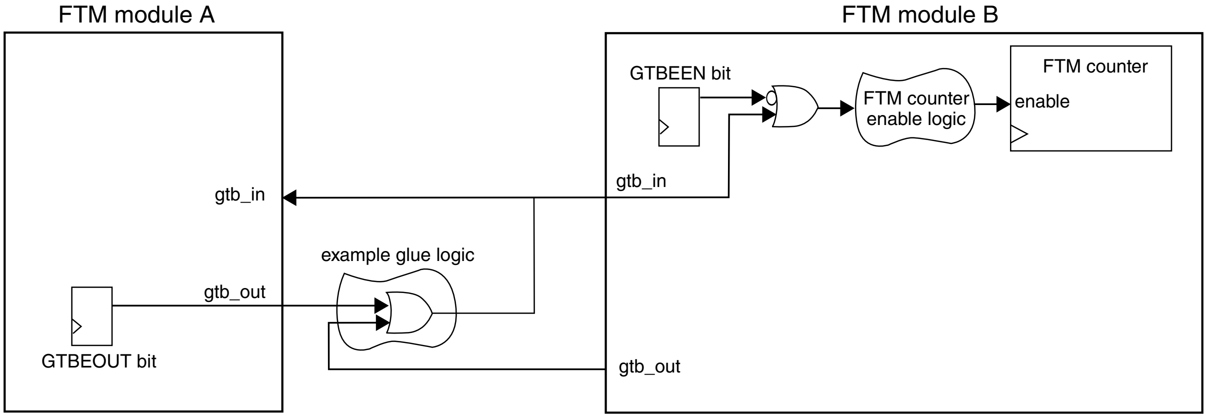


Figure 43-95. Global time base (GTB) block diagram

The GTB functionality is implemented by the GTBEEN and GTBEOUT bits in the CONF register, the input signal *gtb_in*, and the output signal *gtb_out*. The GTBEEN bit enables gtb_in to control the FTM counter enable signal:

- If GTBEEN = 0, each one of FTM modules works independently according to their configured mode.
- If GTBEEN = 1, the FTM counter update is enabled only when gtb_in is 1.

In the configuration described in the preceding figure, FTM modules A and B have their FTM counters enabled if at least one of the gtb_out signals from one of the FTM modules is 1. There are several possible configurations for the interconnection of the gtb_in and gtb_out signals, represented by the example glue logic shown in the figure. Note that these configurations are chip-dependent and implemented outside of the FTM modules. See the chip-specific FTM information for the chip's specific implementation.

NOTE

- In order to use the GTB signals to synchronize the FTM counter of different FTM modules, the configuration of each FTM module should guarantee that its FTM counter starts counting as soon as the gtb_in signal is 1.
- The GTB feature does not provide continuous synchronization of FTM counters, meaning that the FTM counters may lose synchronization during FTM operation. The GTB feature only allows the FTM counters to *start* their operation synchronously.

43.4.27.1 Enabling the global time base (GTB)

To enable the GTB feature, follow these steps for each participating FTM module:

1. Stop the FTM counter: Write 00b to SC[CLKS].
2. Program the FTM to the intended configuration. The FTM counter mode needs to be consistent across all participating modules.
3. Write 1 to CONF[GTBEEN] and write 0 to CONF[GTBEOUT] at the same time.
4. Select the intended FTM counter clock source in SC[CLKS]. The clock source needs to be consistent across all participating modules.
5. Reset the FTM counter: Write any value to the CNT register.

To initiate the GTB feature in the configuration described in the preceding figure, write 1 to CONF[GTBEOUT] in the FTM module used as the time base.

43.5 Reset overview

The FTM is reset whenever any chip reset occurs.

When the FTM exits from reset:

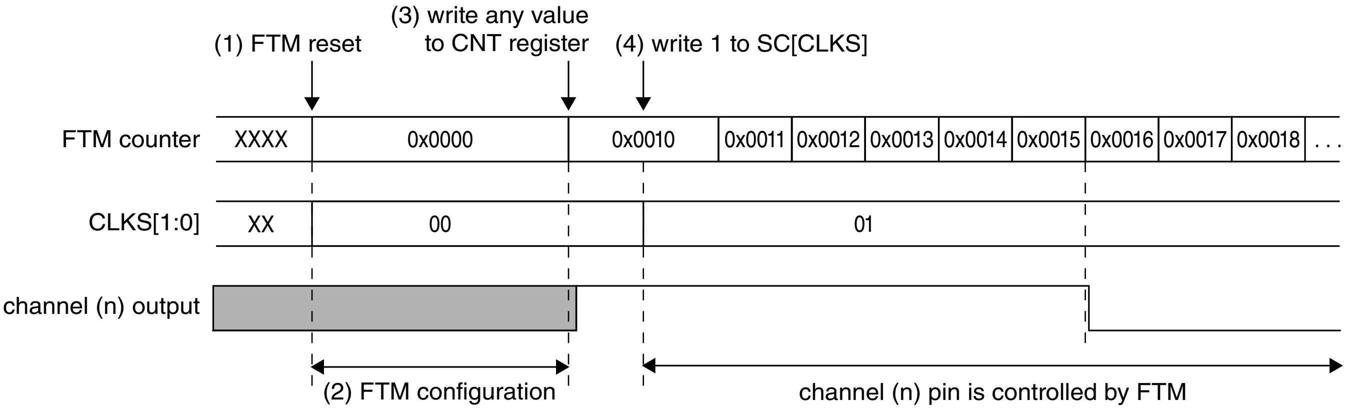
- the FTM counter and the prescaler counter are zero and are stopped (CLKS[1:0] = 00b);
- the timer overflow interrupt is zero, see Timer Overflow Interrupt;
- the channels interrupts are zero, see Channel (n) Interrupt;
- the fault interrupt is zero, see Fault Interrupt;
- the channels are in input capture mode, see Input Capture mode;
- the channels outputs are zero;
- the channels pins are not controlled by FTM (ELS(n)B:ELS(n)A = 0:0) (See the table in the description of CnSC register).

The following figure shows the FTM behavior after the reset. At the reset (item 1), the FTM counter is disabled (see the description of the CLKS field in the Status and Control register), its value is updated to zero and the pins are not controlled by FTM (See the table in the description of CnSC register).

After the reset, the FTM should be configurated (item 2). It is necessary to define the FTM counter mode, the FTM counting limits (MOD and CNTIN registers value), the channels mode and CnV registers value according to the channels mode.

Thus, it is recommended to write any value to CNT register (item 3). This write updates the FTM counter with the CNTIN register value and the channels output with its initial value (except for channels in output compare mode) (Counter reset).

The next step is to select the FTM counter clock by the CLKS[1:0] bits (item 4). It is important to highlight that the pins are only controlled by FTM when CLKS[1:0] bits are different from zero (See the table in the description of CnSC register).



NOTES:
- CNTIN = 0x0010
- Channel (n) is in low-true combine mode with CNTIN < C(n)V < C(n+1)V < MOD
- C(n)V = 0x0015

Figure 43-96. FTM behavior after reset when the channel (n) is in Combine mode

The following figure shows an example when the channel (n) is in Output Compare mode and the channel (n) output is toggled when there is a match. In the Output Compare mode, the channel output is not updated to its initial value when there is a write to CNT register (item 3). In this case, use the software output control (Software output control) or the initialization (Initialization) to update the channel output to the selected value (item 4).

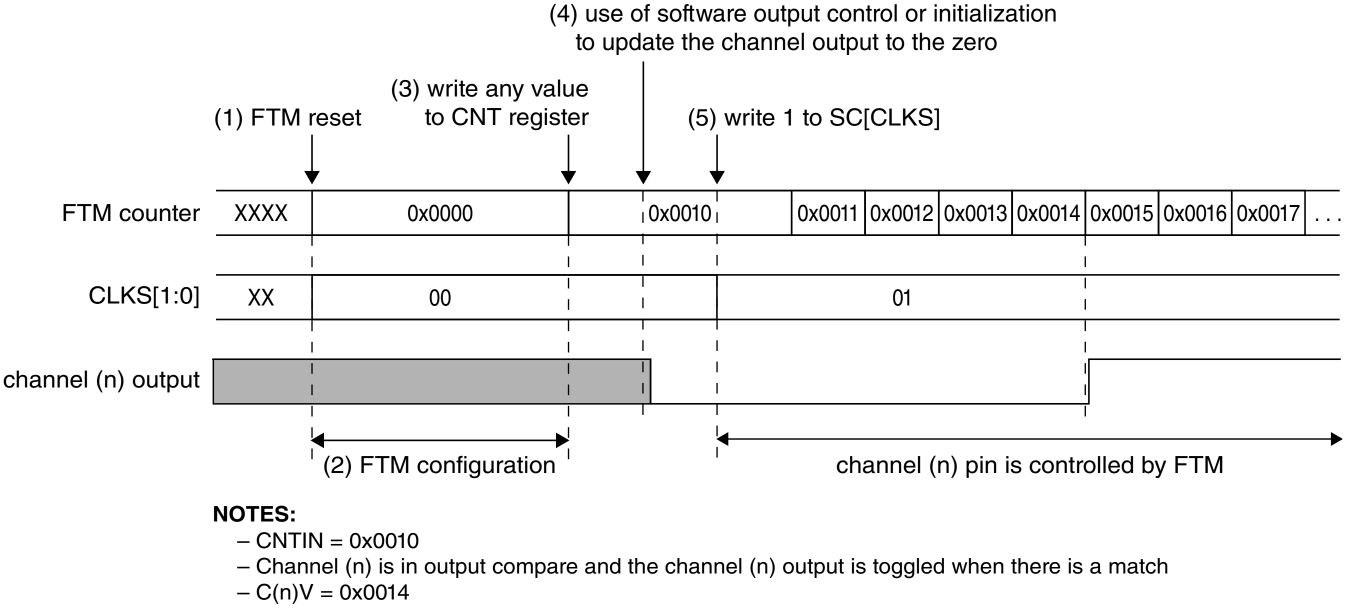

(4) use of software output control or initialization to update the channel output to the zero
(3) write any value to CNT register
(1) FTM reset
(5) write 1 to SC[CLKS]
FTM counter
XXXX
0x0000
0x0010
0x0011
0x0012
0x0013
0x0014
0x0015
0x0016
0x0017
. . .
CLKS[1:0]
XX
00
01
channel (n) output
(2) FTM configuration
channel (n) pin is controlled by FTM
NOTES:
– CNTIN = 0x0010
– Channel (n) is in output compare and the channel (n) output is toggled when there is a match
– C(n)V = 0x0014

Figure 43-97. FTM behavior after reset when the channel (n) is in Output Compare mode

43.6 FTM Interrupts

43.6.1 Timer Overflow Interrupt

The timer overflow interrupt is generated when (TOIE = 1) and (TOF = 1).

43.6.2 Channel (n) Interrupt

The channel (n) interrupt is generated when (CHnIE = 1) and (CHnF = 1).

43.6.3 Fault Interrupt

The fault interrupt is generated when (FAULTIE = 1) and (FAULTF = 1).

43.7 Initialization Procedure

The following initialization procedure is recommended to configure the FlexTimer operation. This procedure can also be used to do a new configuration of the FlexTimer operation.

- Define the POL bits.
- Mask the channels outputs using SYNCHOM = 0. Two clocks after the write to OUTMASK, the channels output are in the safe value.
- (Re)Configuration FTM counter and channels to generation of periodic signals - Disable the clock. If the selected mode is Quadrature Decoder, then disable this mode. Examples of the (re)configuration:
  - Write to MOD.
  - Write to CNTIN.
  - Select OC, EPWM, CPWM, Combine, Complement modes for all channels that will be used
  - Select the high-true and low-true channels modes.
  - Write to CnV for all channels that will be used .
  - (Re)Configure deadtime and fault control.
  - Do not use the SWOC without SW synchronization (see item 6).
  - Do not use the Inverting without SW synchronization (see item 6).
  - Do not use the Initialization.
  - Do not change the polarity control.
  - Do not configure the HW synchronization
- Write any value to CNT. The FTM Counter is reset and the channels output are updated according to new configuration.
- Enable the clock. Write to CLKS[1:0] bits a value different from zero. If in the Quadrature Decoder mode, enable this mode.
- Configure the SW synchronization for SWOC (if it is necessary), Inverting (if it is necessary) and Output Mask (always)
  - Select synchronization for Output Mask Write to SYNC (SWSYNC = 0, TRIG2 = 0, TRIG1 = 0, TRIG0 = 0, SYNCHOM = 1, REINIT = 0, CNTMAX = 0, CNTMIN = 0)
  - Write to SYNCONF.
    - HW Synchronization can not be enabled (HWSOC = 0, HWINVC = 0, HWOM = 0, HWWRBUF = 0, HWRSTCNT = 0, HWTRIGMODE = 0).
    - SW Synchronization for SWOC (if it is necessary): SWSOC = [0/1] and SWOC = [0/1].
    - SW Synchronization for Inverting (if it is necessary): SWINVC = [0/1] and INVC = [0/1].
    - SW Synchronization for SWOM (always): SWOM = 1. No enable the SW Synchronization for write buffers (because the writes to registers with write buffer are done using CLKS[1:0] = 2’b00): SWWRBUF = 0 and CNTINC = 0 .
    - SW Synchronization for counter reset (always): SWRSTCNT = 1.
    - Enhanced synchronization (always): SYNCMODE = 1

  - If the SWOC is used (SWSOC = 1 and SWOC = 1), then write to SWOCTRL register.
  - If the Inverting is used (SWINVC = 1 and INVC = 1), then write to INVCTRL register.
  - Write to OUTMASK to enable the masked channels.
- Generate the Software Trigger Write to SYNC (SWSYNC = 1, TRIG2 = 0, TRIG1 = 0, TRIG0 = 0, SYNCHOM = 1, REINIT = 0, CNTMAX = 0, CNTMIN = 0)

Chapter 44
MCU: Periodic Interrupt Timer (PIT)

44.1 Introduction

NOTE

> For the chip-specific implementation details of this module's instances, see the chip configuration information.

The PIT module is an array of timers that can be used to raise interrupts and trigger DMA channels.

44.1.1 Block diagram

The following figure shows the block diagram of the PIT module.

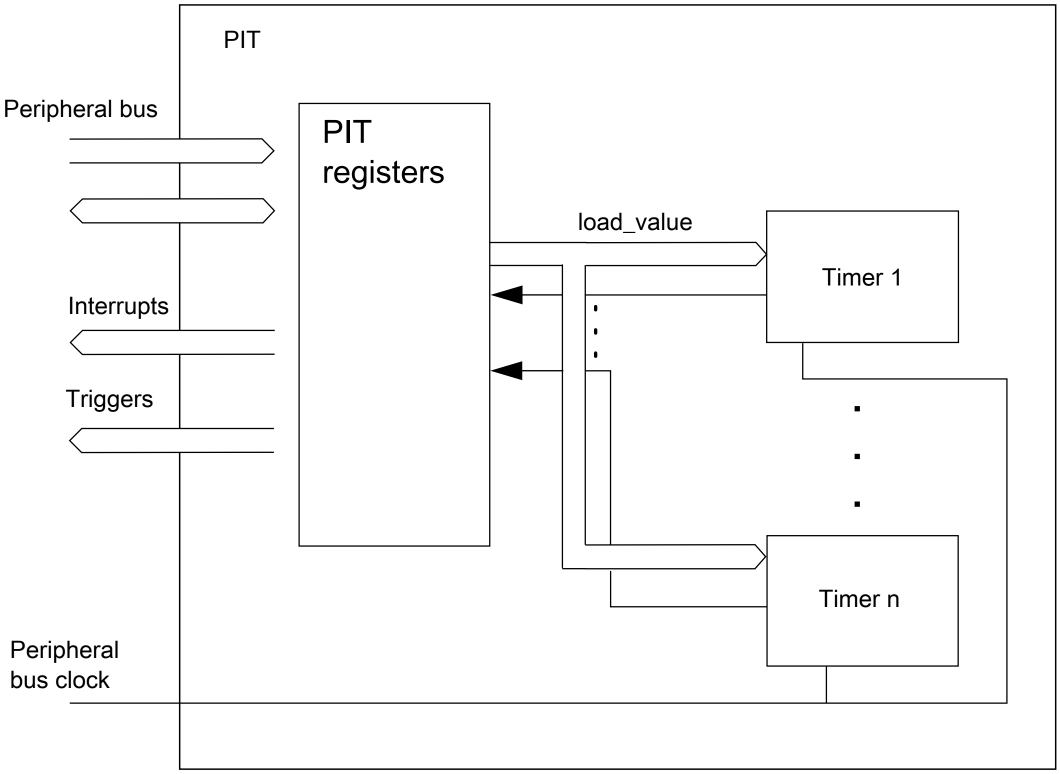


Figure 44-1. Block diagram of the PIT

> **NOTE**
>
> See the chip-specific PIT information for the number of PIT channels used in this MCU.

44.1.2 Features

The main features of this block are:

- Ability of timers to generate DMA trigger pulses
- Ability of timers to generate interrupts
- Maskable interrupts
- Independent timeout periods for each timer

44.2 Signal description

The PIT module has no external pins.

44.3 Memory map/register description

This section provides a detailed description of all registers accessible in the PIT module.

- Reserved registers will read as 0, writes will have no effect.
- See the chip-specific PIT information for the number of PIT channels used in this MCU.

PIT memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4003_7000	PIT Module Control Register (PIT_MCR)	32	R/W	0000_0006h	44.3.1/1099
4003_7100	Timer Load Value Register (PIT_LDVAL0)	32	R/W	0000_0000h	44.3.2/1101
4003_7104	Current Timer Value Register (PIT_CVAL0)	32	R	0000_0000h	44.3.3/1101
4003_7108	Timer Control Register (PIT_TCTRL0)	32	R/W	0000_0000h	44.3.4/1102
4003_710C	Timer Flag Register (PIT_TFLG0)	32	R/W	0000_0000h	44.3.5/1102
4003_7110	Timer Load Value Register (PIT_LDVAL1)	32	R/W	0000_0000h	44.3.2/1101
4003_7114	Current Timer Value Register (PIT_CVAL1)	32	R	0000_0000h	44.3.3/1101
4003_7118	Timer Control Register (PIT_TCTRL1)	32	R/W	0000_0000h	44.3.4/1102
4003_711C	Timer Flag Register (PIT_TFLG1)	32	R/W	0000_0000h	44.3.5/1102
4003_7120	Timer Load Value Register (PIT_LDVAL2)	32	R/W	0000_0000h	44.3.2/1101
4003_7124	Current Timer Value Register (PIT_CVAL2)	32	R	0000_0000h	44.3.3/1101
4003_7128	Timer Control Register (PIT_TCTRL2)	32	R/W	0000_0000h	44.3.4/1102
4003_712C	Timer Flag Register (PIT_TFLG2)	32	R/W	0000_0000h	44.3.5/1102
4003_7130	Timer Load Value Register (PIT_LDVAL3)	32	R/W	0000_0000h	44.3.2/1101
4003_7134	Current Timer Value Register (PIT_CVAL3)	32	R	0000_0000h	44.3.3/1101
4003_7138	Timer Control Register (PIT_TCTRL3)	32	R/W	0000_0000h	44.3.4/1102
4003_713C	Timer Flag Register (PIT_TFLG3)	32	R/W	0000_0000h	44.3.5/1102

44.3.1 PIT Module Control Register (PIT_MCR)

This register enables or disables the PIT timer clocks and controls the timers when the PIT enters the Debug mode.

Access: User read/write

Address: 4003_7000h base + 0h offset = 4003_7000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0													Reserved	MDIS	FRZ
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0

PIT_MCR field descriptions

Field	Description
31–3 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
2 Reserved	This field is reserved.
1 MDIS	Module Disable - (PIT section) Disables the standard timers. This field must be enabled before any other setup is done. 0 Clock for standard PIT timers is enabled. 1 Clock for standard PIT timers is disabled.
0 FRZ	Freeze Allows the timers to be stopped when the device enters the Debug mode. 0 Timers continue to run in Debug mode. 1 Timers are stopped in Debug mode.

44.3.2 Timer Load Value Register (PIT_LDVAL*n*)

These registers select the timeout period for the timer interrupts.

Access: User read/write

Address: 4003_7000h base + 100h offset + (16d × i), where i=0d to 3d

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

PIT_LDVAL*n* field descriptions

Field	Description
TSV	Timer Start Value Sets the timer start value. The timer will count down until it reaches 0, then it will generate an interrupt and load this register value again. Writing a new value to this register will not restart the timer; instead the value will be loaded after the timer expires. To abort the current cycle and start a timer period with the new value, the timer must be disabled and enabled again.

44.3.3 Current Timer Value Register (PIT_CVAL*n*)

These registers indicate the current timer position.

Access: User read only

Address: 4003_7000h base + 104h offset + (16d × i), where i=0d to 3d

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

PIT_CVAL*n* field descriptions

Field	Description
TVL	Current Timer Value Represents the current timer value, if the timer is enabled. **NOTE:** • If the timer is disabled, do not use this field as its value is unreliable. • The timer uses a downcounter. The timer values are frozen in Debug mode if MCR[FRZ] is set.

44.3.4 Timer Control Register (PIT_TCTRL*n*)

These registers contain the control bits for each timer.

Access: User read/write

Address: 4003_7000h base + 108h offset + (16d × i), where i=0d to 3d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0													CHN	TIE	TEN
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PIT_TCTRL*n* field descriptions

Field	Description
31–3 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
2 CHN	Chain Mode When activated, Timer n-1 needs to expire before timer n can decrement by 1. Timer 0 cannot be chained. 0 Timer is not chained. 1 Timer is chained to previous timer. For example, for Channel 2, if this field is set, Timer 2 is chained to Timer 1.
1 TIE	Timer Interrupt Enable When an interrupt is pending, or, TFLGn[TIF] is set, enabling the interrupt will immediately cause an interrupt event. To avoid this, the associated TFLGn[TIF] must be cleared first. 0 Interrupt requests from Timer n are disabled. 1 Interrupt will be requested whenever TIF is set.
0 TEN	Timer Enable Enables or disables the timer. 0 Timer n is disabled. 1 Timer n is enabled.

44.3.5 Timer Flag Register (PIT_TFLG*n*)

These registers hold the PIT interrupt flags.

Access: User read/write

Address: 4003_7000h base + 10Ch offset + (16d × i), where i=0d to 3d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0															TIF
W																w1c
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PIT_TFLG*n* field descriptions

Field	Description
31–1 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
0 TIF	Timer Interrupt Flag Sets to 1 at the end of the timer period. Writing 1 to this flag clears it. Writing 0 has no effect. If enabled, or, when TCTRLn[TIE] = 1, TIF causes an interrupt request. 0 Timeout has not yet occurred. 1 Timeout has occurred.

44.4 Functional description

This section provides the functional description of the module.

44.4.1 General operation

This section gives detailed information on the internal operation of the module. Each timer can be used to generate trigger pulses and interrupts. Each interrupt is available on a separate interrupt line.

44.4.1.1 Timers

The timers generate triggers at periodic intervals, when enabled. The timers load the start values as specified in their LDVAL registers, count down to 0 and then load the respective start value again. Each time a timer reaches 0, it will generate a trigger pulse and set the interrupt flag.

All interrupts can be enabled or masked by setting TCTRLn[TIE]. A new interrupt can be generated only after the previous one is cleared.

If desired, the current counter value of the timer can be read via the CVAL registers.

The counter period can be restarted, by first disabling, and then enabling the timer with TCTRLn[TEN]. See the following figure.

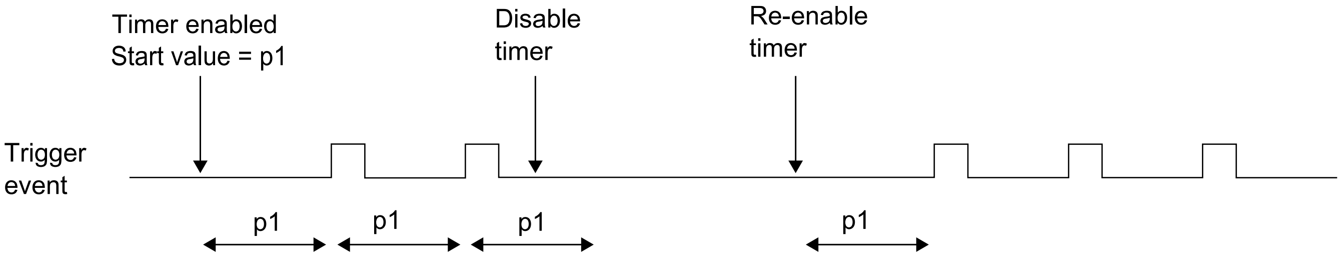


Figure 44-2. Stopping and starting a timer

The counter period of a running timer can be modified, by first disabling the timer, setting a new load value, and then enabling the timer again. See the following figure.

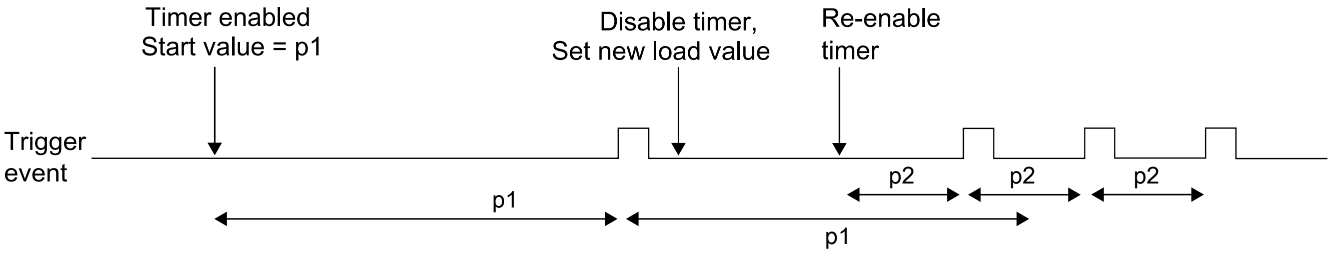


Figure 44-3. Modifying running timer period

It is also possible to change the counter period without restarting the timer by writing LDVAL with the new load value. This value will then be loaded after the next trigger event. See the following figure.

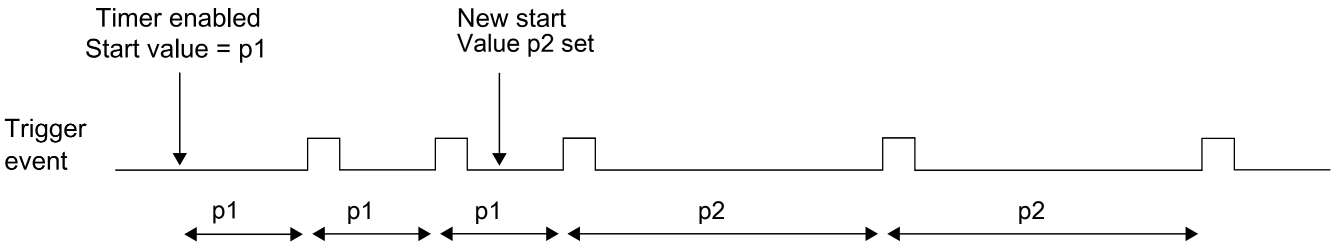


Figure 44-4. Dynamically setting a new load value

44.4.1.2 Debug mode

In Debug mode, the timers will be frozen based on MCR[FRZ]. This is intended to aid software development, allowing the developer to halt the processor, investigate the current state of the system, for example, the timer values, and then continue the operation.

44.4.2 Interrupts

All the timers support interrupt generation. See the MCU specification for related vector addresses and priorities.

Timer interrupts can be enabled by setting TCTRLn[TIE]. TFLGn[TIF] are set to 1 when a timeout occurs on the associated timer, and are cleared to 0 by writing a 1 to the corresponding TFLGn[TIF].

44.4.3 Chained timers

When a timer has chain mode enabled, it will only count after the previous timer has expired. So if timer n-1 has counted down to 0, counter n will decrement the value by one. This allows to chain some of the timers together to form a longer timer. The first timer (timer 0) cannot be chained to any other timer.

44.5 Initialization and application information

In the example configuration:

- The PIT clock has a frequency of 50 MHz.
- Timer 1 creates an interrupt every 5.12 ms.
- Timer 3 creates a trigger event every 30 ms.

The PIT module must be activated by writing a 0 to MCR[MDIS].

The 50 MHz clock frequency equates to a clock period of 20 ns. Timer 1 needs to trigger every 5.12 ms/20 ns = 256,000 cycles and Timer 3 every 30 ms/20 ns = 1,500,000 cycles. The value for the LDVAL register trigger is calculated as:

LDVAL trigger = (period / clock period) -1

This means LDVAL1 and LDVAL3 must be written with 0x0003E7FF and 0x0016E35F respectively.

The interrupt for Timer 1 is enabled by setting TCTRL1[TIE]. The timer is started by writing 1 to TCTRL1[TEN].

Timer 3 shall be used only for triggering. Therefore, Timer 3 is started by writing a 1 to TCTRL3[TEN]. TCTRL3[TIE] stays at 0.

The following example code matches the described setup:

```
// turn on PIT
PIT_MCR = 0x00;


// Timer 1
PIT_LDVAL1 = 0x0003E7FF; // setup timer 1 for 256000 cycles
PIT_TCTRL1 = TIE; // enable Timer 1 interrupts
PIT_TCTRL1 |= TEN; // start Timer 1


// Timer 3
PIT_LDVAL3 = 0x0016E35F; // setup timer 3 for 1500000 cycles
PIT_TCTRL3 |= TEN; // start Timer 3
```

44.6 Example configuration for chained timers

In the example configuration:

- The PIT clock has a frequency of 100 MHz.
- Timers 1 and 2 are available.
- An interrupt shall be raised every 1 minute.

The PIT module needs to be activated by writing a 0 to MCR[MDIS].

The 100 MHz clock frequency equates to a clock period of 10 ns, so the PIT needs to count for 6000 million cycles, which is more than a single timer can do. So, Timer 1 is set up to trigger every 6 s (600 million cycles). Timer 2 is chained to Timer 1 and programmed to trigger 10 times.

The value for the LDVAL register trigger is calculated as number of cycles-1, so LDVAL1 receives the value 0x23C345FF and LDVAL2 receives the value 0x00000009.

The interrupt for Timer 2 is enabled by setting TCTRL2[TIE], the Chain mode is activated by setting TCTRL2[CHN], and the timer is started by writing a 1 to TCTRL2[TEN]. TCTRL1[TEN] needs to be set, and TCTRL1[CHN] and TCTRL1[TIE] are cleared.

The following example code matches the described setup:

```
// turn on PIT
PIT_MCR = 0x00;


// Timer 2
PIT_LDVAL2 = 0x00000009; // setup Timer 2 for 10 counts
PIT_TCTRL2 = TIE; // enable Timer 2 interrupt
PIT_TCTRL2 |= CHN; // chain Timer 2 to Timer 1
PIT_TCTRL2 |= TEN; // start Timer 2


// Timer 1
PIT_LDVAL1 = 0x23C345FF; // setup Timer 1 for 600 000 000 cycles
PIT_TCTRL1 = TEN; // start Timer 1
```

Chapter 45
MCU: Low-Power Timer (LPTMR)

45.1 Introduction

NOTE

> For the chip-specific implementation details of this module's instances, see the chip configuration information.

The low-power timer (LPTMR) can be configured to operate as a time counter with optional prescaler, or as a pulse counter with optional glitch filter, across all power modes, including the low-leakage modes. It can also continue operating through most system reset events, allowing it to be used as a time of day counter.

45.1.1 Features

The features of the LPTMR module include:

- 16-bit time counter or pulse counter with compare
  - Optional interrupt can generate asynchronous wakeup from any low-power mode
  - Hardware trigger output
  - Counter supports free-running mode or reset on compare
- Configurable clock source for prescaler/glitch filter
- Configurable input source for pulse counter
  - Rising-edge or falling-edge

45.1.2 Modes of operation

The following table describes the operation of the LPTMR module in various modes.

Table 45-1. Modes of operation

Modes	Description
Run	The LPTMR operates normally.
Wait	The LPTMR continues to operate normally and may be configured to exit the low-power mode by generating an interrupt request.
Stop	The LPTMR continues to operate normally and may be configured to exit the low-power mode by generating an interrupt request.
Low-Leakage	The LPTMR continues to operate normally and may be configured to exit the low-power mode by generating an interrupt request.
Debug	The LPTMR operates normally.

45.2 LPTMR signal descriptions

Table 45-2. LPTMR signal descriptions

Signal	I/O	Description
LPTMR0_ALT*n*	I	Pulse Counter Input pin

45.2.1 Detailed signal descriptions

Table 45-3. LPTMR interface—detailed signal descriptions

<table>
<tr><th>Signal</th><th>I/O</th><th colspan="2">Description</th></tr>
<tr><td rowspan="3">LPTMR_ALT<i>n</i></td><td rowspan="3">I</td><td colspan="2">Pulse Counter Input
The LPTMR can select one of the input pins to be used in Pulse Counter mode.</td></tr>
<tr><td>State meaning</td><td>Assertion—If configured for pulse counter mode with active-high input, then assertion causes the CNR to increment.
Deassertion—If configured for pulse counter mode with active-low input, then deassertion causes the CNR to increment.</td></tr>
<tr><td>Timing</td><td>Assertion or deassertion may occur at any time; input may assert asynchronously to the bus clock.</td></tr>
</table>

45.3 Memory map and register definition

NOTE

The LPTMR registers are reset only on a POR or LVD event. See LPTMR power and reset for more details.

LPTMR memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4004_0000	Low Power Timer Control Status Register (LPTMR0_CSR)	32	R/W	0000_0000h	45.3.1/1111
4004_0004	Low Power Timer Prescale Register (LPTMR0_PSR)	32	R/W	0000_0000h	45.3.2/1113
4004_0008	Low Power Timer Compare Register (LPTMR0_CMR)	32	R/W	0000_0000h	45.3.3/1114
4004_000C	Low Power Timer Counter Register (LPTMR0_CNR)	32	R/W	0000_0000h	45.3.4/1115

45.3.1 Low Power Timer Control Status Register (LPTMRx_CSR)

Address: 4004_0000h base + 0h offset = 4004_0000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								TCF	TIE	TPS		TPP	TFC	TMS	TEN
W									w1c							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

LPTMRx_CSR field descriptions

Field	Description
31–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7 TCF	Timer Compare Flag TCF is set when the LPTMR is enabled and the CNR equals the CMR and increments. TCF is cleared when the LPTMR is disabled or a logic 1 is written to it. 0 The value of CNR is not equal to CMR and increments. 1 The value of CNR is equal to CMR and increments.
6 TIE	Timer Interrupt Enable When TIE is set, the LPTMR Interrupt is generated whenever TCF is also set.

Table continues on the next page...

LPTMRx_CSR field descriptions (continued)

Field	Description
	0 Timer interrupt disabled. 1 Timer interrupt enabled.
5–4 TPS	Timer Pin Select Configures the input source to be used in Pulse Counter mode. TPS must be altered only when the LPTMR is disabled. The input connections vary by device. See the chip configuration details for information on the connections to these inputs. 00 Pulse counter input 0 is selected. 01 Pulse counter input 1 is selected. 10 Pulse counter input 2 is selected. 11 Pulse counter input 3 is selected.
3 TPP	Timer Pin Polarity Configures the polarity of the input source in Pulse Counter mode. TPP must be changed only when the LPTMR is disabled. 0 Pulse Counter input source is active-high, and the CNR will increment on the rising-edge. 1 Pulse Counter input source is active-low, and the CNR will increment on the falling-edge.
2 TFC	Timer Free-Running Counter When clear, TFC configures the CNR to reset whenever TCF is set. When set, TFC configures the CNR to reset on overflow. TFC must be altered only when the LPTMR is disabled. 0 CNR is reset whenever TCF is set. 1 CNR is reset on overflow.
1 TMS	Timer Mode Select Configures the mode of the LPTMR. TMS must be altered only when the LPTMR is disabled. 0 Time Counter mode. 1 Pulse Counter mode.
0 TEN	Timer Enable When TEN is clear, it resets the LPTMR internal logic, including the CNR and TCF. When TEN is set, the LPTMR is enabled. While writing 1 to this field, CSR[5:1] must not be altered. 0 LPTMR is disabled and internal logic is reset. 1 LPTMR is enabled.

45.3.2 Low Power Timer Prescale Register (LPTMRx_PSR)

Address: 4004_0000h base + 4h offset = 4004_0004h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0									PRESCALE				PBYP	PCS	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

LPTMRx_PSR field descriptions

Field	Description
31–7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
6–3 PRESCALE	Prescale Value Configures the size of the Prescaler in Time Counter mode or width of the glitch filter in Pulse Counter mode. PRESCALE must be altered only when the LPTMR is disabled. 0000 Prescaler divides the prescaler clock by 2; glitch filter does not support this configuration. 0001 Prescaler divides the prescaler clock by 4; glitch filter recognizes change on input pin after 2 rising clock edges. 0010 Prescaler divides the prescaler clock by 8; glitch filter recognizes change on input pin after 4 rising clock edges. 0011 Prescaler divides the prescaler clock by 16; glitch filter recognizes change on input pin after 8 rising clock edges. 0100 Prescaler divides the prescaler clock by 32; glitch filter recognizes change on input pin after 16 rising clock edges. 0101 Prescaler divides the prescaler clock by 64; glitch filter recognizes change on input pin after 32 rising clock edges. 0110 Prescaler divides the prescaler clock by 128; glitch filter recognizes change on input pin after 64 rising clock edges. 0111 Prescaler divides the prescaler clock by 256; glitch filter recognizes change on input pin after 128 rising clock edges. 1000 Prescaler divides the prescaler clock by 512; glitch filter recognizes change on input pin after 256 rising clock edges. 1001 Prescaler divides the prescaler clock by 1024; glitch filter recognizes change on input pin after 512 rising clock edges. 1010 Prescaler divides the prescaler clock by 2048; glitch filter recognizes change on input pin after 1024 rising clock edges. 1011 Prescaler divides the prescaler clock by 4096; glitch filter recognizes change on input pin after 2048 rising clock edges. 1100 Prescaler divides the prescaler clock by 8192; glitch filter recognizes change on input pin after 4096 rising clock edges. 1101 Prescaler divides the prescaler clock by 16,384; glitch filter recognizes change on input pin after 8192 rising clock edges.

Table continues on the next page...

LPTMRx_PSR field descriptions (continued)

Field	Description
	1110 Prescaler divides the prescaler clock by 32,768; glitch filter recognizes change on input pin after 16,384 rising clock edges. 1111 Prescaler divides the prescaler clock by 65,536; glitch filter recognizes change on input pin after 32,768 rising clock edges.
2 PBYP	Prescaler Bypass When PBYP is set, the selected prescaler clock in Time Counter mode or selected input source in Pulse Counter mode directly clocks the CNR. When PBYP is clear, the CNR is clocked by the output of the prescaler/glitch filter. PBYP must be altered only when the LPTMR is disabled. 0 Prescaler/glitch filter is enabled. 1 Prescaler/glitch filter is bypassed.
PCS	Prescaler Clock Select Selects the clock to be used by the LPTMR prescaler/glitch filter. PCS must be altered only when the LPTMR is disabled. The clock connections vary by device. **NOTE:** See the chip configuration details for information on the connections to these inputs. 00 Prescaler/glitch filter clock 0 selected. 01 Prescaler/glitch filter clock 1 selected. 10 Prescaler/glitch filter clock 2 selected. 11 Prescaler/glitch filter clock 3 selected.

45.3.3 Low Power Timer Compare Register (LPTMRx_CMR)

Address: 4004_0000h base + 8h offset = 4004_0008h

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

LPTMRx_CMR field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
COMPARE	Compare Value When the LPTMR is enabled and the CNR equals the value in the CMR and increments, TCF is set and the hardware trigger asserts until the next time the CNR increments. If the CMR is 0, the hardware trigger will remain asserted until the LPTMR is disabled. If the LPTMR is enabled, the CMR must be altered only when TCF is set.

45.3.4 Low Power Timer Counter Register (LPTMRx_CNR)

NOTE

See LPTMR counter for details on how to read counter value.

Address: 4004_0000h base + Ch offset = 4004_000Ch

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

LPTMRx_CNR field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
COUNTER	Counter Value

45.4 Functional description

45.4.1 LPTMR power and reset

The LPTMR remains powered in all power modes, including low-leakage modes. If the LPTMR is not required to remain operating during a low-power mode, then it must be disabled before entering the mode.

The LPTMR is reset only on global Power On Reset (POR) or Low Voltage Detect (LVD). When configuring the LPTMR registers, the CSR must be initially written with the timer disabled, before configuring the PSR and CMR. Then, CSR[TIE] must be set as the last step in the initialization. This ensures the LPTMR is configured correctly and the LPTMR counter is reset to zero following a warm reset.

45.4.2 LPTMR clocking

The LPTMR prescaler/glitch filter can be clocked by one of the four clocks. The clock source must be enabled before the LPTMR is enabled.

NOTE

The clock source selected need to be configured to remain enabled in low-power modes, otherwise the LPTMR will not operate during low-power modes.

In Pulse Counter mode with the prescaler/glitch filter bypassed, the selected input source directly clocks the CNR and no other clock source is required. To minimize power in this case, configure the prescaler clock source for a clock that is not toggling.

NOTE

The clock source or pulse input source selected for the LPTMR should not exceed the frequency f_{LPTMR} defined in the device datasheet.

45.4.3 LPTMR prescaler/glitch filter

The LPTMR prescaler and glitch filter share the same logic which operates as a prescaler in Time Counter mode and as a glitch filter in Pulse Counter mode.

NOTE

The prescaler/glitch filter configuration must not be altered when the LPTMR is enabled.

45.4.3.1 Prescaler enabled

In Time Counter mode, when the prescaler is enabled, the output of the prescaler directly clocks the CNR. When the LPTMR is enabled, the CNR will increment every 2^2 to 2^{16} prescaler clock cycles. After the LPTMR is enabled, the first increment of the CNR will take an additional one or two prescaler clock cycles due to synchronization logic.

45.4.3.2 Prescaler bypassed

In Time Counter mode, when the prescaler is bypassed, the selected prescaler clock increments the CNR on every clock cycle. When the LPTMR is enabled, the first increment will take an additional one or two prescaler clock cycles due to synchronization logic.

45.4.3.3 Glitch filter

In Pulse Counter mode, when the glitch filter is enabled, the output of the glitch filter directly clocks the CNR. When the LPTMR is first enabled, the output of the glitch filter is asserted, that is, logic 1 for active-high and logic 0 for active-low. The following table shows the change in glitch filter output with the selected input source.

If	Then
The selected input source remains deasserted for at least 2^1 to 2^{15} consecutive prescaler clock rising edges	The glitch filter output will also deassert.
The selected input source remains asserted for at least 2^1 to 2^{15} consecutive prescaler clock rising-edges	The glitch filter output will also assert.

NOTE

The input is only sampled on the rising clock edge.

The CNR will increment each time the glitch filter output asserts. In Pulse Counter mode, the maximum rate at which the CNR can increment is once every 2^2 to 2^{16} prescaler clock edges. When first enabled, the glitch filter will wait an additional one or two prescaler clock edges due to synchronization logic.

45.4.3.4 Glitch filter bypassed

In Pulse Counter mode, when the glitch filter is bypassed, the selected input source increments the CNR every time it asserts. Before the LPTMR is first enabled, the selected input source is forced to be asserted. This prevents the CNR from incrementing if the selected input source is already asserted when the LPTMR is first enabled.

45.4.4 LPTMR compare

When the CNR equals the value of the CMR and increments, the following events occur:

- CSR[TCF] is set.
- LPTMR interrupt is generated if CSR[TIE] is also set.
- LPTMR hardware trigger is generated.
- CNR is reset if CSR[TFC] is clear.

When the LPTMR is enabled, the CMR can be altered only when CSR[TCF] is set. When updating the CMR, the CMR must be written and CSR[TCF] must be cleared before the LPTMR counter has incremented past the new LPTMR compare value.

45.4.5 LPTMR counter

The CNR increments by one on every:

- Prescaler clock in Time Counter mode with prescaler bypassed
- Prescaler output in Time Counter mode with prescaler enabled
- Input source assertion in Pulse Counter mode with glitch filter bypassed
- Glitch filter output in Pulse Counter mode with glitch filter enabled

The CNR is reset when the LPTMR is disabled or if the counter register overflows. If CSR[TFC] is cleared, then the CNR is also reset whenever CSR[TCF] is set.

The CNR continues incrementing when the core is halted in Debug mode.

The CNR cannot be initialized, but can be read at any time. On each read of the CNR, software must first write to the CNR with any value. This will synchronize and register the current value of the CNR into a temporary register. The contents of the temporary register are returned on each read of the CNR.

When reading the CNR, the bus clock must be at least two times faster than the rate at which the LPTMR counter is incrementing, otherwise incorrect data may be returned.

45.4.6 LPTMR hardware trigger

The LPTMR hardware trigger asserts at the same time the CSR[TCF] is set and can be used to trigger hardware events in other peripherals without software intervention. The hardware trigger is always enabled.

When	Then
The CMR is set to 0 with CSR[TFC] clear	The LPTMR hardware trigger will assert on the first compare and does not deassert.
The CMR is set to a nonzero value, or, if CSR[TFC] is set	The LPTMR hardware trigger will assert on each compare and deassert on the following increment of the CNR.

45.4.7 LPTMR interrupt

The LPTMR interrupt is generated whenever CSR[TIE] and CSR[TCF] are set. CSR[TCF] is cleared by disabling the LPTMR or by writing a logic 1 to it.

CSR[TIE] can be altered and CSR[TCF] can be cleared while the LPTMR is enabled.

The LPTMR interrupt is generated asynchronously to the system clock and can be used to generate a wakeup from any low-power mode, including the low-leakage modes, provided the LPTMR is enabled as a wakeup source.

Chapter 46
MCU: Carrier Modulator Transmitter (CMT)

46.1 Introduction

NOTE

> For the chip-specific implementation details of this module's instances, see the chip configuration information.

The carrier modulator transmitter (CMT) module provides the means to generate the protocol timing and carrier signals for a wide variety of encoding schemes. The CMT incorporates hardware to off-load the critical and/or lengthy timing requirements associated with signal generation from the CPU, releasing much of its bandwidth to handle other tasks such as:

- Code data generation
- Data decompression, or,
- Keyboard scanning

The CMT does not include dedicated hardware configurations for specific protocols, but is intended to be sufficiently programmable in its function to handle the timing requirements of most protocols with minimal CPU intervention.

When the modulator is disabled, certain CMT registers can be used to change the state of the infrared output (IRO) signal directly. This feature allows for the generation of future protocol timing signals not readily producible by the current architecture.

46.2 Features

The features of this module include:

- Four modes of operation:
  - Time; with independent control of high and low times

  - Baseband
  - Frequency-shift key (FSK)
  - Direct software control of the IRO signal
- Extended space operation in Time, Baseband, and FSK modes
- Selectable input clock divider
- Interrupt on end-of-cycle
  - Ability to disable the IRO signal and use as timer interrupt

46.3 Block diagram

The following figure presents the block diagram of the CMT module.

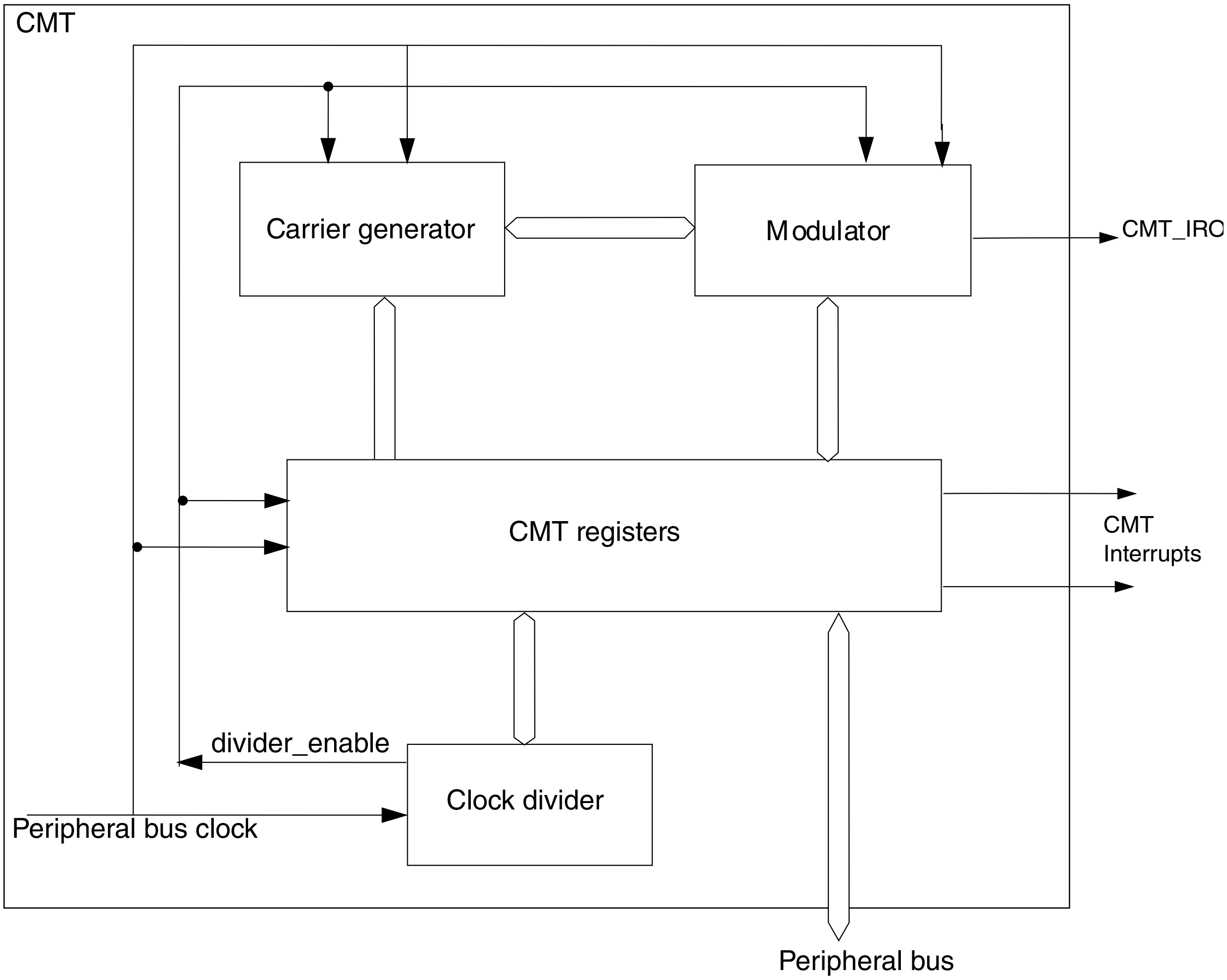


Figure 46-1. CMT module block diagram

46.4 Modes of operation

The following table describes the operation of the CMT module operates in various modes.

Table 46-1. Modes of operation

Modes	Description
Time	In Time mode, the user independently defines the high and low times of the carrier signal to determine both period and duty cycle
Baseband	When MSC[BASE] is set, the carrier output (f_{cg}) to the modulator is held high continuously to allow for the generation of baseband protocols.

Table continues on the next page...

Table 46-1. Modes of operation (continued)

Modes	Description
Frequency-shift key	This mode allows the carrier generator to alternate between two sets of high and low times. When operating in FSK mode, the generator will toggle between the two sets when instructed by the modulator, allowing the user to dynamically switch between two carrier frequencies without CPU intervention.

The following table summarizes the modes of operation of the CMT module.

Table 46-2. CMT modes of operation

Mode	MSC[MCGEN][1]	MSC[BASE]	MSC[FSK][2]	MSC[EXSPC]	Comment
Time	1	0	0	0	f_{cg} controlled by primary high and low registers. f_{cg} transmitted to the IRO signal when modulator gate is open.
Baseband	1	1	X	0	f_{cg} is always high. The IRO signal is high when the modulator gate is open.
FSK	1	0	1	0	f_{cg} control alternates between primary high/low registers and secondary high/low registers. f_{cg} transmitted to the IRO signal when modulator gate is open.
Extended Space	1	X	X	1	Setting MSC[EXSPC] causes subsequent modulator cycles to be spaces (modulator out not asserted) for the duration of the modulator period (mark and space times).
IRO Latch	0	X	X	X	OC[IROL] controls the state of the IRO signal.

[1] To prevent spurious operation, initialize all data and control registers before beginning a transmission when MSC[MCGEN]=1.

[2] This field is not double-buffered and must not be changed during a transmission while MSC[MCGEN]=1.

NOTE

> The assignment of module modes to core modes is chip-specific. For module-to-core mode assignments, see the chapter that describes how modules are configured.

46.4.1 Wait mode operation

During Wait mode, the CMT if enabled, will continue to operate normally. However, there is no change in operating modes of CMT during Wait mode, because the CPU is not operating.

46.4.2 Stop mode operation

This section describes the CMT Stop mode operations.

46.4.2.1 Normal Stop mode operation

During Normal Stop mode, clocks to the CMT module are halted. No registers are affected.

The CMT module will resume upon exit from Normal Stop mode because the clocks are halted. Software must ensure that the Normal Stop mode is not entered while the modulator is still in operation so as to prevent the IRO signal from being asserted while in Normal Stop mode. This may require a timeout period from the time that MSC[MCGEN] is cleared to allow the last modulator cycle to complete.

46.4.2.2 Low-Power Stop mode operation

During Low-Power Stop mode, the CMT module is completely powered off internally and the IRO signal state is latched and held at the time when the CMT enters this mode. To prevent the IRO signal from being asserted during Low-Power Stop mode, the software must assure that the signal is not active when entering Low-Power Stop mode. Upon wakeup from Low-Power Stop mode, the CMT module will be in the reset state.

46.5 CMT external signal descriptions

The following table shows the description of the external signal.

Table 46-3. CMT signal description

Signal	Description	I/O
CMT_IRO	Infrared Output	O

46.5.1 CMT_IRO — Infrared Output

This output signal is driven by the modulator output when MSC[MCGEN] and OC[IROPEN] are set. The IRO signal starts a valid transmission with a delay, after MSC[MCGEN] bit be asserted to high, that can be calculated based on two register bits. Table 46-4 shows how to calculate this delay.

The following table describes conditions for the IRO signal to be active.

If	Then
MSC[MCGEN] is cleared and OC[IROPEN] is set	The signal is driven by OC[IROL] . This enables user software to directly control the state of the IRO signal by writing to OC[IROL] .
OC[IROPEN] is cleared	The signal is disabled and is not driven by the CMT module. Therefore, CMT can be configured as a modulo timer for generating periodic interrupts without causing signal activity.

Table 46-4. CMT_IRO signal delay calculation

Condition	Delay (bus clock cycles)
MSC[CMTDIV] = 0	PPS[PPSDIV] + 2
MSC[CMTDIV] > 0	(PPS[PPSDIV] *2) + 3

46.6 Memory map/register definition

The following registers control and monitor the CMT operation.

The address of a register is the sum of a base address and an address offset. The base address is defined at the chip level. The address offset is defined at the module level.

CMT memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4006_2000	CMT Carrier Generator High Data Register 1 (CMT_CGH1)	8	R/W	Undefined	46.6.1/1127
4006_2001	CMT Carrier Generator Low Data Register 1 (CMT_CGL1)	8	R/W	Undefined	46.6.2/1128
4006_2002	CMT Carrier Generator High Data Register 2 (CMT_CGH2)	8	R/W	Undefined	46.6.3/1128
4006_2003	CMT Carrier Generator Low Data Register 2 (CMT_CGL2)	8	R/W	Undefined	46.6.4/1129
4006_2004	CMT Output Control Register (CMT_OC)	8	R/W	00h	46.6.5/1129
4006_2005	CMT Modulator Status and Control Register (CMT_MSC)	8	R/W	00h	46.6.6/1130
4006_2006	CMT Modulator Data Register Mark High (CMT_CMD1)	8	R/W	Undefined	46.6.7/1132

Table continues on the next page...

CMT memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4006_2007	CMT Modulator Data Register Mark Low (CMT_CMD2)	8	R/W	Undefined	46.6.8/1133
4006_2008	CMT Modulator Data Register Space High (CMT_CMD3)	8	R/W	Undefined	46.6.9/1133
4006_2009	CMT Modulator Data Register Space Low (CMT_CMD4)	8	R/W	Undefined	46.6.10/ 1134
4006_200A	CMT Primary Prescaler Register (CMT_PPS)	8	R/W	00h	46.6.11/ 1134
4006_200B	CMT Direct Memory Access Register (CMT_DMA)	8	R/W	00h	46.6.12/ 1135

46.6.1 CMT Carrier Generator High Data Register 1 (CMT_CGH1)

This data register contains the primary high value for generating the carrier output.

Address: 4006_2000h base + 0h offset = 4006_2000h

Bit	7	6	5	4	3	2	1	0
Read Write	PH							
Reset	x*	x*	x*	x*	x*	x*	x*	x*

* Notes:

- x = Undefined at reset.

CMT_CGH1 field descriptions

Field	Description
PH	Primary Carrier High Time Data Value Contains the number of input clocks required to generate the carrier high time period. When operating in Time mode, this register is always selected. When operating in FSK mode, this register and the secondary register pair are alternately selected under the control of the modulator. The primary carrier high time value is undefined out of reset. This register must be written to nonzero values before the carrier generator is enabled to avoid spurious results.

46.6.2 CMT Carrier Generator Low Data Register 1 (CMT_CGL1)

This data register contains the primary low value for generating the carrier output.

Address: 4006_2000h base + 1h offset = 4006_2001h

Bit	7	6	5	4	3	2	1	0
Read Write	PL							
Reset	x*	x*	x*	x*	x*	x*	x*	x*

* Notes:

- x = Undefined at reset.

CMT_CGL1 field descriptions

Field	Description
PL	Primary Carrier Low Time Data Value Contains the number of input clocks required to generate the carrier low time period. When operating in Time mode, this register is always selected. When operating in FSK mode, this register and the secondary register pair are alternately selected under the control of the modulator. The primary carrier low time value is undefined out of reset. This register must be written to nonzero values before the carrier generator is enabled to avoid spurious results.

46.6.3 CMT Carrier Generator High Data Register 2 (CMT_CGH2)

This data register contains the secondary high value for generating the carrier output.

Address: 4006_2000h base + 2h offset = 4006_2002h

Bit	7	6	5	4	3	2	1	0
Read Write	SH							
Reset	x*	x*	x*	x*	x*	x*	x*	x*

* Notes:

- x = Undefined at reset.

CMT_CGH2 field descriptions

Field	Description
SH	Secondary Carrier High Time Data Value Contains the number of input clocks required to generate the carrier high time period. When operating in Time mode, this register is never selected. When operating in FSK mode, this register and the primary register pair are alternately selected under control of the modulator.

CMT_CGH2 field descriptions (continued)

Field	Description
	The secondary carrier high time value is undefined out of reset. This register must be written to nonzero values before the carrier generator is enabled when operating in FSK mode.

46.6.4 CMT Carrier Generator Low Data Register 2 (CMT_CGL2)

This data register contains the secondary low value for generating the carrier output.

Address: 4006_2000h base + 3h offset = 4006_2003h

Bit	7	6	5	4	3	2	1	0
Read Write	SL							
Reset	x*	x*	x*	x*	x*	x*	x*	x*

* Notes:

- x = Undefined at reset.

CMT_CGL2 field descriptions

Field	Description
SL	Secondary Carrier Low Time Data Value Contains the number of input clocks required to generate the carrier low time period. When operating in Time mode, this register is never selected. When operating in FSK mode, this register and the primary register pair are alternately selected under the control of the modulator. The secondary carrier low time value is undefined out of reset. This register must be written to nonzero values before the carrier generator is enabled when operating in FSK mode.

46.6.5 CMT Output Control Register (CMT_OC)

This register is used to control the IRO signal of the CMT module.

Address: 4006_2000h base + 4h offset = 4006_2004h

Bit	7	6	5	4	3	2	1	0
Read	IROL	CMTPOL	IROPEN	0				
Write				▨	▨	▨	▨	▨
Reset	0	0	0	0	0	0	0	0

CMT_OC field descriptions

Field	Description
7 IROL	IRO Latch Control Reads the state of the IRO latch. Writing to IROL changes the state of the IRO signal when MSC[MCGEN] is cleared and IROPEN is set.
6 CMTPOL	CMT Output Polarity Controls the polarity of the IRO signal. 0 The IRO signal is active-low. 1 The IRO signal is active-high.
5 IROPEN	IRO Pin Enable Enables and disables the IRO signal. When the IRO signal is enabled, it is an output that drives out either the CMT transmitter output or the state of IROL depending on whether MSC[MCGEN] is set or not. Also, the state of output is either inverted or non-inverted, depending on the state of CMTPOL. When the IRO signal is disabled, it is in a high-impedance state and is unable to draw any current. This signal is disabled during reset. 0 The IRO signal is disabled. 1 The IRO signal is enabled as output.
Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

46.6.6 CMT Modulator Status and Control Register (CMT_MSC)

This register contains the modulator and carrier generator enable (MCGEN), end of cycle interrupt enable (EOCIE), FSK mode select (FSK), baseband enable (BASE), extended space (EXSPC), prescaler (CMTDIV) bits, and the end of cycle (EOCF) status bit.

Address: 4006_2000h base + 5h offset = 4006_2005h

Bit	7	6	5	4	3	2	1	0
Read	EOCF	CMTDIV		EXSPC	BASE	FSK	EOCIE	MCGEN
Write								
Reset	0	0	0	0	0	0	0	0

CMT_MSC field descriptions

Field	Description
7 EOCF	End Of Cycle Status Flag Sets when:

Table continues on the next page...

CMT_MSC field descriptions (continued)

Field	Description
	• The modulator is not currently active and MCGEN is set to begin the initial CMT transmission. • At the end of each modulation cycle while MCGEN is set. This is recognized when a match occurs between the contents of the space period register and the down counter. At this time, the counter is initialized with, possibly new contents of the mark period buffer, CMD1 and CMD2, and the space period register is loaded with, possibly new contents of the space period buffer, CMD3 and CMD4. This flag is cleared by reading MSC followed by an access of CMD2 or CMD4, or by the DMA transfer. 0 End of modulation cycle has not occured since the flag last cleared. 1 End of modulator cycle has occurred.
6–5 CMTDIV	CMT Clock Divide Prescaler Causes the CMT to be clocked at the IF signal frequency, or the IF frequency divided by 2 ,4, or 8 . This field must not be changed during a transmission because it is not double-buffered. 00 IF ÷ 1 01 IF ÷ 2 10 IF ÷ 4 11 IF ÷ 8
4 EXSPC	Extended Space Enable Enables the extended space operation. 0 Extended space is disabled. 1 Extended space is enabled.
3 BASE	Baseband Enable When set, BASE disables the carrier generator and forces the carrier output high for generation of baseband protocols. When BASE is cleared, the carrier generator is enabled and the carrier output toggles at the frequency determined by values stored in the carrier data registers. This field is cleared by reset. This field is not double-buffered and must not be written to during a transmission. 0 Baseband mode is disabled. 1 Baseband mode is enabled.
2 FSK	FSK Mode Select Enables FSK operation. 0 The CMT operates in Time or Baseband mode. 1 The CMT operates in FSK mode.
1 EOCIE	End of Cycle Interrupt Enable Requests to enable a CPU interrupt when EOCF is set if EOCIE is high.

Table continues on the next page...

CMT_MSC field descriptions (continued)

Field	Description
	0 CPU interrupt is disabled. 1 CPU interrupt is enabled.
0 MCGEN	Modulator and Carrier Generator Enable Setting MCGEN will initialize the carrier generator and modulator and will enable all clocks. When enabled, the carrier generator and modulator will function continuously. When MCGEN is cleared, the current modulator cycle will be allowed to expire before all carrier and modulator clocks are disabled to save power and the modulator output is forced low. **NOTE:** To prevent spurious operation, the user should initialize all data and control registers before enabling the system. 0 Modulator and carrier generator disabled 1 Modulator and carrier generator enabled

46.6.7 CMT Modulator Data Register Mark High (CMT_CMD1)

The contents of this register are transferred to the modulator down counter upon the completion of a modulation period.

Address: 4006_2000h base + 6h offset = 4006_2006h

Bit	7	6	5	4	3	2	1	0
Read Write	MB[15:8]							
Reset	x*	x*	x*	x*	x*	x*	x*	x*

* Notes:
- x = Undefined at reset.

CMT_CMD1 field descriptions

Field	Description
MB[15:8]	MB[15:8] Controls the upper mark periods of the modulator for all modes.

46.6.8 CMT Modulator Data Register Mark Low (CMT_CMD2)

The contents of this register are transferred to the modulator down counter upon the completion of a modulation period.

Address: 4006_2000h base + 7h offset = 4006_2007h

Bit	7	6	5	4	3	2	1	0
Read Write	MB[7:0]							
Reset	x*	x*	x*	x*	x*	x*	x*	x*

* Notes:
- x = Undefined at reset.

CMT_CMD2 field descriptions

Field	Description
MB[7:0]	MB[7:0] Controls the lower mark periods of the modulator for all modes.

46.6.9 CMT Modulator Data Register Space High (CMT_CMD3)

The contents of this register are transferred to the space period register upon the completion of a modulation period.

Address: 4006_2000h base + 8h offset = 4006_2008h

Bit	7	6	5	4	3	2	1	0
Read Write	SB[15:8]							
Reset	x*	x*	x*	x*	x*	x*	x*	x*

* Notes:
- x = Undefined at reset.

CMT_CMD3 field descriptions

Field	Description
SB[15:8]	SB[15:8] Controls the upper space periods of the modulator for all modes.

46.6.10 CMT Modulator Data Register Space Low (CMT_CMD4)

The contents of this register are transferred to the space period register upon the completion of a modulation period.

Address: 4006_2000h base + 9h offset = 4006_2009h

Bit	7	6	5	4	3	2	1	0
Read Write	SB[7:0]							
Reset	x*	x*	x*	x*	x*	x*	x*	x*

* Notes:
- x = Undefined at reset.

CMT_CMD4 field descriptions

Field	Description
SB[7:0]	SB[7:0] Controls the lower space periods of the modulator for all modes.

46.6.11 CMT Primary Prescaler Register (CMT_PPS)

This register is used to set the Primary Prescaler Divider field (PPSDIV).

Address: 4006_2000h base + Ah offset = 4006_200Ah

Bit	7	6	5	4	3	2	1	0
Read	0				PPSDIV			
Write								
Reset	0	0	0	0	0	0	0	0

CMT_PPS field descriptions

Field	Description
7–4 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
PPSDIV	Primary Prescaler Divider Divides the CMT clock to generate the Intermediate Frequency clock enable to the secondary prescaler. 0000 Bus clock ÷ 1 0001 Bus clock ÷ 2 0010 Bus clock ÷ 3

Table continues on the next page...

CMT_PPS field descriptions (continued)

Field	Description
	0011 Bus clock ÷ 4 0100 Bus clock ÷ 5 0101 Bus clock ÷ 6 0110 Bus clock ÷ 7 0111 Bus clock ÷ 8 1000 Bus clock ÷ 9 1001 Bus clock ÷ 10 1010 Bus clock ÷ 11 1011 Bus clock ÷ 12 1100 Bus clock ÷ 13 1101 Bus clock ÷ 14 1110 Bus clock ÷ 15 1111 Bus clock ÷ 16

46.6.12 CMT Direct Memory Access Register (CMT_DMA)

This register is used to enable/disable direct memory access (DMA).

Address: 4006_2000h base + Bh offset = 4006_200Bh

Bit	7	6	5	4	3	2	1	0
Read	0							DMA
Write								
Reset	0	0	0	0	0	0	0	0

CMT_DMA field descriptions

Field	Description
7–1 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
0 DMA	DMA Enable Enables the DMA protocol. 0 DMA transfer request and done are disabled. 1 DMA transfer request and done are enabled.

46.7 Functional description

The CMT module primarily consists of clock divider, carrier generator, and modulator.

46.7.1 Clock divider

The CMT was originally designed to be based on an 8 MHz bus clock that could be divided by 1, 2, 4, or 8 according to the specification. To be compatible with higher bus frequency, the primary prescaler (PPS) was developed to receive a higher frequency and generate a clock enable signal called intermediate frequency (IF). This IF must be approximately equal to 8 MHz and will work as a clock enable to the secondary prescaler. The following figure shows the clock divider block diagram.

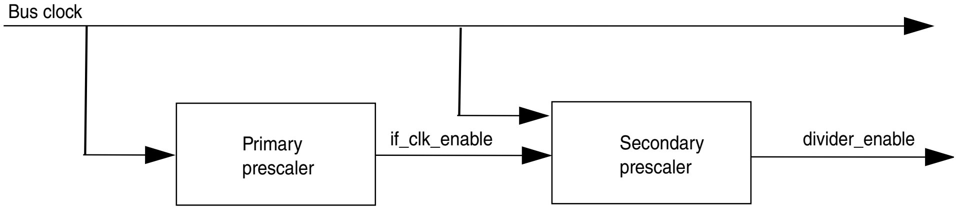


Figure 46-2. Clock divider block diagram

For compatibility with previous versions of CMT, when bus clock = 8 MHz, the PPS must be configured to zero. The PPS counter is selected according to the bus clock to generate an intermediate frequency approximately equal to 8 MHz.

46.7.2 Carrier generator

The carrier generator resolution is 125 ns when operating with an 8 MHz intermediate frequency signal and the secondary prescaler is set to divide by 1, or, when MSC[CMTDIV] = 00. The carrier generator can generate signals with periods between 250 ns (4 MHz) and 127.5 µs (7.84 kHz) in steps of 125 ns. The following table shows the relationship between the clock divide bits and the carrier generator resolution, minimum carrier generator period, and minimum modulator period.

Table 46-5. Clock divider

Bus clock (MHz)	MSC[CMTDIV]	Carrier generator resolution (µs)	Min. carrier generator period (µs)	Min. modulator period (µs)
8	00	0.125	0.25	1.0
8	01	0.25	0.5	2.0
8	10	0.5	1.0	4.0
8	11	1.0	2.0	8.0

The possible duty cycle options depend upon the number of counts required to complete the carrier period. For example, 1.6 MHz signal has a period of 625 ns and will therefore require 5 x 125 ns counts to generate. These counts may be split between high and low times, so the duty cycles available will be:

- 20% with one high and four low times
- 40% with two high and three low times
- 60% with three high and two low times, and
- 80% with four high and one low time

.

For low-frequency signals with large periods, high-resolution duty cycles as a percentage of the total period, are possible.

The carrier signal is generated by counting a register-selected number of input clocks (125 ns for an 8 MHz bus) for both the carrier high time and the carrier low time. The period is determined by the total number of clocks counted. The duty cycle is determined by the ratio of high-time clocks to total clocks counted. The high and low time values are user-programmable and are held in two registers.

An alternate set of high/low count values is held in another set of registers to allow the generation of dual-frequency FSK protocols without CPU intervention.

Note

> Only nonzero data values are allowed. The carrier generator will not work if any of the count values are equal to zero.

MSC[MCGEN] must be set and MSC[BASE] must be cleared to enable carrier generator clocks. When MSC[BASE] is set, the carrier output to the modulator is held high continuously. The following figure represents the block diagram of the clock generator.

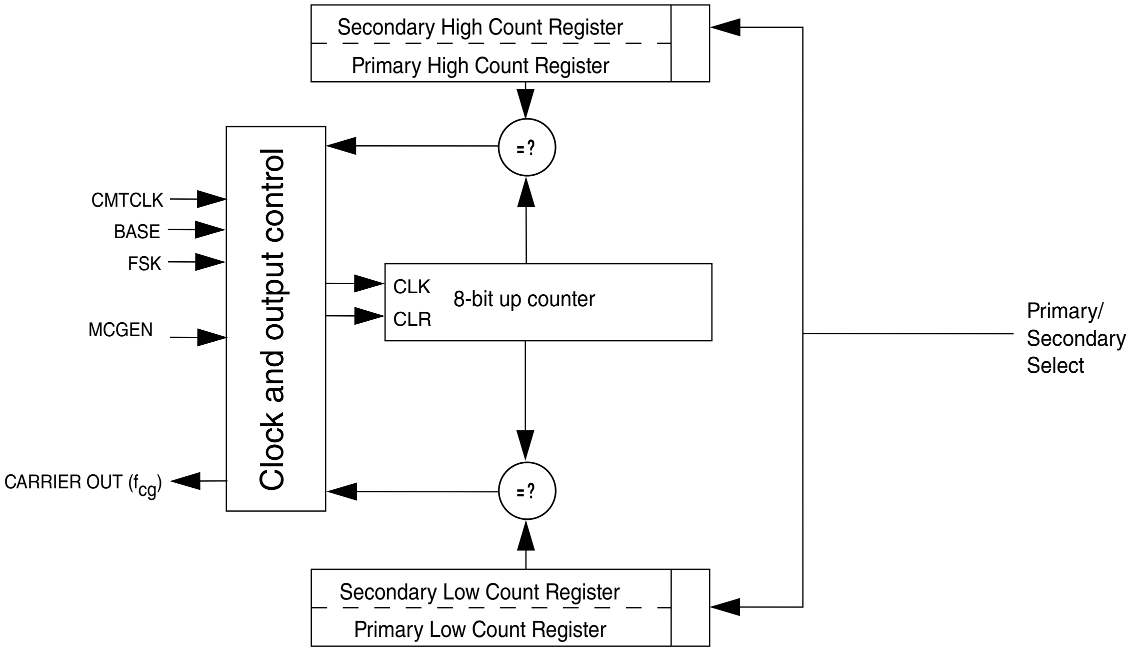


Figure 46-3. Carrier generator block diagram

The high/low time counter is an 8-bit up counter. After each increment, the contents of the counter are compared with the appropriate high or low count value register. When the compare value is reached, the counter is reset to a value of 0x01, and the compare is redirected to the other count value register.

Assuming that the high time count compare register is currently active, a valid compare will cause the carrier output to be driven low. The counter will continue to increment starting at the reset value of 0x01. When the value stored in the selected low count value register is reached, the counter will again be reset and the carrier output will be driven high.

The cycle repeats, automatically generating a periodic signal which is directed to the modulator. The lower frequency with maximum period, f_{max}, and highest frequency with minimum period, f_{min}, which can be generated, are defined as:

$f_{max} = f_{CMTCLK} \div (2 * 1)$ Hz

$f_{min} = f_{CMTCLK} \div (2 * (2^8 - 1))$ Hz

In the general case, the carrier generator output frequency is:

$f_{cg} = f_{CMTCLK} \div (\text{High count} + \text{Low count})$ Hz

Where: $0 < \text{High count} < 256$ and

$0 < \text{Low count} < 256$

The duty cycle of the carrier signal is controlled by varying the ratio of high time to low + high time. As the input clock period is fixed, the duty cycle resolution will be proportional to the number of counts required to generate the desired carrier period.

$$\mathrm{DutyCycle} = \frac{\mathrm{Highcount}}{\mathrm{Highcount{+}Lowcount}}$$

46.7.3 Modulator

The modulator block controls the state of the infrared out signal (IRO). The modulator output is gated on to the IRO signal when the modulator/carrier generator is enabled. . When the modulator/carrier generator is disabled, the IRO signal is controlled by the state of the IRO latch. OC[CMTPOL] enables the IRO signal to be active-high or active-low.

The following table describes the functions of the modulators in different modes:

Table 46-6. Mode functions

Mode	Function
Time	The modulator can gate the carrier onto the modulator output.
Baseband	The modulator can control the logic level of the modulator output.
FSK	The modulator can count carrier periods and instruct the carrier generator to alternate between two carrier frequencies whenever a modulation period consisting of mark and space counts, expires.

The modulator provides a simple method to control protocol timing. The modulator has a minimum resolution of 1.0 μs with an 8 MHz. It can count bus clocks to provide real-time control, or carrier clocks for self-clocked protocols.

The modulator includes a 17-bit down counter with underflow detection. The counter is loaded from the 16-bit modulation mark period buffer registers, CMD1 and CMD2. The most significant bit is loaded with a logic 0 and serves as a sign bit.

When	Then
The counter holds a positive value	The modulator gate is open and the carrier signal is driven to the transmitter block.
The counter underflows	The modulator gate is closed and a 16-bit comparator is enabled which compares the logical complement of the value of the down counter with the contents of the modulation space period register which has been loaded from the registers, CMD3 and CMD4.

When a match is obtained, the cycle repeats by opening the modulator gate, reloading the counter with the contents of CMD1 and CMD2, and reloading the modulation space period register with the contents of CMD3 and CMD4.

The modulation space period is activated when the carrier signal is low to prohibit cutting off the high pulse of a carrier signal. If the carrier signal is high, the modulator extends the mark period until the carrier signal becomes low. To deassert the space period and assert the mark period, the carrier signal must have gone low to ensure that a space period is not erroneously shortened.

If the contents of the modulation space period register are all zeroes, the match will be immediate and no space period will be generated, for instance, for FSK protocols that require successive bursts of different frequencies).

MSC[MCGEN] must be set to enable the modulator timer.

The following figure presents the block diagram of the modulator.

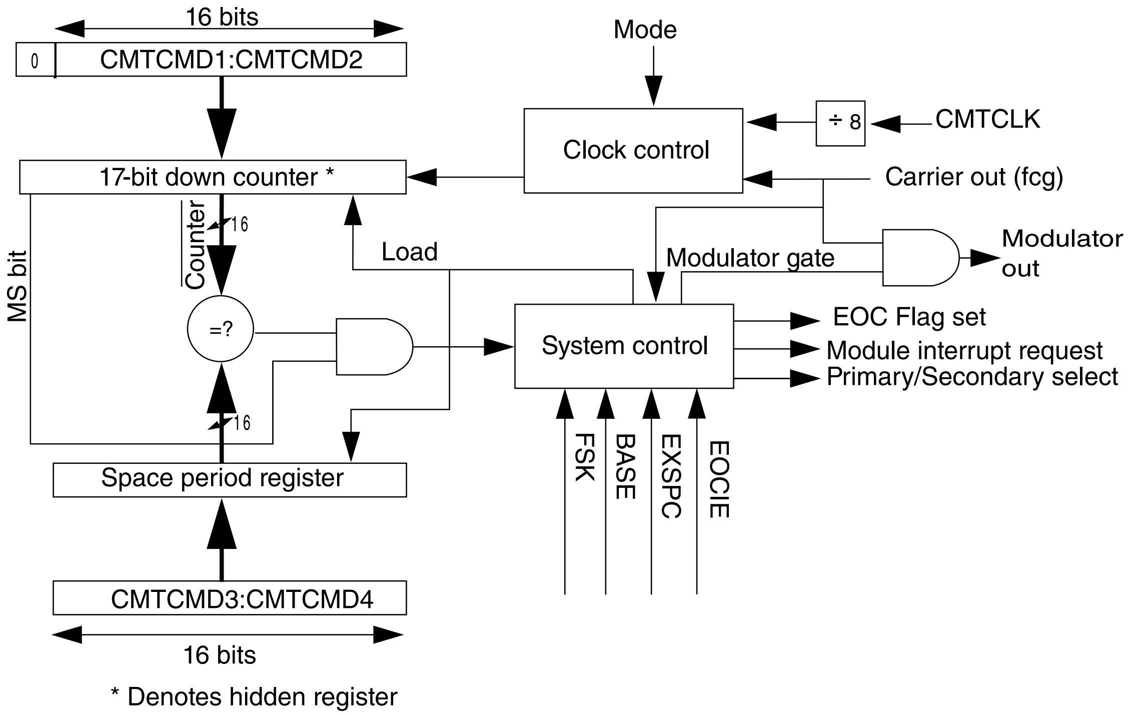


Figure 46-4. Modulator block diagram

46.7.3.1 Time mode

When the modulator operates in Time mode, or, when MSC[MCGEN] is set, and MSC[BASE] and MSC[FSK] are cleared:

- The modulation mark period consists of an integer number of (CMTCLK ÷ 8) clock periods.
- The modulation space period consists of 0 or an integer number of (CMTCLK ÷ 8) clock periods.

With an 8 MHz IF and MSC[CMTDIV] = 00, the modulator resolution is 1 μs and has a maximum mark and space period of about 65.535 ms each . See Figure 46-5 for an example of the Time and Baseband mode outputs.

The mark and space time equations for Time and Baseband mode are:

$t_{mark} = (CMD1:CMD2 + 1) \div (f_{CMTCLK} \div 8)$

$t_{space} = CMD3:CMD4 \div (f_{CMTCLK} \div 8)$

where CMD1:CMD2 and CMD3:CMD4 are the decimal values of the concatenated registers.

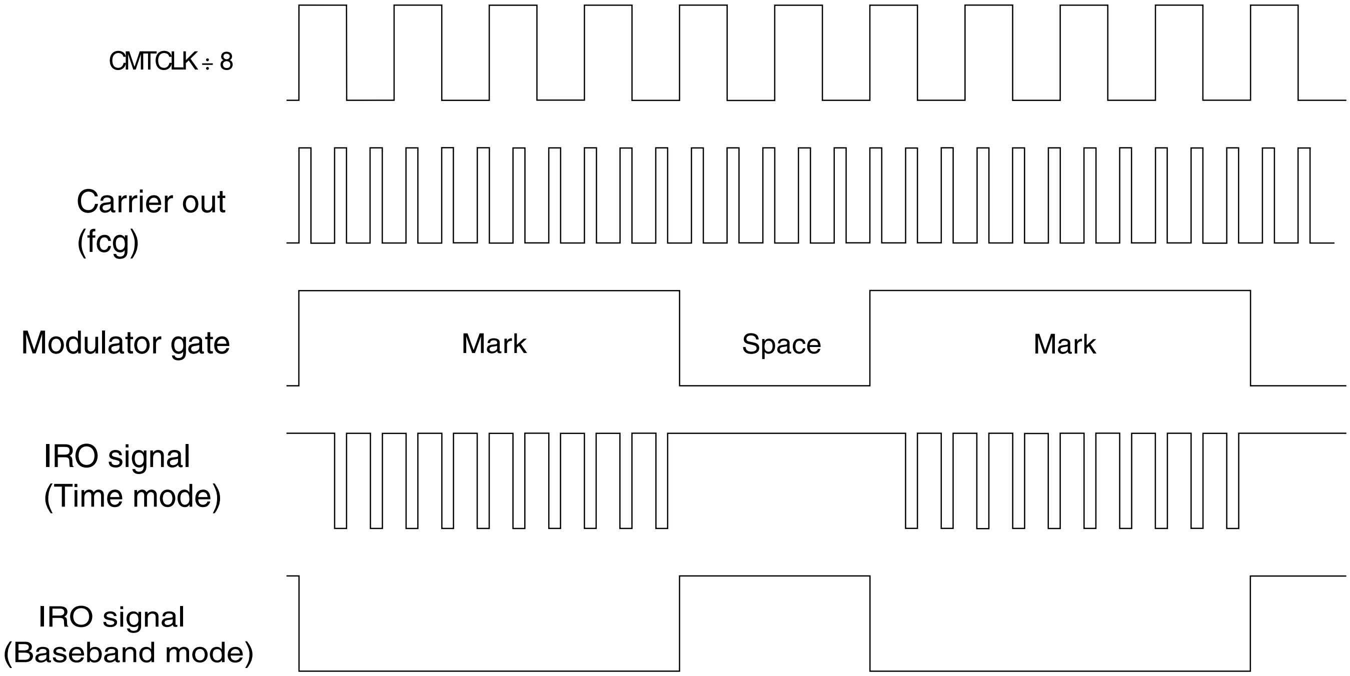


Figure 46-5. Example: CMT output in Time and Baseband modes with OC[CMTPOL]=0

46.7.3.2 Baseband mode

Baseband mode, that is, when MSC[MCGEN] and MSC[BASE] are set, is a derivative of Time mode, where the mark and space period is based on (CMTCLK ÷ 8) counts. The mark and space calculations are the same as in Time mode.

In this mode, the modulator output will be at a logic 1 for the duration of the mark period and at a logic 0 for the duration of a space period. See Figure 46-5 for an example of the output for both Baseband and Time modes. In the example, the carrier out frequency (f_{cg}) is generated with a high count of 0x01 and a low count of 0x02 that results in a divide of 3 of CMTCLK with a 33% duty cycle. The modulator down counter was loaded with the value 0x0003 and the space period register with 0x0002.

Note

The waveforms in Figure 46-5 and Figure 46-6 are for the purpose of conceptual illustration and are not meant to represent precise timing relationships between the signals shown.

46.7.3.3 FSK mode

When the modulator operates in FSK mode, that is, when MSC[MCGEN] and MSC[FSK] are set, and MSC[BASE] is cleared:

- The modulation mark and space periods consist of an integer number of carrier clocks (space period can be zero).
- When the mark period expires, the space period is transparently started as in Time mode.
- The carrier generator toggles between primary and secondary data register values whenever the modulator space period expires.

The space period provides an interpulse gap (no carrier). If CMD3:CMD4 = 0x0000, then the modulator and carrier generator will switch between carrier frequencies without a gap or any carrier glitches (zero space).

Using timing data for carrier burst and interpulse gap length calculated by the CPU, FSK mode can automatically generate a phase-coherent, dual-frequency FSK signal with programmable burst and interburst gaps.

The mark and space time equations for FSK mode are:

$t_{mark} = (\text{CMD1:CMD2} + 1) \div f_{cg}$

$t_{space} = (\text{CMD3:CMD4}) \div f_{cg}$

Where f_{cg} is the frequency output from the carrier generator. The example in Figure 46-6 shows what the IRO signal looks like in FSK mode with the following values:

- CMD1:CMD2 = 0x0003
- CMD3:CMD4 = 0x0002
- Primary carrier high count = 0x01
- Primary carrier low count = 0x02

- Secondary carrier high count = 0x03
- Secondary carrier low count = 0x01

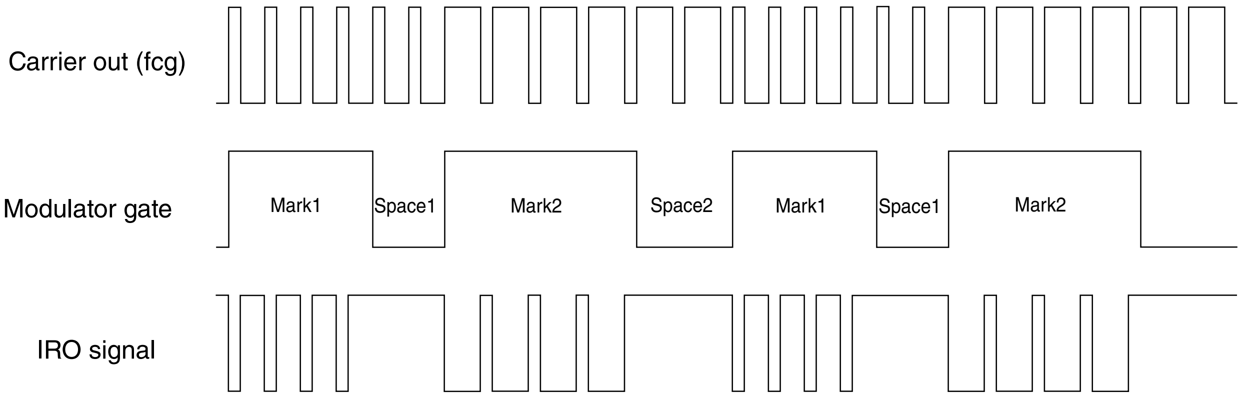


Figure 46-6. Example: CMT output in FSK mode

46.7.4 Extended space operation

In either Time, Baseband, or FSK mode, the space period can be made longer than the maximum possible value of the space period register. Setting MSC[EXSPC] will force the modulator to treat the next modulation period beginning with the next load of the counter and space period register, as a space period equal in length to the mark and space counts combined. Subsequent modulation periods will consist entirely of these extended space periods with no mark periods. Clearing MSC[EXSPC] will return the modulator to standard operation at the beginning of the next modulation period.

