WPR1516 Sub-Family Reference Manual

Supports: MWPR1516CFM(R) and MWPR1516CALR

Document Number: WPR1516RM
Rev 2, 12/2014

Contents

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

Chapter 1
About This Document

Chapter 2
Introduction

Chapter 3
Core Overview

Chapter 4
Memory and Memory Map

Chapter 5
Clock Distribution

Chapter 6
Reset and Boot

Chapter 11
Port Control (PORT)

Chapter 12
System Integration Module (SIM)

Chapter 13
Power Management Controller (PMC)

Chapter 14
Miscellaneous Control Module (MCM)

Chapter 15
Peripheral Bridge (AIPS-Lite)

Chapter 16
Crossbar Switch Lite (AXBS-Lite)

Chapter 17
Watchdog timer (WDOG)

Chapter 18
Flash Memory Module (FTMRE)

Chapter 19
Internal Clock Source (ICS)

Chapter 20
Oscillator (OSC)

Chapter 21
Interrupt (IRQ)

Chapter 22
Analog-to-Digital Converter (ADC)

Chapter 23
Analog comparator (ACMP)

Chapter 24
Programmable Gain Amplifier (PGA)

Chapter 25
FlexTimer Module (FTM)

Chapter 26
Frequency-Shift Keying Demodulation Timer (FSKDT)

Chapter 27
Communication and Clamp Controller (CNC)

Chapter 28
Linear Low Dropout Voltage Regulator Controller (LDO)

Chapter 29
Periodic Interrupt Timer (PIT)

Chapter 30
Real-Time Counter (RTC)

Chapter 31
Inter-Integrated Circuit (I2C)

Chapter 32
Universal Asynchronous Receiver/Transmitter (UART)

Chapter 33
General-Purpose Input/Output (GPIO)

Chapter 1
About This Document

1.1 Overview

1.1.1 Purpose

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

1.1.2 Audience

This document is primarily 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.

Chapter 2
Introduction

2.1 Overview

This chapter provides an overview of the WPR1516 of ARM® Cortex™-M0+ MCUs. It also presents high-level descriptions of the modules available on the devices covered by this document.

2.2 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 2-1. Module functional categories

Module category	Description
ARM Cortex-M0+ core	• 32-bit MCU core from ARM's Cortex-M class, 1.77 CoreMark®/MHz from single-cycle access memories, 24 MHz CPU frequency
System	• System integration module (SIM) • Power management and mode controllers (PMC) • Miscellaneous control module (MCM) • Peripheral bridge (AIPS) • Watchdog (WDOG)

Memories	• Internal memories include:
	  • 16 KB flash memory
	  • 4 KB SRAM

Clocks	• External crystal oscillator or resonator
	  • Low range: 31.25–39.0625 kHz
	  • High range: 4–24 MHz
	• External square wave input clock
	• Internal clock references
	  • 31.25 to 39.0625 kHz oscillator
	  • 20 kHz LPO oscillator
	• Frequency-locked loop (FLL) range: 40–50 MHz

Security	• Watchdog (WDOG) with independent clock source
UHV	• One communication and clamp controller (CNC)

Table continues on the next page...

Table 2-1. Module functional categories (continued)

Module category	Description
	• One power management controller (PMC) • One linear low dropout voltage regulator controller (LDO)
Analog	• One 12-bit analog-to-digital converters (ADC) with up to 12 channels • One analog comparators (ACMP) with internal 6-bit digital-to-analog converter (DAC) • One programmable gain amplifier (PGA) with differential input and output
Timers	• Two 2-channel FTMs with basic TPM function • One 2-channel periodic interrupt timer (PIT) • One real time clock (RTC) • One FSK demodulation timer (FSKDT) • System tick timer (SysTick)
Communications	• One inter-integrated circuit (I2C) module • One universal asynchronous receiver/transmitter (UART) module
Human-Machine Interfaces (HMI)	• General purpose input/output (GPIO) controller

2.2.1 ARM® Cortex™-M0+ Core Modules

The following core modules are available on this device.

Table 2-2. Core modules

Module	Description
ARM Cortex-M0+	The ARM Cortex-M0+ is the newest member of the Cortex M Series of processors targeting microcontroller applications focused on very cost sensitive, deterministic, interrupt driven environments. The Cortex M0+ processor is based on the ARMv6 Architecture and Thumb®-2 ISA and is 100% instruction set compatible with its predecessor, the Cortex-M0 core, and upward compatible to Cortex-M3 and M4 cores. The ARM® Cortex™-M0+ improvements include an ARMv6 Thumb-2 DSP, ported from the ARMv6-A/R profile architectures, that provide 32-bit instructions with SIMD (single instruction multiple data) DSP style multiply-accumulates and saturating arithmetic to support single cycle 32x32 multiplier.
Nested vectored interrupt controller (NVIC)	The ARMv6-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. 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.
Asynchronous wakeup interrupt controller (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. One debug interface is supported: • Serial Wire Debug (SWD)

2.2.2 System modules

The following system modules are available on this device.

Table 2-3. System modules

Module	Description
System integration module (SIM)	The SIM includes integration logic and several module configuration settings.
Power management controller (PMC)	The PMC provides the user with multiple power options. Multiple 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 reset (LVR) with reset (brownout) capability and selectable LVR trip points.
Miscellaneous control module (MCM)	The MCM includes integration logic and details.
Peripheral bridge (AIPS-Lite)	The peripheral bridge converts the ARM AHB interface to an interface to access a majority of peripherals on the device.
Watchdog (WDOG)	The WDOG monitors internal system operation and forces a reset in case of failure. It can run from an independent 20 kHz low-power oscillator with a programmable refresh window to detect deviations in program flow or system frequency.

2.2.3 Memories and memory interfaces

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

Table 2-4. Memories and memory interfaces

Module	Description
Flash memory (FTMRE)	Flash memory — 16 KB of the non-volatile flash memory that can execute program code
SRAM	4 KB internal system RAM.

2.2.4 Clocks

The following clock modules are available on this device.

Table 2-5. Clock modules

Module	Description
Internal Clock Source (ICS)	ICS module containing an internal reference clock (ICSIRCLK) and a frequency-locked-loop (FLL)

Table continues on the next page...

Table 2-5. Clock modules (continued)

Module	Description
System oscillator (OSC)	The system oscillator, in conjunction with an external crystal or resonator, generates a reference clock for the MCU.
Low-Power Oscillator (LPO)	The PMC module contains a 20 kHz low-power oscillator which acts as a standalone low-frequency clock source in all modes.

2.2.5 Security and integrity modules

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

Table 2-6. Security and integrity modules

Module	Description
Watchdog (WDOG)	The WDOG monitors internal system operation and forces a reset in case of failure. It can run from an independent 20 kHz low-power oscillator with a programmable refresh window to detect deviations in program flow or system frequency.

2.2.6 Analog modules

The following analog modules are available on this device:

Table 2-7. Analog modules

Module	Description
Analog-to-digital converters (ADC)	12-bit successive-approximation ADC module with up to 12 channels.
Analog comparators (ACMP)	One comparator with support of analog input voltages across the full range of the supply voltage and CPU interrupt. ACMP0 is further capable to trigger an ADC acquisition and FTM update.
6-bit digital-to-analog converters (DAC)	64-tap resistor ladder network which provides a selectable voltage reference for comparator.
Programmer gain amplifier (PGA)	Differential input voltage amplifier with 8 to 20 programmable gain.

2.2.7 Ultra high voltage modules

The following ultra high voltage modules are available on this device:

Table 2-8. Ultra high voltage modules

Module	Description
Communication and Clamp Controller (CNC)	The CNC module is designed for WPC-Qi compliant wireless power receiver. It acts as communications and control unit of the power receiver.
Power Management Controller (PMC)	The PMC provides internal power to other analog modules and the MCU from an external DC source. It also provides power-on reset and low voltage detection.
Linear Low Dropout Voltage Regulator Controller (LDO)	The LDO module is used to deliver power to wireless power receiver loads with the external N-FET power devices. It can provide up to 5V/3A power supply to loading system with precious voltage and current control.