46.7.4.1 EXSPC operation in Time mode

To calculate the length of an extended space in Time or Baseband mode, add the mark and space times and multiply by the number of modulation periods when MSC[EXSPC] is set.

$t_{exspace} = (t_{mark} + t_{space})$ * (number of modulation periods)

For an example of extended space operation, see Figure 46-7.

Note

> The extended space enable feature can be used to emulate a zero mark event.

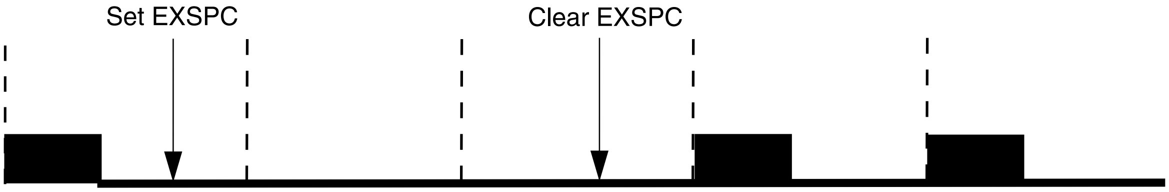


Figure 46-7. Extended space operation

46.7.4.2 EXSPC operation in FSK mode

In FSK mode, the modulator continues to count carrier out clocks, alternating between the primary and secondary registers at the end of each modulation period.

To calculate the length of an extended space in FSK mode, it is required to know whether MSC[EXSPC] was set on a primary or secondary modulation period, and the total number of both primary and secondary modulation periods completed while MSC[EXSPC] is high. A status bit for the current modulation is not accessible to the CPU. If necessary, software must maintain tracking of the current primary or secondary modulation cycle. The extended space period ends at the completion of the space period time of the modulation period during which MSC[EXSPC]is cleared.

The following table depicts the equations which can be used to calculate the extended space period depending on when MSC[EXSPC] is set.

If	Then
MSC[EXSPC] was set during a primary modulation cycle	Use the equation: $t_{exspace} = (t_{space})_p + (t_{mark} + t_{space})_s + (t_{mark} + t_{space})_p +...$
MSC[EXSPC] bit was set during a secondary modulation cycle	Use the equation: $t_{exspace} = (t_{space})_s + (t_{mark} + t_{space})_p + (t_{mark} + t_{space})_s +...$

Where the subscripts p and s refer to mark and space times for the primary and secondary modulation cycles.

46.8 CMT interrupts and DMA

The CMT generates an interrupt request or a DMA transfer request according to MSC[EOCIE], MSC[EOCF], DMA[DMA] bits.

Table 46-7. DMA transfer request x CMT interrupt request

MSC[EOCF]	DMA[DMA]	MSC[EOCIE]	DMA transfer request	CMT interrupt request
0	X	X	0	0
1	X	0	0	0
1	0	1	0	1
1	1	1	1	0

MSC[EOCF] is set:

- When the modulator is not currently active and MSC[MCGEN] is set to begin the initial CMT transmission.
- At the end of each modulation cycle when the counter is reloaded from CMD1:CMD2, while MSC[MCGEN] is set.

When MSC[MCGEN] is cleared and then set before the end of the modulation cycle, MSC[EOCF] will not be set when MSC[MCGEN] is set, but will become set at the end of the current modulation cycle.

When MSC[MCGEN] becomes disabled, the CMT module does not set MSC[EOCF] at the end of the last modulation cycle.

If MSC[EOCIE] is high when MSC[EOCF] is set, the CMT module will generate an interrupt request or a DMA transfer request.

MSC[EOCF] must be cleared to prevent from being generated by another event like interrupt or DMA request, after exiting the service routine. See the following table.

Table 46-8. How to clear MSC[EOCF]

DMA[DMA]	MSC[EOCIE]	Description
0	X	MSC[EOCF] is cleared by reading MSC followed by an access of CMD2 or CMD4.
1	X	MSC[EOCF] is cleared by the CMT DMA transfer done.

The EOC interrupt is coincident with:

- Loading the down-counter with the contents of CMD1:CMD2
- Loading the space period register with the contents of CMD3:CMD4

The EOC interrupt provides a means for the user to reload new mark/space values into the modulator data registers. Modulator data register updates will take effect at the end of the current modulation cycle.

NOTE

The down-counter and space period register are updated at the end of every modulation cycle, irrespective of interrupt handling and the state of MSC[EOCF].

Chapter 47
MCU: Real Time Clock (RTC)

47.1 Introduction

NOTE

For the chip-specific implementation details of this module's instances, see the chip configuration information.

47.1.1 Features

The RTC module features include:

- Independent power supply, POR, and 32 kHz crystal oscillator
- 32-bit seconds counter with roll-over protection and 32-bit alarm
- 16-bit prescaler with compensation that can correct errors between 0.12 ppm and 3906 ppm
- Register write protection
  - Lock register requires VBAT POR or software reset to enable write access
  - Access control registers require system reset to enable read and/or write access
- 1 Hz square wave output with optional interrupt
- 64-bit monotonic counter with roll-over protection
- Tamper time seconds register that records when the time was invalidated

47.1.2 Modes of operation

The RTC remains functional in all low power modes and can generate an interrupt to exit any low power mode. It operates in one of two modes of operation: chip power-up and chip power-down.

During chip power-down, RTC is powered from the backup power supply (VBAT) and is electrically isolated from the rest of the chip but continues to increment the time counter (if enabled) and retain the state of the RTC registers. The RTC registers are not accessible.

During chip power-up, RTC remains powered from the backup power supply (VBAT). All RTC registers are accessible by software and all functions are operational. If enabled, the 32.768 kHz clock can be supplied to the rest of the chip.

47.1.3 RTC signal descriptions

Table 47-1. RTC signal descriptions

Signal	Description	I/O
EXTAL32	32.768 kHz oscillator input	I
XTAL32	32.768 kHz oscillator output	O
RTC_CLKOUT	1 Hz square-wave output or OSCERCLK	O

47.1.3.1 RTC clock output

The clock to the seconds counter is available on the RTC_CLKOUT signal. It is a 1 Hz square wave output. See RTC_CLKOUT options for details.

47.2 Register definition

All registers must be accessed using 32-bit writes and all register accesses incur three wait states.

Write accesses to any register and read accesses to tamper and monotonic registers by non-supervisor mode software, when the supervisor access bit in the control register is clear, will terminate with a bus error.

Read accesses to other registers by non-supervisor mode software complete as normal.

Writing to a register protected by the write access register or lock register does not generate a bus error, but the write will not complete.

Reading a register protected by the read access register does not generate a bus error, but the register will read zero.

RTC memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4003_D000	RTC Time Seconds Register (RTC_TSR)	32	R/W	0000_0000h	47.2.1/1149
4003_D004	RTC Time Prescaler Register (RTC_TPR)	32	R/W	0000_0000h	47.2.2/1150
4003_D008	RTC Time Alarm Register (RTC_TAR)	32	R/W	0000_0000h	47.2.3/1150
4003_D00C	RTC Time Compensation Register (RTC_TCR)	32	R/W	0000_0000h	47.2.4/1151
4003_D010	RTC Control Register (RTC_CR)	32	R/W	0000_0000h	47.2.5/1152
4003_D014	RTC Status Register (RTC_SR)	32	R/W	0000_0001h	47.2.6/1154
4003_D018	RTC Lock Register (RTC_LR)	32	R/W	0000_FFFFh	47.2.7/1155
4003_D01C	RTC Interrupt Enable Register (RTC_IER)	32	R/W	0000_0007h	47.2.8/1157
4003_D020	RTC Tamper Time Seconds Register (RTC_TTSR)	32	R	Undefined	47.2.9/1158
4003_D024	RTC Monotonic Enable Register (RTC_MER)	32	R/W	0000_0000h	47.2.10/ 1159
4003_D028	RTC Monotonic Counter Low Register (RTC_MCLR)	32	R/W	0000_0000h	47.2.11/ 1159
4003_D02C	RTC Monotonic Counter High Register (RTC_MCHR)	32	R/W	0000_0000h	47.2.12/ 1160
4003_D800	RTC Write Access Register (RTC_WAR)	32	R/W	0000_FFFFh	47.2.13/ 1160
4003_D804	RTC Read Access Register (RTC_RAR)	32	R/W	0000_FFFFh	47.2.14/ 1163

47.2.1 RTC Time Seconds Register (RTC_TSR)

Address: 4003_D000h base + 0h offset = 4003_D000h

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

RTC_TSR field descriptions

Field	Description
TSR	Time Seconds Register When the time counter is enabled, the TSR is read only and increments once a second provided SR[TOF] or SR[TIF] are not set. The time counter will read as zero when SR[TOF] or SR[TIF] are set. When the time counter is disabled, the TSR can be read or written. Writing to the TSR when the time counter is disabled will clear the SR[TOF] and/or the SR[TIF]. Writing to TSR with zero is supported, but not

RTC_TSR field descriptions (continued)

Field	Description
	recommended because TSR will read as zero when SR[TIF] or SR[TOF] are set (indicating the time is invalid).

47.2.2 RTC Time Prescaler Register (RTC_TPR)

Address: 4003_D000h base + 4h offset = 4003_D004h

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

RTC_TPR field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
TPR	Time Prescaler Register When the time counter is enabled, the TPR is read only and increments every 32.768 kHz clock cycle. The time counter will read as zero when SR[TOF] or SR[TIF] are set. When the time counter is disabled, the TPR can be read or written. The TSR[TSR] increments when bit 14 of the TPR transitions from a logic one to a logic zero.

47.2.3 RTC Time Alarm Register (RTC_TAR)

Address: 4003_D000h base + 8h offset = 4003_D008h

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

RTC_TAR field descriptions

Field	Description
TAR	Time Alarm Register When the time counter is enabled, the SR[TAF] is set whenever the TAR[TAR] equals the TSR[TSR] and the TSR[TSR] increments. Writing to the TAR clears the SR[TAF].

47.2.4 RTC Time Compensation Register (RTC_TCR)

Address: 4003_D000h base + Ch offset = 4003_D00Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CIC								TCV								CIR								TCR							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

RTC_TCR field descriptions

Field	Description
31–24 CIC	Compensation Interval Counter Current value of the compensation interval counter. If the compensation interval counter equals zero then it is loaded with the contents of the CIR. If the CIC does not equal zero then it is decremented once a second.
23–16 TCV	Time Compensation Value Current value used by the compensation logic for the present second interval. Updated once a second if the CIC equals 0 with the contents of the TCR field. If the CIC does not equal zero then it is loaded with zero (compensation is not enabled for that second increment).
15–8 CIR	Compensation Interval Register Configures the compensation interval in seconds from 1 to 256 to control how frequently the TCR should adjust the number of 32.768 kHz cycles in each second. The value written should be one less than the number of seconds. For example, write zero to configure for a compensation interval of one second. This register is double buffered and writes do not take affect until the end of the current compensation interval.
TCR	Time Compensation Register Configures the number of 32.768 kHz clock cycles in each second. This register is double buffered and writes do not take affect until the end of the current compensation interval. 80h Time Prescaler Register overflows every 32896 clock cycles. ... ... FFh Time Prescaler Register overflows every 32769 clock cycles. 00h Time Prescaler Register overflows every 32768 clock cycles. 01h Time Prescaler Register overflows every 32767 clock cycles. .... 7Fh Time Prescaler Register overflows every 32641 clock cycles.

47.2.5 RTC Control Register (RTC_CR)

Address: 4003_D000h base + 10h offset = 4003_D010h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0								0							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	Reserved	SC2P	SC4P	SC8P	SC16P	CLKO	OSCE	0			WPS	UM	SUP	WPE	SWR
W		0														
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

RTC_CR field descriptions

Field	Description
31–24 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
23–15 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
14 Reserved	This field is reserved. It must always be written to 0.
13 SC2P	Oscillator 2pF Load Configure 0 Disable the load. 1 Enable the additional load.
12 SC4P	Oscillator 4pF Load Configure 0 Disable the load. 1 Enable the additional load.

Table continues on the next page...

RTC_CR field descriptions (continued)

Field	Description
11 SC8P	Oscillator 8pF Load Configure 0 Disable the load. 1 Enable the additional load.
10 SC16P	Oscillator 16pF Load Configure 0 Disable the load. 1 Enable the additional load.
9 CLKO	Clock Output 0 The 32 kHz clock is output to other peripherals. 1 The 32 kHz clock is not output to other peripherals.
8 OSCE	Oscillator Enable 0 32.768 kHz oscillator is disabled. 1 32.768 kHz oscillator is enabled. After setting this bit, wait the oscillator startup time before enabling the time counter to allow the 32.768 kHz clock time to stabilize.
7–5 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
4 WPS	Wakeup Pin Select The wakeup pin is optional and not available on all devices. 0 Wakeup pin asserts (active low, open drain) if the RTC interrupt asserts or the wakeup pin is turned on. 1 Wakeup pin instead outputs the RTC 32kHz clock, provided the wakeup pin is turned on and the 32kHz clock is output to other peripherals.
3 UM	Update Mode Allows SR[TCE] to be written even when the Status Register is locked. When set, the SR[TCE] can always be written if the SR[TIF] or SR[TOF] are set or if the SR[TCE] is clear. Allows the monotonic enable register to be written when it is locked. When set, the monotonic enable register can always be written if the SR[TIF] or SR[MOF] are set or if the montonic counter enable is clear. 0 Registers cannot be written when locked. 1 Registers can be written when locked under limited conditions.
2 SUP	Supervisor Access Configures non-supervisor mode write access to all RTC registers and non-supervisor mode read access to RTC tamper/monotonic registers 0 Non-supervisor mode write accesses are not supported and generate a bus error. 1 Non-supervisor mode write accesses are supported.
1 WPE	Wakeup Pin Enable The wakeup pin is optional and not available on all devices. 0 Wakeup pin is disabled. 1 Wakeup pin is enabled and wakeup pin asserts if the RTC interrupt asserts or the wakeup pin is turned on.

Table continues on the next page...

RTC_CR field descriptions (continued)

Field	Description
0 SWR	Software Reset 0 No effect. 1 Resets all RTC registers except for the SWR bit and the RTC_WAR and RTC_RAR registers . The SWR bit is cleared by VBAT POR and by software explicitly clearing it.

47.2.6 RTC Status Register (RTC_SR)

Address: 4003_D000h base + 14h offset = 4003_D014h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0											TCE	MOF	TAF	TOF	TIF
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

RTC_SR field descriptions

Field	Description
31–5 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
4 TCE	Time Counter Enable When time counter is disabled the TSR register and TPR register are writeable, but do not increment. When time counter is enabled the TSR register and TPR register are not writeable, but increment. 0 Time counter is disabled. 1 Time counter is enabled.
3 MOF	Monotonic Overflow Flag Monotonic overflow flag is set when the monotonic counter is enabled and the monotonic counter high overflows. The monotonic counter does not increment and will read as zero when this bit is set. This bit is cleared by writing the monotonic counter high register when the monotonic counter is disabled. 0 Monotonic counter overflow has not occurred. 1 Monotonic counter overflow has occurred and monotonic counter is read as zero.
2 TAF	Time Alarm Flag Time alarm flag is set when the TAR[TAR] equals the TSR[TSR] and the TSR[TSR] increments. This bit is cleared by writing the TAR register. 0 Time alarm has not occurred. 1 Time alarm has occurred.

Table continues on the next page...

RTC_SR field descriptions (continued)

Field	Description
1 TOF	Time Overflow Flag Time overflow flag is set when the time counter is enabled and overflows. The TSR and TPR do not increment and read as zero when this bit is set. This bit is cleared by writing the TSR register when the time counter is disabled. 0 Time overflow has not occurred. 1 Time overflow has occurred and time counter is read as zero.
0 TIF	Time Invalid Flag The time invalid flag is set on VBAT POR or software reset. The TSR and TPR do not increment and read as zero when this bit is set. This bit is cleared by writing the TSR register when the time counter is disabled. The monotonic counter register is held in reset whenever the time invalid flag is set. 0 Time is valid. 1 Time is invalid and time counter is read as zero.

47.2.7 RTC Lock Register (RTC_LR)

Address: 4003_D000h base + 18h offset = 4003_D018h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved				MCHL	MCLL	MEL	TTSL	1	LRL	SRL	CRL	TCL	1		
W	1															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

RTC_LR field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
15–12 Reserved	This field is reserved.
11 MCHL	Monotonic Counter High Lock After being cleared, this bit can be set only by VBAT POR or software reset. 0 Monotonic Counter High Register is locked and writes are ignored. 1 Monotonic Counter High Register is not locked and writes complete as normal.
10 MCLL	Monotonic Counter Low Lock After being cleared, this bit can be set only by VBAT POR or software reset. 0 Monotonic Counter Low Register is locked and writes are ignored. 1 Monotonic Counter Low Register is not locked and writes complete as normal.
9 MEL	Monotonic Enable Lock After being cleared, this bit can be set only by VBAT POR or software reset. 0 Monotonic Enable Register is locked and writes are ignored. 1 Monotonic Enable Register is not locked and writes complete as normal.
8 TTSL	Tamper Time Seconds Lock After being cleared, this bit can be set only by VBAT POR or software reset. 0 Tamper Time Seconds Register is locked and writes are ignored. 1 Tamper Time Seconds Register is not locked and writes complete as normal.
7 Reserved	This field is reserved. This read-only field is reserved and always has the value 1.
6 LRL	Lock Register Lock After being cleared, this bit can be set only by VBAT POR or software reset. 0 Lock Register is locked and writes are ignored. 1 Lock Register is not locked and writes complete as normal.
5 SRL	Status Register Lock After being cleared, this bit can be set only by VBAT POR or software reset. 0 Status Register is locked and writes are ignored. 1 Status Register is not locked and writes complete as normal.
4 CRL	Control Register Lock After being cleared, this bit can only be set by VBAT POR. 0 Control Register is locked and writes are ignored. 1 Control Register is not locked and writes complete as normal.
3 TCL	Time Compensation Lock After being cleared, this bit can be set only by VBAT POR or software reset.

Table continues on the next page...

RTC_LR field descriptions (continued)

Field	Description
	0 Time Compensation Register is locked and writes are ignored. 1 Time Compensation Register is not locked and writes complete as normal.
Reserved	This field is reserved. This read-only field is reserved and always has the value 1.

47.2.8 RTC Interrupt Enable Register (RTC_IER)

Address: 4003_D000h base + 1Ch offset = 4003_D01Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								WPON	Reserved		TSIE	MOIE	TAIE	TOIE	TIIE
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1

RTC_IER field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
15–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7 WPON	Wakeup Pin On The wakeup pin is optional and not available on all devices. Whenever the wakeup pin is enabled and this bit is set, the wakeup pin will assert. 0 No effect. 1 If the wakeup pin is enabled, then the wakeup pin will assert.
6–5 Reserved	This field is reserved.
4 TSIE	Time Seconds Interrupt Enable The seconds interrupt is an edge-sensitive interrupt with a dedicated interrupt vector. It is generated once a second and requires no software overhead (there is no corresponding status flag to clear). 0 Seconds interrupt is disabled. 1 Seconds interrupt is enabled.

Table continues on the next page...

RTC_IER field descriptions (continued)

Field	Description
3 MOIE	Monotonic Overflow Interrupt Enable 0 Monotonic overflow flag does not generate an interrupt. 1 Monotonic overflow flag does generate an interrupt.
2 TAIE	Time Alarm Interrupt Enable 0 Time alarm flag does not generate an interrupt. 1 Time alarm flag does generate an interrupt.
1 TOIE	Time Overflow Interrupt Enable 0 Time overflow flag does not generate an interrupt. 1 Time overflow flag does generate an interrupt.
0 TIIE	Time Invalid Interrupt Enable 0 Time invalid flag does not generate an interrupt. 1 Time invalid flag does generate an interrupt.

47.2.9 RTC Tamper Time Seconds Register (RTC_TTSR)

Address: 4003_D000h base + 20h offset = 4003_D020h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TTS																															
W																																
Reset	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*

* Notes:

- x = Undefined at reset.

RTC_TTSR field descriptions

Field	Description
TTS	Tamper Time Seconds If the time invalid flag is set then reading this register returns the contents of the time seconds register at the point at which the time invalid flag was set. If the time invalid flag is clear then reading this register returns zero. Writing the tamper time seconds register with any value will set the time invalid flag.

47.2.10 RTC Monotonic Enable Register (RTC_MER)

Address: 4003_D000h base + 24h offset = 4003_D024h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0											MCE	0			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

RTC_MER field descriptions

Field	Description
31–5 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
4 MCE	Monotonic Counter Enable 0 Writes to the monotonic counter load the counter with the value written. 1 Writes to the monotonic counter increment the counter.
Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

47.2.11 RTC Monotonic Counter Low Register (RTC_MCLR)

Address: 4003_D000h base + 28h offset = 4003_D028h

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

RTC_MCLR field descriptions

Field	Description
MCL	Monotonic Counter Low When the time invalid flag is set, the monotonic counter is held in reset. When the monotonic counter enable is clear, a write to this register will load the counter with the value written. When the monotonic counter enable is set, a write to this register will cause it to increment. A write to monotonic counter low that causes it to overflow will also increment monotonic counter high.

47.2.12 RTC Monotonic Counter High Register (RTC_MCHR)

Address: 4003_D000h base + 2Ch offset = 4003_D02Ch

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

RTC_MCHR field descriptions

Field	Description
MCH	Monotonic Counter High When the time invalid flag is set, the monotonic counter is held in reset. When the monotonic counter enable is clear, a write to this register will load the counter with the value written. When the monotonic counter enable is set, a write to this register will cause it to increment. A write to monotonic counter low that causes it to overflow will also increment monotonic counter high.

47.2.13 RTC Write Access Register (RTC_WAR)

Address: 4003_D000h base + 800h offset = 4003_D800h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved				MCHW	MCLW	MERW	TTSW	IERW	LRW	SRW	CRW	TCRW	TARW	TPRW	TSRW
W	1															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

RTC_WAR field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
15–12 Reserved	This field is reserved.
11 MCHW	Monotonic Counter High Write After being cleared, this bit is set only by system reset. It is not affected by VBAT POR or software reset. 0 Writes to the Monotonic Counter High Register are ignored. 1 Writes to the Monotonic Counter High Register complete as normal.
10 MCLW	Monotonic Counter Low Write After being cleared, this bit is set only by system reset. It is not affected by VBAT POR or software reset. 0 Writes to the Monotonic Counter Low Register are ignored. 1 Writes to the Monotonic Counter Low Register complete as normal.
9 MERW	Monotonic Enable Register Write After being cleared, this bit is set only by system reset. It is not affected by VBAT POR or software reset. 0 Writes to the Monotonic Enable Register are ignored. 1 Writes to the Monotonic Enable Register complete as normal.
8 TTSW	Tamper Time Seconds Write After being cleared, this bit is set onlyby system reset. It is not affected by VBAT POR or software reset. 0 Writes to the Tamper Time Seconds Register are ignored. 1 Writes to the Tamper Time Seconds Register complete as normal.
7 IERW	Interrupt Enable Register Write After being cleared, this bit is set only by system reset. It is not affected by VBAT POR or software reset. 0 Writes to the Interupt Enable Register are ignored. 1 Writes to the Interrupt Enable Register complete as normal.
6 LRW	Lock Register Write After being cleared, this bit is set only by system reset. It is not affected by VBAT POR or software reset. 0 Writes to the Lock Register are ignored. 1 Writes to the Lock Register complete as normal.
5 SRW	Status Register Write After being cleared, this bit is set only by system reset. It is not affected by VBAT POR or software reset. 0 Writes to the Status Register are ignored. 1 Writes to the Status Register complete as normal.
4 CRW	Control Register Write After being cleared, this bit is set only by system reset. It is not affected by VBAT POR or software reset.

Table continues on the next page...

RTC_WAR field descriptions (continued)

Field	Description
	0 Writes to the Control Register are ignored. 1 Writes to the Control Register complete as normal.
3 TCRW	Time Compensation Register Write After being cleared, this bit is set only by system reset. It is not affected by VBAT POR or software reset. 0 Writes to the Time Compensation Register are ignored. 1 Writes to the Time Compensation Register complete as normal.
2 TARW	Time Alarm Register Write After being cleared, this bit is set only by system reset. It is not affected by VBAT POR or software reset. 0 Writes to the Time Alarm Register are ignored. 1 Writes to the Time Alarm Register complete as normal.
1 TPRW	Time Prescaler Register Write After being cleared, this bit is set only by system reset. It is not affected by VBAT POR or software reset. 0 Writes to the Time Prescaler Register are ignored. 1 Writes to the Time Prescaler Register complete as normal.
0 TSRW	Time Seconds Register Write After being cleared, this bit is set only by system reset. It is not affected by VBAT POR or software reset. 0 Writes to the Time Seconds Register are ignored. 1 Writes to the Time Seconds Register complete as normal.

47.2.14 RTC Read Access Register (RTC_RAR)

Address: 4003_D000h base + 804h offset = 4003_D804h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved				MCHR	MCLR	MERR	TTSR	IERR	LRR	SRR	CRR	TCRR	TARR	TPRR	TSRR
W	1															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

RTC_RAR field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
15–12 Reserved	This field is reserved.
11 MCHR	Monotonic Counter High Read After being cleared, this bit is set only by system reset. It is not affected by VBAT POR or software reset. 0 Reads to the Monotonic Counter High Register are ignored. 1 Reads to the Monotonic Counter High Register complete as normal.
10 MCLR	Monotonic Counter Low Read After being cleared, this bit is set only by system reset. It is not affected by VBAT POR or software reset. 0 Reads to the Monotonic Counter Low Register are ignored. 1 Reads to the Monotonic Counter Low Register complete as normal.
9 MERR	Monotonic Enable Register Read After being cleared, this bit is set only by system reset. It is not affected by VBAT POR or software reset.

Table continues on the next page...

RTC_RAR field descriptions (continued)

Field	Description
	0 Reads to the Monotonic Enable Register are ignored. 1 Reads to the Monotonic Enable Register complete as normal.
8 TTSR	Tamper Time Seconds Read After being cleared, this bit is set only by system reset. It is not affected by VBAT POR or software reset. 0 Reads to the Tamper Time Seconds Register are ignored. 1 Reads to the Tamper Time Seconds Register complete as normal.
7 IERR	Interrupt Enable Register Read After being cleared, this bit is set only by system reset. It is not affected by VBAT POR or software reset. 0 Reads to the Interrupt Enable Register are ignored. 1 Reads to the Interrupt Enable Register complete as normal.
6 LRR	Lock Register Read After being cleared, this bit is set only by system reset. It is not affected by VBAT POR or software reset. 0 Reads to the Lock Register are ignored. 1 Reads to the Lock Register complete as normal.
5 SRR	Status Register Read After being cleared, this bit is set only by system reset. It is not affected by VBAT POR or software reset. 0 Reads to the Status Register are ignored. 1 Reads to the Status Register complete as normal.
4 CRR	Control Register Read After being cleared, this bit is set only by system reset. It is not affected by VBAT POR or software reset. 0 Reads to the Control Register are ignored. 1 Reads to the Control Register complete as normal.
3 TCRR	Time Compensation Register Read After being cleared, this bit is set only by system reset. It is not affected by VBAT POR or software reset. 0 Reads to the Time Compensation Register are ignored. 1 Reads to the Time Compensation Register complete as normal.
2 TARR	Time Alarm Register Read After being cleared, this bit is set only by system reset. It is not affected by VBAT POR or software reset. 0 Reads to the Time Alarm Register are ignored. 1 Reads to the Time Alarm Register complete as normal.
1 TPRR	Time Prescaler Register Read After being cleared, this bit is set only by system reset. It is not affected by VBAT POR or software reset. 0 Reads to the Time Pprescaler Register are ignored. 1 Reads to the Time Prescaler Register complete as normal.

Table continues on the next page...

RTC_RAR field descriptions (continued)

Field	Description
0 TSRR	Time Seconds Register Read After being cleared, this bit is set only by system reset. It is not affected by VBAT POR or software reset. 0 Reads to the Time Seconds Register are ignored. 1 Reads to the Time Seconds Register complete as normal.

47.3 Functional description

47.3.1 Power, clocking, and reset

The RTC is an always powered block that remains active in all low power modes and is powered by the battery power supply (VBAT). The battery power supply ensures that the RTC registers retain their state during chip power-down and that the RTC time counter remains operational.

The time counter within the RTC is clocked by a 32.768 kHz clock and can supply this clock to other peripherals. The 32.768 kHz clock can only be sourced from an external crystal using the oscillator that is part of the RTC module.

The RTC includes its own analog POR block, which generates a VBAT power-on-reset signal whenever the RTC module is powered up and initializes all RTC registers to their default state. A software reset bit can also initialize all RTC registers. The RTC also monitors the chip power supply and electrically isolates itself when the rest of the chip is powered down.

NOTE

An attempt to access an RTC register, except the access control registers, results in a bus error when:

- VBAT is powered down,
- the RTC is electrically isolated, or
- VBAT POR is asserted.

47.3.1.1 Oscillator control

The 32.768 kHz crystal oscillator is disabled at VBAT POR and must be enabled by software. After enabling the cystal oscillator, wait the oscillator startup time before setting SR[TCE] or using the oscillator clock external to the RTC.

The crystal oscillator includes tunable capacitors that can be configured by software. Do not change the capacitance unless the oscillator is disabled.

47.3.1.2 Software reset

Writing 1 to CR[SWR] forces the equivalent of a VBAT POR to the rest of the RTC module. CR[SWR] is not affected by the software reset and must be cleared by software. The access control registers are not affected by either VBAT POR or the software reset; they are reset by the chip reset.

47.3.1.3 Supervisor access

When the supervisor access control bit is clear, only supervisor mode software can write to the RTC registers or read the RTC tamper and monotonic registers, non-supervisor mode software will generate a bus error. Both supervisor and non-supervisor mode software can always read the other RTC registers.

47.3.2 Time counter

The time counter consists of a 32-bit seconds counter that increments once every second and a 16-bit prescaler register that increments once every 32.768 kHz clock cycle.

Reading the time counter (either seconds or prescaler) while it is incrementing may return invalid data due to synchronization of the read data bus. If it is necessary for software to read the prescaler or seconds counter when they could be incrementing, it is recommended that two read accesses are performed and that software verifies that the same data was returned for both reads.

The time seconds register and time prescaler register can be written only when SR[TCE] is clear. Always write to the prescaler register before writing to the seconds register, because the seconds register increments on the falling edge of bit 14 of the prescaler register.

The time prescaler register increments provided SR[TCE] is set, SR[TIF] is clear, SR[TOF] is clear, and the 32.768 kHz clock source is present. After enabling the oscillator, wait the oscillator startup time before setting SR[TCE] to allow time for the oscillator clock output to stabilize.

If the time seconds register overflows then the SR[TOF] will set and the time prescaler register will stop incrementing. Clear SR[TOF] by initializing the time seconds register. The time seconds register and time prescaler register read as zero whenever SR[TOF] is set.

SR[TIF] is set on VBAT POR and software reset and is cleared by initializing the time seconds register. The time seconds register and time prescaler register read as zero whenever SR[TIF] is set.

47.3.3 Compensation

The compensation logic provides an accurate and wide compensation range and can correct errors as high as 3906 ppm and as low as 0.12 ppm. The compensation factor must be calculated externally to the RTC and supplied by software to the compensation register. The RTC itself does not calculate the amount of compensation that is required, although the 1 Hz clock is output to an external pin in support of external calibration logic.

Crystal compensation can be supported by using firmware and crystal characteristics to determine the compensation amount. Temperature compensation can be supported by firmware that periodically measures the external temperature via ADC and updates the compensation register based on a look-up table that specifies the change in crystal frequency over temperature.

The compensation logic alters the number of 32.768 kHz clock cycles it takes for the prescaler register to overflow and increment the time seconds counter. The time compensation value is used to adjust the number of clock cycles between -127 and +128. Cycles are added or subtracted from the prescaler register when the prescaler register equals 0x3FFF and then increments. The compensation interval is used to adjust the frequency at which the time compensation value is used, that is, from once a second to once every 256 seconds.

Updates to the time compensation register will not take effect until the next time the time seconds register increments and provided the previous compensation interval has expired. When the compensation interval is set to other than once a second then the compensation is applied in the first second interval and the remaining second intervals receive no compensation.

Compensation is disabled by configuring the time compensation register to zero.

47.3.4 Time alarm

The Time Alarm register (TAR), SR[TAF], and IER[TAIE] allow the RTC to generate an interrupt at a predefined time. The 32-bit TAR is compared with the 32-bit Time Seconds register (TSR) each time it increments. SR[TAF] will set when TAR equals TSR and TSR increments.

SR[TAF] is cleared by writing TAR. This will usually be the next alarm value, although writing a value that is less than TSR, such as 0, will prevent SR[TAF] from setting again. SR[TAF] cannot otherwise be disabled, although the interrupt it generates is enabled or disabled by IER[TAIE].

47.3.5 Update mode

The Update Mode field in the Control register (CR[UM]) configures software write access to the Time Counter Enable (SR[TCE]) field. When CR[UM] is clear, SR[TCE] can be written only when LR[SRL] is set. When CR[UM] is set, SR[TCE] can also be written when SR[TCE] is clear or when SR[TIF] or SR[TOF] are set. This allows the time seconds and prescaler registers to be initialized whenever time is invalidated, while preventing the time seconds and prescaler registers from being changed on the fly. When LR[SRL] is set, CR[UM] has no effect on SR[TCE].

CR[UM] also configures software write access to the Monotonic Counter Enable (MER[MCE]) bit. When CR[UM] is clear, MER[MCE] can be written only when LR[MEL] is set. When CR[UM] is set, MER[MCE] can also be written when MER[MCE] is clear or when SR[TIF] or SR[MOF] are set. This allows the monotonic counter register to be initialized whenever the monotonic counter is invalid, while preventing the monotonic counter from being changed on the fly. When LR[MEL] is set, CR[UM] has no effect on MCR[MCE].

47.3.6 Monotonic counter

The 64-bit Monotonic Counter is a counter that cannot be exhausted or return to any previous value, once it has been initialized. If the monotonic overflow flag is set, the monotonic counter returns zero and does not increment.

Depending on the value of the monotonic counter enable bit, writing to the monotonic counter either initializes the register with the value written, or increments the register by one (and the value written is ignored).

When the monotonic counter is enabled, the monotonic counter high increments on either a write to the monotonic counter high register or if the monotonic counter low register overflows (due to a write to the monotonic counter low register). The monotonic overflow flag sets when the monotonic counter high register overflows and is cleared by writing the monotonic counter high register when the monotonic counter is disabled.

The monotonic counter is held in reset whenever the time invalid flag is set. Always clear the time invalid flag before initializing the monotonic counter.

47.3.7 Register lock

The Lock register (LR) can be used to block write accesses to certain registers until the next VBAT POR or software reset. Locking the Control register (CR) will disable the software reset. Locking LR will block future updates to LR.

Write accesses to a locked register are ignored and do not generate a bus error.

47.3.8 Access control

The read access and write access registers are implemented in the chip power domain and reset on the chip reset. They are not affected by the VBAT POR or the software reset. They are used to block read or write accesses to each register until the next chip system reset. When accesses are blocked, the bus access is not seen in the VBAT power supply and does not generate a bus error.

47.3.9 Interrupt

The RTC interrupt is asserted whenever a status flag and the corresponding interrupt enable bit are both set. It is always asserted on VBAT POR, and software reset, and when the VBAT power supply is powered down. The RTC interrupt is enabled at the chip level by enabling the chip-specific RTC clock gate control bit. The RTC interrupt can be used to wakeup the chip from any low-power mode.

The optional RTC seconds interrupt is an edge-sensitive interrupt with a dedicated interrupt vector that is generated once a second and requires no software overhead (there is no corresponding status flag to clear). It is enabled in the RTC by the time seconds interrupt enable bit and enabled at the chip level by setting the chip-specific RTC clock gate control bit. This interrupt is optional and may not be implemented on all devices.

Chapter 48
MCU: Universal Serial Bus OTG Controller (USBOTG)

48.1 Introduction

NOTE

For the chip-specific implementation details of this module's instances, see the chip configuration information.

This chapter describes the USB full speed OTG controller. The OTG implementation in this module provides limited host functionality and device solutions for implementing a USB 2.0 full-speed/low-speed compliant peripheral. The OTG logic implements features required by the *On-The-Go and Embedded Host Supplement to the USB 2.0 Specification* (usb.org, 2008). The USB full speed controller interfaces to a USBFS/LS transceiver.

NOTE

This chapter describes the following registers that have similar names: USB_OTGCTL, USB_CTL, USB_CTRL, and USB_CONTROL. These are all separate registers.

48.1.1 References

The following publications are referenced in this document. For copies or updates to these specifications, see the USB Implementers Forum, Inc. website at http://www.usb.org.

- *Universal Serial Bus Specification, Revision 2.0* , 2000, with amendments including those listed below
- *Errata for “USB Revision 2.0 April 27, 2000” as of 12/7/2000*
- *Errata for “USB Revision 2.0 April 27, 2000” as of 12/7/2000*

- *Pull-up / Pull-down Resistors* (USB Engineering Change Notice)
- *Suspend Current Limit Changes* (USB Engineering Change Notice)
- *Device Capacitance* (USB Engineering Change Notice)
- *USB 2.0 Connect Timing Update* (USB Engineering Change Notice as of April 4, 2013)
- *USB 2.0 VBUS Max Limit* (USB Engineering Change Notice)
- *On-The-Go and Embedded Host Supplement to the USB Revision 2.0 Specification*, Revision 2.0 version 1.1a, July 27, 2012
- *Maximum VBUS Voltage* (USB OTGEH Engineering Change Notice)
- *Universal Serial Bus Micro-USB Cables and Connectors Specification*, Revision 1.01, 2007

48.1.2 USB

The USB is a cable bus that supports data exchange between a host computer and a wide range of simultaneously accessible peripherals. The attached peripherals share USB bandwidth through a host-scheduled, token-based protocol. The bus allows peripherals to be attached, configured, used, and detached while the host and other peripherals are in operation.

USB software provides a uniform view of the system for all application software, hiding implementation details that make application software more portable. It manages the dynamic attach and detach of peripherals.

There is only one host in any USB system. The USB interface to the host computer system is referred to as the Host Controller.

There may be multiple USB devices in any system such as human interface devices, speakers, printers, etc. USB devices present a standard USB interface in terms of comprehension, response, and standard capability.

The host initiates transactions to specific peripherals, whereas the device responds to control transactions. The device sends and receives data to and from the host using a standard USB data format. USB 2.0 full-speed /low-speed peripherals operate at 12Mbit/s or 1.5 Mbit/s.

For additional information, see the USB 2.0 specification.

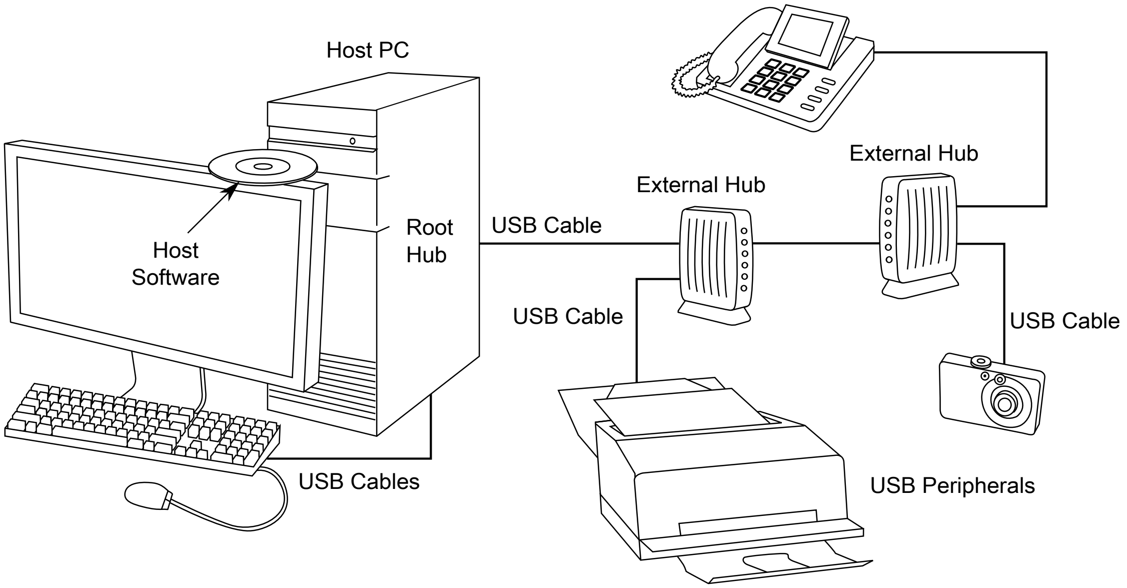


Figure 48-1. Example USB 2.0 system configuration

48.1.3 USB On-The-Go

USB is a popular standard for connecting peripherals and portable consumer electronic devices such as digital cameras and tablets to host PCs. The On-The-Go (OTG) Supplement to the USB Specification extends USB to peer-to-peer application. Using USB OTG technology, consumer electronics, peripherals, and portable devices can connect to each other to exchange data. For example, a digital camera can connect directly to a printer, or a keyboard can connect to a tablet to exchange data.

With the USB On-The-Go product, you can develop a fully USB-compliant peripheral device that can also assume the role of a USB host. Software determines the role of the device based on hardware signals, and then initializes the device in the appropriate mode of operation (host or peripheral) based on how it is connected. After connecting, the devices can negotiate using the OTG protocols to assume the role of host or peripheral based on the task to be accomplished.

For additional information, see the *On-The-Go and Embedded Host Supplement to the USB 2.0 Specification*.

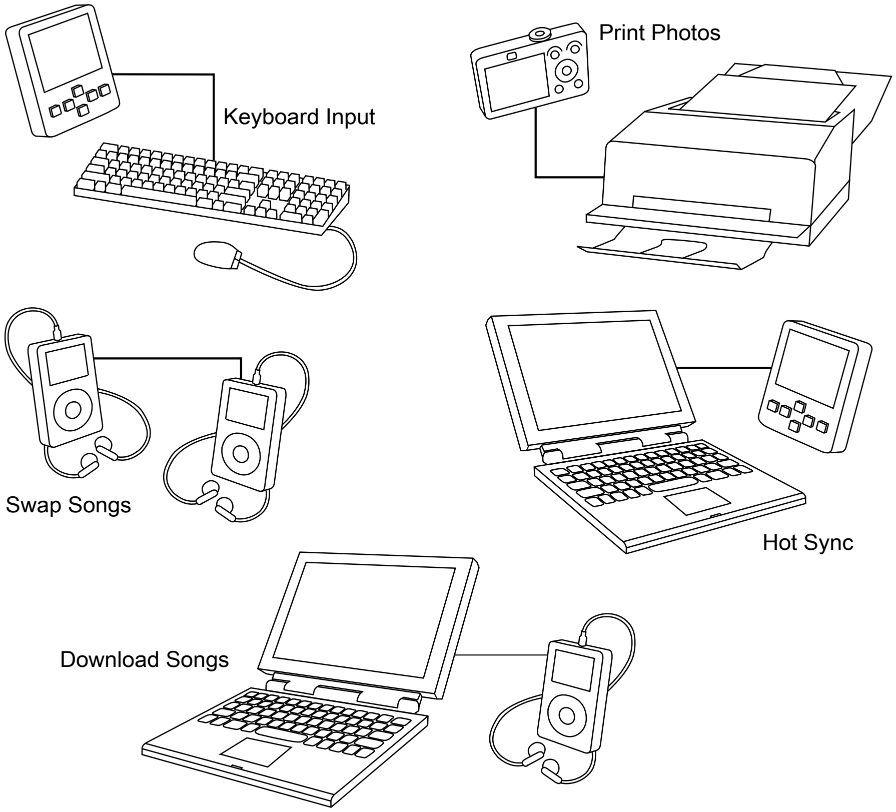


Figure 48-2. Example USB 2.0 On-The-Go configurations

48.1.4 USBFS Features

- USB 1.1 and 2.0 compatible FS device and FS/LS host controller with On-The-Go protocol logic
- 16 bidirectional endpoints
- DMA or FIFO data stream interfaces
- Low-power consumption

48.2 Functional description

USBOTG communicates with the processor core through status registers, control registers, and data structures in memory.

48.2.1 Data Structures

To efficiently manage USB endpoint communications, USBFS implements a Buffer Descriptor Table (BDT) in system memory. See Figure 48-5.

48.2.2 On-chip transceiver required external components

USB system operation requires external components to ensure that driver output impedance, eye diagram, and VBUS cable fault tolerance requirements are met. DM and DP I/O pads must connect through series resistors (approximately 33 Ω each) to the USB connector on the application printed circuit board (PCB). Additionally, signal quality optimizes when these 33 Ω resistors are mounted closer to the processor than to the USB board-level connector. The USB transceiver includes:

- An internal 1.5 kΩ pullup resistor on the USB_DP line for full-speed device (controlled by USB_CONTROL[DPPULLUPNONOTG] or USB_OTGCTL[DPHIGH])
- Internal 15 kΩ pulldown resistors on the USB_DP and USB_DM signals, which are primarily intended for Host mode operation but are also useful in Device mode as explained below.

NOTE

For device operation, the internal 15 kΩ pulldowns should be enabled to keep the DP and DM ports in a known quiescent state when the VBUS detection software determines that the USB connection is not activated, including the cases when no cable to a host is present or when the USB port is not used. The internal 15 kΩ pulldowns should be controlled by USB_CTRL[PDE] in this case.

For host operation, the internal 15 kΩ pulldowns are enabled automatically, as required by the USB 2.0 specification, when USB_CTL[HOSTMODEEN] is asserted high.

The following diagrams provide overviews of host-only, device-only, and dual-role connections, respectively. For more details, see the *Kinetis Peripheral Module Quick Reference*(KQRUG).

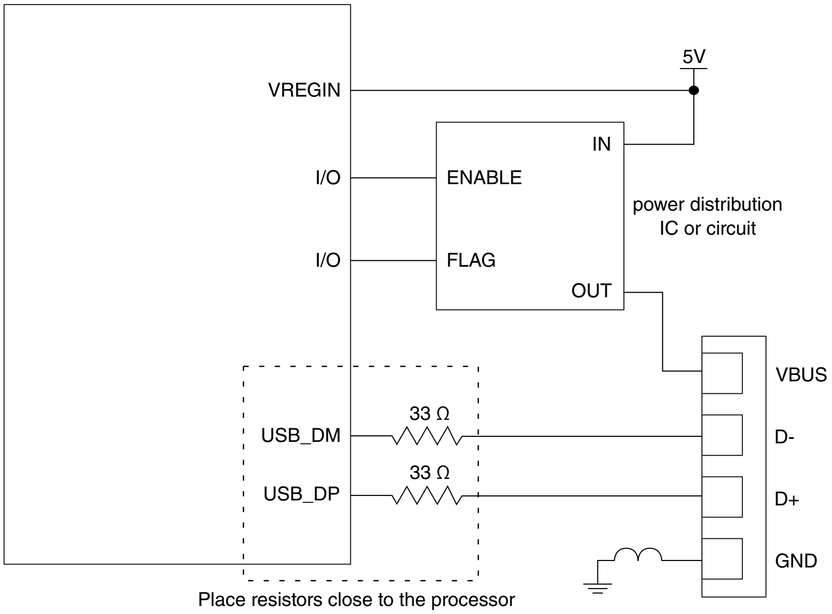


Figure 48-3. Host-only diagram

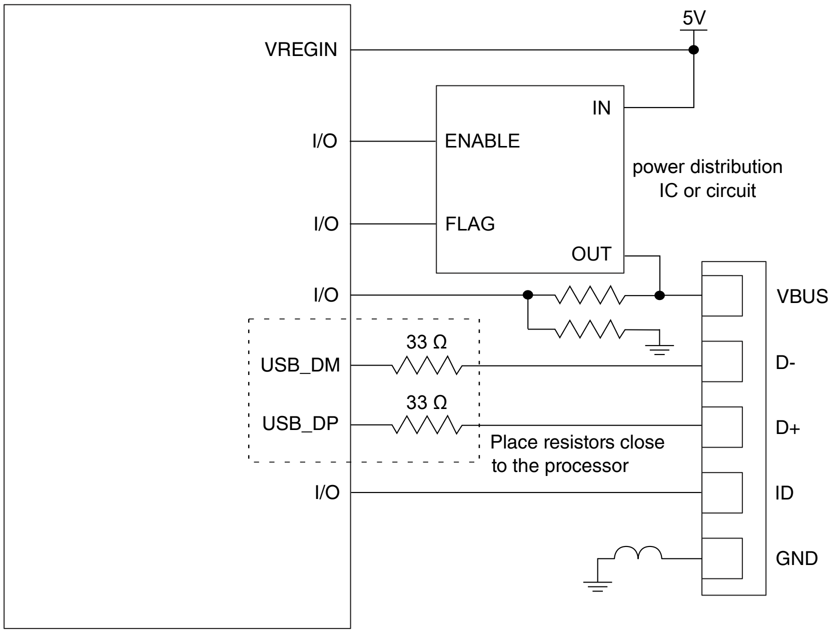


Figure 48-4. Dual-role diagram

48.3 Programmers interface

This section discusses the major components of the programming model for the USB module.

48.3.1 Buffer Descriptor Table

To efficiently manage USB endpoint communications USBFS implements a Buffer Descriptor Table (BDT) in system memory. The BDT resides on a 512-byte boundary in system memory and is pointed to by the BDT Page Registers. Every endpoint direction requires two 8-byte Buffer Descriptor (BD) entries. Therefore, a system with 16 fully bidirectional endpoints would require 512 bytes of system memory to implement the BDT. The two BD entries allows for an EVEN BD and ODD BD entry for each endpoint direction. This allows the microprocessor to process one BD while USBFS is processing the other BD. Double buffering BDs in this way allows USBFS to transfer data easily at the maximum throughput provided by USB.

Software should manage buffers for USBFS by updating the BDT when needed. This allows USBFS to efficiently manage data transmission and reception, while the microprocessor performs communication overhead processing and other function dependent applications. Because the buffers are shared between the microprocessor and USBFS, a simple semaphore mechanism is used to distinguish who is allowed to update the BDT and buffers in system memory. A semaphore, the OWN bit, is cleared to 0 when the BD entry is owned by the microprocessor. The microprocessor is allowed read and write access to the BD entry and the buffer in system memory when the OWN bit is 0. When the OWN bit is set to 1, the BD entry and the buffer in system memory are owned by USBFS. USBFS now has full read and write access and the microprocessor must not modify the BD or its corresponding data buffer. The BD also contains indirect address pointers to where the actual buffer resides in system memory. This indirect address mechanism is shown in the following diagram.

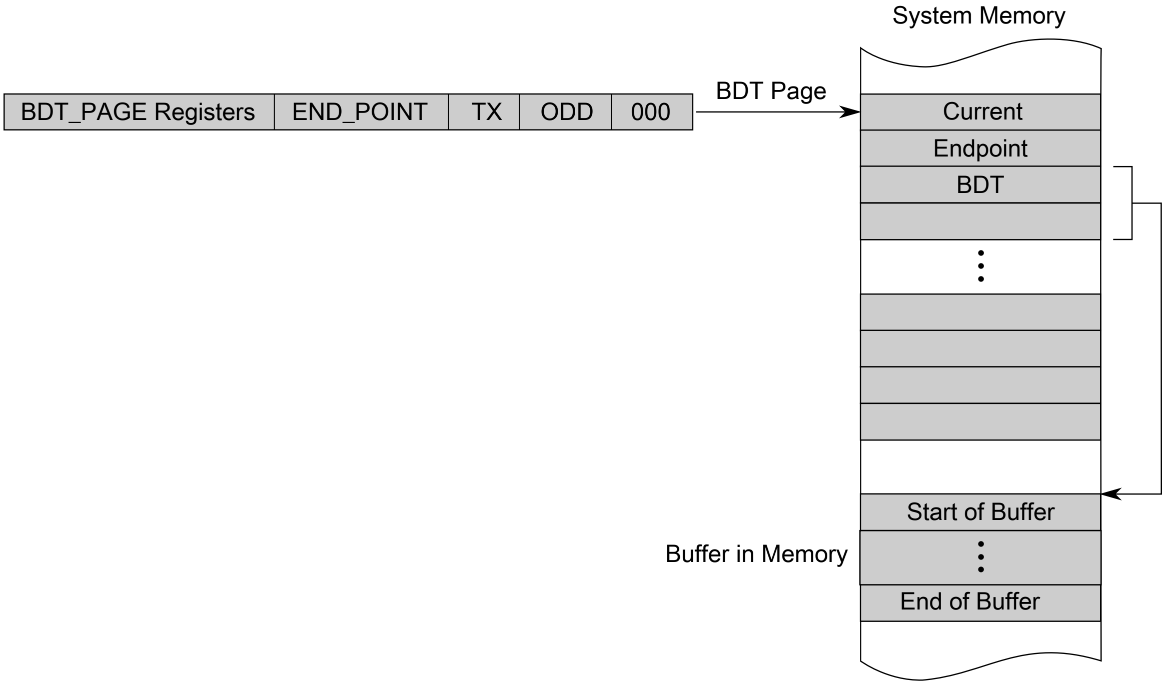


Figure 48-5. Buffer descriptor table

48.3.2 RX vs. TX as a USB peripheral device or USB host

The USBFS core uses software control to switch between two modes of operation:

- USB peripheral device
- USB hosts

In either mode, USB host or USB peripheral device, the same data paths and buffer descriptors are used for the transmission and reception of data. For this reason, a USBFS core-centric nomenclature is used to describe the direction of the data transfer between the USBFS core and USB:

- "RX" (or "receive") describes transfers that move data from USB to memory.
- "TX" (or "transmit") describes transfers that move data from memory to USB.

The following table shows how the data direction corresponds to the USB token type in host and peripheral device applications.

Table 48-1. Data direction for USB host or USB peripheral device

	RX	TX
Device	OUT or SETUP	IN
Host	IN	OUT or SETUP

48.3.3 Addressing BDT entries

An understanding of the addressing mechanism of the Buffer Descriptor Table is useful when accessing endpoint data via USBFS or microprocessor. Some points of interest are:

- The BDT occupies up to 512 bytes of system memory.
- 16 bidirectional endpoints can be supported with a full BDT of 512 bytes.
- 16 bytes are needed for each USB endpoint direction.
- Applications with fewer than 16 endpoints require less RAM to implement the BDT.
- The BDT Page Registers (BDT_PAGE) point to the starting location of the BDT.
- The BDT must be located on a 512-byte boundary in system memory.
- All enabled TX and RX endpoint BD entries are indexed into the BDT to allow easy access via USBFS or MCU core.

When a USB token on an enabled endpoint is received, USBFS uses its integrated DMA controller to interrogate the BDT. USBFS reads the corresponding endpoint BD entry to determine whether it owns the BD and corresponding buffer in system memory.

To compute the entry point into the BDT, the BDT_PAGE registers are concatenated with the current endpoint and the TX and ODD fields to form a 32-bit address. This address mechanism is shown in the following tables:

Table 48-2. BDT Address Calculation

31:24	23:16	15:9	8:5	4	3	2:0
BDT_PAGE_03	BDT_PAGE_02	BDT_PAGE_01[7:1]	Endpoint	TX	ODD	000

Table 48-3. BDT address calculation fields

Field	Description
BDT_PAGE	BDT_PAGE registers in the Control Register Block
ENDPOINT	ENDPOINT field from the USB TOKEN
TX	1 for transmit transfers and 0 for receive transfers
ODD	Maintained within the USBFS SIE. It corresponds to the buffer currently in use. The buffers are used in a ping-pong fashion.

48.3.4 Buffer Descriptors (BDs)

A buffer descriptor provides endpoint buffer control information for USBFS and the processor. The Buffer Descriptors have different meaning based on whether it is USBFS or the processor reading the BD in memory.

The USBFS Controller uses the data stored in the BDs to determine:

- Who owns the buffer in system memory
- Data0 or Data1 PID
- Whether to release ownership upon packet completion
- No address increment (FIFO mode)
- Whether data toggle synchronization is enabled
- How much data is to be transmitted or received
- Where the buffer resides in system memory

While the processor uses the data stored in the BDs to determine:

- Who owns the buffer in system memory
- Data0 or Data1 PID
- The received TOKEN PID
- How much data was transmitted or received
- Where the buffer resides in system memory

The format for the BD is shown in the following figure.

Table 48-4. Buffer descriptor format

31:26	25:16	15:8	7	6	5	4	3	2	1	0
RSVD	BC (10 bits)	RSVD	OWN	DATA0/1	KEEP/ TOK_PID[3]	NINC/ TOK_PID[2]	DTS/ TOK_PID[1]	BDT_STALL/ TOK_PID[0]	0	0
Buffer Address (32-Bits)										

Table 48-5. Buffer descriptor fields

Field	Description
31–26 RSVD	Reserved
25–16 BC	Byte Count Represents the 10-bit byte count. The USBFS SIE changes this field upon the completion of a RX transfer with the byte count of the data received.
15–8 RSVD	Reserved

Table continues on the next page...

Table 48-5. Buffer descriptor fields (continued)

Field	Description
7 OWN	Determines whether the processor or USBFS currently owns the buffer. Except when KEEP=1, the SIE hands ownership back to the processor after completing the token by clearing this bit. This must always be the last byte of the BD that the processor updates when it initializes a BD. 0 The processor has access to the BD. USBFS ignores all other fields in the BD. 1 USBFS has access to the BD. While USBFS owns the BD, the processor should not modify any other fields in the BD.
6 DATA0/1	Defines whether a DATA0 field (DATA0/1=0) or a DATA1 (DATA0/1=1) field was transmitted or received. It is unchanged by USBFS.
5 KEEP/ TOK_PID[3]	This bit has two functions: • KEEP bit—When written by the processor, it serves as the KEEP bit. Typically, this bit is 1 with ISO endpoints feeding a FIFO. The microprocessor is not informed that a token has been processed, the data is simply transferred to or from the FIFO. When KEEP is set, normally the NINC bit is also set to prevent address increment. 0 Allows USBFS to release the BD when a token has been processed. 1 This bit is unchanged by USBFS. Bit 3 of the current token PID is written back to the BD by USBFS. • TOK_PID[3]—If the OWN bit is also set, the BD remains owned by USBFS indefinitely; when written by USB, it serves as the TOK_PID[3] bit. 0 or 1 Bit 3 of the current token PID is written back to the BD by USBFS. Typically, this bit is 1 with ISO endpoints feeding a FIFO. The microprocessor is not informed that a token has been processed, the data is simply transferred to or from the FIFO. When KEEP is set, normally the NINC bit is also set to prevent address increment.
4 NINC/ TOK_PID[2]	No Increment (NINC) Disables the DMA engine address increment. This forces the DMA engine to read or write from the same address. This is useful for endpoints when data needs to be read from or written to a single location such as a FIFO. Typically this bit is set with the KEEP bit for ISO endpoints that are interfacing to a FIFO. 0 USBFS writes bit 2 of the current token PID to the BD. 1 This bit is unchanged by USBFS.
3 DTS/ TOK_PID[1]	Setting this bit enables USBFS to perform Data Toggle Synchronization. • If KEEP=0, bit 1 of the current token PID is written back to the BD. • If KEEP=1, this bit is unchanged by USBFS. 0 Data Toggle Synchronization is disabled. 1 Enables USBFS to perform Data Toggle Synchronization.
2 BDT_STALL TOK_PID[0]	Setting this bit causes USBFS to issue a STALL handshake if a token is received by the SIE that would use the BDT in this location. The BDT is not consumed by the SIE (the owns bit remains set and the rest of the BDT is unchanged) when a BDT_STALL bit is set. • If KEEP=0, bit 0 of the current token PID is written back to the BD. • If KEEP=1, this bit is unchanged by USBFS. 0 No stall issued. 1 The BDT is not consumed by the SIE (the OWN bit remains set and the rest of the BDT is unchanged).

Table continues on the next page...

Table 48-5. Buffer descriptor fields (continued)

Field	Description
	Setting BDT_STALL also causes the corresponding USB_ENDPT*n*[EPSTALL] bit to set. This causes USBOTG to issue a STALL handshake for both directions of the associated endpoint. To clear the stall condition: 1. Clear the associated USB_ENDPT*n*[EPSTALL] bit. 2. Write the BDT to clear OWN and BDT_STALL.
TOK_PID[n]	Bits [5:2] can also represent the current token PID. The current token PID is written back in to the BD by USBFS when a transfer completes. The values written back are the token PID values from the USB specification: • 0x1h for an OUT token. • 0x9h for an IN token. • 0xDh for a SETUP token. In host mode, this field is used to report the last returned PID or a transfer status indication. The possible values returned are: • 0x3h DATA0 • 0xBh DATA1 • 0x2h ACK • 0xEh STALL • 0xAh NAK • 0x0h Bus Timeout • 0xFh Data Error
1–0 Reserved	Reserved, should read as zeroes.
ADDR[31:0]	Address Represents the 32-bit buffer address in system memory. These bits are unchanged by USBFS.

48.3.5 USB transaction

When USBFS transmits or receives data, it computes the BDT address using the address generation shown in "Addressing Buffer Descriptor Entries" table.

If OWN =1, the following process occurs:

1. USBFS reads the BDT.
2. The SIE transfers the data via the DMA to or from the buffer pointed to by the ADDR field of the BD.
3. When the TOKEN is complete, USBFS updates the BDT and, if KEEP=0, changes the OWN bit to 0.
4. The STAT register is updated and the TOK_DNE interrupt is set.
5. When the processor processes the TOK_DNE interrupt, it reads from the status register all the information needed to process the endpoint.
6. At this point, the processor allocates a new BD so that additional USB data can be transmitted or received for that endpoint, and then processes the last BD.

The following figure shows a timeline of how a typical USB token is processed after the BDT is read and OWN=1.

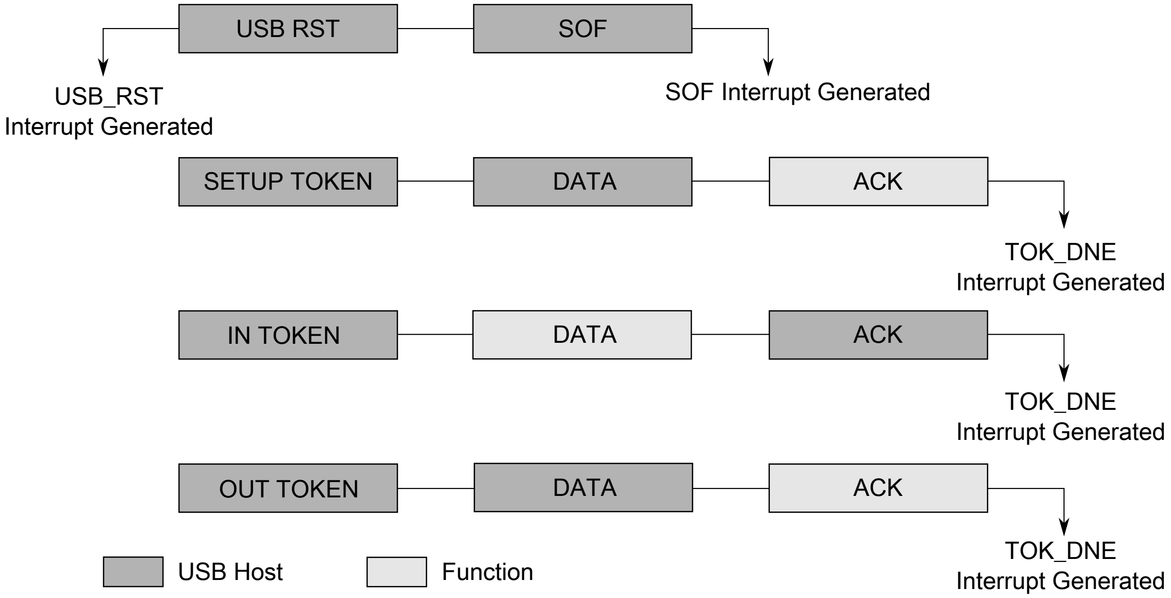

USB RST
SOF
USB_RST Interrupt Generated
SOF Interrupt Generated
SETUP TOKEN
DATA
ACK
TOK_DNE Interrupt Generated
IN TOKEN
DATA
ACK
TOK_DNE Interrupt Generated
OUT TOKEN
DATA
ACK
TOK_DNE Interrupt Generated
USB Host
Function

Figure 48-6. USB token transaction

The USB has two sources for the DMA overrun error:

Memory Latency

The memory latency may be too high and cause the receive FIFO to overflow. This is predominantly a hardware performance issue, usually caused by transient memory access issues.

Oversized Packets

The packet received may be larger than the negotiated *MaxPacket* size. Typically, this is caused by a software bug. For DMA overrun errors due to oversized data packets, the USB specification is ambiguous. It assumes correct software drivers on both sides. NAKing the packet can result in retransmission of the already oversized packet data. Therefore, in response to oversized packets, the USB core continues ACKing the packet for non-isochronous transfers.

Table 48-6. USB responses to DMA overrun errors

Errors due to Memory Latency	Errors due to Oversized Packets
Non-Acknowledgment (NAK) or Bus Timeout (BTO) — See bit 4 in "Error Interrupt Status Register (ERRSTAT)" as appropriate for the class of transaction.	Continues acknowledging (ACKing) the packet for non-isochronous transfers.
—	The data written to memory is clipped to the MaxPacket size so as not to corrupt system memory.

Table continues on the next page...

Table 48-6. USB responses to DMA overrun errors (continued)

Errors due to Memory Latency	Errors due to Oversized Packets
The DMAERR bit is set in the ERRSTAT register for host and device modes of operation. Depending on the values of the INTENB and ERRENB register, the core may assert an interrupt to notify the processor of the DMA error.	Asserts ERRSTAT[DMAERR] ,which can trigger an interrupt and TOKDNE interrupt fires. Note: The TOK_PID field of the BDT is not 1111 because the DMAERR is not due to latency.
• For host mode, the TOKDNE interrupt is generated and the TOK_PID field of the BDT is 1111 to indicate the DMA latency error. Host mode software can decide to retry or move to next scheduled item. • In device mode, the BDT is not written back nor is the TOKDNE interrupt triggered because it is assumed that a second attempt is queued and will succeed in the future.	The packet length field written back to the BDT is the MaxPacket value that represents the length of the clipped data actually written to memory.
From here, the software can decide an appropriate course of action for future transactions such as stalling the endpoint, canceling the transfer, disabling the endpoint, etc.	

48.4 Memory map/Register definitions

This section provides the memory map and detailed descriptions of all USB interface registers.

USB memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4007_2000	Peripheral ID register (USB0_PERID)	8	R	04h	48.4.1/1186
4007_2004	Peripheral ID Complement register (USB0_IDCOMP)	8	R	FBh	48.4.2/1187
4007_2008	Peripheral Revision register (USB0_REV)	8	R	33h	48.4.3/1187
4007_200C	Peripheral Additional Info register (USB0_ADDINFO)	8	R	01h	48.4.4/1188
4007_2010	OTG Interrupt Status register (USB0_OTGISTAT)	8	R/W	00h	48.4.5/1188
4007_2014	OTG Interrupt Control register (USB0_OTGICR)	8	R/W	00h	48.4.6/1189
4007_2018	OTG Status register (USB0_OTGSTAT)	8	R/W	00h	48.4.7/1190
4007_201C	OTG Control register (USB0_OTGCTL)	8	R/W	00h	48.4.8/1191
4007_2080	Interrupt Status register (USB0_ISTAT)	8	R/W	00h	48.4.9/1192
4007_2084	Interrupt Enable register (USB0_INTEN)	8	R/W	00h	48.4.10/ 1193
4007_2088	Error Interrupt Status register (USB0_ERRSTAT)	8	R/W	00h	48.4.11/ 1195
4007_208C	Error Interrupt Enable register (USB0_ERREN)	8	R/W	00h	48.4.12/ 1196
4007_2090	Status register (USB0_STAT)	8	R	00h	48.4.13/ 1197

Table continues on the next page...

USB memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4007_2094	Control register (USB0_CTL)	8	R/W	00h	48.4.14/ 1198
4007_2098	Address register (USB0_ADDR)	8	R/W	00h	48.4.15/ 1199
4007_209C	BDT Page register 1 (USB0_BDTPAGE1)	8	R/W	00h	48.4.16/ 1200
4007_20A0	Frame Number register Low (USB0_FRMNUML)	8	R/W	00h	48.4.17/ 1200
4007_20A4	Frame Number register High (USB0_FRMNUMH)	8	R/W	00h	48.4.18/ 1201
4007_20A8	Token register (USB0_TOKEN)	8	R/W	00h	48.4.19/ 1201
4007_20AC	SOF Threshold register (USB0_SOFTHLD)	8	R/W	00h	48.4.20/ 1202
4007_20B0	BDT Page Register 2 (USB0_BDTPAGE2)	8	R/W	00h	48.4.21/ 1203
4007_20B4	BDT Page Register 3 (USB0_BDTPAGE3)	8	R/W	00h	48.4.22/ 1203
4007_20C0	Endpoint Control register (USB0_ENDPT0)	8	R/W	00h	48.4.23/ 1204
4007_20C4	Endpoint Control register (USB0_ENDPT1)	8	R/W	00h	48.4.23/ 1204
4007_20C8	Endpoint Control register (USB0_ENDPT2)	8	R/W	00h	48.4.23/ 1204
4007_20CC	Endpoint Control register (USB0_ENDPT3)	8	R/W	00h	48.4.23/ 1204
4007_20D0	Endpoint Control register (USB0_ENDPT4)	8	R/W	00h	48.4.23/ 1204
4007_20D4	Endpoint Control register (USB0_ENDPT5)	8	R/W	00h	48.4.23/ 1204
4007_20D8	Endpoint Control register (USB0_ENDPT6)	8	R/W	00h	48.4.23/ 1204
4007_20DC	Endpoint Control register (USB0_ENDPT7)	8	R/W	00h	48.4.23/ 1204
4007_20E0	Endpoint Control register (USB0_ENDPT8)	8	R/W	00h	48.4.23/ 1204
4007_20E4	Endpoint Control register (USB0_ENDPT9)	8	R/W	00h	48.4.23/ 1204
4007_20E8	Endpoint Control register (USB0_ENDPT10)	8	R/W	00h	48.4.23/ 1204
4007_20EC	Endpoint Control register (USB0_ENDPT11)	8	R/W	00h	48.4.23/ 1204
4007_20F0	Endpoint Control register (USB0_ENDPT12)	8	R/W	00h	48.4.23/ 1204

Table continues on the next page...

USB memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4007_20F4	Endpoint Control register (USB0_ENDPT13)	8	R/W	00h	48.4.23/ 1204
4007_20F8	Endpoint Control register (USB0_ENDPT14)	8	R/W	00h	48.4.23/ 1204
4007_20FC	Endpoint Control register (USB0_ENDPT15)	8	R/W	00h	48.4.23/ 1204
4007_2100	USB Control register (USB0_USBCTRL)	8	R/W	C0h	48.4.24/ 1205
4007_2104	USB OTG Observe register (USB0_OBSERVE)	8	R	50h	48.4.25/ 1206
4007_2108	USB OTG Control register (USB0_CONTROL)	8	R/W	00h	48.4.26/ 1206
4007_210C	USB Transceiver Control register 0 (USB0_USBTRC0)	8	R/W	00h	48.4.27/ 1207
4007_2114	Frame Adjust Register (USB0_USBFRMADJUST)	8	R/W	00h	48.4.28/ 1208

48.4.1 Peripheral ID register (USBx_PERID)

Reads back the value of 0x04. This value is defined for the USB peripheral.

Address: 4007_2000h base + 0h offset = 4007_2000h

Bit	7	6	5	4	3	2	1	0
Read	0		ID					
Write								
Reset	0	0	0	0	0	1	0	0

USBx_PERID field descriptions

Field	Description
7–6 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
ID	Peripheral Identification This field always reads 0x4h.

48.4.2 Peripheral ID Complement register (USBx_IDCOMP)

Reads back the complement of the Peripheral ID register. For the USB peripheral, the value is 0xFB.

Address: 4007_2000h base + 4h offset = 4007_2004h

Bit	7	6	5	4	3	2	1	0
Read	1		NID					
Write								
Reset	1	1	1	1	1	0	1	1

USBx_IDCOMP field descriptions

Field	Description
7–6 Reserved	This field is reserved. This read-only field is reserved and always has the value 1.
NID	Ones' complement of PERID[ID]. bits.

48.4.3 Peripheral Revision register (USBx_REV)

Contains the revision number of the USB module.

Address: 4007_2000h base + 8h offset = 4007_2008h

Bit	7	6	5	4	3	2	1	0
Read	REV							
Write								
Reset	0	0	1	1	0	0	1	1

USBx_REV field descriptions

Field	Description
REV	Revision Indicates the revision number of the USB Core.

48.4.4 Peripheral Additional Info register (USBx_ADDINFO)

Reads back the value of the Host Enable bit.

Address: 4007_2000h base + Ch offset = 4007_200Ch

Bit	7	6	5	4	3	2	1	0
Read	0					0		IEHOST
Write								
Reset	0	0	0	0	0	0	0	1

USBx_ADDINFO field descriptions

Field	Description
7–3 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
2–1 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
0 IEHOST	This bit is set if host mode is enabled.

48.4.5 OTG Interrupt Status register (USBx_OTGISTAT)

Records changes of the ID sense and VBUS signals. Software can read this register to determine the event that triggers an interrupt. Only bits that have changed since the last software read are set. Writing a one to a bit clears the associated interrupt.

Address: 4007_2000h base + 10h offset = 4007_2010h

Bit	7	6	5	4	3	2	1	0
Read	Reserved	ONEMSEC	LINE_ STATE_ CHG	0	Reserved	Reserved	0	Reserved
Write	w1c	w1c	w1c		w1c	w1c		w1c
Reset	0	0	0	0	0	0	0	0

USBx_OTGISTAT field descriptions

Field	Description
7 Reserved	This field is reserved. Software should not change the value of this bitfield.
6 ONEMSEC	This bit is set when the 1 millisecond timer expires. This bit stays asserted until cleared by software. The interrupt must be serviced every millisecond to avoid losing 1msec counts.

Table continues on the next page...

USBx_OTGISTAT field descriptions (continued)

Field	Description
5 LINE_STATE_ CHG	This interrupt is set when the USB line state (CTL[SE0] and CTL[JSTATE] bits) are stable without change for 1 millisecond, and the value of the line state is different from the last time when the line state was stable. It is set on transitions between SE0 and J-state, SE0 and K-state, and J-state and K-state. Changes in J-state while SE0 is true do not cause an interrupt. This interrupt can be used in detecting Reset, Resume, Connect, and Data Line Pulse signaling.
4 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
3 Reserved	This field is reserved. Software should not change the value of this bitfield.
2 Reserved	This field is reserved. Software should not change the value of this bitfield.
1 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
0 Reserved	This field is reserved. Software should not change the value of this bitfield.

48.4.6 OTG Interrupt Control register (USBx_OTGICR)

Enables the corresponding interrupt status bits defined in the OTG Interrupt Status Register.

Address: 4007_2000h base + 14h offset = 4007_2014h

Bit	7	6	5	4	3	2	1	0
Read	Reserved	ONEMSECEN	LINESTATEEN	0	Reserved	Reserved	0	Reserved
Write				▨			▨	
Reset	0	0	0	0	0	0	0	0

USBx_OTGICR field descriptions

Field	Description
7 Reserved	Software must not change the value of this bitfield. This field is reserved. 0 The ID interrupt is disabled 1 The ID interrupt is enabled
6 ONEMSECEN	One Millisecond Interrupt Enable 0 Diables the 1ms timer interrupt. 1 Enables the 1ms timer interrupt.
5 LINESTATEEN	Line State Change Interrupt Enable 0 Disables the LINE_STAT_CHG interrupt. 1 Enables the LINE_STAT_CHG interrupt.

Table continues on the next page...

USBx_OTGICR field descriptions (continued)

Field	Description
4 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
3 Reserved	Software must not change the value of this bitfield. This field is reserved. 0 Disables the SESSVLDCHG interrupt. 1 Enables the SESSVLDCHG interrupt.
2 Reserved	Software must not change the value of this bitfield. This field is reserved. 0 Disables the B_SESS_CHG interrupt. 1 Enables the B_SESS_CHG interrupt.
1 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
0 Reserved	Software must not change the value of this bitfield. This field is reserved. 0 Disables the AVBUSCHG interrupt. 1 Enables the AVBUSCHG interrupt.

48.4.7 OTG Status register (USBx_OTGSTAT)

Displays the actual value from the one-millisecond timer and line stable logic.

Address: 4007_2000h base + 18h offset = 4007_2018h

Bit	7	6	5	4
Read	0	ONEMSECEN	LINESTATESTABLE	0
Write				
Reset	0	0	0	0

Bit	3	2	1	0
Read	0	0	0	0
Write				
Reset	0	0	0	0

USBx_OTGSTAT field descriptions

Field	Description
7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0. 0 Indicates a Type A cable is plugged into the USB connector. 1 Indicates no cable is attached or a Type B cable is plugged into the USB connector.

Table continues on the next page...

USBx_OTGSTAT field descriptions (continued)

Field	Description
6 ONEMSECEN	This bit is reserved for the 1ms count, but it is not useful to software.
5 LINESTATESTABLE	Indicates that the internal signals that control the LINE_STATE_CHG field of OTGISTAT are stable for at least 1 ms. This bit is used to provide a hardware debounce of the linestate in detection of Connect, Disconnect and Resume signaling. First read LINE_STATE_CHG field and then read this field. If this field reads as 1, then the value of LINE_STATE_CHG can be considered stable. 0 The LINE_STAT_CHG bit is not yet stable. 1 The LINE_STAT_CHG bit has been debounced and is stable.
4 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
3 Reserved	This field is reserved. This read-only field is reserved and always has the value 0. 0 The VBUS voltage is below the B session valid threshold 1 The VBUS voltage is above the B session valid threshold.
2 Reserved	This field is reserved. This read-only field is reserved and always has the value 0. 0 The VBUS voltage is above the B session end threshold. 1 The VBUS voltage is below the B session end threshold.
1 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
0 Reserved	This field is reserved. This read-only field is reserved and always has the value 0. 0 The VBUS voltage is below the A VBUS Valid threshold. 1 The VBUS voltage is above the A VBUS Valid threshold.

48.4.8 OTG Control register (USBx_OTGCTL)

Controls the operation of VBUS and Data Line termination resistors.

Address: 4007_2000h base + 1Ch offset = 4007_201Ch

Bit	7	6	5	4	3	2	1	0
Read	DPHIGH	0	DPLOW	DMLOW	0	OTGEN	0	
Write								
Reset	0	0	0	0	0	0	0	0

USBx_OTGCTL field descriptions

Field	Description
7 DPHIGH	D+ Data Line pullup resistor enable 0 D+ pullup resistor is not enabled 1 D+ pullup resistor is enabled

Table continues on the next page...

USBx_OTGCTL field descriptions (continued)

Field	Description
6 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
5 DPLOW	D+ Data Line pull-down resistor enable This bit should always be enabled together with bit 4 (DMLOW) 0 D+ pulldown resistor is not enabled. 1 D+ pulldown resistor is enabled.
4 DMLOW	D– Data Line pull-down resistor enable 0 D- pulldown resistor is not enabled. 1 D- pulldown resistor is enabled.
3 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
2 OTGEN	On-The-Go pullup/pulldown resistor enable 0 If USB_EN is 1 and HOST_MODE is 0 in the Control Register (CTL), then the D+ Data Line pull-up resistors are enabled. If HOST_MODE is 1 the D+ and D– Data Line pull-down resistors are engaged. 1 The pull-up and pull-down controls in this register are used.
Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

48.4.9 Interrupt Status register (USBx_ISTAT)

Contains fields for each of the interrupt sources within the USB Module. Each of these fields are qualified with their respective interrupt enable bits. All fields of this register are logically OR'd together along with the OTG Interrupt Status Register (OTGSTAT) to form a single interrupt source for the processor's interrupt controller. After an interrupt bit has been set it may only be cleared by writing a one to the respective interrupt bit. This register contains the value of 0x00 after a reset.

Address: 4007_2000h base + 80h offset = 4007_2080h

Bit	7	6	5	4	3	2	1	0
Read	STALL	ATTACH	RESUME	SLEEP	TOKDNE	SOFTOK	ERROR	USBRST
Write	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c
Reset	0	0	0	0	0	0	0	0

USBx_ISTAT field descriptions

Field	Description
7 STALL	Stall Interrupt In Device mode this bit is asserted when a STALL handshake is sent by the SIE.

Table continues on the next page...

USBx_ISTAT field descriptions (continued)

Field	Description
	In Host mode this bit is set when the USB Module detects a STALL acknowledge during the handshake phase of a USB transaction.This interrupt can be used to determine whether the last USB transaction was completed successfully or stalled.
6 ATTACH	Attach Interrupt This field is set when the USB Module detects an attach of a USB device. This field is only valid if CTL[HOSTMODEEN]=1. This interrupt signifies that a peripheral is now present and must be configured; it is asserted if there have been no transitions on the USB for 2.5 μs and the current bus state is not SE0." 0 No Attach is detected since the last time the ATTACH bit was cleared. 1 A peripheral is now present and must be configured (a stable non-SE0 state is detected for more than 2.5 μs).
5 RESUME	This bit is set when a K-state is observed on the DP/DM signals for 2.5 μs. When not in suspend mode this interrupt must be disabled.
4 SLEEP	This bit is set when the USB Module detects a constant idle on the USB bus for 3 ms. The sleep timer is reset by activity on the USB bus.
3 TOKDNE	This bit is set when the current token being processed has completed. The processor must immediately read the STATUS (STAT) register to determine the EndPoint and BD used for this token. Clearing this bit (by writing a one) causes STAT to be cleared or the STAT holding register to be loaded into the STAT register.
2 SOFTOK	This bit is set when the USB Module receives a Start Of Frame (SOF) token. In Host mode this field is set when the SOF threshold is reached, so that software can prepare for the next SOF.
1 ERROR	This bit is set when any of the error conditions within Error Interrupt Status (ERRSTAT) register occur. The processor must then read the ERRSTAT register to determine the source of the error.
0 USBRST	This bit is set when the USB Module has decoded a valid USB reset. This informs the processor that it should write 0x00 into the address register and enable endpoint 0. USBRST is set after a USB reset has been detected for 2.5 microseconds. It is not asserted again until the USB reset condition has been removed and then reasserted.

48.4.10 Interrupt Enable register (USBx_INTEN)

Contains enable fields for each of the interrupt sources within the USB Module. Setting any of these bits enables the respective interrupt source in the ISTAT register. This register contains the value of 0x00 after a reset.

Address: 4007_2000h base + 84h offset = 4007_2084h

Bit	7	6	5	4	3	2	1	0
Read Write	STALLEN	ATTACHEN	RESUMEEN	SLEEPEN	TOKDNEEN	SOFTOKEN	ERROREN	USBRSTEN
Reset	0	0	0	0	0	0	0	0

USBx_INTEN field descriptions

Field	Description
7 STALLEN	STALL Interrupt Enable

Table continues on the next page...

USBx_INTEN field descriptions (continued)

Field	Description
	0 Diasbles the STALL interrupt. 1 Enables the STALL interrupt.
6 ATTACHEN	ATTACH Interrupt Enable 0 Disables the ATTACH interrupt. 1 Enables the ATTACH interrupt.
5 RESUMEEN	RESUME Interrupt Enable 0 Disables the RESUME interrupt. 1 Enables the RESUME interrupt.
4 SLEEPEN	SLEEP Interrupt Enable 0 Disables the SLEEP interrupt. 1 Enables the SLEEP interrupt.
3 TOKDNEEN	TOKDNE Interrupt Enable 0 Disables the TOKDNE interrupt. 1 Enables the TOKDNE interrupt.
2 SOFTOKEN	SOFTOK Interrupt Enable 0 Disbles the SOFTOK interrupt. 1 Enables the SOFTOK interrupt.
1 ERROREN	ERROR Interrupt Enable 0 Disables the ERROR interrupt. 1 Enables the ERROR interrupt.
0 USBRSTEN	USBRST Interrupt Enable 0 Disables the USBRST interrupt. 1 Enables the USBRST interrupt.

48.4.11 Error Interrupt Status register (USBx_ERRSTAT)

Contains enable bits for each of the error sources within the USB Module. Each of these bits are qualified with their respective error enable bits. All bits of this register are logically OR'd together and the result placed in the ERROR bit of the ISTAT register. After an interrupt bit has been set it may only be cleared by writing a one to the respective interrupt bit. Each bit is set as soon as the error condition is detected. Therefore, the interrupt does not typically correspond with the end of a token being processed. This register contains the value of 0x00 after a reset.

Address: 4007_2000h base + 88h offset = 4007_2088h

Bit	7	6	5	4	3	2	1	0
Read	BTSERR	0	DMAERR	BTOERR	DFN8	CRC16	CRC5EOF	PIDERR
Write	w1c		w1c	w1c	w1c	w1c	w1c	w1c
Reset	0	0	0	0	0	0	0	0

USBx_ERRSTAT field descriptions

Field	Description
7 BTSERR	This bit is set when a bit stuff error is detected. If set, the corresponding packet is rejected due to the error.
6 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
5 DMAERR	This bit is set if the USB Module has requested a DMA access to read a new BDT but has not been given the bus before it needs to receive or transmit data. If processing a TX transfer this would cause a transmit data underflow condition. If processing a RX transfer this would cause a receive data overflow condition. This interrupt is useful when developing device arbitration hardware for the microprocessor and the USB module to minimize bus request and bus grant latency. This bit is also set if a data packet to or from the host is larger than the buffer size allocated in the BDT. In this case the data packet is truncated as it is put in buffer memory.
4 BTOERR	This bit is set when a bus turnaround timeout error occurs. The USB module contains a bus turnaround timer that keeps track of the amount of time elapsed between the token and data phases of a SETUP or OUT TOKEN or the data and handshake phases of a IN TOKEN. If more than 16 bit times are counted from the previous EOP before a transition from IDLE, a bus turnaround timeout error occurs.
3 DFN8	This bit is set if the data field received was not 8 bits in length. USB Specification 1.0 requires that data fields be an integral number of bytes. If the data field was not an integral number of bytes, this bit is set.
2 CRC16	This bit is set when a data packet is rejected due to a CRC16 error.
1 CRC5EOF	This error interrupt has two functions. When the USB Module is operating in peripheral mode (HOSTMODEEN=0), this interrupt detects CRC5 errors in the token packets generated by the host. If set the token packet was rejected due to a CRC5 error. When the USB Module is operating in host mode (HOSTMODEEN=1), this interrupt detects End Of Frame (EOF) error conditions. This occurs when the USB Module is transmitting or receiving data and the SOF counter reaches zero. This interrupt is useful when developing USB packet scheduling software to ensure that no USB transactions cross the start of the next frame.
0 PIDERR	This bit is set when the PID check field fails.

48.4.12 Error Interrupt Enable register (USBx_ERREN)

Contains enable bits for each of the error interrupt sources within the USB module. Setting any of these bits enables the respective interrupt source in ERRSTAT. Each bit is set as soon as the error condition is detected. Therefore, the interrupt does not typically correspond with the end of a token being processed. This register contains the value of 0x00 after a reset.

Address: 4007_2000h base + 8Ch offset = 4007_208Ch

Bit	7	6	5	4	3	2	1	0
Read	BTSERREN	0	DMAERREN	BTOERREN	DFN8EN	CRC16EN	CRC5EOFEN	PIDERREN
Write		▨						
Reset	0	0	0	0	0	0	0	0

USBx_ERREN field descriptions

Field	Description
7 BTSERREN	BTSERR Interrupt Enable 0 Disables the BTSERR interrupt. 1 Enables the BTSERR interrupt.
6 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
5 DMAERREN	DMAERR Interrupt Enable 0 Disables the DMAERR interrupt. 1 Enables the DMAERR interrupt.
4 BTOERREN	BTOERR Interrupt Enable 0 Disables the BTOERR interrupt. 1 Enables the BTOERR interrupt.
3 DFN8EN	DFN8 Interrupt Enable 0 Disables the DFN8 interrupt. 1 Enables the DFN8 interrupt.
2 CRC16EN	CRC16 Interrupt Enable 0 Disables the CRC16 interrupt. 1 Enables the CRC16 interrupt.
1 CRC5EOFEN	CRC5/EOF Interrupt Enable 0 Disables the CRC5/EOF interrupt. 1 Enables the CRC5/EOF interrupt.
0 PIDERREN	PIDERR Interrupt Enable 0 Disables the PIDERR interrupt. 1 Enters the PIDERR interrupt.

48.4.13 Status register (USBx_STAT)

Reports the transaction status within the USB module. When the processor's interrupt controller has received a TOKDNE, interrupt the Status Register must be read to determine the status of the previous endpoint communication. The data in the status register is valid when TOKDNE interrupt is asserted. The Status register is actually a read window into a status FIFO maintained by the USB module. When the USB module uses a BD, it updates the Status register. If another USB transaction is performed before the TOKDNE interrupt is serviced, the USB module stores the status of the next transaction in the STAT FIFO. Thus STAT is actually a four byte FIFO that allows the processor core to process one transaction while the SIE is processing the next transaction. Clearing the TOKDNE bit in the ISTAT register causes the SIE to update STAT with the contents of the next STAT value. If the data in the STAT holding register is valid, the SIE immediately reasserts to TOKDNE interrupt.

Address: 4007_2000h base + 90h offset = 4007_2090h

Bit	7	6	5	4	3	2	1	0
Read	ENDP				TX	ODD	0	
Write								
Reset	0	0	0	0	0	0	0	0

USBx_STAT field descriptions

Field	Description
7–4 ENDP	This four-bit field encodes the endpoint address that received or transmitted the previous token. This allows the processor core to determine the BDT entry that was updated by the last USB transaction.
3 TX	Transmit Indicator 0 The most recent transaction was a receive operation. 1 The most recent transaction was a transmit operation.
2 ODD	This bit is set if the last buffer descriptor updated was in the odd bank of the BDT.
Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

48.4.14 Control register (USBx_CTL)

Provides various control and configuration information for the USB module.

Address: 4007_2000h base + 94h offset = 4007_2094h

Bit	7	6	5	4
Read Write	JSTATE	SE0	TXSUSPENDTOKENBUSY	RESET
Reset	0	0	0	0

Bit	3	2	1	0
Read Write	HOSTMODEEN	RESUME	ODDRST	USBENSOFEN
Reset	0	0	0	0

USBx_CTL field descriptions

Field	Description
7 JSTATE	Live USB differential receiver JSTATE signal The polarity of this signal is affected by the current state of LSEN .
6 SE0	Live USB Single Ended Zero signal
5 TXSUSPENDTOKENBUSY	In Host mode, TOKEN_BUSY is set when the USB module is busy executing a USB token. Software must not write more token commands to the Token Register when TOKEN_BUSY is set. Software should check this field before writing any tokens to the Token Register to ensure that token commands are not lost. In Device mode, TXD_SUSPEND is set when the SIE has disabled packet transmission and reception. Clearing this bit allows the SIE to continue token processing. This bit is set by the SIE when a SETUP Token is received allowing software to dequeue any pending packet transactions in the BDT before resuming token processing.
4 RESET	Setting this bit enables the USB Module to generate USB reset signaling. This allows the USB Module to reset USB peripherals. This control signal is only valid in Host mode (HOSTMODEEN=1). Software must set RESET to 1 for the required amount of time and then clear it to 0 to end reset signaling. 1.0.
3 HOSTMODEEN	When set to 1, this bit enables the USB Module to operate in Host mode. In host mode, the USB module performs USB transactions under the programmed control of the host processor.
2 RESUME	When set to 1 this bit enables the USB Module to execute resume signaling. This allows the USB Module to perform remote wake-up. Software must set RESUME to 1 for the required amount of time and then clear it to 0. If the HOSTMODEEN bit is set, the USB module appends a Low Speed End of Packet to the Resume signaling when the RESUME bit is cleared.
1 ODDRST	Setting this bit to 1 resets all the BDT ODD ping/pong fields to 0, which then specifies the EVEN BDT bank.
0 USBENSOFEN	USB Enable Setting this bit enables the USB-FS to operate; clearing it disables the USB-FS. Setting the bit causes the SIE to reset all of its ODD bits to the BDTs. Therefore, setting this bit resets much of the logic in the SIE. When host mode is enabled, clearing this bit causes the SIE to stop sending SOF tokens.

Table continues on the next page...

USBx_CTL field descriptions (continued)

Field	Description
	0 Disables the USB Module. 1 Enables the USB Module.

48.4.15 Address register (USBx_ADDR)

Holds the unique USB address that the USB module decodes when in Peripheral mode (HOSTMODEEN=0). When operating in Host mode (HOSTMODEEN=1) the USB module transmits this address with a TOKEN packet. This enables the USB module to uniquely address any USB peripheral. In either mode, CTL[USBENSOFEN] must be 1. The Address register is reset to 0x00 after the reset input becomes active or the USB module decodes a USB reset signal. This action initializes the Address register to decode address 0x00 as required by the USB specification.

Address: 4007_2000h base + 98h offset = 4007_2098h

Bit	7	6	5	4	3	2	1	0
Read Write	LSEN	ADDR						
Reset	0	0	0	0	0	0	0	0

USBx_ADDR field descriptions

Field	Description
7 LSEN	Low Speed Enable bit Informs the USB module that the next token command written to the token register must be performed at low speed. This enables the USB module to perform the necessary preamble required for low-speed data transmissions.
ADDR	USB Address Defines the USB address that the USB module decodes in peripheral mode, or transmits when in host mode.

48.4.16 BDT Page register 1 (USBx_BDTPAGE1)

Provides address bits 15 through 9 of the base address where the current Buffer Descriptor Table (BDT) resides in system memory. See Buffer Descriptor Table. The 32-bit BDT Base Address is always aligned on 512-byte boundaries, so bits 8 through 0 of the base address are always zero.

Address: 4007_2000h base + 9Ch offset = 4007_209Ch

Bit	7	6	5	4	3	2	1	0
Read	BDTBA							0
Write								▨
Reset	0	0	0	0	0	0	0	0

USBx_BDTPAGE1 field descriptions

Field	Description
7–1 BDTBA	Provides address bits 15 through 9 of the BDT base address.
0 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

48.4.17 Frame Number register Low (USBx_FRMNUML)

The Frame Number registers (low and high) contain the 11-bit frame number. These registers are updated with the current frame number whenever a SOF TOKEN is received.

Address: 4007_2000h base + A0h offset = 4007_20A0h

Bit	7	6	5	4	3	2	1	0
Read Write	FRM[7:0]							
Reset	0	0	0	0	0	0	0	0

USBx_FRMNUML field descriptions

Field	Description
FRM[7:0]	This 8-bit field and the 3-bit field in the Frame Number Register High are used to compute the address where the current Buffer Descriptor Table (BDT) resides in system memory.

48.4.18 Frame Number register High (USBx_FRMNUMH)

The Frame Number registers (low and high) contain the 11-bit frame number. These registers are updated with the current frame number whenever a SOF TOKEN is received.

Address: 4007_2000h base + A4h offset = 4007_20A4h

Bit	7	6	5	4	3	2	1	0
Read	0					FRM[10:8]		
Write	▨							
Reset	0	0	0	0	0	0	0	0

USBx_FRMNUMH field descriptions

Field	Description
7–3 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
FRM[10:8]	This 3-bit field and the 8-bit field in the Frame Number Register Low are used to compute the address where the current Buffer Descriptor Table (BDT) resides in system memory.

48.4.19 Token register (USBx_TOKEN)

Used to initiate USB transactions when in host mode (HOSTMODEEN=1). When the software needs to execute a USB transaction to a peripheral, it writes the TOKEN type and endpoint to this register. After this register has been written, the USB module begins the specified USB transaction to the address contained in the address register. The processor core must always check that the TOKEN_BUSY bit in the control register is not 1 before writing to the Token Register. This ensures that the token commands are not overwritten before they can be executed. The address register and endpoint control register 0 are also used when performing a token command and therefore must also be written before the Token Register. The address register is used to select the USB peripheral address transmitted by the token command. The endpoint control register determines the handshake and retry policies used during the transfer.

Address: 4007_2000h base + A8h offset = 4007_20A8h

Bit	7	6	5	4	3	2	1	0
Read Write	TOKENPID				TOKENENDPT			
Reset	0	0	0	0	0	0	0	0

USBx_TOKEN field descriptions

Field	Description
7–4 TOKENPID	Contains the token type executed by the USB module. 0001 OUT Token. USB Module performs an OUT (TX) transaction. 1001 IN Token. USB Module performs an In (RX) transaction. 1101 SETUP Token. USB Module performs a SETUP (TX) transaction
TOKENENDPT	Holds the Endpoint address for the token command. The four bit value written must be a valid endpoint.

48.4.20 SOF Threshold register (USBx_SOFTHLD)

The SOF Threshold Register is used only in Host mode (HOSTMODEEN=1). When in Host mode, the 14-bit SOF counter counts the interval between SOF frames. The SOF must be transmitted every 1ms so therefore the SOF counter is loaded with a value of 12000. When the SOF counter reaches zero, a Start Of Frame (SOF) token is transmitted.

The SOF threshold register is used to program the number of USB byte times before the SOF to stop initiating token packet transactions. This register must be set to a value that ensures that other packets are not actively being transmitted when the SOF time counts to zero. When the SOF counter reaches the threshold value, no more tokens are transmitted until after the SOF has been transmitted.

The value programmed into the threshold register must reserve enough time to ensure the worst case transaction completes. In general the worst case transaction is an IN token followed by a data packet from the peripheral device followed by the response from the host. The actual time required is a function of the maximum packet size on the bus.

Typical values for the SOF threshold are:

- 64-byte packets=74;
- 32-byte packets=42;
- 16-byte packets=26;
- 8-byte packets=18.

Address: 4007_2000h base + ACh offset = 4007_20ACh

Bit	7	6	5	4	3	2	1	0
Read Write	CNT							
Reset	0	0	0	0	0	0	0	0

USBx_SOFTHLD field descriptions

Field	Description
CNT	Represents the SOF count threshold in byte times.

48.4.21 BDT Page Register 2 (USBx_BDTPAGE2)

Contains an 8-bit value used to compute the address where the current Buffer Descriptor Table (BDT) resides in system memory. See Buffer Descriptor Table.

Address: 4007_2000h base + B0h offset = 4007_20B0h

Bit	7	6	5	4	3	2	1	0
Read Write	BDTBA							
Reset	0	0	0	0	0	0	0	0

USBx_BDTPAGE2 field descriptions

Field	Description
BDTBA	Provides address bits 23 through 16 of the BDT base address that defines the location of Buffer Descriptor Table resides in system memory.

48.4.22 BDT Page Register 3 (USBx_BDTPAGE3)

Contains an 8-bit value used to compute the address where the current Buffer Descriptor Table (BDT) resides in system memory. See Buffer Descriptor Table.

Address: 4007_2000h base + B4h offset = 4007_20B4h

Bit	7	6	5	4	3	2	1	0
Read Write	BDTBA							
Reset	0	0	0	0	0	0	0	0

USBx_BDTPAGE3 field descriptions

Field	Description
BDTBA	Provides address bits 31 through 24 of the BDT base address that defines the location of Buffer Descriptor Table resides in system memory.

48.4.23 Endpoint Control register (USBx_ENDPT*n*)

Contains the endpoint control bits for each of the 16 endpoints available within the USB module for a decoded address. The format for these registers is shown in the following figure. Endpoint 0 (ENDPT0) is associated with control pipe 0, which is required for all USB functions. Therefore, after a USBRST interrupt occurs the processor core should set ENDPT0 to contain 0x0D.

In Host mode ENDPT0 is used to determine the handshake, retry and low speed characteristics of the host transfer. For Control, Bulk and Interrupt transfers, the EPHSHK bit should be 1. For Isochronous transfers it should be 0. Common values to use for ENDPT0 in host mode are 0x4D for Control, Bulk, and Interrupt transfers, and 0x4C for Isochronous transfers.

The three bits EPCTLDIS, EPRXEN, and EPTXEN define if an endpoint is enabled and define the direction of the endpoint. The endpoint enable/direction control is defined in the following table.

Table 48-7. Endpoint enable and direction control

EPCTLDIS	EPRXEN	EPTXEN	Endpoint enable/direction control
X	0	0	Disable endpoint
X	0	1	Enable endpoint for Tx transfers only
X	1	0	Enable endpoint for Rx transfers only
1	1	1	Enable endpoint for Rx and Tx transfers
0	1	1	Enable Endpoint for RX and TX as well as control (SETUP) transfers.

Address: 4007_2000h base + C0h offset + (4d × i), where i=0d to 15d

Bit	7	6	5	4	3	2	1	0
Read	HOSTWOHUB	RETRYDIS	0	EPCTLDIS	EPRXEN	EPTXEN	EPSTALL	EPHSHK
Write			▨					
Reset	0	0	0	0	0	0	0	0

USBx_ENDPT*n* field descriptions

Field	Description
7 HOSTWOHUB	Host without a hub This is a Host mode only field and is present in the control register for endpoint 0 (ENDPT0) only.

Table continues on the next page...

USBx_ENDPT*n* field descriptions (continued)

Field	Description
	0 Low-speed device connected to Host through a hub. PRE_PID will be generated as required. 1 Low-speed device directly connected. No hub, or no low-speed device attached.
6 RETRYDIS	This is a Host mode only bit and is present in the control register for endpoint 0 (ENDPT0) only. When set this bit causes the host to not retry NAK'ed (Negative Acknowledgement) transactions. When a transaction is NAKed, the BDT PID field is updated with the NAK PID, and the TOKEN_DNE interrupt is set. When this bit is cleared, NAKed transactions are retried in hardware. This bit must be set when the host is attempting to poll an interrupt endpoint.
5 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
4 EPCTLDIS	This bit, when set, disables control (SETUP) transfers. When cleared, control transfers are enabled. This applies if and only if the EPRXEN and EPTXEN bits are also set. See Table 48-7
3 EPRXEN	This bit, when set, enables the endpoint for RX transfers. See Table 48-7
2 EPTXEN	This bit, when set, enables the endpoint for TX transfers. See Table 48-7
1 EPSTALL	When set this bit indicates that the endpoint is stalled. This bit has priority over all other control bits in this register, but it is only valid if EPTXEN=1 or EPRXEN=1. Any access to this endpoint causes the USB Module to return a STALL handshake. After an endpoint is stalled it requires intervention from the Host Controller.
0 EPHSHK	When set this bit enables an endpoint to perform handshaking during a transaction to this endpoint. This bit is generally 1 unless the endpoint is Isochronous.

48.4.24 USB Control register (USBx_USBCTRL)

Address: 4007_2000h base + 100h offset = 4007_2100h

Bit	7	6	5	4	3	2	1	0
Read	SUSP	PDE	0					
Write								
Reset	1	1	0	0	0	0	0	0

USBx_USBCTRL field descriptions

Field	Description
7 SUSP	Places the USB transceiver into the suspend state. 0 USB transceiver is not in suspend state. 1 USB transceiver is in suspend state.
6 PDE	Enables the weak pulldowns on the USB transceiver. 0 Weak pulldowns are disabled on D+ and D–. 1 Weak pulldowns are enabled on D+ and D–.
Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

48.4.25 USB OTG Observe register (USBx_OBSERVE)

Provides visibility on the state of the pull-ups and pull-downs at the transceiver. Useful when interfacing to an external OTG control module via a serial interface.

Address: 4007_2000h base + 104h offset = 4007_2104h

Bit	7	6	5	4	3	2	1	0
Read	DPPU	DPPD	0	DMPD	0			0
Write								
Reset	0	1	0	1	0	0	0	0

USBx_OBSERVE field descriptions

Field	Description
7 DPPU	Provides observability of the D+ Pullup enable at the USB transceiver. 0 D+ pullup disabled. 1 D+ pullup enabled.
6 DPPD	Provides observability of the D+ Pulldown enable at the USB transceiver. 0 D+ pulldown disabled. 1 D+ pulldown enabled.
5 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
4 DMPD	Provides observability of the D- Pulldown enable at the USB transceiver. 0 D– pulldown disabled. 1 D– pulldown enabled.
3–1 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
0 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

48.4.26 USB OTG Control register (USBx_CONTROL)

Address: 4007_2000h base + 108h offset = 4007_2108h

Bit	7	6	5	4
Read	0			DPPULLUPNONOTG
Write				
Reset	0	0	0	0

Bit	3	2	1	0
Read	0			
Write				
Reset	0	0	0	0

USBx_CONTROL field descriptions

Field	Description
7–5 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
4 DPPULLUPNONOTG	Provides control of the DP Pullup in USBOTG, if USB is configured in non-OTG device mode. 0 DP Pullup in non-OTG device mode is not enabled. 1 DP Pullup in non-OTG device mode is enabled.
Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

48.4.27 USB Transceiver Control register 0 (USBx_USBTRC0)

Includes signals for basic operation of the on-chip USB Full Speed transceiver and configuration of the USB data connection that are not otherwise included in the USB Full Speed controller registers.

Address: 4007_2000h base + 10Ch offset = 4007_210Ch

Bit	7	6	5	4
Read		0	USBRESMEN	0
Write	USBRESET			
Reset	0	0	0	0

Bit	3	2	1	0
Read	0		SYNC_DET	USB_RESUME_INT
Write				
Reset	0	0	0	0

USBx_USBTRC0 field descriptions

Field	Description
7 USBRESET	USB Reset Generates a hard reset to USBOTG. After this bit is set and the reset occurs, this bit is automatically cleared. **NOTE: This bit is always read as zero. Wait two USB clock cycles after setting this bit before accessing other USB register bitfields.** 0 Normal USB module operation. 1 Returns the USB module to its reset state.
6 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
5 USBRESMEN	Asynchronous Resume Interrupt Enable

Table continues on the next page...

USBx_USBTRC0 field descriptions (continued)

Field	Description
	This bit, when set, allows the USB module to send an asynchronous wakeup event to the MCU upon detection of resume signaling on the USB bus. The MCU then re-enables clocks to the USB module. It is used for low-power suspend mode when USB module clocks are stopped or the USB transceiver is in Suspend mode. Async wakeup only works in device mode. 0 USB asynchronous wakeup from suspend mode disabled. 1 USB asynchronous wakeup from suspend mode enabled. The asynchronous resume interrupt differs from the synchronous resume interrupt in that it asynchronously detects K-state using the unfiltered state of the D+ and D– pins. This interrupt should only be enabled when the Transceiver is suspended.
4–2 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
1 SYNC_DET	Synchronous USB Interrupt Detect 0 Synchronous interrupt has not been detected. 1 Synchronous interrupt has been detected.
0 USB_RESUME_ INT	USB Asynchronous Interrupt 0 No interrupt was generated. 1 Interrupt was generated because of the USB asynchronous interrupt.

48.4.28 Frame Adjust Register (USBx_USBFRMADJUST)

Address: 4007_2000h base + 114h offset = 4007_2114h

Bit	7	6	5	4	3	2	1	0
Read Write	ADJ							
Reset	0	0	0	0	0	0	0	0

USBx_USBFRMADJUST field descriptions

Field	Description
ADJ	Frame Adjustment In Host mode, the frame adjustment is a twos complement number that adjusts the period of each USB frame in 12-MHz clock periods. A SOF is normally generated every 12,000 12-MHz clock cycles. The Frame Adjust Register can adjust this by -128 to +127 to compensate for inaccuracies in the USB 48-MHz clock. Changes to the ADJ bit take effect at the start of the next frame.

48.5 OTG and Host mode operation

The Host mode logic allows portable, mobile, and other devices using this SOC to function as a USB Embedded Host (EH) Controller. The OTG logic adds an interface to allow the OTG Host Negotiation and Session Request Protocols (HNP and SRP) to be implemented in software.

Host mode is intended for use in handheld-portable devices to allow easy connection to simple HID class devices such as printers and keyboards. It is not intended to perform the functions of a full OHCI or UHCI compatible host controller found on PC motherboards. Host mode allows bulk, isochronous, interrupt and control transfers. Bulk data transfers are performed at nearly the full USB interface bandwidth. Support is provided for ISO transfers, but the number of ISO streams that can be practically supported is affected by the interrupt latency of the processor servicing the Token Done interrupts from the SIE. Custom drivers must be written to support Host mode operation.

Setting the HOST_MODE_EN bit in the CTL register enables Host mode. The USB-FS core can only operate as a peripheral device or in Host mode. It cannot operate in both modes simultaneously. When HOST_MODE is enabled, only endpoint zero is used. All other endpoints should be disabled by software.

48.6 Host Mode Operation Examples

The following sections illustrate the steps required to perform USB host functions using the USB-FS core. For more information about these procedures, see the *Universal Serial Bus Specification, Revision 2.0*, "Chapter 9 USB Device Framework."

To enable host mode and discover a connected device:

1. Enable Host Mode (CTL[HOST_MODE_EN]=1). The pull-down resistors are enabled, and pull-up disabled. Start of Frame (SOF) generation begins. SOF counter loaded with 12,000. Disable SOF packet generation to eliminate noise on the USB by writing the USB enable bit to 0 (CTL[USB_EN]=0).
2. Enable the ATTACH interrupt (INTEN[ATTACHEN]=1).
3. Wait for ATTACH interrupt (ISTAT[ATTACH]). Signaled by USB peripheral device pull-up resistor changing the state of DPLUS or DMINUS from 0 to 1 (SE0 to J or K state).
4. Check the state of the JSTATE and SE0 bits in the control register. If the connecting device is low speed (JSTATE bit is 0), set the low-speed bit in the address registers (ADDR[LS_EN]=1) and the Host Without Hub bit in endpoint 0 register control (ENDPT0[HOSTWOHUB]=1).
5. Enable RESET (CTL[RESET]=1) for 10 ms.
6. Enable SOF packet to keep the connected device from going to suspend (CTL[USB_EN=1]).

7. Enumerate the attached device by sending the appropriate commands to the default control pipe of the connected device.

To complete a control transaction to a connected device:

1. Complete all the steps to discover a connected device
2. Set up the endpoint control register for bidirectional control transfers ENDPT0[4:0] = 0x0d.
3. Place a copy of the device framework setup command in a memory buffer.
4. Initialize current even or odd TX EP0 BDT to transfer the 8 bytes of command data for a device framework command (for example, a GET DEVICE DESCRIPTOR).
    - Set the BDT command word to 0x00080080 –Byte count to 8, OWN bit to 1.
    - Set the BDT buffer address field to the start address of the 8 byte command buffer.
5. Set the USB device address of the peripheral device in the address register (ADDR[6:0]). After the USB bus reset, the device USB address is zero. It is set to some other value usually 1 by the Set Address device framework command.
6. Write the TOKEN register with a SETUP to Endpoint 0, the peripheral device default control pipe (TOKEN=0xD0). This initiates a setup token on the bus followed by a data packet. The device handshake is returned in the BDT PID field after the packets complete. When the BDT is written, a Token Done (ISTAT[TOKDNE]) interrupt is asserted. This completes the setup phase of the setup transaction.
7. To initiate the data phase of the setup transaction (that is, get the data for the GET DEVICE DESCRIPTOR command), set up a buffer in memory for the data to be transferred.
8. Initialize the current even or odd TX EP0 BDT to transfer the data.
    - Set the BDT command word to 0x004000C0 – BC to 64 (the byte count of the data buffer in this case), OWN bit to 1, Data toggle to Data1.
    - Set the BDT buffer address field to the start address of the data buffer
9. Write the TOKEN register with an IN or OUT token to Endpoint 0, the peripheral device default control pipe, an IN token for a GET DEVICE DESCRIPTOR command (TOKEN=0x90). This initiates an IN token on the bus followed by a data packet from the device to the host. When the data packet completes, the BDT is written and a Token Done (ISTAT[DNE]) interrupt is asserted. For control transfers with a single packet data phase this completes the data phase of the setup transaction.

10. To initiate the status phase of the setup transaction, set up a buffer in memory to receive or send the zero length status phase data packet.
11. Initialize the current even or odd TX EP0 BDT to transfer the status data.
    - Set the BDT command word to 0x00000080 – BC to 0 (the byte count of the data buffer in this case), OWN bit to 1, Data toggle to Data1.
    - Set the BDT buffer address field to the start address of the data buffer
12. Write the TOKEN register with a IN or OUT token to Endpoint 0, the peripheral device default control pipe, an OUT token for a GET DEVICE DESCRIPTOR command (TOKEN=0x10). This initiates an OUT token on the bus followed by a zero length data packet from the host to the device. When the data packet completes, the BDT is written with the handshake from the device and a Token Done (ISTAT[TOKDNE]) interrupt is asserted. This completes the data phase of the setup transaction.

To send a full speed bulk data transfer to a peripheral device:

1. Complete all steps to discover a connected device and to configure a connected device. Write the ADDR register with the address of the peripheral device. Typically, there is only one other device on the USB bus in host mode so it is expected that the address is 0x01 and should remain constant.
2. Write 0x1D to ENDPT0 register to enable transmit and receive transfers with handshaking enabled.
3. Setup the even TX EP0 BDT to transfer up to 64 bytes.
4. Set the USB device address of the peripheral device in the address register (ADDR[6:0]).
5. Write the TOKEN register with an OUT token to the desired endpoint. The write to this register triggers the USB-FS transmit state machines to begin transmitting the token and the data.
6. Setup the odd TX EP0 BDT to transfer up to 64 bytes.
7. Write the TOKEN register with an OUT token as in step 4. Two tokens can be queued at a time to allow the packets to be double buffered to achieve maximum throughput.
8. Wait for the TOKDNE interrupt. This indicates that one of the BDTs has been released back to the processor and the transfer has completed. If the peripheral device asserts NAKs, the USB-FS continues to retry the transfer indefinitely without processor intervention unless the ENDPT0[RETRYDIS] is 1. If the retry disable field

is set, the handshake (ACK, NAK, STALL, or ERROR (0xf)) is returned in the BDT PID field. If a stall interrupt occurs, the pending packet must be dequeued and the error condition in the peripheral device cleared. If a Reset interrupt occurs (SE0 for more than 2.5 µs), the peripheral device has detached.

9. After the TOK_DNE interrupt occurs, the BDTs can be examined and the next data packet queued by returning to step 2.

48.7 On-The-Go operation

The USB-OTG core provides sensors and controls to enable On-The-Go (OTG) operation. These sensors are used by the OTG driver software to implement the Host Negotiation Protocol (HNP) and Session Request Protocol (SRP) and give access to the OTG protocol control signals. The following state machines show the OTG operations involved with HNP and SRP protocols from either end of the USB cable.

48.7.1 OTG dual role A device operation

A device is considered the A device because of the type of cable attached. If the USB Type Standard-A or Micro-A plug is plugged into the device, it is considered the A device.

A dual role A device operates as the following flow diagram and state description table illustrates.

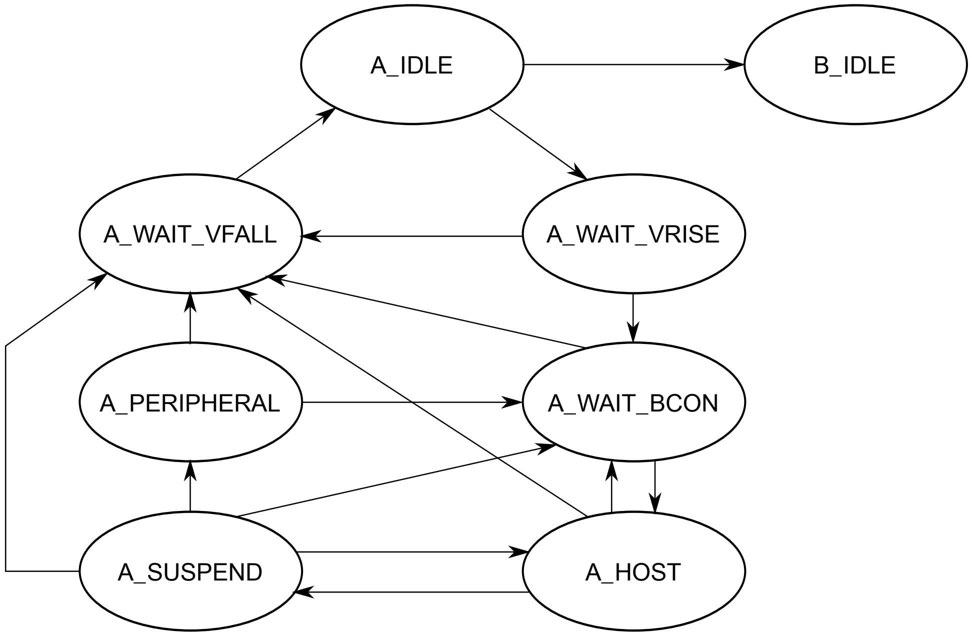


Figure 48-7. Dual role A device flow diagram

Table 48-8. State descriptions for the dual role A device flow

State	Action	Response
A_IDLE	If ID Interrupt. The cable has been unplugged or a Type B cable has been attached. The device now acts as a Type B device.	Go to B_IDLE
	If the A application wants to use the bus or if the B device is doing an SRP as indicated by an A_SESS_VLD Interrupt or Attach or Port Status Change Interrupt check data line for 5 –10 msec pulsing.	Go to A_WAIT_VRISE Turn on DRV_VBUS
A_WAIT_VRISE	If ID Interrupt or if A_VBUS_VLD is false after 100 msec The cable has been changed or the A device cannot support the current required from the B device.	Go to A_WAIT_VFALL Turn off DRV_VBUS
	If A_VBUS_VLD interrupt	Go to A_WAIT_BCON
A_WAIT_BCON	After 200 ms without Attach or ID Interrupt. (This could wait forever if desired.)	Go to A_WAIT_FALL Turn off DRV_VBUS
	A_VBUS_VLD Interrupt and B device attaches	Go to A_HOST Turn on Host mode
A_HOST	Enumerate Device determine OTG Support.	
	If A_VBUS_VLD/ Interrupt or A device is done and does not think it wants to do something soon or the B device disconnects	Go to A_WAIT_VFALL Turn off Host mode Turn off DRV_VBUS
	If the A device is finished with session or if the A device wants to allow the B device to take bus.	Go to A_SUSPEND
	ID Interrupt or the B device disconnects	Go to A_WAIT_BCON
A_SUSPEND	If ID Interrupt, or if 150 ms B disconnect timeout (This timeout value could be longer) or if A_VBUS_VLD\ Interrupt	Go to A_WAIT_VFALL Turn off DRV_VBUS

Table continues on the next page...

Table 48-8. State descriptions for the dual role A device flow (continued)

State	Action	Response
	If HNP enabled, and B disconnects in 150 ms then B device is becoming the host.	Go to A_PERIPHERAL Turn off Host mode
	If A wants to start another session	Go to A_HOST
A_PERIPHERAL	If ID Interrupt or if A_VBUS_VLD interrupt	Go to A_WAIT_VFALL Turn off DRV_VBUS.
	If 3 –200 ms of Bus Idle	Go to A_WAIT_BCON Turn on Host mode
A_WAIT_VFALL	If ID Interrupt or (A_SESS_VLD/ & b_conn/)	Go to A_IDLE

48.7.2 OTG dual role B device operation

A device is considered a B device if it is connected to the bus with a USB Type Standard-B, Mini-B, or Micro-B plug inserted into the local USB receptacle. The Type Mini-B plug and receptacle are now only allowed for dedicated peripheral devices, not dual-role/OTG devices.

A dual role B device operates as the following flow diagram and state description table illustrates.

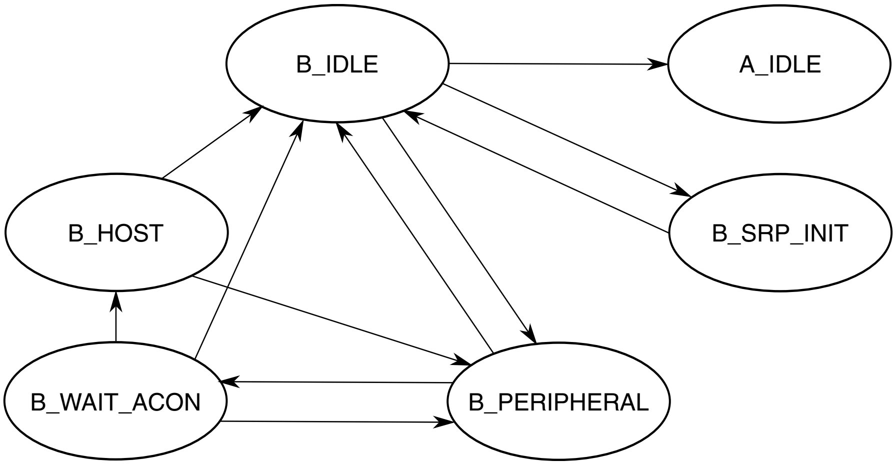


Figure 48-8. Dual role B device flow diagram

Table 48-9. State descriptions for the dual role B device flow

State	Action	Response
B_IDLE	If ID\ Interrupt. A Type A cable has been plugged in and the device should now respond as a Type A device.	Go to A_IDLE

Table continues on the next page...

Table 48-9. State descriptions for the dual role B device flow (continued)

State	Action	Response
	If B_SESS_VLD Interrupt. The A device has turned on VBUS and begins a session.	Go to B_PERIPHERAL Turn on DP_HIGH
	If B application wants the bus and Bus is Idle for 2 ms and the B_SESS_END bit is set, the B device can perform an SRP.	Go to B_SRP_INIT Pulse CHRG_VBUS Pulse DP_HIGH 5-10 ms
B_SRP_INIT	If ID\ Interrupt or SRP Done (SRP must be done in less than 100 ms.)	Go to B_IDLE
B_PERIPHERAL	If HNP enabled and the bus is suspended and B wants the bus, the B device can become the host.	Go to B_WAIT_ACON Turn off DP_HIGH
B_WAIT_ACON	If A connects, an attach interrupt is received	Go to B_HOST Turn on Host Mode
	If ID\ Interrupt or B_SESS_VLD/ Interrupt If the cable changes or if VBUS goes away, the host doesn't support us. Go to B_IDLE	Go to B_IDLE
	If 3.125 ms expires or if a Resume occurs	Go to B_PERIPHERAL
B_HOST	If ID\ Interrupt or B_SESS_VLD\ Interrupt If the cable changes or if VBUS goes away, the host doesn't support us.	Go to B_IDLE
	If B application is done or A disconnects	Go to B_PERIPHERAL

Chapter 49
MCU: USB Device Charger Detection Module (USBDCD)

49.1 Preface

49.1.1 References

The following publications are referenced in this document. For updates to these specifications, see http://www.usb.org.

- *USB Battery Charging Specification Revision 1.1, USB Implementers Forum*
- *Universal Serial Bus Specification Revision 2.0, USB Implementers Forum*

49.1.2 Acronyms and abbreviations

The following table contains acronyms and abbreviations used in this document.

Table 49-1. Acronyms and abbreviated terms

Term	Meaning
FS	Full speed (12 Mbit/s)
HS	High speed (480 Mbit/s)
I_{DEV_DCHG}	Current drawn when the USB device is connected to a dedicated charging port
$I_{DEV\_HCHG\_LFS}$	Current drawn when the USB device is connected to an FS charging host port
I_{DM_SINK}	Current sink for the D– line
I_{DP_SRC}	Current source for the D+ line
I_{SUSP}	Current drawn when the USB device is suspended
LDO	Low dropout
LS	Low Speed (1.5 Mbit/s)
N/A	Not applicable

Table continues on the next page...

Table 49-1. Acronyms and abbreviated terms (continued)

Term	Meaning
OTG	On-The-Go
R_{DM_DWN}	D– pulldown resistance for data pin contact detect
V_{DAT_REF}	Data detect reference voltage for the voltage comparator
V_{DP_SRC}	Voltage source for the D+ line
V_{LGC}	Threshold voltage for logic high

49.1.3 Glossary

The following table shows a glossary of terms used in this document.

Table 49-2. Glossary of terms

Term	Definition
Transceiver	Module that implements the physical layer of the USB standard (FS or LS only).
PHY	Module that implements the physical layer of the USB standard (HS capable).
Attached	Device is physically plugged into USB port, but has *not enabled* either D+ or D– pullup resistor.
Connected	Device is physically plugged into USB port, and has *enabled* either D+ or D– pullup resistor.
Suspended	After 3 ms of no bus activity, the USB device enters suspend mode.
Component	The hardware and software that make up a subsystem.

49.2 Introduction

NOTE

> For the chip-specific implementation details of this module's instances, see the chip configuration information.

The USBDCD module works with the USB transceiver to detect whether the USB device is attached to a charging port, either a dedicated charging port or a charging host. System software coordinates the detection activites of the module and controls an off-chip integrated circuit that performs the battery charging.

49.2.1 Block diagram

The following figure is a high level block diagram of the module.

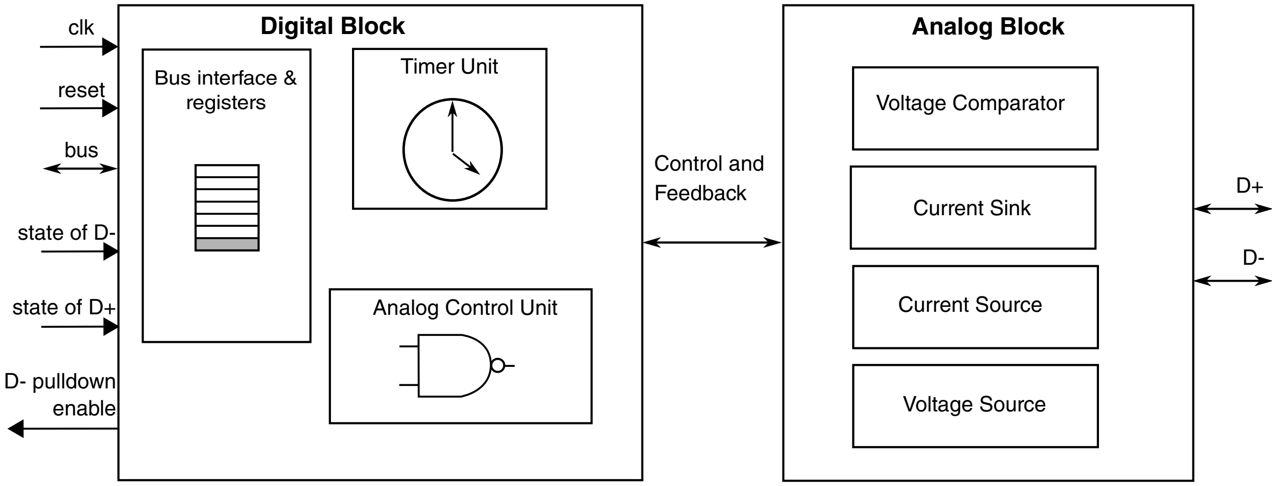


Figure 49-1. Block diagram

The USBDCD module consists of two main blocks:

- A digital block provides the programming interface (memory-mapped registers) and includes the timer unit and the analog control unit.
- An analog block provides the circuitry for the physical detection of the charger, including the voltage source, current source, current sink, and voltage comparator circuitry.

49.2.2 Features

The USBDCD module offers the following features:

- Compliant with the latest industry standard specification: *USB Battery Charging Specification, Revision 1.1*
- Programmable timing parameters default to values required by the industry standards:
  - Having standard default values allows for easy configuration- simply set the clock frequency before enabling the module.
  - Programmability allows the flexibility to meet future updates of the standards.

49.2.3 Modes of operation

The operating modes of the USBDCD module are shown in the following table.

Table 49-3. Module modes and their conditions

Module mode	Description	Conditions when used
Enabled	The module performs the charger detection sequence.	System software should enable the module only when *all* of the following conditions are true: • The system uses a rechargeable battery. • The device is being used in an FS USB device application. • The device has detected that it is attached to the USB cable.
Disabled	The module is not active and is held in a low power state.	System software should disable the module when *either* of the following conditions is true: • The charger detect sequence is complete. • The conditions for being enabled are not met.
Powered Off	The digital supply voltage dvdd is removed.	Low system performance requirements allow putting the device into a very low-power stop mode.

Operating mode transitions are shown in the following table.

Table 49-4. Entering and exiting module modes

Module mode	Entering	Exiting	Mode after exiting
Enabled	Set CONTROL[START].	Set CONTROL[SR].	Disabled
Disabled	Take *either* of the following actions: • Set CONTROL[SR].[1] • Reset the module. By default, the module is disabled.	Set CONTROL[START].	Enabled
Powered Off	Perform the following actions: 1. Put the device into very low-power stop mode. 2. Adjust the supply voltages.	Perform the following actions: 1. Restore the supply voltages. 2. Take the device out of very low-power stop mode.	Disabled

[1] The effect of setting the SR bit is immediate; that is, the module is disabled even if the sequence has not completed.

49.3 Module signal descriptions

This section describes the module signals. The following table shows a summary of module signals that interface with the pins of the device.

Table 49-5. Signal descriptions

Signal	Description	I/O
usb_dm	USB D– analog data signal. The analog block interfaces directly to the D– signal on the USB bus.	I/O
usb_dp	USB D+ analog data signal. The analog block interfaces directly to the D+ signal on the USB bus.	I/O
avdd33[2]	3.3 V regulated analog supply	I
avss	Analog ground	I
dvss	Digital ground	I
dvdd	1.2 V digital supply	I

[2] Voltage must be 3.3 V +/- 10% for full functionality of the module. That is, the charger detection function does not work when this voltage is below 3.0 V, and the CONTROL[START] bit should not be set.

NOTE

> The transceiver module also interfaces to the usb_dm and usb_dp signals. Both modules and the USB host/hub use these signals as bidirectional, tristate signals.

Information about the signal integrity aspects of the lines including shielding, isolated return paths, input or output impedance, packaging, suggested external components, ESD, and other protections can be found in the USB 2.0 specification and in Application information.

49.4 Memory map/Register definition

This section describes the memory map and registers for the USBDCD module.

USBDCD memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4003_5000	Control register (USBDCD_CONTROL)	32	R/W	0001_0000h	49.4.1/1222
4003_5004	Clock register (USBDCD_CLOCK)	32	R/W	0000_00C1h	49.4.2/1223
4003_5008	Status register (USBDCD_STATUS)	32	R	0000_0000h	49.4.3/1225
4003_5010	TIMER0 register (USBDCD_TIMER0)	32	R/W	0010_0000h	49.4.4/1226
4003_5014	TIMER1 register (USBDCD_TIMER1)	32	R/W	000A_0028h	49.4.5/1227
4003_5018	TIMER2 register (USBDCD_TIMER2)	32	R/W	0028_0001h	49.4.6/1228

49.4.1 Control register (USBDCD_CONTROL)

Contains the control and interrupt bit fields.

Address: 4003_5000h base + 0h offset = 4003_5000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0						0	0	0							IE
W							SR	START								
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved							IF	0							0
W																IACK
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

USBDCD_CONTROL field descriptions

Field	Description
31–26 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
25 SR	Software Reset Determines whether a software reset is performed. 0 Do not perform a software reset. 1 Perform a software reset.
24 START	Start Change Detection Sequence Determines whether the charger detection sequence is initiated. 0 Do not start the sequence. Writes of this value have no effect. 1 Initiate the charger detection sequence. If the sequence is already running, writes of this value have no effect.
23–17 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
16 IE	Interrupt Enable Enables/disables interrupts to the system.

Table continues on the next page...

USBDCD_CONTROL field descriptions (continued)

Field	Description
	0 Disable interrupts to the system. 1 Enable interrupts to the system.
15–9 Reserved	This field is reserved.
8 IF	Interrupt Flag Determines whether an interrupt is pending. 0 No interrupt is pending. 1 An interrupt is pending.
7–1 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
0 IACK	Interrupt Acknowledge Determines whether the interrupt is cleared. 0 Do not clear the interrupt. 1 Clear the IF bit (interrupt flag).

49.4.2 Clock register (USBDCD_CLOCK)

Address: 4003_5000h base + 4h offset = 4003_5004h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0				CLOCK_SPEED										0	CLOCK_UNIT
W																
Reset	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	1

USBDCD_CLOCK field descriptions

Field	Description
31–12 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
11–2 CLOCK_SPEED	Numerical Value of Clock Speed in Binary

Table continues on the next page...

USBDCD_CLOCK field descriptions (continued)

Field	Description
	The unit of measure is programmed in CLOCK_UNIT. The valid range is from 1 to 1023 when clock unit is MHz and 4 to 1023 when clock unit is kHz. Examples with CLOCK_UNIT = 1: • For 48 MHz: 0b00_0011_0000 (48) (Default) • For 24 MHz: 0b00_0001_1000 (24) Examples with CLOCK_UNIT = 0: • For 100 kHz: 0b00_0110_0100 (100) • For 500 kHz: 0b01_1111_0100 (500)
1 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
0 CLOCK_UNIT	Unit of Measurement Encoding for Clock Speed Specifies the unit of measure for the clock speed. 0 kHz Speed (between 1 kHz and 1023 kHz) 1 MHz Speed (between 1 MHz and 1023 MHz)

49.4.3 Status register (USBDCD_STATUS)

Provides the current state of the module for system software monitoring.

Address: 4003_5000h base + 8h offset = 4003_5008h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0									ACTIVE	TO	ERR	SEQ_STAT		SEQ_RES	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

USBDCD_STATUS field descriptions

Field	Description
31–23 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
22 ACTIVE	Active Status Indicator Indicates whether the sequence is running. 0 The sequence is not running. 1 The sequence is running.
21 TO	Timeout Flag Indicates whether the detection sequence has passed the timeout threshhold.

Table continues on the next page...

USBDCD_STATUS field descriptions (continued)

Field	Description
	0 The detection sequence has not been running for over 1 s. 1 It has been over 1 s since the data pin contact was detected and debounced.
20 ERR	Error Flag Indicates whether there is an error in the detection sequence. 0 No sequence errors. 1 Error in the detection sequence. See the SEQ_STAT field to determine the phase in which the error occurred.
19–18 SEQ_STAT	Charger Detection Sequence Status Indicates the status of the charger detection sequence. 00 The module is either not enabled, or the module is enabled but the data pins have not yet been detected. 01 Data pin contact detection is complete. 10 Charging port detection is complete. 11 Charger type detection is complete.
17–16 SEQ_RES	Charger Detection Sequence Results Reports how the charger detection is attached. 00 No results to report. 01 Attached to a standard host. Must comply with USB 2.0 by drawing only 2.5 mA (max) until connected. 10 Attached to a charging port. The exact meaning depends on bit 18: • 0: Attached to either a charging host or a dedicated charger. The charger type detection has not completed. • 1: Attached to a charging host. The charger type detection has completed. 11 Attached to a dedicated charger.
Reserved	This field is reserved. **NOTE:** Bits do not always read as 0.

49.4.4 TIMER0 register (USBDCD_TIMER0)

TIMER0 has an TSEQ_INIT field that represents the system latency in ms. Latency is measured from the time when VBUS goes active until the time system software initiates charger detection sequence in USBDCD module. When software sets the CONTROL[START] bit, the Unit Connection Timer (TUNITCON) is initialized with the value of TSEQ_INIT.

Valid values are 0–1023, however the USB Battery Charging Specification requires the entire sequence, including TSEQ_INIT, to be completed in 1s or less.

Address: 4003_5000h base + 10h offset = 4003_5010h

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

USBDCD_TIMER0 field descriptions

Field	Description
31–26 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
25–16 TSEQ_INIT	Sequence Initiation Time TSEQ_INIT represents the system latency (in ms) measured from the time VBUS goes active to the time system software initiates the charger detection sequence in the USBDCD module. When software sets the CONTROL[START] bit, the Unit Connection Timer (TUNITCON) is initialized with the value of TSEQ_INIT. Valid values are 0-1023, but the USB Battery Charging Specification requires the entire sequence, including TSEQ_INIT, to be completed in 1s or less.
15–12 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
TUNITCON	Unit Connection Timer Elapse (in ms) Displays the amount of elapsed time since the event of setting the START bit plus the value of TSEQ_INIT. The timer is automatically initialized with the value of TSEQ_INIT This timer enables compliance with the maximum time allowed to connect T_{UNIT_CON} under the USB Battery Charging Specification, v1.1.If the timer reaches the one second limit, the module triggers an interrupt and sets the error flag STATUS[ERR]. The timer continues counting throughout the charger detection sequence, even when control has been passed to software. As long as the module is active, the timer continues to count until it reaches the maximum value of 0xFFF (4095 ms). The timer does not rollover to zero. A software reset clears the timer.

49.4.5 TIMER1 register (USBDCD_TIMER1)

TIMER1 contains timing parameters. Note that register values can be written that are not compliant with the USB Battery Charging Specification v1.1, so care should be taken when overwriting the default values.

Address: 4003_5000h base + 14h offset = 4003_5014h

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

USBDCD_TIMER1 field descriptions

Field	Description
31–26 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

Table continues on the next page...

USBDCD_TIMER1 field descriptions (continued)

Field	Description
25–16 TDCD_DBNC	Time Period to Debounce D+ Signal Sets the time period (ms) to debounce the D+ signal during the data pin contact detection phase. See "Debouncing the data pin contact" Valid values are 1–1023, but the USB Battery Charging Specification requires a minimum value of 10 ms.
15–10 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
TVDPSRC_ON	Time Period Comparator Enabled This timing parameter is used after detection of the data pin. See "Charging Port Detection". Valid values are 1–1023, but the USB Battery Charging Specification requires a minimum value of 40 ms.

49.4.6 TIMER2 register (USBDCD_TIMER2)

TIMER2 contains timing parameters.

NOTE

> Register values can be written that are not compliant with the USB Battery Charging Specification v1.1, so care should be taken when overwriting the default values.

Address: 4003_5000h base + 18h offset = 4003_5018h

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

USBDCD_TIMER2 field descriptions

Field	Description
31–26 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
25–16 TVDPSRC_CON	Time Period Before Enabling D+ Pullup Sets the time period (ms) that the module waits after charging port detection before system software must enable the D+ pullup to connect to the USB host. Valid values are 1–1023, but the USB Battery Charging Specification requires a minimum value of 40 ms.
15–4 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
CHECK_DM	Time Before Check of D– Line Sets the amount of time (in ms) that the module waits after the device connects to the USB bus until checking the state of the D� line to determine the type of charging port. See "Charger Type Detection." Valid values are 1–15ms.

49.5 Functional description

The sequence of detecting the presence of charging port and type of charging port involves several hardware components, coordinated by system software. This collection of interacting hardware and software is called the USB Battery Charging Subsystem. The following figure shows the USBDCD module as a component of the subsystem. The following table describes the components.

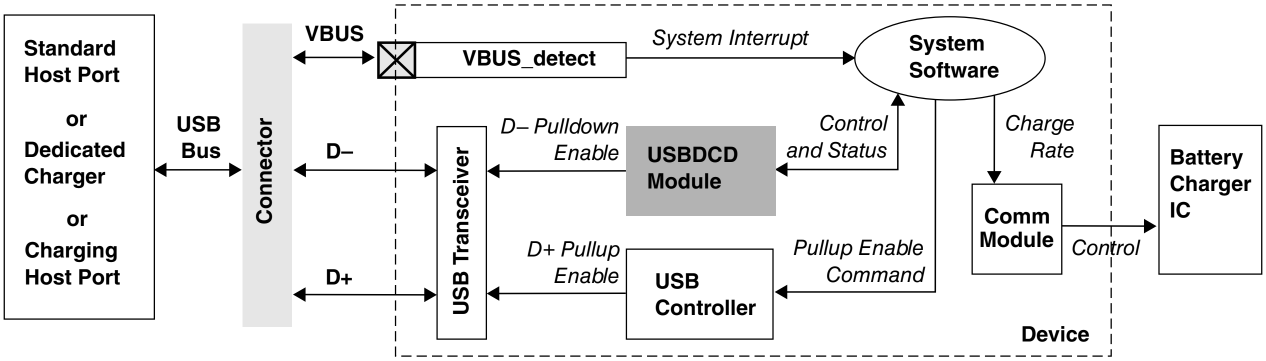


Figure 49-2. USB battery charging subsystem

Table 49-6. USB battery charger subsystem components

Component	Description
Battery Charger IC	The external battery charger IC regulates the charge rate to the rechargable battery. System software is responsible for communicating the appropriate charge rates. <table><tr><th>Charger</th><th>Maximum current drawn[1]</th></tr><tr><td>Standard host port</td><td>up to 500 mA</td></tr><tr><td>Charging host port</td><td>up to 1500 mA</td></tr><tr><td>Dedicated charging port</td><td>up to 1800 mA</td></tr></table> 1. If the USB host has suspended the USB device, system software must configure the system to limit the current drawn from the USB bus to 2.5 mA or less.
Comm Module	A communications module on the device can be used to control the charge rate of the battery charger IC.
System software	Coordinates the detection activities of the subsystem.
USB Controller	The D+ pullup enable control signal plays a role during the charger type detection phase. System software must issue a command to the USB controller to assert this signal. After this pullup is enabled, the device is considered to be connected to the USB bus. The host then attempts to enumerate it. **NOTE:** The USB controller must be used only for USB device applications when using the USBDCD module. For USB host applications, the USBDCD module must be disabled.

Table continues on the next page...

Table 49-6. USB battery charger subsystem components (continued)

Component	Description
USB Transceiver	The USB transceiver contains the pullup resistor for the USB D+ signal and the pulldown resistors for the USB D+ and D– signals. The D+ pullup and the D– pulldown are both used during the charger detection sequence. The USB transceiver also outputs the digital state of the D+ and D– signals from the USB bus. The pullup and pulldown enable signals are controlled by other modules during the charger detection sequence. The D+ pullup enable is output from the USB controller and is under software control. The USBDCD module controls the D–pulldown enable.
USBDCD Module	Detects whether the device has been plugged into either a standard host port, a charging host port, or a dedicated charger.
VBUS_detect	This interrupt pin connected to the USB VBUS signal detects when the device has been plugged into or unplugged from the USB bus. If the system requires waking up from a low power mode on being plugged into the USB port, this interrupt should also be a low power wake up source. If this pin multiplexes other functions, such as GPIO, the pin can be configured as an interrupt so that the USB plug or unplug event can be detected.

1. If the USB host has suspended the USB device, system software must configure the system to limit the current drawn from the USB bus to 2.5 mA or less.

49.5.1 The charger detection sequence

The following figure illustrates the charger detection sequence in a simplified timing diagram based on the USB Battery Charging Specification v1.1.

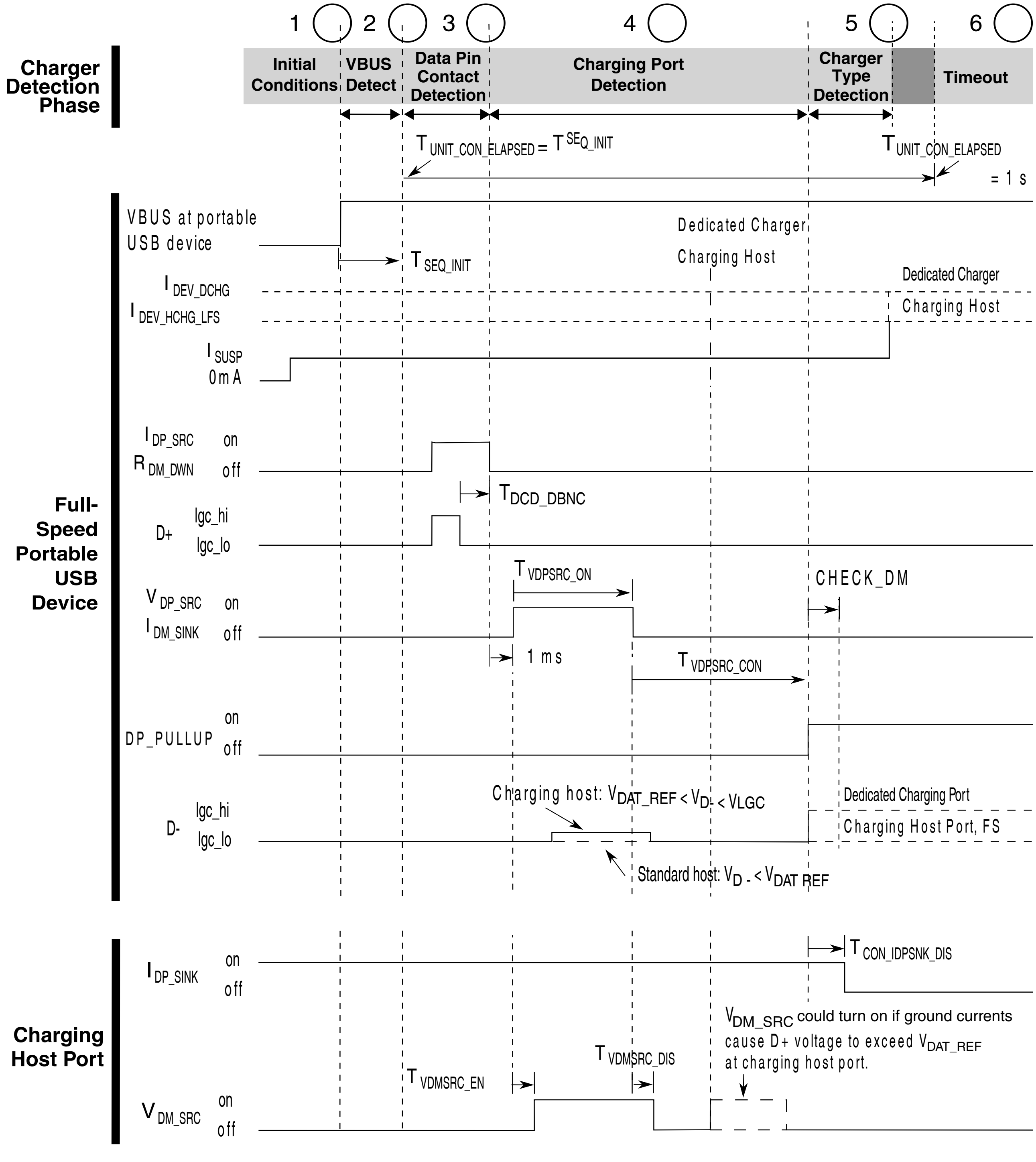


Figure 49-3. Full speed charger detection timing

The following table provides an overview description of the charger detection sequence shown in the preceding figure.

Table 49-7. Overview of the charger detection sequence

Phase		Overview description	Full description
1	Initial Conditions	Initial system conditions that need to be met before the detection sequence is initiated.	Initial System Conditions
2	VBUS Detection	System software detects contact of the VBUS signal with the system interrupt pin VBUS_detect.	VBUS contact detection
3	Data Pin Contact Detection	The USBDCD module detects that the USB data pins D+ and D– have made contact with the USB port.	Data pin contact detection
4	Charging Port Detection	The USBDCD module detects if the port is a standard host or either type of charging port, that is charging host or dedicated charger.	Charging port detection
5	Charger Type Detection	The USBDCD module detects the type of charging port, if applicable.	Charger type detection
6	Sequence Timeout	The USBDCD module did not finish the detection sequence within the timeout interval. The sequence will continue until halted by software.	Charger detection sequence timeout

Timing parameter values used in this module are listed in the following table.

Table 49-8. Timing parameters for the charger detection sequence

Parameter	USB Battery Charging Spec	Module default	Module programmable range
T_{DCD_DBNC}	10 ms min (no max)	10 ms	0– 1023 ms
T_{VDPSRC_ON}[1]	40 ms min (no max)	40 ms	0 –1023 ms
T_{VDPSRC_CON}[1]	40 ms min (no max)	40 ms	0 –1023 ms
CHECK_DM	N/A	1 ms	0– 15 ms
T_{SEQ_INIT}	N/A	16 ms	0 –1023 ms
T_{UNIT_CON}[1]	1 s	N/A	N/A
T_{VDMSRC_EN}[1]	1– 20 ms	From the USB host	N/A
T_{VDMSRC_DIS}[1]	0 –20 ms	From the USB host	N/A
$T_{CON\_IDPSINK\_DIS}$[1]	0– 20 ms	From the USB host	N/A

[1] This parameter is defined by the *USB Battery Charging Specification, v1.1*.

49.5.1.1 Initial System Conditions

The USBDCD module can be used only with FS USB device applications using a rechargable battery. That is, it cannot be used with USB applications that are HS, LS, host, or OTG. In addition, before the USBDCD module's charger detection sequence can be initiated, the system must be:

- Powered-up and in run mode.
- Recently plugged into a USB port.
- Drawing not more than 2.5 mA total system current from the USB bus.

Examples of allowable precursors to this set of initial conditions include:

- A powered-down device is subsequently powered-up upon being plugged into the USB bus.
- A device in a low power mode subsequently enters run mode upon being plugged into the USB bus.

49.5.1.2 VBUS contact detection

Once the device is plugged into a USB port, the VBUS_detect system interrupt is triggered. System software must do the following to initialize the module and start the charger detection sequence:

1. Restore power if the module is powered-off.
2. Set CONTROL[SR] to initiate a software reset.
3. Configure the USBDCD module by programming the CLOCK register and the timing parameters as needed.
4. Set CONTROL[IE] to enable interrupts, or clear the bit if software polling method is used .
5. Set CONTROL[START] to start the charger detection sequence.

49.5.1.3 Data pin contact detection

The module must ensure that the data pins have made contact because the detection sequence depends upon the state of the USB D+ signal. USB plugs and receptables are designed such that when the plug is inserted into the receptable, the power pins make contact before the data pins make contact. See the following figure.

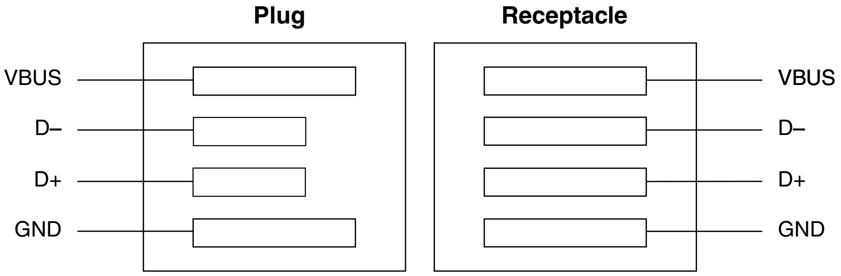


Figure 49-4. Relative pin positions in USB plugs and receptacles

As a result, when a portable USB device is attached to an upstream port, the portable USB device detects VBUS before the data pins have made contact. The time between power pins and data pins making contact depends on how fast the plug is inserted into the receptable. Delays of several hundred milliseconds are possible.

49.5.1.3.1 Debouncing the data pin contact

When system software has initiated the charger detection sequence, as described in Initial System Conditions, the USBDCD module turns on the I_{DP_SRC} current source and enables the R_{DM_DWN} pulldown resistor. If the data pins have not made contact, the D+ line remains high. After the data pins make contact, the D+ line goes low and debouncing begins.

After the D+ line goes low, the module continuously samples the D+ line over the duration of the T_{DCD_DBNC} debounce time interval.By deafult, T_{DCD_DBNC} is 10 ms, but it can be programmed in the TIMER0[TDCD_DBNC] field. See the description of the TIMER0 Register for register information.

When it has remained low for the entire interval, the debouncing is complete. However, if the D+ line returns high during the debounce interval, the module waits until the D+ line goes low again to restart the debouncing. This cycle repeats until either of the following happens:

- The data pin contact has been successfully debounced (see Success in detecting data pin contact (phase completion)).
- A timeout occurs (see Charger detection sequence timeout).

49.5.1.3.2 Success in detecting data pin contact (phase completion)

After successfully debouncing the D+ state, the module does the following:

- Updates the STATUS register to reflect phase completion (See Table 49-11 for field values.)
- Directly proceeds to the next step in the sequence: detection of a charging port (See Charging port detection.)

49.5.1.4 Charging port detection

After it detects that the data pins have made contact, the module waits for a fixed delay of 1 ms, and then attempts to detect whether it is plugged into a charging port. The module connects the following analog units to the USB D+ or D– lines during this phase:

- The voltage source V_{DP_SRC} connects to the D+ line
- The current sink I_{DM_SINK} connects to the D– line
- The voltage comparator connects to the USB D– line, comparing it to the voltage V_{DAT_REF}.

After a time of T_{VDPSRC_ON}, the module samples the D– line. The T_{VDPSRC_ON} parameter is programmable and defaults to 40 ms. After sampling the D– line, the module disconnects the voltage source, current sink, and comparator.

The next steps in the sequence depend on the voltage on the D– line as determined by the voltage comparator. See the following table.

Table 49-9. Sampling D– in the charging port detection phase

If the voltage on D- is...	Then...	See...
Below V_{DAT_REF}	The port is a *standard host* that does not support the USB Battery Charging Specification v1.1.	Standard host port
Above V_{DAT_REF} but below V_{LGC}	The port is a *charging port*.	Charging port
Above V_{LGC}	This is an error condition.	Error in charging port detection

49.5.1.4.1 Standard host port

As part of the charger detection handshake with a standard USB host, the module does the following without waiting for the T_{VDPSRC_CON} interval to elapse:

- Updates the STATUS register to reflect that a standard host has been detected with SEQ_RES = 01. See Table 49-11 for field values.
- Sets CONTROL[IF].
- Generates an interrupt if enabled in CONTROL[IE].

At this point, control has been passed to system software via the interrupt. The rest of the sequence, which detects the type of charging port, is not applicable, so software should perform the following steps:

1. Read the STATUS register.
2. Set CONTROL[IACK] to acknowledge the interrupt.
3. Set CONTROL[SR] to issue a software reset to the module.
4. Disable the module.
5. Communicate the appropriate charge rate to the external battery charger IC; see Table 49-6.

49.5.1.4.2 Charging port

As part of the charger detection handshake with any type of USB host, the module waits until the T_{VDPSRC_CON} interval has elapsed before it does the following:

- Updates the STATUS register to reflect that a charging port has been detected with SEQ_RES = 10. See Table 49-11 for field values.
- Sets CONTROL[IF].
- Generates an interrupt if enabled in CONTROL[IE].

At this point, control has passed to system software via the interrupt. Software should:

1. Read the STATUS register.
2. Set CONTROL[IACK] to acknowledge the interrupt.
3. Issue a command to the USB controller to pullup the USB D+ line.
4. Wait for the module to complete the final phase of the sequence. See Charger type detection.

49.5.1.4.3 Error in charging port detection

For this error condition, the module does the following:

- Updates the STATUS register to reflect the error with SEQ_RES = 00. See Table 49-11 for field values.
- Sets CONTROL[IF].
- Generates an interrupt if enabled in CONTROL[IE].

Note that in this case the module does not wait for the T_{VDPSRC_CON} interval to elapse.

At this point, control has been passed to system software via the interrupt. The rest of the sequence (detecting the type of charging port) is not applicable, so software should:

1. Read the STATUS register.
2. Set CONTROL[IACK] to acknowledge the interrupt.
3. Set CONTROL[SR] to issue a software reset to the module.
4. Disable the module.

49.5.1.5 Charger type detection

After software enables the D+ pullup resistor, the module is notified automatically (via internal signaling) to start the CHECK_DM timer counting down the time interval programmed in the TIMER2[CHECK_DM] field.

After the CHECK_DM time has elapsed, the module samples the USB D– line to determine the type of charger. See the following table.

Table 49-10. Sampling D– in the charger type detection phase

If the voltage on D– is...	Then...	See...
High	The port is a *dedicated charging port*.[1]	Dedicated charging port
Low	The port is a *charging host port*.[2]	Charging host port

[1] In a dedicated charger, the D+ and D– lines are shorted together through a small resistor.
[2] In a charging host port, the D+ and D– lines are not shorted.

49.5.1.5.1 Dedicated charging port

For a dedicated charger, the module does the following:

- Updates the STATUS register to reflect that a dedicated charger has been detected with SEQ_RES = 11. See Table 49-11 for field values.
- Sets CONTROL[IF].
- Generates an interrupt if enabled in CONTROL[IE] bit.

At this point, control has been passed to system software via the interrupt. Software should:

1. Read the STATUS register.

2. Disable the USB controller to prevent transitions on the USB D+ or D– lines from causing spurious interrupt or wakeup events to the system.
3. Set CONTROL[IACK] to acknowledge the interrupt.
4. Set CONTROL[SR] to issue a software reset to the module.
5. Disable the module.
6. Communicate the appropriate charge rate to the external battery charger IC; see Table 49-6.

49.5.1.5.2 Charging host port

For a charging host port, the module does the following:

- Updates the STATUS register to reflect that a charging host port has been detected with SEQ_RES = 10. See Table 49-11 for field values.
- Sets CONTROL[IF].
- Generates an interrupt if enabled in CONTROL[IE].

At this point, control has been passed to system software via the interrupt. Software should:

1. Read the STATUS register.
2. Set CONTROL[IACK] to acknowledge the interrupt.
3. Set CONTROL[SR] to issue a software reset to the module.
4. Disable the module.
5. Communicate the appropriate charge rate to the external battery charger IC; see Table 49-6.

49.5.1.6 Charger detection sequence timeout

The maximum time allowed to connect according to the *USB Battery Charging Specification, v1.1* is one second. If the Unit Connection Timer reaches the one second limit and the sequence is still running as indicated by the STATUS[ACTIVE] bit, the module does the following:

- Updates the STATUS register to reflect that a timeout error has occured. See Table 49-11 for field values.
- Sets the CONTROL[IF] bit.
- Generates an interrupt if enabled in CONTROL[IE].
- The detection sequence continues until explicitly halted by software setting the CONTROL[SR] bit.
- The Unit Connection Timer continues counting. See the description of the TIMER0 Register.

At this point, control has been passed to system software via the interrupt, which has two options: ignore the interrupt and allow more time for the sequence to complete, or halt the sequence. To halt the sequence, software should:

1. Read the STATUS register.
2. Set the CONTROL[IACK] bit to acknowledge the interrupt.
3. Set the CONTROL[SR] bit to issue a software reset to the module.
4. Disable the module.

This timeout function is also useful in case software does not realize that the USB device is unplugged from USB port during the charger detection sequence. If the interrupt occurs but the V_{BUS_DETECT} input is low, software can disable and reset the module.

System software might allow the sequence to run past the timeout interrupt under these conditions:

1. The USB Battery Charging Spec is amended to allow more time. In this case, software should poll TIMER0[$T_{UNITCON}$] periodically to track elapsed time after 1s; or
2. For debug purposes.

Note that the $T_{UNITCON}$ register field will stop incrementing when it reaches its maximum value so it will not rollover to zero and start counting up again.

49.5.2 Interrupts and events

The USBDCD module has an interrupt to alert system software of certain events, which are listed in the following table. All events except the Phase Complete event for the Data Pin Detection phase can trigger an interrupt.

Table 49-11. Events triggering an interrupt by sequence phase

Sequence phase	Event	Event description	STATUS fields[1]	Phase description

Data Pin Detection	Phase Complete	The module has detected data pin contact.
		No interrupt occurs: CONTROL[IF] = 0.
			ERR = 0
			SEQ_STAT = 01
			SEQ_RES = 00
			TO = 0
				VBUS contact detection

Charging Port Detection	Phase Complete	The module has completed the process of identifying if the USB port is a charging port or not.	ERR = 0
			SEQ_STAT = 10
			SEQ_RES = 01 or 10
			TO = 0
				Charging port detection

	Error	The module cannot identify the type of port because the D– line is above the USB's VLGC threshold.	ERR = 1
			SEQ_STAT = 10
			SEQ_RES = 00
			TO = 0
				Error in charging port detection

Charger Type Detection	Phase Complete	The module has completed the process of identifying the charger type detection.
		Note: The ERR flag always reads zero because no known error conditions are checked during this phase.
			ERR = 0
			SEQ_STAT = 11
			SEQ_RES = 11 or 10
			TO = 0
				Charger type detection

Sequence Timeout	Error	The timeout interval from the time the USB device attaches to a USB port until it connects has elapsed.	ERR = 1
			SEQ_STAT = last value
			SEQ_RES = last value[2]
			TO = 1
				Charger detection sequence timeout.

[1] See the description of the Status register for register information.
[2] The SEQ_STAT and SEQ_RES fields retain the values held at the time of the timeout error.

49.5.2.1 Interrupt Handling

Software can read which event caused the interrupt from the STATUS register during the interrupt service routine.

An interrupt is generated only if CONTROL[IE] is set. The CONTROL[IF] bit is always set under interrupt conditions, even if CONTROL[IE] is cleared. In this case, software can poll CONTROL[IF] to determine if an interrupt condition is pending.

Writes to CONTROL[IF] are ignored. To reset CONTROL[IF], set CONTROL[IACK] to acknowledge the interrupt. Writing to CONTROL[IACK] when CONTROL[IF] is cleared has no effect.

49.5.3 Resets

There are two ways to reset various register contents in this module: hardware resets and a software reset.

49.5.3.1 Hardware resets

Hardware resets originate at the system or device level and propagate down to the individual module level. They include start up reset, low-voltage reset, and all other hardware reset sources.

Hardware resets cause the register contents to be restored to their default state as listed in the register descriptions.

49.5.3.2 Software reset

A software reset re-initializes the module's status information, but leaves configuration information unchanged. The software reset allows software to prepare the module without needing to reprogram the same configuration each time the USB device is plugged into a USB port.

Setting CONTROL[SR] initiates a software reset. The following table shows all register fields that are reset to their default values by a software reset.

Table 49-12. Software reset and register fields affected

Register	Fields affected	Fields not affected
CONTROL[1]	IF	IE, START
STATUS	All	None
CLOCK	None	All
TIMER*n*	TUNITCON	All other

[1] CONTROL[SR] and CONTROL[IACK] are self-clearing.

A software reset also returns all internal logic, timers, and counters to their reset states. If the module is already active (STATUS[ACTIVE] = 1), a software reset stops the sequence.

Note

> Software must always initiate a software reset before starting the sequence to ensure the module is in a known state.

49.6 Initialization information

This module has been designed for minimal configuration while retaining significant programmability. The CLOCK register needs to be initialized to the actual system clock frequency, unless the default value already matches the system requirements.

The other registers generally do not need to be modified, because they default to values that comply with the USB Battery Charging Specification v1.1. However, several timing parameters can be changed for a great deal of flexibility if a particular system requires it.

All module configuration must occur *before* initiating the charger detection sequence. Configuration changes made *after* setting CONTROL[START] result in undefined behavior.

49.7 Application information

This section provides application information.

49.7.1 External pullups

Any external pullups applied to the USB D+ or D- data lines must be capable of being disabled to prevent incorrect pullup values or incorrect operation of the USB subsystem.

49.7.2 Dead or weak battery

According to the USB Battery Charging Specification v1.1, a USB device with a dead, weak, or missing battery that is attached to a charging port can remain attached indefinitely drawing up to 1.5A, until the battery is charged to the point that the USB device can connect.

The USBDCD module is compatible with systems that do not check the strength of the battery. Therefore, this module assumes that the battery is good, so the USB device must immediately connect to the USB bus by pulling the D+ line high after the USBDCD module has determined that the device is attached to a charging port.

The module is also compatible with systems that do check the strength of the battery. In these systems, if it is known that the battery is weak or dead, software can delay connecting to the USB while charging at 1.5A. Once the battery is charged to the good battery threshold, software can then connect to the USB host by pulling the D+ line high.

49.7.3 Handling unplug events

If the device is unplugged from the USB bus during the charger detection sequence, the contents of the STATUS register must be ignored and the USBDCD module must get a Software Reset, as described in Software reset.

Chapter 50
MCU: USB Voltage Regulator

50.1 Introduction

NOTE

> For the chip-specific implementation details of this module's instances, see the chip configuration information.

The USB Voltage Regulator module is a LDO linear voltage regulator to provide 3.3V power from an input power supply varying from 2.7 V to 5.5 V. It consists of one 3.3 V power channel. When the input power supply is below 3.6 V, the regulator goes to pass-through mode. The following figure shows the ideal relation between the regulator output and input power supply.

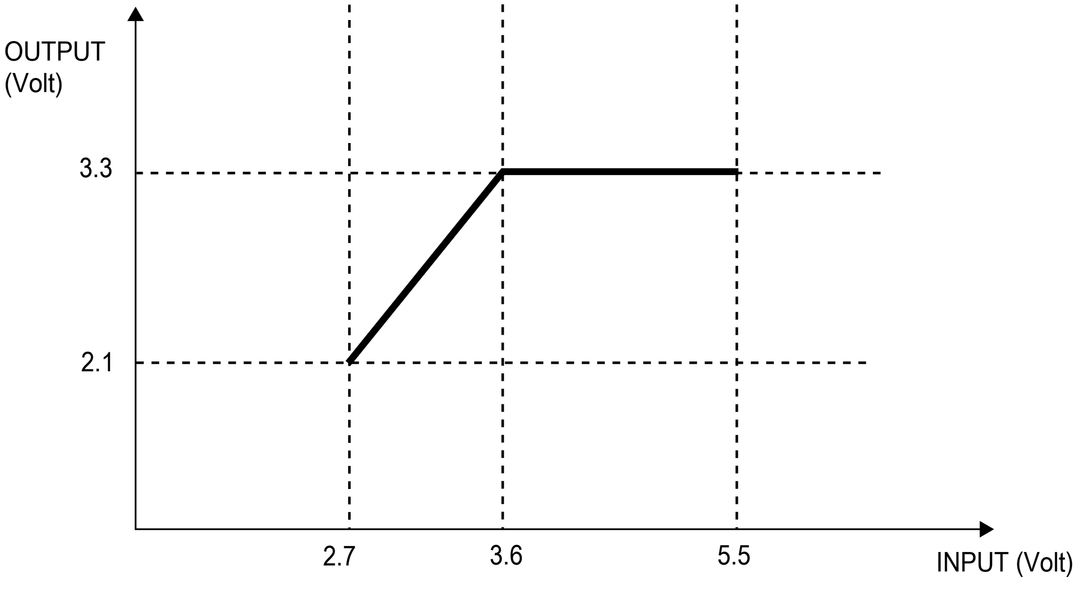


Figure 50-1. Ideal Relation Between the Regulator Output and Input Power Supply

50.1.1 Overview

A simplified block diagram for the USB Voltage Regulator module is shown below.

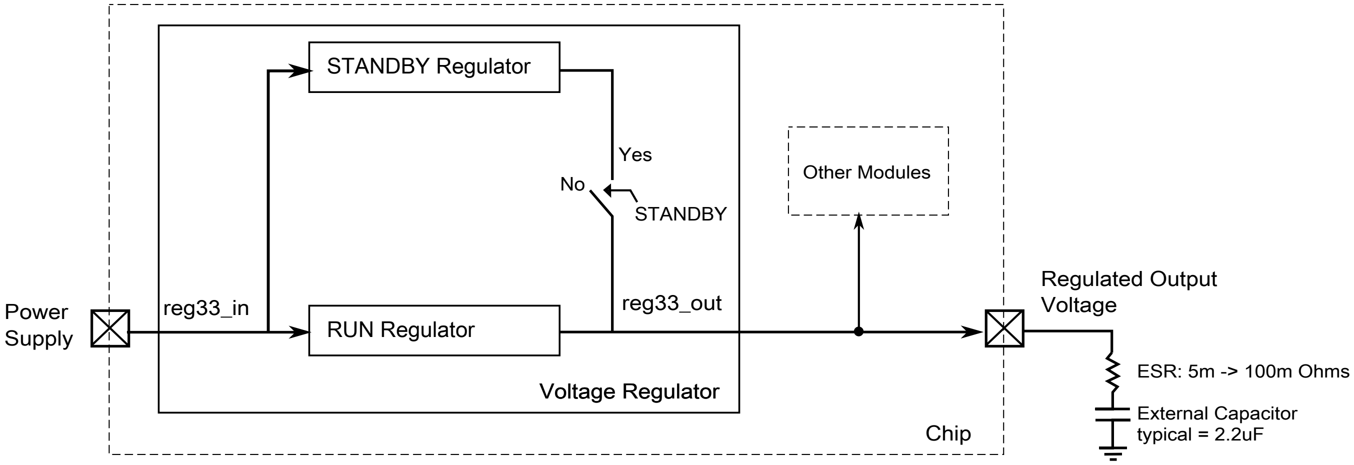


Figure 50-2. USB Voltage Regulator Block Diagram

This module uses 2 regulators in parallel. In run mode, the RUN regulator with the bandgap voltage reference is enabled and can provide up to 120 mA load current. In run mode, the STANDBY regulator and the low power reference are also enabled, but a switch disconnects its output from the external pin. In STANDBY mode, the RUN regulator is disabled and the STANDBY regulator output is connected to the external pin.

Internal power mode signals control whether the module is in RUN or STANDBY mode.

50.1.2 Features

- Low drop-out linear voltage regulator with one power channel (3.3 V).
- Low drop-out voltage: 300 mV.
- Output current: 120 mA.
- Three different power modes: RUN, STANDBY and SHUTDOWN.
- Low quiescent current in RUN mode.
  - Typical value is around 120 μA (one thousand times smaller than the maximum load current).
- Very low quiescent current in STANDBY mode.
  - Typical value is around 1 μA.
- Automatic current limiting if the load current is greater than 290 mA.

- Automatic power-up once some voltage is applied to the regulator input.
- Pass-through mode for regulator input voltages less than 3.6 V
- Small output capacitor: 2.2 µF
- Stable with aluminum, tantalum or ceramic capacitors.

50.1.3 Modes of Operation

The regulator has these power modes:

- RUN—The regulating loop of the RUN regulator and the STANDBY regulator are active, but the switch connecting the STANDBY regulator output to the external pin is open.
- STANDBY—The regulating loop of the RUN regulator is disabled and the standby regulator is active. The switch connecting the STANDBY regulator output to the external pin is closed.
- SHUTDOWN—The module is disabled.

The regulator is enabled by default. This means that once the power supply is provided, the module power-up sequence to RUN mode starts.

50.2 USB Voltage Regulator Module Signal Descriptions

The following table shows the external signals for the regulator.

Table 50-1. USB Voltage Regulator Module Signal Descriptions

Signal	Description	I/O
reg33_in	Unregulated power supply	I
reg33_out	Regulator output voltage	O

Chapter 51
MCU: Serial Peripheral Interface (SPI)

51.1 Introduction

NOTE

> For the chip-specific implementation details of this module's instances, see the chip configuration information.

The serial peripheral interface (SPI) module provides a synchronous serial bus for communication between a chip and an external peripheral device.