2.2.8 Timer modules

The following timer modules are available on this device:

Table 2-9. Timer modules

Module	Description
FlexTimer 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 complementary outputs, or independent channels with independent outputs • Software control of PWM outputs • Configurable channel polarity • Programmable interrupt on input capture, reference compare, overflowed counter
Periodic interrupt timers (PIT)	• One general purpose interrupt timer • Interrupt timers for triggering ADC conversions • 32-bit counter resolution • Clocked by bus clock frequency

Real-time counter (RTC)	• 16-bit up-counter
	  • 16-bit modulo match limit
	  • Software controllable periodic interrupt on match
	• Software selectable clock sources for input to prescaler with programmable 16-bit prescaler
	  • Bus clock
	  • IRC clock (31.25~39.0625 kHz)
	  • LPO (~20 kHz)
	  • System oscillator output clock

FSK Demodulation Timer (FSKDT)	• One frequency-shift keyed signal input channel • 16-bit free-running counter • Three 16-bit phase counters • 16-bit current position number • Input signal edge-to-edge period error detection • Software module reset • Message bit-stream detection • Message byte packing

2.2.9 Communication interfaces

The following communication interfaces are available on this device:

Table 2-10. Communication modules

Module	Description
Inter-integrated circuit (I2C)	One I2C module for inter device communications. Also supports the System Management Bus (SMBus) Specification, version 2.
Universal asynchronous receiver/ transmitters (UART)	One asynchronous serial bus communication interface (UART) modules with optional 13-bit break, full duplex non-return to zero (NRZ), and LIN extension support.

2.2.10 Human-machine interfaces

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

Table 2-11. HMI modules

Module	Description
Port Control (PORT)	Two sets of I/O ports.
General purpose input/output (GPIO)	Up to 13 general purpose input or output (GPIO) pins.
Interrupt (IRQ)	• An external interrupt pin IRQ. • IRQ Interrupt Control bits • Programmable edge-only or edge and level interrupt sensitivity • Automatic interrupt acknowledge • Internal pullup device

2.2.11 Orderable part numbers

The following table summarizes the part numbers of the devices covered by this document.

Table 2-12. Orderable part numbers summary

Freescale part number	CPU frequency	Pin count	Package	Total flash memory	RAM	Temperature range
MWPR1516CFM(R)	24 MHz	32	QFN	16 KB	4 KB	-40 to 85 °C
MWPR1516CALR	24 MHz	36	WLCSP	16 KB	4 KB	-40 to 85 °C

Chapter 3
Core Overview

3.1 Introduction

The enhanced ARM Cortex M0+ is the member of the Cortex-M Series of processors targeting microcontroller cores focused on very cost sensitive, low power applications. It has a single 32-bit AMBA AHB-Lite interface and includes an NVIC component. The processor supports the ARMv6-M instruction set (Thumb) architecture including all but three 16-bit Thumb opcodes (52 total) plus seven 32-bit instructions. It is upward compatible with other Cortex-M profile processors.

3.1.1 ARM Cortex M0+ core

The ARM Cortex M0+ parameter settings are as follows:

Table 3-1. ARM Cortex-M0+ parameter settings

Parameter	Verilog name	Value	Description
Arch Clock Gating	ACG	1 = Present	Implements architectural clock gating
DAP Slave Port Support	AHBSLV	1	Support any AHB debug access port (like the CM4 DAP)
DAP ROM Table Base	BASEADDR	0xF000_2003	Base address for DAP ROM table
Endianess	BE	0	Little endian control for data transfers
Breakpoints	BKPT	2	Implements 2 breakpoints
Debug Support	DBG	1 = Present	—
Halt Event Support	HALTEV	1 = Present	—
I/O Port	IOP	0 = No Present	Not implements single-cycle ld/st accesses to special addr space
IRQ Mask Enable	IRQDIS	0x0	—
Debug Port Protocol	JTAGnSW	0 = SWD	SWD protocol, not JTAG
Core Memory Protection	MPU	0 = Absent	No MPU
Number of IRQs	NUMIRQ	32	Assume full NVIC request vector

Table continues on the next page...

Table 3-1. ARM Cortex-M0+ parameter settings (continued)

Parameter	Verilog name	Value	Description
Reset all regs	RAR	0 = Standard	Do not force all registers to be async reset
Multiplier	SMUL	0 = Fast Mul	Implements single-cycle multiplier
Multi-drop Support	SWMD	0 = Absent	Do not include serial wire support for multi-drop
System Tick Timer	SYST	1 = Present	Implements system tick timer (for CM4 compatibility)
DAP Target ID	TARGETID	0	—
User/Privileged	USER	1 = Present	Implements processor operating modes
Vector Table Offset Register	VTOR	1 = Present	Implements relocation of exception vector table
WIC Support	WIC	1 = Present	Implements WIC interface
WIC Requests	WICLINES	34	Exact number of wakeup IRQs is 34
Watchpoints	WPT	2	Implements 2 watchpoints

For details on the ARM Cortex-M0+ processor core, see the ARM website: **arm.com**.

3.1.2 Buses, interconnects, and interfaces

The ARM Cortex-M0+ core has one bus interfaces:

- single 32-bit AMBA-3 AHB-Lite system interface that provides connections to peripherals and all system memory, which includes flash and RAM.

3.1.3 System tick timer

The CLKSOURCE field in SysTick Control and Status register selects either the core clock (when CLKSOURCE = 1) or a divide-by-16 of the core clock (when CLKSOURCE = 0). Because the timing reference is a variable frequency, the TENMS field in the SysTick Calibration Value Register is always zero.

3.1.4 Debug facilities

This device supports standard ARM 2-pin SWD debug port.

3.1.5 Core privilege levels

The core on this device is implemented with both Privileged and Unprivileged 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

3.2 Asynchronous wakeup interrupt controller (AWIC)

3.2.1 AWIC overview

The primary function of the AWIC block 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.

3.2.2 Wakeup sources

The device uses the following internal and external inputs to the AWIC module.

Table 3-2. AWIC stop wakeup sources

Wake-up source	Description
Available system resets	$\overline{\text{RESET}}$ pin when LPO is its clock source
IRQ	IRQ pin
CNC	AD_IN status change
I2C	Address match wake-up
UART	UART active edge detect at RX
RTC	Alarm interrupt
Non-maskable interrupt	$\overline{\text{NMI}}$ pin

3.3 Nested vectored interrupt controller (NVIC)

3.3.1 Interrupt priority levels

This device supports 4 priority levels for interrupts. Therefore, in the NVIC each source in the IPR registers contains 2 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	0	0	IRQ2		0	0	0	0	0	0	IRQ1		0	0	0	0	0	0	IRQ0		0	0	0	0	0	0
W																																

3.3.2 Non-maskable interrupt

The non-maskable interrupt request to the NVIC is controlled by the external $\overline{\text{NMI}}$ signal. The pin, which the $\overline{\text{NMI}}$ signal is multiplexed on, must be configured for the $\overline{\text{NMI}}$ function to generate the non-maskable interrupt request.

3.3.3 Interrupt channel assignments

The interrupt vector 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 3-4. Interrupt vector assignments

Address	Vector	IRQ[1]	NVIC IPR register number[2]	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	—	—	—	—

Table continues on the next page...

Table 3-4. Interrupt vector assignments (continued)

Address	Vector	IRQ[1]	NVIC IPR register number[2]	Source module	Source description
0x0000_0014	5	—	—	—	—
0x0000_0018	6	—	—	—	—
0x0000_001C	7	—	—	—	—
0x0000_0020	8	—	—	—	—
0x0000_0024	9	—	—	—	—
0x0000_0028	10	—	—	—	—
0x0000_002C	11	—	—	ARM core	Supervisor call (SVCall)
0x0000_0030	12	—	—	—	—
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	—	—
0x0000_0044	17	1	0	—	—
0x0000_0048	18	2	0	—	—
0x0000_004C	19	3	0	—	—
0x0000_0050	20	4	1	—	—
0x0000_0054	21	5	1	FTMRE	Command complete
0x0000_0058	22	6	1	PMC	Low-voltage warning
0x0000_005C	23	7	1	IRQ	External interrupt
0x0000_0060	24	8	2	I2C0	Single interrupt vector for all sources
0x0000_0064	25	9	2	—	—
0x0000_0068	26	10	2	—	—
0x0000_006C	27	11	2	—	—
0x0000_0070	28	12	3	UART0	Status and error
0x0000_0074	29	13	3	—	—
0x0000_0078	30	14	3	ADC	ADC Error or conversion sequence abort
0x0000_007C	31	15	3	ADC	ADC conversion complete interrupt
0x0000_0080	32	16	4	ACMP0	Analog comparator 0 interrupt
0x0000_0084	33	17	4	FTM0	Single interrupt vector for all sources
0x0000_0088	34	18	4	FTM1	Single interrupt vector for all sources
0x0000_008C	35	19	4	—	—
0x0000_0090	36	20	5	RTC	RTC overflow
0x0000_0094	37	21	5	—	—
0x0000_0098	38	22	5	PIT_CH0	PIT CH0 overflow
0x0000_009C	39	23	5	PIT_CH1	PIT CH1 overflow

Table continues on the next page...

Table 3-4. Interrupt vector assignments (continued)

Address	Vector	IRQ[1]	NVIC IPR register number[2]	Source module	Source description
0x0000_00A0	40	24	6	—	—
0x0000_00A4	41	25	6	—	—
0x0000_00A8	42	26	6	FSKDT	Single interrupt vector for all sources
0x0000_00AC	43	27	6	ICS	Clock loss of lock
0x0000_00B0	44	28	7	WDOG	Watchdog timeout
0x0000_00B4	45	29	7	—	—
0x0000_00B8	46	30	7	CNC	Single interrupt vector for all sources
0x0000_00BC	47	31	7	LDO	Single interrupt vector for all sources

1. Indicates the NVIC's interrupt source number.
2. Indicates the NVIC's IPR register number used for this IRQ. The equation to calculate this value is: IRQ div 4

Chapter 4
Memory and Memory Map

4.1 Flash memory

The device contains 16 KB flash, with 512-byte sector size.

The flash access time supports single cycle access when the CPU operating speed is 24 MHz (1:1).

Table 4-1. WPR1516 flash memory size

Freescale part number	Program flash (KB)	Block 0 (P-Flash) address range
MWPR1516CFM(R)	16	0x0000_0000 – 0x0000_3FFF
MWPR1516CALR	16	0x0000_0000 – 0x0000_3FFF

4.1.1 Flash memory map

The flash memory 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.

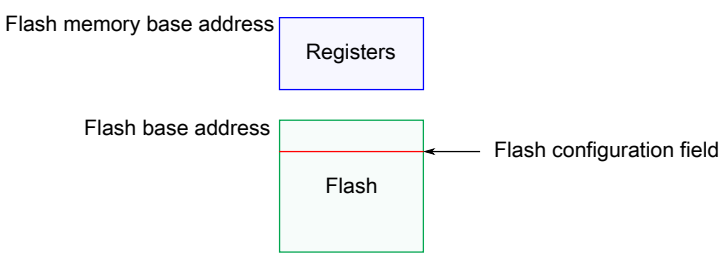


Figure 4-1. Flash memory map

The on-chip Flash is implemented in a portion of the allocated Flash range to form a contiguous block in the memory map beginning at address 0x0000_0000. See Flash memory for details of supported ranges.

Accesses to the flash memory ranges outside the amount of Flash on the device causes the bus cycle to be terminated with an error followed by the appropriate response in the requesting bus master. Read collision events in which flash memory is accessed while a flash memory resource is being manipulated by a flash command also generates a bus error response.

4.1.2 Flash security

How flash security is implemented on this device is described in Chip Security.

4.1.3 Flash Modes

The flash memory chapter defines two modes of operation - NVM normal and NVM special modes. On this device, The flash memory only operates in NVM normal mode. All references to NVM special mode should be ignored.

4.1.4 Erase all flash contents

In addition to software, the entire flash memory may be erased external to the flash memory via the SW-DP debug port by setting MDM-AP CONTROL[0] (bit 0 of the MDM-AP Control register). MDM-AP STATUS[0] (bit 0 of the MDM-AP Status register) is set to indicate the mass erase command has been accepted. MDM-AP CONTROL[0] is cleared when the mass erase completes.

4.2 SRAM

The SRAM supports single cycle access (zero wait states) at all core speeds.

The amounts of SRAM for the devices covered in this document are:

Table 4-2. SRAM size

Freescale part number	SRAM
MWPR1516	4 KB

4.2.1 SRAM ranges

The on-chip SRAM is split into two ranges, 1/4 is allocated SRAM_L and 3/4 is allocated to 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/4)] to 0x1FFF_FFFF
- SRAM_U = 0x2000_0000 to [0x2000_0000+(SRAM_size*(3/4))-1]

This is illustrated in the following figure.

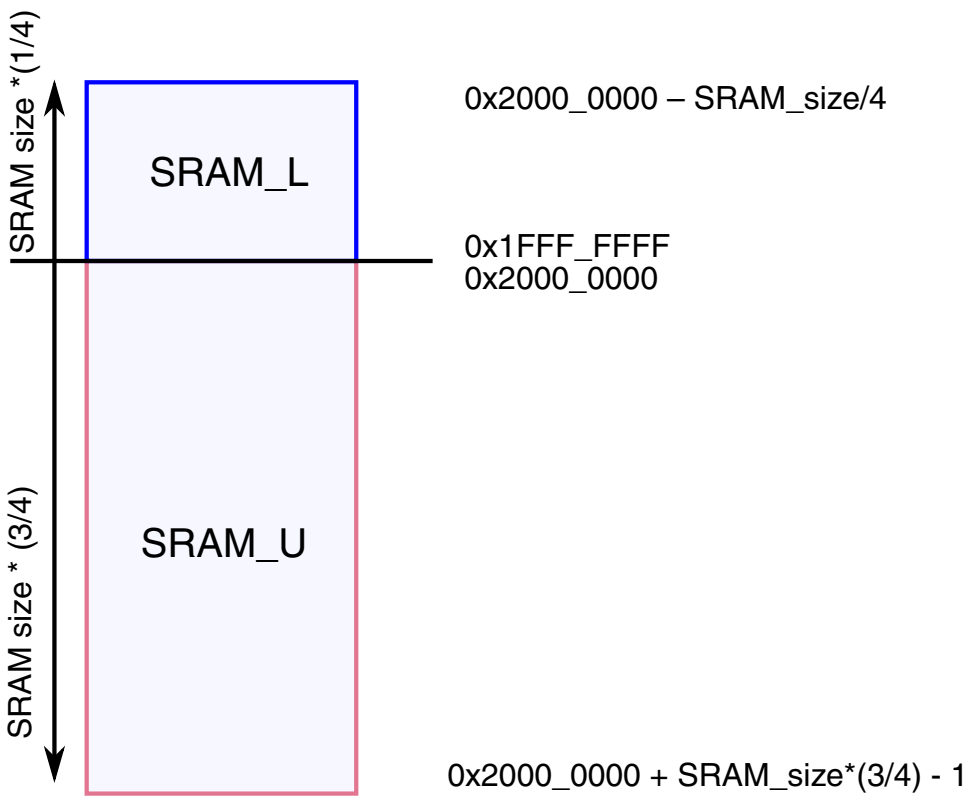


Figure 4-2. SRAM blocks memory map

For example, for a device containing 16 KB of SRAM, the ranges are:

- SRAM_L: 0x1FFF_F000 – 0x1FFF_FFFF
- SRAM_U: 0x2000_0000 – 0x2000_2FFF

4.3 Memory map

4.3.1 Introduction

This device contains various memories and memory-mapped peripherals which are located in a 4 GB memory space. This chapter describes the memory and peripheral locations within that memory space.

4.3.2 System memory map

The following table shows the high-level device memory map.

Table 4-3. System memory map

System 32-bit Address Range	Destination Slave	Access
0x0000_0000–0x07FF_FFFF[1]	Program flash and read-only data (Includes exception vectors in first 196 bytes)	Cortex-M0+ core
0x0800_0000–0x0FFF_FFFF	Reserved	—
0x1000_0000– 0x1FFF_FBFF	Reserved	—
0x1FFF_FC00-0x1FFF_FFFF[2]	SRAM_L: Lower SRAM	Cortex-M0+ core
0x2000_0000-0x2000_0BFF	SRAM_U: Upper SRAM	Cortex-M0+ core
0x2000_0C00–0x21FF_FFFF	Reserved	—
0x2200_0000–0x2201_7FFF	Reserved	—
0x2201_8000–0x23FF_FFFF	Reserved	—
0x2400_0000–0x3FFF_FFFF	Reserved	—
0x4000_0000–0x4007_FFFF	AIPS peripherals	Cortex-M0+ core
0x4008_0000–0x400F_EFFF	Reserved	—
0x400F_F000–0x400F_FFFF	Reserved	—
0x4010_0000–0x43FF_FFFF	Reserved	—
0x4400_0000–0x5FFF_FFFF	Reserved	—
0x6000_0000–0xDFFF_FFFF	Reserved	—
0xE000_0000–0xE00F_FFFF	Private peripherals	Cortex-M0+ core
0xE010_0000–0xEFFF_FFFF	Reserved	—
0xF000_0000–0xF000_0FFF	Reserved	—
0xF000_1000–0xF000_1FFF	Reserved	—
0xF000_2000–0xF000_2FFF	ROM table[3]	Cortex-M0+ core
0xF000_3000–0xF000_3FFF	Miscellaneous control module (MCM)	Cortex-M0+ core
0xF000_4000–0xF7FF_FFFF	Reserved	—
0xF800_0000–0xFFFF_FFFF	Reserved	Cortex-M0+ core

1. The program flash always begins at 0x0000_0000 but the end of implemented flash varies depending on the amount of flash implemented for a particular device. See Flash memory for details.
2. This range varies depending on SRAM sizes. See SRAM for details.
3. This device implements a system ROM table which is used to redirect to ARM Cortex M0+ (Flycatcher) ROM table in CoreSight debug system.