51.1.1 Block Diagram

The block diagram of this module is as follows:

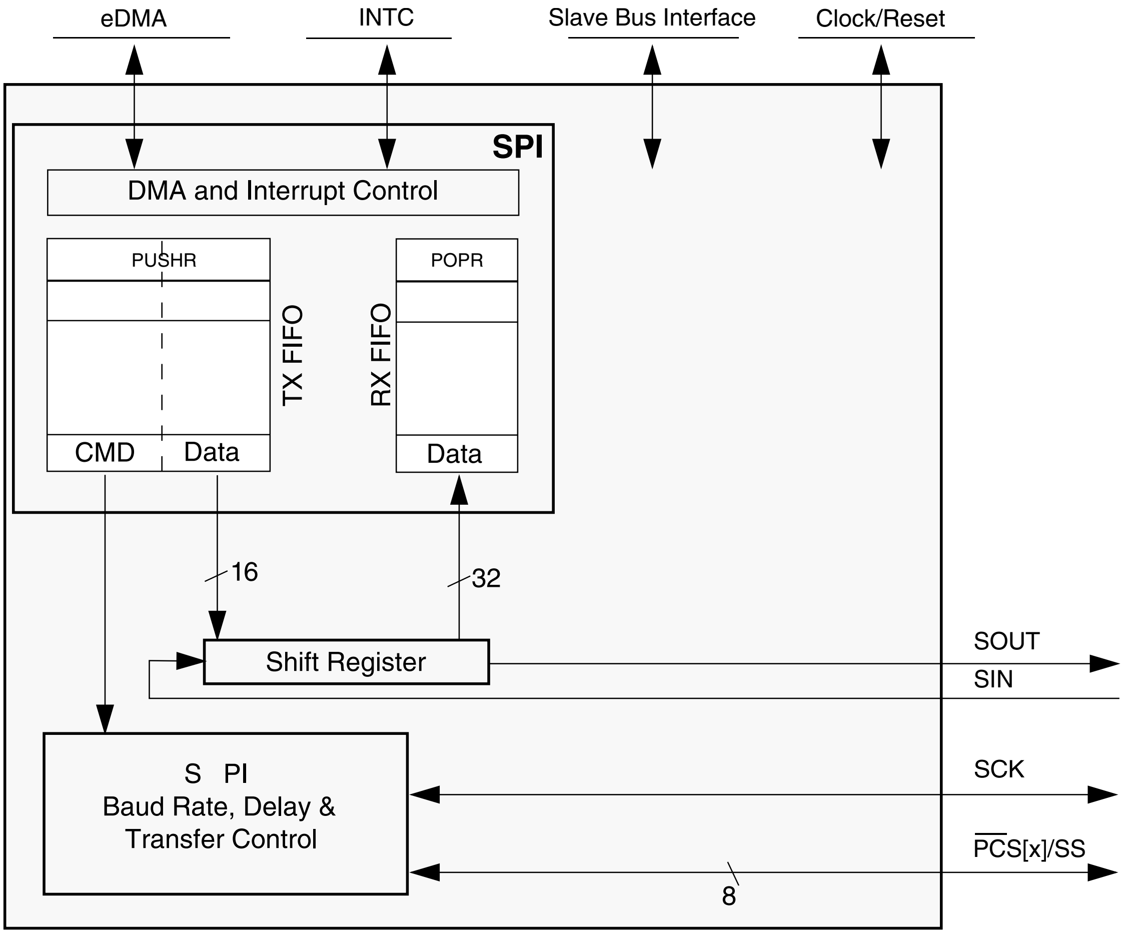


Figure 51-1. SPI Block Diagram

51.1.2 Features

The module supports the following features:

- Full-duplex, three-wire synchronous transfers
- Master mode
- Slave mode
- Data streaming operation in Slave mode with continuous slave selection
- Buffered transmit operation using the transmit first in first out (TX FIFO) with depth of 4 entries
- Buffered receive operation using the receive FIFO (RX FIFO) with depth of 4 entries
- Asynchronous clocking scheme for Register and Protocol Interfaces

- TX and RX FIFOs can be disabled individually for low-latency updates to SPI queues
- Visibility into TX and RX FIFOs for ease of debugging
- Programmable transfer attributes on a per-frame basis:
    - two transfer attribute registers
    - Serial clock (SCK) with programmable polarity and phase
    - Various programmable delays
    - Programmable serial frame size: 4 to 16 bits
        - SPI frames longer than 16 bits can be supported using the continuous selection format.
    - Continuously held chip select capability
- 5 peripheral chip selects (PCSes), expandable to 32 with external demultiplexer
- Deglitching support for up to 16 peripheral chip selects (PCSes) with external demultiplexer
- DMA support for adding entries to TX FIFO and removing entries from RX FIFO:
    - TX FIFO is not full (TFFF)
    - RX FIFO is not empty (RFDF)
- Interrupt conditions:
    - End of Queue reached (EOQF)
    - TX FIFO is not full (TFFF)
    - Transfer of current frame complete (TCF)
    - Attempt to transmit with an empty Transmit FIFO (TFUF)
    - RX FIFO is not empty (RFDF)
    - Frame received while Receive FIFO is full (RFOF)
- Global interrupt request line
- Modified SPI transfer formats for communication with slower peripheral devices
- Power-saving architectural features:

- Support for Stop mode
- Support for Doze mode

51.1.3 Interface configurations

51.1.3.1 SPI configuration

The Serial Peripheral Interface (SPI) configuration allows the module to send and receive serial data. This configuration allows the module to operate as a basic SPI block with internal FIFOs supporting external queue operation. Transmitted data and received data reside in separate FIFOs. The host CPU or a DMA controller read the received data from the Receive FIFO and write transmit data to the Transmit FIFO.

For queued operations, the SPI queues can reside in system RAM, external to the module. Data transfers between the queues and the module FIFOs are accomplished by a DMA controller or host CPU. The following figure shows a system example with DMA, SPI, and external queues in system RAM.

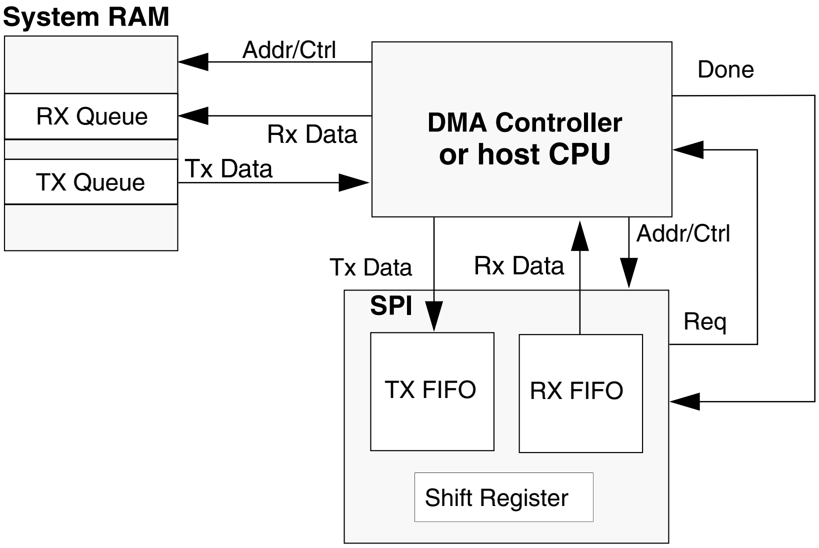


Figure 51-2. SPI with queues and DMA

51.1.4 Modes of Operation

The module supports the following modes of operation that can be divided into two categories:

- Module-specific modes:

  - Master mode
  - Slave mode
  - Module Disable mode
- Chip-specific modes:
  - External Stop mode
  - Debug mode

The module enters module-specific modes when the host writes a module register. The chip-specific modes are controlled by signals external to the module. The chip-specific modes are modes that a chip may enter in parallel to the block-specific modes.

51.1.4.1 Master Mode

Master mode allows the module to initiate and control serial communication. In this mode, these signals are controlled by the module and configured as outputs:

- SCK
- SOUT
- PCS[*x*]

51.1.4.2 Slave Mode

Slave mode allows the module to communicate with SPI bus masters. In this mode, the module responds to externally controlled serial transfers. The SCK signal and the PCS[0]/$\overline{\text{SS}}$ signals are configured as inputs and driven by an SPI bus master.

51.1.4.3 Module Disable Mode

The Module Disable mode can be used for chip power management. The clock to the non-memory mapped logic in the module can be stopped while in the Module Disable mode.

51.1.4.4 External Stop Mode

External Stop mode is used for chip power management. The module supports the Peripheral Bus Stop mode mechanism. When a request is made to enter External Stop mode, it acknowledges the request and completes the transfer that is in progress. When the module reaches the frame boundary, it signals that the protocol clock to the module may be shut off.

51.1.4.5 Debug Mode

Debug mode is used for system development and debugging. The MCR[FRZ] bit controls module behavior in the Debug mode:

- If the bit is set, the module stops all serial transfers, when the chip is in debug mode.
- If the bit is cleared, the chip debug mode has no effect on the module.

51.2 Module signal descriptions

This table describes the signals on the boundary of the module that may connect off chip (in alphabetical order).

Table 51-1. Module signal descriptions

Signal	Master mode	Slave mode	I/O
PCS0/$\overline{\text{SS}}$	Peripheral Chip Select 0 (O)	Slave Select (I)	I/O
PCS[1:3]	Peripheral Chip Selects 1–3	(Unused)	O
PCS4	Peripheral Chip Select 4	(Unused)	O
SCK	Serial Clock (O)	Serial Clock (I)	I/O
SIN	Serial Data In	Serial Data In	I
SOUT	Serial Data Out	Serial Data Out	O

51.2.1 PCS0/$\overline{\text{SS}}$—Peripheral Chip Select/Slave Select

Master mode: Peripheral Chip Select 0 (O)—Selects an SPI slave to receive data transmitted from the module.

Slave mode: Slave Select (I)—Selects the module to receive data transmitted from an SPI master.

51.2.2 PCS1–PCS3—Peripheral Chip Selects 1–3

Master mode: Peripheral Chip Selects 1–3 (O)—Select an SPI slave to receive data transmitted by the module.

Slave mode: Unused

51.2.3 PCS4—Peripheral Chip Select 4

Master mode: Peripheral Chip Select 4 (O)—Selects an SPI slave to receive data transmitted by the module.

Slave mode: Unused

51.2.4 SCK—Serial Clock

Master mode: Serial Clock (O)—Supplies a clock signal from the module to SPI slaves.

Slave mode: Serial Clock (I)—Supplies a clock signal to the module from an SPI master.

51.2.5 SIN—Serial Input

Master mode: Serial Input (I)—Receives serial data.

Slave mode: Serial Input (I)—Receives serial data.

51.2.6 SOUT—Serial Output

Master mode: Serial Output (O)—Transmits serial data.

Slave mode: Serial Output (O)—Transmits serial data.

NOTE

Serial Data Out output buffers are controlled through SIU (or SIUL) and cannot be controlled through the module.

51.3 Memory Map/Register Definition

Register accesses to memory addresses that are reserved or undefined result in a transfer error. Write access to the POPR and RXFRn also results in a transfer error.

SPI memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4002_C000	Module Configuration Register (SPI0_MCR)	32	R/W	0000_4001h	51.3.1/1258
4002_C008	Transfer Count Register (SPI0_TCR)	32	R/W	0000_0000h	51.3.2/1261
4002_C00C	Clock and Transfer Attributes Register (In Master Mode) (SPI0_CTAR0)	32	R/W	7800_0000h	51.3.3/1262
4002_C00C	Clock and Transfer Attributes Register (In Slave Mode) (SPI0_CTAR0_SLAVE)	32	R/W	7800_0000h	51.3.4/1266
4002_C010	Clock and Transfer Attributes Register (In Master Mode) (SPI0_CTAR1)	32	R/W	7800_0000h	51.3.3/1262
4002_C02C	Status Register (SPI0_SR)	32	R/W	0200_0000h	51.3.5/1268
4002_C030	DMA/Interrupt Request Select and Enable Register (SPI0_RSER)	32	R/W	0000_0000h	51.3.6/1271
4002_C034	PUSH TX FIFO Register In Master Mode (SPI0_PUSHR)	32	R/W	0000_0000h	51.3.7/1273
4002_C034	PUSH TX FIFO Register In Slave Mode (SPI0_PUSHR_SLAVE)	32	R/W	0000_0000h	51.3.8/1275
4002_C038	POP RX FIFO Register (SPI0_POPR)	32	R	0000_0000h	51.3.9/1275
4002_C03C	Transmit FIFO Registers (SPI0_TXFR0)	32	R	0000_0000h	51.3.10/ 1276
4002_C040	Transmit FIFO Registers (SPI0_TXFR1)	32	R	0000_0000h	51.3.10/ 1276
4002_C044	Transmit FIFO Registers (SPI0_TXFR2)	32	R	0000_0000h	51.3.10/ 1276
4002_C048	Transmit FIFO Registers (SPI0_TXFR3)	32	R	0000_0000h	51.3.10/ 1276
4002_C07C	Receive FIFO Registers (SPI0_RXFR0)	32	R	0000_0000h	51.3.11/ 1276
4002_C080	Receive FIFO Registers (SPI0_RXFR1)	32	R	0000_0000h	51.3.11/ 1276
4002_C084	Receive FIFO Registers (SPI0_RXFR2)	32	R	0000_0000h	51.3.11/ 1276
4002_C088	Receive FIFO Registers (SPI0_RXFR3)	32	R	0000_0000h	51.3.11/ 1276
4002_D000	Module Configuration Register (SPI1_MCR)	32	R/W	0000_4001h	51.3.1/1258
4002_D008	Transfer Count Register (SPI1_TCR)	32	R/W	0000_0000h	51.3.2/1261
4002_D00C	Clock and Transfer Attributes Register (In Master Mode) (SPI1_CTAR0)	32	R/W	7800_0000h	51.3.3/1262
4002_D00C	Clock and Transfer Attributes Register (In Slave Mode) (SPI1_CTAR0_SLAVE)	32	R/W	7800_0000h	51.3.4/1266
4002_D010	Clock and Transfer Attributes Register (In Master Mode) (SPI1_CTAR1)	32	R/W	7800_0000h	51.3.3/1262
4002_D02C	Status Register (SPI1_SR)	32	R/W	0200_0000h	51.3.5/1268
4002_D030	DMA/Interrupt Request Select and Enable Register (SPI1_RSER)	32	R/W	0000_0000h	51.3.6/1271
4002_D034	PUSH TX FIFO Register In Master Mode (SPI1_PUSHR)	32	R/W	0000_0000h	51.3.7/1273

Table continues on the next page...

SPI memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4002_D034	PUSH TX FIFO Register In Slave Mode (SPI1_PUSHR_SLAVE)	32	R/W	0000_0000h	51.3.8/1275
4002_D038	POP RX FIFO Register (SPI1_POPR)	32	R	0000_0000h	51.3.9/1275
4002_D03C	Transmit FIFO Registers (SPI1_TXFR0)	32	R	0000_0000h	51.3.10/ 1276
4002_D040	Transmit FIFO Registers (SPI1_TXFR1)	32	R	0000_0000h	51.3.10/ 1276
4002_D044	Transmit FIFO Registers (SPI1_TXFR2)	32	R	0000_0000h	51.3.10/ 1276
4002_D048	Transmit FIFO Registers (SPI1_TXFR3)	32	R	0000_0000h	51.3.10/ 1276
4002_D07C	Receive FIFO Registers (SPI1_RXFR0)	32	R	0000_0000h	51.3.11/ 1276
4002_D080	Receive FIFO Registers (SPI1_RXFR1)	32	R	0000_0000h	51.3.11/ 1276
4002_D084	Receive FIFO Registers (SPI1_RXFR2)	32	R	0000_0000h	51.3.11/ 1276
4002_D088	Receive FIFO Registers (SPI1_RXFR3)	32	R	0000_0000h	51.3.11/ 1276

51.3.1 Module Configuration Register (SPIx_MCR)

Contains bits to configure various attributes associated with the module operations. The HALT and MDIS bits can be changed at any time, but the effect takes place only on the next frame boundary. Only the HALT and MDIS bits in the MCR can be changed, while the module is in the Running state.

Address: Base address + 0h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	MSTR	CONT_SCKE	DCONF		FRZ	MTFE	Reserved	ROOE	Reserved			PCSIS				
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	DOZE	MDIS	DIS_TXF	DIS_RXF	0	0	SMPL_PT		0					Reserved	Reserved	HALT
W					CLR_TXF	CLR_RXF										
Reset	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1

SPIx_MCR field descriptions

Field	Description
31 MSTR	Master/Slave Mode Select Enables either Master mode (if supported) or Slave mode (if supported) operation. 0 Enables Slave mode 1 Enables Master mode
30 CONT_SCKE	Continuous SCK Enable Enables the Serial Communication Clock (SCK) to run continuously. 0 Continuous SCK disabled. 1 Continuous SCK enabled.
29–28 DCONF	SPI Configuration. Selects among the different configurations of the module. 00 SPI 01 Reserved 10 Reserved 11 Reserved
27 FRZ	Freeze Enables transfers to be stopped on the next frame boundary when the device enters Debug mode. 0 Do not halt serial transfers in Debug mode. 1 Halt serial transfers in Debug mode.
26 MTFE	Modified Transfer Format Enable Enables a modified transfer format to be used. 0 Modified SPI transfer format disabled. 1 Modified SPI transfer format enabled.
25 Reserved	This field is reserved.
24 ROOE	Receive FIFO Overflow Overwrite Enable In the RX FIFO overflow condition, configures the module to ignore the incoming serial data or overwrite existing data. If the RX FIFO is full and new data is received, the data from the transfer, generating the overflow, is ignored or shifted into the shift register. 0 Incoming data is ignored. 1 Incoming data is shifted into the shift register.
23–21 Reserved	Reserved. This field is reserved.
20–16 PCSIS	Peripheral Chip Select x Inactive State Determines the inactive state of PCSx. Refer to the chip-specific SPI information for the number of PCS signals used in this chip. **NOTE:** The effect of this bit only takes place when module is enabled. Ensure that this bit is configured correctly before enabling the DSPI interface.

Table continues on the next page...

SPIx_MCR field descriptions (continued)

Field	Description
	0 The inactive state of PCSx is low. 1 The inactive state of PCSx is high.
15 DOZE	Doze Enable Provides support for an externally controlled Doze mode power-saving mechanism. 0 Doze mode has no effect on the module. 1 Doze mode disables the module.
14 MDIS	Module Disable Allows the clock to be stopped to the non-memory mapped logic in the module effectively putting it in a software-controlled power-saving state. The reset value of the MDIS bit is parameterized, with a default reset value of 1. When the module is used in Slave Mode, it is recommended to leave this bit 0, because a slave doesn't have control over master transactions. 0 Enables the module clocks. 1 Allows external logic to disable the module clocks.
13 DIS_TXF	Disable Transmit FIFO When the TX FIFO is disabled, the transmit part of the module operates as a simplified double-buffered SPI. This bit can be written only when the MDIS bit is cleared. 0 TX FIFO is enabled. 1 TX FIFO is disabled.
12 DIS_RXF	Disable Receive FIFO When the RX FIFO is disabled, the receive part of the module operates as a simplified double-buffered SPI. This bit can only be written when the MDIS bit is cleared. 0 RX FIFO is enabled. 1 RX FIFO is disabled.
11 CLR_TXF	Clear TX FIFO Flushes the TX FIFO. Writing a 1 to CLR_TXF clears the TX FIFO Counter. The CLR_TXF bit is always read as zero. 0 Do not clear the TX FIFO counter. 1 Clear the TX FIFO counter.
10 CLR_RXF	CLR_RXF Flushes the RX FIFO. Writing a 1 to CLR_RXF clears the RX Counter. The CLR_RXF bit is always read as zero. 0 Do not clear the RX FIFO counter. 1 Clear the RX FIFO counter.
9–8 SMPL_PT	Sample Point Controls when the module master samples SIN in Modified Transfer Format. This field is valid only when CPHA bit in CTARn[CPHA] is 0. 00 0 protocol clock cycles between SCK edge and SIN sample 01 1 protocol clock cycle between SCK edge and SIN sample

Table continues on the next page...

SPIx_MCR field descriptions (continued)

Field	Description
	10 2 protocol clock cycles between SCK edge and SIN sample 11 Reserved
7–3 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
2 Reserved	This field is reserved.
1 Reserved	This field is reserved.
0 HALT	Halt The HALT bit starts and stops frame transfers. See Start and Stop of Module transfers 0 Start transfers. 1 Stop transfers.

51.3.2 Transfer Count Register (SPIx_TCR)

TCR contains a counter that indicates the number of SPI transfers made. The transfer counter is intended to assist in queue management. Do not write the TCR when the module is in the Running state.

Address: Base address + 8h offset

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

SPIx_TCR field descriptions

Field	Description
31–16 SPI_TCNT	SPI Transfer Counter Counts the number of SPI transfers the module makes. The SPI_TCNT field increments every time the last bit of an SPI frame is transmitted. A value written to SPI_TCNT presets the counter to that value. SPI_TCNT is reset to zero at the beginning of the frame when the CTCNT field is set in the executing SPI command. The Transfer Counter wraps around; incrementing the counter past 65535 resets the counter to zero.
Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

51.3.3 Clock and Transfer Attributes Register (In Master Mode) (SPIx_CTAR*n*)

CTAR registers are used to define different transfer attributes. Do not write to the CTAR registers while the module is in the Running state.

In Master mode, the CTAR registers define combinations of transfer attributes such as frame size, clock phase and polarity, data bit ordering, baud rate, and various delays. In slave mode, a subset of the bitfields in CTAR0 are used to set the slave transfer attributes.

When the module is configured as an SPI master, the CTAS field in the command portion of the TX FIFO entry selects which of the CTAR registers is used. When the module is configured as an SPI bus slave, it uses the CTAR0 register.

Address: Base address + Ch offset + (4d × i), where i=0d to 1d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	DBR	FMSZ				CPOL	CPHA	LSBFE	PCSSCK		PASC		PDT		PBR	
Reset	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CSSCK				ASC				DT				BR			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SPIx_CTAR*n* field descriptions

<table>
<tr><th>Field</th><th>Description</th></tr>
<tr><td>31
DBR</td><td>Double Baud Rate
Doubles the effective baud rate of the Serial Communications Clock (SCK). This field is used only in master mode. It effectively halves the Baud Rate division ratio, supporting faster frequencies, and odd division ratios for the Serial Communications Clock (SCK). When the DBR bit is set, the duty cycle of the Serial Communications Clock (SCK) depends on the value in the Baud Rate Prescaler and the Clock Phase bit as listed in the following table. See the BR field description for details on how to compute the baud rate.
<b>Table 51-2. SPI SCK Duty Cycle</b>
<table>
<tr><th>DBR</th><th>CPHA</th><th>PBR</th><th>SCK Duty Cycle</th></tr>
<tr><td>0</td><td>any</td><td>any</td><td>50/50</td></tr>
<tr><td>1</td><td>0</td><td>00</td><td>50/50</td></tr>
<tr><td>1</td><td>0</td><td>01</td><td>33/66</td></tr>
<tr><td>1</td><td>0</td><td>10</td><td>40/60</td></tr>
</table>
</td></tr>
</table>

Table continues on the next page...

SPIx_CTAR*n* field descriptions (continued)

<table>
<tr><th>Field</th><th>Description</th></tr>
<tr><td></td><td>
<b>Table 51-2. SPI SCK Duty Cycle (continued)</b>
<table>
<tr><th>DBR</th><th>CPHA</th><th>PBR</th><th>SCK Duty Cycle</th></tr>
<tr><td>1</td><td>0</td><td>11</td><td>43/57</td></tr>
<tr><td>1</td><td>1</td><td>00</td><td>50/50</td></tr>
<tr><td>1</td><td>1</td><td>01</td><td>66/33</td></tr>
<tr><td>1</td><td>1</td><td>10</td><td>60/40</td></tr>
<tr><td>1</td><td>1</td><td>11</td><td>57/43</td></tr>
</table>

0 The baud rate is computed normally with a 50/50 duty cycle.
1 The baud rate is doubled with the duty cycle depending on the Baud Rate Prescaler.
</td></tr>
<tr><td>30–27
FMSZ</td><td>Frame Size
The number of bits transferred per frame is equal to the FMSZ value plus 1. Regardless of the transmission mode, the minimum valid frame size value is 4.</td></tr>
<tr><td>26
CPOL</td><td>Clock Polarity
Selects the inactive state of the Serial Communications Clock (SCK). This bit is used in both master and slave mode. For successful communication between serial devices, the devices must have identical clock polarities. When the Continuous Selection Format is selected, switching between clock polarities without stopping the module can cause errors in the transfer due to the peripheral device interpreting the switch of clock polarity as a valid clock edge.
<b>NOTE:</b> In case of Continuous SCK mode, when the module goes in low power mode(disabled), inactive state of SCK is not guaranted.
0 The inactive state value of SCK is low.
1 The inactive state value of SCK is high.</td></tr>
<tr><td>25
CPHA</td><td>Clock Phase
Selects which edge of SCK causes data to change and which edge causes data to be captured. This bit is used in both master and slave mode. For successful communication between serial devices, the devices must have identical clock phase settings. In Continuous SCK mode, the bit value is ignored and the transfers are done as if the CPHA bit is set to 1.
0 Data is captured on the leading edge of SCK and changed on the following edge.
1 Data is changed on the leading edge of SCK and captured on the following edge.</td></tr>
<tr><td>24
LSBFE</td><td>LSB First
Specifies whether the LSB or MSB of the frame is transferred first.
0 Data is transferred MSB first.
1 Data is transferred LSB first.</td></tr>
<tr><td>23–22
PCSSCK</td><td>PCS to SCK Delay Prescaler
Selects the prescaler value for the delay between assertion of PCS and the first edge of the SCK. See the CSSCK field description for information on how to compute the PCS to SCK Delay. Refer PCS to SCK Delay (t_{CSC}) for more details.</td></tr>
</table>

Table continues on the next page...

SPIx_CTARn field descriptions (continued)

Field	Description
	00 PCS to SCK Prescaler value is 1. 01 PCS to SCK Prescaler value is 3. 10 PCS to SCK Prescaler value is 5. 11 PCS to SCK Prescaler value is 7.
21–20 PASC	After SCK Delay Prescaler Selects the prescaler value for the delay between the last edge of SCK and the negation of PCS. See the ASC field description for information on how to compute the After SCK Delay. Refer After SCK Delay (t_{ASC}) for more details. 00 Delay after Transfer Prescaler value is 1. 01 Delay after Transfer Prescaler value is 3. 10 Delay after Transfer Prescaler value is 5. 11 Delay after Transfer Prescaler value is 7.
19–18 PDT	Delay after Transfer Prescaler Selects the prescaler value for the delay between the negation of the PCS signal at the end of a frame and the assertion of PCS at the beginning of the next frame. The PDT field is only used in master mode. See the DT field description for details on how to compute the Delay after Transfer. Refer Delay after Transfer (t_{DT}) for more details. 00 Delay after Transfer Prescaler value is 1. 01 Delay after Transfer Prescaler value is 3. 10 Delay after Transfer Prescaler value is 5. 11 Delay after Transfer Prescaler value is 7.
17–16 PBR	Baud Rate Prescaler Selects the prescaler value for the baud rate. This field is used only in master mode. The baud rate is the frequency of the SCK. The protocol clock is divided by the prescaler value before the baud rate selection takes place. See the BR field description for details on how to compute the baud rate. 00 Baud Rate Prescaler value is 2. 01 Baud Rate Prescaler value is 3. 10 Baud Rate Prescaler value is 5. 11 Baud Rate Prescaler value is 7.
15–12 CSSCK	PCS to SCK Delay Scaler Selects the scaler value for the PCS to SCK delay. This field is used only in master mode. The PCS to SCK Delay is the delay between the assertion of PCS and the first edge of the SCK. The delay is a multiple of the protocol clock period, and it is computed according to the following equation: t_{CSC} = (1/f_P) x PCSSCK x CSSCK. The following table lists the delay scaler values.

Table 51-3. Delay Scaler Encoding

Field Value	Delay Scaler Value
0000	2
0001	4
0010	8

Table continues on the next page...

SPIx_CTAR*n* field descriptions (continued)

<table>
<tr><th>Field</th><th>Description</th></tr>
<tr><td></td><td>
<b>Table 51-3. Delay Scaler Encoding (continued)</b>
<table>
<tr><th>Field Value</th><th>Delay Scaler Value</th></tr>
<tr><td>0011</td><td>16</td></tr>
<tr><td>0100</td><td>32</td></tr>
<tr><td>0101</td><td>64</td></tr>
<tr><td>0110</td><td>128</td></tr>
<tr><td>0111</td><td>256</td></tr>
<tr><td>1000</td><td>512</td></tr>
<tr><td>1001</td><td>1024</td></tr>
<tr><td>1010</td><td>2048</td></tr>
<tr><td>1011</td><td>4096</td></tr>
<tr><td>1100</td><td>8192</td></tr>
<tr><td>1101</td><td>16384</td></tr>
<tr><td>1110</td><td>32768</td></tr>
<tr><td>1111</td><td>65536</td></tr>
</table>
Refer PCS to SCK Delay (t_{CSC}) for more details.
</td></tr>
<tr><td>11–8
ASC</td><td>After SCK Delay Scaler
Selects the scaler value for the After SCK Delay. This field is used only in master mode. The After SCK Delay is the delay between the last edge of SCK and the negation of PCS. The delay is a multiple of the protocol clock period, and it is computed according to the following equation:
$t_{ASC} = (1/f_P) \text{ x PASC x ASC}$
See Delay Scaler Encoding table in CTARn[CSSCK] bit field description for scaler values. Refer After SCK Delay (t_{ASC}) for more details.</td></tr>
<tr><td>7–4
DT</td><td>Delay After Transfer Scaler
Selects the Delay after Transfer Scaler. This field is used only in master mode. The Delay after Transfer is the time between the negation of the PCS signal at the end of a frame and the assertion of PCS at the beginning of the next frame.
In the Continuous Serial Communications Clock operation, the DT value is fixed to one SCK clock period, The Delay after Transfer is a multiple of the protocol clock period, and it is computed according to the following equation:
$t_{DT} = (1/f_P) \text{ x PDT x DT}$
See Delay Scaler Encoding table in CTARn[CSSCK] bit field description for scaler values.</td></tr>
<tr><td>BR</td><td>Baud Rate Scaler
Selects the scaler value for the baud rate. This field is used only in master mode. The prescaled protocol clock is divided by the Baud Rate Scaler to generate the frequency of the SCK. The baud rate is computed according to the following equation:
SCK baud rate = $(f_P /\text{PBR}) \text{ x } [(1+\text{DBR})/\text{BR}]$
The following table lists the baud rate scaler values.</td></tr>
</table>

Table continues on the next page...

SPIx_CTAR*n* field descriptions (continued)

Field	Description

Table 51-4. Baud Rate Scaler

CTARn[BR]	Baud Rate Scaler Value
0000	2
0001	4
0010	6
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

51.3.4 Clock and Transfer Attributes Register (In Slave Mode) (SPIx_CTAR*n*_SLAVE)

When the module is configured as an SPI bus slave, the CTAR0 register is used.

Address: Base address + Ch offset + (0d × i), where i=0d to 0d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved	FMSZ				CPOL	CPHA	0		Reserved	Reserved					
W																
Reset	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SPIx_CTAR*n*_SLAVE field descriptions

Field	Description
31 Reserved	Always write the reset value to this field. This field is reserved.
30–27 FMSZ	Frame Size The number of bits transfered per frame is equal to the FMSZ field value plus 1. Note that the minimum valid value of frame size is 4.
26 CPOL	Clock Polarity Selects the inactive state of the Serial Communications Clock (SCK). **NOTE:** In case of Continuous SCK mode, when the module goes in low power mode(disabled), inactive state of SCK is not guaranted. 0 The inactive state value of SCK is low. 1 The inactive state value of SCK is high.
25 CPHA	Clock Phase Selects which edge of SCK causes data to change and which edge causes data to be captured. This bit is used in both master and slave mode. For successful communication between serial devices, the devices must have identical clock phase settings. In Continuous SCK mode, the bit value is ignored and the transfers are done as if the CPHA bit is set to 1. 0 Data is captured on the leading edge of SCK and changed on the following edge. 1 Data is changed on the leading edge of SCK and captured on the following edge.
24–23 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
22 Reserved	This field is reserved.
Reserved	This field is reserved.

51.3.5 Status Register (SPIx_SR)

SR contains status and flag bits. The bits reflect the status of the module and indicate the occurrence of events that can generate interrupt or DMA requests. Software can clear flag bits in the SR by writing a 1 to them. Writing a 0 to a flag bit has no effect. This register may not be writable in Module Disable mode due to the use of power saving mechanisms.

Address: Base address + 2Ch offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	TCF	TXRXS	0	EOQF	TFUF	0	TFFF	0	0	0	0	0	RFOF	0	RFDF	0
W	w1c			w1c	w1c		w1c						w1c		w1c	
Reset	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TXCTR				TXNXTPTR				RXCTR				POPNXTPTR			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SPIx_SR field descriptions

Field	Description
31 TCF	Transfer Complete Flag Indicates that all bits in a frame have been shifted out. TCF remains set until it is cleared by writing a 1 to it. 0 Transfer not complete. 1 Transfer complete.
30 TXRXS	TX and RX Status Reflects the run status of the module.

Table continues on the next page...

SPIx_SR field descriptions (continued)

Field	Description
	0 Transmit and receive operations are disabled (The module is in Stopped state). 1 Transmit and receive operations are enabled (The module is in Running state).
29 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
28 EOQF	End of Queue Flag Indicates that the last entry in a queue has been transmitted when the module is in Master mode. The EOQF bit is set when the TX FIFO entry has the EOQ bit set in the command halfword and the end of the transfer is reached. The EOQF bit remains set until cleared by writing a 1 to it. When the EOQF bit is set, the TXRXS bit is automatically cleared. 0 EOQ is not set in the executing command. 1 EOQ is set in the executing SPI command.
27 TFUF	Transmit FIFO Underflow Flag Indicates an underflow condition in the TX FIFO. The transmit underflow condition is detected only for SPI blocks operating in Slave mode and SPI configuration. TFUF is set when the TX FIFO of the module operating in SPI Slave mode is empty and an external SPI master initiates a transfer. The TFUF bit remains set until cleared by writing 1 to it. 0 No TX FIFO underflow. 1 TX FIFO underflow has occurred.
26 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
25 TFFF	Transmit FIFO Fill Flag Provides a method for the module to request more entries to be added to the TX FIFO. The TFFF bit is set while the TX FIFO is not full. The TFFF bit can be cleared by writing 1 to it or by acknowledgement from the DMA controller to the TX FIFO full request. **NOTE:** The reset value of this bit is 0 when the module is disabled,(MCR[MDIS]=1). 0 TX FIFO is full. 1 TX FIFO is not full.
24 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
23 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
22 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
21 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
20 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
19 RFOF	Receive FIFO Overflow Flag Indicates an overflow condition in the RX FIFO. The field is set when the RX FIFO and shift register are full and a transfer is initiated. The bit remains set until it is cleared by writing a 1 to it. 0 No Rx FIFO overflow. 1 Rx FIFO overflow has occurred.

Table continues on the next page...

SPIx_SR field descriptions (continued)

Field	Description
18 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
17 RFDF	Receive FIFO Drain Flag Provides a method for the module to request that entries be removed from the RX FIFO. The bit is set while the RX FIFO is not empty. The RFDF bit can be cleared by writing 1 to it or by acknowledgement from the DMA controller when the RX FIFO is empty. 0 RX FIFO is empty. 1 RX FIFO is not empty.
16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
15–12 TXCTR	TX FIFO Counter Indicates the number of valid entries in the TX FIFO. The TXCTR is incremented every time the PUSHR is written. The TXCTR is decremented every time an SPI command is executed and the SPI data is transferred to the shift register.
11–8 TXNXTPTR	Transmit Next Pointer Indicates which TX FIFO entry is transmitted during the next transfer. The TXNXTPTR field is updated every time SPI data is transferred from the TX FIFO to the shift register.
7–4 RXCTR	RX FIFO Counter Indicates the number of entries in the RX FIFO. The RXCTR is decremented every time the POPR is read. The RXCTR is incremented every time data is transferred from the shift register to the RX FIFO.
POPNXTPTR	Pop Next Pointer Contains a pointer to the RX FIFO entry to be returned when the POPR is read. The POPNXTPTR is updated when the POPR is read.

51.3.6 DMA/Interrupt Request Select and Enable Register (SPIx_RSER)

RSER controls DMA and interrupt requests. Do not write to the RSER while the module is in the Running state.

Address: Base address + 30h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	TCF_RE	Reserved	Reserved	EOQF_RE	TFUF_RE	Reserved	TFFF_RE	TFFF_DIRS	Reserved	Reserved	Reserved	Reserved	RFOF_RE	Reserved	RFDF_RE	RFDF_DIRS
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved	Reserved	0													
W			▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SPIx_RSER field descriptions

Field	Description
31 TCF_RE	Transmission Complete Request Enable Enables TCF flag in the SR to generate an interrupt request. 0 TCF interrupt requests are disabled. 1 TCF interrupt requests are enabled.
30 Reserved	Always write the reset value to this field. This field is reserved.
29 Reserved	Always write the reset value to this field. This field is reserved.
28 EOQF_RE	Finished Request Enable Enables the EOQF flag in the SR to generate an interrupt request. 0 EOQF interrupt requests are disabled. 1 EOQF interrupt requests are enabled.
27 TFUF_RE	Transmit FIFO Underflow Request Enable Enables the TFUF flag in the SR to generate an interrupt request.

Table continues on the next page...

SPIx_RSER field descriptions (continued)

Field	Description
	0 TFUF interrupt requests are disabled. 1 TFUF interrupt requests are enabled.
26 Reserved	Always write the reset value to this field. This field is reserved.
25 TFFF_RE	Transmit FIFO Fill Request Enable Enables the TFFF flag in the SR to generate a request. The TFFF_DIRS bit selects between generating an interrupt request or a DMA request. 0 TFFF interrupts or DMA requests are disabled. 1 TFFF interrupts or DMA requests are enabled.
24 TFFF_DIRS	Transmit FIFO Fill DMA or Interrupt Request Select Selects between generating a DMA request or an interrupt request. When SR[TFFF] and RSER[TFFF_RE] are set, this field selects between generating an interrupt request or a DMA request. 0 TFFF flag generates interrupt requests. 1 TFFF flag generates DMA requests.
23 Reserved	Always write the reset value to this field. This field is reserved.
22 Reserved	Always write the reset value to this field. This field is reserved.
21 Reserved	Always write the reset value to this field. This field is reserved.
20 Reserved	Always write the reset value to this field. This field is reserved.
19 RFOF_RE	Receive FIFO Overflow Request Enable Enables the RFOF flag in the SR to generate an interrupt request. 0 RFOF interrupt requests are disabled. 1 RFOF interrupt requests are enabled.
18 Reserved	Always write the reset value to this field. This field is reserved.
17 RFDF_RE	Receive FIFO Drain Request Enable Enables the RFDF flag in the SR to generate a request. The RFDF_DIRS bit selects between generating an interrupt request or a DMA request. 0 RFDF interrupt or DMA requests are disabled. 1 RFDF interrupt or DMA requests are enabled.
16 RFDF_DIRS	Receive FIFO Drain DMA or Interrupt Request Select

Table continues on the next page...

SPIx_RSER field descriptions (continued)

Field	Description
	Selects between generating a DMA request or an interrupt request. When the RFDF flag bit in the SR is set, and the RFDF_RE bit in the RSER is set, the RFDF_DIRS bit selects between generating an interrupt request or a DMA request. 0 Interrupt request. 1 DMA request.
15 Reserved	Always write the reset value to this field. This field is reserved.
14 Reserved	Always write the reset value to this field. This field is reserved.
Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

51.3.7 PUSH TX FIFO Register In Master Mode (SPIx_PUSHR)

Specifies data to be transferred to the TX FIFO. An 8- or 16-bit write access transfers all 32 bits to the TX FIFO. In Master mode, the register transfers 16 bits of data and 16 bits of command information. A read access of PUSHR returns the topmost TX FIFO entry.

When the module is disabled, writing to this register does not update the FIFO. Therefore, any reads performed while the module is disabled return the last PUSHR write performed while the module was still enabled.

Address: Base address + 34h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	CONT	CTAS			EOQ	CTCNT	Reserved		Reserved			PCS				
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	TXDATA															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SPIx_PUSHR field descriptions

Field	Description
31 CONT	Continuous Peripheral Chip Select Enable Selects a continuous selection format. The bit is used in SPI Master mode. The bit enables the selected PCS signals to remain asserted between transfers. 0 Return PCSn signals to their inactive state between transfers. 1 Keep PCSn signals asserted between transfers.
30–28 CTAS	Clock and Transfer Attributes Select Selects which CTAR to use in master mode to specify the transfer attributes for the associated SPI frame. In SPI Slave mode, CTAR0 is used. See the chip specific section for details to determine how many CTARs this device has. You should not program a value in this field for a register that is not present. 000 CTAR0 001 CTAR1 010 Reserved 011 Reserved 100 Reserved 101 Reserved 110 Reserved 111 Reserved
27 EOQ	End Of Queue Host software uses this bit to signal to the module that the current SPI transfer is the last in a queue. At the end of the transfer, the EOQF bit in the SR is set. 0 The SPI data is not the last data to transfer. 1 The SPI data is the last data to transfer.
26 CTCNT	Clear Transfer Counter Clears the TCNT field in the TCR register. The TCNT field is cleared before the module starts transmitting the current SPI frame. 0 Do not clear the TCR[TCNT] field. 1 Clear the TCR[TCNT] field.
25–24 Reserved	Always write the reset value to this field. This field is reserved.
23–21 Reserved	Always write the reset value to this field. This field is reserved.
20–16 PCS	Select which PCS signals are to be asserted for the transfer. Refer to the chip-specific SPI information for the number of PCS signals used in this chip. 0 Negate the PCS[x] signal. 1 Assert the PCS[x] signal.
TXDATA	Transmit Data Holds SPI data to be transferred according to the associated SPI command.

51.3.8 PUSH TX FIFO Register In Slave Mode (SPIx_PUSHR_SLAVE)

Specifies data to be transferred to the TX FIFO in slave mode. An 8- or 16-bit write access to PUSHR transfers the 16-bit TXDATA field to the TX FIFO.

Address: Base address + 34h offset

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

SPIx_PUSHR_SLAVE field descriptions

Field	Description
31–16 Reserved	This field is reserved.
TXDATA	Transmit Data Holds SPI data to be transferred according to the associated SPI command.

51.3.9 POP RX FIFO Register (SPIx_POPR)

POPR is used to read the RX FIFO. Eight- or sixteen-bit read accesses to the POPR have the same effect on the RX FIFO as 32-bit read accesses. A write to this register will generate a Transfer Error.

Address: Base address + 38h offset

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

SPIx_POPR field descriptions

Field	Description
RXDATA	Received Data Contains the SPI data from the RX FIFO entry to which the Pop Next Data Pointer points.

51.3.10 Transmit FIFO Registers (SPIx_TXFR*n*)

TXFRn registers provide visibility into the TX FIFO for debugging purposes. Each register is an entry in the TX FIFO. The registers are read-only and cannot be modified. Reading the TXFRx registers does not alter the state of the TX FIFO.

Address: Base address + 3Ch offset + (4d × i), where i=0d to 3d

<table>
<tr><td>Bit</td><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>R</td><td colspan="16">TXCMD_TXDATA</td><td colspan="16">TXDATA</td></tr>
<tr><td>W</td><td colspan="16"></td><td colspan="16"></td></tr>
<tr><td>Reset</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
</table>

SPIx_TXFR*n* field descriptions

Field	Description
31–16 TXCMD_ TXDATA	Transmit Command or Transmit Data In Master mode the TXCMD field contains the command that sets the transfer attributes for the SPI data. In Slave mode, this field is reserved.
TXDATA	Transmit Data Contains the SPI data to be shifted out.

51.3.11 Receive FIFO Registers (SPIx_RXFR*n*)

RXFRn provide visibility into the RX FIFO for debugging purposes. Each register is an entry in the RX FIFO. The RXFR registers are read-only. Reading the RXFRx registers does not alter the state of the RX FIFO.

Address: Base address + 7Ch offset + (4d × i), where i=0d to 3d

<table>
<tr><td>Bit</td><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>R</td><td colspan="32">RXDATA</td></tr>
<tr><td>W</td><td colspan="32"></td></tr>
<tr><td>Reset</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
</table>

SPIx_RXFR*n* field descriptions

Field	Description
RXDATA	Receive Data Contains the received SPI data.

SPIx_RXFR*n* field descriptions (continued)

Field	Description

51.4 Functional description

The module supports full-duplex, synchronous serial communications between chips and peripheral devices. The SPI configuration transfers data serially using a shift register and a selection of programmable transfer attributes.

The module has the following configuration

- The SPI Configuration in which the module operates as a basic SPI or a queued SPI.

The DCONF field in the Module Configuration Register (MCR) determines the module Configuration. SPI configuration is selected when DCONF within SPIx_MCR is 0b00.

The CTARn registers hold clock and transfer attributes. The SPI configuration allows to select which CTAR to use on a frame by frame basis by setting a field in the SPI command.

See Clock and Transfer Attributes Register (In Master Mode) (SPI_CTAR*n*) for information on the fields of CTAR registers.

Typical master to slave connections are shown in the following figure. When a data transfer operation is performed, data is serially shifted a predetermined number of bit positions. Because the modules are linked, data is exchanged between the master and the slave. The data that was in the master shift register is now in the shift register of the slave, and vice versa. At the end of a transfer, the Transfer Control Flag(TCF) bit in the Shift Register(SR) is set to indicate a completed frame transfer.

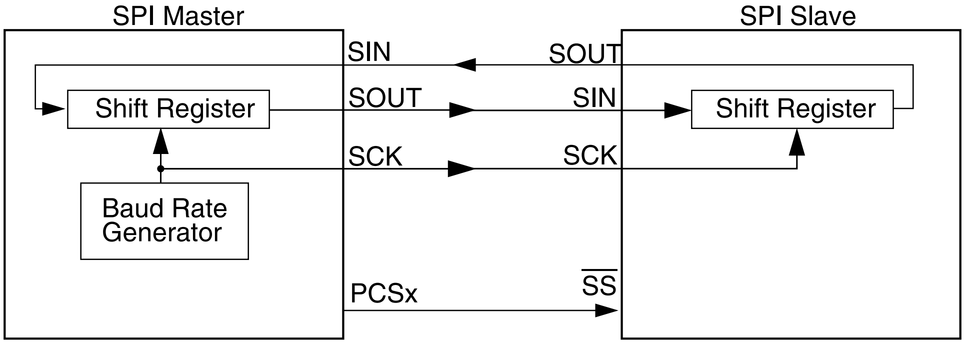


Figure 51-3. Serial protocol overview

Generally, more than one slave device can be connected to the module master. 5 Peripheral Chip Select (PCS) signals of the module masters can be used to select which of the slaves to communicate with. Refer to the chip specific section for details on the number of PCS signals used in this chip.

The SPI configuration shares transfer protocol and timing properties which are described independently of the configuration in Transfer formats. The transfer rate and delay settings are described in Module baud rate and clock delay generation.

51.4.1 Start and Stop of module transfers

The module has two operating states: Stopped and Running. Both the states are independent of it's configuration. The default state of the module is Stopped. In the Stopped state, no serial transfers are initiated in Master mode and no transfers are responded to in Slave mode. The Stopped state is also a safe state for writing the various configuration registers of the module without causing undetermined results. In the Running state serial transfers take place.

The TXRXS bit in the SR indicates the state of module. The bit is set if the module is in Running state.

The module starts or transitions to Running when all of the following conditions are true:

- SR[EOQF] bit is clear
- Chip is not in the Debug mode or the MCR[FRZ] bit is clear
- MCR[HALT] bit is clear

The module stops or transitions from Running to Stopped after the current frame when any one of the following conditions exist:

- SR[EOQF] bit is set
- Chip in the Debug mode and the MCR[FRZ] bit is set
- MCR[HALT] bit is set

State transitions from Running to Stopped occur on the next frame boundary if a transfer is in progress, or immediately if no transfers are in progress.

51.4.2 Serial Peripheral Interface (SPI) configuration

The SPI configuration transfers data serially using a shift register and a selection of programmable transfer attributes. The module is in SPI configuration when the DCONF field in the MCR is 0b00. The SPI frames can be 32 bits long. The host CPU or a DMA controller transfers the SPI data from the external to the module RAM queues to a TX FIFO buffer. The received data is stored in entries in the RX FIFO buffer. The host CPU or the DMA controller transfers the received data from the RX FIFO to memory external to the module. The operation of FIFO buffers is described in the following sections:

- Transmit First In First Out (TX FIFO) buffering mechanism
- Transmit First In First Out (TX FIFO) buffering mechanism
- Receive First In First Out (RX FIFO) buffering mechanism

The interrupt and DMA request conditions are described in Interrupts/DMA requests.

The SPI configuration supports two block-specific modes—Master mode and Slave mode.In Master mode the module initiates and controls the transfer according to the fields of the executing SPI Command. In Slave mode, the module responds only to transfers initiated by a bus master external to it and the SPI command field space is reserved.

51.4.2.1 Master mode

In SPI Master mode, the module initiates the serial transfers by controlling the SCK and the PCS signals. The executing SPI Command determines which CTARs will be used to set the transfer attributes and which PCS signals to assert. The command field also contains various bits that help with queue management and transfer protocol. See PUSH TX FIFO Register In Master Mode (SPI_PUSHR) for details on the SPI command fields. The data in the executing TX FIFO entry is loaded into the shift register and shifted out on the Serial Out (SOUT) pin. In SPI Master mode, each SPI frame to be transmitted has a command associated with it, allowing for transfer attribute control on a frame by frame basis.

51.4.2.2 Slave mode

In SPI Slave mode the module responds to transfers initiated by an SPI bus master. It does not initiate transfers. Certain transfer attributes such as clock polarity, clock phase, and frame size must be set for successful communication with an SPI master. The SPI Slave mode transfer attributes are set in the CTAR0. The data is shifted out with MSB first. Shifting out of LSB is not supported in this mode.

51.4.2.3 FIFO disable operation

The FIFO disable mechanisms allow SPI transfers without using the TX FIFO or RX FIFO. The module operates as a double-buffered simplified SPI when the FIFOs are disabled. The Transmit and Receive side of the FIFOs are disabled separately. Setting the MCR[DIS_TXF] bit disables the TX FIFO, and setting the MCR[DIS_RXF] bit disables the RX FIFO.

The FIFO disable mechanisms are transparent to the user and to host software. Transmit data and commands are written to the PUSHR and received data is read from the POPR.

When the TX FIFO is disabled:

- SR[TFFF], SR[TFUF] and SR[TXCTR] behave as if there is a one-entry FIFO
- The contents of TXFRs, SR[TXNXTPTR] are undefined

Similarly, when the RX FIFO is disabled, the RFDF, RFOF, and RXCTR fields in the SR behave as if there is a one-entry FIFO, but the contents of the RXFR registers and POPNXTPTR are undefined.

51.4.2.4 Transmit First In First Out (TX FIFO) buffering mechanism

The TX FIFO functions as a buffer of SPI data for transmission. The TX FIFO holds 4 words, each consisting of SPI data. The number of entries in the TX FIFO is device-specific. SPI data is added to the TX FIFO by writing to the Data Field of module PUSH FIFO Register (PUSHR). TX FIFO entries can only be removed from the TX FIFO by being shifted out or by flushing the TX FIFO.

The TX FIFO Counter field (TXCTR) in the module Status Register (SR) indicates the number of valid entries in the TX FIFO. The TXCTR is updated every time a 8- or 16-bit write takes place to PUSHR[TXDATA] or SPI data is transferred into the shift register from the TX FIFO.

The TXNXTPTR field indicates the TX FIFO Entry that will be transmitted during the next transfer. The TXNXTPTR field is incremented every time SPI data is transferred from the TX FIFO to the shift register. The maximum value of the field is equal to the maximum implemented TXFR number and it rolls over after reaching the maximum.

51.4.2.4.1 Filling the TX FIFO

Host software or other intelligent blocks can add (push) entries to the TX FIFO by writing to the PUSHR. When the TX FIFO is not full, the TX FIFO Fill Flag (TFFF) in the SR is set. The TFFF bit is cleared when TX FIFO is full and the DMA controller indicates that a write to PUSHR is complete. Writing a '1' to the TFFF bit also clears it. The TFFF can generate a DMA request or an interrupt request. See Transmit FIFO Fill Interrupt or DMA Request for details.

The module ignores attempts to push data to a full TX FIFO, and the state of the TX FIFO does not change and no error condition is indicated.

51.4.2.4.2 Draining the TX FIFO

The TX FIFO entries are removed (drained) by shifting SPI data out through the shift register. Entries are transferred from the TX FIFO to the shift register and shifted out as long as there are valid entries in the TX FIFO. Every time an entry is transferred from the TX FIFO to the shift register, the TX FIFO Counter decrements by one. At the end of a transfer, the TCF bit in the SR is set to indicate the completion of a transfer. The TX FIFO is flushed by writing a '1' to the CLR_TXF bit in MCR.

If an external bus master initiates a transfer with a module slave while the slave's TX FIFO is empty, the Transmit FIFO Underflow Flag (TFUF) in the slave's SR is set. See Transmit FIFO Underflow Interrupt Request for details.

51.4.2.5 Receive First In First Out (RX FIFO) buffering mechanism

The RX FIFO functions as a buffer for data received on the SIN pin. The RX FIFO holds 4 received SPI data frames. The number of entries in the RX FIFO is device-specific. SPI data is added to the RX FIFO at the completion of a transfer when the received data in the shift register is transferred into the RX FIFO. SPI data are removed (popped) from the RX FIFO by reading the module POP RX FIFO Register (POPR). RX FIFO entries can only be removed from the RX FIFO by reading the POPR or by flushing the RX FIFO.

The RX FIFO Counter field (RXCTR) in the module's Status Register (SR) indicates the number of valid entries in the RX FIFO. The RXCTR is updated every time the POPR is read or SPI data is copied from the shift register to the RX FIFO.

The POPNXTPTR field in the SR points to the RX FIFO entry that is returned when the POPR is read. The POPNXTPTR contains the positive offset from RXFR0 in a number of 32-bit registers. For example, POPNXTPTR equal to two means that the RXFR2 contains the received SPI data that will be returned when the POPR is read. The POPNXTPTR field is incremented every time the POPR is read. The maximum value of the field is equal to the maximum implemented RXFR number and it rolls over after reaching the maximum.

51.4.2.5.1 Filling the RX FIFO

The RX FIFO is filled with the received SPI data from the shift register. While the RX FIFO is not full, SPI frames from the shift register are transferred to the RX FIFO. Every time an SPI frame is transferred to the RX FIFO, the RX FIFO Counter is incremented by one.

If the RX FIFO and shift register are full and a transfer is initiated, the RFOF bit in the SR is set indicating an overflow condition. Depending on the state of the ROOE bit in the MCR, the data from the transfer that generated the overflow is either ignored or shifted in to the shift register. If the ROOE bit is set, the incoming data is shifted in to the shift register. If the ROOE bit is cleared, the incoming data is ignored.

51.4.2.5.2 Draining the RX FIFO

Host CPU or a DMA can remove (pop) entries from the RX FIFO by reading the module POP RX FIFO Register (POPR). A read of the POPR decrements the RX FIFO Counter by one. Attempts to pop data from an empty RX FIFO are ignored and the RX FIFO Counter remains unchanged. The data, read from the empty RX FIFO, is undetermined.

When the RX FIFO is not empty, the RX FIFO Drain Flag (RFDF) in the SR is set. The RFDF bit is cleared when the RX_FIFO is empty and the DMA controller indicates that a read from POPR is complete or by writing a 1 to it.

51.4.3 Module baud rate and clock delay generation

The SCK frequency and the delay values for serial transfer are generated by dividing the system clock frequency by a prescaler and a scaler with the option for doubling the baud rate. The following figure shows conceptually how the SCK signal is generated.

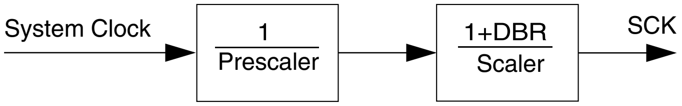


Figure 51-4. Communications clock prescalers and scalers

51.4.3.1 Baud rate generator

The baud rate is the frequency of the SCK. The protocol clock is divided by a prescaler (PBR) and scaler (BR) to produce SCK with the possibility of halving the scaler division. The DBR, PBR, and BR fields in the CTARs select the frequency of SCK by the formula in the BR field description. The following table shows an example of how to compute the baud rate.

Table 51-5. Baud rate computation example

f_P	PBR	Prescaler	BR	Scaler	DBR	Baud rate
100 MHz	0b00	2	0b0000	2	0	25 Mb/s
20 MHz	0b00	2	0b0000	2	1	10 Mb/s

NOTE

The clock frequencies mentioned in the preceding table are given as an example. Refer to the clocking chapter for the frequency used to drive this module in the device.

51.4.3.2 PCS to SCK Delay (t_{CSC})

The PCS to SCK delay is the length of time from assertion of the PCS signal to the first SCK edge. See Figure 51-5 for an illustration of the PCS to SCK delay. The PCSSCK and CSSCK fields in the CTAR*x* registers select the PCS to SCK delay by the formula in the CSSCK field description. The following table shows an example of how to compute the PCS to SCK delay.

Table 51-6. PCS to SCK delay computation example

f_{SYS}	PCSSCK	Prescaler	CSSCK	Scaler	PCS to SCK Delay
100 MHz	0b01	3	0b0100	32	0.96 μs

NOTE

The clock frequency mentioned in the preceding table is given as an example. Refer to the clocking chapter for the frequency used to drive this module in the device.

51.4.3.3 After SCK Delay (t_{ASC})

The After SCK Delay is the length of time between the last edge of SCK and the negation of PCS. See Figure 51-5 and Figure 51-6 for illustrations of the After SCK delay. The PASC and ASC fields in the CTAR*x* registers select the After SCK Delay by the formula in the ASC field description. The following table shows an example of how to compute the After SCK delay.

Table 51-7. After SCK Delay computation example

f_P	PASC	Prescaler	ASC	Scaler	After SCK Delay
100 MHz	0b01	3	0b0100	32	0.96 μs

NOTE

The clock frequency mentioned in the preceding table is given as an example. Refer to the clocking chapter for the frequency used to drive this module in the device.

51.4.3.4 Delay after Transfer (t_{DT})

The Delay after Transfer is the minimum time between negation of the PCS signal for a frame and the assertion of the PCS signal for the next frame. See Figure 51-5 for an illustration of the Delay after Transfer. The PDT and DT fields in the CTAR*x* registers select the Delay after Transfer by the formula in the DT field description. The following table shows an example of how to compute the Delay after Transfer.

Table 51-8. Delay after Transfer computation example

f_P	PDT	Prescaler	DT	Scaler	Delay after Transfer
100 MHz	0b01	3	0b1110	32768	0.98 ms

NOTE

The clock frequency mentioned in the preceding table is given as an example. Refer to the clocking chapter for the frequency used to drive this module in the device.

When in Non-Continuous Clock mode the t_{DT} delay is configured according to the equation specified in the CTAR[DT] field description. When in Continuous Clock mode, the delay is fixed at 1 SCK period.

51.4.4 Transfer formats

The SPI serial communication is controlled by the Serial Communications Clock (SCK) signal and the PCS signals. The SCK signal provided by the master device synchronizes shifting and sampling of the data on the SIN and SOUT pins. The PCS signals serve as enable signals for the slave devices.

In Master mode, the CPOL and CPHA bits in the Clock and Transfer Attributes Registers (CTARn) select the polarity and phase of the serial clock, SCK.

- CPOL - Selects the idle state polarity of the SCK
- CPHA - Selects if the data on SOUT is valid before or on the first SCK edge

Even though the bus slave does not control the SCK signal, in Slave mode the values of CPOL and CPHA must be identical to the master device settings to ensure proper transmission. In SPI Slave mode, only CTAR0 is used.

The module supports four different transfer formats:

- Classic SPI with CPHA=0
- Classic SPI with CPHA=1
- Modified Transfer Format with CPHA = 0
- Modified Transfer Format with CPHA = 1

A modified transfer format is supported to allow for high-speed communication with peripherals that require longer setup times. The module can sample the incoming data later than halfway through the cycle to give the peripheral more setup time. The MTFE bit in the MCR selects between Classic SPI Format and Modified Transfer Format.

In the interface configurations, the module provides the option of keeping the PCS signals asserted between frames. See Continuous Selection Format for details.

51.4.4.1 Classic SPI Transfer Format (CPHA = 0)

The transfer format shown in following figure is used to communicate with peripheral SPI slave devices where the first data bit is available on the first clock edge. In this format, the master and slave sample their SIN pins on the odd-numbered SCK edges and change the data on their SOUT pins on the even-numbered SCK edges.

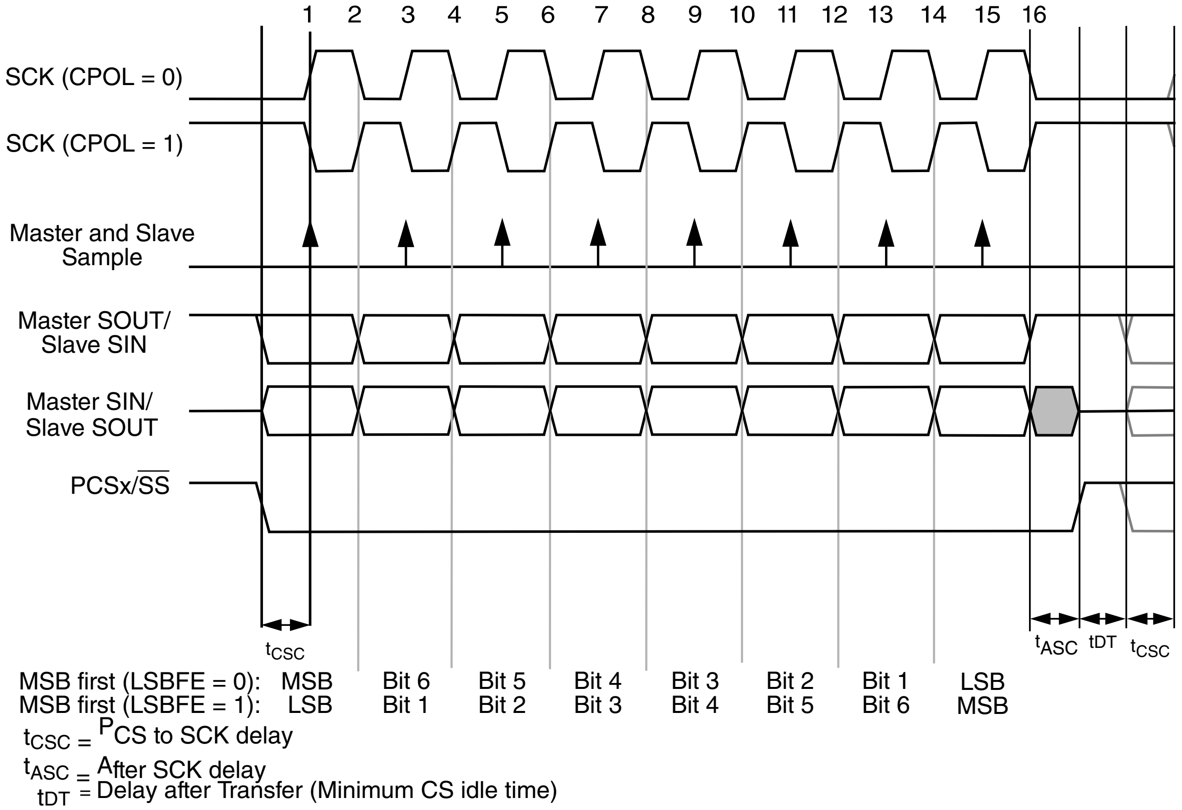


Figure 51-5. Module transfer timing diagram (MTFE=0, CPHA=0, FMSZ=8)

The master initiates the transfer by placing its first data bit on the SOUT pin and asserting the appropriate peripheral chip select signals to the slave device. The slave responds by placing its first data bit on its SOUT pin. After the t_{CSC} delay elapses, the master outputs the first edge of SCK. The master and slave devices use this edge to sample the first input data bit on their serial data input signals. At the second edge of the SCK, the master and slave devices place their second data bit on their serial data output signals. For the rest of the frame the master and the slave sample their SIN pins on the odd-numbered clock edges and changes the data on their SOUT pins on the even-numbered clock edges. After the last clock edge occurs, a delay of t_{ASC} is inserted before the master negates the PCS signals. A delay of t_{DT} is inserted before a new frame transfer can be initiated by the master.

51.4.4.2 Classic SPI Transfer Format (CPHA = 1)

This transfer format shown in the following figure is used to communicate with peripheral SPI slave devices that require the first SCK edge before the first data bit becomes available on the slave SOUT pin. In this format, the master and slave devices change the data on their SOUT pins on the odd-numbered SCK edges and sample the data on their SIN pins on the even-numbered SCK edges.

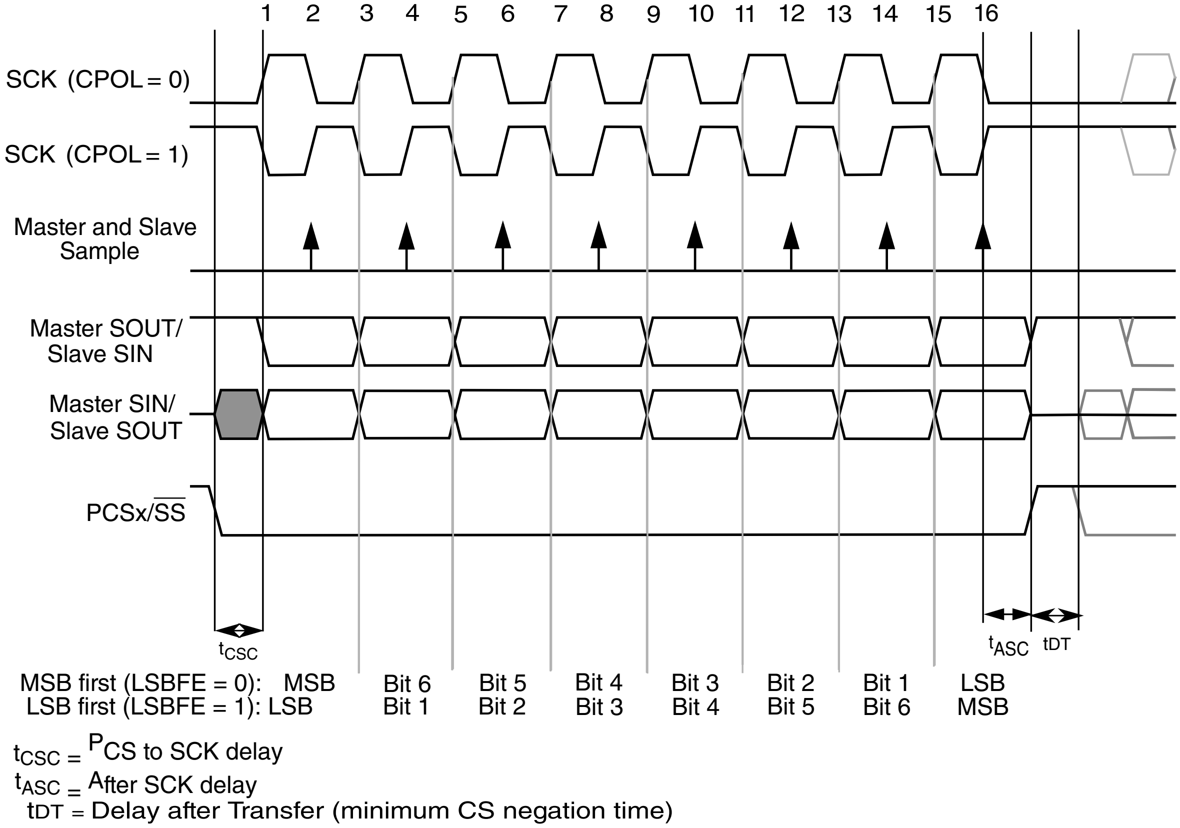


Figure 51-6. Module transfer timing diagram (MTFE=0, CPHA=1, FMSZ=8)

The master initiates the transfer by asserting the PCS signal to the slave. After the t_{CSC} delay has elapsed, the master generates the first SCK edge and at the same time places valid data on the master SOUT pin. The slave responds to the first SCK edge by placing its first data bit on its slave SOUT pin.

At the second edge of the SCK the master and slave sample their SIN pins. For the rest of the frame the master and the slave change the data on their SOUT pins on the odd-numbered clock edges and sample their SIN pins on the even-numbered clock edges. After the last clock edge occurs, a delay of t_{ASC} is inserted before the master negates the PCS signal. A delay of t_{DT} is inserted before a new frame transfer can be initiated by the master.

51.4.4.3 Modified SPI Transfer Format (MTFE = 1, CPHA = 0)

In this Modified Transfer Format both the master and the slave sample later in the SCK period than in Classic SPI mode to allow the logic to tolerate more delays in device pads and board traces. These delays become a more significant fraction of the SCK period as the SCK period decreases with increasing baud rates.

The master and the slave place data on the SOUT pins at the assertion of the PCS signal. After the PCS to SCK delay has elapsed the first SCK edge is generated. The slave samples the master SOUT signal on every odd numbered SCK edge. The DSPI in the slave mode when the MTFE bit is set also places new data on the slave SOUT on every odd numbered clock edge. Regular external slave, configured with CPHA=0 format drives its SOUT output at every even numbered SCK clock edge.

The DSPI master places its second data bit on the SOUT line one protocol clock after odd numbered SCK edge if the protocol clock frequency to SCK frequency ratio is higher than three. If this ratio is below four the master changes SOUT at odd numbered SCK edge. The point where the master samples the SIN is selected by the DSPI_MCR[SMPL_PT] field. The master sample point can be delayed by one or two protocol clock cycles. The SMPL_PT field should be set to 0 if the protocol to SCK frequency ratio is less than 4. However if this ratio is less than 4, the actual sample point is delayed by one protocol clock cycle automatically by the design.

The following timing diagrams illustrate the DSPI operation with MTFE=1. Timing delays shown are:

- T_{csc} - PCS to SCK assertion delay
- T_{acs} - After SCK PCS negation delay
- T_{su_ms} - master SIN setup time
- T_{hd_ms} - master SIN hold time
- T_{vd_sl} - slave data output valid time, time between slave data output SCK driving edge and data becomes valid.
- T_{su_sl} - data setup time on slave data input
- T_{hd_sl} - data hold time on slave data input
- T_{sys} - protocol clock period.

The following figure shows the modified transfer format for CPHA = 0 and Fsys/Fsck = 4. Only the condition where CPOL = 0 is illustrated. Solid triangles show the data sampling clock edges. The two possible slave behavior are shown.

- Signal, marked "SOUT of Ext Slave", presents regular SPI slave serial output.
- Signal, marked "SOUT of DSPI Slave", presents DSPI in the slave mode with MTFE bit set.

Other MTFE = 1 diagrams show DSPI SIN input as being driven by a regular external SPI slave, configured according DSPI master CPHA programming.

Note

In the following diagrams, f_{sys} represents the protocol clock frequency from which the Baud frequency f_{sck} is derived.

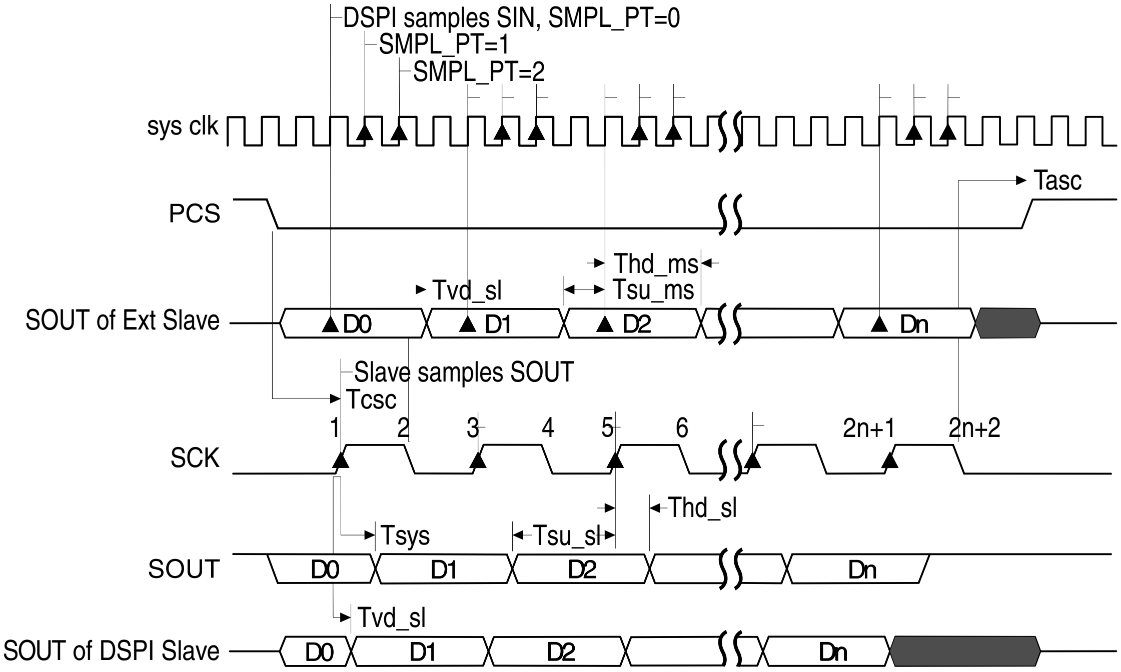


Figure 51-7. DSPI Modified Transfer Format (MTFE=1, CPHA=0, $f_{sck} = f_{sys}/4$)

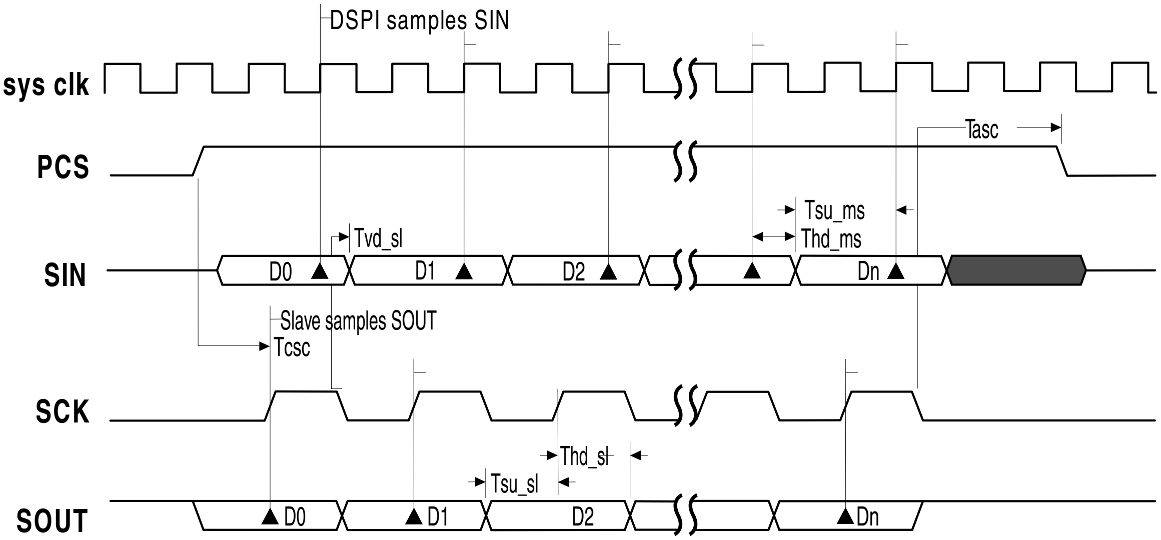


Figure 51-8. DSPI Modified Transfer Format (MTFE=1, CPHA=0, $f_{sck} = f_{sys}/2$)

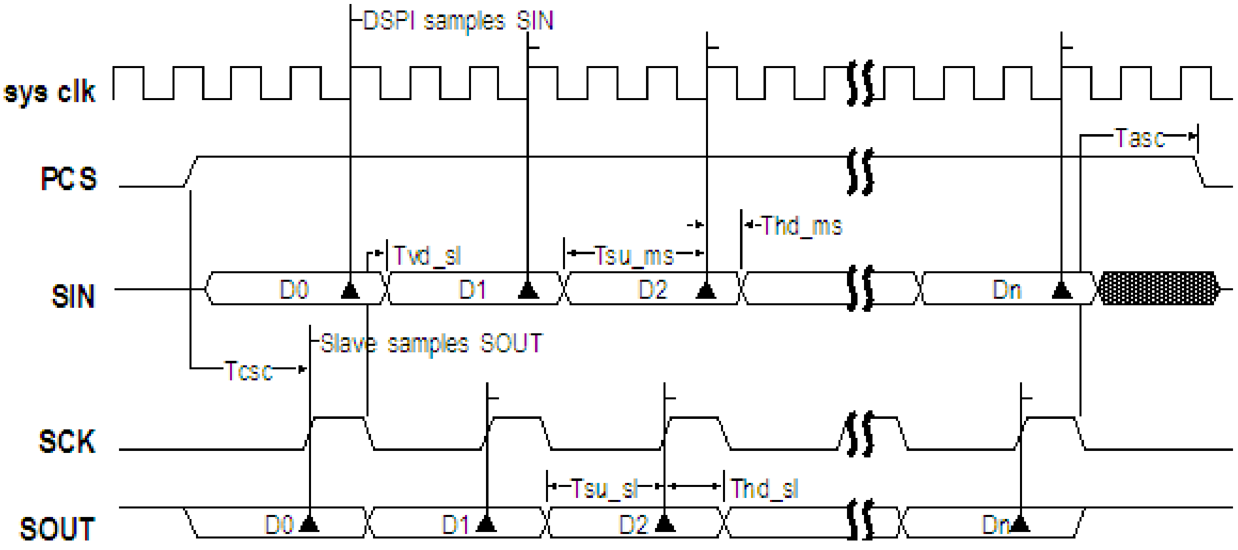


Figure 51-9. DSPI Modified Transfer Format (MTFE=1, CPHA=0, f_{sck} = f_{sys}/3)

51.4.4.4 Modified SPI Transfer Format (MTFE = 1, CPHA = 1)

The following figures show the Modified Transfer Format for CPHA = 1. Only the condition, where CPOL = 0 is shown. At the start of a transfer the DSPI asserts the PCS signal to the slave device. After the PCS to SCK delay has elapsed the master and the slave put data on their SOUT pins at the first edge of SCK . The slave samples the master SOUT signal on the even numbered edges of SCK. The master samples the slave SOUT signal on the odd numbered SCK edges starting with the third SCK edge. The slave samples the last bit on the last edge of the SCK. The master samples the last slave SOUT bit one half SCK cycle after the last edge of SCK. No clock edge will be visible on the master SCK pin during the sampling of the last bit. **The SCK to PCS delay and the After SCK delay must be greater or equal to half of the SCK period.**

NOTE

When MTFE=1 with continuous SCK enabled (MCR [CONT_SCKE] =1) in master mode, configure CTAR[LSBFE]=0 for correct operations while receiving unequal length frames. If PUSHR[CONT] is also set for back to back frame transfer, also configure the frame size of the first frame as less than or equal to the frame size of the next frame. In this scenario, make sure that for all received frames, the bits are read equal to their respective frame sizes and any extra bits during POP operation are masked.

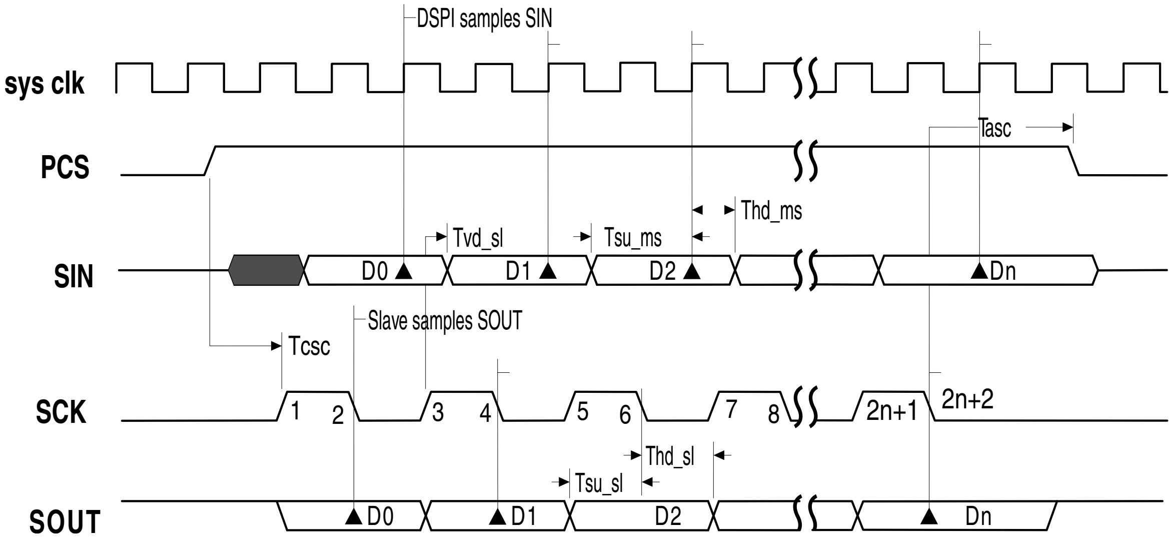


Figure 51-10. DSPI Modified Transfer Format (MTFE=1, CPHA=1, $f_{sck} = f_{sys}/2$)

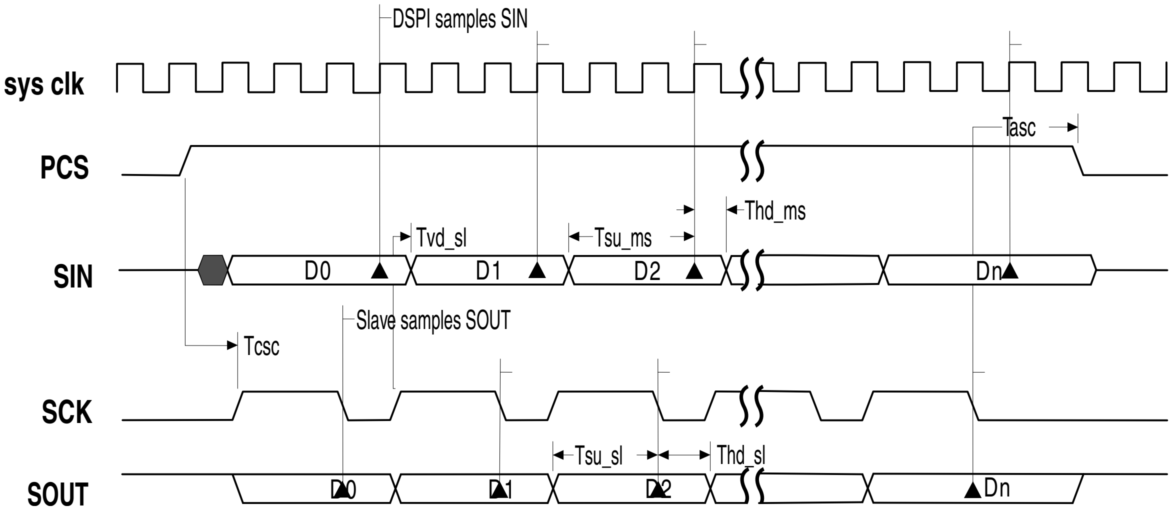


Figure 51-11. DSPI Modified Transfer Format (MTFE=1, CPHA=1, $f_{sck} = f_{sys}/3$)

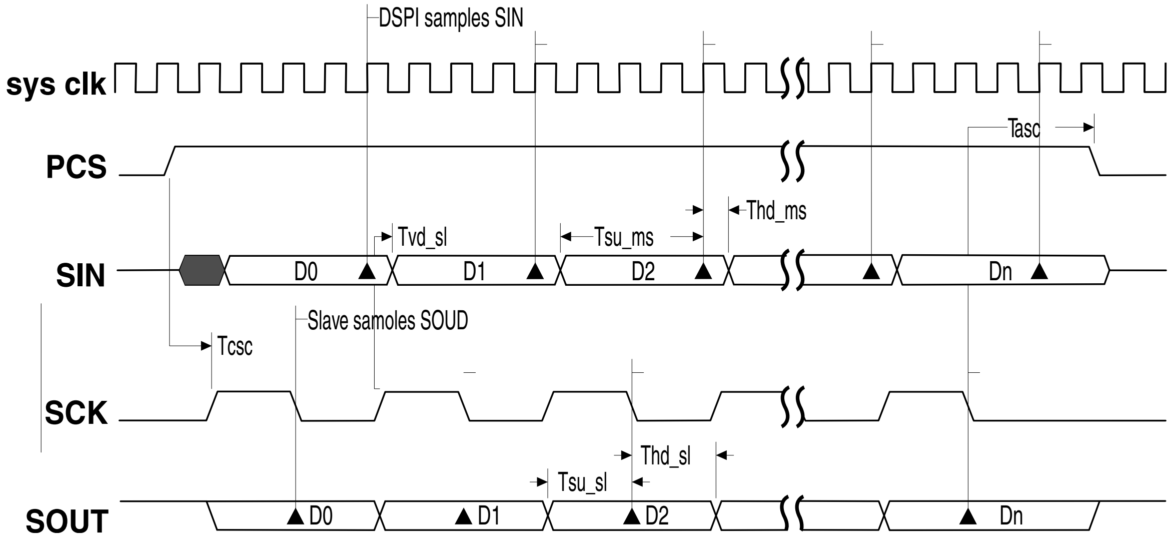


Figure 51-12. DSPI Modified Transfer Format (MTFE=1, CPHA=1, $f_{sck} = f_{sys}/4$)

51.4.4.5 Continuous Selection Format

Some peripherals must be deselected between every transfer. Other peripherals must remain selected between several sequential serial transfers. The Continuous Selection Format provides the flexibility to handle the following case. The Continuous Selection Format is enabled for the SPI configuration by setting the CONT bit in the SPI command.

When the CONT bit = 0, the module drives the asserted Chip Select signals to their idle states in between frames. The idle states of the Chip Select signals are selected by the PCSISn bits in the MCR. The following timing diagram is for two four-bit transfers with CPHA = 1 and CONT = 0.

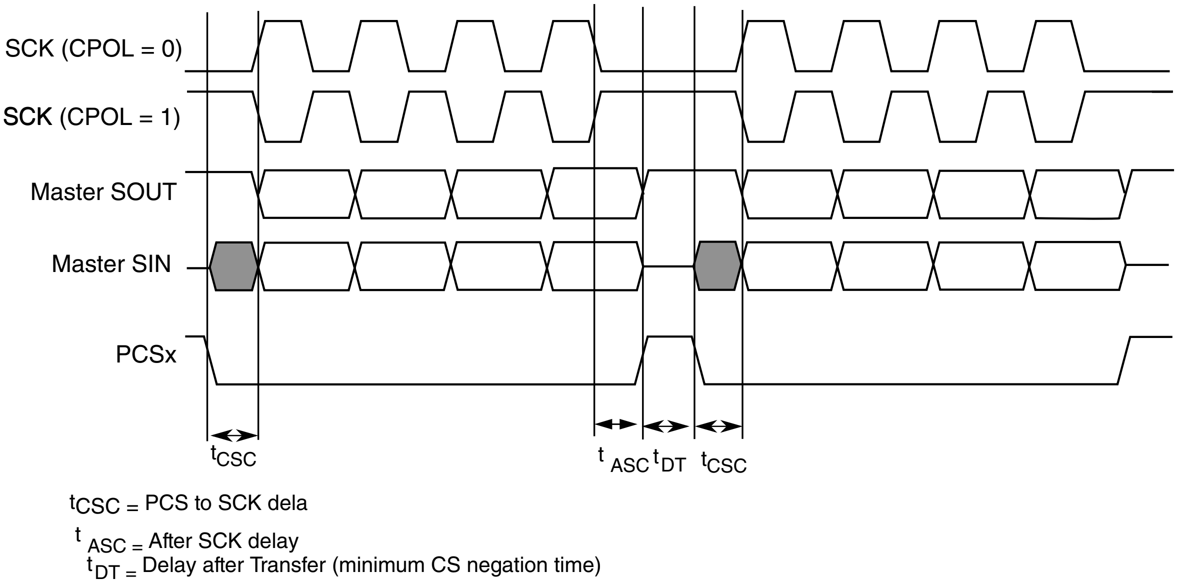


Figure 51-13. Example of non-continuous format (CPHA=1, CONT=0)

When the CONT bit = 1, the PCS signal remains asserted for the duration of the two transfers. The Delay between Transfers (t_{DT}) is not inserted between the transfers. The following figure shows the timing diagram for two four-bit transfers with CPHA = 1 and CONT = 1.