4.3.3 Peripheral bridge (AIPS-Lite) memory map

The peripheral memory map is accessible via one slave port on the crossbar in the 0x4000_0000–0x400F_FFFF region. The device implements one peripheral bridge that defines a 1024 KB address space.

The three regions associated with this space are:

- A 128 KB region, partitioned as 32 spaces, each 4 KB in size and reserved for on-platform peripheral devices. The AIPS controller generates unique module enables for all 32 spaces.
- A 384 KB region, partitioned as 96 spaces, each 4 KB in size and reserved for off-platform modules. The AIPS controller generates unique module enables for all 96 spaces.
- The last slot is a 4 KB region beginning at 0x400F_F000 for accessing the GPIO module.

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.

4.3.3.1 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.

4.3.3.2 Peripheral bridge (AIPS-Lite) memory map

NOTE

- Slots 0-95 and 128 are for 32-bit data width modules, with the exception that slots 32, 49, and 82 are for 8-bit data width modules (flash memory, IRQ, and WDOG) and slot 52 and 59, are for 16-bit data width modules(FSKDT and ADC)
- Slots 96-127 are for 8-bit data width modules , with the exception that slots 110 and 112 are for 16-bit data width modules (CNC and LDO).

Table 4-4. Peripheral bridge 0 slot assignments

System 32-bit base address	Slot number	Module
0x4000_0000	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	—
0x4000_9000	9	—
0x4000_A000	10	—
0x4000_B000	11	—
0x4000_C000	12	—
0x4000_D000	13	—
0x4000_E000	14	—
0x4000_F000	15	GPIO controller
0x4001_0000	16	—
0x4001_1000	17	—
0x4001_2000	18	—
0x4001_3000	19	—
0x4001_4000	20	—
0x4001_5000	21	—
0x4001_6000	22	—
0x4001_7000	23	—

Table continues on the next page...

Table 4-4. Peripheral bridge 0 slot assignments (continued)

System 32-bit base address	Slot number	Module
0x4001_8000	24	—
0x4001_9000	25	—
0x4001_A000	26	—
0x4001_B000	27	—
0x4001_C000	28	—
0x4001_D000	29	—
0x4001_E000	30	—
0x4001_F000	31	—
0x4002_0000	32	Flash memory (FTMRE)
0x4002_1000	33	—
0x4002_2000	34	—
0x4002_3000	35	—
0x4002_4000	36	—
0x4002_5000	37	—
0x4002_6000	38	—
0x4002_7000	39	—
0x4002_8000	40	—
0x4002_9000	41	—
0x4002_A000	42	—
0x4002_B000	43	—
0x4002_C000	44	—
0x4002_D000	45	—
0x4002_E000	46	—
0x4002_F000	47	—
0x4003_0000	48	—
0x4003_1000	49	IRQ controller (IRQ)
0x4003_2000	50	—
0x4003_3000	51	—
0x4003_4000	52	FSK demodulation timer (FSKDT)
0x4003_5000	53	—
0x4003_6000	54	—
0x4003_7000	55	Periodic interrupt timers (PIT)
0x4003_8000	56	Flex timer 0 (FTM0)
0x4003_9000	57	Flex timer 1 (FTM1)
0x4003_A000	58	—
0x4003_B000	59	Analog-to-digital converter (ADC)
0x4003_C000	60	—
0x4003_D000	61	Real time clock (RTC)
0x4003_E000	62	—

Table continues on the next page...

Table 4-4. Peripheral bridge 0 slot assignments (continued)

System 32-bit base address	Slot number	Module
0x4003_F000	63	—
0x4004_0000	64	—
0x4004_1000	65	—
0x4004_2000	66	—
0x4004_3000	67	—
0x4004_4000	68	—
0x4004_5000	69	—
0x4004_6000	70	—
0x4004_7000	71	—
0x4004_8000	72	System integration module (SIM)
0x4004_9000	73	Port controller (PORT)
0x4004_A000	74	—
0x4004_B000	75	—
0x4004_C000	76	—
0x4004_D000	77	—
0x4004_E000	78	—
0x4004_F000	79	—
0x4005_0000	80	—
0x4005_1000	81	—
0x4005_2000	82	Watchdog (WDOG)
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	—
0x4006_2000	98	—
0x4006_3000	99	—
0x4006_4000	100	Internal clock source (ICS)
0x4006_5000	101	System oscillator (OSC)

Table continues on the next page...

Table 4-4. Peripheral bridge 0 slot assignments (continued)

System 32-bit base address	Slot number	Module
0x4006_6000	102	Inter-integrated circuit 0 (I2C0)
0x4006_7000	103	—
0x4006_8000	104	—
0x4006_9000	105	—
0x4006_A000	106	Universal asynchronous receiver/transmitter 0 (UART0)
0x4006_B000	107	
0x4006_C000	108	
0x4006_D000	109	—
0x4006_E000	110	WPC communication and clamp controller (CNC)
0x4006_F000	111	
0x4007_0000	112	Linear regulator controller (LDO)
0x4007_1000	113	Differential input voltage PGA (PGA)
0x4007_2000	114	—
0x4007_3000	115	Analog comparator 0 (ACMP0)
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	—
0x4007_D000	125	Power management controller (PMC)
0x4007_E000	126	—
0x4007_F000	127	—
0x400F_F000	128	—

Chapter 5
Clock Distribution

5.1 Introduction

This chapter presents the clock architecture for the device, the overview of the clocks and includes a terminology section.

The Cortex M0+ resides within a synchronous core platform, where the processor and bus masters, Flash and peripherals clocks can be configured independently.

The ICS module is used for main system clock generation. The ICS module controls which clock sources (internal references, external crystals or external clock signals) generate the source of the system clocks.

5.2 Programming model

The selection and multiplexing of system clock sources is controlled and programmed via the ICS 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.

5.3 High-level device clocking diagram

This device contains following on-chip clock sources:

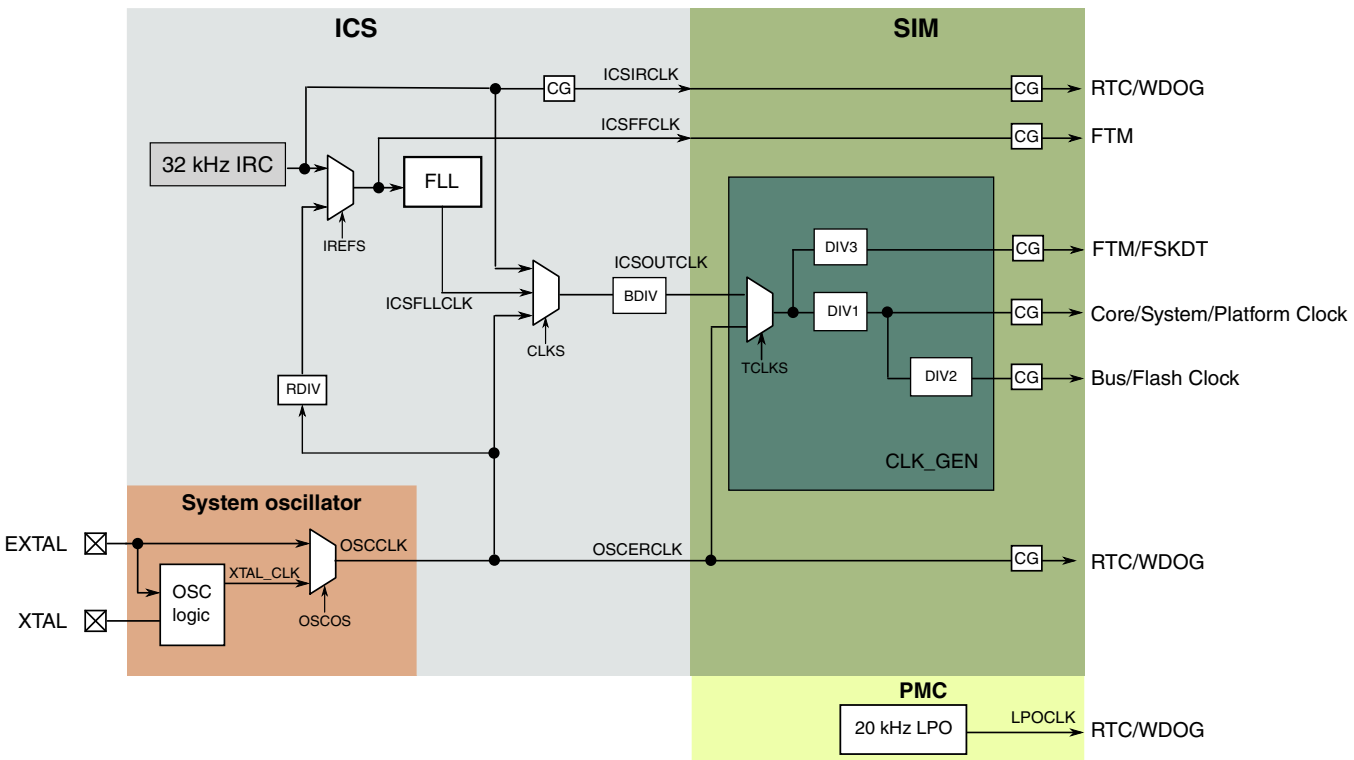
- Internal Clock Source (ICS) module: The main clock source generator providing bus clock and other reference clocks to peripherals
- Low-Power Oscillator (LPO) module: The on-chip low-power oscillator providing 20 kHz reference clock to RTC and Watchdog (WDOG)

Figure 5-1 shows how clocks from the ICS modules are distributed to the microcontroller's other function units. Some modules in the microcontroller have selectable clock input.

The following registers of ICS, and SIM modules control the multiplexers, dividers, and clock gates shown in the figure:

Table 5-1. Registers controlling multiplexers, dividers, and clock gate

	OSC	ICS	SIM
Multiplexers	OSC_CR	ICS_C1	–
Dividers	–	ICS_C2	SIM_CLKDIV
Clock gates	–	–	SIM_SCGC



CG — Clock gate
Note: See subsequent sections for details on where these clocks are used.

Figure 5-1. Clocking diagram

5.4 Clock definitions

The following table describes the clocks in Figure 5-1.

Table 5-2. Clock definitions

Clock name	Description
Core clock	ICSOUTCLK divided by DIV1, clocks the ARM Cortex-M0+ core. It's the CPU HCLK
Platform clock	ICSOUTCLK divided by DIV1, clocks the crossbar switch and NVIC, It's the free-running FCLK
System clock	ICSOUTCLK divided by DIV1, clocks the bus masters directly
Bus clock	System clock divided by DIV2, clocks the bus slaves and peripherals
Flash clock	System clock divided by DIV2, clocks the Flash memory module. It is same as Bus clock in this device
Timer clock	ICSOUTCLK divided by DIV3, clocks the FTM and FSKDT modules
Debug clock	Debug logic clock. On this device it is derived from platform clock
SWD clock	DAP interface clock. SWD clock is typically driven by an external debugger and completely asynchronous to Core clock and platform clock
ICSIRCLK	ICS output of the internal 32 kHz IRC reference clock. ICSIRCLK can be selected as the clock source of RTC or WDOG modules
ICSOUTCLK	ICS output of either IRC, ICSFLLCLK or ICS's external reference clock that sources the core, system, bus, and flash clock
ICSFLLCLK	Output of the FLL
ICSFFCLK	ICS output of the fixed frequency clock. ICSFFCLK can be selected as clock source for the FTM modules. The frequency of the ICSFFCLK is determined by the setting of the ICS
OSCCLK	System oscillator output of the internal oscillator or sourced directly from EXTAL. Used as ICS external reference clock
OSCERCLK	System oscillator output sourced from OSCCLK that can be selected as the clock source of RTC, WDOG.
LPOCLK	PMC 20 kHz output, The LPOCLK can be selected as the clock source to the RTC or WDOG modules

5.4.1 Device clock summary

The following table provides more information regarding the on-chip clocks.

Table 5-3. Clock summary

Clock name	Run mode frequency	Clock source	Clock is disabled when…
Core clock	Up to 24 MHz	ICSOUTCLK clock divider	In Wait and Stop modes
Platform clock	Up to 24 MHz	ICSOUTCLK clock divider	In Stop mode
System clock	Up to 24 MHz	ICSOUTCLK clock divider	In Stop mode
Timer clock	Up to 48 MHz	ICSOUTCLK clock divider	In Stop mode
Bus clock	Up to 24 MHz	ICSOUTCLK clock divider	In Stop mode

Table continues on the next page...

Table 5-3. Clock summary (continued)

Clock name	Run mode frequency	Clock source	Clock is disabled when…
Debug clock	Up to 24 MHz	Derive from Platform clock	Debug not enabled
SWD clock	Up to 24MHz	SWD_CLK pin	Input from external clock, so will not be disabled.
Flash clock	Up to 24 MHz	ICSOUTCLK clock divider	In Stop mode
Internal reference (ICSIRCLK)[1]	31.25–39.0625 kHz IRC	IRC	ICS_C1[IRCLKEN] =0
External reference (OSCERCLK)[1]	DC up to 48 MHz (bypass), 31.25–39.0625 kHz or 4–24 MHz (crystal)	System OSC	OSC_CR[OSCEN] =0
FLL out clock (ICSFLLCLK)	40-50 MHz	System OSC or IRC	In Stop mode, or FLL not enabled
ICS Fixed Frequency clock (ICSFFCLK)	31.25–39.0625 kHz	System OSC or IRC	In Stop mode
LPOCLK	20 kHz	PMC	Optional be OFF in STOP mode

1. This clock does not support being enabled during entering stop mode.

5.4.2 Clock distribution

The following figure shows a simplified clock distribution diagram

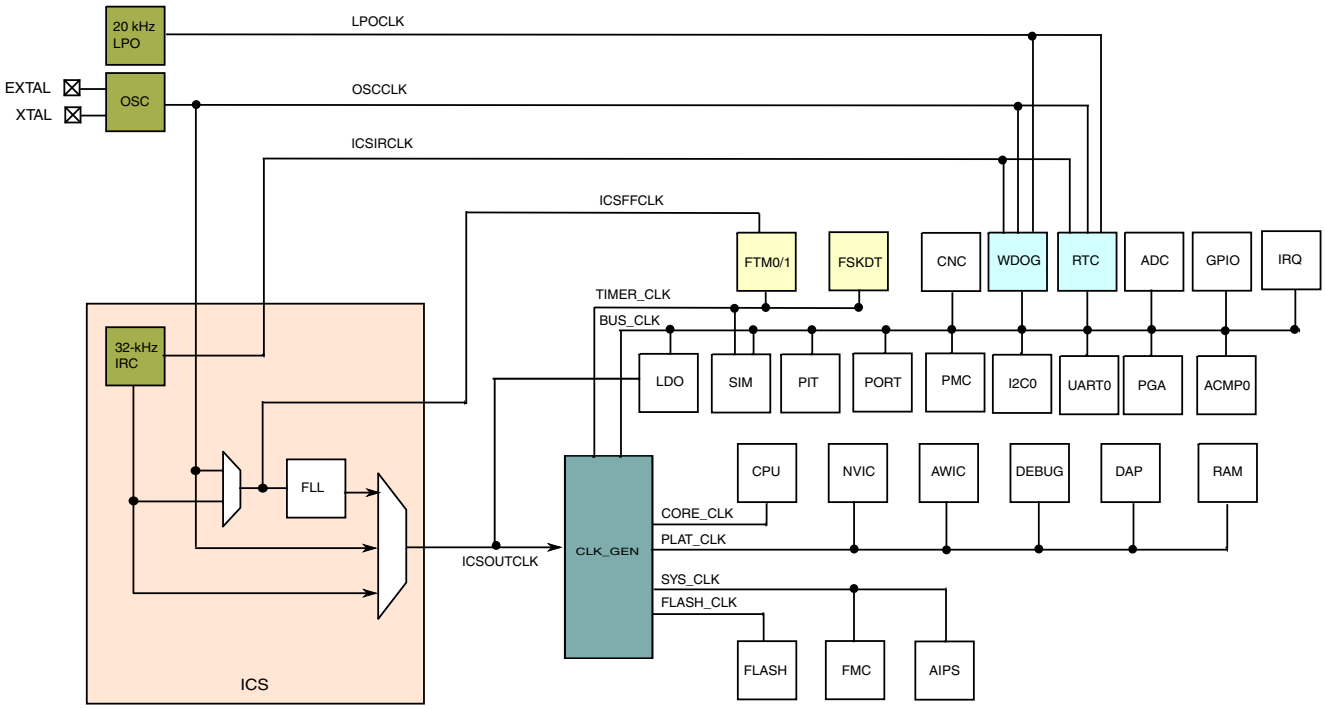


Figure 5-2. High-level clock distribution diagram

NOTE

Clock divide and gating are not shown in clock distribution diagram.

5.5 Internal clocking sources

The internal clock sources on this device are as following:

- On-chip RC oscillator range of 31.25–39.0625 kHz as the reference of FLL input with 0.2% trim step and 1% accuracy across temperature range of 0 °C to 70 °C and 1.5% accuracy across temperature of –40 °C to 85 °C.
- On-chip internal 20 kHz oscillator as the low-frequency low-power source for RTC, WDOG and PORT glitch filter according to specific use case requirement. User can do calibration to let 20 kHZ fall in ± 5% range.

The following table shows the frequency availability of this device:

Table 5-4. ICS bus frequency availability with internal reference

Reference		Bus
FEI (high range)	BDIV = 0	40 MHz ~ 50 MHz[1]
	BDIV = 1	20 MHz ~ 25 MHz

Table continues on the next page...

Table 5-4. ICS bus frequency availability with internal reference (continued)

Reference		Bus
	BDIV = 2	10 MHz ~ 12.5 MHz
	BDIV = 4	5 MHz ~ 6.25 MHz
	BDIV = 8	2.5 MHz ~ 3.125 MHz
	BDIV = 16	1.25 MHz ~ 1.5625 MHz
	BDIV = 32	625 kHz ~ 781.25 kHz
	BDIV = 64	312.5 kHz ~ 390.625 kHz
	BDIV = 128	156.25 kHz ~ 195.3125 kHz

1. On this device, core and system clock can work up to 24 MHz, timer clock can be up to 48 MHz. Hence, the FLL clock output must be equal or less than 48 MHz and the bus and core frequency must be equal or less than 24 MHz.

5.6 Clock gating

The clock to each module can be individually gated on and off using the SIM_SCGC. Prior to initializing a module, set the corresponding bit in SIM_SCGC 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.

NOTE

Do not disable the SIM_SCGC[ADC] when ADC is running, otherwise, it may cause system stalling.

5.7 Module clocks

The following table summarizes the clocks associated with each module.

Table 5-5. Module clocks

Module	Bus interface clock	Internal clocks	I/O interface clocks
Core modules			
ARM Cortex-M0+ core	Platform clock	Core clock	—
NVIC	Platform clock	—	—
DAP	Platform clock	—	SWD_CLK
System modules			
PORT	Bus clock	LPOCLK[1]	—
AXBS_Lite	Platform clock	—	—
AIPS_Lite	Platform clock	Bus clock	—
PMC	Bus clock	LPOCLK	—

Table continues on the next page...

Table 5-5. Module clocks (continued)

Module	Bus interface clock	Internal clocks	I/O interface clocks
SIM	Bus clock	Timer clock	—
MCM	Platform clock	—	—
WDOG	Bus clock	Bus clock LPOCLK ICSIRCLK OSCERCLK	—
Clocks			
ICS	Bus clock	ICSOUTCLK ICSFLLCLK ICSIRCLK OSCERCLK	—
Memory and memory interfaces			
Flash Controller	System clock	—	—
Flash memory	Flash clock	—	—
SRAM	Platform clock	—	—
UHV			
CNC	Bus clock	Bus clock	—
LDO5V3A	Bus clock	Bus clock ICSOUTCLK[2]	—
Analog			
ADC	Bus clock	—	—
ACMP0	Bus clock	—	—
PGA	Bus clock	—	—
Timers			
PIT	Bus clock	—	—
FSKDT	Timer clock	—	—
FTM0	Timer clock	Timer clock ICSFFCLK	—
FTM1	Timer clock	Timer clock ICSFFCLK	—
RTC	Bus clock	Bus clock LPOCLK ICSIRCLK OSCERCLK	RTC_CLKOUT
Communication interfaces			
I^2C0	Bus clock	—	I2C0_SCL
UART0	Bus clock	—	—
Human-machine interfaces			
GPIO	System clock	—	—

Table continues on the next page...

Table 5-5. Module clocks (continued)

Module	Bus interface clock	Internal clocks	I/O interface clocks
IRQ	Bus clock	—	—

1. The LPOCLK is used for the glitch filter function.
2. This clock is for charge pump generation.

Chapter 6
Reset and Boot

6.1 Introduction

The following reset sources are supported in this MCU:

Table 6-1. Reset sources

Reset sources	Description
POR reset	• Power-on reset (POR)
System resets	• External pin reset (PIN) • Low-voltage detect (LVD) • Watchdog (WDOG) timer • ICS loss of clock (LOC) reset • Stop mode acknowledge error (SACKERR) • Software reset (SW) • Lockup reset (LOCKUP) • MDM DAP system reset

Each of the system reset sources has an associated bit in the System Reset Status and ID Register (SIM_SRSID).

The MCU can exit and reset in functional mode where the CPU is executing code (default) or the CPU is in a debug halted state.

6.2 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.

6.2.1 Power-on reset (POR)

The MCU is powered by the PMC on-chip LDO whose input is the wireless power receiver rectified voltage. When the LDO output (VDD5V) is initially applied to the MCU or when it drops below the power-on reset voltage level (V_{POR}), the POR circuit causes a POR reset condition.

When the on-chip LDO output voltage rises, the LVD circuit holds the MCU in reset until the supply has risen above the LVD low threshold (V_{LVDL}). POR and LVD fields of the System Reset Status and ID Register (SIM_SRSID) register (SIM_SRSID[POR] and SIM_SRSID[LVD]) are set following a POR.

6.2.2 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 program counter (PC) from vector-table offset 4
- The Link Register (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 (except that the SWD_DIO/SWD_CLK, $\overline{\text{NMI}}$ and $\overline{\text{RESET}}$ pins could be enabled after system reset according to the System Options Register 0 (SIM_SOPT0) setting). The pins with analog functions assigned to them default to their analog function after reset.

6.2.2.1 External pin reset ($\overline{\text{RESET}}$)

This pin has an internal pullup resistor. Asserting $\overline{\text{RESET}}$ wakes the device from any mode.

After POR reset, the PTB0 pin functions as $\overline{\text{RESET}}$. SIM_SOPT0[RSTPE] must be programmed to enable the other functions. When this field is clear, this pin can function as GPIO or other alternative functions .

6.2.2.1.1 Reset pin filter

The $\overline{\text{RESET}}$ pin filter supports filtering from both the 20 kHz LPO clock and the bus clock. It can be used as a simple low-pass filter to filter any glitch that is introduced from the pin of $\overline{\text{RESET}}$.

The glitch width threshold can be adjusted easily by setting Port Filter Control Register 0 (PORT_IOFLT0) between 1~4096 BUSCLKs (or 1~128 LPOCLKs). This configurable glitch filter can replace an on-board external analog filter, and greatly improve the EMC performance. Setting Port Filter Control Register 0 (PORT_IOFLT0) can configure the filter of the whole port.

6.2.2.2 Low-voltage detect (LVD)

This device includes a system to protect against low-voltage conditions in order to protect memory contents and control MCU system states during supply voltage variations. This system consists of a power-on-reset (POR) circuit, and an LVD circuit with a user selectable trip voltage, V_{LVD}

The LVD circuit can be enabled/disabled and the LVD voltage threshold can be configured through its configure bits.

6.2.2.3 Watchdog timer

The 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 watchdog. If this periodic refreshing does not occur, the watchdog issues a system reset. The WDOG reset causes SIM_SRSID[WDOG] to set.