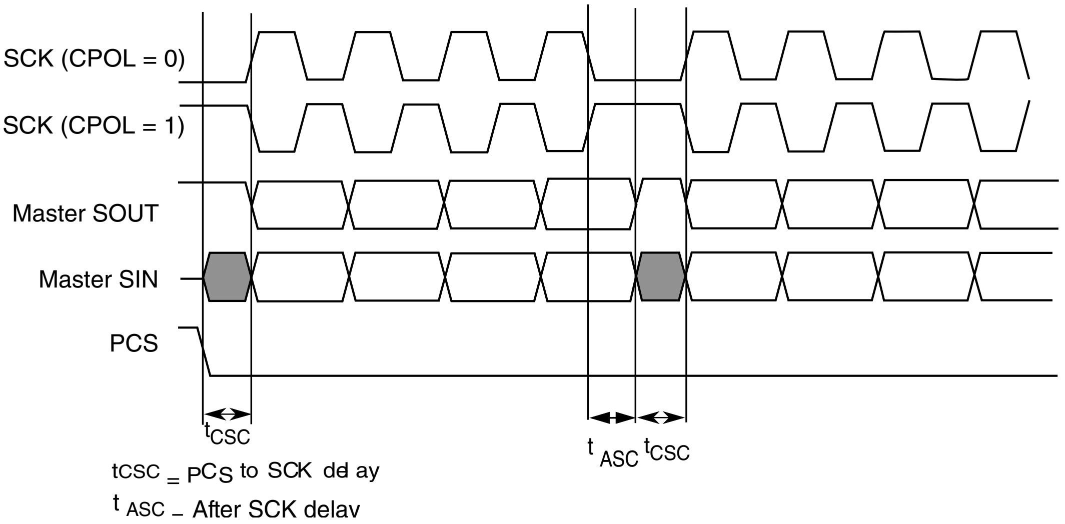


Figure 51-14. Example of continuous transfer (CPHA=1, CONT=1)

When using the module with continuous selection follow these rules:

- All transmit commands must have the same PCSn bits programming.
- The CTARs, selected by transmit commands, must be programmed with the same transfer attributes. Only FMSZ field can be programmed differently in these CTARs.

- When transmitting multiple frames in this mode, the user software must ensure that the last frame has the PUSHR[CONT] bit deasserted in Master mode and the user software must provide sufficient frames in the TX_FIFO to be sent out in Slave mode and the master deasserts the PCSn at end of transmission of the last frame.
- PUSHR[CONT] must be deasserted before asserting MCR[HALT] in master mode. This will make sure that the PCSn signals are deasserted. Asserting MCR[HALT] during continuous transfer will cause the PCSn signals to remain asserted and hence Slave Device cannot transition from Running to Stopped state.

NOTE

User must fill the TX FIFO with the number of entries that will be concatenated together under one PCS assertion for both master and slave before the TX FIFO becomes empty.

When operating in Slave mode, ensure that when the last entry in the TX FIFO is completely transmitted, that is, the corresponding TCF flag is asserted and TXFIFO is empty, the slave is deselected for any further serial communication; otherwise, an underflow error occurs.

51.4.5 Continuous Serial Communications Clock

The module provides the option of generating a Continuous SCK signal for slave peripherals that require a continuous clock.

Continuous SCK is enabled by setting the CONT_SCKE bit in the MCR. Enabling this bit generates the Continuous SCK only if MCR[HALT] bit is low. Continuous SCK is valid in all configurations.

Continuous SCK is only supported for CPHA=1. Clearing CPHA is ignored if the CONT_SCKE bit is set. Continuous SCK is supported for Modified Transfer Format.

Clock and transfer attributes for the Continuous SCK mode are set according to the following rules:

- When the module is in SPI configuration, CTAR0 is used initially. At the start of each SPI frame transfer, the CTAR specified by the CTAS for the frame is used.
- In all configurations, the currently selected CTAR remains in use until the start of a frame with a different CTAR specified, or the Continuous SCK mode is terminated.

It is recommended to keep the baud rate the same while using the Continuous SCK. Switching clock polarity between frames while using Continuous SCK can cause errors in the transfer. Continuous SCK operation is not guaranteed if the module is put into the External Stop mode or Module Disable mode.

Enabling Continuous SCK disables the PCS to SCK delay and the Delay after Transfer (t_{DT}) is fixed to one SCK cycle. The following figure is the timing diagram for Continuous SCK format with Continuous Selection disabled.

NOTE

> In Continuous SCK mode, for the SPI transfer CTAR0 should always be used, and the TX FIFO must be cleared using the MCR[CLR_TXF] field before initiating transfer.

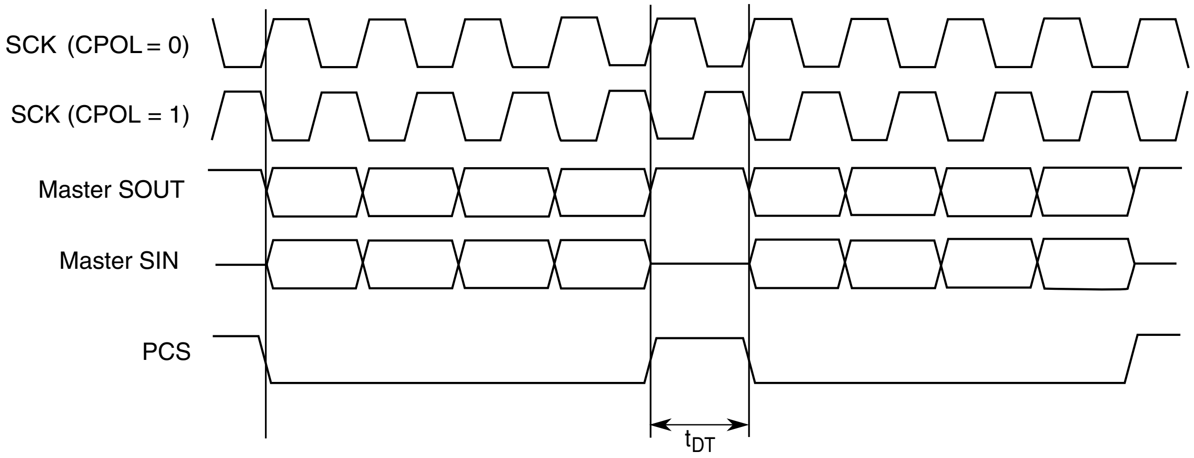


Figure 51-15. Continuous SCK Timing Diagram (CONT=0)

If the CONT bit in the TX FIFO entry is set, PCS remains asserted between the transfers. Under certain conditions, SCK can continue with PCS asserted, but with no data being shifted out of SOUT, that is, SOUT pulled high. This can cause the slave to receive incorrect data. Those conditions include:

- Continuous SCK with CONT bit set, but no data in the TX FIFO.
- Continuous SCK with CONT bit set and entering Stopped state (refer to Start and Stop of module transfers).
- Continuous SCK with CONT bit set and entering Stop mode or Module Disable mode.

The following figure shows timing diagram for Continuous SCK format with Continuous Selection enabled.

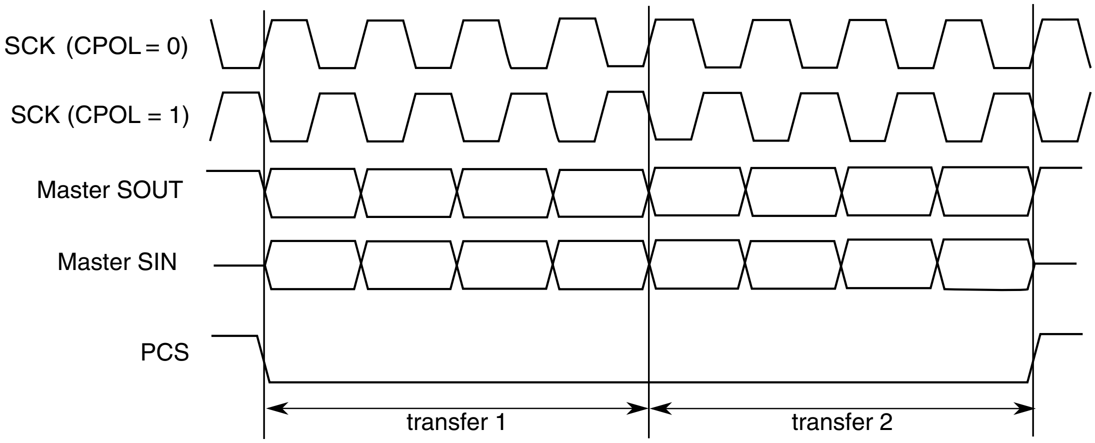


Figure 51-16. Continuous SCK timing diagram (CONT=1)

51.4.6 Slave Mode Operation Constraints

Slave mode logic shift register is buffered. This allows data streaming operation, when the module is permanently selected and data is shifted in with a constant rate.

The transmit data is transferred at second SCK clock edge of the each frame to the shift register if the $\overline{SS}$ signal is asserted and any time when transmit data is ready and $\overline{SS}$ signal is negated.

Received data is transferred to the receive buffer at last SCK edge of each frame, defined by frame size programmed to the CTAR0/1 register. Then the data from the buffer is transferred to the RXFIFO or DDR register.

If the $\overline{SS}$ negates before that last SCK edge, the data from shift register is lost.

51.4.7 Interrupts/DMA requests

The module has several conditions that can generate only interrupt requests and two conditions that can generate interrupt or DMA requests. The following table lists these conditions.

Table 51-9. Interrupt and DMA request conditions

Condition	Flag	Interrupt	DMA
End of Queue (EOQ)	EOQF	Yes	-
TX FIFO Fill	TFFF	Yes	Yes
Transfer Complete	TCF	Yes	-
TX FIFO Underflow	TFUF	Yes	-

Table continues on the next page...

Table 51-9. Interrupt and DMA request conditions (continued)

Condition	Flag	Interrupt	DMA
RX FIFO Drain	RFDF	Yes	Yes
RX FIFO Overflow	RFOF	Yes	-

Each condition has a flag bit in the module Status Register (SR) and a Request Enable bit in the DMA/Interrupt Request Select and Enable Register (RSER). Certain flags (as shown in above table) generate interrupt requests or DMA requests depending on configuration of RSER register.

The module also provides a global interrupt request line, which is asserted when any of individual interrupt requests lines is asserted.

51.4.7.1 End Of Queue interrupt request

The End Of Queue (EOQ) interrupt request indicates that the end of a transmit queue is reached. The module generates the interrupt request when EOQ interrupt requests are enabled (RSER[EOQF_RE]) and the EOQ bit in the executing SPI command is 1.

The module generates the interrupt request when the last bit of the SPI frame with EOQ bit set is transmitted.

51.4.7.2 Transmit FIFO Fill Interrupt or DMA Request

The Transmit FIFO Fill Request indicates that the TX FIFO is not full. The Transmit FIFO Fill Request is generated when the number of entries in the TX FIFO is less than the maximum number of possible entries, and the TFFF_RE bit in the RSER is set. The TFFF_DIRS bit in the RSER selects whether a DMA request or an interrupt request is generated.

NOTE

TFFF flag clears automatically when DMA is used to fill TX FIFO.

To clear TFFF when not using DMA, follow these steps for every PUSH performed using CPU to fill TX FIFO:

1. Wait until TFFF = 1.
2. Write data to PUSHR using CPU.
3. Clear TFFF by writing a 1 to its location. If TX FIFO is not full, this flag will not clear.

51.4.7.3 Transfer Complete Interrupt Request

The Transfer Complete Request indicates the end of the transfer of a serial frame. The Transfer Complete Request is generated at the end of each frame transfer when the TCF_RE bit is set in the RSER.

51.4.7.4 Transmit FIFO Underflow Interrupt Request

The Transmit FIFO Underflow Request indicates that an underflow condition in the TX FIFO has occurred. The transmit underflow condition is detected only for the module operating in Slave mode and SPI configuration . The TFUF bit is set when the TX FIFO of the module is empty, and a transfer is initiated from an external SPI master. If the TFUF bit is set while the TFUF_RE bit in the RSER is set, an interrupt request is generated.

51.4.7.5 Receive FIFO Drain Interrupt or DMA Request

The Receive FIFO Drain Request indicates that the RX FIFO is not empty. The Receive FIFO Drain Request is generated when the number of entries in the RX FIFO is not zero, and the RFDF_RE bit in the RSER is set. The RFDF_DIRS bit in the RSER selects whether a DMA request or an interrupt request is generated.

51.4.7.6 Receive FIFO Overflow Interrupt Request

The Receive FIFO Overflow Request indicates that an overflow condition in the RX FIFO has occurred. A Receive FIFO Overflow request is generated when RX FIFO and shift register are full and a transfer is initiated. The RFOF_RE bit in the RSER must be set for the interrupt request to be generated.

Depending on the state of the ROOE bit in the MCR, the data from the transfer that generated the overflow is either ignored or shifted in to the shift register. If the ROOE bit is set, the incoming data is shifted in to the shift register. If the ROOE bit is cleared, the incoming data is ignored.

51.4.8 Power saving features

The module supports following power-saving strategies:

- External Stop mode
- Module Disable mode – Clock gating of non-memory mapped logic

51.4.8.1 Stop mode (External Stop mode)

This module supports the Stop mode protocol. When a request is made to enter External Stop mode, the module acknowledges the request . If a serial transfer is in progress, then this module waits until it reaches the frame boundary before it is ready to have its clocks shut off . While the clocks are shut off, this module's memory-mapped logic is not accessible. This also puts the module in STOPPED state. The SR[TXRXS] bit is cleared to indicate STOPPED state. The states of the interrupt and DMA request signals cannot be changed while in External Stop mode.

51.4.8.2 Module Disable mode

Module Disable mode is a block-specific mode that the module can enter to save power. Host CPU can initiate the Module Disable mode by setting the MDIS bit in the MCR. The Module Disable mode can also be initiated by hardware.

When the MDIS bit is set, the module negates the Clock Enable signal at the next frame boundary. Once the Clock Enable signal is negated, it is said to have entered Module Disable Mode. This also puts the module in STOPPED state. The SR[TXRXS] bit is cleared to indicate STOPPED state.If implemented, the Clock Enable signal can stop the clock to the non-memory mapped logic. When Clock Enable is negated, the module is in a dormant state, but the memory mapped registers are still accessible. Certain read or write operations have a different effect when the module is in the Module Disable mode. Reading the RX FIFO Pop Register does not change the state of the RX FIFO. Similarly, writing to the PUSHR Register does not change the state of the TX FIFO. Clearing either of the FIFOs has no effect in the Module Disable mode. Changes to the DIS_TXF and DIS_RXF fields of the MCR have no effect in the Module Disable mode. In the Module Disable mode, all status bits and register flags in the module return the correct values when read, but writing to them has no effect. Writing to the TCR during Module Disable mode has no effect. Interrupt and DMA request signals cannot be cleared while in the Module Disable mode.

51.5 Initialization/application information

This section describes how to initialize the module.

51.5.1 How to manage queues

The queues are not part of the module, but it includes features in support of queue management. Queues are primarily supported in SPI configuration.

1. When module executes last command word from a queue, the EOQ bit in the command word is set to indicate it that this is the last entry in the queue.
2. At the end of the transfer, corresponding to the command word with EOQ set is sampled, the EOQ flag (EOQF) in the SR is set.
3. The setting of the EOQF flag disables serial transmission and reception of data, putting the module in the Stopped state. The TXRXS bit is cleared to indicate the Stopped state.
4. The DMA can continue to fill TX FIFO until it is full or step 5 occurs.
5. Disable DMA transfers by disabling the DMA enable request for the DMA channel assigned to TX FIFO and RX FIFO. This is done by clearing the corresponding DMA enable request bits in the DMA Controller.
6. Ensure all received data in RX FIFO has been transferred to memory receive queue by reading the RXCNT in SR or by checking RFDF in the SR after each read operation of the POPR.
7. Modify DMA descriptor of TX and RX channels for new queues
8. Flush TX FIFO by writing a 1 to the CLR_TXF bit in the MCR. Flush RX FIFO by writing a '1' to the CLR_RXF bit in the MCR.
9. Clear transfer count either by setting CTCNT bit in the command word of the first entry in the new queue or via CPU writing directly to SPI_TCNT field in the TCR.
10. Enable DMA channel by enabling the DMA enable request for the DMA channel assigned to the module TX FIFO, and RX FIFO by setting the corresponding DMA set enable request bit.
11. Enable serial transmission and serial reception of data by clearing the EOQF bit.

51.5.2 Switching Master and Slave mode

When changing modes in the module, follow the steps below to guarantee proper operation.

1. Halt it by setting MCR[HALT].
2. Clear the transmit and receive FIFOs by writing a 1 to the CLR_TXF and CLR_RXF bits in MCR.
3. Set the appropriate mode in MCR[MSTR] and enable it by clearing MCR[HALT].

51.5.3 Initializing Module in Master/Slave Modes

Once the appropriate mode in MCR[MSTR] is configured, the module is enabled by clearing MCR[HALT]. It should be ensured that module Slave is enabled before enabling it's Master. This ensures the Slave is ready to be communicated with, before Master initializes communication.

51.5.4 Baud rate settings

The following table shows the baud rate that is generated based on the combination of the baud rate prescaler PBR and the baud rate scaler BR in the CTARs. The values calculated assume a 100 MHz protocol frequency and the double baud rate DBR bit is cleared.

NOTE

> The clock frequency mentioned above is given as an example in this chapter. See the clocking chapter for the frequency used to drive this module in the device.

Table 51-10. Baud rate values (bps)

		Baud rate divider prescaler values			
		2	3	5	7
Baud Rate Scaler Values	2	25.0M	16.7M	10.0M	7.14M
	4	12.5M	8.33M	5.00M	3.57M
	6	8.33M	5.56M	3.33M	2.38M
	8	6.25M	4.17M	2.50M	1.79M
	16	3.12M	2.08M	1.25M	893k
	32	1.56M	1.04M	625k	446k
	64	781k	521k	312k	223k
	128	391k	260k	156k	112k
	256	195k	130k	78.1k	55.8k
	512	97.7k	65.1k	39.1k	27.9k
	1024	48.8k	32.6k	19.5k	14.0k
	2048	24.4k	16.3k	9.77k	6.98k

Table continues on the next page...

Table 51-10. Baud rate values (bps) (continued)

		Baud rate divider prescaler values			
		2	3	5	7
	4096	12.2k	8.14k	4.88k	3.49k
	8192	6.10k	4.07k	2.44k	1.74k
	16384	3.05k	2.04k	1.22k	872
	32768	1.53k	1.02k	610	436

51.5.5 Delay settings

The following table shows the values for the Delay after Transfer (t_{DT}) and CS to SCK Delay (T_{CSC}) that can be generated based on the prescaler values and the scaler values set in the CTARs. The values calculated assume a 100 MHz protocol frequency.

NOTE

The clock frequency mentioned above is given as an example in this chapter. See the clocking chapter for the frequency used to drive this module in the device.

Table 51-11. Delay values

		Delay prescaler values			
		1	3	5	7
Delay scaler values	2	20.0 ns	60.0 ns	100.0 ns	140.0 ns
	4	40.0 ns	120.0 ns	200.0 ns	280.0 ns
	8	80.0 ns	240.0 ns	400.0 ns	560.0 ns
	16	160.0 ns	480.0 ns	800.0 ns	1.1 μs
	32	320.0 ns	960.0 ns	1.6 μs	2.2 μs
	64	640.0 ns	1.9 μs	3.2 μs	4.5 μs
	128	1.3 μs	3.8 μs	6.4 μs	9.0 μs
	256	2.6 μs	7.7 μs	12.8 μs	17.9 μs
	512	5.1 μs	15.4 μs	25.6 μs	35.8 μs
	1024	10.2 μs	30.7 μs	51.2 μs	71.7 μs
	2048	20.5 μs	61.4 μs	102.4 μs	143.4 μs
	4096	41.0 μs	122.9 μs	204.8 μs	286.7 μs
	8192	81.9 μs	245.8 μs	409.6 μs	573.4 μs
	16384	163.8 μs	491.5 μs	819.2 μs	1.1 ms
	32768	327.7 μs	983.0 μs	1.6 ms	2.3 ms
	65536	655.4 μs	2.0 ms	3.3 ms	4.6 ms

51.5.6 Calculation of FIFO pointer addresses

Complete visibility of the FIFO contents is available through the FIFO registers, and valid entries can be identified through a memory-mapped pointer and counter for each FIFO. The pointer to the first-in entry in each FIFO is memory mapped. For the TX FIFO the first-in pointer is the Transmit Next Pointer (TXNXTPTR). For the RX FIFO the first-in pointer is the Pop Next Pointer (POPNXTPTR). The following figure illustrates the concept of first-in and last-in FIFO entries along with the FIFO Counter. The TX FIFO is chosen for the illustration, but the concepts carry over. See Transmit First In First Out (TX FIFO) buffering mechanism and Receive First In First Out (RX FIFO) buffering mechanism for details on the FIFO operation.

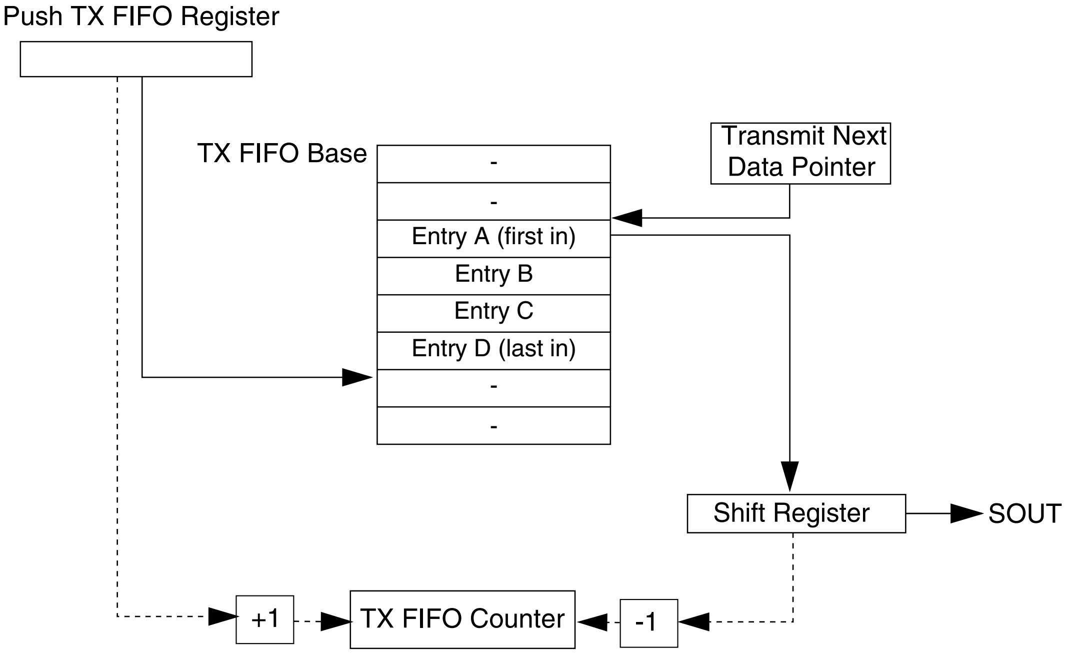


Figure 51-17. TX FIFO pointers and counter

51.5.6.1 Address Calculation for the First-in Entry and Last-in Entry in the TX FIFO

The memory address of the first-in entry in the TX FIFO is computed by the following equation:

$$\text{First-in EntryAddress} = \text{TXFIFOBase} + (4 \times \text{TXNXTPTR})$$

The memory address of the last-in entry in the TX FIFO is computed by the following equation:

$$\text{Last-inEntryaddress} = \text{TXFIFOBase} + 4 \times (\text{TXCTR} + \text{TXNXTPTR} - 1)\text{mod}(\text{TXFIFOdepth})$$

TX FIFO Base - Base address of TX FIFO
TXCTR - TX FIFO Counter
TXNXTPTR - Transmit Next Pointer
TX FIFO Depth - Transmit FIFO depth, implementation specific

51.5.6.2 Address Calculation for the First-in Entry and Last-in Entry in the RX FIFO

The memory address of the first-in entry in the RX FIFO is computed by the following equation:

$$\text{First-in EntryAddress} = \text{RX FIFOBase} + (4 \times \text{POPNXTPTR})$$

The memory address of the last-in entry in the RX FIFO is computed by the following equation:

$$\text{Last-inEntryaddress} = \text{RX FIFO Base} + 4 \times (\text{RXCTR} + \text{POPNXTPTR} - 1)\text{mod}(\text{RXFIFOdepth})$$

RX FIFO Base - Base address of RX FIFO
RXCTR - RX FIFO counter
POPNXTPTR - Pop Next Pointer
RX FIFO Depth - Receive FIFO depth, implementation specific

Chapter 52
MCU: Inter-Integrated Circuit (I2C)

52.1 Introduction

NOTE

> For the chip-specific implementation details of this module's instances, see the chip configuration information.

The inter-integrated circuit (I^{2}C, I2C, or IIC) module provides a method of communication between a number of devices.

The interface is designed to operate up to 100 kbit/s with maximum bus loading and timing. The I2C device is capable of operating at higher baud rates, up to a maximum of clock/20, with reduced bus loading. The maximum communication length and the number of devices that can be connected are limited by a maximum bus capacitance of 400 pF. The I2C module also complies with the *System Management Bus (SMBus) Specification, version 2*.

52.1.1 Features

The I2C module has the following features:

- Compatible with *The I^{2}C-Bus Specification*
- Multimaster operation
- Software programmable for one of 64 different serial clock frequencies
- Software-selectable acknowledge bit
- Interrupt-driven byte-by-byte data transfer
- Arbitration-lost interrupt with automatic mode switching from master to slave
- Calling address identification interrupt
- START and STOP signal generation and detection
- Repeated START signal generation and detection
- Acknowledge bit generation and detection

- Bus busy detection
- General call recognition
- 10-bit address extension
- Support for *System Management Bus (SMBus) Specification, version 2*
- Programmable input glitch filter
- Low power mode wakeup on slave address match
- Range slave address support
- DMA support

52.1.2 Modes of operation

The I2C module's operation in various low power modes is as follows:

- Run mode: This is the basic mode of operation. To conserve power in this mode, disable the module.
- Wait mode: The module continues to operate when the core is in Wait mode and can provide a wakeup interrupt.
- Stop mode: The module is inactive in Stop mode for reduced power consumption, except that address matching is enabled in Stop mode. The STOP instruction does not affect the I2C module's register states.

52.1.3 Block diagram

The following figure is a functional block diagram of the I2C module.

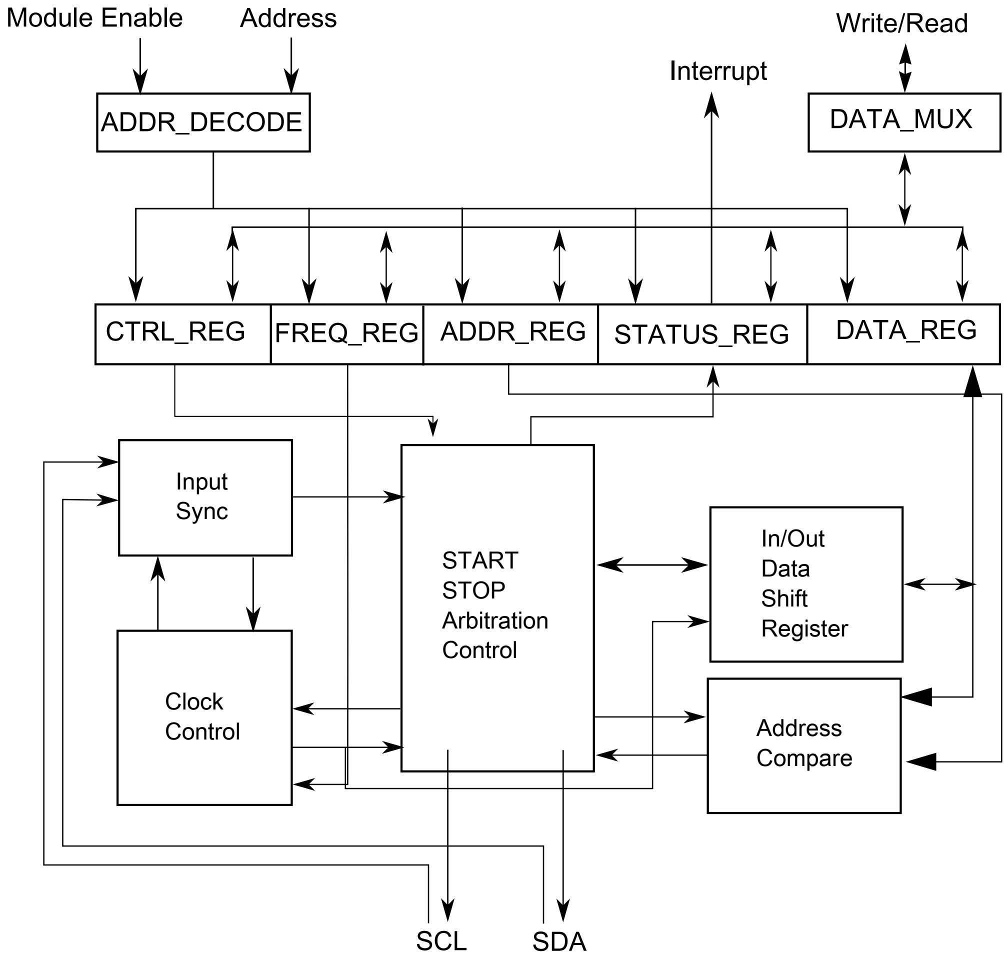


Figure 52-1. I2C Functional block diagram

52.2 I^{2}C signal descriptions

The signal properties of I^{2}C are shown in the table found here.

Table 52-1. I^{2}C signal descriptions

Signal	Description	I/O
SCL	Bidirectional serial clock line of the I^{2}C system.	I/O
SDA	Bidirectional serial data line of the I^{2}C system.	I/O

52.3 Memory map/register definition

This section describes in detail all I2C registers accessible to the end user.

I2C memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4006_6000	I2C Address Register 1 (I2C0_A1)	8	R/W	00h	52.3.1/1309
4006_6001	I2C Frequency Divider register (I2C0_F)	8	R/W	00h	52.3.2/1309
4006_6002	I2C Control Register 1 (I2C0_C1)	8	R/W	00h	52.3.3/1310
4006_6003	I2C Status register (I2C0_S)	8	R/W	80h	52.3.4/1312
4006_6004	I2C Data I/O register (I2C0_D)	8	R/W	00h	52.3.5/1314
4006_6005	I2C Control Register 2 (I2C0_C2)	8	R/W	00h	52.3.6/1314
4006_6006	I2C Programmable Input Glitch Filter Register (I2C0_FLT)	8	R/W	00h	52.3.7/1315
4006_6007	I2C Range Address register (I2C0_RA)	8	R/W	00h	52.3.8/1316
4006_6008	I2C SMBus Control and Status register (I2C0_SMB)	8	R/W	00h	52.3.9/1316
4006_6009	I2C Address Register 2 (I2C0_A2)	8	R/W	C2h	52.3.10/ 1318
4006_600A	I2C SCL Low Timeout Register High (I2C0_SLTH)	8	R/W	00h	52.3.11/ 1318
4006_600B	I2C SCL Low Timeout Register Low (I2C0_SLTL)	8	R/W	00h	52.3.12/ 1319
4006_7000	I2C Address Register 1 (I2C1_A1)	8	R/W	00h	52.3.1/1309
4006_7001	I2C Frequency Divider register (I2C1_F)	8	R/W	00h	52.3.2/1309
4006_7002	I2C Control Register 1 (I2C1_C1)	8	R/W	00h	52.3.3/1310
4006_7003	I2C Status register (I2C1_S)	8	R/W	80h	52.3.4/1312
4006_7004	I2C Data I/O register (I2C1_D)	8	R/W	00h	52.3.5/1314
4006_7005	I2C Control Register 2 (I2C1_C2)	8	R/W	00h	52.3.6/1314
4006_7006	I2C Programmable Input Glitch Filter Register (I2C1_FLT)	8	R/W	00h	52.3.7/1315
4006_7007	I2C Range Address register (I2C1_RA)	8	R/W	00h	52.3.8/1316
4006_7008	I2C SMBus Control and Status register (I2C1_SMB)	8	R/W	00h	52.3.9/1316
4006_7009	I2C Address Register 2 (I2C1_A2)	8	R/W	C2h	52.3.10/ 1318
4006_700A	I2C SCL Low Timeout Register High (I2C1_SLTH)	8	R/W	00h	52.3.11/ 1318
4006_700B	I2C SCL Low Timeout Register Low (I2C1_SLTL)	8	R/W	00h	52.3.12/ 1319

52.3.1 I2C Address Register 1 (I2Cx_A1)

This register contains the slave address to be used by the I2C module.

Address: Base address + 0h offset

Bit	7	6	5	4	3	2	1	0
Read	AD[7:1]							0
Write								
Reset	0	0	0	0	0	0	0	0

I2Cx_A1 field descriptions

Field	Description
7–1 AD[7:1]	Address Contains the primary slave address used by the I2C module when it is addressed as a slave. This field is used in the 7-bit address scheme and the lower seven bits in the 10-bit address scheme.
0 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

52.3.2 I2C Frequency Divider register (I2Cx_F)

Address: Base address + 1h offset

Bit	7	6	5	4	3	2	1	0
Read Write	MULT		ICR					
Reset	0	0	0	0	0	0	0	0

I2Cx_F field descriptions

Field	Description
7–6 MULT	Multiplier Factor Defines the multiplier factor (mul). This factor is used along with the SCL divider to generate the I2C baud rate. 00 mul = 1 01 mul = 2 10 mul = 4 11 Reserved
ICR	ClockRate Prescales the I2C module clock for bit rate selection. This field and the MULT field determine the I2C baud rate, the SDA hold time, the SCL start hold time, and the SCL stop hold time. For a list of values corresponding to each ICR setting, see I2C divider and hold values. The SCL divider multiplied by multiplier factor (mul) determines the I2C baud rate. `I2C baud rate = I2C module clock speed (Hz)/(mul × SCL divider)`

Table continues on the next page...

I2Cx_F field descriptions (continued)

Field	Description

	The SDA hold time is the delay from the falling edge of SCL (I2C clock) to the changing of SDA (I2C data).
	`SDA hold time = I2C module clock period (s) × mul × SDA hold value`
	The SCL start hold time is the delay from the falling edge of SDA (I2C data) while SCL is high (start condition) to the falling edge of SCL (I2C clock).
	`SCL start hold time = I2C module clock period (s) × mul × SCL start hold value`
	The SCL stop hold time is the delay from the rising edge of SCL (I2C clock) to the rising edge of SDA (I2C data) while SCL is high (stop condition).
	`SCL stop hold time = I2C module clock period (s) × mul × SCL stop hold value`
	For example, if the I2C module clock speed is 8 MHz, the following table shows the possible hold time values with different ICR and MULT selections to achieve an I^2C baud rate of 100 kbit/s.

MULT	ICR	Hold times (μs)		
		SDA	SCL Start	SCL Stop
2h	00h	3.500	3.000	5.500
1h	07h	2.500	4.000	5.250
1h	0Bh	2.250	4.000	5.250
0h	14h	2.125	4.250	5.125
0h	18h	1.125	4.750	5.125

52.3.3 I2C Control Register 1 (I2Cx_C1)

Address: Base address + 2h offset

Bit	7	6	5	4	3	2	1	0
Read	IICEN	IICIE	MST	TX	TXAK	0	WUEN	DMAEN
Write						RSTA		
Reset	0	0	0	0	0	0	0	0

I2Cx_C1 field descriptions

Field	Description
7 IICEN	I2C Enable Enables I2C module operation. 0 Disabled 1 Enabled
6 IICIE	I2C Interrupt Enable Enables I2C interrupt requests.

Table continues on the next page...

I2Cx_C1 field descriptions (continued)

Field	Description
	0 Disabled 1 Enabled
5 MST	Master Mode Select When MST is changed from 0 to 1, a START signal is generated on the bus and master mode is selected. When this bit changes from 1 to 0, a STOP signal is generated and the mode of operation changes from master to slave. 0 Slave mode 1 Master mode
4 TX	Transmit Mode Select Selects the direction of master and slave transfers. In master mode this bit must be set according to the type of transfer required. Therefore, for address cycles, this bit is always set. When addressed as a slave this bit must be set by software according to the SRW bit in the status register. 0 Receive 1 Transmit
3 TXAK	Transmit Acknowledge Enable Specifies the value driven onto the SDA during data acknowledge cycles for both master and slave receivers. The value of SMB[FACK] affects NACK/ACK generation. **NOTE:** SCL is held low until TXAK is written. 0 An acknowledge signal is sent to the bus on the following receiving byte (if FACK is cleared) or the current receiving byte (if FACK is set). 1 No acknowledge signal is sent to the bus on the following receiving data byte (if FACK is cleared) or the current receiving data byte (if FACK is set).
2 RSTA	Repeat START Writing 1 to this bit generates a repeated START condition provided it is the current master. This bit will always be read as 0. Attempting a repeat at the wrong time results in loss of arbitration.
1 WUEN	Wakeup Enable The I2C module can wake the MCU from low power mode with no peripheral bus running when slave address matching occurs. 0 Normal operation. No interrupt generated when address matching in low power mode. 1 Enables the wakeup function in low power mode.
0 DMAEN	DMA Enable Enables or disables the DMA function. 0 All DMA signalling disabled. 1 DMA transfer is enabled. While SMB[FACK] = 0, the following conditions trigger the DMA request: • a data byte is received, and either address or data is transmitted. (ACK/NACK is automatic) • the first byte received matches the A1 register or is a general call address.

Table continues on the next page...

I2Cx_C1 field descriptions (continued)

Field	Description
	If any address matching occurs, S[IAAS] and S[TCF] are set. If the direction of transfer is known from master to slave, then it is not required to check S[SRW]. With this assumption, DMA can also be used in this case. In other cases, if the master reads data from the slave, then it is required to rewrite the C1 register operation. With this assumption, DMA cannot be used. When FACK = 1, an address or a data byte is transmitted.

52.3.4 I2C Status register (I2Cx_S)

Address: Base address + 3h offset

Bit	7	6	5	4	3	2	1	0
Read	TCF	IAAS	BUSY	ARBL	RAM	SRW	IICIF	RXAK
Write				w1c			w1c	
Reset	1	0	0	0	0	0	0	0

I2Cx_S field descriptions

Field	Description
7 TCF	Transfer Complete Flag Acknowledges a byte transfer; TCF is set on the completion of a byte transfer. This bit is valid only during or immediately following a transfer to or from the I2C module. TCF is cleared by reading the I2C data register in receive mode or by writing to the I2C data register in transmit mode. 0 Transfer in progress 1 Transfer complete
6 IAAS	Addressed As A Slave This bit is set by one of the following conditions: • The calling address matches the programmed primary slave address in the A1 register, or matches the range address in the RA register (which must be set to a nonzero value and under the condition I2C_C2[RMEN] = 1). • C2[GCAEN] is set and a general call is received. • SMB[SIICAEN] is set and the calling address matches the second programmed slave address. • ALERTEN is set and an SMBus alert response address is received • RMEN is set and an address is received that is within the range between the values of the A1 and RA registers. IAAS sets before the ACK bit. The CPU must check the SRW bit and set TX/RX accordingly. Writing the C1 register with any value clears this bit. 0 Not addressed 1 Addressed as a slave
5 BUSY	Bus Busy Indicates the status of the bus regardless of slave or master mode. This bit is set when a START signal is detected and cleared when a STOP signal is detected.

Table continues on the next page...

I2Cx_S field descriptions (continued)

Field	Description
	0 Bus is idle 1 Bus is busy
4 ARBL	Arbitration Lost This bit is set by hardware when the arbitration procedure is lost. The ARBL bit must be cleared by software, by writing 1 to it. 0 Standard bus operation. 1 Loss of arbitration.
3 RAM	Range Address Match This bit is set to 1 by any of the following conditions, if I2C_C2[RMEN] = 1: • Any nonzero calling address is received that matches the address in the RA register. • The calling address is within the range of values of the A1 and RA registers. **NOTE:** For the RAM bit to be set to 1 correctly, C1[IICIE] must be set to 1. Writing the C1 register with any value clears this bit to 0. 0 Not addressed 1 Addressed as a slave
2 SRW	Slave Read/Write When addressed as a slave, SRW indicates the value of the $R/\overline{W}$ command bit of the calling address sent to the master. 0 Slave receive, master writing to slave 1 Slave transmit, master reading from slave
1 IICIF	Interrupt Flag This bit sets when an interrupt is pending. This bit must be cleared by software by writing 1 to it, such as in the interrupt routine. One of the following events can set this bit: • One byte transfer, including ACK/NACK bit, completes if FACK is 0. An ACK or NACK is sent on the bus by writing 0 or 1 to TXAK after this bit is set in receive mode. • One byte transfer, excluding ACK/NACK bit, completes if FACK is 1. • Match of slave address to calling address including primary slave address, range slave address, alert response address, second slave address, or general call address. • Arbitration lost • In SMBus mode, any timeouts except SCL and SDA high timeouts 0 No interrupt pending 1 Interrupt pending
0 RXAK	Receive Acknowledge 0 Acknowledge signal was received after the completion of one byte of data transmission on the bus 1 No acknowledge signal detected

52.3.5 I2C Data I/O register (I2Cx_D)

Address: Base address + 4h offset

Bit	7	6	5	4	3	2	1	0
Read Write	DATA							
Reset	0	0	0	0	0	0	0	0

I2Cx_D field descriptions

Field	Description
DATA	Data In master transmit mode, when data is written to this register, a data transfer is initiated. The most significant bit is sent first. In master receive mode, reading this register initiates receiving of the next byte of data. **NOTE:** When making the transition out of master receive mode, switch the I2C mode before reading the Data register to prevent an inadvertent initiation of a master receive data transfer. In slave mode, the same functions are available after an address match occurs. The C1[TX] bit must correctly reflect the desired direction of transfer in master and slave modes for the transmission to begin. For example, if the I2C module is configured for master transmit but a master receive is desired, reading the Data register does not initiate the receive. Reading the Data register returns the last byte received while the I2C module is configured in master receive or slave receive mode. The Data register does not reflect every byte that is transmitted on the I2C bus, and neither can software verify that a byte has been written to the Data register correctly by reading it back. In master transmit mode, the first byte of data written to the Data register following assertion of MST (start bit) or assertion of RSTA (repeated start bit) is used for the address transfer and must consist of the calling address (in bits 7-1) concatenated with the required R/$\overline{W}$ bit (in position bit 0).

52.3.6 I2C Control Register 2 (I2Cx_C2)

Address: Base address + 5h offset

Bit	7	6	5	4	3	2	1	0
Read Write	GCAEN	ADEXT	HDRS	SBRC	RMEN	AD[10:8]		
Reset	0	0	0	0	0	0	0	0

I2Cx_C2 field descriptions

Field	Description
7 GCAEN	General Call Address Enable Enables general call address. 0 Disabled 1 Enabled

Table continues on the next page...

I2Cx_C2 field descriptions (continued)

Field	Description
6 ADEXT	Address Extension Controls the number of bits used for the slave address. 0 7-bit address scheme 1 10-bit address scheme
5 HDRS	High Drive Select Controls the drive capability of the I2C pads. 0 Normal drive mode 1 High drive mode
4 SBRC	Slave Baud Rate Control Enables independent slave mode baud rate at maximum frequency, which forces clock stretching on SCL in very fast I2C modes. To a slave, an example of a "very fast" mode is when the master transfers at 40 kbit/s but the slave can capture the master's data at only 10 kbit/s. 0 The slave baud rate follows the master baud rate and clock stretching may occur 1 Slave baud rate is independent of the master baud rate
3 RMEN	Range Address Matching Enable This bit controls the slave address matching for addresses between the values of the A1 and RA registers. When this bit is set, a slave address matching occurs for any address greater than the value of the A1 register and less than or equal to the value of the RA register. 0 Range mode disabled. No address matching occurs for an address within the range of values of the A1 and RA registers. 1 Range mode enabled. Address matching occurs when a slave receives an address within the range of values of the A1 and RA registers.
AD[10:8]	Slave Address Contains the upper three bits of the slave address in the 10-bit address scheme. This field is valid only while the ADEXT bit is set.

52.3.7 I2C Programmable Input Glitch Filter Register (I2Cx_FLT)

Address: Base address + 6h offset

Bit	7	6	5	4	3	2	1	0
Read	Reserved	0		FLT				
Write								
Reset	0	0	0	0	0	0	0	0

I2Cx_FLT field descriptions

Field	Description
7 Reserved	This field is reserved. Writing this bit has no effect.

Table continues on the next page...

I2Cx_FLT field descriptions (continued)

Field	Description
6–5 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
FLT	I2C Programmable Filter Factor Controls the width of the glitch, in terms of I2C module clock cycles, that the filter must absorb. For any glitch whose size is less than or equal to this width setting, the filter does not allow the glitch to pass. 00h No filter/bypass 01-1Fh Filter glitches up to width of *n* I2C module clock cycles, where *n*=1-31d

52.3.8 I2C Range Address register (I2Cx_RA)

Address: Base address + 7h offset

Bit	7	6	5	4	3	2	1	0
Read	RAD							0
Write								▨
Reset	0	0	0	0	0	0	0	0

I2Cx_RA field descriptions

Field	Description
7–1 RAD	Range Slave Address This field contains the slave address to be used by the I2C module. The field is used in the 7-bit address scheme. If I2C_C2[RMEN] is set to 1, any nonzero value write enables this register. This register value can be considered as a maximum boundary in the range matching mode.
0 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

52.3.9 I2C SMBus Control and Status register (I2Cx_SMB)

NOTE

When the SCL and SDA signals are held high for a length of time greater than the high timeout period, the SHTF1 flag sets. Before reaching this threshold, while the system is detecting how long these signals are being held high, a master assumes that the bus is free. However, the SHTF1 bit is set to 1 in the bus transmission process with the idle bus state.

NOTE

When the TCKSEL bit is set, there is no need to monitor the SHTF1 bit because the bus speed is too high to match the protocol of SMBus.

Address: Base address + 8h offset

Bit	7	6	5	4	3	2	1	0
Read	FACK	ALERTEN	SIICAEN	TCKSEL	SLTF	SHTF1	SHTF2	SHTF2IE
Write					w1c		w1c	
Reset	0	0	0	0	0	0	0	0

I2Cx_SMB field descriptions

Field	Description
7 FACK	Fast NACK/ACK Enable For SMBus packet error checking, the CPU must be able to issue an ACK or NACK according to the result of receiving data byte. 0 An ACK or NACK is sent on the following receiving data byte 1 Writing 0 to TXAK after receiving a data byte generates an ACK. Writing 1 to TXAK after receiving a data byte generates a NACK.
6 ALERTEN	SMBus Alert Response Address Enable Enables or disables SMBus alert response address matching. **NOTE:** After the host responds to a device that used the alert response address, you must use software to put the device's address on the bus. The alert protocol is described in the SMBus specification. 0 SMBus alert response address matching is disabled 1 SMBus alert response address matching is enabled
5 SIICAEN	Second I2C Address Enable Enables or disables SMBus device default address. 0 I2C address register 2 matching is disabled 1 I2C address register 2 matching is enabled
4 TCKSEL	Timeout Counter Clock Select Selects the clock source of the timeout counter. 0 Timeout counter counts at the frequency of the I2C module clock / 64 1 Timeout counter counts at the frequency of the I2C module clock
3 SLTF	SCL Low Timeout Flag This bit is set when the SLT register (consisting of the SLTH and SLTL registers) is loaded with a non-zero value (LoValue) and an SCL low timeout occurs. Software clears this bit by writing a logic 1 to it. **NOTE:** The low timeout function is disabled when the SLT register's value is 0. 0 No low timeout occurs 1 Low timeout occurs
2 SHTF1	SCL High Timeout Flag 1 This read-only bit sets when SCL and SDA are held high more than clock × LoValue / 512, which indicates the bus is free. This bit is cleared automatically. 0 No SCL high and SDA high timeout occurs 1 SCL high and SDA high timeout occurs

Table continues on the next page...

I2Cx_SMB field descriptions (continued)

Field	Description
1 SHTF2	SCL High Timeout Flag 2 This bit sets when SCL is held high and SDA is held low more than clock × LoValue / 512. Software clears this bit by writing 1 to it. 0 No SCL high and SDA low timeout occurs 1 SCL high and SDA low timeout occurs
0 SHTF2IE	SHTF2 Interrupt Enable Enables SCL high and SDA low timeout interrupt. 0 SHTF2 interrupt is disabled 1 SHTF2 interrupt is enabled

52.3.10 I2C Address Register 2 (I2Cx_A2)

Address: Base address + 9h offset

Bit	7	6	5	4	3	2	1	0
Read	SAD							0
Write								▨
Reset	1	1	0	0	0	0	1	0

I2Cx_A2 field descriptions

Field	Description
7–1 SAD	SMBus Address Contains the slave address used by the SMBus. This field is used on the device default address or other related addresses.
0 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

52.3.11 I2C SCL Low Timeout Register High (I2Cx_SLTH)

Address: Base address + Ah offset

Bit	7	6	5	4	3	2	1	0
Read Write	SSLT[15:8]							
Reset	0	0	0	0	0	0	0	0

I2Cx_SLTH field descriptions

Field	Description
SSLT[15:8]	SSLT[15:8] Most significant byte of SCL low timeout value that determines the timeout period of SCL low.

52.3.12 I2C SCL Low Timeout Register Low (I2Cx_SLTL)

Address: Base address + Bh offset

Bit	7	6	5	4	3	2	1	0
Read Write	SSLT[7:0]							
Reset	0	0	0	0	0	0	0	0

I2Cx_SLTL field descriptions

Field	Description
SSLT[7:0]	SSLT[7:0] Least significant byte of SCL low timeout value that determines the timeout period of SCL low.

52.4 Functional description

This section provides a comprehensive functional description of the I2C module.

52.4.1 I2C protocol

The I2C bus system uses a serial data line (SDA) and a serial clock line (SCL) for data transfers.

All devices connected to it must have open drain or open collector outputs. A logic AND function is exercised on both lines with external pull-up resistors. The value of these resistors depends on the system.

Normally, a standard instance of communication is composed of four parts:

1. START signal
2. Slave address transmission
3. Data transfer
4. STOP signal

The STOP signal should not be confused with the CPU STOP instruction. The following figure illustrates I2C bus system communication.

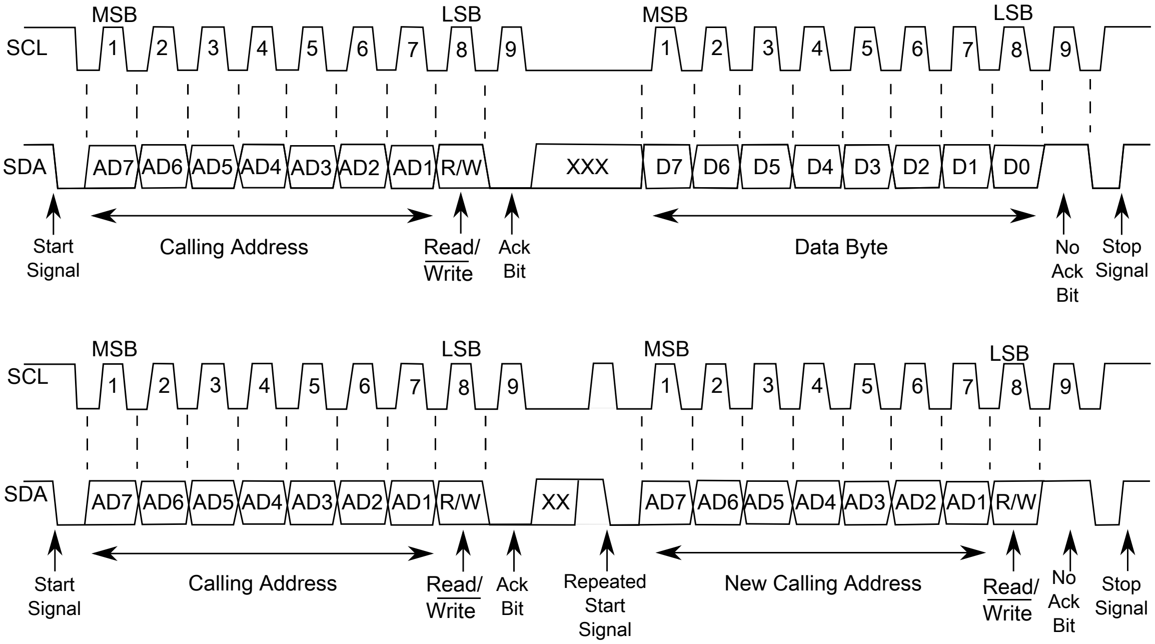


Figure 52-2. I2C bus transmission signals

52.4.1.1 START signal

The bus is free when no master device is engaging the bus (both SCL and SDA are high). When the bus is free, a master may initiate communication by sending a START signal. A START signal is defined as a high-to-low transition of SDA while SCL is high. This signal denotes the beginning of a new data transfer—each data transfer might contain several bytes of data—and brings all slaves out of their idle states.

52.4.1.2 Slave address transmission

Immediately after the START signal, the first byte of a data transfer is the slave address transmitted by the master. This address is a 7-bit calling address followed by an R/$\overline{\text{W}}$ bit. The R/$\overline{\text{W}}$ bit tells the slave the desired direction of data transfer.

- 1 = Read transfer: The slave transmits data to the master
- 0 = Write transfer: The master transmits data to the slave

Only the slave with a calling address that matches the one transmitted by the master responds by sending an acknowledge bit. The slave sends the acknowledge bit by pulling SDA low at the ninth clock.

No two slaves in the system can have the same address. If the I2C module is the master, it must not transmit an address that is equal to its own slave address. The I2C module cannot be master and slave at the same time. However, if arbitration is lost during an address cycle, the I2C module reverts to slave mode and operates correctly even if it is being addressed by another master.

52.4.1.3 Data transfers

When successful slave addressing is achieved, data transfer can proceed on a byte-by-byte basis in the direction specified by the R/$\overline{\text{W}}$ bit sent by the calling master.

All transfers that follow an address cycle are referred to as data transfers, even if they carry subaddress information for the slave device.

Each data byte is 8 bits long. Data may be changed only while SCL is low. Data must be held stable while SCL is high. There is one clock pulse on SCL for each data bit, and the MSB is transferred first. Each data byte is followed by a ninth (acknowledge) bit, which is signaled from the receiving device by pulling SDA low at the ninth clock. In summary, one complete data transfer needs nine clock pulses.

If the slave receiver does not acknowledge the master in the ninth bit, the slave must leave SDA high. The master interprets the failed acknowledgement as an unsuccessful data transfer.

If the master receiver does not acknowledge the slave transmitter after a data byte transmission, the slave interprets it as an end to data transfer and releases the SDA line.

In the case of a failed acknowledgement by either the slave or master, the data transfer is aborted and the master does one of two things:

- Relinquishes the bus by generating a STOP signal.
- Commences a new call by generating a repeated START signal.

52.4.1.4 STOP signal

The master can terminate the communication by generating a STOP signal to free the bus. A STOP signal is defined as a low-to-high transition of SDA while SCL is asserted.

The master can generate a STOP signal even if the slave has generated an acknowledgement, at which point the slave must release the bus.

52.4.1.5 Repeated START signal

The master may generate a START signal followed by a calling command without generating a STOP signal first. This action is called a repeated START. The master uses a repeated START to communicate with another slave or with the same slave in a different mode (transmit/receive mode) without releasing the bus.

52.4.1.6 Arbitration procedure

The I2C bus is a true multimaster bus that allows more than one master to be connected on it.

If two or more masters try to control the bus at the same time, a clock synchronization procedure determines the bus clock. The bus clock's low period is equal to the longest clock low period, and the high period is equal to the shortest one among the masters.

The relative priority of the contending masters is determined by a data arbitration procedure. A bus master loses arbitration if it transmits logic level 1 while another master transmits logic level 0. The losing masters immediately switch to slave receive mode and stop driving SDA output. In this case, the transition from master to slave mode does not generate a STOP condition. Meanwhile, hardware sets a status bit to indicate the loss of arbitration.

52.4.1.7 Clock synchronization

Because wire AND logic is performed on SCL, a high-to-low transition on SCL affects all devices connected on the bus. The devices start counting their low period and, after a device's clock has gone low, that device holds SCL low until the clock reaches its high state. However, the change of low to high in this device clock might not change the state of SCL if another device clock is still within its low period. Therefore, the synchronized clock SCL is held low by the device with the longest low period. Devices with shorter low periods enter a high wait state during this time; see the following diagram. When all applicable devices have counted off their low period, the synchronized clock SCL is released and pulled high. Afterward there is no difference between the device clocks and the state of SCL, and all devices start counting their high periods. The first device to complete its high period pulls SCL low again.

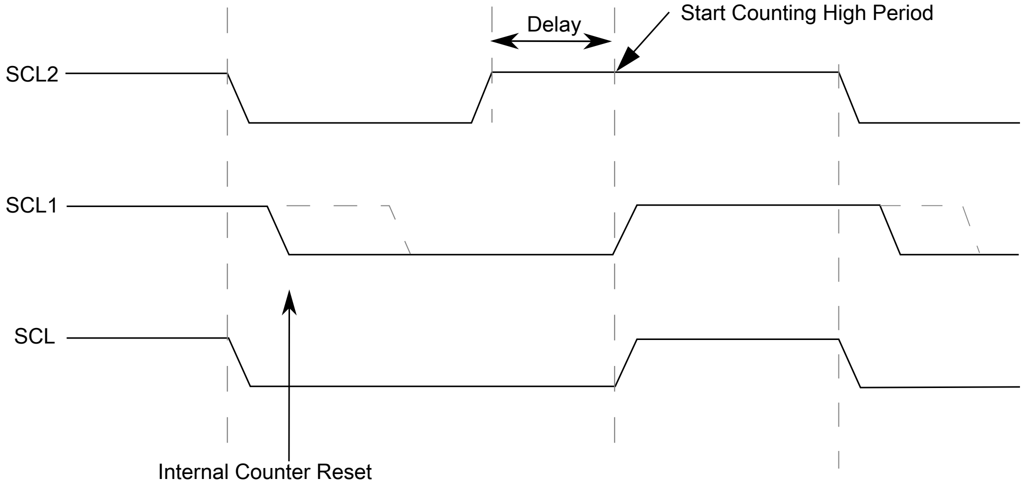


Figure 52-3. I2C clock synchronization

52.4.1.8 Handshaking

The clock synchronization mechanism can be used as a handshake in data transfers. A slave device may hold SCL low after completing a single byte transfer (9 bits). In this case, it halts the bus clock and forces the master clock into wait states until the slave releases SCL.

52.4.1.9 Clock stretching

The clock synchronization mechanism can be used by slaves to slow down the bit rate of a transfer. After the master drives SCL low, a slave can drive SCL low for the required period and then release it. If the slave's SCL low period is greater than the master's SCL low period, the resulting SCL bus signal's low period is stretched. In other words, the SCL bus signal's low period is increased to be the same length as the slave's SCL low period.

52.4.1.10 I2C divider and hold values

NOTE

For some cases on some devices, the SCL divider value may vary by ±2 or ±4 when ICR's value ranges from 00h to 0Fh. These potentially varying SCL divider values are highlighted in the following table. For the actual SCL divider values for your device, see the chip-specific details about the I2C module.

Table 52-2. I2C divider and hold values

ICR (hex)	SCL divider	SDA hold value	SCL hold (start) value	SCL hold (stop) value		ICR (hex)	SCL divider (clocks)	SDA hold (clocks)	SCL hold (start) value	SCL hold (stop) value
00	20	7	6	11		20	160	17	78	81
01	22	7	7	12		21	192	17	94	97
02	24	8	8	13		22	224	33	110	113
03	26	8	9	14		23	256	33	126	129
04	28	9	10	15		24	288	49	142	145
05	30	9	11	16		25	320	49	158	161
06	34	10	13	18		26	384	65	190	193
07	40	10	16	21		27	480	65	238	241
08	28	7	10	15		28	320	33	158	161
09	32	7	12	17		29	384	33	190	193
0A	36	9	14	19		2A	448	65	222	225
0B	40	9	16	21		2B	512	65	254	257
0C	44	11	18	23		2C	576	97	286	289
0D	48	11	20	25		2D	640	97	318	321
0E	56	13	24	29		2E	768	129	382	385
0F	68	13	30	35		2F	960	129	478	481
10	48	9	18	25		30	640	65	318	321
11	56	9	22	29		31	768	65	382	385
12	64	13	26	33		32	896	129	446	449
13	72	13	30	37		33	1024	129	510	513
14	80	17	34	41		34	1152	193	574	577
15	88	17	38	45		35	1280	193	638	641
16	104	21	46	53		36	1536	257	766	769
17	128	21	58	65		37	1920	257	958	961
18	80	9	38	41		38	1280	129	638	641
19	96	9	46	49		39	1536	129	766	769
1A	112	17	54	57		3A	1792	257	894	897
1B	128	17	62	65		3B	2048	257	1022	1025
1C	144	25	70	73		3C	2304	385	1150	1153
1D	160	25	78	81		3D	2560	385	1278	1281
1E	192	33	94	97		3E	3072	513	1534	1537
1F	240	33	118	121		3F	3840	513	1918	1921

52.4.2 10-bit address

For 10-bit addressing, 0x11110 is used for the first 5 bits of the first address byte. Various combinations of read/write formats are possible within a transfer that includes 10-bit addressing.

52.4.2.1 Master-transmitter addresses a slave-receiver

The transfer direction is not changed. When a 10-bit address follows a START condition, each slave compares the first 7 bits of the first byte of the slave address (11110XX) with its own address and tests whether the eighth bit ($R/\overline{W}$ direction bit) is 0. It is possible that more than one device finds a match and generates an acknowledge (A1). Each slave that finds a match compares the 8 bits of the second byte of the slave address with its own address, but only one slave finds a match and generates an acknowledge (A2). The matching slave remains addressed by the master until it receives a STOP condition (P) or a repeated START condition (Sr) followed by a different slave address.

Table 52-3. Master-transmitter addresses slave-receiver with a 10-bit address

S	Slave address first 7 bits 11110 + AD10 + AD9	$R/\overline{W}$ 0	A1	Slave address second byte AD[8:1]	A2	Data	A	...	Data	A/A	P

After the master-transmitter has sent the first byte of the 10-bit address, the slave-receiver sees an I2C interrupt. User software must ensure that for this interrupt, the contents of the Data register are ignored and not treated as valid data.

52.4.2.2 Master-receiver addresses a slave-transmitter

The transfer direction is changed after the second $R/\overline{W}$ bit. Up to and including acknowledge bit A2, the procedure is the same as that described for a master-transmitter addressing a slave-receiver. After the repeated START condition (Sr), a matching slave remembers that it was addressed before. This slave then checks whether the first seven bits of the first byte of the slave address following Sr are the same as they were after the START condition (S), and it tests whether the eighth ($R/\overline{W}$) bit is 1. If there is a match, the slave considers that it has been addressed as a transmitter and generates acknowledge A3. The slave-transmitter remains addressed until it receives a STOP condition (P) or a repeated START condition (Sr) followed by a different slave address.

After a repeated START condition (Sr), all other slave devices also compare the first seven bits of the first byte of the slave address with their own addresses and test the eighth ($R/\overline{W}$) bit. However, none of them are addressed because $R/\overline{W}$ = 1 (for 10-bit devices), or the 11110XX slave address (for 7-bit devices) does not match.

Table 52-4. Master-receiver addresses a slave-transmitter with a 10-bit address

S	Slave address first 7 bits 11110 + AD10 + AD9	$R/\overline{W}$ 0	A1	Slave address second byte AD[8:1]	A2	Sr	Slave address first 7 bits 11110 + AD10 + AD9	$R/\overline{W}$ 1	A3	Data	A	...	Data	A	P

After the master-receiver has sent the first byte of the 10-bit address, the slave-transmitter sees an I2C interrupt. User software must ensure that for this interrupt, the contents of the Data register are ignored and not treated as valid data.

52.4.3 Address matching

All received addresses can be requested in 7-bit or 10-bit address format.

- AD[7:1] in Address Register 1, which contains the I2C primary slave address, always participates in the address matching process. It provides a 7-bit address.
- If the ADEXT bit is set, AD[10:8] in Control Register 2 participates in the address matching process. It extends the I2C primary slave address to a 10-bit address.

Additional conditions that affect address matching include:

- If the GCAEN bit is set, general call participates the address matching process.
- If the ALERTEN bit is set, alert response participates the address matching process.
- If the SIICAEN bit is set, Address Register 2 participates in the address matching process.
- If the RMEN bit is set, when the Range Address register is programmed to a nonzero value, any address within the range of values of Address Register 1 (excluded) and the Range Address register (included) participates in the address matching process. The Range Address register must be programmed to a value greater than the value of Address Register 1.

When the I2C module responds to one of these addresses, it acts as a slave-receiver and the IAAS bit is set after the address cycle. Software must read the Data register after the first byte transfer to determine that the address is matched.

52.4.4 System management bus specification

SMBus provides a control bus for system and power management related tasks. A system can use SMBus to pass messages to and from devices instead of tripping individual control lines.

Removing the individual control lines reduces pin count. Accepting messages ensures future expandability. With the system management bus, a device can provide manufacturer information, tell the system what its model/part number is, save its state for a suspend event, report different types of errors, accept control parameters, and return its status.

52.4.4.1 Timeouts

The $T_{TIMEOUT,MIN}$ parameter allows a master or slave to conclude that a defective device is holding the clock low indefinitely or a master is intentionally trying to drive devices off the bus. The slave device must release the bus (stop driving the bus and let SCL and SDA float high) when it detects any single clock held low longer than $T_{TIMEOUT,MIN}$. Devices that have detected this condition must reset their communication and be able to receive a new START condition within the timeframe of $T_{TIMEOUT,MAX}$.

SMBus defines a clock low timeout, $T_{TIMEOUT}$, of 35 ms, specifies $T_{LOW:SEXT}$ as the cumulative clock low extend time for a slave device, and specifies $T_{LOW:MEXT}$ as the cumulative clock low extend time for a master device.

52.4.4.1.1 SCL low timeout

If the SCL line is held low by a slave device on the bus, no further communication is possible. Furthermore, the master cannot force the SCL line high to correct the error condition. To solve this problem, the SMBus protocol specifies that devices participating in a transfer must detect any clock cycle held low longer than a timeout value condition. Devices that have detected the timeout condition must reset the communication. When the I2C module is an active master, if it detects that SMBCLK low has exceeded the value of $T_{TIMEOUT,MIN}$, it must generate a stop condition within or after the current data byte in the transfer process. When the I2C module is a slave, if it detects the $T_{TIMEOUT,MIN}$ condition, it resets its communication and is then able to receive a new START condition.

52.4.4.1.2 SCL high timeout

When the I2C module has determined that the SMBCLK and SMBDAT signals have been high for at least $T_{HIGH:MAX}$, it assumes that the bus is idle.

A HIGH timeout occurs after a START condition appears on the bus but before a STOP condition appears on the bus. Any master detecting this scenario can assume the bus is free when either of the following occurs:

- SHTF1 rises.
- The BUSY bit is high and SHTF1 is high.

When the SMBDAT signal is low and the SMBCLK signal is high for a period of time, another kind of timeout occurs. The time period must be defined in software. SHTF2 is used as the flag when the time limit is reached. This flag is also an interrupt resource, so it triggers IICIF.

52.4.4.1.3 CSMBCLK TIMEOUT MEXT and CSMBCLK TIMEOUT SEXT

The following figure illustrates the definition of the timeout intervals $T_{LOW:SEXT}$ and $T_{LOW:MEXT}$. When in master mode, the I2C module must not cumulatively extend its clock cycles for a period greater than $T_{LOW:MEXT}$ within a byte, where each byte is defined as START-to-ACK, ACK-to-ACK, or ACK-to-STOP. When CSMBCLK TIMEOUT MEXT occurs, SMBus MEXT rises and also triggers the SLTF.

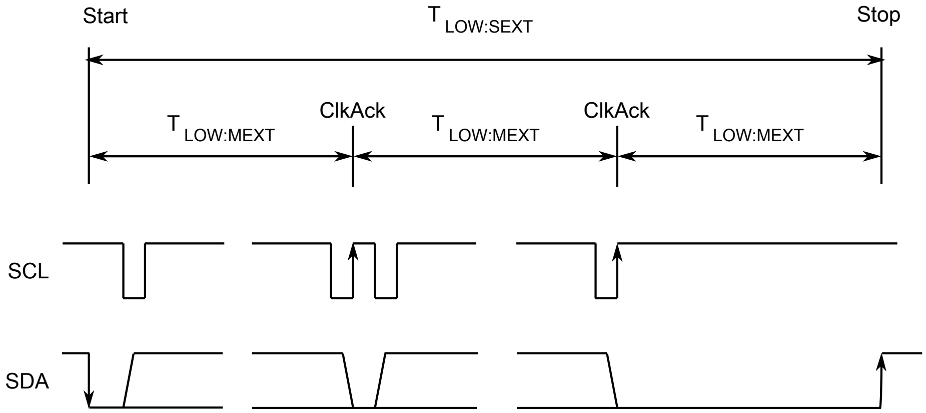


Figure 52-4. Timeout measurement intervals

A master is allowed to abort the transaction in progress to any slave that violates the $T_{LOW:SEXT}$ or $T_{TIMEOUT,MIN}$ specifications. To abort the transaction, the master issues a STOP condition at the conclusion of the byte transfer in progress. When a slave, the I2C module must not cumulatively extend its clock cycles for a period greater than $T_{LOW:SEXT}$ during any message from the initial START to the STOP. When CSMBCLK TIMEOUT SEXT occurs, SEXT rises and also triggers SLTF.

NOTE

CSMBCLK TIMEOUT SEXT and CSMBCLK TIMEOUT MEXT are optional functions that are implemented in the second step.

52.4.4.2 FAST ACK and NACK

To improve reliability and communication robustness, implementation of packet error checking (PEC) by SMBus devices is optional for SMBus devices but required for devices participating in and only during the address resolution protocol (ARP) process. The PEC is a CRC-8 error checking byte, calculated on all the message bytes. The PEC is appended to the message by the device that supplied the last data byte. If the PEC is present but not correct, a NACK is issued by the receiver. Otherwise an ACK is issued. To calculate the CRC-8 by software, this module can hold the SCL line low after receiving the eighth SCL (8th bit) if this byte is a data byte. So software can determine whether an ACK or NACK should be sent to the bus by setting or clearing the TXAK bit if the FACK (fast ACK/NACK enable) bit is enabled.

SMBus requires a device always to acknowledge its own address, as a mechanism to detect the presence of a removable device (such as a battery or docking station) on the bus. In addition to indicating a slave device busy condition, SMBus uses the NACK mechanism to indicate the reception of an invalid command or invalid data. Because such a condition may occur on the last byte of the transfer, SMBus devices are required to have the ability to generate the not acknowledge after the transfer of each byte and before the completion of the transaction. This requirement is important because SMBus does not provide any other resend signaling. This difference in the use of the NACK signaling has implications on the specific implementation of the SMBus port, especially in devices that handle critical system data such as the SMBus host and the SBS components.

NOTE

In the last byte of master receive slave transmit mode, the master must send a NACK to the bus, so FACK must be switched off before the last byte transmits.

52.4.5 Resets

The I2C module is disabled after a reset. The I2C module cannot cause a core reset.

52.4.6 Interrupts

The I2C module generates an interrupt when any of the events in the table found here occur, provided that the IICIE bit is set.

The interrupt is driven by the IICIF bit (of the I2C Status Register) and masked with the IICIE bit (of the I2C Control Register 1). The IICIF bit must be cleared (by software) by writing 1 to it in the interrupt routine. The SMBus timeouts interrupt is driven by SLTF and masked with the IICIE bit. The SLTF bit must be cleared by software by writing 1 to it in the interrupt routine. You can determine the interrupt type by reading the Status Register.

NOTE

In master receive mode, the FACK bit must be set to zero before the last byte transfer.

Table 52-5. Interrupt summary

Interrupt source	Status	Flag	Local enable
Complete 1-byte transfer	TCF	IICIF	IICIE
Match of received calling address	IAAS	IICIF	IICIE
Arbitration lost	ARBL	IICIF	IICIE
SMBus SCL low timeout	SLTF	IICIF	IICIE
SMBus SCL high SDA low timeout	SHTF2	IICIF	IICIE & SHTF2IE
Wakeup from stop or wait mode	IAAS	IICIF	IICIE & WUEN

52.4.6.1 Byte transfer interrupt

The Transfer Complete Flag (TCF) bit is set at the falling edge of the ninth clock to indicate the completion of a byte and acknowledgement transfer. When FACK is enabled, TCF is then set at the falling edge of eighth clock to indicate the completion of byte.

52.4.6.2 Address detect interrupt

When the calling address matches the programmed slave address (I2C Address Register) or when the GCAEN bit is set and a general call is received, the IAAS bit in the Status Register is set. The CPU is interrupted, provided the IICIE bit is set. The CPU must check the SRW bit and set its Tx mode accordingly.

52.4.6.3 Exit from low-power/stop modes

The slave receive input detect circuit and address matching feature are still active on low power modes (wait and stop). An asynchronous input matching slave address or general call address brings the CPU out of low power/stop mode if the interrupt is not masked. Therefore, TCF and IAAS both can trigger this interrupt.

52.4.6.4 Arbitration lost interrupt

The I2C is a true multimaster bus that allows more than one master to be connected on it. If two or more masters try to control the bus at the same time, the relative priority of the contending masters is determined by a data arbitration procedure. The I2C module asserts the arbitration-lost interrupt when it loses the data arbitration process and the ARBL bit in the Status Register is set.

Arbitration is lost in the following circumstances:

1. SDA is sampled as low when the master drives high during an address or data transmit cycle.
2. SDA is sampled as low when the master drives high during the acknowledge bit of a data receive cycle.
3. A START cycle is attempted when the bus is busy.
4. A repeated START cycle is requested in slave mode.
5. A STOP condition is detected when the master did not request it.

The ARBL bit must be cleared (by software) by writing 1 to it.

52.4.6.5 Timeout interrupt in SMBus

When the IICIE bit is set, the I2C module asserts a timeout interrupt (outputs SLTF and SHTF2) upon detection of any of the mentioned timeout conditions, with one exception. The SCL high and SDA high TIMEOUT mechanism must not be used to influence the timeout interrupt output, because this timeout indicates an idle condition on the bus. SHTF1 rises when it matches the SCL high and SDA high TIMEOUT and falls automatically just to indicate the bus status. The SHTF2's timeout period is the same as that of SHTF1, which is short compared to that of SLTF, so another control bit, SHTF2IE, is added to enable or disable it.

52.4.7 Programmable input glitch filter

An I2C glitch filter has been added outside legacy I2C modules but within the I2C package. This filter can absorb glitches on the I2C clock and data lines for the I2C module.

The width of the glitch to absorb can be specified in terms of the number of (half) I2C module clock cycles. A single Programmable Input Glitch Filter control register is provided. Effectively, any down-up-down or up-down-up transition on the data line that occurs within the number of clock cycles programmed in this register is ignored by the I2C module. The programmer must specify the size of the glitch (in terms of I2C module clock cycles) for the filter to absorb and not pass.

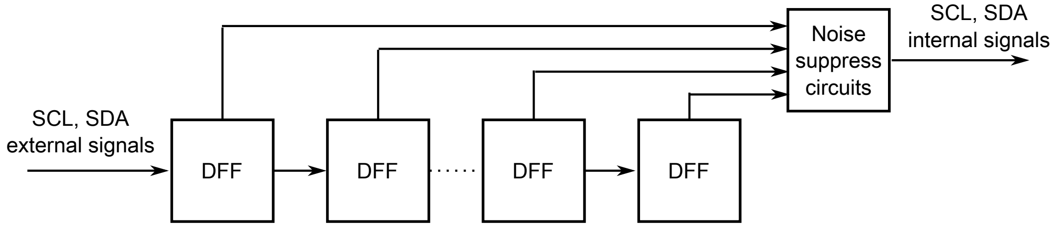


Figure 52-5. Programmable input glitch filter diagram

52.4.8 Address matching wake-up

When a primary, range, or general call address match occurs when the I2C module is in slave receive mode, the MCU wakes from a low power mode where no peripheral bus is running.

Data sent on the bus that is the same as a target device address might also wake the target MCU.

After the address matching IAAS bit is set, an interrupt is sent at the end of address matching to wake the core. The IAAS bit must be cleared after the clock recovery.

NOTE

After the system recovers and is in Run mode, restart the I2C module if it is needed to transfer packets. To avoid I2C transfer problems resulting from the situation, firmware should prevent the MCU execution of a STOP instruction when the I2C module is in the middle of a transfer.

52.4.9 DMA support

If the DMAEN bit is cleared and the IICIE bit is set, an interrupt condition generates an interrupt request.

If the DMAEN bit is set and the IICIE bit is set, an interrupt condition generates a DMA request instead. DMA requests are generated by the transfer complete flag (TCF).

If the DMAEN bit is set, only the TCF initiates a DMA request. All other events generate CPU interrupts.

NOTE

Before the last byte of master receive mode, TXAK must be set to send a NACK after the last byte's transfer. Therefore, the DMA must be disabled before the last byte's transfer.

NOTE

In 10-bit address mode transmission, the addresses to send occupy 2–3 bytes. During this transfer period, the DMA must be disabled because the C1 register is written to send a repeat start or to change the transfer direction.

52.5 Initialization/application information

Module Initialization (Slave)

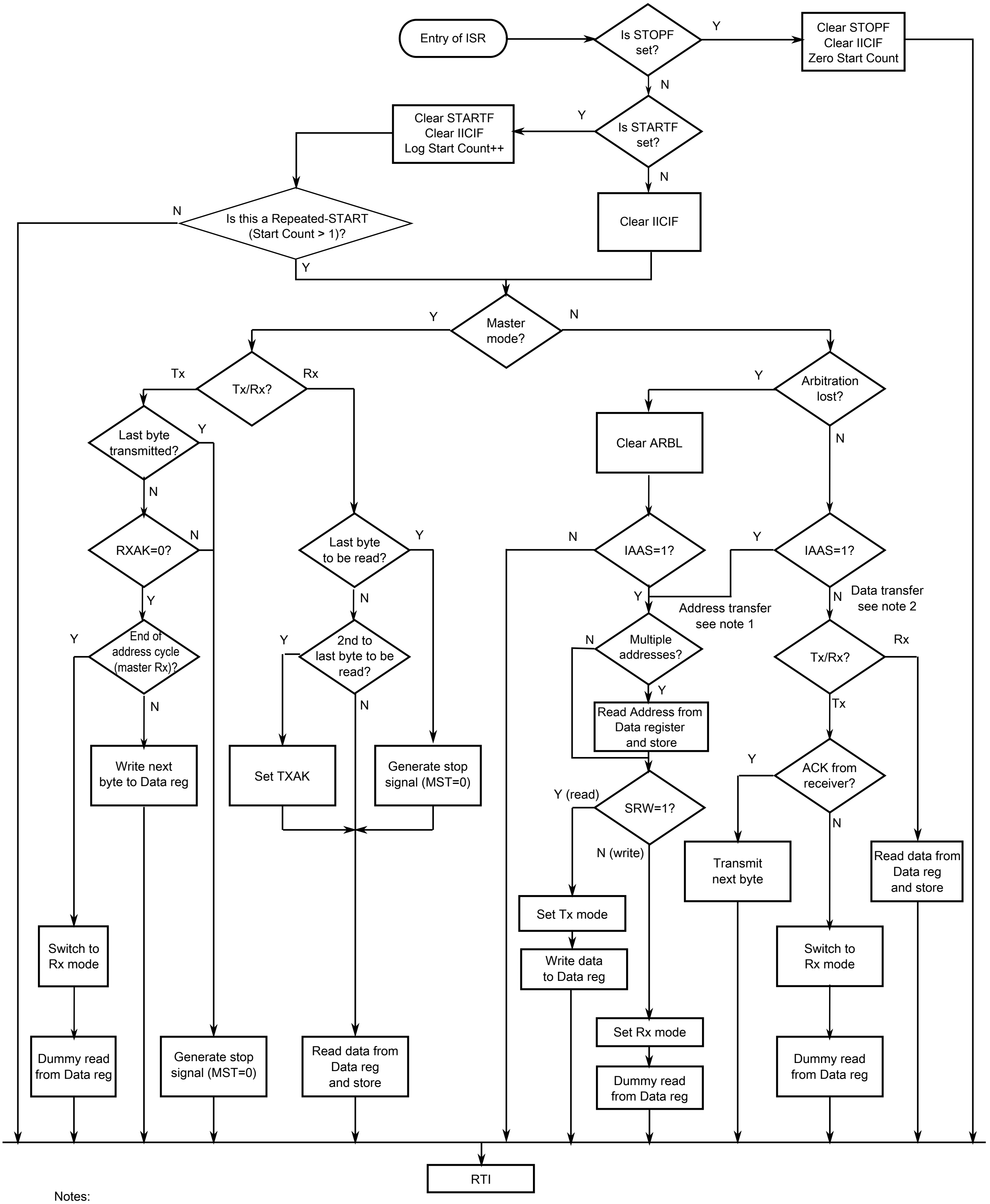
1. Write: Control Register 2
   - to enable or disable general call
   - to select 10-bit or 7-bit addressing mode
2. Write: Address Register 1 to set the slave address
3. Write: Control Register 1 to enable the I2C module and interrupts
4. Initialize RAM variables (IICEN = 1 and IICIE = 1) for transmit data
5. Initialize RAM variables used to achieve the routine shown in the following figure

Module Initialization (Master)

1. Write: Frequency Divider register to set the I2C baud rate (see example in description of ICR)
2. Write: Control Register 1 to enable the I2C module and interrupts
3. Initialize RAM variables (IICEN = 1 and IICIE = 1) for transmit data
4. Initialize RAM variables used to achieve the routine shown in the following figure
5. Write: Control Register 1 to enable TX

6. Write: Control Register 1 to enable MST (master mode)
7. Write: Data register with the address of the target slave (the LSB of this byte determines whether the communication is master receive or transmit)

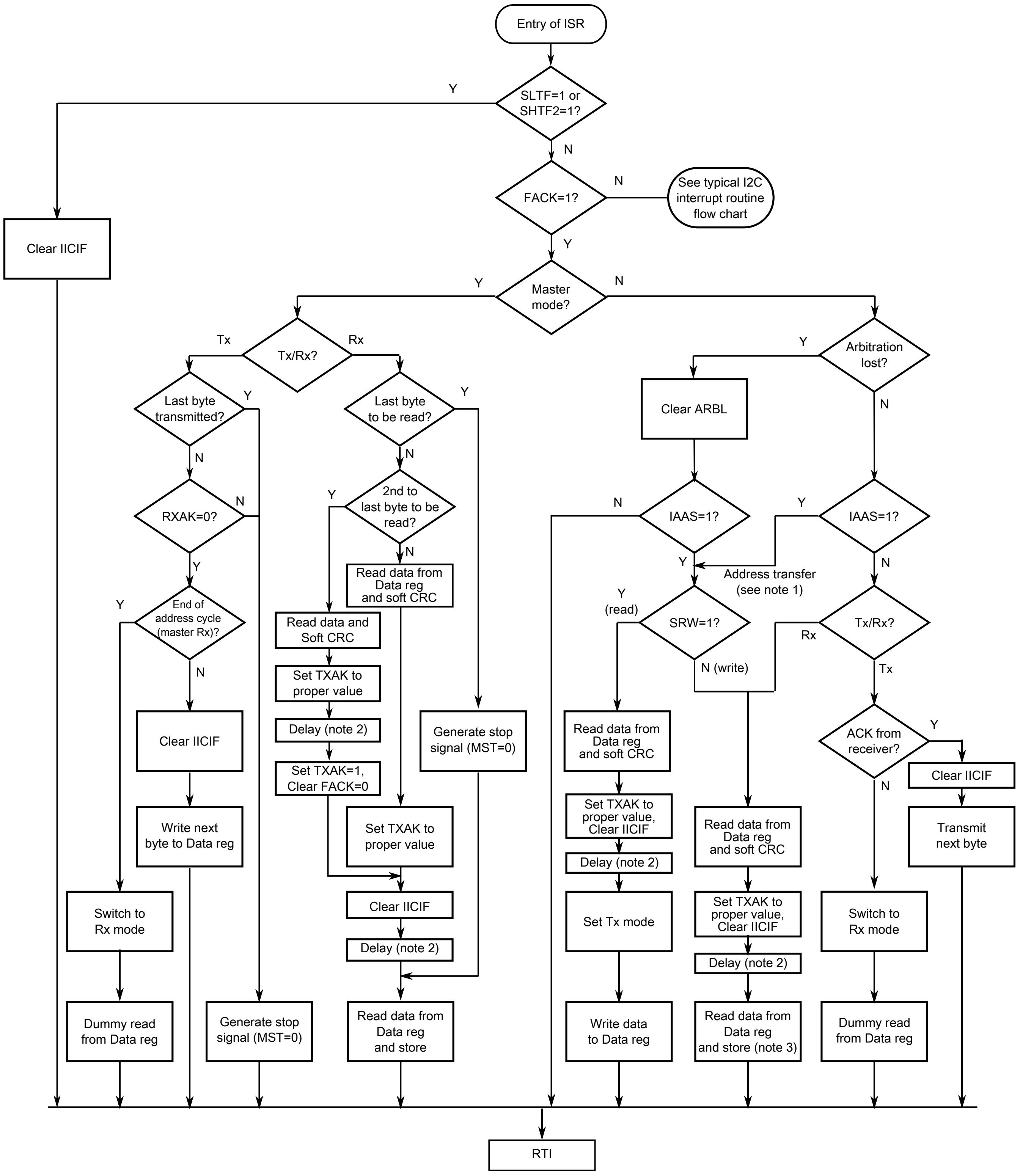
The routine shown in the following figure encompasses both master and slave I2C operations. For slave operation, an incoming I2C message that contains the proper address begins I2C communication. For master operation, communication must be initiated by writing the Data register. An example of an I2C driver which implements many of the steps described here is available in AN4342: Using the Inter-Integrated Circuit on ColdFire+ and Kinetis .



Notes:

1. If general call is enabled, check to determine if the received address is a general call address (0x00). If the received address is a general call address, the general call must be handled by user software.
2. When 10-bit addressing addresses a slave, the slave sees an interrupt following the first byte of the extended address. Ensure that for this interrupt, the contents of the Data register are ignored and not treated as a valid data transfer.

Figure 52-6. Typical I2C interrupt routine



Notes:

1. If general call or SIICAEN is enabled, check to determine if the received address is a general call address (0x00) or an SMBus device default address. In either case, they must be handled by user software.
2. In receive mode, one bit time delay may be needed before the first and second data reading, to wait for the possible longest time period (in worst case) of the 9th SCL cycle.
3. This read is a dummy read in order to reset the SMBus receiver state machine.

Figure 52-7. Typical I2C SMBus interrupt routine

Chapter 53
MCU: Universal Asynchronous Receiver/Transmitter (UART)

53.1 Introduction

NOTE

> For the chip-specific implementation details of this module's instances, see the chip configuration information.

The UART allows asynchronous serial communication with peripheral devices and CPUs.