6.2.2.4 ICS loss-of-clock (LOC)

The ICS on this chip supports external reference clock monitor with reset capability.

In FBE, FEE, or FBELP modes, if 1 is written to ICS_C4[CME], the clock monitor is enabled. If the external reference falls below a certain frequency, such as f_{loc_high} or f_{loc_low} depending on OSC_CR[RANGE], the MCU will reset. SIM_SRSID[LOC] will be set to indicate the error.

In FBELP or FBILP modes, the FLL is not on, so the external reference clock monitor will not function even if 1 is written to ICS_C4[CME].

External reference clock monitor uses FLL as the internal reference clock. The FLL must be functional before ICS_C4[CME] is set.

6.2.2.5 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 20 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.

6.2.2.6 Software reset (SW)

The SYSRESETREQ field 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.

6.2.2.7 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 SIM_SRSID[LOCKUP] to set.

6.2.2.8 MDM-AP system reset request

Set the System Reset Request field in the MDM-AP Control register to initiate a system reset. This is the primary method for resets via the SWD interface. The system reset is held until this bit is cleared.

Set the Core Hold Reset field in the MDM-AP Control register to hold the core in reset as the rest of the chip comes out of system reset.

6.2.3 MCU resets

A variety of resets are generated by the MCU to reset different modules.

6.2.3.1 POR Only

The POR Only reset asserts on the POR reset source only. It resets the PMC and RTC.

The POR Only reset also causes all other reset types to occur.

6.2.3.2 Chip POR

The Chip POR asserts on POR and LVD reset sources. It resets the Reset Pin Filter registers and parts of the SIM and ICS.

The Chip POR also causes the Chip Reset (including Early Chip Reset) to occur.

6.2.3.3 Early Chip Reset

The Early Chip Reset asserts on all reset sources. It resets only the flash memory module and ARM platform. It negates before flash memory initialization begins ("earlier" than when the Chip Reset negates).

6.2.3.4 Chip Reset

Chip Reset asserts on all reset sources and only negates after the $\overline{\text{RESET}}$ pin has also negated. It resets the remaining modules (the modules not reset by other reset types).

6.3 Boot

This section describes the boot sequence, including sources and options.

Some configuration information such as clock trim values stored in factory programmed flash locations is auto-loaded.

6.3.1 Boot sources

The CM0+ core adds support for a programmable Vector Table Offset Register (VTOR[1]) to relocate the exception vector table. This device supports booting from internal flash and RAM.

1. VTOR: refer to Vector Table Offset Register in the ARMv6-M Architecture Reference Manual.

This device supports booting from internal flash with the reset vectors located at addresses 0x0 (initial SP_main), 0x4 (initial PC), and RAM with relocating the exception vector table to RAM.

6.3.2 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 Reset Controller 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 (about 4.2 µs), and the ICS is enabled in its default clocking mode.
2. The $\overline{\text{RESET}}$ pin is released. If $\overline{\text{RESET}}$ pin 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 core clock is enabled and the system is released from reset.
3. The NVM starts internal initialization. Flash Controller is released from reset and begins initialization operations while the core is still halted before the flash initialization completes.
4. When the flash Initialization completes(16 µs) , the core 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. The CPU begins execution at the PC location.

Subsequent system resets follow this same reset flow.

Chapter 7
Power Management

7.1 Introduction

This chapter describes the various chip power modes and functionality of the individual modules in these modes.

7.2 Power modes

The power management controller (PMC) provides the user with multiple power options. The different modes of operation are supported to allow the user to optimize power consumption for the level of functionality needed.

The device supports Run, Wait,and Stop modes which are easy to use for customers both from different power consumption level and functional requirement. I/O states are held in all the modes.

- Run mode—CPU clocks can be run at full speed and the internal supply is fully regulated.
- Wait mode—CPU shuts down to conserve power; system clocks and bus clock are running and full regulation is maintained.
- Stop mode—voltage regulator is in standby.

The three modes of operation are Run, Wait, and Stop. The WFI and WFE instruction invokes both Wait and Stop modes for the chip.

Table 7-1. Chip power modes

Power mode	Description	Core mode	Normal recover 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 7-1. Chip power modes (continued)

Power mode	Description	Core mode	Normal recover 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. NVIC is disabled; AWIC is used to wake up from interrupt; peripheral clocks are stopped.	Sleep Deep	Interrupt

7.3 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.

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.

7.4 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. The standard behavior is shown with some exceptions.

Table 7-2. Module operation in low-power modes

Modules	RUN	WAIT	Stop
Core modules			
CPU	On	Standby	Standby
NVIC	On	On	Standby
System modules			
PMC	Full regulation	Full regulation	Loose regulation
WDOG	On	On	Optional on
LVD	On	On	Standby
DBG	On	On	Standby
Clock			
ICS	On	On	Standby
OSC	On	On	Standby
LPO	On	On	Optional on

Table continues on the next page...

Table 7-2. Module operation in low-power modes (continued)

Modules	RUN	WAIT	Stop
Memory			
Flash	On	On	Standby
RAM	On	Standby[1]	Standby
UHV			
CNC	On	On	Optional on[2]
LDO	On	On	Standby
TIMER			
FTM	On	On	Standby
PIT	On	On	Standby
FSKDT	On	On	Standby
RTC	On	On	Optional on
Analog			
ADC	On	On	Standby
ACMP	On	On	Standby
Communication Interfaces			
UART	On	On	Standby[3]
I2C	On	On	Standby[4]
Human-machine interfaces			
IRQ	On	On	Standby[5]
I/O	On	On	State held

1. SRAM memory enable signal disables internal clock signal and masks the address and data inputs when held low, RAM clock can be active in Wait mode.
2. Supports wake-up on Vad_in LVP/OVP in Stop mode
3. Supports wake-up on edge in Stop mode
4. Supports address match wake-up in Stop mode
5. Supports pin interrupt wake-up in Stop mode

Chapter 8
Security

8.1 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 of the effects of security on non-flash modules.

8.2 Flash security

The flash module provides security information to the MCU based on the state held by the FTMRE_FSEC[SEC]. The MCU, in turn, confirms the security request and limits access to flash resources. During reset, the flash module initializes the Flash Security Register (FTMRE_FSEC) using data read from the security byte of the flash configuration field.

NOTE

The security features apply only to external accesses: debug. CPU accesses to the flash are not affected by the status of Flash Security Register (FTMRE_FSEC).

In the unsecured state, all flash commands are available on the programming interfaces either from the debug port (SWD) or user code execution. When the flash is secured (FTMRE_FSEC[SEC] = 00, 01, or 11), the programmer interfaces are only allowed to launch mass erase operations. Additionally, in this mode, the debug port has no access to memory locations.

8.3 Security interactions with other modules

The flash security settings are used by the system 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.

8.3.1 Security interactions with debug

When flash security is active, the SWD port cannot access the memory resources of the MCU.

Although most debug functions are disabled, the debugger can write to the Flash Mass Erase in Progress field of 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.

Chapter 9
Debug

9.1 Introduction

This device's debug is based on the ARM CoreSight™ architecture and is configured to provide the maximum flexibility as allowed by the restrictions of the pinout and other available resources.

It provides register and memory accessibility from the external debugger interface, basic run/halt control plus 2 breakpoints and 2 watchpoints.

This device supports only one debug interface, Serial Wire Debug (SWD).

9.2 Debug port pin descriptions

The debug port pins default to their SWD functionality after power-on-reset (POR).

Table 9-1. Serial wire debug pin description

Pin Name	Type	Description
SWD_CLK	Input	Serial Wire Clock. This pin is the clock for debug logic when in the Serial Wire Debug mode. [1]
SWD_DIO	Input / Output	Serial Wire Debug Data input/output. The SWD_DIO pin is used by an external debug tool for communication and device control. This pin is pulled up internally.

1. The pad library of this device does not support on-chip pull down; it supports only pullup controlled by PTAPE0, external pulldown resistor is required to fully support SWD protocol.

9.3 SWD 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 Figure 9-1. 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 using SWD. The MDM-AP is accessible as Debug Access Port 1 with the available registers shown in the table below.