53.1.1 Features

The UART includes the following features:

- Full-duplex operation
- Standard mark/space non-return-to-zero (NRZ) format
- Selectable IrDA 1.4 return-to-zero-inverted (RZI) format with programmable pulse width
- 13-bit baud rate selection with /32 fractional divide, based on the module clock frequency
- Programmable 8-bit or 9-bit data format
- Separately enabled transmitter and receiver
- Programmable transmitter output polarity
- Programmable receive input polarity
- Up to 14-bit break character transmission.

- 11-bit break character detection option
- Independent FIFO structure for transmit and receive
- Two receiver wakeup methods:
    - Idle line wakeup
    - Address mark wakeup
- Address match feature in the receiver to reduce address mark wakeup ISR overhead
- Ability to select MSB or LSB to be first bit on wire
- Hardware flow control support for request to send (RTS) and clear to send (CTS) signals
- Support for ISO 7816 protocol to interface with SIM cards and smart cards
    - Support for T=0 and T=1 protocols
    - Automatic retransmission of NACK'd packets with programmable retry threshold
    - Support for 11 and 12 ETU transfers
    - Detection of initial packet and automated transfer parameter programming
    - Interrupt-driven operation with seven ISO-7816 specific interrupts:
        - Wait time violated
        - Character wait time violated
        - Block wait time violated
        - Initial frame detected
        - Transmit error threshold exceeded
        - Receive error threshold exceeded
        - Guard time violated
- Interrupt-driven operation with flags, not specific to ISO-7816 support
    - Transmitter data buffer at or below watermark
    - Transmission complete
    - Receiver data buffer at or above watermark
    - Idle receiver input

  - Receiver data buffer overrun
  - Receiver data buffer underflow
  - Transmit data buffer overflow
  - Noise error
  - Framing error
  - Parity error
  - Active edge on receive pin
  - LIN break detect
- Receiver framing error detection
- Hardware parity generation and checking
- 1/16 bit-time noise detection
- DMA interface

53.1.2 Modes of operation

The UART functions in the same way in all the normal modes.

It has the following low power modes:

- Wait mode
- Stop mode

53.1.2.1 Run mode

This is the normal mode of operation.

53.1.2.2 Wait mode

UART operation in the Wait mode depends on the state of the C1[UARTSWAI] field.

- If C1[UARTSWAI] is cleared, and the CPU is in Wait mode, the UART operates normally.
- If C1[UARTSWAI] is set, and the CPU is in Wait mode, the UART clock generation ceases and the UART module enters a power conservation state.

C1[UARTSWAI] does not initiate any power down or power up procedures for the ISO-7816 smartcard interface.

Setting C1[UARTSWAI] does not affect the state of the C2[RE] or C2[TE].

If C1[UARTSWAI] is set, any ongoing transmission or reception stops at the Wait mode entry. The transmission or reception resumes when either an internal or external interrupt brings the CPU out of Wait mode. Bringing the CPU out of Wait mode by reset aborts any ongoing transmission or reception and resets the UART.

53.1.2.3 Stop mode

The UART is inactive during Stop mode for reduced power consumption. The STOP instruction does not affect the UART register states, but the UART module clock is disabled. The UART operation resumes after an external interrupt brings the CPU out of Stop mode. Bringing the CPU out of Stop mode by reset aborts any ongoing transmission or reception and resets the UART. Entering or leaving Stop mode does not initiate any power down or power up procedures for the ISO-7816 smartcard interface.

53.2 UART signal descriptions

The UART signals are shown in the following table.

Table 53-1. UART signal descriptions

Signal	Description	I/O
$\overline{\text{CTS}}$	Clear to send	I
$\overline{\text{RTS}}$	Request to send	O
RXD	Receive data	I
TXD	Transmit data	O

53.2.1 Detailed signal descriptions

The detailed signal descriptions of the UART are shown in the following table.

Table 53-2. UART—Detailed signal descriptions

Signal	I/O	Description	
$\overline{\text{CTS}}$	I	Clear to send. Indicates whether the UART can start transmitting data when flow control is enabled.	
		State meaning	Asserted—Data transmission can start. Negated—Data transmission cannot start.
		Timing	Assertion—When transmitting device's $\overline{\text{RTS}}$ asserts. Negation—When transmitting device's $\overline{\text{RTS}}$ deasserts.
$\overline{\text{RTS}}$	O	Request to send. When driven by the receiver, indicates whether the UART is ready to receive data. When driven by the transmitter, can enable an external transceiver during transmission.	
		State meaning	Asserted—When driven by the receiver, ready to receive data. When driven by the transmitter, enable the external transmitter. Negated—When driven by the receiver, not ready to receive data. When driven by the transmitter, disable the external transmitter.
		Timing	Assertion—Can occur at any time; can assert asynchronously to the other input signals. Negation—Can occur at any time; can deassert asynchronously to the other input signals.
RXD	I	Receive data. Serial data input to receiver.	
		State meaning	Whether RXD is interpreted as a 1 or 0 depends on the bit encoding method along with other configuration settings.
		Timing	Sampled at a frequency determined by the module clock divided by the baud rate.
TXD	O	Transmit data. Serial data output from transmitter.	
		State meaning	Whether TXD is interpreted as a 1 or 0 depends on the bit encoding method along with other configuration settings.
		Timing	Driven at the beginning or within a bit time according to the bit encoding method along with other configuration settings. Otherwise, transmissions are independent of reception timing.

53.3 Memory map and registers

This section provides a detailed description of all memory and registers.

Accessing reserved addresses within the memory map results in a transfer error. None of the contents of the implemented addresses are modified as a result of that access.

Only byte accesses are supported.

UART memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4006_A000	UART Baud Rate Registers: High (UART0_BDH)	8	R/W	00h	53.3.1/1345
4006_A001	UART Baud Rate Registers: Low (UART0_BDL)	8	R/W	04h	53.3.2/1346
4006_A002	UART Control Register 1 (UART0_C1)	8	R/W	00h	53.3.3/1347
4006_A003	UART Control Register 2 (UART0_C2)	8	R/W	00h	53.3.4/1348
4006_A004	UART Status Register 1 (UART0_S1)	8	R	C0h	53.3.5/1350
4006_A005	UART Status Register 2 (UART0_S2)	8	R/W	00h	53.3.6/1353
4006_A006	UART Control Register 3 (UART0_C3)	8	R/W	00h	53.3.7/1355
4006_A007	UART Data Register (UART0_D)	8	R/W	00h	53.3.8/1356
4006_A008	UART Match Address Registers 1 (UART0_MA1)	8	R/W	00h	53.3.9/1357
4006_A009	UART Match Address Registers 2 (UART0_MA2)	8	R/W	00h	53.3.10/ 1358
4006_A00A	UART Control Register 4 (UART0_C4)	8	R/W	00h	53.3.11/ 1358
4006_A00B	UART Control Register 5 (UART0_C5)	8	R/W	00h	53.3.12/ 1359
4006_A00C	UART Extended Data Register (UART0_ED)	8	R	00h	53.3.13/ 1360
4006_A00D	UART Modem Register (UART0_MODEM)	8	R/W	00h	53.3.14/ 1361
4006_A00E	UART Infrared Register (UART0_IR)	8	R/W	00h	53.3.15/ 1362
4006_A010	UART FIFO Parameters (UART0_PFIFO)	8	R/W	See section	53.3.16/ 1363
4006_A011	UART FIFO Control Register (UART0_CFIFO)	8	R/W	00h	53.3.17/ 1364
4006_A012	UART FIFO Status Register (UART0_SFIFO)	8	R/W	C0h	53.3.18/ 1365
4006_A013	UART FIFO Transmit Watermark (UART0_TWFIFO)	8	R/W	00h	53.3.19/ 1366
4006_A014	UART FIFO Transmit Count (UART0_TCFIFO)	8	R	00h	53.3.20/ 1367
4006_A015	UART FIFO Receive Watermark (UART0_RWFIFO)	8	R/W	01h	53.3.21/ 1367
4006_A016	UART FIFO Receive Count (UART0_RCFIFO)	8	R	00h	53.3.22/ 1368
4006_A018	UART 7816 Control Register (UART0_C7816)	8	R/W	00h	53.3.23/ 1368
4006_A019	UART 7816 Interrupt Enable Register (UART0_IE7816)	8	R/W	00h	53.3.24/ 1370
4006_A01A	UART 7816 Interrupt Status Register (UART0_IS7816)	8	R/W	00h	53.3.25/ 1371
4006_A01B	UART 7816 Wait Parameter Register (UART0_WP7816T0)	8	R/W	0Ah	53.3.26/ 1372

Table continues on the next page...

UART memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4006_A01B	UART 7816 Wait Parameter Register (UART0_WP7816T1)	8	R/W	0Ah	53.3.27/ 1373
4006_A01C	UART 7816 Wait N Register (UART0_WN7816)	8	R/W	00h	53.3.28/ 1373
4006_A01D	UART 7816 Wait FD Register (UART0_WF7816)	8	R/W	01h	53.3.29/ 1374
4006_A01E	UART 7816 Error Threshold Register (UART0_ET7816)	8	R/W	00h	53.3.30/ 1374
4006_A01F	UART 7816 Transmit Length Register (UART0_TL7816)	8	R/W	00h	53.3.31/ 1375
4006_B000	UART Baud Rate Registers: High (UART1_BDH)	8	R/W	00h	53.3.1/1345
4006_B001	UART Baud Rate Registers: Low (UART1_BDL)	8	R/W	04h	53.3.2/1346
4006_B002	UART Control Register 1 (UART1_C1)	8	R/W	00h	53.3.3/1347
4006_B003	UART Control Register 2 (UART1_C2)	8	R/W	00h	53.3.4/1348
4006_B004	UART Status Register 1 (UART1_S1)	8	R	C0h	53.3.5/1350
4006_B005	UART Status Register 2 (UART1_S2)	8	R/W	00h	53.3.6/1353
4006_B006	UART Control Register 3 (UART1_C3)	8	R/W	00h	53.3.7/1355
4006_B007	UART Data Register (UART1_D)	8	R/W	00h	53.3.8/1356
4006_B008	UART Match Address Registers 1 (UART1_MA1)	8	R/W	00h	53.3.9/1357
4006_B009	UART Match Address Registers 2 (UART1_MA2)	8	R/W	00h	53.3.10/ 1358
4006_B00A	UART Control Register 4 (UART1_C4)	8	R/W	00h	53.3.11/ 1358
4006_B00B	UART Control Register 5 (UART1_C5)	8	R/W	00h	53.3.12/ 1359
4006_B00C	UART Extended Data Register (UART1_ED)	8	R	00h	53.3.13/ 1360
4006_B00D	UART Modem Register (UART1_MODEM)	8	R/W	00h	53.3.14/ 1361
4006_B00E	UART Infrared Register (UART1_IR)	8	R/W	00h	53.3.15/ 1362
4006_B010	UART FIFO Parameters (UART1_PFIFO)	8	R/W	See section	53.3.16/ 1363
4006_B011	UART FIFO Control Register (UART1_CFIFO)	8	R/W	00h	53.3.17/ 1364
4006_B012	UART FIFO Status Register (UART1_SFIFO)	8	R/W	C0h	53.3.18/ 1365
4006_B013	UART FIFO Transmit Watermark (UART1_TWFIFO)	8	R/W	00h	53.3.19/ 1366
4006_B014	UART FIFO Transmit Count (UART1_TCFIFO)	8	R	00h	53.3.20/ 1367
4006_B015	UART FIFO Receive Watermark (UART1_RWFIFO)	8	R/W	01h	53.3.21/ 1367

Table continues on the next page...

UART memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4006_B016	UART FIFO Receive Count (UART1_RCFIFO)	8	R	00h	53.3.22/ 1368
4006_B018	UART 7816 Control Register (UART1_C7816)	8	R/W	00h	53.3.23/ 1368
4006_B019	UART 7816 Interrupt Enable Register (UART1_IE7816)	8	R/W	00h	53.3.24/ 1370
4006_B01A	UART 7816 Interrupt Status Register (UART1_IS7816)	8	R/W	00h	53.3.25/ 1371
4006_B01B	UART 7816 Wait Parameter Register (UART1_WP7816T0)	8	R/W	0Ah	53.3.26/ 1372
4006_B01B	UART 7816 Wait Parameter Register (UART1_WP7816T1)	8	R/W	0Ah	53.3.27/ 1373
4006_B01C	UART 7816 Wait N Register (UART1_WN7816)	8	R/W	00h	53.3.28/ 1373
4006_B01D	UART 7816 Wait FD Register (UART1_WF7816)	8	R/W	01h	53.3.29/ 1374
4006_B01E	UART 7816 Error Threshold Register (UART1_ET7816)	8	R/W	00h	53.3.30/ 1374
4006_B01F	UART 7816 Transmit Length Register (UART1_TL7816)	8	R/W	00h	53.3.31/ 1375
4006_C000	UART Baud Rate Registers: High (UART2_BDH)	8	R/W	00h	53.3.1/1345
4006_C001	UART Baud Rate Registers: Low (UART2_BDL)	8	R/W	04h	53.3.2/1346
4006_C002	UART Control Register 1 (UART2_C1)	8	R/W	00h	53.3.3/1347
4006_C003	UART Control Register 2 (UART2_C2)	8	R/W	00h	53.3.4/1348
4006_C004	UART Status Register 1 (UART2_S1)	8	R	C0h	53.3.5/1350
4006_C005	UART Status Register 2 (UART2_S2)	8	R/W	00h	53.3.6/1353
4006_C006	UART Control Register 3 (UART2_C3)	8	R/W	00h	53.3.7/1355
4006_C007	UART Data Register (UART2_D)	8	R/W	00h	53.3.8/1356
4006_C008	UART Match Address Registers 1 (UART2_MA1)	8	R/W	00h	53.3.9/1357
4006_C009	UART Match Address Registers 2 (UART2_MA2)	8	R/W	00h	53.3.10/ 1358
4006_C00A	UART Control Register 4 (UART2_C4)	8	R/W	00h	53.3.11/ 1358
4006_C00B	UART Control Register 5 (UART2_C5)	8	R/W	00h	53.3.12/ 1359
4006_C00C	UART Extended Data Register (UART2_ED)	8	R	00h	53.3.13/ 1360
4006_C00D	UART Modem Register (UART2_MODEM)	8	R/W	00h	53.3.14/ 1361
4006_C00E	UART Infrared Register (UART2_IR)	8	R/W	00h	53.3.15/ 1362
4006_C010	UART FIFO Parameters (UART2_PFIFO)	8	R/W	See section	53.3.16/ 1363

Table continues on the next page...

UART memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4006_C011	UART FIFO Control Register (UART2_CFIFO)	8	R/W	00h	53.3.17/ 1364
4006_C012	UART FIFO Status Register (UART2_SFIFO)	8	R/W	C0h	53.3.18/ 1365
4006_C013	UART FIFO Transmit Watermark (UART2_TWFIFO)	8	R/W	00h	53.3.19/ 1366
4006_C014	UART FIFO Transmit Count (UART2_TCFIFO)	8	R	00h	53.3.20/ 1367
4006_C015	UART FIFO Receive Watermark (UART2_RWFIFO)	8	R/W	01h	53.3.21/ 1367
4006_C016	UART FIFO Receive Count (UART2_RCFIFO)	8	R	00h	53.3.22/ 1368
4006_C018	UART 7816 Control Register (UART2_C7816)	8	R/W	00h	53.3.23/ 1368
4006_C019	UART 7816 Interrupt Enable Register (UART2_IE7816)	8	R/W	00h	53.3.24/ 1370
4006_C01A	UART 7816 Interrupt Status Register (UART2_IS7816)	8	R/W	00h	53.3.25/ 1371
4006_C01B	UART 7816 Wait Parameter Register (UART2_WP7816T0)	8	R/W	0Ah	53.3.26/ 1372
4006_C01B	UART 7816 Wait Parameter Register (UART2_WP7816T1)	8	R/W	0Ah	53.3.27/ 1373
4006_C01C	UART 7816 Wait N Register (UART2_WN7816)	8	R/W	00h	53.3.28/ 1373
4006_C01D	UART 7816 Wait FD Register (UART2_WF7816)	8	R/W	01h	53.3.29/ 1374
4006_C01E	UART 7816 Error Threshold Register (UART2_ET7816)	8	R/W	00h	53.3.30/ 1374
4006_C01F	UART 7816 Transmit Length Register (UART2_TL7816)	8	R/W	00h	53.3.31/ 1375

53.3.1 UART Baud Rate Registers: High (UARTx_BDH)

This register, along with the BDL register, controls the prescale divisor for UART baud rate generation. To update the 13-bit baud rate setting (SBR[12:0]), first write to BDH to buffer the high half of the new value and then write to BDL. The working value in BDH does not change until BDL is written.

BDL is reset to a nonzero value, but after reset, the baud rate generator remains disabled until the first time the receiver or transmitter is enabled, that is, when C2[RE] or C2[TE] is set.

Address: Base address + 0h offset

Bit	7	6	5	4	3	2	1	0
Read	LBKDIE	RXEDGIE	0	SBR				
Write								
Reset	0	0	0	0	0	0	0	0

UARTx_BDH field descriptions

Field	Description
7 LBKDIE	LIN Break Detect Interrupt Enable Enables the LIN break detect flag, LBKDIF, to generate interrupt requests, 0 LBKDIF interrupt requests disabled. 1 LBKDIF interrupt requests enabled.
6 RXEDGIE	RxD Input Active Edge Interrupt Enable Enables the receive input active edge, RXEDGIF, to generate interrupt requests. 0 Hardware interrupts from RXEDGIF disabled using polling. 1 RXEDGIF interrupt request enabled.
5 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
SBR	UART Baud Rate Bits The baud rate for the UART is determined by the 13 SBR fields. See Baud rate generation for details. **NOTE:** • The baud rate generator is disabled until C2[TE] or C2[RE] is set for the first time after reset.The baud rate generator is disabled when SBR = 0. • Writing to BDH has no effect without writing to BDL, because writing to BDH puts the data in a temporary location until BDL is written.

53.3.2 UART Baud Rate Registers: Low (UARTx_BDL)

This register, along with the BDH register, controls the prescale divisor for UART baud rate generation. To update the 13-bit baud rate setting, SBR[12:0], first write to BDH to buffer the high half of the new value and then write to BDL. The working value in BDH does not change until BDL is written. BDL is reset to a nonzero value, but after reset, the baud rate generator remains disabled until the first time the receiver or transmitter is enabled, that is, when C2[RE] or C2[TE] is set.

Address: Base address + 1h offset

Bit	7	6	5	4	3	2	1	0
Read	SBR							
Write								
Reset	0	0	0	0	0	1	0	0

UARTx_BDL field descriptions

Field	Description
SBR	UART Baud Rate Bits The baud rate for the UART is determined by the 13 SBR fields. See Baud rate generation for details. **NOTE:** • The baud rate generator is disabled until C2[TE] or C2[RE] is set for the first time after reset.The baud rate generator is disabled when SBR = 0. • Writing to BDH has no effect without writing to BDL, because writing to BDH puts the data in a temporary location until BDL is written. • When the 1/32 narrow pulse width is selected for infrared (IrDA), the baud rate fields must be even, the least significant bit is 0. See MODEM register for more details.

53.3.3 UART Control Register 1 (UARTx_C1)

This read/write register controls various optional features of the UART system.

Address: Base address + 2h offset

Bit	7	6	5	4	3	2	1	0
Read Write	LOOPS	UARTSWAI	RSRC	M	WAKE	ILT	PE	PT
Reset	0	0	0	0	0	0	0	0

UARTx_C1 field descriptions

Field	Description
7 LOOPS	Loop Mode Select When LOOPS is set, the RxD pin is disconnected from the UART and the transmitter output is internally connected to the receiver input. The transmitter and the receiver must be enabled to use the loop function. 0 Normal operation. 1 Loop mode where transmitter output is internally connected to receiver input. The receiver input is determined by RSRC.
6 UARTSWAI	UART Stops in Wait Mode 0 UART clock continues to run in Wait mode. 1 UART clock freezes while CPU is in Wait mode.
5 RSRC	Receiver Source Select This field has no meaning or effect unless the LOOPS field is set. When LOOPS is set, the RSRC field determines the source for the receiver shift register input. 0 Selects internal loop back mode. The receiver input is internally connected to transmitter output. 1 Single wire UART mode where the receiver input is connected to the transmit pin input signal.
4 M	9-bit or 8-bit Mode Select This field must be set when C7816[ISO_7816E] is set/enabled. 0 Normal—start + 8 data bits (MSB/LSB first as determined by MSBF) + stop. 1 Use—start + 9 data bits (MSB/LSB first as determined by MSBF) + stop.

Table continues on the next page...

UARTx_C1 field descriptions (continued)

Field	Description
3 WAKE	Receiver Wakeup Method Select Determines which condition wakes the UART: • Address mark in the most significant bit position of a received data character, or • An idle condition on the receive pin input signal. 0 Idle line wakeup. 1 Address mark wakeup.
2 ILT	Idle Line Type Select Determines when the receiver starts counting logic 1s as idle character bits. The count begins either after a valid start bit or after the stop bit. If the count begins after the start bit, then a string of logic 1s preceding the stop bit can cause false recognition of an idle character. Beginning the count after the stop bit avoids false idle character recognition, but requires properly synchronized transmissions. **NOTE:** • In case the UART is programmed with ILT = 1, a logic of 1'b0 is automatically shifted after a received stop bit, therefore resetting the idle count. • In case the UART is programmed for IDLE line wakeup (RWU = 1 and WAKE = 0), ILT has no effect on when the receiver starts counting logic 1s as idle character bits. In idle line wakeup, an idle character is recognized at anytime the receiver sees 10, 11, or 12 1s depending on the M, PE, and C4[M10] fields. 0 Idle character bit count starts after start bit. 1 Idle character bit count starts after stop bit.
1 PE	Parity Enable Enables the parity function. When parity is enabled, parity function inserts a parity bit in the bit position immediately preceding the stop bit. This field must be set when C7816[ISO_7816E] is set/enabled. 0 Parity function disabled. 1 Parity function enabled.
0 PT	Parity Type Determines whether the UART generates and checks for even parity or odd parity. With even parity, an even number of 1s clears the parity bit and an odd number of 1s sets the parity bit. With odd parity, an odd number of 1s clears the parity bit and an even number of 1s sets the parity bit. This field must be cleared when C7816[ISO_7816E] is set/enabled. 0 Even parity. 1 Odd parity.

53.3.4 UART Control Register 2 (UARTx_C2)

This register can be read or written at any time.

Address: Base address + 3h offset

Bit	7	6	5	4	3	2	1	0
Read Write	TIE	TCIE	RIE	ILIE	TE	RE	RWU	SBK
Reset	0	0	0	0	0	0	0	0

UARTx_C2 field descriptions

Field	Description
7 TIE	Transmitter Interrupt or DMA Transfer Enable. Enables S1[TDRE] to generate interrupt requests or DMA transfer requests, based on the state of C5[TDMAS]. **NOTE:** If C2[TIE] and C5[TDMAS] are both set, then TCIE must be cleared, and D[D] must not be written unless servicing a DMA request. 0 TDRE interrupt and DMA transfer requests disabled. 1 TDRE interrupt or DMA transfer requests enabled.
6 TCIE	Transmission Complete Interrupt Enable Enables the transmission complete flag, S1[TC], to generate interrupt requests . 0 TC interrupt requests disabled. 1 TC interrupt requests enabled.
5 RIE	Receiver Full Interrupt or DMA Transfer Enable Enables S1[RDRF] to generate interrupt requests or DMA transfer requests, based on the state of C5[RDMAS]. 0 RDRF interrupt and DMA transfer requests disabled. 1 RDRF interrupt or DMA transfer requests enabled.
4 ILIE	Idle Line Interrupt Enable Enables the idle line flag, S1[IDLE], to generate interrupt requests 0 IDLE interrupt requests disabled. 1 IDLE interrupt requests enabled.
3 TE	Transmitter Enable Enables the UART transmitter. TE can be used to queue an idle preamble by clearing and then setting TE. When C7816[ISO_7816E] is set/enabled and C7816[TTYPE] = 1, this field is automatically cleared after the requested block has been transmitted. This condition is detected when TL7816[TLEN] = 0 and four additional characters are transmitted. 0 Transmitter off. 1 Transmitter on.
2 RE	Receiver Enable Enables the UART receiver. 0 Receiver off. 1 Receiver on.
1 RWU	Receiver Wakeup Control This field can be set to place the UART receiver in a standby state. RWU automatically clears when an RWU event occurs, that is, an IDLE event when C1[WAKE] is clear or an address match when C1[WAKE] is set. This field must be cleared when C7816[ISO_7816E] is set. **NOTE:** RWU must be set only with C1[WAKE] = 0 (wakeup on idle) if the channel is currently not idle. This can be determined by S2[RAF]. If the flag is set to wake up an IDLE event and the channel

Table continues on the next page...

UARTx_C2 field descriptions (continued)

Field	Description
	is already idle, it is possible that the UART will discard data. This is because the data must be received or a LIN break detected after an IDLE is detected before IDLE is allowed to reasserted. 0 Normal operation. 1 RWU enables the wakeup function and inhibits further receiver interrupt requests. Normally, hardware wakes the receiver by automatically clearing RWU.
0 SBK	Send Break Toggling SBK sends one break character from the following: See Transmitting break characters for the number of logic 0s for the different configurations. Toggling implies clearing the SBK field before the break character has finished transmitting. As long as SBK is set, the transmitter continues to send complete break characters (10, 11, or 12 bits, or 13 or 14 bits). Ensure that C2[TE] is asserted atleast 1 clock before assertion of this bit. • 10, 11, or 12 logic 0s if S2[BRK13] is cleared • 13 or 14 logic 0s if S2[BRK13] is set. This field must be cleared when C7816[ISO_7816E] is set. 0 Normal transmitter operation. 1 Queue break characters to be sent.

53.3.5 UART Status Register 1 (UARTx_S1)

The S1 register provides inputs to the MCU for generation of UART interrupts or DMA requests. This register can also be polled by the MCU to check the status of its fields. To clear a flag, the status register should be read followed by a read or write to D register, depending on the interrupt flag type. Other instructions can be executed between the two steps as long the handling of I/O is not compromised, but the order of operations is important for flag clearing. When a flag is configured to trigger a DMA request, assertion of the associated DMA done signal from the DMA controller clears the flag.

NOTE

- If the condition that results in the assertion of the flag, interrupt, or DMA request is not resolved prior to clearing the flag, the flag, and interrupt/DMA request, reasserts. For example, if the DMA or interrupt service routine fails to write sufficient data to the transmit buffer to raise it above the watermark level, the flag reasserts and generates another interrupt or DMA request.
- Reading an empty data register to clear one of the flags of the S1 register causes the FIFO pointers to become misaligned. A receive FIFO flush reinitializes the pointers. A better way to prevent this situation is to always leave one

byte in FIFO and this byte will be read eventually in clearing the flag bit.

Address: Base address + 4h offset

Bit	7	6	5	4	3	2	1	0
Read	TDRE	TC	RDRF	IDLE	OR	NF	FE	PF
Write								
Reset	1	1	0	0	0	0	0	0

UARTx_S1 field descriptions

Field	Description
7 TDRE	Transmit Data Register Empty Flag TDRE will set when the number of datawords in the transmit buffer (D and C3[T8])is equal to or less than the number indicated by TWFIFO[TXWATER]. A character that is in the process of being transmitted is not included in the count. To clear TDRE, read S1 when TDRE is set and then write to the UART data register (D). For more efficient interrupt servicing, all data except the final value to be written to the buffer must be written to D/C3[T8]. Then S1 can be read before writing the final data value, resulting in the clearing of the TRDE flag. This is more efficient because the TDRE reasserts until the watermark has been exceeded. So, attempting to clear the TDRE with every write will be ineffective until sufficient data has been written. 0 The amount of data in the transmit buffer is greater than the value indicated by TWFIFO[TXWATER]. 1 The amount of data in the transmit buffer is less than or equal to the value indicated by TWFIFO[TXWATER] at some point in time since the flag has been cleared.
6 TC	Transmit Complete Flag TC is set when the transmit buffer is empty and no data, preamble, or break character is being transmitted. When TC is set, the transmit data output signal becomes idle (logic 1). TC is cleared by reading S1 with TC set and then doing one of the following: When C7816[ISO_7816E] is set/enabled, this field is set after any NACK signal has been received, but prior to any corresponding guard times expiring. • Writing to D to transmit new data. • Queuing a preamble by clearing and then setting C2[TE]. • Queuing a break character by writing 1 to SBK in C2. 0 Transmitter active (sending data, a preamble, or a break). 1 Transmitter idle (transmission activity complete).
5 RDRF	Receive Data Register Full Flag RDRF is set when the number of datawords in the receive buffer is equal to or more than the number indicated by RWFIFO[RXWATER]. A dataword that is in the process of being received is not included in the count. To clear RDRF, read S1 when RDRF is set and then read D. For more efficient interrupt and DMA operation, read all data except the final value from the buffer, using D/C3[T8]/ED. Then read S1 and the final data value, resulting in the clearing of the RDRF flag. Even if RDRF is set, data will continue to be received until an overrun condition occurs.RDRF is prevented from setting while S2[LBKDE] is set. Additionally, when S2[LBKDE] is set, the received datawords are stored in the receive buffer but over-write each other. 0 The number of datawords in the receive buffer is less than the number indicated by RXWATER. 1 The number of datawords in the receive buffer is equal to or greater than the number indicated by RXWATER at some point in time since this flag was last cleared.
4 IDLE	Idle Line Flag

Table continues on the next page...

UARTx_S1 field descriptions (continued)

Field	Description
	After the IDLE flag is cleared, a frame must be received (although not necessarily stored in the data buffer, for example if C2[RWU] is set), or a LIN break character must set the S2[LBKDIF] flag before an idle condition can set the IDLE flag. To clear IDLE, read UART status S1 with IDLE set and then read D. IDLE is set when either of the following appear on the receiver input: • 10 consecutive logic 1s if C1[M] = 0 • 11 consecutive logic 1s if C1[M] = 1 and C4[M10] = 0 • 12 consecutive logic 1s if C1[M] = 1, C4[M10] = 1, and C1[PE] = 1 Idle detection is not supported when7816Eis set/enabled and hence this flag is ignored. **NOTE:** When RWU is set and WAKE is cleared, an idle line condition sets the IDLE flag if RWUID is set, else the IDLE flag does not become set. 0 Receiver input is either active now or has never become active since the IDLE flag was last cleared. 1 Receiver input has become idle or the flag has not been cleared since it last asserted.
3 OR	Receiver Overrun Flag OR is set when software fails to prevent the receive data register from overflowing with data. The OR bit is set immediately after the stop bit has been completely received for the dataword that overflows the buffer and all the other error flags (FE, NF, and PF) are prevented from setting. The data in the shift register is lost, but the data already in the UART data registers is not affected. If the OR flag is set, no data is stored in the data buffer even if sufficient room exists. Additionally, while the OR flag is set, the RDRF and IDLE flags are blocked from asserting, that is, transition from an inactive to an active state. To clear OR, read S1 when OR is set and then read D. See functional description for more details regarding the operation of the OR bit.If LBKDE is enabled and a LIN Break is detected, the OR field asserts if S2[LBKDIF] is not cleared before the next data character is received. In 7816 mode, it is possible to configure a NACK to be returned by programing C7816[ONACK]. 0 No overrun has occurred since the last time the flag was cleared. 1 Overrun has occurred or the overrun flag has not been cleared since the last overrun occured.
2 NF	Noise Flag NF is set when the UART detects noise on the receiver input. NF does not become set in the case of an overrun or while the LIN break detect feature is enabled (S2[LBKDE] = 1). When NF is set, it indicates only that a dataword has been received with noise since the last time it was cleared. There is no guarantee that the first dataword read from the receive buffer has noise or that there is only one dataword in the buffer that was received with noise unless the receive buffer has a depth of one. To clear NF, read S1 and then read D. 0 No noise detected since the last time this flag was cleared. If the receive buffer has a depth greater than 1 then there may be data in the receiver buffer that was received with noise. 1 At least one dataword was received with noise detected since the last time the flag was cleared.
1 FE	Framing Error Flag FE is set when a logic 0 is accepted as the stop bit. FE does not set in the case of an overrun or while the LIN break detect feature is enabled (S2[LBKDE] = 1). FE inhibits further data reception until it is cleared. To clear FE, read S1 with FE set and then read D. The last data in the receive buffer represents the data that was received with the frame error enabled. Framing errors are not supported when 7816E is set/enabled. However, if this flag is set, data is still not received in 7816 mode. 0 No framing error detected. 1 Framing error.
0 PF	Parity Error Flag

Table continues on the next page...

UARTx_S1 field descriptions (continued)

Field	Description
	PF is set when PE is set and the parity of the received data does not match its parity bit. The PF is not set in the case of an overrun condition. When PF is set, it indicates only that a dataword was received with parity error since the last time it was cleared. There is no guarantee that the first dataword read from the receive buffer has a parity error or that there is only one dataword in the buffer that was received with a parity error, unless the receive buffer has a depth of one. To clear PF, read S1 and then read D., S2[LBKDE] is disabled,Within the receive buffer structure the received dataword is tagged if it is received with a parity error. This information is available by reading the ED register prior to reading the D register. 0 No parity error detected since the last time this flag was cleared. If the receive buffer has a depth greater than 1, then there may be data in the receive buffer what was received with a parity error. 1 At least one dataword was received with a parity error since the last time this flag was cleared.

53.3.6 UART Status Register 2 (UARTx_S2)

The S2 register provides inputs to the MCU for generation of UART interrupts or DMA requests. Also, this register can be polled by the MCU to check the status of these bits. This register can be read or written at any time, with the exception of the MSBF and RXINV bits, which should be changed by the user only between transmit and receive packets.

Address: Base address + 5h offset

Bit	7	6	5	4	3	2	1	0
Read	LBKDIF	RXEDGIF	MSBF	RXINV	RWUID	BRK13	LBKDE	RAF
Write	w1c	w1c						
Reset	0	0	0	0	0	0	0	0

UARTx_S2 field descriptions

Field	Description
7 LBKDIF	LIN Break Detect Interrupt Flag LBKDIF is set when LBKDE is set and a LIN break character is detected on the receiver input. The LIN break characters are 11 consecutive logic 0s if C1[M] = 0 or 12 consecutive logic 0s if C1[M] = 1. LBKDIF is set after receiving the last LIN break character. LBKDIF is cleared by writing a 1 to it. 0 No LIN break character detected. 1 LIN break character detected.
6 RXEDGIF	RxD Pin Active Edge Interrupt Flag RXEDGIF is set when an active edge occurs on the RxD pin. The active edge is falling if RXINV = 0, and rising if RXINV=1. RXEDGIF is cleared by writing a 1 to it. See for additional details. RXEDGIF description **NOTE:** The active edge is detected only in two wire mode and on receiving data coming from the RxD pin. 0 No active edge on the receive pin has occurred. 1 An active edge on the receive pin has occurred.

Table continues on the next page...

UARTx_S2 field descriptions (continued)

Field	Description
5 MSBF	Most Significant Bit First Setting this field reverses the order of the bits that are transmitted and received on the wire. This field does not affect the polarity of the bits, the location of the parity bit, or the location of the start or stop bits. This field is automatically set when C7816[INIT] and C7816[ISO7816E] are enabled and an initial character is detected in T = 0 protocol mode. 0 LSB (bit0) is the first bit that is transmitted following the start bit. Further, the first bit received after the start bit is identified as bit0. 1 MSB (bit8, bit7 or bit6) is the first bit that is transmitted following the start bit, depending on the setting of C1[M] and C1[PE]. Further, the first bit received after the start bit is identified as bit8, bit7, or bit6, depending on the setting of C1[M] and C1[PE].
4 RXINV	Receive Data Inversion Setting this field reverses the polarity of the received data input. In NRZ format, a one is represented by a mark and a zero is represented by a space for normal polarity, and the opposite for inverted polarity. In IrDA format, a zero is represented by short high pulse in the middle of a bit time remaining idle low for a one for normal polarity. A zero is represented by a short low pulse in the middle of a bit time remaining idle high for a one for inverted polarity.This field is automatically set when C7816[INIT] and C7816[ISO7816E] are enabled and an initial character is detected in T = 0 protocol mode. **NOTE:** Setting RXINV inverts the RxD input for data bits, start and stop bits, break, and idle. When C7816[ISO7816E] is set/enabled, only the data bits and the parity bit are inverted. 0 Receive data is not inverted. 1 Receive data is inverted.
3 RWUID	Receive Wakeup Idle Detect When RWU is set and WAKE is cleared, this field controls whether the idle character that wakes the receiver sets S1[IDLE]. This field must be cleared when C7816[ISO7816E] is set/enabled. 0 S1[IDLE] is not set upon detection of an idle character. 1 S1[IDLE] is set upon detection of an idle character.
2 BRK13	Break Transmit Character Length Determines whether the transmit break character is 10, 11, or 12 bits long, or 13 or 14 bits long. See for the length of the break character for the different configurations. The detection of a framing error is not affected by this field. Transmitting break characters 0 Break character is 10, 11, or 12 bits long. 1 Break character is 13 or 14 bits long.
1 LBKDE	LIN Break Detection Enable Enables the LIN Break detection feature. While LBKDE is set, S1[RDRF], S1[NF], S1[FE], and S1[PF] are prevented from setting. When LBKDE is set, see . Overrun operationLBKDE must be cleared when C7816[ISO7816E] is set. 0 Break character detection is disabled. 1 Break character is detected at length of 11 bit times if C1[M] = 0 or 12 bits time if C1[M] = 1.
0 RAF	Receiver Active Flag RAF is set when the UART receiver detects a logic 0 during the RT1 time period of the start bit search. RAF is cleared when the receiver detects an idle character when C7816[ISO7816E] is cleared/disabled.

Table continues on the next page...

UARTx_S2 field descriptions (continued)

Field	Description
	When C7816[ISO7816E] is enabled, the RAF is cleared if the C7816[TTYPE] = 0 expires or the C7816[TTYPE] = 1 expires. **NOTE:** In case C7816[ISO7816E] is set and C7816[TTYPE] = 0, it is possible to configure the guard time to 12. However, if a NACK is required to be transmitted, the data transfer actually takes 13 ETU with the 13th ETU slot being a inactive buffer. Therefore, in this situation, the RAF may deassert one ETU prior to actually being inactive. 0 UART receiver idle/inactive waiting for a start bit. 1 UART receiver active, RxD input not idle.

53.3.7 UART Control Register 3 (UARTx_C3)

Writing R8 does not have any effect. TXDIR and TXINV can be changed only between transmit and receive packets.

Address: Base address + 6h offset

Bit	7	6	5	4	3	2	1	0
Read	R8	T8	TXDIR	TXINV	ORIE	NEIE	FEIE	PEIE
Write	▨							
Reset	0	0	0	0	0	0	0	0

UARTx_C3 field descriptions

Field	Description
7 R8	Received Bit 8 R8 is the ninth data bit received when the UART is configured for 9-bit data format, that is, if C1[M] = 1 or C4[M10] = 1. The R8 value corresponds to the current data value in the UARTx_D register. To read the 9th bit, read the value of UARTx_C3[R8], then read the UARTx_D register.
6 T8	Transmit Bit 8 T8 is the ninth data bit transmitted when the UART is configured for 9-bit data format, that is, if C1[M] = 1 or C4[M10] = 1. **NOTE:** If the value of T8 is the same as in the previous transmission, T8 does not have to be rewritten. The same value is transmitted until T8 is rewritten. To correctly transmit the 9th bit, write UARTx_C3[T8] to the desired value, then write the UARTx_D register with the remaining data.
5 TXDIR	Transmitter Pin Data Direction in Single-Wire mode Determines whether the TXD pin is used as an input or output in the single-wire mode of operation. This field is relevant only to the single wire mode. When C7816[ISO7816E] is set/enabled and C7816[TTYPE] = 1, this field is automatically cleared after the requested block is transmitted. This condition is detected when TL7816[TLEN] = 0 and 4 additional characters are transmitted. Additionally, if C7816[ISO7816E] is set/enabled and C7816[TTYPE] = 0 and a NACK is being transmitted, the hardware automatically overrides this field as needed. In this situation, TXDIR does not reflect the temporary state associated with the NACK.

Table continues on the next page...

UARTx_C3 field descriptions (continued)

Field	Description
	0 TXD pin is an input in single wire mode. 1 TXD pin is an output in single wire mode.
4 TXINV	Transmit Data Inversion. Setting this field reverses the polarity of the transmitted data output. In NRZ format, a one is represented by a mark and a zero is represented by a space for normal polarity, and the opposite for inverted polarity. In IrDA format, a zero is represented by short high pulse in the middle of a bit time remaining idle low for a one for normal polarity, and a zero is represented by short low pulse in the middle of a bit time remaining idle high for a one for inverted polarity.This field is automatically set when C7816[INIT] and C7816[ISO7816E] are enabled and an initial character is detected in T = 0 protocol mode. **NOTE:** Setting TXINV inverts all transmitted values, including idle, break, start, and stop bits. In loop mode, if TXINV is set, the receiver gets the transmit inversion bit when RXINV is disabled. When C7816[ISO7816E] is set/enabled then only the transmitted data bits and parity bit are inverted. 0 Transmit data is not inverted. 1 Transmit data is inverted.
3 ORIE	Overrun Error Interrupt Enable Enables the overrun error flag, S1[OR], to generate interrupt requests. 0 OR interrupts are disabled. 1 OR interrupt requests are enabled.
2 NEIE	Noise Error Interrupt Enable Enables the noise flag, S1[NF], to generate interrupt requests. 0 NF interrupt requests are disabled. 1 NF interrupt requests are enabled.
1 FEIE	Framing Error Interrupt Enable Enables the framing error flag, S1[FE], to generate interrupt requests. 0 FE interrupt requests are disabled. 1 FE interrupt requests are enabled.
0 PEIE	Parity Error Interrupt Enable Enables the parity error flag, S1[PF], to generate interrupt requests. 0 PF interrupt requests are disabled. 1 PF interrupt requests are enabled.

53.3.8 UART Data Register (UARTx_D)

This register is actually two separate registers. Reads return the contents of the read-only receive data register and writes go to the write-only transmit data register.

NOTE

- In 8-bit or 9-bit data format, only UART data register (D) needs to be accessed to clear the S1[RDRF] bit (assuming receiver buffer level is less than RWFIFO[RXWATER]). The C3 register needs to be read, prior to the D register, only if the ninth bit of data needs to be captured. Similarly, the ED register needs to be read, prior to the D register, only if the additional flag data for the dataword needs to be captured.

- In the normal 8-bit mode (M bit cleared) if the parity is enabled, you get seven data bits and one parity bit. That one parity bit is loaded into the D register. So, for the data bits, mask off the parity bit from the value you read out of this register.
- When transmitting in 9-bit data format and using 8-bit write instructions, write first to transmit bit 8 in UART control register 3 (C3[T8]), then D. A write to C3[T8] stores the data in a temporary register. If D register is written first, and then the new data on data bus is stored in D, the temporary value written by the last write to C3[T8] gets stored in the C3[T8] register.

Address: Base address + 7h offset

Bit	7	6	5	4	3	2	1	0
Read Write	RT							
Reset	0	0	0	0	0	0	0	0

UARTx_D field descriptions

Field	Description
RT	Reads return the contents of the read-only receive data register and writes go to the write-only transmit data register.

53.3.9 UART Match Address Registers 1 (UARTx_MA1)

The MA1 and MA2 registers are compared to input data addresses when the most significant bit is set and the associated C4[MAEN] field is set. If a match occurs, the following data is transferred to the data register. If a match fails, the following data is discarded. These registers can be read and written at anytime.

Address: Base address + 8h offset

Bit	7	6	5	4	3	2	1	0
Read Write	MA							
Reset	0	0	0	0	0	0	0	0

UARTx_MA1 field descriptions

Field	Description
MA	Match Address

53.3.10 UART Match Address Registers 2 (UARTx_MA2)

These registers can be read and written at anytime. The MA1 and MA2 registers are compared to input data addresses when the most significant bit is set and the associated C4[MAEN] field is set. If a match occurs, the following data is transferred to the data register. If a match fails, the following data is discarded.

Address: Base address + 9h offset

Bit	7	6	5	4	3	2	1	0
Read Write	MA							
Reset	0	0	0	0	0	0	0	0

UARTx_MA2 field descriptions

Field	Description
MA	Match Address

53.3.11 UART Control Register 4 (UARTx_C4)

Address: Base address + Ah offset

Bit	7	6	5	4	3	2	1	0
Read Write	MAEN1	MAEN2	M10	BRFA				
Reset	0	0	0	0	0	0	0	0

UARTx_C4 field descriptions

Field	Description
7 MAEN1	Match Address Mode Enable 1 See Match address operation for more information. 0 All data received is transferred to the data buffer if MAEN2 is cleared. 1 All data received with the most significant bit cleared, is discarded. All data received with the most significant bit set, is compared with contents of MA1 register. If no match occurs, the data is discarded. If match occurs, data is transferred to the data buffer. This field must be cleared when C7816[ISO7816E] is set/enabled.
6 MAEN2	Match Address Mode Enable 2 See Match address operation for more information.

Table continues on the next page...

UARTx_C4 field descriptions (continued)

Field	Description
	0 All data received is transferred to the data buffer if MAEN1 is cleared. 1 All data received with the most significant bit cleared, is discarded. All data received with the most significant bit set, is compared with contents of MA2 register. If no match occurs, the data is discarded. If a match occurs, data is transferred to the data buffer. This field must be cleared when C7816[ISO7816E] is set/enabled.
5 M10	10-bit Mode select Causes a tenth, non-memory mapped bit to be part of the serial transmission. This tenth bit is generated and interpreted as a parity bit. The M10 field does not affect the LIN send or detect break behavior. If M10 is set, then both C1[M] and C1[PE] must also be set. This field must be cleared when C7816[ISO7816E] is set/enabled. See Data format (non ISO-7816) for more information. 0 The parity bit is the ninth bit in the serial transmission. 1 The parity bit is the tenth bit in the serial transmission.
BRFA	Baud Rate Fine Adjust This bit field is used to add more timing resolution to the average baud frequency, in increments of 1/32. See Baud rate generation for more information.

53.3.12 UART Control Register 5 (UARTx_C5)

Address: Base address + Bh offset

Bit	7	6	5	4	3	2	1	0
Read	TDMAS	0	RDMAS	0	0			
Write								
Reset	0	0	0	0	0	0	0	0

UARTx_C5 field descriptions

Field	Description
7 TDMAS	Transmitter DMA Select Configures the transmit data register empty flag, S1[TDRE], to generate interrupt or DMA requests if C2[TIE] is set. **NOTE:** • If C2[TIE] is cleared, TDRE DMA and TDRE interrupt request signals are not asserted when the TDRE flag is set, regardless of the state of TDMAS. • If C2[TIE] and TDMAS are both set, then C2[TCIE] must be cleared, and D must not be written unless a DMA request is being serviced. 0 If C2[TIE] is set and the S1[TDRE] flag is set, the TDRE interrupt request signal is asserted to request interrupt service. 1 If C2[TIE] is set and the S1[TDRE] flag is set, the TDRE DMA request signal is asserted to request a DMA transfer.
6 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
5 RDMAS	Receiver Full DMA Select

Table continues on the next page...

UARTx_C5 field descriptions (continued)

Field	Description
	Configures the receiver data register full flag, S1[RDRF], to generate interrupt or DMA requests if C2[RIE] is set. **NOTE:** If C2[RIE] is cleared, and S1[RDRF] is set, the RDRF DMA and RDFR interrupt request signals are not asserted, regardless of the state of RDMAS. 0 If C2[RIE] and S1[RDRF] are set, the RDFR interrupt request signal is asserted to request an interrupt service. 1 If C2[RIE] and S1[RDRF] are set, the RDRF DMA request signal is asserted to request a DMA transfer.
4 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

53.3.13 UART Extended Data Register (UARTx_ED)

This register contains additional information flags that are stored with a received dataword. This register may be read at any time but contains valid data only if there is a dataword in the receive FIFO.

NOTE

- The data contained in this register represents additional information regarding the conditions on which a dataword was received. The importance of this data varies with the application, and in some cases maybe completely optional. These fields automatically update to reflect the conditions of the next dataword whenever D is read.
- If S1[NF] and S1[PF] have not been set since the last time the receive buffer was empty, the NOISY and PARITYE fields will be zero.

Address: Base address + Ch offset

Bit	7	6	5	4	3	2	1	0
Read	NOISY	PARITYE	0					
Write	▨	▨	▨					
Reset	0	0	0	0	0	0	0	0

UARTx_ED field descriptions

Field	Description
7 NOISY	The current received dataword contained in D and C3[R8] was received with noise.

Table continues on the next page...

UARTx_ED field descriptions (continued)

Field	Description
	0 The dataword was received without noise. 1 The data was received with noise.
6 PARITYE	The current received dataword contained in D and C3[R8] was received with a parity error. 0 The dataword was received without a parity error. 1 The dataword was received with a parity error.
Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

53.3.14 UART Modem Register (UARTx_MODEM)

The MODEM register controls options for setting the modem configuration.

NOTE

RXRTSE, TXRTSPOL, TXRTSE, and TXCTSE must all be cleared when C7816[ISO7816EN] is enabled. This will cause the RTS to deassert during ISO-7816 wait times. The ISO-7816 protocol does not use the RTS and CTS signals.

Address: Base address + Dh offset

Bit	7	6	5	4	3	2	1	0
Read	0				RXRTSE	TXRTSPOL	TXRTSE	TXCTSE
Write								
Reset	0	0	0	0	0	0	0	0

UARTx_MODEM field descriptions

Field	Description
7–4 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
3 RXRTSE	Receiver request-to-send enable Allows the RTS output to control the CTS input of the transmitting device to prevent receiver overrun. **NOTE:** Do not set both RXRTSE and TXRTSE. 0 The receiver has no effect on RTS. 1 RTS is deasserted if the number of characters in the receiver data register (FIFO) is equal to or greater than RWFIFO[RXWATER]. RTS is asserted when the number of characters in the receiver data register (FIFO) is less than RWFIFO[RXWATER]. See Hardware flow control
2 TXRTSPOL	Transmitter request-to-send polarity Controls the polarity of the transmitter RTS. TXRTSPOL does not affect the polarity of the receiver RTS. RTS will remain negated in the active low state unless TXRTSE is set. 0 Transmitter RTS is active low. 1 Transmitter RTS is active high.

Table continues on the next page...

UARTx_MODEM field descriptions (continued)

Field	Description
1 TXRTSE	Transmitter request-to-send enable Controls RTS before and after a transmission. 0 The transmitter has no effect on RTS. 1 When a character is placed into an empty transmitter data buffer , RTS asserts one bit time before the start bit is transmitted. RTS deasserts one bit time after all characters in the transmitter data buffer and shift register are completely sent, including the last stop bit. (FIFO) **NOTE:** Ensure that C2[TE] is asserted before assertion of this bit.
0 TXCTSE	Transmitter clear-to-send enable TXCTSE controls the operation of the transmitter. TXCTSE can be set independently from the state of TXRTSE and RXRTSE. 0 CTS has no effect on the transmitter. 1 Enables clear-to-send operation. The transmitter checks the state of CTS each time it is ready to send a character. If CTS is asserted, the character is sent. If CTS is deasserted, the signal TXD remains in the mark state and transmission is delayed until CTS is asserted. Changes in CTS as a character is being sent do not affect its transmission.

53.3.15 UART Infrared Register (UARTx_IR)

The IR register controls options for setting the infrared configuration.

Address: Base address + Eh offset

Bit	7	6	5	4	3	2	1	0
Read	0					IREN	TNP	
Write								
Reset	0	0	0	0	0	0	0	0

UARTx_IR field descriptions

Field	Description
7–3 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
2 IREN	Infrared enable Enables/disables the infrared modulation/demodulation. 0 IR disabled. 1 IR enabled.
TNP	Transmitter narrow pulse Enables whether the UART transmits a 1/16, 3/16, 1/32, or 1/4 narrow pulse. 00 3/16. 01 1/16. 10 1/32. 11 1/4.

53.3.16 UART FIFO Parameters (UARTx_PFIFO)

This register provides the ability for the programmer to turn on and off FIFO functionality. It also provides the size of the FIFO that has been implemented. This register may be read at any time. This register must be written only when C2[RE] and C2[TE] are cleared/not set and when the data buffer/FIFO is empty.

Address: Base address + 10h offset

Bit	7	6	5	4	3	2	1	0
Read	TXFE	TXFIFOSIZE			RXFE	RXFIFOSIZE		
Write								
Reset	0	*	*	*	0	*	*	*

* Notes:

- TXFIFOSIZE field: The reset value depends on whether the specific UART instance supports the FIFO and on the size of that FIFO. See the Chip Configuration details for more information on the FIFO size supported for each UART instance.
- RXFIFOSIZE field: The reset value depends on whether the specific UART instance supports the FIFO and on the size of that FIFO. See the Chip Configuration details for more information on the FIFO size supported for each UART instance.

UARTx_PFIFO field descriptions

Field	Description
7 TXFE	Transmit FIFO Enable When this field is set, the built in FIFO structure for the transmit buffer is enabled. The size of the FIFO structure is indicated by TXFIFOSIZE. If this field is not set, the transmit buffer operates as a FIFO of depth one dataword regardless of the value in TXFIFOSIZE. Both C2[TE] and C2[RE] must be cleared prior to changing this field. Additionally, TXFLUSH and RXFLUSH commands must be issued immediately after changing this field. 0 Transmit FIFO is not enabled. Buffer is depth 1. (Legacy support). 1 Transmit FIFO is enabled. Buffer is depth indicated by TXFIFOSIZE.
6–4 TXFIFOSIZE	Transmit FIFO. Buffer Depth The maximum number of transmit datawords that can be stored in the transmit buffer. This field is read only. 000 Transmit FIFO/Buffer depth = 1 dataword. 001 Transmit FIFO/Buffer depth = 4 datawords. 010 Transmit FIFO/Buffer depth = 8 datawords. 011 Transmit FIFO/Buffer depth = 16 datawords. 100 Transmit FIFO/Buffer depth = 32 datawords. 101 Transmit FIFO/Buffer depth = 64 datawords. 110 Transmit FIFO/Buffer depth = 128 datawords. 111 Reserved.
3 RXFE	Receive FIFO Enable

Table continues on the next page...

UARTx_PFIFO field descriptions (continued)

Field	Description
	When this field is set, the built in FIFO structure for the receive buffer is enabled. The size of the FIFO structure is indicated by the RXFIFOSIZE field. If this field is not set, the receive buffer operates as a FIFO of depth one dataword regardless of the value in RXFIFOSIZE. Both C2[TE] and C2[RE] must be cleared prior to changing this field. Additionally, TXFLUSH and RXFLUSH commands must be issued immediately after changing this field. 0 Receive FIFO is not enabled. Buffer is depth 1. (Legacy support) 1 Receive FIFO is enabled. Buffer is depth indicted by RXFIFOSIZE.
RXFIFOSIZE	Receive FIFO. Buffer Depth The maximum number of receive datawords that can be stored in the receive buffer before an overrun occurs. This field is read only. 000 Receive FIFO/Buffer depth = 1 dataword. 001 Receive FIFO/Buffer depth = 4 datawords. 010 Receive FIFO/Buffer depth = 8 datawords. 011 Receive FIFO/Buffer depth = 16 datawords. 100 Receive FIFO/Buffer depth = 32 datawords. 101 Receive FIFO/Buffer depth = 64 datawords. 110 Receive FIFO/Buffer depth = 128 datawords. 111 Reserved.

53.3.17 UART FIFO Control Register (UARTx_CFIFO)

This register provides the ability to program various control fields for FIFO operation. This register may be read or written at any time. Note that writing to TXFLUSH and RXFLUSH may result in data loss and requires careful action to prevent unintended/ unpredictable behavior. Therefore, it is recommended that TE and RE be cleared prior to flushing the corresponding FIFO.

Address: Base address + 11h offset

Bit	7	6	5	4	3	2	1	0
Read	0	0	0			RXOFE	TXOFE	RXUFE
Write	TXFLUSH	RXFLUSH						
Reset	0	0	0	0	0	0	0	0

UARTx_CFIFO field descriptions

Field	Description
7 TXFLUSH	Transmit FIFO/Buffer Flush Writing to this field causes all data that is stored in the transmit FIFO/buffer to be flushed. This does not affect data that is in the transmit shift register. 0 No flush operation occurs. 1 All data in the transmit FIFO/Buffer is cleared out.

Table continues on the next page...

UARTx_CFIFO field descriptions (continued)

Field	Description
6 RXFLUSH	Receive FIFO/Buffer Flush Writing to this field causes all data that is stored in the receive FIFO/buffer to be flushed. This does not affect data that is in the receive shift register. 0 No flush operation occurs. 1 All data in the receive FIFO/buffer is cleared out.
5–3 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
2 RXOFE	Receive FIFO Overflow Interrupt Enable When this field is set, the RXOF flag generates an interrupt to the host. 0 RXOF flag does not generate an interrupt to the host. 1 RXOF flag generates an interrupt to the host.
1 TXOFE	Transmit FIFO Overflow Interrupt Enable When this field is set, the TXOF flag generates an interrupt to the host. 0 TXOF flag does not generate an interrupt to the host. 1 TXOF flag generates an interrupt to the host.
0 RXUFE	Receive FIFO Underflow Interrupt Enable When this field is set, the RXUF flag generates an interrupt to the host. 0 RXUF flag does not generate an interrupt to the host. 1 RXUF flag generates an interrupt to the host.

53.3.18 UART FIFO Status Register (UARTx_SFIFO)

This register provides status information regarding the transmit and receiver buffers/ FIFOs, including interrupt information. This register may be written to or read at any time.

Address: Base address + 12h offset

Bit	7	6	5	4	3	2	1	0
Read	TXEMPT	RXEMPT	0			RXOF	TXOF	RXUF
Write						w1c	w1c	w1c
Reset	1	1	0	0	0	0	0	0

UARTx_SFIFO field descriptions

Field	Description
7 TXEMPT	Transmit Buffer/FIFO Empty Asserts when there is no data in the Transmit FIFO/buffer. This field does not take into account data that is in the transmit shift register.

Table continues on the next page...

UARTx_SFIFO field descriptions (continued)

Field	Description
	0 Transmit buffer is not empty. 1 Transmit buffer is empty.
6 RXEMPT	Receive Buffer/FIFO Empty Asserts when there is no data in the receive FIFO/Buffer. This field does not take into account data that is in the receive shift register. 0 Receive buffer is not empty. 1 Receive buffer is empty.
5–3 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
2 RXOF	Receiver Buffer Overflow Flag Indicates that more data has been written to the receive buffer than it can hold. This field will assert regardless of the value of CFIFO[RXOFE]. However, an interrupt will be issued to the host only if CFIFO[RXOFE] is set. This flag is cleared by writing a 1. 0 No receive buffer overflow has occurred since the last time the flag was cleared. 1 At least one receive buffer overflow has occurred since the last time the flag was cleared.
1 TXOF	Transmitter Buffer Overflow Flag Indicates that more data has been written to the transmit buffer than it can hold. This field will assert regardless of the value of CFIFO[TXOFE]. However, an interrupt will be issued to the host only if CFIFO[TXOFE] is set. This flag is cleared by writing a 1. 0 No transmit buffer overflow has occurred since the last time the flag was cleared. 1 At least one transmit buffer overflow has occurred since the last time the flag was cleared.
0 RXUF	Receiver Buffer Underflow Flag Indicates that more data has been read from the receive buffer than was present. This field will assert regardless of the value of CFIFO[RXUFE]. However, an interrupt will be issued to the host only if CFIFO[RXUFE] is set. This flag is cleared by writing a 1. 0 No receive buffer underflow has occurred since the last time the flag was cleared. 1 At least one receive buffer underflow has occurred since the last time the flag was cleared.

53.3.19 UART FIFO Transmit Watermark (UARTx_TWFIFO)

This register provides the ability to set a programmable threshold for notification of needing additional transmit data. This register may be read at any time but must be written only when C2[TE] is not set. Changing the value of the watermark will not clear the S1[TDRE] flag.

Address: Base address + 13h offset

Bit	7	6	5	4	3	2	1	0
Read Write	TXWATER							
Reset	0	0	0	0	0	0	0	0

UARTx_TWFIFO field descriptions

Field	Description
TXWATER	Transmit Watermark When the number of datawords in the transmit FIFO/buffer is equal to or less than the value in this register field, an interrupt via S1[TDRE] or a DMA request via C5[TDMAS] is generated as determined by C5[TDMAS] and C2[TIE]. For proper operation, the value in TXWATER must be set to be less than the size of the transmit buffer/FIFO size as indicated by PFIFO[TXFIFOSIZE] and PFIFO[TXFE].

53.3.20 UART FIFO Transmit Count (UARTx_TCFIFO)

This is a read only register that indicates how many datawords are currently in the transmit buffer/FIFO. It may be read at any time.

Address: Base address + 14h offset

Bit	7	6	5	4	3	2	1	0
Read	TXCOUNT							
Write	▨							
Reset	0	0	0	0	0	0	0	0

UARTx_TCFIFO field descriptions

Field	Description
TXCOUNT	Transmit Counter The value in this register indicates the number of datawords that are in the transmit FIFO/buffer. If a dataword is being transmitted, that is, in the transmit shift register, it is not included in the count. This value may be used in conjunction with PFIFO[TXFIFOSIZE] to calculate how much room is left in the transmit FIFO/buffer.

53.3.21 UART FIFO Receive Watermark (UARTx_RWFIFO)

This register provides the ability to set a programmable threshold for notification of the need to remove data from the receiver FIFO/buffer. This register may be read at any time but must be written only when C2[RE] is not asserted. Changing the value in this register will not clear S1[RDRF].

Address: Base address + 15h offset

Bit	7	6	5	4	3	2	1	0
Read Write	RXWATER							
Reset	0	0	0	0	0	0	0	1

UARTx_RWFIFO field descriptions

Field	Description
RXWATER	Receive Watermark When the number of datawords in the receive FIFO/buffer is equal to or greater than the value in this register field, an interrupt via S1[RDRF] or a DMA request via C5[RDMAS] is generated as determined by C5[RDMAS] and C2[RIE]. For proper operation, the value in RXWATER must be set to be less than the receive FIFO/buffer size as indicated by PFIFO[RXFIFOSIZE] and PFIFO[RXFE] and must be greater than 0.

53.3.22 UART FIFO Receive Count (UARTx_RCFIFO)

This is a read only register that indicates how many datawords are currently in the receive FIFO/buffer. It may be read at any time.

Address: Base address + 16h offset

Bit	7	6	5	4	3	2	1	0
Read	RXCOUNT							
Write	▨							
Reset	0	0	0	0	0	0	0	0

UARTx_RCFIFO field descriptions

Field	Description
RXCOUNT	Receive Counter The value in this register indicates the number of datawords that are in the receive FIFO/buffer. If a dataword is being received, that is, in the receive shift register, it is not included in the count. This value may be used in conjunction with PFIFO[RXFIFOSIZE] to calculate how much room is left in the receive FIFO/buffer.

53.3.23 UART 7816 Control Register (UARTx_C7816)

The C7816 register is the primary control register for ISO-7816 specific functionality. This register is specific to 7816 functionality and the values in this register have no effect on UART operation and should be ignored if ISO_7816E is not set/enabled. This register may be read at any time but values must be changed only when ISO_7816E is not set.

Address: Base address + 18h offset

Bit	7	6	5	4	3	2	1	0
Read	0			ONACK	ANACK	INIT	TTYPE	ISO_7816E
Write	▨							
Reset	0	0	0	0	0	0	0	0

UARTx_C7816 field descriptions

Field	Description
7–5 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
4 ONACK	Generate NACK on Overflow When this field is set, the receiver automatically generates a NACK response if a receive buffer overrun occurs, as indicated by S1[OR]. In many systems, this results in the transmitter resending the packet that overflowed until the retransmit threshold for that transmitter is reached. A NACK is generated only if TTYPE=0. This field operates independently of ANACK. See . Overrun NACK considerations 0 The received data does not generate a NACK when the receipt of the data results in an overflow event. 1 If the receiver buffer overflows, a NACK is automatically sent on a received character.
3 ANACK	Generate NACK on Error When this field is set, the receiver automatically generates a NACK response if a parity error occurs or if INIT is set and an invalid initial character is detected. A NACK is generated only if TTYPE = 0. If ANACK is set, the UART attempts to retransmit the data indefinitely. To stop retransmission attempts, clear C2[TE] or ISO_7816E and do not set until S1[TC] sets C2[TE] again. 0 No NACK is automatically generated. 1 A NACK is automatically generated if a parity error is detected or if an invalid initial character is detected.
2 INIT	Detect Initial Character When this field is set, all received characters are searched for a valid initial character. If an invalid initial character is identified, and ANACK is set, a NACK is sent. All received data is discarded and error flags blocked (S1[NF], S1[OR], S1[FE], S1[PF], IS7816[WT], IS7816[CWT], IS7816[BWT], IS7816[GTV]) until a valid initial character is detected. Upon detecting a valid initial character, the configuration values S2[MSBF], C3[TXINV], and S2[RXINV] are automatically updated to reflect the initial character that was received. The actual INIT data value is not stored in the receive buffer. Additionally, upon detection of a valid initial character, IS7816[INITD] is set and an interrupt issued as programmed by IE7816[INITDE]. When a valid initial character is detected, INIT is automatically cleared. This Initial Character Detect feature is supported only in T = 0 protocol mode. 0 Normal operating mode. Receiver does not seek to identify initial character. 1 Receiver searches for initial character.
1 TTYPE	Transfer Type Indicates the transfer protocol being used. See ISO-7816 / smartcard support for more details. 0 T = 0 per the ISO-7816 specification. 1 T = 1 per the ISO-7816 specification.
0 ISO_7816E	ISO-7816 Functionality Enabled Indicates that the UART is operating according to the ISO-7816 protocol. **NOTE:** This field must be modified only when no transmit or receive is occurring. If this field is changed during a data transfer, the data being transmitted or received may be transferred incorrectly. 0 ISO-7816 functionality is turned off/not enabled. 1 ISO-7816 functionality is turned on/enabled.

53.3.24 UART 7816 Interrupt Enable Register (UARTx_IE7816)

The IE7816 register controls which flags result in an interrupt being issued. This register is specific to 7816 functionality, the corresponding flags that drive the interrupts are not asserted when 7816E is not set/enabled. However, these flags may remain set if they are asserted while 7816E was set and not subsequently cleared. This register may be read or written to at any time.

Address: Base address + 19h offset

Bit	7	6	5	4	3	2	1	0
Read	WTE	CWTE	BWTE	INITDE	0	GTVE	TXTE	RXTE
Write					▨			
Reset	0	0	0	0	0	0	0	0

UARTx_IE7816 field descriptions

Field	Description
7 WTE	Wait Timer Interrupt Enable 0 The assertion of IS7816[WT] does not result in the generation of an interrupt. 1 The assertion of IS7816[WT] results in the generation of an interrupt.
6 CWTE	Character Wait Timer Interrupt Enable 0 The assertion of IS7816[CWT] does not result in the generation of an interrupt. 1 The assertion of IS7816[CWT] results in the generation of an interrupt.
5 BWTE	Block Wait Timer Interrupt Enable 0 The assertion of IS7816[BWT] does not result in the generation of an interrupt. 1 The assertion of IS7816[BWT] results in the generation of an interrupt.
4 INITDE	Initial Character Detected Interrupt Enable 0 The assertion of IS7816[INITD] does not result in the generation of an interrupt. 1 The assertion of IS7816[INITD] results in the generation of an interrupt.
3 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
2 GTVE	Guard Timer Violated Interrupt Enable 0 The assertion of IS7816[GTV] does not result in the generation of an interrupt. 1 The assertion of IS7816[GTV] results in the generation of an interrupt.
1 TXTE	Transmit Threshold Exceeded Interrupt Enable 0 The assertion of IS7816[TXT] does not result in the generation of an interrupt. 1 The assertion of IS7816[TXT] results in the generation of an interrupt.
0 RXTE	Receive Threshold Exceeded Interrupt Enable 0 The assertion of IS7816[RXT] does not result in the generation of an interrupt. 1 The assertion of IS7816[RXT] results in the generation of an interrupt.

53.3.25 UART 7816 Interrupt Status Register (UARTx_IS7816)

The IS7816 register provides a mechanism to read and clear the interrupt flags. All flags/ interrupts are cleared by writing a 1 to the field location. Writing a 0 has no effect. All bits are "sticky", meaning they indicate that only the flag condition that occurred since the last time the bit was cleared, not that the condition currently exists. The status flags are set regardless of whether the corresponding field in the IE7816 is set or cleared. The IE7816 controls only if an interrupt is issued to the host processor. This register is specific to 7816 functionality and the values in this register have no affect on UART operation and should be ignored if 7816E is not set/enabled. This register may be read or written at anytime.

Address: Base address + 1Ah offset

Bit	7	6	5	4	3	2	1	0
Read	WT	CWT	BWT	INITD	0	GTV	TXT	RXT
Write	w1c	w1c	w1c	w1c		w1c	w1c	w1c
Reset	0	0	0	0	0	0	0	0

UARTx_IS7816 field descriptions

Field	Description
7 WT	Wait Timer Interrupt Indicates that the wait time, the time between the leading edge of a character being transmitted and the leading edge of the next response character, has exceeded the programmed value. This flag asserts only when C7816[TTYPE] = 0. This interrupt is cleared by writing 1. 0 Wait time (WT) has not been violated. 1 Wait time (WT) has been violated.
6 CWT	Character Wait Timer Interrupt Indicates that the character wait time, the time between the leading edges of two consecutive characters in a block, has exceeded the programmed value. This flag asserts only when C7816[TTYPE] = 1. This interrupt is cleared by writing 1. 0 Character wait time (CWT) has not been violated. 1 Character wait time (CWT) has been violated.
5 BWT	Block Wait Timer Interrupt Indicates that the block wait time, the time between the leading edge of first received character of a block and the leading edge of the last character the previously transmitted block, has exceeded the programmed value. This flag asserts only when C7816[TTYPE] = 1.This interrupt is cleared by writing 1. 0 Block wait time (BWT) has not been violated. 1 Block wait time (BWT) has been violated.
4 INITD	Initial Character Detected Interrupt Indicates that a valid initial character is received. This interrupt is cleared by writing 1.

Table continues on the next page...

UARTx_IS7816 field descriptions (continued)

Field	Description
	0 A valid initial character has not been received. 1 A valid initial character has been received.
3 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
2 GTV	Guard Timer Violated Interrupt Indicates that one or more of the character guard time, block guard time, or guard time are violated. This interrupt is cleared by writing 1. 0 A guard time (GT, CGT, or BGT) has not been violated. 1 A guard time (GT, CGT, or BGT) has been violated.
1 TXT	Transmit Threshold Exceeded Interrupt Indicates that the transmit NACK threshold has been exceeded as indicated by ET7816[TXTHRESHOLD]. Regardless of whether this flag is set, the UART continues to retransmit indefinitely. This flag asserts only when C7816[TTYPE] = 0. If 7816E is cleared/disabled, ANACK is cleared/disabled, C2[TE] is cleared/disabled, C7816[TTYPE] = 1, or packet is transferred without receiving a NACK, the internal NACK detection counter is cleared and the count restarts from zero on the next received NACK. This interrupt is cleared by writing 1. 0 The number of retries and corresponding NACKS does not exceed the value in ET7816[TXTHRESHOLD]. 1 The number of retries and corresponding NACKS exceeds the value in ET7816[TXTHRESHOLD].
0 RXT	Receive Threshold Exceeded Interrupt Indicates that there are more than ET7816[RXTHRESHOLD] consecutive NACKS generated in response to parity errors on received data. This flag requires ANACK to be set. Additionally, this flag asserts only when C7816[TTYPE] = 0. Clearing this field also resets the counter keeping track of consecutive NACKS. The UART will continue to attempt to receive data regardless of whether this flag is set. If 7816E is cleared/disabled, RE is cleared/disabled, C7816[TTYPE] = 1, or packet is received without needing to issue a NACK, the internal NACK detection counter is cleared and the count restarts from zero on the next transmitted NACK. This interrupt is cleared by writing 1. 0 The number of consecutive NACKS generated as a result of parity errors and buffer overruns is less than or equal to the value in ET7816[RXTHRESHOLD]. 1 The number of consecutive NACKS generated as a result of parity errors and buffer overruns is greater than the value in ET7816[RXTHRESHOLD].

53.3.26 UART 7816 Wait Parameter Register (UARTx_WP7816T0)

The WP7816 register contains constants used in the generation of various wait timer counters. To save register space, this register is used differently when C7816[TTYPE] = 0 and C7816[TTYPE] = 1. This register may be read at any time. This register must be written to only when C7816[ISO_7816E] is not set.

Address: Base address + 1Bh offset

Bit	7	6	5	4	3	2	1	0
Read Write	WI							
Reset	0	0	0	0	1	0	1	0

UARTx_WP7816T0 field descriptions

Field	Description
WI	Wait Time Integer (C7816[TTYPE] = 0) Used to calculate the value used for the WT counter. It represents a value between 1 and 255. The value of zero is not valid. This value is used only when C7816[TTYPE] = 0. See Wait time and guard time parameters.

53.3.27 UART 7816 Wait Parameter Register (UARTx_WP7816T1)

The WP7816 register contains constants used in the generation of various wait timer counters. To save register space, this register is used differently when C7816[TTYPE] = 0 and C7816[TTYPE] = 1. This register may be read at any time. This register must be written to only when C7816[ISO_7816E] is not set.

Address: Base address + 1Bh offset

Bit	7	6	5	4	3	2	1	0
Read Write	CWI				BWI			
Reset	0	0	0	0	1	0	1	0

UARTx_WP7816T1 field descriptions

Field	Description
7–4 CWI	Character Wait Time Integer (C7816[TTYPE] = 1) Used to calculate the value used for the CWT counter. It represents a value between 0 and 15. This value is used only when C7816[TTYPE] = 1. See Wait time and guard time parameters .
BWI	Block Wait Time Integer(C7816[TTYPE] = 1) Used to calculate the value used for the BWT counter. It represent a value between 0 and 15. This value is used only when C7816[TTYPE] = 1. See Wait time and guard time parameters .

53.3.28 UART 7816 Wait N Register (UARTx_WN7816)

The WN7816 register contains a parameter that is used in the calculation of the guard time counter. This register may be read at any time. This register must be written to only when C7816[ISO_7816E] is not set.

Address: Base address + 1Ch offset

Bit	7	6	5	4	3	2	1	0
Read Write	GTN							
Reset	0	0	0	0	0	0	0	0

UARTx_WN7816 field descriptions

Field	Description
GTN	Guard Band N Defines a parameter used in the calculation of GT, CGT, and BGT counters. The value represents an integer number between 0 and 255. See Wait time and guard time parameters .

53.3.29 UART 7816 Wait FD Register (UARTx_WF7816)

The WF7816 contains parameters that are used in the generation of various counters including GT, CGT, BGT, WT, and BWT. This register may be read at any time. This register must be written to only when C7816[ISO_7816E] is not set.

Address: Base address + 1Dh offset

Bit	7	6	5	4	3	2	1	0
Read Write	GTFD							
Reset	0	0	0	0	0	0	0	1

UARTx_WF7816 field descriptions

Field	Description
GTFD	FD Multiplier Used as another multiplier in the calculation of WT and BWT. This value represents a number between 1 and 255. The value of 0 is invalid. This value is not used in baud rate generation. See Wait time and guard time parameters and Baud rate generation .

53.3.30 UART 7816 Error Threshold Register (UARTx_ET7816)

The ET7816 register contains fields that determine the number of NACKs that must be received or transmitted before the host processor is notified. This register may be read at anytime. This register must be written to only when C7816[ISO_7816E] is not set.

Address: Base address + 1Eh offset

Bit	7	6	5	4	3	2	1	0
Read Write	TXTHRESHOLD				RXTHRESHOLD			
Reset	0	0	0	0	0	0	0	0

UARTx_ET7816 field descriptions

Field	Description
7–4 TXTHRESHOLD	Transmit NACK Threshold The value written to this field indicates the maximum number of failed attempts (NACKs) a transmitted character can have before the host processor is notified. This field is meaningful only when

Table continues on the next page...

UARTx_ET7816 field descriptions (continued)

Field	Description
	C7816[TTYPE] = 0 and C7816[ANACK] = 1. The value read from this field represents the number of consecutive NACKs that have been received since the last successful transmission. This counter saturates at 4'hF and does not wrap around. Regardless of how many NACKs that are received, the UART continues to retransmit indefinitely. This flag only asserts when C7816[TTYPE] = 0. For additional information see the IS7816[TXT] field description. 0 TXT asserts on the first NACK that is received. 1 TXT asserts on the second NACK that is received.
RXTHRESHOLD	Receive NACK Threshold The value written to this field indicates the maximum number of consecutive NACKs generated as a result of a parity error or receiver buffer overruns before the host processor is notified. After the counter exceeds that value in the field, the IS7816[RXT] is asserted. This field is meaningful only when C7816[TTYPE] = 0. The value read from this field represents the number of consecutive NACKs that have been transmitted since the last successful reception. This counter saturates at 4'hF and does not wrap around. Regardless of the number of NACKs sent, the UART continues to receive valid packets indefinitely. For additional information, see IS7816[RXT] field description.

53.3.31 UART 7816 Transmit Length Register (UARTx_TL7816)

The TL7816 register is used to indicate the number of characters contained in the block being transmitted. This register is used only when C7816[TTYPE] = 1. This register may be read at anytime. This register must be written only when C2[TE] is not enabled.

Address: Base address + 1Fh offset

Bit	7	6	5	4	3	2	1	0
Read Write	TLEN							
Reset	0	0	0	0	0	0	0	0

UARTx_TL7816 field descriptions

Field	Description
TLEN	Transmit Length This value plus four indicates the number of characters contained in the block being transmitted. This register is automatically decremented by 1 for each character in the information field portion of the block. Additionally, this register is automatically decremented by 1 for the first character of a CRC in the epilogue field. Therefore, this register must be programmed with the number of bytes in the data packet if an LRC is being transmitted, and the number of bytes + 1 if a CRC is being transmitted. This register is not decremented for characters that are assumed to be part of the Prologue field, that is, the first three characters transmitted in a block, or the LRC or last CRC character in the Epilogue field, that is, the last character transmitted. This field must be programed or adjusted only when C2[TE] is cleared.

53.4 Functional description

This section provides a complete functional description of the UART block.

The UART allows full duplex, asynchronous, NRZ serial communication between the CPU and remote devices, including other CPUs. The UART transmitter and receiver operate independently, although they use the same baud rate generator. The CPU monitors the status of the UART, writes the data to be transmitted, and processes received data.

53.4.1 Transmitter

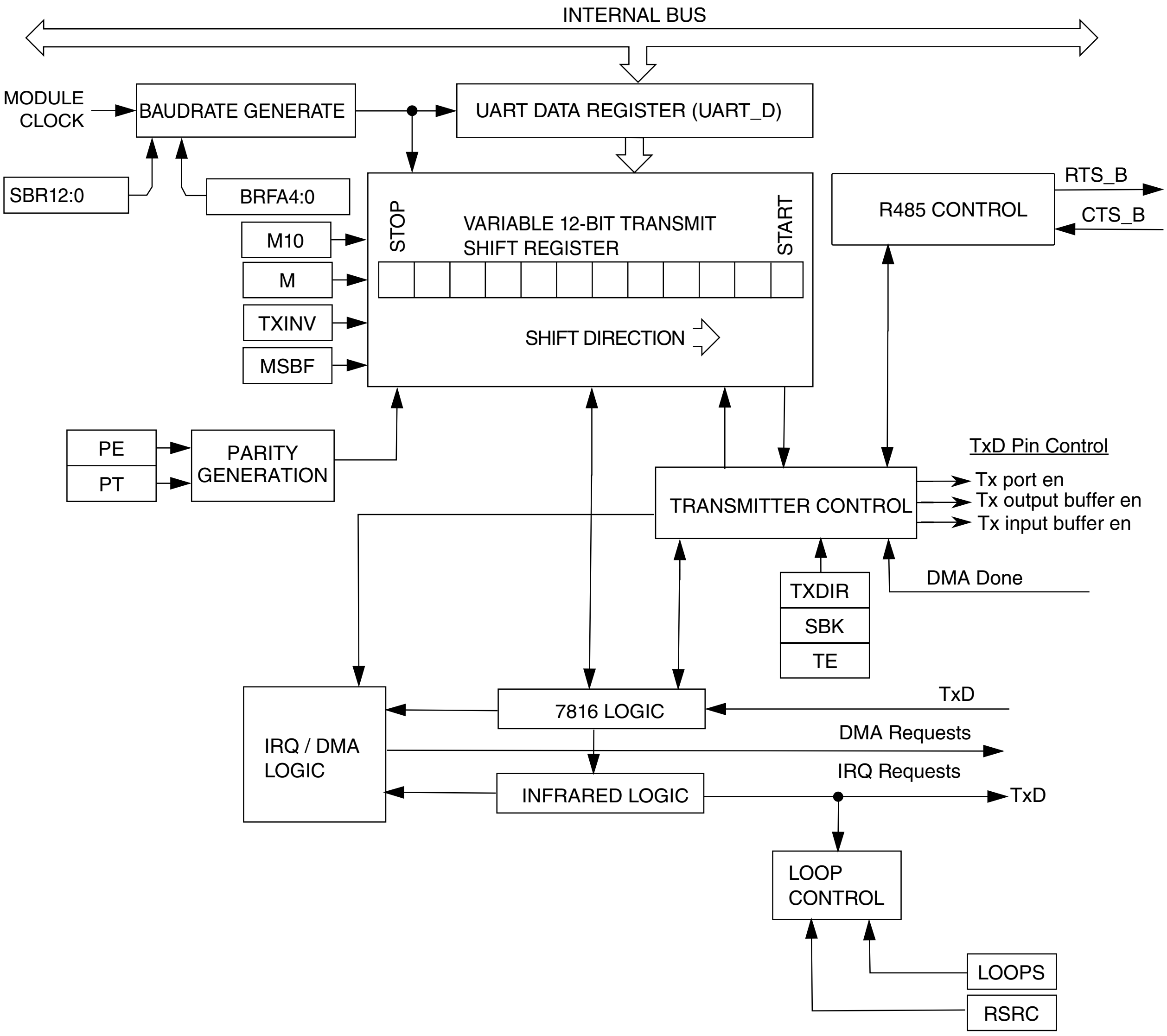


Figure 53-1. Transmitter Block Diagram

53.4.1.1 Transmitter character length

The UART transmitter can accommodate either 8, 9, or 10-bit data characters. The state of the C1[M] and C1[PE] bits and the C4[M10] bit determine the length of data characters. When transmitting 9-bit data, bit C3[T8] is the ninth bit (bit 8).

53.4.1.2 Transmission bit order

When S2[MSBF] is set, the UART automatically transmits the MSB of the data word as the first bit after the start bit. Similarly, the LSB of the data word is transmitted immediately preceding the parity bit, or the stop bit if parity is not enabled. All necessary bit ordering is handled automatically by the module. Therefore, the format of the data written to D for transmission is completely independent of the S2[MSBF] setting.

53.4.1.3 Character transmission

To transmit data, the MCU writes the data bits to the UART transmit buffer using UART data registers C3[T8] and D. Data in the transmit buffer is then transferred to the transmitter shift register as needed. The transmit shift register then shifts a frame out through the transmit data output signal after it has prefaced it with any required start and stop bits. The UART data registers, C3[T8] and D, provide access to the transmit buffer structure.

The UART also sets a flag, the transmit data register empty flag S1[TDRE], and generates an interrupt or DMA request (C5[TDMAS]) whenever the number of datawords in the transmit buffer is equal to or less than the value indicated by TWFIFO[TXWATER]. The transmit driver routine may respond to this flag by writing additional datawords to the transmit buffer using C3[T8]/D as space permits.

See Application information for specific programing sequences.

Setting C2[TE] automatically loads the transmit shift register with the following preamble:

- 10 logic 1s if C1[M] = 0
- 11 logic 1s if C1[M] = 1 and C4[M10] = 0
- 12 logic 1s if C1[M] = 1, C4[M10] = 1, C1[PE] = 1

After the preamble shifts out, control logic transfers the data from the D register into the transmit shift register. The transmitter automatically transmits the correct start bit and stop bit before and after the dataword.

When C7816[ISO_7816E] = 1, setting C2[TE] does not result in a preamble being generated. The transmitter starts transmitting as soon as the corresponding guard time expires. When C7816[TTYPE] = 0, the value in GT is used. When C7816[TTYPE] = 1, the value in BGT is used, because C2[TE] will remain asserted until the end of the block transfer. C2[TE] is automatically cleared when C7816[TTYPE] = 1 and the block being transmitted has completed. When C7816[TTYPE] = 0, the transmitter listens for a NACK indication. If no NACK is received, it is assumed that the character was correctly received. If a NACK is received, the transmitter resends the data, assuming that the number of retries for that character, that is, the number of NACKs received, is less than or equal to the value in ET7816[TXTHRESHOLD].

Hardware supports odd or even parity. When parity is enabled, the bit immediately preceding the stop bit is the parity bit.

When the transmit shift register is not transmitting a frame, the transmit data output signal goes to the idle condition, logic 1. If at any time software clears C2[TE], the transmitter enable signal goes low and the transmit signal goes idle.

If the software clears C2[TE] while a transmission is in progress, the character in the transmit shift register continues to shift out, provided S1[TC] was cleared during the data write sequence. To clear S1[TC], the S1 register must be read followed by a write to D register.

If S1[TC] is cleared during character transmission and C2[TE] is cleared, the transmission enable signal is deasserted at the completion of the current frame. Following this, the transmit data out signal enters the idle state even if there is data pending in the UART transmit data buffer. To ensure that all the data written in the FIFO is transmitted on the link before clearing C2[TE], wait for S1[TC] to set. Alternatively, the same can be achieved by setting TWFIFO[TXWATER] to 0x0 and waiting for S1[TDRE] to set.

53.4.1.4 Transmitting break characters

Setting C2[SBK] loads the transmit shift register with a break character. A break character contains all logic 0s and has no start, stop, or parity bit. Break character length depends on C1[M], C1[PE], S2[BRK13] and C4[M10]. See the following table.

Table 53-3. Transmit break character length

S2[BRK13]	C1[M]	C4[M10]	C1[PE]	Bits transmitted
0	0	—	—	10
0	1	1	0	11
0	1	1	1	12
1	0	—	—	13
1	1	—	—	14

As long as C2[SBK] is set, the transmitter logic continuously loads break characters into the transmit shift register. After the software clears C2[SBK], the shift register finishes transmitting the last break character and then transmits at least one logic 1. The automatic logic 1 at the end of a break character guarantees the recognition of the start bit of the next character. Break bits are not supported when C7816[ISO_7816E] is set/enabled.

NOTE

When queuing a break character, it will be transmitted following the completion of the data value currently being shifted out from the shift register. This means that, if data is queued in the data buffer to be transmitted, the break character preempts that queued data. The queued data is then transmitted after the break character is complete.

53.4.1.5 Idle characters

An idle character contains all logic 1s and has no start, stop, or parity bit. Idle character length depends on C1[M], C1[PE] and C4[M10]. The preamble is a synchronizing idle character that begins the first transmission initiated after setting C2[TE]. When C7816[ISO_7816E] is set/enabled, idle characters are not sent or detected. When data is not being transmitted, the data I/O line is in an inactive state.

If C2[TE] is cleared during a transmission, the transmit data output signal becomes idle after completion of the transmission in progress. Clearing and then setting C2[TE] during a transmission queues an idle character to be sent after the dataword currently being transmitted.

Note

When queuing an idle character, the idle character will be transmitted following the completion of the data value currently being shifted out from the shift register. This means that if data is queued in the data buffer to be transmitted, the idle character preempts that queued data. The queued data is then transmitted after the idle character is complete.

If C2[TE] is cleared and the transmission is completed, the UART is not the master of the TXD pin.

53.4.1.6 Hardware flow control

The transmitter supports hardware flow control by gating the transmission with the value of CTS. If the clear-to-send operation is enabled, the character is transmitted when CTS is asserted. If CTS is deasserted in the middle of a transmission with characters remaining in the receiver data buffer, the character in the shift register is sent and TXD remains in the mark state until CTS is reasserted.

If the clear-to-send operation is disabled, the transmitter ignores the state of CTS. Also, if the transmitter is forced to send a continuous low condition because it is sending a break character, the transmitter ignores the state of CTS regardless of whether the clear-to-send operation is enabled.

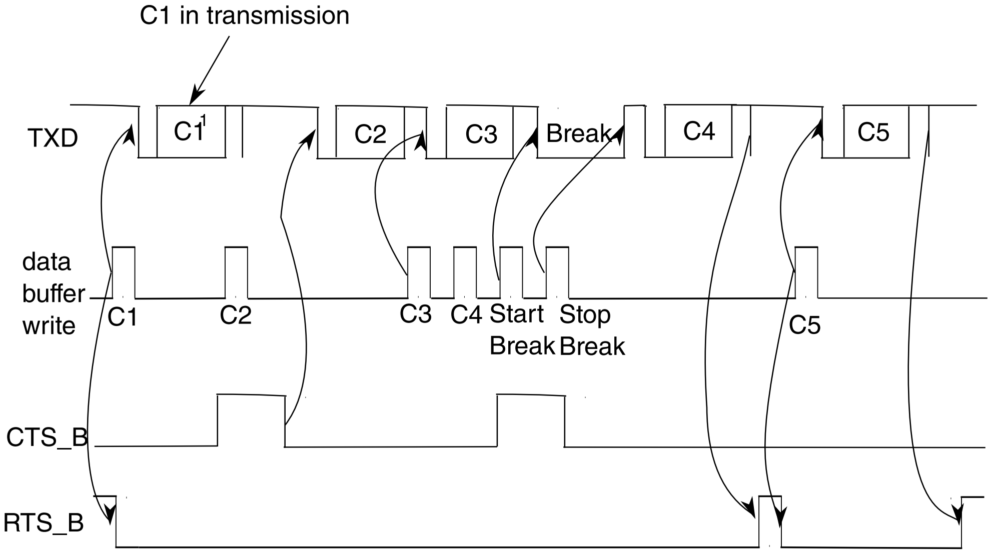
The transmitter's CTS signal can also be enabled even if the same UART receiver's RTS signal is disabled.

53.4.1.7 Transceiver driver enable

The transmitter can use RTS as an enable signal for the driver of an external transceiver. See Transceiver driver enable using RTS for details. If the request-to-send operation is enabled, when a character is placed into an empty transmitter data buffer, RTS asserts one bit time before the start bit is transmitted. RTS remains asserted for the whole time that the transmitter data buffer has any characters. RTS deasserts one bit time after all characters in the transmitter data buffer and shift register are completely sent, including the last stop bit. Transmitting a break character also asserts RTS, with the same assertion and deassertion timing as having a character in the transmitter data buffer.

The transmitter's RTS signal asserts only when the transmitter is enabled. However, the transmitter's RTS signal is unaffected by its CTS signal. RTS will remain asserted until the transfer is completed, even if the transmitter is disabled mid-way through a data transfer.

The following figure shows the functional timing information for the transmitter. Along with the actual character itself, TXD shows the start bit. The stop bit is also indicated, with a dashed line if necessary.



1. Cn = transmit characters

Figure 53-2. Transmitter RTS and CTS timing diagram

53.4.2 Receiver

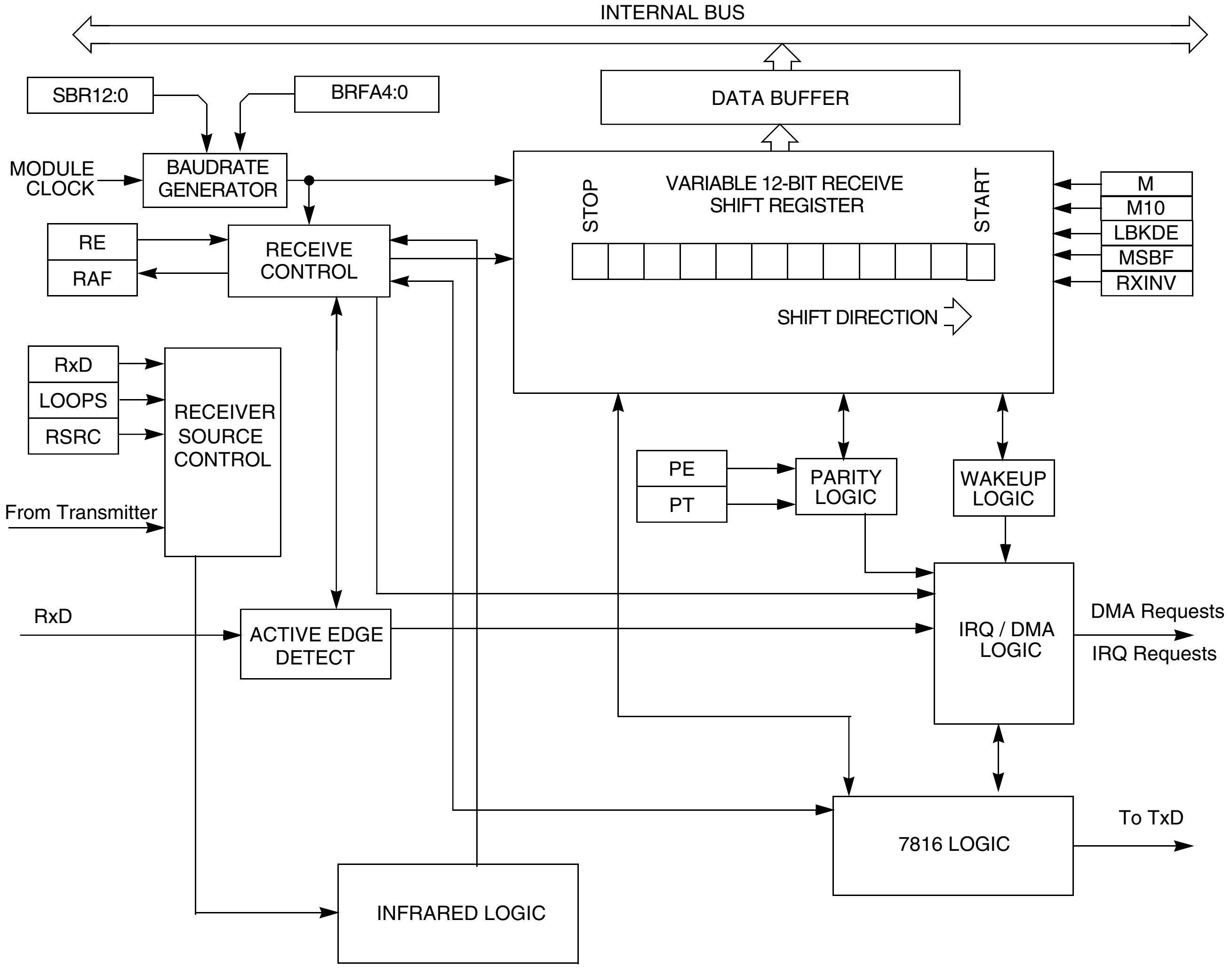


Figure 53-3. UART receiver block diagram

53.4.2.1 Receiver character length

The UART receiver can accommodate 8-, 9-, or 10-bit data characters. The states of C1[M], C1[PE] and C4[M10] determine the length of data characters. When receiving 9 or 10-bit data, C3[R8] is the ninth bit (bit 8).

53.4.2.2 Receiver bit ordering

When S2[MSBF] is set, the receiver operates such that the first bit received after the start bit is the MSB of the dataword. Similarly, the bit received immediately preceding the parity bit, or the stop bit if parity is not enabled, is treated as the LSB for the dataword. All necessary bit ordering is handled automatically by the module. Therefore, the format of the data read from receive data buffer is completely independent of S2[MSBF].

53.4.2.3 Character reception

During UART reception, the receive shift register shifts a frame in from the unsynchronized receiver input signal. After a complete frame shifts into the receive shift register, the data portion of the frame transfers to the UART receive buffer. Additionally, the noise and parity error flags that are calculated during the receive process are also captured in the UART receive buffer. The receive data buffer is accessible via the D and C3[T8] registers. Additional received information flags regarding the receive dataword can be read in ED register. S1[RDRF] is set if the number of resulting datawords in the receive buffer is equal to or greater than the number indicated by RWFIFO[RXWATER]. If the C2[RIE] is also set, RDRF generates an RDRF interrupt request. Alternatively, by programming C5[RDMAS], a DMA request can be generated.

When C7816[ISO_7816E] is set/enabled and C7816[TTYPE] = 0, character reception operates slightly differently. Upon receipt of the parity bit, the validity of the parity bit is checked. If C7816[ANACK] is set and the parity check fails, or if INIT and the received character is not a valid initial character, then a NACK is sent by the receiver. If the number of consecutive receive errors exceeds the threshold set by ET7816[RXTHRESHOLD], then IS7816[RXT] is set and an interrupt generated if IE7816[RXTE] is set. If an error is detected due to parity or an invalid initial character, the data is not transferred from the receive shift register to the receive buffer. Instead, the data is overwritten by the next incoming data.

When the C7816[ISO_7816E] is set/enabled, C7816[ONACK] is set/enabled, and the received character results in the receive buffer overflowing, a NACK is issued by the receiver. Additionally, S1[OR] is set and an interrupt is issued if required, and the data in the shift register is discarded.

53.4.2.4 Data sampling

The receiver samples the unsynchronized receiver input signal at the RT clock rate. The RT clock is an internal signal with a frequency 16 times the baud rate. To adjust for baud rate mismatch, the RT clock (see the following figure) is re-synchronized:

- After every start bit.
- After the receiver detects a data bit change from logic 1 to logic 0 (after the majority of data bit samples at RT8, RT9, and RT10 returns a valid logic 1 and the majority of the next RT8, RT9, and RT10 samples returns a valid logic 0).

To locate the start bit, data recovery logic does an asynchronous search for a logic 0 preceded by three logic 1s. When the falling edge of a possible start bit occurs, the RT clock begins to count to 16.

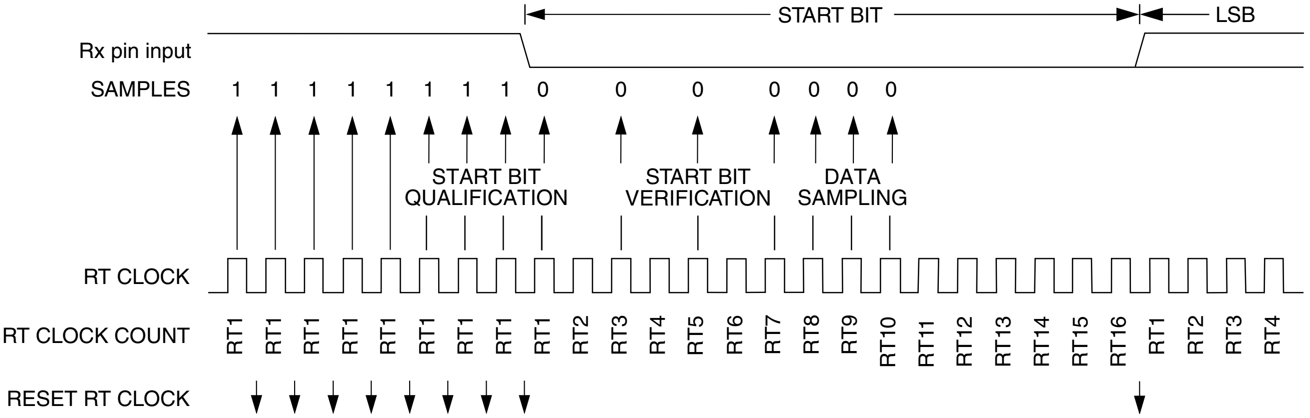


Figure 53-4. Receiver data sampling

To verify the start bit and to detect noise, data recovery logic takes samples at RT3, RT5, and RT7 when C7816[ISO_7816E] is cleared/disabled and RT8, RT9 and RT10 when C7816[ISO_7816E] is set/enabled. The following table summarizes the results of the start bit verification samples.

Table 53-4. Start bit verification

RT3, RT5, and RT7 samples RT8, RT9, RT10 samples when 7816E	Start bit verification	Noise flag
000	Yes	0
001	Yes	1
010	Yes	1
011	No	0
100	Yes	1
101	No	0
110	No	0
111	No	0

If start bit verification is not successful, the RT clock is reset and a new search for a start bit begins.

To determine the value of a data bit and to detect noise, recovery logic takes samples at RT8, RT9, and RT10. The following table summarizes the results of the data bit samples.

Table 53-5. Data bit recovery

RT8, RT9, and RT10 samples	Data bit determination	Noise flag
000	0	0
001	0	1
010	0	1
011	1	1
100	0	1
101	1	1
110	1	1
111	1	0

Note

The RT8, RT9, and RT10 samples do not affect start bit verification. If any or all of the RT8, RT9, and RT10 start bit samples are logic 1s following a successful start bit verification, the noise flag (S1[NF]) is set and the receiver assumes that the bit is a start bit (logic 0). With the exception of when C7816[ISO_7816E] is set/enabled, where the values of RT8, RT9 and RT10 exclusively determine if a start bit exists.

To verify a stop bit and to detect noise, recovery logic takes samples at RT8, RT9, and RT10. The following table summarizes the results of the stop bit samples. In the event that C7816[ISO_7816E] is set/enabled and C7816[TTYPE] = 0, verification of a stop bit does not take place. Rather, starting with RT8 the receiver transmits a NACK as programmed until time RT9 of the following time period. Framing Error detection is not supported when C7816[ISO_7816E] is set/enabled.

Table 53-6. Stop bit recovery

RT8, RT9, and RT10 samples	Framing error flag	Noise flag
000	1	0
001	1	1
010	1	1
011	0	1
100	1	1
101	0	1
110	0	1
111	0	0

In the following figure, the verification samples RT3 and RT5 determine that the first low detected was noise and not the beginning of a start bit. In this example C7816[ISO_7816E] = 0. The RT clock is reset and the start bit search begins again. The noise flag is not set because the noise occurred before the start bit was found.

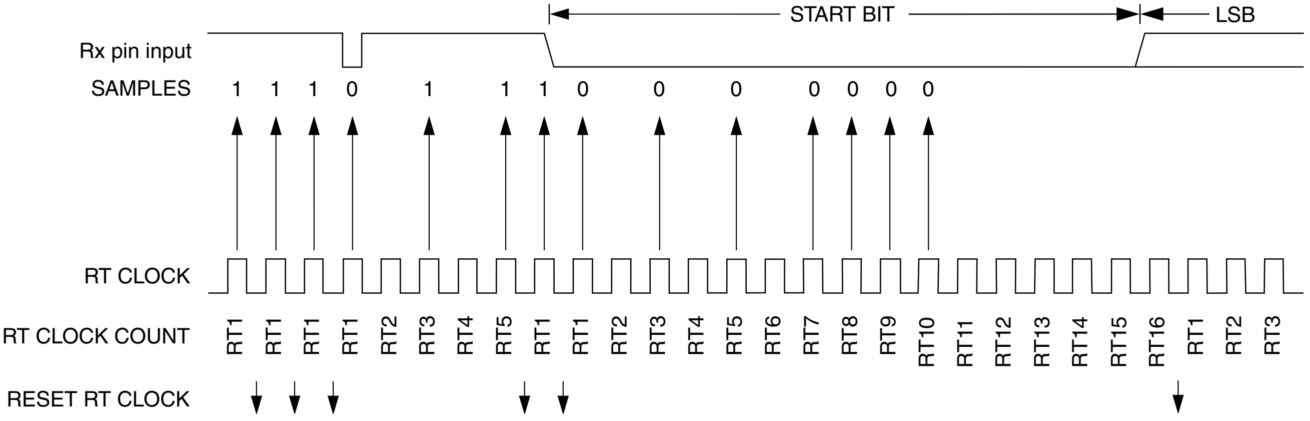

START BIT
LSB
Rx pin input
SAMPLES
1 1 1 0 1 1 1 0 0 0 0 0 0 0
RT CLOCK
RT CLOCK COUNT
RT1 RT1 RT1 RT1 RT2 RT3 RT4 RT5 RT1 RT1 RT2 RT3 RT4 RT5 RT6 RT7 RT8 RT9 RT10 RT11 RT12 RT13 RT14 RT15 RT16 RT1 RT2 RT3
RESET RT CLOCK

Figure 53-5. Start bit search example 1 (C7816[ISO_7816E] = 0)

In the following figure, verification sample at RT3 is high. In this example C7816[ISO_7816E] = 0. The RT3 sample sets the noise flag. Although the perceived bit time is misaligned, the data samples RT8, RT9, and RT10 are within the bit time and data recovery is successful.

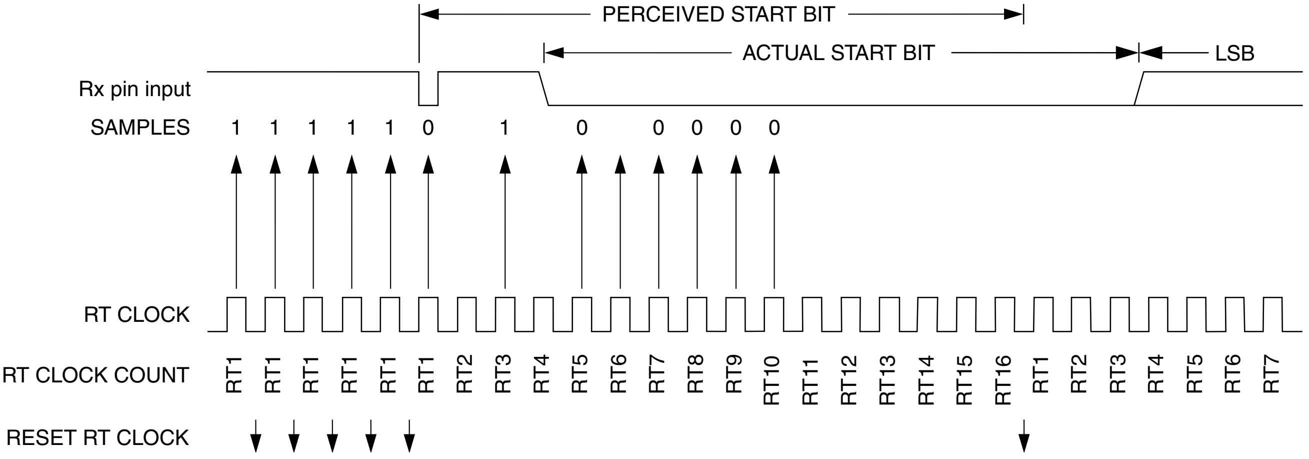

PERCEIVED START BIT
ACTUAL START BIT
LSB
Rx pin input
SAMPLES
1 1 1 1 1 0 1 0 0 0 0 0
RT CLOCK
RT CLOCK COUNT
RT1 RT1 RT1 RT1 RT1 RT1 RT2 RT3 RT4 RT5 RT6 RT7 RT8 RT9 RT10 RT11 RT12 RT13 RT14 RT15 RT16 RT1 RT2 RT3 RT4 RT5 RT6 RT7
RESET RT CLOCK

Figure 53-6. Start bit search example 2 (C7816[ISO_7816E] = 0)

In the following figure, a large burst of noise is perceived as the beginning of a start bit, although the test sample at RT5 is high. In this example C7816[ISO_7816E] = 0. The RT5 sample sets the noise flag. Although this is a worst-case misalignment of perceived bit time, the data samples RT8, RT9, and RT10 are within the bit time and data recovery is successful.

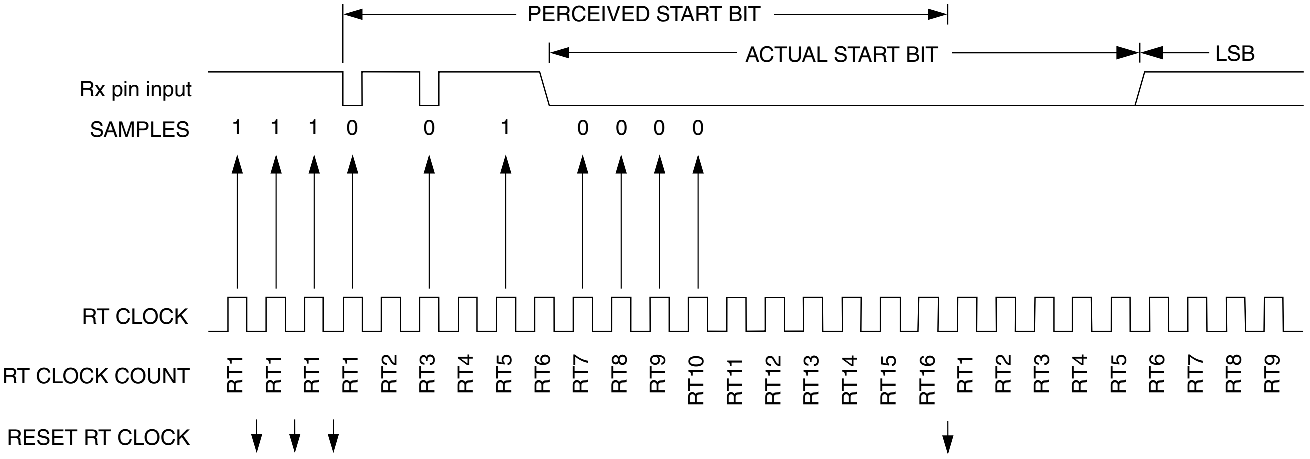

PERCEIVED START BIT
ACTUAL START BIT
LSB
Rx pin input
SAMPLES
1 1 1 0 0 1 0 0 0 0
RT CLOCK
RT CLOCK COUNT
RT1 RT1 RT1 RT1 RT2 RT3 RT4 RT5 RT6 RT7 RT8 RT9 RT10 RT11 RT12 RT13 RT14 RT15 RT16 RT1 RT2 RT3 RT4 RT5 RT6 RT7 RT8 RT9
RESET RT CLOCK

Figure 53-7. Start bit search example 3 (C7816[ISO_7816E] = 0)

The following figure shows the effect of noise early in the start bit time. In this example C7816[ISO_7816E] = 0. Although this noise does not affect proper synchronization with the start bit time, it does set the noise flag.

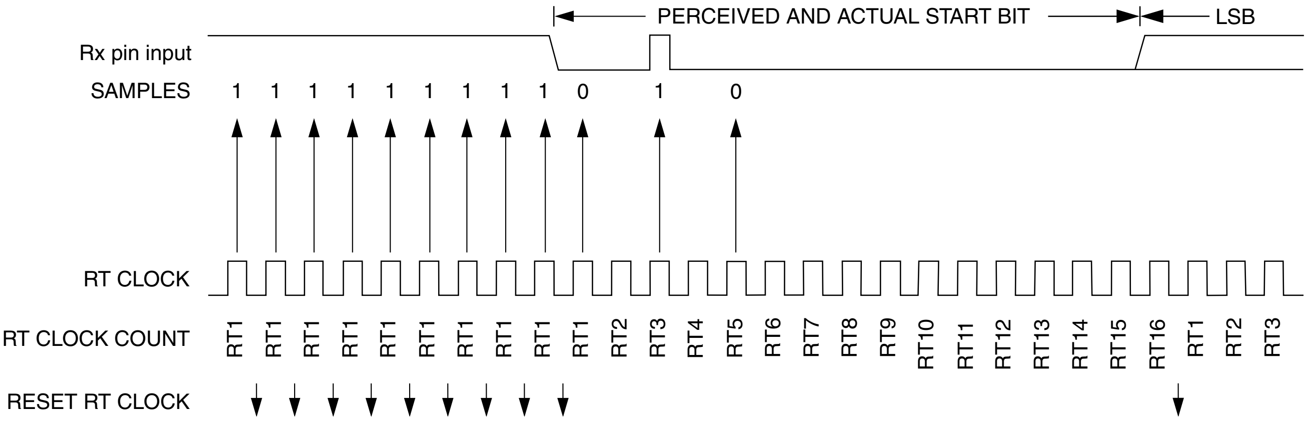

PERCEIVED AND ACTUAL START BIT
LSB
Rx pin input
SAMPLES
1 1 1 1 1 1 1 1 1 0 1 0
RT CLOCK
RT CLOCK COUNT
RT1 RT1 RT1 RT1 RT1 RT1 RT1 RT1 RT1 RT1 RT2 RT3 RT4 RT5 RT6 RT7 RT8 RT9 RT10 RT11 RT12 RT13 RT14 RT15 RT16 RT1 RT2 RT3
RESET RT CLOCK

Figure 53-8. Start bit search example 4 (C7816[ISO_7816E] = 0)

The following figure shows a burst of noise near the beginning of the start bit that resets the RT clock. In this example C7816[ISO_7816E] = 0. The sample after the reset is low but is not preceded by three high samples that would qualify as a falling edge. Depending on the timing of the start bit search and on the data, the frame may be missed entirely or it may set the framing error flag.

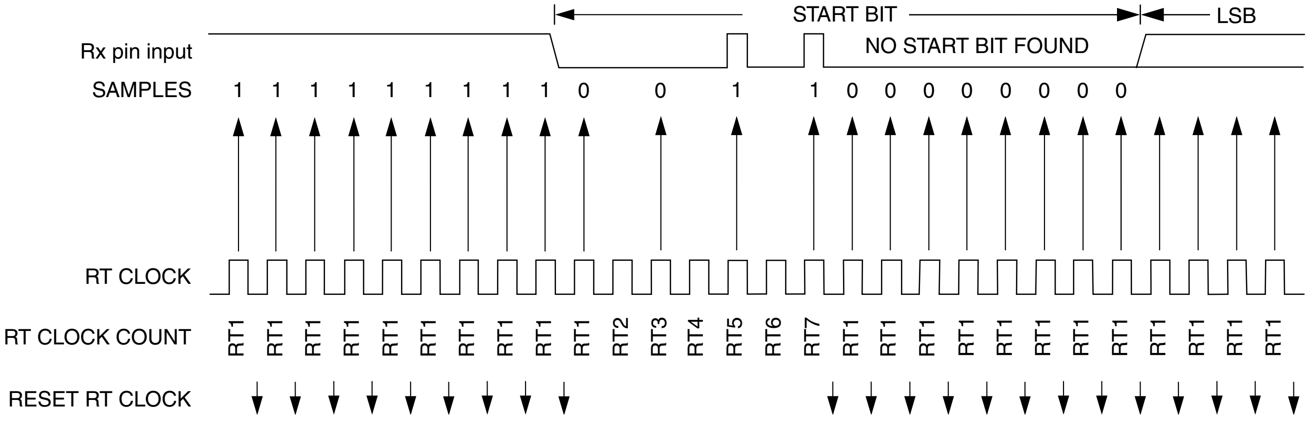


Figure 53-9. Start bit search example 5 (C7816[ISO_7816E] = 0)

In the following figure, a noise burst makes the majority of data samples RT8, RT9, and RT10 high. In this example C7816[ISO_7816E] = 0. This sets the noise flag but does not reset the RT clock. In start bits only, the RT8, RT9, and RT10 data samples are ignored. In this example, if C7816[ISO_7816E] = 1 then a start bit would not have been detected at all since at least two of the three samples (RT8, RT9, RT10) were high.

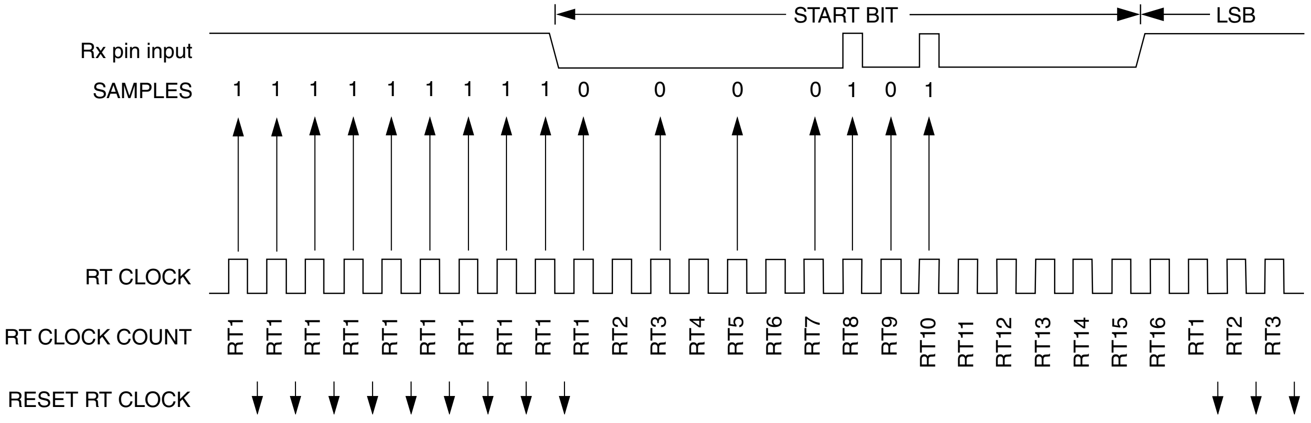

START BIT
LSB
Rx pin input
SAMPLES
1 1 1 1 1 1 1 1 1 0 0 0 0 1 0 1
RT CLOCK
RT CLOCK COUNT
RT1 RT1 RT1 RT1 RT1 RT1 RT1 RT1 RT1 RT1 RT2 RT3 RT4 RT5 RT6 RT7 RT8 RT9 RT10 RT11 RT12 RT13 RT14 RT15 RT16 RT1 RT2 RT3
RESET RT CLOCK

Figure 53-10. Start bit search example 6

53.4.2.5 Framing errors

If the data recovery logic does not detect a logic 1 where the stop bit should be in an incoming frame, it sets the framing error flag, S1[FE], if S2[LBKDE] is disabled. When S2[LBKDE] is disabled, a break character also sets the S1[FE] because a break character has no stop bit. S1[FE] is set at the same time that received data is placed in the receive data buffer. Framing errors are not supported when C7816[ISO7816E] is set/enabled. However, if S1[FE] is set, data will not be received when C7816[ISO7816E] is set.

53.4.2.6 Receiving break characters

The UART recognizes a break character when a start bit is followed by eight, nine, or ten logic 0 data bits and a logic 0 where the stop bit should be. Receiving a break character has these effects on UART registers:

- Sets the framing error flag, S1[FE].
- Writes an all 0 dataword to the data buffer, which may cause S1[RDRF] to set, depending on the watermark and number of values in the data buffer.
- May set the overrun flag, S1[OR], noise flag, S1[NF], parity error flag, S1[PE], or the receiver active flag, S2[RAF].

The detection threshold for a break character can be adjusted when using an internal oscillator in a LIN system by setting S2[LBKDE]. The UART break character detection threshold depends on C1[M], C1[PE], S2[LBKDE] and C4[M10]. See the following table.

Table 53-7. Receive break character detection threshold

LBKDE	M	M10	PE	Threshold (bits)
0	0	—	—	10
0	1	0	—	11
0	1	1	1	12
1	0	—	—	11
1	1	—	—	12

While S2[LBKDE] is set, it will have these effects on the UART registers:

- Prevents S1[RDRF], S1[FE], S1[NF], and S1[PF] from being set. However, if they are already set, they will remain set.
- Sets the LIN break detect interrupt flag, S2[LBKDIF], if a LIN break character is received.

Break characters are not detected or supported when C7816[ISO_7816E] is set/enabled.

53.4.2.7 Hardware flow control

To support hardware flow control, the receiver can be programmed to automatically deassert and assert RTS.

- RTS remains asserted until the transfer is complete, even if the transmitter is disabled midway through a data transfer. See Transceiver driver enable using RTS for more details.
- If the receiver request-to-send functionality is enabled, the receiver automatically deasserts RTS if the number of characters in the receiver data register is equal to or greater than receiver data buffer's watermark, RWFIFO[RXWATER].
- The receiver asserts RTS when the number of characters in the receiver data register is less than the watermark. It is not affected if RDRF is asserted.
- Even if RTS is deasserted, the receiver continues to receive characters until the receiver data buffer is full or is overrun.
- If the receiver request-to-send functionality is disabled, the receiver RTS remains deasserted.

The following figure shows receiver hardware flow control functional timing. Along with the actual character itself, RXD shows the start bit. The stop bit can also indicated, with a dashed line, if necessary. The watermark is set to 2.

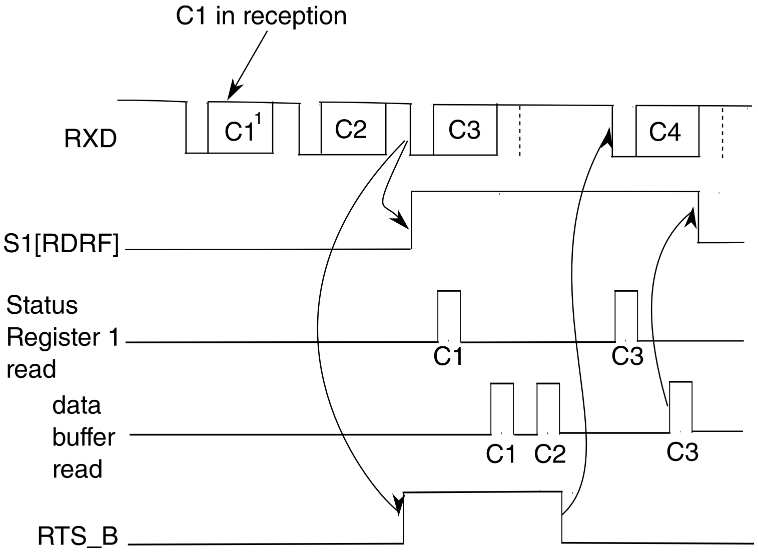


Figure 53-11. Receiver hardware flow control timing diagram

53.4.2.8 Infrared decoder

The infrared decoder converts the received character from the IrDA format to the NRZ format used by the receiver. It also has a 16-RT clock counter that filters noise and indicates when a 1 is received.

53.4.2.8.1 Start bit detection

When S2[RXINV] is cleared, the first rising edge of the received character corresponds to the start bit. The infrared decoder resets its counter. At this time, the receiver also begins its start bit detection process. After the start bit is detected, the receiver synchronizes its bit times to this start bit time. For the rest of the character reception, the infrared decoder's counter and the receiver's bit time counter count independently from each other.

53.4.2.8.2 Noise filtering

Any further rising edges detected during the first half of the infrared decoder counter are ignored by the decoder. Any pulses less than one RT clocks can be undetected by it regardless of whether it is seen in the first or second half of the count.

53.4.2.8.3 Low-bit detection

During the second half of the decoder count, a rising edge is decoded as a 0, which is sent to the receiver. The decoder counter is also reset.

53.4.2.8.4 High-bit detection

At 16-RT clocks after the previous rising edge, if a rising edge is not seen, then the decoder sends a 1 to the receiver.

If the next bit is a 0, which arrives late, then a low-bit is detected according to Low-bit detection. The value sent to the receiver is changed from 1 to a 0. Then, if a noise pulse occurs outside the receiver's bit time sampling period, then the delay of a 0 is not recorded as noise.

53.4.2.9 Baud rate tolerance

A transmitting device may be operating at a baud rate below or above the receiver baud rate. Accumulated bit time misalignment can cause one of the three stop bit data samples (RT8, RT9, and RT10) to fall outside the actual stop bit. A noise error will occur if the RT8, RT9, and RT10 samples are not all the same logical values. A framing error will occur if the receiver clock is misaligned in such a way that the majority of the RT8, RT9, and RT10 stop bit samples are a logic 0.

As the receiver samples an incoming frame, it resynchronizes the RT clock on any valid falling edge within the frame. Resynchronization within frames corrects a misalignment between transmitter bit times and receiver bit times.

53.4.2.9.1 Slow data tolerance

The following figure shows how much a slow received frame can be misaligned without causing a noise error or a framing error. The slow stop bit begins at RT8 instead of RT1 but arrives in time for the stop bit data samples at RT8, RT9, and RT10.

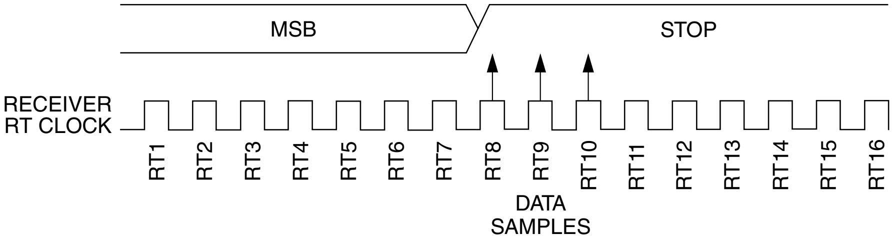


Figure 53-12. Slow data

For an 8-bit data character, data sampling of the stop bit takes the receiver 154 RT cycles (9 bit times × 16 RT cycles + 10 RT cycles).

With the misaligned character shown in the Figure 53-12, the receiver counts 154 RT cycles at the point when the count of the transmitting device is 147 RT cycles (9 bit times × 16 RT cycles + 3 RT cycles).

The maximum percent difference between the receiver count and the transmitter count of a slow 8-bit data character with no errors is:

$$((154 - 147) \div 154) \times 100 = 4.54\%$$

For a 9-bit data character, data sampling of the stop bit takes the receiver 170 RT cycles (10 bit times × 16 RT cycles + 10 RT cycles).

With the misaligned character shown in the Figure 53-12, the receiver counts 170 RT cycles at the point when the count of the transmitting device is 163 RT cycles (10 bit times × 16 RT cycles + 3 RT cycles).

The maximum percent difference between the receiver count and the transmitter count of a slow 9-bit character with no errors is:

$$((170 - 163) \div 170) \times 100 = 4.12\%$$

53.4.2.9.2 Fast data tolerance

The following figure shows how much a fast received frame can be misaligned. The fast stop bit ends at RT10 instead of RT16 but is still sampled at RT8, RT9, and RT10.

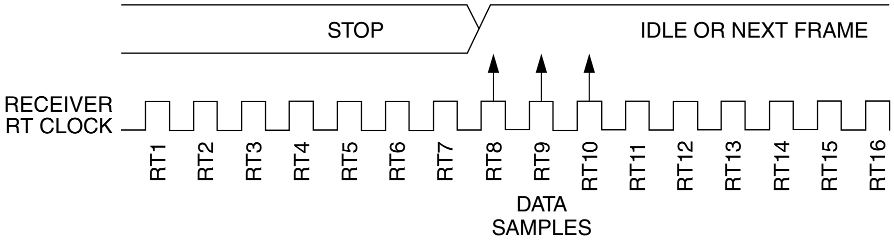


Figure 53-13. Fast data

For an 8-bit data character, data sampling of the stop bit takes the receiver 154 RT cycles (9 bit times × 16 RT cycles + 10 RT cycles).

With the misaligned character shown in the Figure 53-13, the receiver counts 154 RT cycles at the point when the count of the transmitting device is 160 RT cycles (10 bit times × 16 RT cycles).

The maximum percent difference between the receiver count and the transmitter count of a fast 8-bit character with no errors is:

$$((154 - 160) \div 154) \times 100 = 3.90\%$$

For a 9-bit data character, data sampling of the stop bit takes the receiver 170 RT cycles (10 bit times × 16 RT cycles + 10 RT cycles).

With the misaligned character shown in the Figure 53-13, the receiver counts 170 RT cycles at the point when the count of the transmitting device is 176 RT cycles (11 bit times × 16 RT cycles).

The maximum percent difference between the receiver count and the transmitter count of a fast 9-bit character with no errors is:

$$((170 - 176) \div 170) \times 100 = 3.53\%$$

53.4.2.10 Receiver wakeup

C1[WAKE] determines how the UART is brought out of the standby state to process an incoming message. C1[WAKE] enables either idle line wakeup or address mark wakeup.

Receiver wakeup is not supported when C7816[ISO_7816E] is set/enabled because multi-receiver systems are not allowed.

53.4.2.10.1 Idle input line wakeup (C1[WAKE] = 0)

In this wakeup method, an idle condition on the unsynchronized receiver input signal clears C2[RWU] and wakes the UART. The initial frame or frames of every message contain addressing information. All receivers evaluate the addressing information, and receivers for which the message is addressed process the frames that follow. Any receiver for which a message is not addressed can set its C2[RWU] and return to the standby state. C2[RWU] remains set and the receiver remains in standby until another idle character appears on the unsynchronized receiver input signal.

Idle line wakeup requires that messages be separated by at least one idle character and that no message contains idle characters.

When C2[RWU] is 1 and S2[RWUID] is 0, the idle character that wakes the receiver does not set S1[IDLE] or the receive data register full flag, S1[RDRF]. The receiver wakes and waits for the first data character of the next message which is stored in the receive data buffer. When S2[RWUID] and C2[RWU] are set and C1[WAKE] is cleared, any idle condition sets S1[IDLE] and generates an interrupt if enabled.

Idle input line wakeup is not supported when C7816[ISO_7816E] is set/enabled.

53.4.2.10.2 Address mark wakeup (C1[WAKE] = 1)

In this wakeup method, a logic 1 in the bit position immediately preceding the stop bit of a frame clears C2[RWU] and wakes the UART. A logic 1 in the bit position immediately preceeding the stop bit marks a frame as an address frame that contains addressing information. All receivers evaluate the addressing information, and the receivers for which the message is addressed process the frames that follow. Any receiver for which a message is not addressed can set its C2[RWU] and return to the standby state. C2[RWU] remains set and the receiver remains in standby until another address frame appears on the unsynchronized receiver input signal.

A logic 1 in the bit position immediately preceding the stop bit clears the receiver's C2[RWU] after the stop bit is received and places the received data into the receiver data buffer. Note that if Match Address operation is enabled i.e. C4[MAEN1] or C4[MAEN2] is set, then received frame is transferred to receive buffer only if the comparison matches.

Address mark wakeup allows messages to contain idle characters but requires that the bit position immediately preceding the stop bit be reserved for use in address frames.

If module is in standby mode and nothing triggers to wake the UART, no error flag is set even if an invalid error condition is detected on the receiving data line.

Address mark wakeup is not supported when C7816[ISO_7816E] is set/enabled.

53.4.2.10.3 Match address operation

Match address operation is enabled when C4[MAEN1] or C4[MAEN2] is set. In this function, a frame received by the RX pin with a logic 1 in the bit position of the address mark is considered an address and is compared with the associated MA1 or MA2 register.

The frame is transferred to the receive buffer, and S1[RDRF] is set, only if the comparison matches. All subsequent frames received with a logic 0 in the bit position of the address mark are considered to be data associated with the address and are transferred to the receive data buffer. If no marked address match occurs, then no transfer is made to the receive data buffer, and all following frames with logic 0 in the bit position of the address mark are also discarded. If both C4[MAEN1] and C4[MAEN2] are negated, the receiver operates normally and all data received is transferred to the receive data buffer.

Match address operation functions in the same way for both MA1 and MA2 registers. Note that the position of the address mark is the same as the Parity Bit when parity is enabled for 8 bit and 9 bit data formats.

- If only one of C4[MAEN1] and C4[MAEN2] is asserted, a marked address is compared only with the associated match register and data is transferred to the receive data buffer only on a match.
- If C4[MAEN1] and C4[MAEN2] are asserted, a marked address is compared with both match registers and data is transferred only on a match with either register.

Address match operation is not supported when C7816[ISO_7816E] is set/enabled.

53.4.3 Baud rate generation

A 13-bit modulus counter and a 5-bit fractional fine-adjust counter in the baud rate generator derive the baud rate for both the receiver and the transmitter. The value from 1 to 8191 written to SBR[12:0] determines the module clock divisor. The SBR bits are in the UART baud rate registers, BDH and BDL. The baud rate clock is synchronized with the module clock and drives the receiver. The fractional fine-adjust counter adds fractional delays to the baud rate clock to allow fine trimming of the baud rate to match the system baud rate. The transmitter is driven by the baud rate clock divided by 16. The receiver has an acquisition rate of 16 samples per bit time.

Baud rate generation is subject to two sources of error:

- Integer division of the module clock may not give the exact target frequency. This error can be reduced with the fine-adjust counter.
- Synchronization with the module clock can cause phase shift.

The Table 53-8 lists the available baud divisor fine adjust values.

UART baud rate = UART module clock / (16 × (SBR[12:0] + BRFD))

The following table lists some examples of achieving target baud rates with a module clock frequency of 10.2 MHz, with and without fractional fine adjustment.

Table 53-8. Baud rates (example: module clock = 10.2 MHz)

Bits SBR (decimal)	Bits BRFA	BRFD value	Receiver clock (Hz)	Transmitter clock (Hz)	Target Baud rate	Error (%)
17	00000	0	600,000.0	37,500.0	38,400	2.3
16	10011	19/32=0.59375	614,689.3	38,418.08	38,400	0.047
33	00000	0	309,090.9	19,318.2	19,200	0.62
33	00110	6/32=0.1875	307,344.6	19,209.04	19,200	0.047
66	00000	0	154,545.5	9659.1	9600	0.62
133	00000	0	76,691.7	4793.2	4800	0.14
266	00000	0	38,345.9	2396.6	2400	0.14
531	00000	0	19,209.0	1200.6	1200	0.11
1062	00000	0	9604.5	600.3	600	0.05
2125	00000	0	4800.0	300.0	300	0.00
4250	00000	0	2400.0	150.0	150	0.00
5795	00000	0	1760.1	110.0	110	0.00

Table 53-9. Baud rate fine adjust

BRFA	Baud Rate Fractional Divisor (BRFD)
0 0 0 0 0	0/32 = 0
0 0 0 0 1	1/32 = 0.03125
0 0 0 1 0	2/32 = 0.0625
0 0 0 1 1	3/32 = 0.09375
0 0 1 0 0	4/32 = 0.125
0 0 1 0 1	5/32 = 0.15625
0 0 1 1 0	6/32 = 0.1875
0 0 1 1 1	7/32 = 0.21875
0 1 0 0 0	8/32 = 0.25
0 1 0 0 1	9/32 = 0.28125
0 1 0 1 0	10/32 = 0.3125
0 1 0 1 1	11/32 = 0.34375
0 1 1 0 0	12/32 = 0.375
0 1 1 0 1	13/32 = 0.40625
0 1 1 1 0	14/32 = 0.4375
0 1 1 1 1	15/32 = 0.46875
1 0 0 0 0	16/32 = 0.5
1 0 0 0 1	17/32 = 0.53125
1 0 0 1 0	18/32 = 0.5625

Table continues on the next page...

Table 53-9. Baud rate fine adjust (continued)

BRFA	Baud Rate Fractional Divisor (BRFD)
1 0 0 1 1	19/32 = 0.59375
1 0 1 0 0	20/32 = 0.625
1 0 1 0 1	21/32 = 0.65625
1 0 1 1 0	22/32 = 0.6875
1 0 1 1 1	23/32 = 0.71875
1 1 0 0 0	24/32 = 0.75
1 1 0 0 1	25/32 = 0.78125
1 1 0 1 0	26/32 = 0.8125
1 1 0 1 1	27/32 = 0.84375
1 1 1 0 0	28/32 = 0.875
1 1 1 0 1	29/32 = 0.90625
1 1 1 1 0	30/32 = 0.9375
1 1 1 1 1	31/32 = 0.96875

53.4.4 Data format (non ISO-7816)

Each data character is contained in a frame that includes a start bit and a stop bit. The rest of the data format depends upon C1[M], C1[PE], S2[MSBF] and C4[M10].

53.4.4.1 Eight-bit configuration

Clearing C1[M] configures the UART for 8-bit data characters, that is, eight bits are memory mapped in D. A frame with eight data bits has a total of 10 bits. The most significant bit of the eight data bits can be used as an address mark to wake the receiver. If the most significant bit is used in this way, then it serves as an address or data indication, leaving the remaining seven bits as actual data. When C1[PE] is set, the eighth data bit is automatically calculated as the parity bit. See the following table.

Table 53-10. Configuration of 8-bit data format

UART_C1[PE]	Start bit	Data bits	Address bits	Parity bits	Stop bit
0	1	8	0	0	1
0	1	7	1	0	1
1	1	7	0	1	1

53.4.4.2 Nine-bit configuration

When C1[M] is set and C4[M10] is cleared, the UART is configured for 9-bit data characters. If C1[PE] is enabled, the ninth bit is either C3[T8/R8] or the internally generated parity bit. This results in a frame consisting of a total of 11 bits. In the event that the ninth data bit is selected to be C3[T8], it will remain unchanged after transmission and can be used repeatedly without rewriting it, unless the value needs to be changed. This feature may be useful when the ninth data bit is being used as an address mark.

When C1[M] and C4[M10] are set, the UART is configured for 9-bit data characters, but the frame consists of a total of 12 bits. The 12 bits include the start and stop bits, the 9 data character bits, and a tenth internal data bit. Note that if C4[M10] is set, C1[PE] must also be set. In this case, the tenth bit is the internally generated parity bit. The ninth bit can either be used as an address mark or a ninth data bit.

See the following table.

Table 53-11. Configuration of 9-bit data formats

C1[PE]	UC1[M]	C1[M10]	Start bit	Data bits	Address bits	Parity bits	Stop bit
0	0	0	See Eight-bit configuration				
0	0	1	Invalid configuration				
0	1	0	1	9	0	0	1
0	1	0	1	8	1	0	1
0	1	1	Invalid Configuration				
1	0	0	See Eight-bit configuration				
1	0	1	Invalid Configuration				
1	1	0	1	8	0	1	1
1	1	1	1	9	0	1	1
1	1	1	1	8	1	1	1

Note

Unless in 9-bit mode with M10 set, do not use address mark wakeup with parity enabled.

53.4.4.3 Timing examples

Timing examples of these configurations in the NRZ mark/space data format are illustrated in the following figures. The timing examples show all of the configurations in the following sub-sections along with the LSB and MSB first variations.

53.4.4.3.1 Eight-bit format with parity disabled

The most significant bit can be used for address mark wakeup.

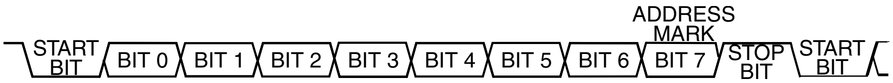


Figure 53-14. Eight bits of data with LSB first

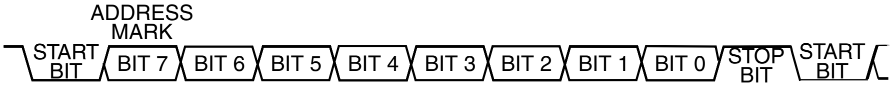


Figure 53-15. Eight bits of data with MSB first

53.4.4.3.2 Eight-bit format with parity enabled

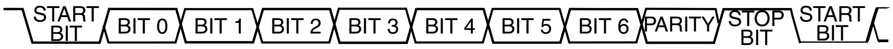


Figure 53-16. Seven bits of data with LSB first and parity

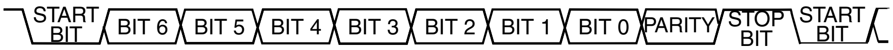


Figure 53-17. Seven bits of data with MSB first and parity

53.4.4.3.3 Nine-bit format with parity disabled

The most significant bit can be used for address mark wakeup.

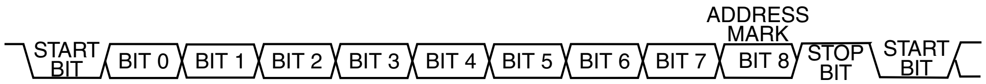


Figure 53-18. Nine bits of data with LSB first

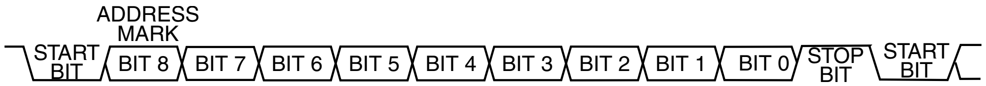


Figure 53-19. Nine bits of data with MSB first

53.4.4.3.4 Nine-bit format with parity enabled

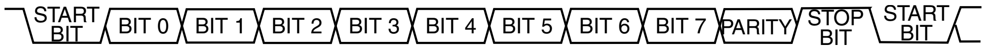


Figure 53-20. Eight bits of data with LSB first and parity

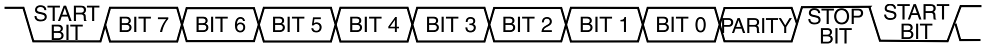


Figure 53-21. Eight bits of data with MSB first and parity

53.4.4.3.5 Non-memory mapped tenth bit for parity

The most significant memory-mapped bit can be used for address mark wakeup.

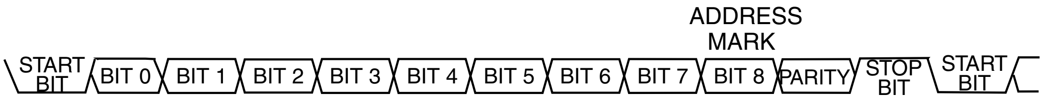


Figure 53-22. Nine bits of data with LSB first and parity

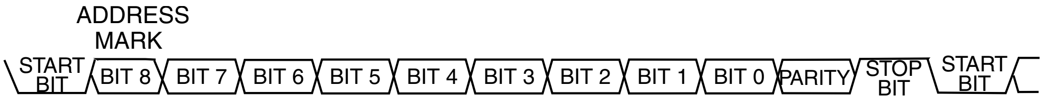


Figure 53-23. Nine bits of data with MSB first and parity

53.4.5 Single-wire operation

Normally, the UART uses two pins for transmitting and receiving. In single wire operation, the RXD pin is disconnected from the UART and the UART implements a half-duplex serial connection. The UART uses the TXD pin for both receiving and transmitting.

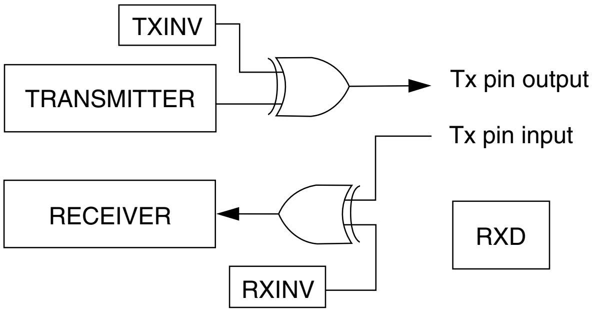


Figure 53-24. Single-wire operation (C1[LOOPS] = 1, C1[RSRC] = 1)

Enable single wire operation by setting C1[LOOPS] and the receiver source field, C1[RSRC]. Setting C1[LOOPS] disables the path from the unsynchronized receiver input signal to the receiver. Setting C1[RSRC] connects the receiver input to the output of the

TXD pin driver. Both the transmitter and receiver must be enabled (C2[TE] = 1 and C2[RE] = 1). When C7816[ISO_7816EN] is set, it is not required that both C2[TE] and C2[RE] are set.

53.4.6 Loop operation

In loop operation, the transmitter output goes to the receiver input. The unsynchronized receiver input signal is disconnected from the UART.

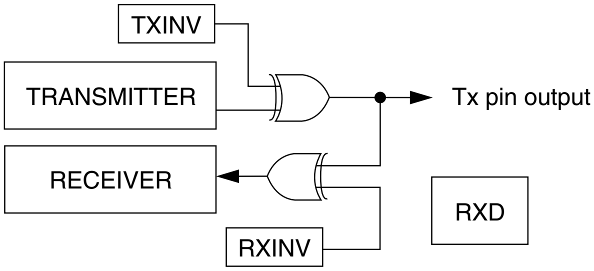


Figure 53-25. Loop operation (C1[LOOPS] = 1, C1[RSRC] = 0)

Enable loop operation by setting C1[LOOPS] and clearing C1[RSRC]. Setting C1[LOOPS] disables the path from the unsynchronized receiver input signal to the receiver. Clearing C1[RSRC] connects the transmitter output to the receiver input. Both the transmitter and receiver must be enabled (C2[TE] = 1 and C2[RE] = 1). When C7816[ISO_7816EN] is set, it is not required that both C2[TE] and C2[RE] are set.

53.4.7 ISO-7816/smartcard support

The UART provides mechanisms to support the ISO-7816 protocol that is commonly used to interface with smartcards. The ISO-7816 protocol is an NRZ, single wire, half-duplex interface. The TxD pin is used in open-drain mode because the data signal is used for both transmitting and receiving. There are multiple subprotocols within the ISO-7816 standard. The UART supports both T = 0 and T = 1 protocols. The module also provides for automated initial character detection and configuration, which allows for support of both direct convention and inverse convention data formats. A variety of interrupts specific to 7816 are provided in addition to the general interrupts to assist software. Additionally, the module is able to provide automated NACK responses and has programmed automated retransmission of failed packets. An assortment of programmable timeouts and guard band times are also supported.

The term elemental time unit (ETU) is frequently used in the context of ISO-7816. This concept is used to relate the frequency that the system (UART) is running at and the frequency that data is being transmitted and received. One ETU represents the time it takes to transmit or receive a single bit. For example, a standard 7816 packet, excluding any guard time or NACK elements is 10 ETUs (start bit, 8 data bits, and a parity bit). Guard times and wait times are also measured in ETUs.,

NOTE

> The ISO-7816 specification may have certain configuration options that are reserved. To maintain maximum flexibility to support future 7816 enhancements or devices that may not strictly conform to the specification, the UART does not prevent those options being used. Further, the UART may provide configuration options that exceed the flexibility of options explicitly allowed by the 7816 specification. Failure to correctly configure the UART may result in unexpected behavior or incompatibility with the ISO-7816 specification.

53.4.7.1 Initial characters

In ISO-7816 with T = 0 mode, the UART can be configured to use C7816[INIT] to detect the next valid initial character, referred to by the ISO-7816 specifically as a TS character. When the initial character is detected, the UART provides the host processor with an interrupt if IE7816[INITDE] is set. Additionally, the UART will alter S2[MSBF], C3[TXINV], and S2[RXINV] automatically, based on the initial character. The corresponding initial character and resulting register settings are listed in the following table.

Table 53-12. Initial character automated settings

Initial character (bit 1-10)	Initial character (hex)	MSBF	TXINV	RXINV
LHHL LLL LLH inverse convention	3F	1	1	1
LHHL HHH LLH direct convention	3B	0	0	0

S2[MSBF], C3[TXINV], and S2[RXINV] must be reset to their default values before C7816[INIT] is set. Once C7816[INIT] is set, the receiver searches all received data for the first valid initial character. Detecting a Direct Convention Initial Character will cause no change to S2[MSBF], C3[TXINV], and S2[RXINV], while detecting an Inverse Convention Initial Character will cause these fields to set automatically. All data received, which is not a valid initial character, is ignored and all flags resulting from the invalid data are blocked from asserting. If C7816[ANACK] is set, a NACK is returned for invalid received initial characters and an RXT interrupt is generated as programmed.

53.4.7.2 Protocol T = 0

When T = 0 protocol is selected, a relatively complex error detection scheme is used. Data characters are formatted as illustrated in the following figure. This scheme is also used for answer to reset and Peripheral Pin Select (PPS) formats.

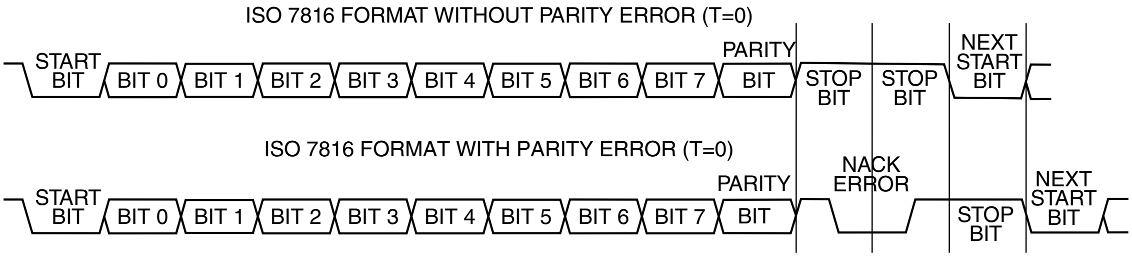


Figure 53-26. ISO-7816 T = 0 data format

As with other protocols supported by the UART, the data character includes a start bit. However, in this case, there are two stop bits rather than the typical single stop bit. In addition to a standard even parity check, the receiver has the ability to generate and return a NACK during the second half of the first stop bit period. The NACK must be at least one time period (ETU) in length and no more than two time periods (ETU) in length. The transmitter must wait for at least two time units (ETU) after detection of the error signal before attempting to retransmit the character.

It is assumed that the UART and the device (smartcard) know in advance which device is receiving and which is transmitting. No special mechanism is supplied by the UART to control receive and transmit in the mode other than C2[TE] and C2[RE]. Initial Character Detect feature is also supported in this mode.

53.4.7.3 Protocol T = 1

When T = 1 protocol is selected, the NACK error detection scheme is not used. Rather, the parity bit is used on a character basis and a CRC or LRC is used on the block basis, that is, for each group of characters. In this mode, the data format allows for a single stop bit although additional inactive bit periods may be present between the stop bit and the next start bit. Data characters are formatted as illustrated in the following figure.

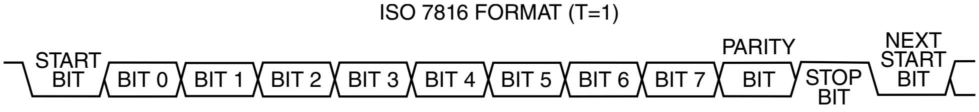


Figure 53-27. ISO 7816 T=1 data format

The smallest data unit that is transferred is a block. A block is made up of several data characters and may vary in size depending on the block type. The UART does not provide a mechanism to decode the block type. As part of the block, an LRC or CRC is included. The UART does not calculate the CRC or LRC for transmitted blocks, nor does it verify the validity of the CRC or LRC for received blocks. The 7816 protocol requires that the initiator and the smartcard (device) takes alternate turns in transmitting and receiving blocks. When the UART detects that the last character in a block has been transmitted it will automatically clear C2[TE], C3[TXDIR] and enter receive mode. Therefore, the software must program the transmit buffer with the next data to be transmitted and then enable C2[TE] and set C3[TXDIR], once the software has determined that the last character of the received block has been received. The UART detects that the last character of the transmit block has been sent when TL7816[TLEN] = 0 and four additional characters have been sent. The four additional characters are made up of three prior to TL7816[TLEN] decrementing (prologue) and one after TL7816[TLEN] = 0, the final character of the epilogue.

53.4.7.4 Wait time and guard time parameters

The ISO-7816 specification defines several wait time and guard time parameters. The UART allows for flexible configuration and violation detection of these settings. On reset, the wait time (IS7816[WT]) defaults to 9600 ETUs and guard time (GT) to 12 ETUs. These values are controlled by parameters in the WP7816, WN7816, and WF7816 registers. Additionally, the value of C7816[TTYPE] also factors into the calculation. The formulae used to calculate the number ETUs for each wait time and guard time value are shown in Table 53-13.

Wait time (WT) is defined as the maximum allowable time between the leading edge of a character transmitted by the smartcard device and the leading edge of the previous character that was transmitted by the UART or the device. Similarly, character wait time (CWT) is defined as the maximum allowable time between the leading edge of two characters within the same block. Block wait time (BWT) is defined as the maximum time between the leading edge character of the last block received by the smartcard device and the leading edge of the first character transmitted by the smartcard device.

Guard time (GT) is defined as the minimum allowable time between the leading edge of two consecutive characters. Character guard time (CGT) is the minimum allowable time between the leading edges of two consecutive characters in the same direction, that is, transmission or reception. Block guard time (BGT) is the minimum allowable time between the leading edges of two consecutive characters in opposite directions, that is, transmission then reception or reception then transmission.

The GT and WT counters reset whenever C7816[TTYPE] = 1 or C7816[ISO_7816E] = 0 or a new dataword start bit has been received or transmitted as specified by the counter descriptions. The CWT, CGT, BWT, BGT counters reset whenever C7816[TTYPE] = 0 or C7816[ISO_7816E] = 0 or a new dataword start bit is received or transmitted as specified by the counter descriptions. When C7816[TTYPE] = 1, some of the counter values require an assumption regarding the first data transferred when the UART first starts. This assumption is required when the 7816E is disabled, when transition from C7816[TTYPE] = 0 to C7816[TTYPE] = 1 or when coming out of reset. In this case, it is assumed that the previous non-existent transfer was a received transfer.

The UART will automatically handle GT, CGT, and BGT such that the UART will not send a packet before the corresponding guard time expiring.

Table 53-13. Wait and guard time calculations

Parameter	Reset value [ETU]	C7816[TTYPE] = 0 [ETU]	C7816[TTYPE] = 1 [ETU]
Wait time (WT)	9600	((WI + 1) × 960 × (GTFD + 1)) - 1	Not used
Character wait time (CWT)	Not used	Not used	$11 + 2^{(CWI - 1)}$
Block wait time (BWT)	Not used	Not used	$10 + 2^{BWI} \times 960 \times (GTFD + 1)$
Guard time (GT)	12	**GTN not equal to 255** 12 + GTN **GTN equal to 255** 12	Not used
Character guard time (CGT)	Not used	Not used	**GTN not equal to 255** 12 + GTN **GTN equal to 255** 11
Block guard time (BGT)	Not used	Not used	22

53.4.7.5 Baud rate generation

The value in WF7816[GTFD] does not impact the clock frequency. SBR and BRFD are used to generate the clock frequency. This clock frequency is used by the UART only and is not seen by the smartcard device. The transmitter clocks operates at 1/16 the frequency of the receive clock so that the receiver is able to sample the received value 16 times during the ETU.

53.4.7.6 UART restrictions in ISO-7816 operation

Due to the flexibility of the UART module, there are several features and interrupts that are not supported while running in ISO-7816 mode. These restrictions are documented within the register field definitions.

53.4.8 Infrared interface

The UART provides the capability of transmitting narrow pulses to an IR LED and receiving narrow pulses and transforming them to serial bits, which are sent to the UART. The IrDA physical layer specification defines a half-duplex infrared communication link for exchanging data. The full standard includes data rates up to 16 Mbits/s. This design covers data rates only between 2.4 kbits/s and 115.2 kbits/s.

The UART has an infrared transmit encoder and receive decoder. The UART transmits serial bits of data that are encoded by the infrared submodule to transmit a narrow pulse for every zero bit. No pulse is transmitted for every one bit. When receiving data, the IR pulses are detected using an IR photo diode and transformed to CMOS levels by the IR receive decoder, external from the MCU. The narrow pulses are then stretched by the infrared receive decoder to get back to a serial bit stream to be received by the UART. The polarity of transmitted pulses and expected receive pulses can be inverted so that a direct connection can be made to external IrDA transceiver modules that use active low pulses.

The infrared submodule receives its clock sources from the UART. One of these two clocks are selected in the infrared submodule to generate either 3/16, 1/16, 1/32, or 1/4 narrow pulses during transmission.

53.4.8.1 Infrared transmit encoder

The infrared transmit encoder converts serial bits of data from transmit shift register to the TXD signal. A narrow pulse is transmitted for a zero bit and no pulse for a one bit. The narrow pulse is sent in the middle of the bit with a duration of 1/32, 1/16, 3/16, or 1/4 of a bit time. A narrow high pulse is transmitted for a zero bit when C3[TXINV] is cleared, while a narrow low pulse is transmitted for a zero bit when C3[TXINV] is set.

53.4.8.2 Infrared receive decoder

The infrared receive block converts data from the RXD signal to the receive shift register. A narrow pulse is expected for each zero received and no pulse is expected for each one received. A narrow high pulse is expected for a zero bit when S2[RXINV] is cleared, while a narrow low pulse is expected for a zero bit when S2[RXINV] is set. This receive decoder meets the edge jitter requirement as defined by the IrDA serial infrared physical layer specification.

53.5 Reset

All registers reset to a particular value are indicated in Memory map and registers.

53.6 System level interrupt sources

There are several interrupt signals that are sent from the UART. The following table lists the interrupt sources generated by the UART. The local enables for the UART interrupt sources are described in this table. Details regarding the individual operation of each interrupt are contained under various sub-sections of Memory map and registers. However, RXEDGIF description also outlines additional details regarding the RXEDGIF interrupt because of its complexity of operation. Any of the UART interrupt requests listed in the table can be used to bring the CPU out of Wait mode.

Table 53-14. UART interrupt sources

Interrupt Source	Flag	Local enable	DMA select
Transmitter	TDRE	TIE	TDMAS = 0
Transmitter	TC	TCIE	-
Receiver	IDLE	ILIE	-
Receiver	RDRF	RIE	RDMAS = 0
Receiver	LBKDIF	LBKDIE	-
Receiver	RXEDGIF	RXEDGIE	-
Receiver	OR	ORIE	-
Receiver	NF	NEIE	-
Receiver	FE	FEIE	-
Receiver	PF	PEIE	-
Receiver	RXUF	RXUFE	-
Transmitter	TXOF	TXOFE	-
Receiver	WT	WTWE	-
Receiver	CWT	CWTE	-

Table continues on the next page...

Table 53-14. UART interrupt sources (continued)

Interrupt Source	Flag	Local enable	DMA select
Receiver	BWT	BWTE	-
Receiver	INITD	INITDE	-
Receiver	TXT	TXTE	-
Receiver	RXT	RXTE	-
Receiver	GTV	GTVE	-

53.6.1 RXEDGIF description

S2[RXEDGIF] is set when an active edge is detected on the RxD pin. Therefore, the active edge can be detected only when in two wire mode. A RXEDGIF interrupt is generated only when S2[RXEDGIF] is set. If RXEDGIE is not enabled before S2[RXEDGIF] is set, an interrupt is not generated.

53.6.1.1 RxD edge detect sensitivity

Edge sensitivity can be software programmed to be either falling or rising. The polarity of the edge sensitivity is selected using S2[RXINV]. To detect the falling edge, S2[RXINV] is programmed to 0. To detect the rising edge, S2[RXINV] is programmed to 1.

Synchronizing logic is used prior to detect edges. Prior to detecting an edge, the receive data on RxD input must be at the deasserted logic level. A falling edge is detected when the RxD input signal is seen as a logic 1 (the deasserted level) during one module clock cycle, and then a logic 0 (the asserted level) during the next cycle. A rising edge is detected when the input is seen as a logic 0 during one module clock cycle and then a logic 1 during the next cycle.

53.6.1.2 Clearing RXEDGIF interrupt request

Writing a logic 1 to S2[RXEDGIF] immediately clears the RXEDGIF interrupt request even if the RxD input remains asserted. S2[RXEDGIF] remains set if another active edge is detected on RxD while attempting to clear S2[RXEDGIF] by writing a 1 to it.

53.6.1.3 Exit from low-power modes

The receive input active edge detect circuit is still active on low power modes (Wait and Stop). An active edge on the receive input brings the CPU out of low power mode if the interrupt is not masked (S2[RXEDGIF] = 1).

53.7 DMA operation

In the transmitter, S1[TDRE] can be configured to assert a DMA transfer request. In the receiver, S1[RDRF], can be configured to assert a DMA transfer request. The following table shows the configuration field settings required to configure each flag for DMA operation.

Table 53-15. DMA configuration

Flag	Request enable bit	DMA select bit
TDRE	TIE = 1	TDMAS = 1
RDRF	RIE = 1	RDMAS = 1

When a flag is configured for a DMA request, its associated DMA request is asserted when the flag is set. When S1[RDRF] is configured as a DMA request, the clearing mechanism of reading S1, followed by reading D, does not clear the associated flag. The DMA request remains asserted until an indication is received that the DMA transactions are done. When this indication is received, the flag bit and the associated DMA request is cleared. If the DMA operation failed to remove the situation that caused the DMA request, another request is issued.

53.8 Application information

This section describes the UART application information.

53.8.1 Transmit/receive data buffer operation

The UART has independent receive and transmit buffers. The size of these buffers may vary depending on the implementation of the module. The implemented size of the buffers is a fixed constant via PFIFO[TXFIFOSIZE] and PFIFO[RXFIFOSIZE]. Additionally, legacy support is provided that allows for the FIFO structure to operate as a depth of one. This is the default/reset behavior of the module and can be adjusted using the PFIFO[RXFE] and PFIFO[TXFE] bits. Individual watermark levels are also provided for transmit and receive.

There are multiple ways to ensure that a data block, which is a set of characters, has completed transmission. These methods include:

1. Set TXFIFO[TXWATER] to 0. TDRE asserts when there is no further data in the transmit buffer. Alternatively the S1[TC] flag can be used to indicate when the transmit shift register is also empty.
2. Poll TCFIFO[TXCOUNT]. Assuming that only data for a data block has been put into the data buffer, when TCFIFO[TXCOUNT] = 0, all data has been transmitted or is in the process of transmission.
3. S1[TC] can be monitored. When S1[TC] asserts, it indicates that all data has been transmitted and there is no data currently being transmitted in the shift register.

53.8.2 ISO-7816 initialization sequence

This section outlines how to program the UART for ISO-7816 operation. Elements such as procedures to power up or power down the smartcard, and when to take those actions, are beyond the scope of this description. To set up the UART for ISO-7816 operation:

1. Select a baud rate. Write this value to the UART baud registers (BDH/L) to begin the baud rate generator. Remember that the baud rate generator is disabled when the baud rate is zero. Writing to the BDH has no effect without also writing to BDL. According to the 7816 specification the initial (default) baud rating setting should be Fi = 372 and Di = 1 and a maximum frequency of 5 MHz. In other words, the BDH, BDL, and C4 registers should be programmed such that the transmission frequency provided to the smartcard device must be 1/372th of the clock and must not exceed 5 MHz.
2. Write to set BDH[LBKDIE] = 0.
3. Write to C1 to configure word length, parity, and other configuration fields (LOOPS, RSRC) and set C1[M] = 1, C1[PE] = 1, and C1[PT] = 0.
4. Write to set S2[RWUID] = 0 and S2[LBKDE] = 0.
5. Write to set MODEM[RXRTSE] = 0, MODEM[TXRTSPOL] = 0, MODEM[TXRTSE] = 0, and MODEM[TXCTSE] = 0.

6. Write to set up interrupt enable fields desired (C3[ORIE], C3[NEIE], C3[PEIE], and C3[FEIE])
7. Write to set C4[MAEN1] = 0 and C4[MAEN2] = 0.
8. Write to C5 register and configure DMA control register fields as desired for application.
9. Write to set C7816[INIT] = 1,C7816[TTYPE] = 0, and C7816[ISO_7816E] = 1. Program C7816[ONACK] and C7816[ANACK] as desired.
10. Write to IE7816 to set interrupt enable parameters as desired.
11. Write to ET7816 and set as desired.
12. Write to set C2[ILIE] = 0, C2[RE] = 1, C2[TE] = 1, C2[RWU] = 0, and C2[SBK] = 0. Set up interrupt enables C2[TIE], C2[TCIE], and C2[RIE] as desired.

At this time, the UART will start listening for an initial character. After being identified, it will automatically adjust S2[MSBF], C3[TXINV], and S2[RXINV]. The software must then receive and process an answer to reset. Upon processing the answer to reset, the software must write to set C2[RE] = 0 and C2[TE] = 0. The software should then adjust 7816 specific and UART generic parameters to match and configure data that was received during the answer on reset period. After the new settings have been programmed, including the new baud rate and C7816[TTYPE], C2[RE] and C2[TE] can be reenabled as required.

53.8.2.1 Transmission procedure for (C7816[TTYPE] = 0)

When the protocol selected is C7816[TTYPE] = 0, it is assumed that the software has a prior knowledge of who should be transmitting and receiving. Therefore, no mechanism is provided for automated transmission/receipt control. The software must monitor S1[TDRE], or configure for an interrupt, and provide additional data for transmission, as appropriate. Additionally, software should set C2[TE] = 1 and control TXDIR whenever it is the UART's turn to transmit information. For ease of monitoring, it is suggested that only data be transmitted until the next receiver/transmit switchover is loaded into the transmit FIFO/buffer.

53.8.2.2 Transmission procedure for (C7816[TTYPE] = 1)

When the protocol selected is C7816[TTYPE] = 1, data is transferred in blocks. Before starting a transmission, the software must write the size, in number of bytes, for the Information Field portion of the block into TLEN. If a CRC is being transmitted for the block, the value in TLEN must be one more than the size of the information field. The software must then set C2[TE] = 1 and C2[RE] = 1. The software must then monitor S1[TDRE]/interrupt and write the prologue, information, and epilogue field to the transmit buffer. TLEN automatically decrements, except for prologue bytes and the final epilogue byte. When the final epilogue byte has been transmitted, the UART automatically clears C2[TE] and C3[TXDIR] to 0, and the UART automatically starts capturing the response to the block that was transmitted. After the software has detected the receipt of the response, the transmission process must be repeated as needed with sufficient urgency to ensure that the block wait time and character wait times are not violated.

53.8.3 Initialization sequence (non ISO-7816)

To initiate a UART transmission:

1. Configure the UART.
   a. Select a baud rate. Write this value to the UART baud registers (BDH/L) to begin the baud rate generator. Remember that the baud rate generator is disabled when the baud rate is zero. Writing to the BDH has no effect without also writing to BDL.
   b. Write to C1 to configure word length, parity, and other configuration bits (LOOPS, RSRC, M, WAKE, ILT, PE, and PT). Write to C4, MA1, and MA2 to configure.
   c. Enable the transmitter, interrupts, receiver, and wakeup as required, by writing to C2 (TIE, TCIE, RIE, ILIE, TE, RE, RWU, and SBK), S2 (MSBF and BRK13), and C3 (ORIE, NEIE, PEIE, and FEIE). A preamble or idle character is then shifted out of the transmitter shift register.
2. Transmit procedure for each byte.
   a. Monitor S1[TDRE] by reading S1 or responding to the TDRE interrupt. The amount of free space in the transmit buffer directly using TCFIFO[TXCOUNT] can also be monitored.

   b. If the TDRE flag is set, or there is space in the transmit buffer, write the data to be transmitted to (C3[T8]/D). A new transmission will not result until data exists in the transmit buffer.

3. Repeat step 2 for each subsequent transmission.

Note

During normal operation, S1[TDRE] is set when the shift register is loaded with the next data to be transmitted from the transmit buffer and the number of datawords contained in the transmit buffer is less than or equal to the value in TWFIFO[TXWATER]. This occurs 9/16ths of a bit time after the start of the stop bit of the previous frame.

To separate messages with preambles with minimum idle line time, use this sequence between messages.

1. Write the last dataword of the first message to C3[T8]/D.
2. Wait for S1[TDRE] to go high with TWFIFO[TXWATER] = 0, indicating the transfer of the last frame to the transmit shift register.
3. Queue a preamble by clearing and then setting C2[TE].
4. Write the first and subsequent datawords of the second message to C3[T8]/D.

53.8.4 Overrun (OR) flag implications

To be flexible, the overrun flag (OR) operates slight differently depending on the mode of operation. There may be implications that need to be carefully considered. This section clarifies the behavior and the resulting implications. Regardless of mode, if a dataword is received while S1[OR] is set, S1[RDRF] and S1[IDLE] are blocked from asserting. If S1[RDRF] or S1[IDLE] were previously asserted, they will remain asserted until cleared.

53.8.4.1 Overrun operation

The assertion of S1[OR] indicates that a significant event has occurred. The assertion indicates that received data has been lost because there was a lack of room to store it in the data buffer. Therefore, while S1[OR] is set, no further data is stored in the data buffer until S1[OR] is cleared. This ensures that the application will be able to handle the overrun condition.

In most applications, because the total amount of lost data is known, the application will attempt to return the system to a known state. Before S1[OR] is cleared, all received data will be dropped. For this, the software does the following.

1. Remove data from the receive data buffer. This could be done by reading data from the data buffer and processing it if the data in the FIFO was still valuable when the overrun event occurred, or using CFIFO[RXFLUSH] to clear the buffer.
2. Clear S1[OR]. Note that if data was cleared using CFIFO[RXFLUSH], then clearing S1[OR] will result in SFIFO[RXUF] asserting. This is because the only way to clear S1[OR] requires reading additional information from the FIFO. Care should be taken to disable the SFIFO[RXUF] interrupt prior to clearing the OR flag and then clearing SFIFO[RXUF] after the OR flag has been cleared.

Note that, in some applications, if an overrun event is responded to fast enough, the lost data can be recovered. For example, when C7816[ISO_7816E] is asserted, C7816[TTYPE]=1 and C7816[ONACK] = 1, the application may reasonably be able to determine whether the lost data will be resent by the device. In this scenario, flushing the receiver data buffer may not be required. Rather, if S1[OR] is cleared, the lost data may be resent and therefore may be recoverable.

When LIN break detect (LBKDE) is asserted, S1[OR] has significantly different behavior than in other modes. S1[OR] will be set, regardless of how much space is actually available in the data buffer, if a LIN break character has been detected and the corresponding flag, S2[LBKDIF], is not cleared before the first data character is received after S2[LBKDIF] asserted. This behavior is intended to allow the software sufficient time to read the LIN break character from the data buffer to ensure that a break character was actually detected. The checking of the break character was used on some older implementations and is therefore supported for legacy reasons. Applications that do not require this checking can simply clear S2[LBKDIF] without checking the stored value to ensure it is a break character.

53.8.5 Overrun NACK considerations

When C7816[ISO_7816E] is enabled and C7816[TTYPE] = 0, the retransmission feature of the 7816 protocol can be used to help avoid lost data when the data buffer overflows. Using C7816[ONACK], the module can be programmed to issue a NACK on an overflow event. Assuming that the smartcard device has implemented retransmission, the lost data will be retransmitted. While useful, there is a programming implication that may require special consideration. The need to transmit a NACK must be determined and committed to prior to the dataword being fully received. While the NACK is being received, it is possible that the application code will read the data buffer such that sufficient room will be made to store the dataword that is being NACKed. Even if room has been made in the data buffer after the transmission of a NACK is completed, the received data will always be discarded as a result of an overflow and the ET7816[RXTHRESHOLD] value will be incremented by one. However, if sufficient space now exists to write the received data which was NACK'ed, S1[OR] will be blocked and kept from asserting.

53.8.6 Match address registers

The two match address registers allow a second match address function for a broadcast or general call address to the serial bus, as an example.

53.8.7 Modem feature

This section describes the modem features.

53.8.7.1 Ready-to-receive using RTS

To help to stop overrun of the receiver data buffer, the RTS signal can be used by the receiver to indicate to another UART that it is ready to receive data. The other UART can send the data when its CTS signal is asserted. This handshaking conforms to the TIA-232-E standard. A transceiver is necessary if the required voltage levels of the communication link do not match the voltage levels of the UART's RTS and CTS signals.

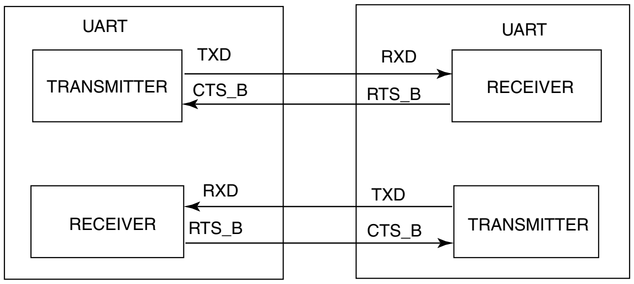


Figure 53-28. Ready-to-receive

The transmitter's CTS signal can be used for hardware flow control whether its RTS signal is used for hardware flow control, transceiver driver enable, or not at all.

53.8.7.2 Transceiver driver enable using RTS

RS-485 is a multiple drop communication protocol in which the UART transceiver's driver is 3-stated unless the UART is driving. The RTS signal can be used by the transmitter to enable the driver of a transceiver. The polarity of RTS can be matched to the polarity of the transceiver's driver enable signal. See the following figure.

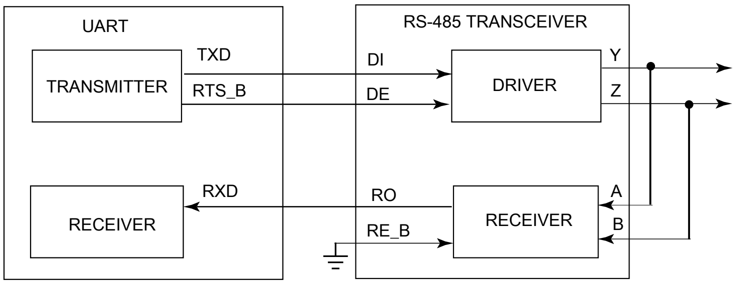


Figure 53-29. Transceiver driver enable using RTS

In the figure, the receiver enable signal is asserted. Another option for this connection is to connect RTS_B to both DE and RE_B. The transceiver's receiver is disabled while driving. A pullup can pull RXD to a non-floating value during this time. This option can be refined further by operating the UART in single wire mode, freeing the RXD pin for other uses.

53.8.8 IrDA minimum pulse width

The IrDA specifies a minimum pulse width of 1.6 μs. The UART hardware does not include a mechanism to restrict/force the pulse width to be greater than or equal to 1.6 μs. However, configuring the baud rate to 115.2 kbit/s and the narrow pulse width to 3/16 of a bit time results in a pulse width of 1.6 μs.

53.8.9 Clearing 7816 wait timer (WT, BWT, CWT) interrupts

The 7816 wait timer interrupts associated with IS7816[WT], IS7816[BWT], and IS7816[CWT] will automatically reassert if they are cleared and the wait time is still violated. This behavior is similar to most of the other interrupts on the UART. In most cases, if the condition that caused the interrupt to trigger still exists when the interrupt is cleared, then the interrupt will reassert. For example, consider the following scenario:

1. IS7816[WT] is programmed to assert after 9600 cycles of unresponsiveness.
2. The 9600 cycles pass without a response resulting in the WT interrupt asserting.
3. The IS7816[WT] is cleared at cycle 9700 by the interrupt service routine.

4. After the WT interrupt has been cleared, the smartcard remains unresponsive. At cycle 9701 the WT interrupt will be reasserted.

If the intent of clearing the interrupt is such that it does not reassert, the interrupt service routine must remove or clear the condition that originally caused the interrupt to assert prior to clearing the interrupt. There are multiple ways that this can be accomplished, including ensuring that an event that results in the wait timer resetting occurs, such as, the transmission of another packet.

53.8.10 Legacy and reverse compatibility considerations

Recent versions of the UART have added several new features. Whenever reasonably possible, reverse compatibility was maintained. However, in some cases this was either not feasible or the behavior was deemed as not intended. This section describes several differences to legacy operation that resulted from these recent enhancements. If application code from previous versions is used, it must be reviewed and modified to take the following items into account. Depending on the application code, additional items that are not listed here may also need to be considered.

1. Various reserved registers and register bits are used, such as, MSFB and M10.
2. This module now generates an error when invalid address spaces are used.
3. While documentation indicated otherwise, in some cases it was possible for S1[IDLE] to assert even if S1[OR] was set.
4. S1[OR] will be set only if the data buffer (FIFO) does not have sufficient room. Previously, the data buffer was always a fixed size of one and the S1[OR] flag would set so long as S1[RDRF] was set even if there was room in the data buffer. While the clearing mechanism has remained the same for S1[RDRF], keeping the OR flag assertion tied to the RDRF event rather than the data buffer being full would have greatly reduced the usefulness of the buffer when its size is larger than one.
5. Previously, when C2[RWU] was set (and WAKE = 0), the IDLE flag could reassert up to every bit period causing an interrupt and requiring the host processor to reassert C2[RWU]. This behavior has been modified. Now, when C2[RWU] is set (and WAKE = 0), at least one non-idle bit must be detected before an idle can be detected.

Chapter 54
MCU: Integrated Interchip Sound (I2S) / Synchronous Audio Interface (SAI)

54.1 Introduction

NOTE

> For the chip-specific implementation details of this module's instances, see the chip configuration information.

The I^2S (or I2S) module provides a synchronous audio interface (SAI) that supports full-duplex serial interfaces with frame synchronization such as I^2S, AC97, TDM, and codec/DSP interfaces.

54.1.1 Features

Note that some of the features are not supported across all SAI instances; see the chip-specific information in the first section of this chapter.

- Transmitter with independent bit clock and frame sync supporting 1 data line
- Receiver with independent bit clock and frame sync supporting 1 data line
- Maximum Frame Size of 16 words
- Word size of between 8-bits and 32-bits
- Word size configured separately for first word and remaining words in frame
- Asynchronous 8 × 32-bit FIFO for each transmit and receive channel
- Supports graceful restart after FIFO error

54.1.2 Block diagram

The following block diagram also shows the module clocks.

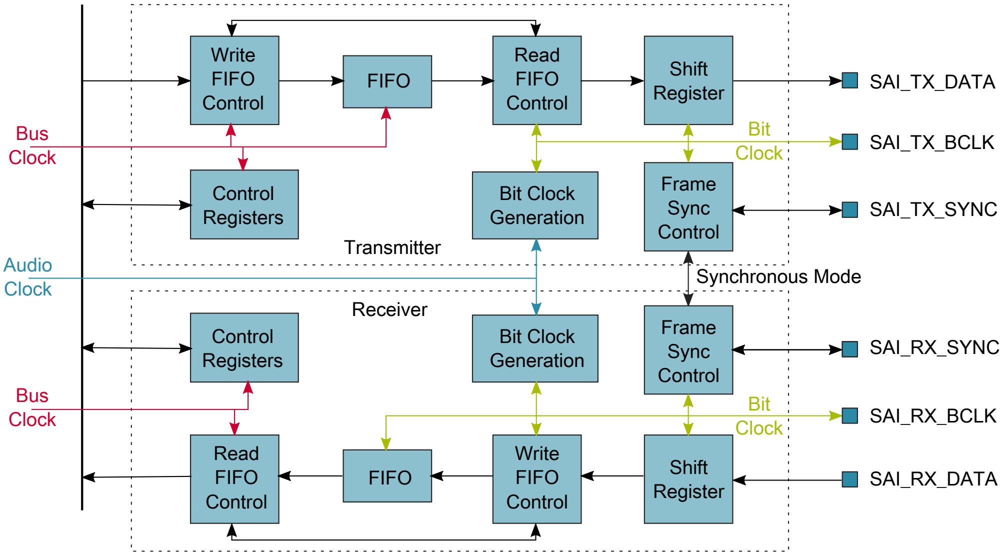


Figure 54-1. I^{2}S/SAI block diagram

54.1.3 Modes of operation

The module operates in these power modes: Run mode, stop modes, low-leakage modes, and Debug mode.

54.1.3.1 Run mode

In Run mode, the SAI transmitter and receiver operate normally.

54.1.3.2 Stop modes

In Stop mode, the SAI transmitter and/or receiver can continue operating provided the appropriate Stop Enable bit is set (TCSR[STOPE] and/or RCSR[STOPE], respectively), and provided the transmitter and/or receiver is/are using an externally generated bit clock or an Audio Master Clock that remains operating in Stop mode. The SAI transmitter and/or receiver can generate an asynchronous interrupt to wake the CPU from Stop mode.

In Stop mode, if the Transmitter Stop Enable (TCSR[STOPE]) bit is clear, the transmitter is disabled after completing the current transmit frame, and, if the Receiver Stop Enable (RCSR[STOPE]) bit is clear, the receiver is disabled after completing the current receive frame. Entry into Stop mode is prevented–not acknowledged–while waiting for the transmitter and receiver to be disabled at the end of the current frame.

54.1.3.3 Low-leakage modes

When entering low-leakage modes, the Stop Enable (TCSR[STOPE] and RCSR[STOPE]) bits are ignored and the SAI is disabled after completing the current transmit and receive Frames. Entry into stop mode is prevented (not acknowledged) while waiting for the transmitter and receiver to be disabled at the end of the current frame.

54.1.3.4 Debug mode

In Debug mode, the SAI transmitter and/or receiver can continue operating provided the Debug Enable bit is set. When TCSR[DBGE] or RCSR[DBGE] bit is clear and Debug mode is entered, the SAI is disabled after completing the current transmit or receive frame. The transmitter and receiver bit clocks are not affected by Debug mode.

54.2 External signals

Name	Function	I/O
SAI_TX_BCLK	Transmit Bit Clock. The bit clock is an input when externally generated and an output when internally generated.	I/O
SAI_TX_SYNC	Transmit Frame Sync. The frame sync is an input sampled synchronously by the bit clock when externally generated and an output generated synchronously by the bit clock when internally generated.	I/O
SAI_TX_DATA	Transmit Data. The transmit data is generated synchronously by the bit clock and is tristated whenever not transmitting a word.	O
SAI_RX_BCLK	Receive Bit Clock. The bit clock is an input when externally generated and an output when internally generated.	I/O
SAI_RX_SYNC	Receive Frame Sync. The frame sync is an input sampled synchronously by the bit clock when externally generated and an output generated synchronously by the bit clock when internally generated.	I/O
SAI_RX_DATA	Receive Data. The receive data is sampled synchronously by the bit clock.	I
SAI_MCLK	Audio Master Clock. The master clock is an input when externally generated and an output when internally generated.	I/O

54.3 Memory map and register definition

A read or write access to an address from offset 0x108 and above will result in a bus error.

I2S memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4002_F000	SAI Transmit Control Register (I2S0_TCSR)	32	R/W	0000_0000h	54.3.1/1423
4002_F004	SAI Transmit Configuration 1 Register (I2S0_TCR1)	32	R/W	0000_0000h	54.3.2/1426
4002_F008	SAI Transmit Configuration 2 Register (I2S0_TCR2)	32	R/W	0000_0000h	54.3.3/1426
4002_F00C	SAI Transmit Configuration 3 Register (I2S0_TCR3)	32	R/W	0000_0000h	54.3.4/1428
4002_F010	SAI Transmit Configuration 4 Register (I2S0_TCR4)	32	R/W	0000_0000h	54.3.5/1429
4002_F014	SAI Transmit Configuration 5 Register (I2S0_TCR5)	32	R/W	0000_0000h	54.3.6/1430
4002_F020	SAI Transmit Data Register (I2S0_TDR0)	32	W (always reads 0)	0000_0000h	54.3.7/1431
4002_F040	SAI Transmit FIFO Register (I2S0_TFR0)	32	R	0000_0000h	54.3.8/1432
4002_F060	SAI Transmit Mask Register (I2S0_TMR)	32	R/W	0000_0000h	54.3.9/1432
4002_F080	SAI Receive Control Register (I2S0_RCSR)	32	R/W	0000_0000h	54.3.10/ 1433
4002_F084	SAI Receive Configuration 1 Register (I2S0_RCR1)	32	R/W	0000_0000h	54.3.11/ 1436
4002_F088	SAI Receive Configuration 2 Register (I2S0_RCR2)	32	R/W	0000_0000h	54.3.12/ 1437
4002_F08C	SAI Receive Configuration 3 Register (I2S0_RCR3)	32	R/W	0000_0000h	54.3.13/ 1438
4002_F090	SAI Receive Configuration 4 Register (I2S0_RCR4)	32	R/W	0000_0000h	54.3.14/ 1439
4002_F094	SAI Receive Configuration 5 Register (I2S0_RCR5)	32	R/W	0000_0000h	54.3.15/ 1441
4002_F0A0	SAI Receive Data Register (I2S0_RDR0)	32	R	0000_0000h	54.3.16/ 1441
4002_F0C0	SAI Receive FIFO Register (I2S0_RFR0)	32	R	0000_0000h	54.3.17/ 1442
4002_F0E0	SAI Receive Mask Register (I2S0_RMR)	32	R/W	0000_0000h	54.3.18/ 1442
4002_F100	SAI MCLK Control Register (I2S0_MCR)	32	R/W	0000_0000h	54.3.19/ 1443
4002_F104	SAI MCLK Divide Register (I2S0_MDR)	32	R/W	0000_0000h	54.3.20/ 1444

54.3.1 SAI Transmit Control Register (I2Sx_TCSR)

Address: 4002_F000h base + 0h offset = 4002_F000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	TE	STOPE	DBGE	BCE	0		0	SR	0			WSF	SEF	FEF	FWF	FRF
W							FR					w1c	w1c	w1c		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0			WSIE	SEIE	FEIE	FWIE	FRIE	0			0			FWDE	FRDE
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

I2Sx_TCSR field descriptions

Field	Description
31 TE	Transmitter Enable Enables/disables the transmitter. When software clears this field, the transmitter remains enabled, and this bit remains set, until the end of the current frame. 0 Transmitter is disabled. 1 Transmitter is enabled, or transmitter has been disabled and has not yet reached end of frame.
30 STOPE	Stop Enable Configures transmitter operation in Stop mode. This field is ignored and the transmitter is disabled in all low-leakage stop modes. 0 Transmitter disabled in Stop mode. 1 Transmitter enabled in Stop mode.
29 DBGE	Debug Enable

Table continues on the next page...

I2Sx_TCSR field descriptions (continued)

Field	Description
	Enables/disables transmitter operation in Debug mode. The transmit bit clock is not affected by debug mode. 0 Transmitter is disabled in Debug mode, after completing the current frame. 1 Transmitter is enabled in Debug mode.
28 BCE	Bit Clock Enable Enables the transmit bit clock, separately from the TE. This field is automatically set whenever TE is set. When software clears this field, the transmit bit clock remains enabled, and this bit remains set, until the end of the current frame. 0 Transmit bit clock is disabled. 1 Transmit bit clock is enabled.
27–26 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
25 FR	FIFO Reset Resets the FIFO pointers. Reading this field will always return zero. FIFO pointers should only be reset when the transmitter is disabled or the FIFO error flag is set. 0 No effect. 1 FIFO reset.
24 SR	Software Reset When set, resets the internal transmitter logic including the FIFO pointers. Software-visible registers are not affected, except for the status registers. 0 No effect. 1 Software reset.
23–21 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
20 WSF	Word Start Flag Indicates that the start of the configured word has been detected. Write a logic 1 to this field to clear this flag. 0 Start of word not detected. 1 Start of word detected.
19 SEF	Sync Error Flag Indicates that an error in the externally-generated frame sync has been detected. Write a logic 1 to this field to clear this flag. 0 Sync error not detected. 1 Frame sync error detected.
18 FEF	FIFO Error Flag Indicates that an enabled transmit FIFO has underrun. Write a logic 1 to this field to clear this flag. 0 Transmit underrun not detected. 1 Transmit underrun detected.

Table continues on the next page...

I2Sx_TCSR field descriptions (continued)

Field	Description
17 FWF	FIFO Warning Flag Indicates that an enabled transmit FIFO is empty. 0 No enabled transmit FIFO is empty. 1 Enabled transmit FIFO is empty.
16 FRF	FIFO Request Flag Indicates that the number of words in an enabled transmit channel FIFO is less than or equal to the transmit FIFO watermark. 0 Transmit FIFO watermark has not been reached. 1 Transmit FIFO watermark has been reached.
15–13 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
12 WSIE	Word Start Interrupt Enable Enables/disables word start interrupts. 0 Disables interrupt. 1 Enables interrupt.
11 SEIE	Sync Error Interrupt Enable Enables/disables sync error interrupts. 0 Disables interrupt. 1 Enables interrupt.
10 FEIE	FIFO Error Interrupt Enable Enables/disables FIFO error interrupts. 0 Disables the interrupt. 1 Enables the interrupt.
9 FWIE	FIFO Warning Interrupt Enable Enables/disables FIFO warning interrupts. 0 Disables the interrupt. 1 Enables the interrupt.
8 FRIE	FIFO Request Interrupt Enable Enables/disables FIFO request interrupts. 0 Disables the interrupt. 1 Enables the interrupt.
7–5 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
4–2 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
1 FWDE	FIFO Warning DMA Enable

Table continues on the next page...

I2Sx_TCSR field descriptions (continued)

Field	Description
	Enables/disables DMA requests. 0 Disables the DMA request. 1 Enables the DMA request.
0 FRDE	FIFO Request DMA Enable Enables/disables DMA requests. 0 Disables the DMA request. 1 Enables the DMA request.

54.3.2 SAI Transmit Configuration 1 Register (I2Sx_TCR1)

Address: 4002_F000h base + 4h offset = 4002_F004h

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

I2Sx_TCR1 field descriptions

Field	Description
31–3 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
TFW	Transmit FIFO Watermark Configures the watermark level for all enabled transmit channels.

54.3.3 SAI Transmit Configuration 2 Register (I2Sx_TCR2)

This register must not be altered when TCSR[TE] is set.

Address: 4002_F000h base + 8h offset = 4002_F008h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	SYNC		BCS	BCI	MSEL		BCP	BCD	0							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								DIV							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

I2Sx_TCR2 field descriptions

Field	Description
31–30 SYNC	Synchronous Mode Configures between asynchronous and synchronous modes of operation. When configured for a synchronous mode of operation, the receiver must be configured for asynchronous operation. 00 Asynchronous mode. 01 Synchronous with receiver. 10 Synchronous with another SAI transmitter. 11 Synchronous with another SAI receiver.
29 BCS	Bit Clock Swap This field swaps the bit clock used by the transmitter. When the transmitter is configured in asynchronous mode and this bit is set, the transmitter is clocked by the receiver bit clock (SAI_RX_BCLK). This allows the transmitter and receiver to share the same bit clock, but the transmitter continues to use the transmit frame sync (SAI_TX_SYNC). When the transmitter is configured in synchronous mode, the transmitter BCS field and receiver BCS field must be set to the same value. When both are set, the transmitter and receiver are both clocked by the transmitter bit clock (SAI_TX_BCLK) but use the receiver frame sync (SAI_RX_SYNC). 0 Use the normal bit clock source. 1 Swap the bit clock source.
28 BCI	Bit Clock Input When this field is set and using an internally generated bit clock in either synchronous or asynchronous mode, the bit clock actually used by the transmitter is delayed by the pad output delay (the transmitter is clocked by the pad input as if the clock was externally generated). This has the effect of decreasing the data input setup time, but increasing the data output valid time. The slave mode timing from the datasheet should be used for the transmitter when this bit is set. In synchronous mode, this bit allows the transmitter to use the slave mode timing from the datasheet, while the receiver uses the master mode timing. This field has no effect when configured for an externally generated bit clock . 0 No effect. 1 Internal logic is clocked as if bit clock was externally generated.
27–26 MSEL	MCLK Select Selects the audio Master Clock option used to generate an internally generated bit clock. This field has no effect when configured for an externally generated bit clock. **NOTE:** Depending on the device, some Master Clock options might not be available. See the chip-specific information for the meaning of each option. 01 Master Clock (MCLK) 1 option selected. 10 Master Clock (MCLK) 2 option selected. 11 Master Clock (MCLK) 3 option selected.
25 BCP	Bit Clock Polarity Configures the polarity of the bit clock. 0 Bit clock is active high with drive outputs on rising edge and sample inputs on falling edge. 1 Bit clock is active low with drive outputs on falling edge and sample inputs on rising edge.

Table continues on the next page...

I2Sx_TCR2 field descriptions (continued)

Field	Description
24 BCD	Bit Clock Direction Configures the direction of the bit clock. 0 Bit clock is generated externally in Slave mode. 1 Bit clock is generated internally in Master mode.
23–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
DIV	Bit Clock Divide Divides down the audio master clock to generate the bit clock when configured for an internal bit clock. The division value is (DIV + 1) * 2.

54.3.4 SAI Transmit Configuration 3 Register (I2Sx_TCR3)

This register must not be altered when TCSR[TE] is set.

Address: 4002_F000h base + Ch offset = 4002_F00Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0								0							TCE
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0												WDFL			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

I2Sx_TCR3 field descriptions

Field	Description
31–24 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
23–17 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
16 TCE	Transmit Channel Enable Enables the corresponding data channel for transmit operation. A channel must be enabled before its FIFO is accessed. 0 Transmit data channel N is disabled. 1 Transmit data channel N is enabled.
15–4 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
WDFL	Word Flag Configuration

Table continues on the next page...

I2Sx_TCR3 field descriptions (continued)

Field	Description
	Configures which word sets the start of word flag. The value written must be one less than the word number. For example, writing 0 configures the first word in the frame. When configured to a value greater than TCR4[FRSZ], then the start of word flag is never set.

54.3.5 SAI Transmit Configuration 4 Register (I2Sx_TCR4)

This register must not be altered when TCSR[TE] is set.

Address: 4002_F000h base + 10h offset = 4002_F010h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0			0	0		0		0				FRSZ			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0			SYWD					0			MF	FSE	0	FSP	FSD
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

I2Sx_TCR4 field descriptions

Field	Description
31–29 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
28 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
27–26 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
25–24 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
23–20 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
19–16 FRSZ	Frame size Configures the number of words in each frame. The value written must be one less than the number of words in the frame. For example, write 0 for one word per frame. The maximum supported frame size is 16 words.
15–13 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
12–8 SYWD	Sync Width Configures the length of the frame sync in number of bit clocks. The value written must be one less than the number of bit clocks. For example, write 0 for the frame sync to assert for one bit clock only. The sync width cannot be configured longer than the first word of the frame.
7–5 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

Table continues on the next page...

I2Sx_TCR4 field descriptions (continued)

Field	Description
4 MF	MSB First Configures whether the LSB or the MSB is transmitted first. 0 LSB is transmitted first. 1 MSB is transmitted first.
3 FSE	Frame Sync Early 0 Frame sync asserts with the first bit of the frame. 1 Frame sync asserts one bit before the first bit of the frame.
2 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
1 FSP	Frame Sync Polarity Configures the polarity of the frame sync. 0 Frame sync is active high. 1 Frame sync is active low.
0 FSD	Frame Sync Direction Configures the direction of the frame sync. 0 Frame sync is generated externally in Slave mode. 1 Frame sync is generated internally in Master mode.

54.3.6 SAI Transmit Configuration 5 Register (I2Sx_TCR5)

This register must not be altered when TCSR[TE] is set.

Address: 4002_F000h base + 14h offset = 4002_F014h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0			WNW					0			W0W					0			FBT					0							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

I2Sx_TCR5 field descriptions

Field	Description
31–29 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
28–24 WNW	Word N Width Configures the number of bits in each word, for each word except the first in the frame. The value written must be one less than the number of bits per word. Word width of less than 8 bits is not supported.
23–21 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

Table continues on the next page...

I2Sx_TCR5 field descriptions (continued)

Field	Description
20–16 W0W	Word 0 Width Configures the number of bits in the first word in each frame. The value written must be one less than the number of bits in the first word. Word width of less than 8 bits is not supported if there is only one word per frame.
15–13 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
12–8 FBT	First Bit Shifted Configures the bit index for the first bit transmitted for each word in the frame. If configured for MSB First, the index of the next bit transmitted is one less than the current bit transmitted. If configured for LSB First, the index of the next bit transmitted is one more than the current bit transmitted. The value written must be greater than or equal to the word width when configured for MSB First. The value written must be less than or equal to 31-word width when configured for LSB First.
Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

54.3.7 SAI Transmit Data Register (I2S*x*_TDR*n*)

Address: 4002_F000h base + 20h offset + (4d × i), where i=0d to 0d

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

I2S*x*_TDR*n* field descriptions

Field	Description
TDR	Transmit Data Register The corresponding TCR3[TCE] bit must be set before accessing the channel's transmit data register. Writes to this register when the transmit FIFO is not full will push the data written into the transmit data FIFO. Writes to this register when the transmit FIFO is full are ignored.

54.3.8 SAI Transmit FIFO Register (I2Sx_TFR*n*)

The MSB of the read and write pointers is used to distinguish between FIFO full and empty conditions. If the read and write pointers are identical, then the FIFO is empty. If the read and write pointers are identical except for the MSB, then the FIFO is full.

Address: 4002_F000h base + 40h offset + (4d × i), where i=0d to 0d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	0											WFP			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0												RFP			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

I2Sx_TFR*n* field descriptions

Field	Description
31 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
30–20 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
19–16 WFP	Write FIFO Pointer FIFO write pointer for transmit data channel.
15–4 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
RFP	Read FIFO Pointer FIFO read pointer for transmit data channel.

54.3.9 SAI Transmit Mask Register (I2Sx_TMR)

This register is double-buffered and updates:

1. When TCSR[TE] is first set
2. At the end of each frame.

This allows the masked words in each frame to change from frame to frame.

Address: 4002_F000h base + 60h offset = 4002_F060h

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

I2Sx_TMR field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
TWM	Transmit Word Mask Configures whether the transmit word is masked (transmit data pin tristated and transmit data not read from FIFO) for the corresponding word in the frame. 0 Word N is enabled. 1 Word N is masked. The transmit data pins are tri-stated when masked.

54.3.10 SAI Receive Control Register (I2Sx_RCSR)

Address: 4002_F000h base + 80h offset = 4002_F080h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	RE	STOPE	DBGE	BCE	0		0	SR	0			WSF	SEF	FEF	FWF	FRF
W							FR					w1c	w1c	w1c		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0			WSIE	SEIE	FEIE	FWIE	FRIE	0			0			FWDE	FRDE
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

I2Sx_RCSR field descriptions

Field	Description
31 RE	Receiver Enable Enables/disables the receiver. When software clears this field, the receiver remains enabled, and this bit remains set, until the end of the current frame. 0 Receiver is disabled. 1 Receiver is enabled, or receiver has been disabled and has not yet reached end of frame.
30 STOPE	Stop Enable Configures receiver operation in Stop mode. This bit is ignored and the receiver is disabled in all low-leakage stop modes. 0 Receiver disabled in Stop mode. 1 Receiver enabled in Stop mode.
29 DBGE	Debug Enable Enables/disables receiver operation in Debug mode. The receive bit clock is not affected by Debug mode. 0 Receiver is disabled in Debug mode, after completing the current frame. 1 Receiver is enabled in Debug mode.
28 BCE	Bit Clock Enable Enables the receive bit clock, separately from RE. This field is automatically set whenever RE is set. When software clears this field, the receive bit clock remains enabled, and this field remains set, until the end of the current frame. 0 Receive bit clock is disabled. 1 Receive bit clock is enabled.
27–26 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
25 FR	FIFO Reset Resets the FIFO pointers. Reading this field will always return zero. FIFO pointers should only be reset when the receiver is disabled or the FIFO error flag is set. 0 No effect. 1 FIFO reset.
24 SR	Software Reset Resets the internal receiver logic including the FIFO pointers. Software-visible registers are not affected, except for the status registers. 0 No effect. 1 Software reset.
23–21 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
20 WSF	Word Start Flag Indicates that the start of the configured word has been detected. Write a logic 1 to this field to clear this flag.

Table continues on the next page...

I2Sx_RCSR field descriptions (continued)

Field	Description
	0 Start of word not detected. 1 Start of word detected.
19 SEF	Sync Error Flag Indicates that an error in the externally-generated frame sync has been detected. Write a logic 1 to this field to clear this flag. 0 Sync error not detected. 1 Frame sync error detected.
18 FEF	FIFO Error Flag Indicates that an enabled receive FIFO has overflowed. Write a logic 1 to this field to clear this flag. 0 Receive overflow not detected. 1 Receive overflow detected.
17 FWF	FIFO Warning Flag Indicates that an enabled receive FIFO is full. 0 No enabled receive FIFO is full. 1 Enabled receive FIFO is full.
16 FRF	FIFO Request Flag Indicates that the number of words in an enabled receive channel FIFO is greater than the receive FIFO watermark. 0 Receive FIFO watermark not reached. 1 Receive FIFO watermark has been reached.
15–13 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
12 WSIE	Word Start Interrupt Enable Enables/disables word start interrupts. 0 Disables interrupt. 1 Enables interrupt.
11 SEIE	Sync Error Interrupt Enable Enables/disables sync error interrupts. 0 Disables interrupt. 1 Enables interrupt.
10 FEIE	FIFO Error Interrupt Enable Enables/disables FIFO error interrupts. 0 Disables the interrupt. 1 Enables the interrupt.
9 FWIE	FIFO Warning Interrupt Enable Enables/disables FIFO warning interrupts.

Table continues on the next page...

I2Sx_RCSR field descriptions (continued)

Field	Description
	0 Disables the interrupt. 1 Enables the interrupt.
8 FRIE	FIFO Request Interrupt Enable Enables/disables FIFO request interrupts. 0 Disables the interrupt. 1 Enables the interrupt.
7–5 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
4–2 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
1 FWDE	FIFO Warning DMA Enable Enables/disables DMA requests. 0 Disables the DMA request. 1 Enables the DMA request.
0 FRDE	FIFO Request DMA Enable Enables/disables DMA requests. 0 Disables the DMA request. 1 Enables the DMA request.

54.3.11 SAI Receive Configuration 1 Register (I2Sx_RCR1)

Address: 4002_F000h base + 84h offset = 4002_F084h

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

I2Sx_RCR1 field descriptions

Field	Description
31–3 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
RFW	Receive FIFO Watermark Configures the watermark level for all enabled receiver channels.

54.3.12 SAI Receive Configuration 2 Register (I2Sx_RCR2)

This register must not be altered when RCSR[RE] is set.

Address: 4002_F000h base + 88h offset = 4002_F088h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	SYNC		BCS	BCI	MSEL		BCP	BCD	0							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								DIV							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

I2Sx_RCR2 field descriptions

Field	Description
31–30 SYNC	Synchronous Mode Configures between asynchronous and synchronous modes of operation. When configured for a synchronous mode of operation, the transmitter must be configured for asynchronous operation. 00 Asynchronous mode. 01 Synchronous with transmitter. 10 Synchronous with another SAI receiver. 11 Synchronous with another SAI transmitter.
29 BCS	Bit Clock Swap This field swaps the bit clock used by the receiver. When the receiver is configured in asynchronous mode and this bit is set, the receiver is clocked by the transmitter bit clock (SAI_TX_BCLK). This allows the transmitter and receiver to share the same bit clock, but the receiver continues to use the receiver frame sync (SAI_RX_SYNC). When the receiver is configured in synchronous mode, the transmitter BCS field and receiver BCS field must be set to the same value. When both are set, the transmitter and receiver are both clocked by the receiver bit clock (SAI_RX_BCLK) but use the transmitter frame sync (SAI_TX_SYNC). 0 Use the normal bit clock source. 1 Swap the bit clock source.
28 BCI	Bit Clock Input When this field is set and using an internally generated bit clock in either synchronous or asynchronous mode, the bit clock actually used by the receiver is delayed by the pad output delay (the receiver is clocked by the pad input as if the clock was externally generated). This has the effect of decreasing the data input setup time, but increasing the data output valid time. The slave mode timing from the datasheet should be used for the receiver when this bit is set. In synchronous mode, this bit allows the receiver to use the slave mode timing from the datasheet, while the transmitter uses the master mode timing. This field has no effect when configured for an externally generated bit clock . 0 No effect. 1 Internal logic is clocked as if bit clock was externally generated.

Table continues on the next page...

I2Sx_RCR2 field descriptions (continued)

Field	Description
27–26 MSEL	MCLK Select Selects the audio Master Clock option used to generate an internally generated bit clock. This field has no effect when configured for an externally generated bit clock. **NOTE:** Depending on the device, some Master Clock options might not be available. See the chip-specific information for the availability and chip-specific meaning of each option. 00 Bus Clock selected. 01 Master Clock (MCLK) 1 option selected. 10 Master Clock (MCLK) 2 option selected. 11 Master Clock (MCLK) 3 option selected.
25 BCP	Bit Clock Polarity Configures the polarity of the bit clock. 0 Bit Clock is active high with drive outputs on rising edge and sample inputs on falling edge. 1 Bit Clock is active low with drive outputs on falling edge and sample inputs on rising edge.
24 BCD	Bit Clock Direction Configures the direction of the bit clock. 0 Bit clock is generated externally in Slave mode. 1 Bit clock is generated internally in Master mode.
23–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
DIV	Bit Clock Divide Divides down the audio master clock to generate the bit clock when configured for an internal bit clock. The division value is (DIV + 1) * 2.

54.3.13 SAI Receive Configuration 3 Register (I2Sx_RCR3)

This register must not be altered when RCSR[RE] is set.

Address: 4002_F000h base + 8Ch offset = 4002_F08Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0								0							RCE
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0												WDFL			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

I2Sx_RCR3 field descriptions

Field	Description
31–24 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
23–17 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
16 RCE	Receive Channel Enable Enables the corresponding data channel for receive operation. A channel must be enabled before its FIFO is accessed. 0 Receive data channel N is disabled. 1 Receive data channel N is enabled.
15–4 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
WDFL	Word Flag Configuration Configures which word the start of word flag is set. The value written should be one less than the word number (for example, write zero to configure for the first word in the frame). When configured to a value greater than the Frame Size field, then the start of word flag is never set.

54.3.14 SAI Receive Configuration 4 Register (I2Sx_RCR4)

This register must not be altered when RCSR[RE] is set.

Address: 4002_F000h base + 90h offset = 4002_F090h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0			0	0		0		0				FRSZ			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0			SYWD					0			MF	FSE	0	FSP	FSD
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

I2Sx_RCR4 field descriptions

Field	Description
31–29 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
28 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
27–26 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
25–24 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

Table continues on the next page...

I2Sx_RCR4 field descriptions (continued)

Field	Description
23–20 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
19–16 FRSZ	Frame Size Configures the number of words in each frame. The value written must be one less than the number of words in the frame. For example, write 0 for one word per frame. The maximum supported frame size is 16 words.
15–13 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
12–8 SYWD	Sync Width Configures the length of the frame sync in number of bit clocks. The value written must be one less than the number of bit clocks. For example, write 0 for the frame sync to assert for one bit clock only. The sync width cannot be configured longer than the first word of the frame.
7–5 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
4 MF	MSB First Configures whether the LSB or the MSB is received first. 0 LSB is received first. 1 MSB is received first.
3 FSE	Frame Sync Early 0 Frame sync asserts with the first bit of the frame. 1 Frame sync asserts one bit before the first bit of the frame.
2 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
1 FSP	Frame Sync Polarity Configures the polarity of the frame sync. 0 Frame sync is active high. 1 Frame sync is active low.
0 FSD	Frame Sync Direction Configures the direction of the frame sync. 0 Frame Sync is generated externally in Slave mode. 1 Frame Sync is generated internally in Master mode.

54.3.15 SAI Receive Configuration 5 Register (I2S*x*_RCR5)

This register must not be altered when RCSR[RE] is set.

Address: 4002_F000h base + 94h offset = 4002_F094h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0			WNW					0			W0W					0			FBT					0							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

I2Sx_RCR5 field descriptions

Field	Description
31–29 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
28–24 WNW	Word N Width Configures the number of bits in each word, for each word except the first in the frame. The value written must be one less than the number of bits per word. Word width of less than 8 bits is not supported.
23–21 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
20–16 W0W	Word 0 Width Configures the number of bits in the first word in each frame. The value written must be one less than the number of bits in the first word. Word width of less than 8 bits is not supported if there is only one word per frame.
15–13 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
12–8 FBT	First Bit Shifted Configures the bit index for the first bit received for each word in the frame. If configured for MSB First, the index of the next bit received is one less than the current bit received. If configured for LSB First, the index of the next bit received is one more than the current bit received. The value written must be greater than or equal to the word width when configured for MSB First. The value written must be less than or equal to 31-word width when configured for LSB First.
Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

54.3.16 SAI Receive Data Register (I2S*x*_RDR*n*)

Reading this register introduces one additional peripheral clock wait state on each read.

Address: 4002_F000h base + A0h offset + (4d × i), where i=0d to 0d

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

I2Sx_RDR*n* field descriptions

Field	Description
RDR	Receive Data Register The corresponding RCR3[RCE] bit must be set before accessing the channel's receive data register. Reads from this register when the receive FIFO is not empty will return the data from the top of the receive FIFO. Reads from this register when the receive FIFO is empty are ignored.

54.3.17 SAI Receive FIFO Register (I2Sx_RFR*n*)

The MSB of the read and write pointers is used to distinguish between FIFO full and empty conditions. If the read and write pointers are identical, then the FIFO is empty. If the read and write pointers are identical except for the MSB, then the FIFO is full.

Address: 4002_F000h base + C0h offset + (4d × i), where i=0d to 0d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0												WFP			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0											RFP			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

I2Sx_RFR*n* field descriptions

Field	Description
31–20 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
19–16 WFP	Write FIFO Pointer FIFO write pointer for receive data channel.
15 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
14–4 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
RFP	Read FIFO Pointer FIFO read pointer for receive data channel.

54.3.18 SAI Receive Mask Register (I2Sx_RMR)

This register is double-buffered and updates:

1. When RCSR[RE] is first set
2. At the end of each frame

This allows the masked words in each frame to change from frame to frame.

Address: 4002_F000h base + E0h offset = 4002_F0E0h

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

I2Sx_RMR field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
RWM	Receive Word Mask Configures whether the receive word is masked (received data ignored and not written to receive FIFO) for the corresponding word in the frame. 0 Word N is enabled. 1 Word N is masked.

54.3.19 SAI MCLK Control Register (I2Sx_MCR)

The MCLK Control Register (MCR) controls the clock source and direction of the audio master clock.

Address: 4002_F000h base + 100h offset = 4002_F100h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	DUF	MOE	0				MICS		0							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

I2Sx_MCR field descriptions

Field	Description
31 DUF	Divider Update Flag Provides the status of on-the-fly updates to the MCLK divider ratio.

Table continues on the next page...

I2Sx_MCR field descriptions (continued)

Field	Description
	0 MCLK divider ratio is not being updated currently. 1 MCLK divider ratio is updating on-the-fly. Further updates to the MCLK divider ratio are blocked while this flag remains set.
30 MOE	MCLK Output Enable Enables the MCLK divider and configures the MCLK signal pin as an output. When software clears this field, it remains set until the MCLK divider is fully disabled. 0 MCLK signal pin is configured as an input that bypasses the MCLK divider. 1 MCLK signal pin is configured as an output from the MCLK divider and the MCLK divider is enabled.
29–26 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
25–24 MICS	MCLK Input Clock Select Selects the clock input to the MCLK divider. This field cannot be changed while the MCLK divider is enabled. See the chip-specific information for the connections to these inputs. 00 MCLK divider input clock 0 is selected. 10 MCLK divider input clock 2 is selected. 11 MCLK divider input clock 3 is selected.
Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

54.3.20 SAI MCLK Divide Register (I2Sx_MDR)

The MCLK Divide Register (MDR) configures the MCLK divide ratio. Although the MDR can be changed when the MCLK divider clock is enabled, additional writes to the MDR are blocked while MCR[DUF] is set. Writes to the MDR when the MCLK divided clock is disabled do not set MCR[DUF].

Address: 4002_F000h base + 104h offset = 4002_F104h

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

I2Sx_MDR field descriptions

Field	Description
31–20 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
19–12 FRACT	MCLK Fraction Sets the MCLK divide ratio such that: MCLK output = MCLK input * ((FRACT + 1) / (DIVIDE + 1)). FRACT must be set equal or less than the value in the DIVIDE field.

Table continues on the next page...

I2Sx_MDR field descriptions (continued)

Field	Description
	NOTE: When using fractional divide values, the MCLK duty cycle will not always be 50/50. See Audio master clock.
DIVIDE	MCLK Divide Sets the MCLK divide ratio such that: MCLK output = MCLK input * ((FRACT + 1) / (DIVIDE + 1)). FRACT must be set equal or less than the value in the DIVIDE field. **NOTE:** When using fractional divide values, the MCLK duty cycle will not always be 50/50. See Audio master clock.

54.4 Functional description

This section provides a complete functional description of the block.

54.4.1 SAI clocking

The SAI clocks include:

- The audio master clock
- The bit clock
- The bus clock

54.4.1.1 Audio master clock

The audio master clock is used to generate the bit clock when the receiver or transmitter is configured for an internally generated bit clock. The transmitter and receiver can independently select between the bus clock and up to three audio master clocks to generate the bit clock.

Each SAI peripheral can control the input clock selection, pin direction and divide ratio of one audio master clock. The input clock selection and pin direction cannot be altered if an SAI module using that audio master clock has been enabled. The MCLK divide ratio can be altered while an SAI is using that master clock, although the change in the divide ratio takes several cycles. MCR[DUF] can be polled to determine when the divide ratio change has completed.

The audio master clock generation and selection is chip-specific. Refer to chip-specific clocking information about how the audio master clocks are generated. A typical implementation appears in the following figure.

Figure 54-2. SAI master clock generation

The MCLK fractional clock divider uses both clock edges from the input clock to generate a divided down clock that will approximate the output frequency, but without creating any new clock edges. Configuring FRACT and DIVIDE to the same value will result in a divide by 1 clock, while configuring FRACT higher than DIVIDE is not supported. The duty cycle can range from 66/33 when FRACT is set to one less than DIVIDE down to 50/50 for integer divide ratios, and will approach 50/50 for large non-integer divide ratios. There is no cycle to cycle jitter or duty cycle variance when the divide ratio is an integer or half integer, otherwise the divider output will oscillate between the two divided frequencies that are the closest integer or half integer divisors of the divider input clock frequency. The maximum jitter is therefore equal to half the divider input clock period, since both edges of the input clock are used in generating the divided clock.

54.4.1.2 Bit clock

The SAI transmitter and receiver support asynchronous free-running bit clocks that can be generated internally from an audio master clock or supplied externally. There is also the option for synchronous bit clock and frame sync operation between the receiver and transmitter or between multiple SAI peripherals.

Externally generated bit clocks must be:

- Enabled before the SAI transmitter or receiver is enabled
- Disabled after the SAI transmitter or receiver is disabled and completes its current frames

If the SAI transmitter or receiver is using an externally generated bit clock in asynchronous mode and that bit clock is generated by an SAI that is disabled in stop mode, then the transmitter or receiver should be disabled by software before entering stop mode. This issue does not apply when the transmitter or receiver is in a synchronous mode because all synchronous SAIs are enabled and disabled simultaneously.

54.4.1.3 Bus clock

The bus clock is used by the control and configuration registers and to generate synchronous interrupts and DMA requests.

NOTE

Although there is no specific minimum bus clock frequency specified, the bus clock frequency must be fast enough (relative to the bit clock frequency) to ensure that the FIFOs can be serviced, without generating either a transmitter FIFO underrun or receiver FIFO overflow condition.

54.4.2 SAI resets

The SAI is asynchronously reset on system reset. The SAI has a software reset and a FIFO reset.

54.4.2.1 Software reset

The SAI transmitter includes a software reset that resets all transmitter internal logic, including the bit clock generation, status flags, and FIFO pointers. It does not reset the configuration registers. The software reset remains asserted until cleared by software.

The SAI receiver includes a software reset that resets all receiver internal logic, including the bit clock generation, status flags and FIFO pointers. It does not reset the configuration registers. The software reset remains asserted until cleared by software.

54.4.2.2 FIFO reset

The SAI transmitter includes a FIFO reset that synchronizes the FIFO write pointer to the same value as the FIFO read pointer. This empties the FIFO contents and is to be used after TCSR[FEF] is set, and before the FIFO is re-initialized and TCSR[FEF] is cleared. The FIFO reset is asserted for one cycle only.

The SAI receiver includes a FIFO reset that synchronizes the FIFO read pointer to the same value as the FIFO write pointer. This empties the FIFO contents and is to be used after the RCSR[FEF] is set and any remaining data has been read from the FIFO, and before the RCSR[FEF] is cleared. The FIFO reset is asserted for one cycle only.

54.4.3 Synchronous modes

The SAI transmitter and receiver can operate synchronously to each other.

54.4.3.1 Synchronous mode

The SAI transmitter and receiver can be configured to operate with synchronous bit clock and frame sync.

If the transmitter bit clock and frame sync are to be used by both the transmitter and receiver:

- The transmitter must be configured for asynchronous operation and the receiver for synchronous operation.
- In synchronous mode, the receiver is enabled only when both the transmitter and receiver are enabled.
- It is recommended that the transmitter is the last enabled and the first disabled.

If the receiver bit clock and frame sync are to be used by both the transmitter and receiver:

- The receiver must be configured for asynchronous operation and the transmitter for synchronous operation.
- In synchronous mode, the transmitter is enabled only when both the receiver and transmitter are both enabled.
- It is recommended that the receiver is the last enabled and the first disabled.

When operating in synchronous mode, only the bit clock, frame sync, and transmitter/receiver enable are shared. The transmitter and receiver otherwise operate independently, although configuration registers must be configured consistently across both the transmitter and receiver.

54.4.4 Frame sync configuration

When enabled, the SAI continuously transmits and/or receives frames of data. Each frame consists of a fixed number of words and each word consists of a fixed number of bits. Within each frame, any given word can be masked causing the receiver to ignore that word and the transmitter to tri-state for the duration of that word.

The frame sync signal is used to indicate the start of each frame. A valid frame sync requires a rising edge (if active high) or falling edge (if active low) to be detected and the transmitter or receiver cannot be busy with a previous frame. A valid frame sync is also ignored (slave mode) or not generated (master mode) for the first four bit clock cycles after enabling the transmitter or receiver.

The transmitter and receiver frame sync can be configured independently with any of the following options:

- Externally generated or internally generated
- Active high or active low
- Assert with the first bit in frame or asserts one bit early
- Assert for a duration between 1 bit clock and the first word length
- Frame length from 1 to 16 words per frame
- Word length to support 8 to 32 bits per word
    - First word length and remaining word lengths can be configured separately
- Words can be configured to transmit/receive MSB first or LSB first

These configuration options cannot be changed after the SAI transmitter or receiver is enabled.

54.4.5 Data FIFO

Each transmit and receive channel includes a FIFO of size 8 × 32-bit. The FIFO data is accessed using the SAI Transmit/Receive Data Registers.

54.4.5.1 Data alignment

Data in the FIFO can be aligned anywhere within the 32-bit wide register through the use of the First Bit Shifted configuration field, which selects the bit index (between 31 and 0) of the first bit shifted.

Examples of supported data alignment and the required First Bit Shifted configuration are illustrated in Figure 54-3 for LSB First configurations and Figure 54-4 for MSB First configurations.

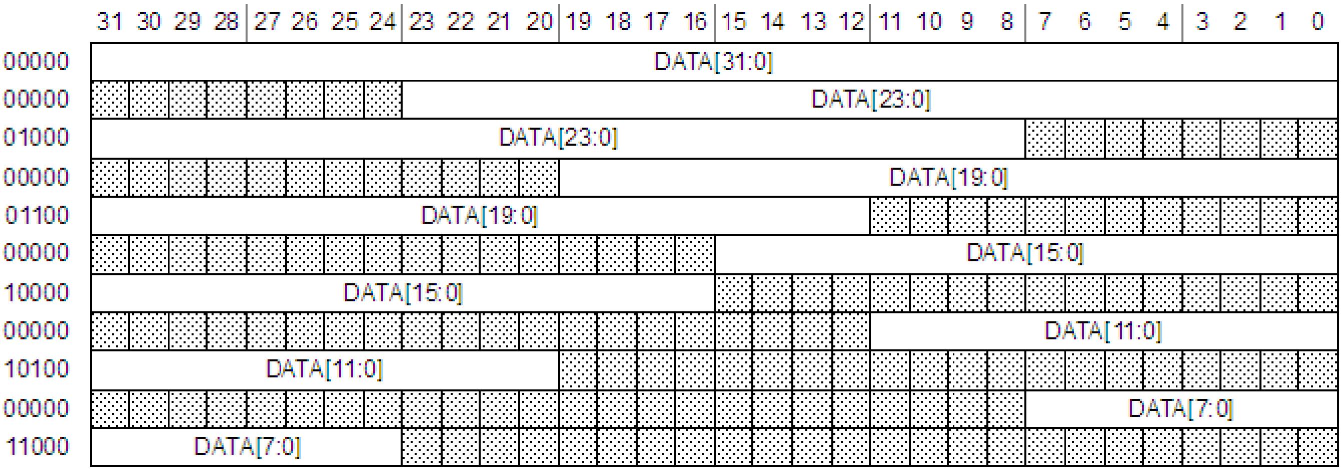


Figure 54-3. SAI first bit shifted, LSB first

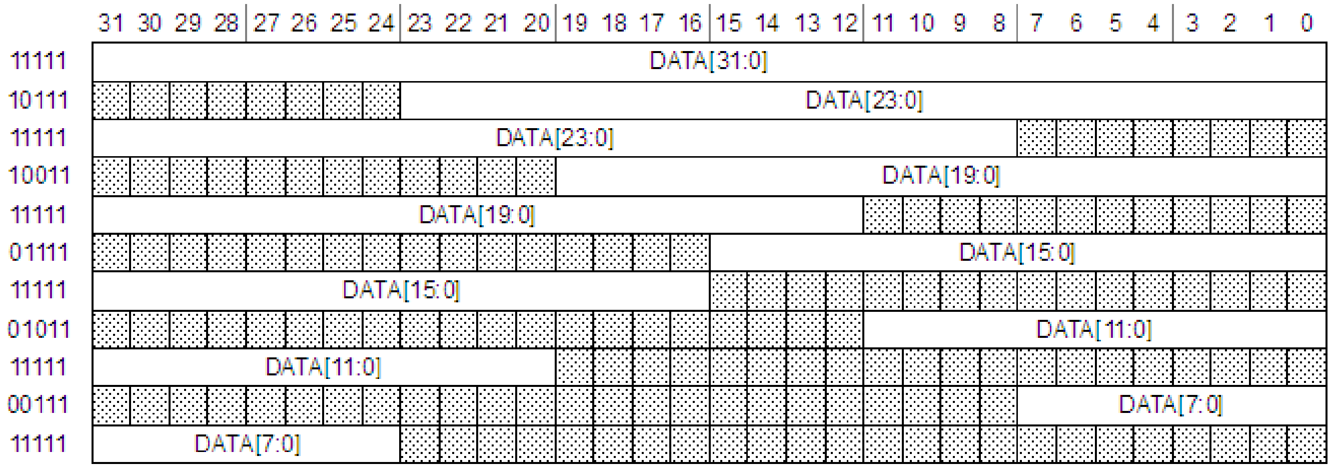


Figure 54-4. SAI first bit shifted, MSB first

54.4.5.2 FIFO pointers

When writing to a TDR, the WFP of the corresponding TFR increments after each valid write. The SAI supports 8-bit, 16-bit and 32-bit writes to the TDR and the FIFO pointer will increment after each individual write. Note that 8-bit writes should only be used when transmitting up to 8-bit data and 16-bit writes should only be used when transmitting up to 16-bit data.

Writes to a TDR are ignored if the corresponding bit of TCR3[TCE] is clear or if the FIFO is full. If the Transmit FIFO is empty, the TDR must be written at least three bit clocks before the start of the next unmasked word to avoid a FIFO underrun.

When reading an RDR, the RFP of the corresponding RFR increments after each valid read. The SAI supports 8-bit, 16-bit and 32-bit reads from the RDR and the FIFO pointer will increment after each individual read. Note that 8-bit reads should only be used when receiving up to 8-bit data and 16-bit reads should only be used when receiving up to 16-bit data.

Reads from an RDR are ignored if the corresponding bit of RCR3[RCE] is clear or if the FIFO is empty. If the Receive FIFO is full, the RDR must be read at least three bit clocks before the end of an unmasked word to avoid a FIFO overrun.

54.4.6 Word mask register

The SAI transmitter and receiver each contain a word mask register, namely TMR and RMR, that can be used to mask any word in the frame. Because the word mask register is double buffered, software can update it before the end of each frame to mask a particular word in the next frame.

The TMR causes the Transmit Data pin to be tri-stated for the length of each selected word and the transmit FIFO is not read for masked words.

The RMR causes the received data for each selected word to be discarded and not written to the receive FIFO.

54.4.7 Interrupts and DMA requests

The SAI transmitter and receiver generate separate interrupts and separate DMA requests, but support the same status flags. Asynchronous versions of the transmitter and receiver interrupts are generated to wake up the CPU from stop mode.

54.4.7.1 FIFO request flag

The FIFO request flag is set based on the number of entries in the FIFO and the FIFO watermark configuration.

The transmit FIFO request flag is set when the number of entries in any of the enabled transmit FIFOs is less than or equal to the transmit FIFO watermark configuration and is cleared when the number of entries in each enabled transmit FIFO is greater than the transmit FIFO watermark configuration.

The receive FIFO request flag is set when the number of entries in any of the enabled receive FIFOs is greater than the receive FIFO watermark configuration and is cleared when the number of entries in each enabled receive FIFO is less than or equal to the receive FIFO watermark configuration.

The FIFO request flag can generate an interrupt or a DMA request.

54.4.7.2 FIFO warning flag

The FIFO warning flag is set based on the number of entries in the FIFO.

The transmit warning flag is set when the number of entries in any of the enabled transmit FIFOs is empty and is cleared when the number of entries in each enabled transmit FIFO is not empty.

The receive warning flag is set when the number of entries in any of the enabled receive FIFOs is full and is cleared when the number of entries in each enabled receive FIFO is not full.

The FIFO warning flag can generate an Interrupt or a DMA request.

54.4.7.3 FIFO error flag

The transmit FIFO error flag is set when the any of the enabled transmit FIFOs underflow. After it is set, all enabled transmit channels repeat the last valid word read from the transmit FIFO until TCSR[FEF] is cleared and the next transmit frame starts. All enabled transmit FIFOs must be reset and initialized with new data before TCSR[FEF] is cleared.

RCSR[FEF] is set when the any of the enabled receive FIFOs overflow. After it is set, all enabled receive channels discard received data until RCSR[FEF] is cleared and the next next receive frame starts. All enabled receive FIFOs should be emptied before RCSR[FEF] is cleared.

The FIFO error flag can generate only an interrupt.

54.4.7.4 Sync error flag

The sync error flag, TCSR[SEF] or RCSR[SEF], is set when configured for an externally generated frame sync and the external frame sync asserts when the transmitter or receiver is busy with the previous frame. The external frame sync assertion is ignored and the sync error flag is set. When the sync error flag is set, the transmitter or receiver continues checking for frame sync assertion when idle or at the end of each frame.

The sync error flag can generate an interrupt only.

54.4.7.5 Word start flag

The word start flag is set at the start of the second bit clock for the selected word, as configured by the Word Flag register field.

The word start flag can generate an interrupt only.

Chapter 55
MCU: General-Purpose Input/Output (GPIO)

55.1 Introduction

NOTE

> For the chip-specific implementation details of this module's instances, see the chip configuration information.

The GPIO registers support 8-bit, 16-bit or 32-bit accesses.

The GPIO data direction and output data registers control the direction and output data of each pin when the pin is configured for the GPIO function. The GPIO input data register displays the logic value on each pin when the pin is configured for any digital function, provided the corresponding Port Control and Interrupt module for that pin is enabled.

Efficient bit manipulation of the general-purpose outputs is supported through the addition of set, clear, and toggle write-only registers for each port output data register.

55.1.1 Features

Features of the GPIO module include:

- Port Data Input register visible in all digital pin-multiplexing modes
- Port Data Output register with corresponding set/clear/toggle registers
- Port Data Direction register

NOTE

> The GPIO module is clocked by system clock.

55.1.2 Modes of operation

The following table depicts different modes of operation and the behavior of the GPIO module in these modes.

Table 55-1. Modes of operation

Modes of operation	Description
Run	The GPIO module operates normally.
Wait	The GPIO module operates normally.
Stop	The GPIO module is disabled.
Debug	The GPIO module operates normally.

55.1.3 GPIO signal descriptions

Table 55-2. GPIO signal descriptions

GPIO signal descriptions	Description	I/O
PORTA31–PORTA0	General-purpose input/output	I/O
PORTB31–PORTB0	General-purpose input/output	I/O
PORTC31–PORTC0	General-purpose input/output	I/O
PORTD31–PORTD0	General-purpose input/output	I/O
PORTE31–PORTE0	General-purpose input/output	I/O

NOTE

Not all pins within each port are implemented on each device. See the chapter on signal multiplexing for the number of GPIO ports available in the device.

55.1.3.1 Detailed signal description

Table 55-3. GPIO interface-detailed signal descriptions

Signal	I/O	Description	
PORTA31–PORTA0 PORTB31–PORTB0 PORTC31–PORTC0 PORTD31–PORTD0 PORTE31–PORTE0	I/O	General-purpose input/output	
		State meaning	Asserted: The pin is logic 1. Deasserted: The pin is logic 0.
		Timing	Assertion: When output, this signal occurs on the rising-edge of the system clock. For input, it may occur at any time and input may be asserted asynchronously to the system clock. Deassertion: When output, this signal occurs on the rising-edge of the system clock. For input, it may occur

Table 55-3. GPIO interface-detailed signal descriptions

Signal	I/O	Description
		at any time and input may be asserted asynchronously to the system clock.

NOTE

Not all pins within each port are implemented on each device. See the chapter on signal multiplexing for the number of GPIO ports available in the device.

55.2 Memory map and register definition

Any read or write access to the GPIO memory space that is outside the valid memory map results in a bus error.

GPIO memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
400F_F000	Port Data Output Register (GPIOA_PDOR)	32	R/W	0000_0000h	55.2.1/1456
400F_F004	Port Set Output Register (GPIOA_PSOR)	32	W (always reads 0)	0000_0000h	55.2.2/1457
400F_F008	Port Clear Output Register (GPIOA_PCOR)	32	W (always reads 0)	0000_0000h	55.2.3/1458
400F_F00C	Port Toggle Output Register (GPIOA_PTOR)	32	W (always reads 0)	0000_0000h	55.2.4/1458
400F_F010	Port Data Input Register (GPIOA_PDIR)	32	R	0000_0000h	55.2.5/1459
400F_F014	Port Data Direction Register (GPIOA_PDDR)	32	R/W	0000_0000h	55.2.6/1459
400F_F040	Port Data Output Register (GPIOB_PDOR)	32	R/W	0000_0000h	55.2.1/1456
400F_F044	Port Set Output Register (GPIOB_PSOR)	32	W (always reads 0)	0000_0000h	55.2.2/1457
400F_F048	Port Clear Output Register (GPIOB_PCOR)	32	W (always reads 0)	0000_0000h	55.2.3/1458
400F_F04C	Port Toggle Output Register (GPIOB_PTOR)	32	W (always reads 0)	0000_0000h	55.2.4/1458
400F_F050	Port Data Input Register (GPIOB_PDIR)	32	R	0000_0000h	55.2.5/1459
400F_F054	Port Data Direction Register (GPIOB_PDDR)	32	R/W	0000_0000h	55.2.6/1459

GPIO memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
400F_F080	Port Data Output Register (GPIOC_PDOR)	32	R/W	0000_0000h	55.2.1/1456
400F_F084	Port Set Output Register (GPIOC_PSOR)	32	W (always reads 0)	0000_0000h	55.2.2/1457
400F_F088	Port Clear Output Register (GPIOC_PCOR)	32	W (always reads 0)	0000_0000h	55.2.3/1458
400F_F08C	Port Toggle Output Register (GPIOC_PTOR)	32	W (always reads 0)	0000_0000h	55.2.4/1458
400F_F090	Port Data Input Register (GPIOC_PDIR)	32	R	0000_0000h	55.2.5/1459
400F_F094	Port Data Direction Register (GPIOC_PDDR)	32	R/W	0000_0000h	55.2.6/1459
400F_F0C0	Port Data Output Register (GPIOD_PDOR)	32	R/W	0000_0000h	55.2.1/1456
400F_F0C4	Port Set Output Register (GPIOD_PSOR)	32	W (always reads 0)	0000_0000h	55.2.2/1457
400F_F0C8	Port Clear Output Register (GPIOD_PCOR)	32	W (always reads 0)	0000_0000h	55.2.3/1458
400F_F0CC	Port Toggle Output Register (GPIOD_PTOR)	32	W (always reads 0)	0000_0000h	55.2.4/1458
400F_F0D0	Port Data Input Register (GPIOD_PDIR)	32	R	0000_0000h	55.2.5/1459
400F_F0D4	Port Data Direction Register (GPIOD_PDDR)	32	R/W	0000_0000h	55.2.6/1459
400F_F100	Port Data Output Register (GPIOE_PDOR)	32	R/W	0000_0000h	55.2.1/1456
400F_F104	Port Set Output Register (GPIOE_PSOR)	32	W (always reads 0)	0000_0000h	55.2.2/1457
400F_F108	Port Clear Output Register (GPIOE_PCOR)	32	W (always reads 0)	0000_0000h	55.2.3/1458
400F_F10C	Port Toggle Output Register (GPIOE_PTOR)	32	W (always reads 0)	0000_0000h	55.2.4/1458
400F_F110	Port Data Input Register (GPIOE_PDIR)	32	R	0000_0000h	55.2.5/1459
400F_F114	Port Data Direction Register (GPIOE_PDDR)	32	R/W	0000_0000h	55.2.6/1459

55.2.1 Port Data Output Register (GPIOx_PDOR)

This register configures the logic levels that are driven on each general-purpose output pins.

NOTE

Do not modify pin configuration registers associated with pins not available in your selected package. All unbonded pins not available in your package will default to DISABLE state for lowest power consumption.

Address: Base address + 0h offset

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

GPIOx_PDOR field descriptions

Field	Description
PDO	Port Data Output Register bits for unbonded pins return a undefined value when read. 0 Logic level 0 is driven on pin, provided pin is configured for general-purpose output. 1 Logic level 1 is driven on pin, provided pin is configured for general-purpose output.

55.2.2 Port Set Output Register (GPIOx_PSOR)

This register configures whether to set the fields of the PDOR.

Address: Base address + 4h offset

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

GPIOx_PSOR field descriptions

Field	Description
PTSO	Port Set Output Writing to this register will update the contents of the corresponding bit in the PDOR as follows: 0 Corresponding bit in PDORn does not change. 1 Corresponding bit in PDORn is set to logic 1.

55.2.3 Port Clear Output Register (GPIOx_PCOR)

This register configures whether to clear the fields of PDOR.

Address: Base address + 8h offset

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

GPIOx_PCOR field descriptions

Field	Description
PTCO	Port Clear Output Writing to this register will update the contents of the corresponding bit in the Port Data Output Register (PDOR) as follows: 0 Corresponding bit in PDORn does not change. 1 Corresponding bit in PDORn is cleared to logic 0.

55.2.4 Port Toggle Output Register (GPIOx_PTOR)

Address: Base address + Ch offset

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

GPIOx_PTOR field descriptions

Field	Description
PTTO	Port Toggle Output Writing to this register will update the contents of the corresponding bit in the PDOR as follows: 0 Corresponding bit in PDORn does not change. 1 Corresponding bit in PDORn is set to the inverse of its existing logic state.

55.2.5 Port Data Input Register (GPIOx_PDIR)

NOTE

Do not modify pin configuration registers associated with pins not available in your selected package. All unbonded pins not available in your package will default to DISABLE state for lowest power consumption.

Address: Base address + 10h offset

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

GPIOx_PDIR field descriptions

Field	Description
PDI	Port Data Input Reads 0 at the unimplemented pins for a particular device. Pins that are not configured for a digital function read 0. If the Port Control and Interrupt module is disabled, then the corresponding bit in PDIR does not update. 0 Pin logic level is logic 0, or is not configured for use by digital function. 1 Pin logic level is logic 1.

55.2.6 Port Data Direction Register (GPIOx_PDDR)

The PDDR configures the individual port pins for input or output.

Address: Base address + 14h offset

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

GPIOx_PDDR field descriptions

Field	Description
PDD	Port Data Direction Configures individual port pins for input or output. 0 Pin is configured as general-purpose input, for the GPIO function. 1 Pin is configured as general-purpose output, for the GPIO function.

GPIOx_PDDR field descriptions (continued)

Field	Description

55.3 Functional description

55.3.1 General-purpose input

The logic state of each pin is available via the Port Data Input registers, provided the pin is configured for a digital function and the corresponding Port Control and Interrupt module is enabled.

The Port Data Input registers return the synchronized pin state after any enabled digital filter in the Port Control and Interrupt module. The input pin synchronizers are shared with the Port Control and Interrupt module, so that if the corresponding Port Control and Interrupt module is disabled, then synchronizers are also disabled. This reduces power consumption when a port is not required for general-purpose input functionality.

55.3.2 General-purpose output

The logic state of each pin can be controlled via the port data output registers and port data direction registers, provided the pin is configured for the GPIO function. The following table depicts the conditions for a pin to be configured as input/output.

If	Then
A pin is configured for the GPIO function and the corresponding port data direction register bit is clear.	The pin is configured as an input.
A pin is configured for the GPIO function and the corresponding port data direction register bit is set.	The pin is configured as an output and and the logic state of the pin is equal to the corresponding port data output register.

To facilitate efficient bit manipulation on the general-purpose outputs, pin data set, pin data clear, and pin data toggle registers exist to allow one or more outputs within one port to be set, cleared, or toggled from a single register write.

The corresponding Port Control and Interrupt module does not need to be enabled to update the state of the port data direction registers and port data output registers including the set/clear/toggle registers.

Chapter 56
MCU: JTAG Controller (JTAGC)

56.1 Introduction

NOTE

> For the chip-specific implementation details of this module's instances, see the chip configuration information.

The JTAGC block provides the means to test chip functionality and connectivity while remaining transparent to system logic when not in test mode. Testing is performed via a boundary scan technique, as defined in the IEEE 1149.1-2001 standard. All data input to and output from the JTAGC block is communicated in serial format.

56.1.1 Block diagram

The following is a simplified block diagram of the JTAG Controller (JTAGC) block. Refer to the chip-specific configuration information as well as Register description for more information about the JTAGC registers.

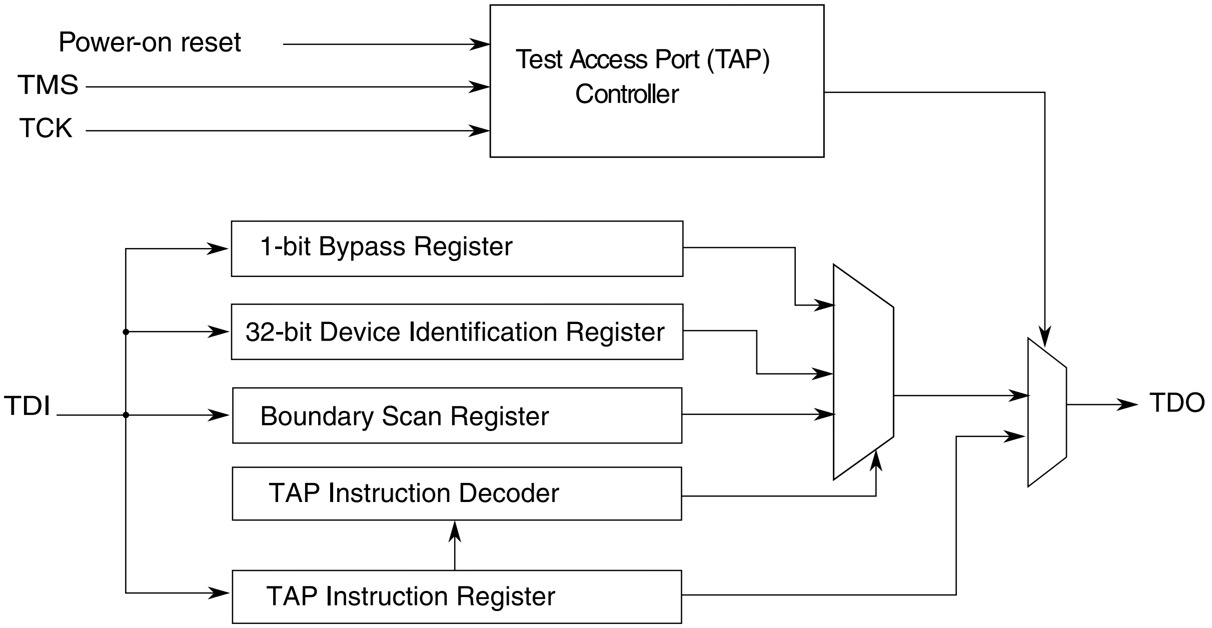


Figure 56-1. JTAG (IEEE 1149.1) block diagram

56.1.2 Features

The JTAGC block is compliant with the IEEE 1149.1-2001 standard, and supports the following features:

- IEEE 1149.1-2001 Test Access Port (TAP) interface
  - 4 pins (TDI, TMS, TCK, and TDO)
- Instruction register that supports several IEEE 1149.1-2001 defined instructions as well as several public and private device-specific instructions. Refer to Table 56-3 for a list of supported instructions.
- Bypass register, boundary scan register, and device identification register.
- TAP controller state machine that controls the operation of the data registers, instruction register and associated circuitry.

56.1.3 Modes of operation

The JTAGC block uses a power-on reset indication as its primary reset signals. Several IEEE 1149.1-2001 defined test modes are supported, as well as a bypass mode.

56.1.3.1 Reset

The JTAGC block is placed in reset when either power-on reset is asserted, or the TMS input is held high for enough consecutive rising edges of TCK to sequence the TAP controller state machine into the Test-Logic-Reset state. Holding TMS high for five consecutive rising edges of TCK guarantees entry into the Test-Logic-Reset state regardless of the current TAP controller state. Asserting power-on reset results in asynchronous entry into the reset state. While in reset, the following actions occur:

- The TAP controller is forced into the Test-Logic-Reset state, thereby disabling the test logic and allowing normal operation of the on-chip system logic to continue unhindered
- The instruction register is loaded with the IDCODE instruction

56.1.3.2 IEEE 1149.1-2001 defined test modes

The JTAGC block supports several IEEE 1149.1-2001 defined test modes. A test mode is selected by loading the appropriate instruction into the instruction register while the JTAGC is enabled. Supported test instructions include EXTEST, HIGHZ, CLAMP, SAMPLE and SAMPLE/PRELOAD. Each instruction defines the set of data register(s) that may operate and interact with the on-chip system logic while the instruction is current. Only one test data register path is enabled to shift data between TDI and TDO for each instruction.

The boundary scan register is enabled for serial access between TDI and TDO when the EXTEST, SAMPLE or SAMPLE/PRELOAD instructions are active. The single-bit bypass register shift stage is enabled for serial access between TDI and TDO when the BYPASS, HIGHZ, CLAMP or reserved instructions are active. The functionality of each test mode is explained in more detail in JTAGC block instructions.

56.1.3.3 Bypass mode

When no test operation is required, the BYPASS instruction can be loaded to place the JTAGC block into bypass mode. While in bypass mode, the single-bit bypass shift register is used to provide a minimum-length serial path to shift data between TDI and TDO.

56.2 External signal description

The JTAGC consists of a set of signals that connect to off chip development tools and allow access to test support functions. The JTAGC signals are outlined in the following table and described in the following sections.

Table 56-1. JTAG signal properties

Name	I/O	Function	Reset State	Pull
TCK	Input	Test Clock	—	Down
TDI	Input	Test Data In	—	Up
TDO	Output	Test Data Out	High Z	—
TMS	Input	Test Mode Select	—	Up

56.2.1 TCK—Test clock input

Test Clock Input (TCK) is an input pin used to synchronize the test logic and control register access through the TAP.

56.2.2 TDI—Test data input

Test Data Input (TDI) is an input pin that receives serial test instructions and data. TDI is sampled on the rising edge of TCK.

56.2.3 TDO—Test data output

Test Data Output (TDO) is an output pin that transmits serial output for test instructions and data. TDO is three-stateable and is actively driven only in the Shift-IR and Shift-DR states of the TAP controller state machine, which is described in TAP controller state machine.

56.2.4 TMS—Test mode select

Test Mode Select (TMS) is an input pin used to sequence the IEEE 1149.1-2001 test control state machine. TMS is sampled on the rising edge of TCK.

56.3 Register description

This section provides a detailed description of the JTAGC block registers accessible through the TAP interface, including data registers and the instruction register. Individual bit-level descriptions and reset states of each register are included. These registers are not memory-mapped and can only be accessed through the TAP.

56.3.1 Instruction register

The JTAGC block uses a 4-bit instruction register as shown in the following figure. The instruction register allows instructions to be loaded into the block to select the test to be performed or the test data register to be accessed or both. Instructions are shifted in through TDI while the TAP controller is in the Shift-IR state, and latched on the falling edge of TCK in the Update-IR state. The latched instruction value can only be changed in the Update-IR and Test-Logic-Reset TAP controller states. Synchronous entry into the Test-Logic-Reset state results in the IDCODE instruction being loaded on the falling edge of TCK. Asynchronous entry into the Test-Logic-Reset state results in asynchronous loading of the IDCODE instruction. During the Capture-IR TAP controller state, the instruction shift register is loaded with the value 0001b , making this value the register's read value when the TAP controller is sequenced into the Shift-IR state.

	3	2	1	0
R	0	0	0	1
W	Instruction Code			
Reset:	0	0	0	1

Figure 56-2. Instruction register

56.3.2 Bypass register

The bypass register is a single-bit shift register path selected for serial data transfer between TDI and TDO when the BYPASS, CLAMP, HIGHZ or reserve instructions are active. After entry into the Capture-DR state, the single-bit shift register is set to a logic 0. Therefore, the first bit shifted out after selecting the bypass register is always a logic 0.

56.3.3 Device identification register

The device identification (JTAG ID) register, shown in the following figure, allows the revision number, part number, manufacturer, and design center responsible for the design of the part to be determined through the TAP. The device identification register is selected for serial data transfer between TDI and TDO when the IDCODE instruction is active. Entry into the Capture-DR state while the device identification register is selected loads the IDCODE into the shift register to be shifted out on TDO in the Shift-DR state. No action occurs in the Update-DR state.

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Part Revision Number				Design Center						Part Identification Number					
W																
Reset	PRN				DC						PIN					

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Part Identification Number				Manufacturer Identity Code											1
W																
Reset	PIN (contd.)				MIC											1

The following table describes the device identification register functions.

Table 56-2. Device identification register field descriptions

Field	Description
PRN	Part Revision Number. Contains the revision number of the part. Value is 0x0.
DC	Design Center. Indicates the design center. Value is 0x2C.
PIN	Part Identification Number. Contains the part number of the device. Value is TBD.
MIC	Manufacturer Identity Code. Contains the reduced Joint Electron Device Engineering Council (JEDEC) ID. Value is 0x00E.
IDCODE ID	IDCODE Register ID. Identifies this register as the device identification register and not the bypass register. Always set to 1.

56.3.4 Boundary scan register

The boundary scan register is connected between TDI and TDO when the EXTEST, SAMPLE or SAMPLE/PRELOAD instructions are active. It is used to capture input pin data, force fixed values on output pins, and select a logic value and direction for bidirectional pins. Each bit of the boundary scan register represents a separate boundary scan register cell, as described in the IEEE 1149.1-2001 standard and discussed in Boundary scan. The size of the boundary scan register and bit ordering is device-dependent and can be found in the device BSDL file.

56.4 Functional description

This section explains the JTAGC functional description.

56.4.1 JTAGC reset configuration

While in reset, the TAP controller is forced into the Test-Logic-Reset state, thus disabling the test logic and allowing normal operation of the on-chip system logic. In addition, the instruction register is loaded with the IDCODE instruction.

56.4.2 IEEE 1149.1-2001 (JTAG) Test Access Port

The JTAGC block uses the IEEE 1149.1-2001 TAP for accessing registers. This port can be shared with other TAP controllers on the MCU. Ownership of the port is determined by the value of the currently loaded instruction.

Data is shifted between TDI and TDO though the selected register starting with the least significant bit, as illustrated in the following figure. This applies for the instruction register, test data registers, and the bypass register.

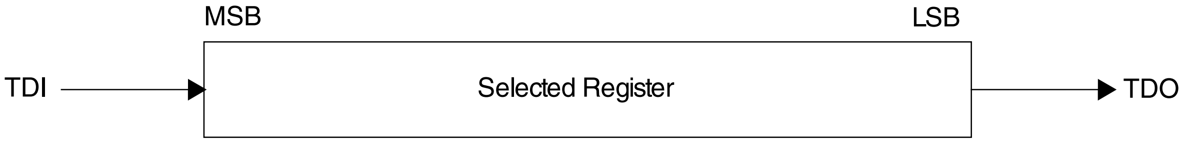
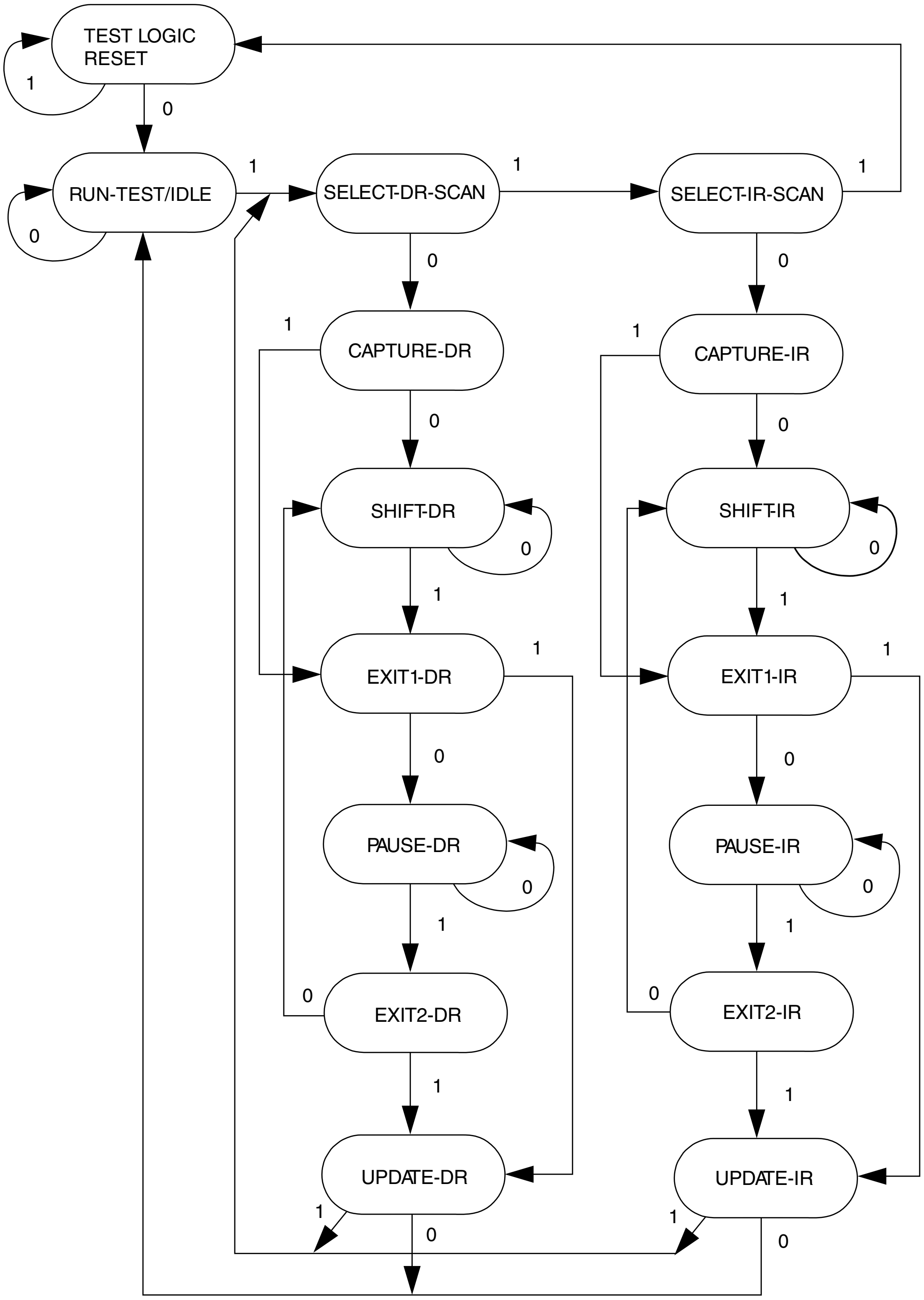


Figure 56-3. Shifting data through a register

56.4.3 TAP controller state machine

The TAP controller is a synchronous state machine that interprets the sequence of logical values on the TMS pin. The following figure shows the machine's states. The value shown next to each state is the value of the TMS signal sampled on the rising edge of the TCK signal. As the following figure shows, holding TMS at logic 1 while clocking TCK through a sufficient number of rising edges also causes the state machine to enter the Test-Logic-Reset state.



The value shown adjacent to each state transition in this figure represents the value of TMS at the time of a rising edge of TCK.

Figure 56-4. IEEE 1149.1-2001 TAP controller finite state machine

56.4.3.1 Enabling the TAP controller

The JTAGC TAP controller is enabled by setting the JTAGC enable to a logic 1 value.

56.4.3.2 Selecting an IEEE 1149.1-2001 register

Access to the JTAGC data registers is achieved by loading the instruction register with any of the JTAGC block instructions while the JTAGC is enabled. Instructions are shifted in via the Select-IR-Scan path and loaded in the Update-IR state. At this point, all data register access is performed via the Select-DR-Scan path.

The Select-DR-Scan path is used to read or write the register data by shifting in the data (LSB first) during the Shift-DR state. When reading a register, the register value is loaded into the IEEE 1149.1-2001 shifter during the Capture-DR state. When writing a register, the value is loaded from the IEEE 1149.1-2001 shifter to the register during the Update-DR state. When reading a register, there is no requirement to shift out the entire register contents. Shifting may be terminated once the required number of bits have been acquired.

56.4.4 JTAGC block instructions

The JTAGC block implements the IEEE 1149.1-2001 defined instructions listed in the following table. This section gives an overview of each instruction; refer to the IEEE 1149.1-2001 standard for more details. All undefined opcodes are reserved.

Table 56-3. 4-bit JTAG instructions

Instruction	Code[3:0]	Instruction summary
IDCODE	0000	Selects device identification register for shift
EZPORT	0001	Enables the EZPORT function for the SoC
SAMPLE/PRELOAD	0010	Selects boundary scan register for shifting, sampling, and preloading without disturbing functional operation
SAMPLE	0011	Selects boundary scan register for shifting and sampling without disturbing functional operation
EXTEST	0100	Selects boundary scan register and applies preloaded values to output pins. **NOTE:** Execution of this instruction asserts functional reset.
Factory debug reserved	0101	Intended for factory debug only
Factory debug reserved	0110	Intended for factory debug only
Factory debug reserved	0111	Intended for factory debug only
ARM JTAG-DP Reserved	1000	This instruction goes the ARM JTAG-DP controller. See the ARM JTAG-DP documentation for more information.
HIGHZ	1001	Selects bypass register and three-states all output pins. **NOTE:** Execution of this instruction asserts functional reset.

Table continues on the next page...

Table 56-3. 4-bit JTAG instructions (continued)

Instruction	Code[3:0]	Instruction summary
ARM JTAG-DP Reserved	1010	This instruction goes the ARM JTAG-DP controller. See the ARM JTAG-DP documentation for more information.
ARM JTAG-DP Reserved	1011	This instruction goes the ARM JTAG-DP controller. See the ARM JTAG-DP documentation for more information.
CLAMP	1100	Selects bypass register and applies preloaded values to output pins. **NOTE:** Execution of this instruction asserts functional reset.
EZPORT	1101	Enables the EZPORT function for the SoC
ARM JTAG-DP Reserved	1110	This instruction goes the ARM JTAG-DP controller. See the ARM JTAG-DP documentation for more information.
BYPASS	1111	Selects bypass register for data operations

56.4.4.1 IDCODE instruction

IDCODE selects the 32-bit device identification register as the shift path between TDI and TDO. This instruction allows interrogation of the MCU to determine its version number and other part identification data. IDCODE is the instruction placed into the instruction register when the JTAGC block is reset.

56.4.4.2 EZPORT instruction

The EZPORT instruction allows for the EZPORT module to program the on-chip flash from a simple 4-pin interface. The JTAGC forces the core into a reset state and forces the EZPORT mode select/chip select low. In this mode, the flash can be programmed through the JTAG test port pins, which are connected to the EZPORT module.

56.4.4.3 SAMPLE/PRELOAD instruction

The SAMPLE/PRELOAD instruction has two functions:

- The SAMPLE portion of the instruction obtains a sample of the system data and control signals present at the MCU input pins and just before the boundary scan register cells at the output pins. This sampling occurs on the rising edge of TCK in the Capture-DR state when the SAMPLE/PRELOAD instruction is active. The sampled data is viewed by shifting it through the boundary scan register to the TDO output during the Shift-DR state. Both the data capture and the shift operation are transparent to system operation.

- The PRELOAD portion of the instruction initializes the boundary scan register cells before selecting the EXTEST or CLAMP instructions to perform boundary scan tests. This is achieved by shifting in initialization data to the boundary scan register during the Shift-DR state. The initialization data is transferred to the parallel outputs of the boundary scan register cells on the falling edge of TCK in the Update-DR state. The data is applied to the external output pins by the EXTEST or CLAMP instruction. System operation is not affected.

56.4.4.4 SAMPLE instruction

The SAMPLE instruction obtains a sample of the system data and control signals present at the MCU input pins and just before the boundary scan register cells at the output pins. This sampling occurs on the rising edge of TCK in the Capture-DR state when the SAMPLE instruction is active. The sampled data is viewed by shifting it through the boundary scan register to the TDO output during the Shift-DR state. There is no defined action in the Update-DR state. Both the data capture and the shift operation are transparent to system operation.

56.4.4.5 EXTEST External test instruction

EXTEST selects the boundary scan register as the shift path between TDI and TDO. It allows testing of off-chip circuitry and board-level interconnections by driving preloaded data contained in the boundary scan register onto the system output pins. Typically, the preloaded data is loaded into the boundary scan register using the SAMPLE/PRELOAD instruction before the selection of EXTEST. EXTEST asserts the internal system reset for the MCU to force a predictable internal state while performing external boundary scan operations.

56.4.4.6 HIGHZ instruction

HIGHZ selects the bypass register as the shift path between TDI and TDO. While HIGHZ is active all output drivers are placed in an inactive drive state (e.g., high impedance). HIGHZ also asserts the internal system reset for the MCU to force a predictable internal state.

56.4.4.7 CLAMP instruction

CLAMP allows the state of signals driven from MCU pins to be determined from the boundary scan register while the bypass register is selected as the serial path between TDI and TDO. CLAMP enhances test efficiency by reducing the overall shift path to a single bit (the bypass register) while conducting an EXTEST type of instruction through the boundary scan register. CLAMP also asserts the internal system reset for the MCU to force a predictable internal state.

56.4.4.8 BYPASS instruction

BYPASS selects the bypass register, creating a single-bit shift register path between TDI and TDO. BYPASS enhances test efficiency by reducing the overall shift path when no test operation of the MCU is required. This allows more rapid movement of test data to and from other components on a board that are required to perform test functions. While the BYPASS instruction is active the system logic operates normally.

56.4.5 Boundary scan

The boundary scan technique allows signals at component boundaries to be controlled and observed through the shift-register stage associated with each pad. Each stage is part of a larger boundary scan register cell, and cells for each pad are interconnected serially to form a shift-register chain around the border of the design. The boundary scan register consists of this shift-register chain, and is connected between TDI and TDO when the EXTEST, SAMPLE, or SAMPLE/PRELOAD instructions are loaded. The shift-register chain contains a serial input and serial output, as well as clock and control signals.

56.5 Initialization/Application information

The test logic is a static logic design, and TCK can be stopped in either a high or low state without loss of data. However, the system clock is not synchronized to TCK internally. Any mixed operation using both the test logic and the system functional logic requires external synchronization.

To initialize the JTAGC block and enable access to registers, the following sequence is required:

1. Place the JTAGC in reset through TAP controller state machine transitions controlled by TMS
2. Load the appropriate instruction for the test or action to be performed

Appendix A
Release Notes for Revision 3

A.1 General changes throughout

- Removed support of VREF and UART3 throughout the manual.
- Updated Figure 2-2 Modem Simplified Block Diagram.
- Corrected a few pin names with prefix "m" in Table 3-1.
- Removed support of USB_CLKIN through out.
- Updated missing reset values of Modem and Indirect Modem registers.
- Added the following registers to the Indirect Modem memory map: TMR_PRESCALE, RX_BYTE_COUNT, RX_WTR_MARK, TXDELAY, ACKDELAY.
- Updated register bit map structure of XTAL Control (Indirect Modem_XTAL_CTRL) register.
- Updated "ADC Reference Options" and "CMP external references" topics.
- Added chip-specific topic, Voltage Reference Selection, to Chapter 40 MCU: Analog-to-Digital Converter (ADC) to include information about voltage references specific to MKW2xD.
- Removed support of Enable Asynchronous Request in Stop register from DMA (DMA_EARS).
- Added support of OTG to the USBOTG chapter and removed support of Keep Alive feature from the chapter.
- Added MCLK Divide Register (MDR) to I2S memory map.

How to Reach Us:

Home Page:
nxp.com

Web Support:
nxp.com/support

Information in this document is provided solely to enable system and software implementers to use NXP products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits based on the information in this document. NXP reserves the right to make changes without further notice to any products herein.

NXP makes no warranty, representation, or guarantee regarding the suitability of its products for any particular purpose, nor does NXP assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. “Typical” parameters that may be provided in NXP data sheets and/or specifications can and do vary in different applications, and actual performance may vary over time. All operating parameters, including “typicals,” must be validated for each customer application by customer’s technical experts. NXP does not convey any license under its patent rights nor the rights of others. NXP sells products pursuant to standard terms and conditions of sale, which can be found at the following address: nxp.com/SalesTermsandConditions.