Table 9-2. MDM-AP register summary

Address	Register	Description
0x0100_0000	Status	See MDM-AP Status Register
0x0100_0004	Control	See MDM-AP Control Register
0x0100_00FC	IDR	Read-only identification register that always reads as 0x001C_0020

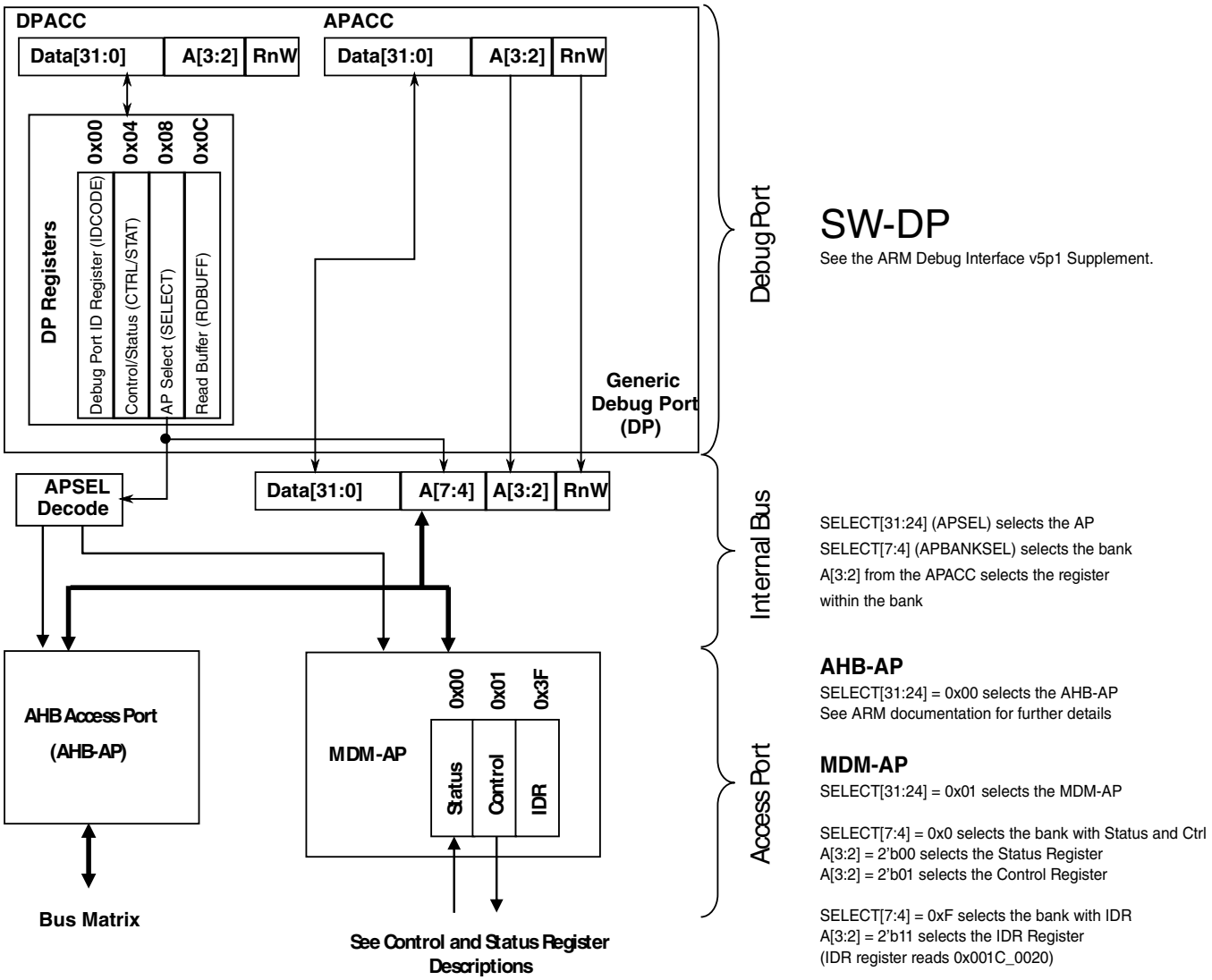


Figure 9-1. MDM AP addressing

9.3.1 MDM-AP Status Register

Table 9-3. MDM-AP Status register assignments

Bit	Name	Description
0	Flash Mass Erase Acknowledge	The Flash Mass Erase Acknowledge field is cleared after POR reset. The field is also cleared at launch of a mass erase command due to write of Flash Mass Erase in Progress field in MDM AP Control Register. The Flash Mass Erase Acknowledge is set after Flash control logic has started the mass erase operation.
1	Flash Ready	Indicates that flash memory has been initialized and debugger can be configured even if system is continuing to be held in reset via the debugger. 0 Flash is under initialization.

Table continues on the next page...

Table 9-3. MDM-AP Status register assignments (continued)

Bit	Name	Description
		1 Flash is ready.
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 field indicates when the part is locked and no system bus access is possible. 0 Device is unsecured. 1 Device is secured.
3	System Reset	Indicates the system reset state. 0 System is in reset. 1 System is not in reset.
4	Reserved	
5 – 15	Reserved for future use	Always read 0.
16	Core Halted	Indicates the core has entered Debug Halt mode 0 Core is not halted. 1 Core is halted.
17	Core SLEEPDEEP	SLEEPDEEP=1 indicates the core has entered Stop mode.
18	Core SLEEPING	SLEEPING=1 indicates the core has entered Wait mode.
19 – 31	Reserved for future use	Always reads 0.

9.3.2 MDM-AP Control Register

Table 9-4. 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.
1	Debug Disable	N	Set to disable debug. Clear to allow debug operation. When set, it overrides the C_DEBUGEN field within the DHCSR[2] and forces to disable Debug logic.
2	Debug Request	N	Set to force the core to halt. If the core is in Stop or Wait mode, this field can be used to wake the core and transition to a halted state.
3	System Reset Request	Y	Set to force a system reset. The system remains held in reset until this field is cleared.
4	Core Hold	N	Configuration field 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.

Table continues on the next page...

Table 9-4. MDM-AP Control register assignments (continued)

Bit	Name	Secure[1]	Description
5– 31	Reserved for future use	N	

1. Command available in secure mode
2. DHCSR: refer to the Debug Halting Control and Status Register in the ARMv6-M Architecture Reference Mannual.

9.4 Debug resets

The debug system receives the following sources of reset:

- 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.
- Writing 1 to the SYSRESETREQ field 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.

9.5 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.

The active debug will prevent the chip entering low power mode. In case that the chip is already in low-power mode, a debug request from MDM-AP control register will wake the chip from low-power mode.

9.6 Debug & security

When flash security is enabled, 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 status register and can determine the current security state of the device. In the case of a secure device, the debugger has the capability of performing only a mass erase operation.

Chapter 10
Signal Multiplexing and Signal Descriptions

10.1 Introduction

This chapter illustrates which of this device's signals are multiplexed on which external pin.

The System Options Register 0 (SIM_SOPT0) register controls which signal is present on the external pin. Refer to that register to find the detailed control operation of a specific multiplexed pin.

10.2 Pinout

10.2.1 Signal multiplexing and pin assignments

The following table shows the signals available on each pin and the locations of these pins on the devices supported by this document.

NOTE

VDD1 and VDD2 must be short on PCB.

PTA6 and PTA7 are true open drain pins. The external pullup resistor must be added to make them output correct values in using I2C0, GPIO, and UART0.

The NC pin must be floating, and do not tie it to the any of the VDD or VSS.

32 QFN	36 WLCSP	Pin Name	Default	ALT0	ALT1	ALT2	ALT3
—	C3	VSS1	VSS1	VSS1			
—	C4	VSS1	VSS1	VSS1			

32 QFN	36 WLCSP	Pin Name	Default	ALT0	ALT1	ALT2	ALT3
—	D3	VDD1	VDD1	VDD1			
—	D4	VDD1	VDD1	VDD1			
1	A1	VREC	VREC	VREC			
2	A2	VDD1	VDD1	VDD1			
3	A3	VSS1	VSS1	VSS1			
4	B3	PTA0	DISABLED	PTA0	SBAR_IN1	EXTAL	
5	B4	PTA1	DISABLED	PTA1	SBAR_OUT0	XTAL	
6	A4	PTA2	DISABLED	PTA2	BUSOUT	SBAR_OUT1	FTM0_CH0
7	A5	PTA3	DISABLED	PTA3	CLAMP	ACMP0_OUT	FTM0_CH1
8	A6	PTA4	SWDIO	PTA4	SWDIO		FTM1_CH0
9	B6	PTA5	SWCLK	PTA5	SWCLK		FTM1_CH1
10	B5	PTA6	DISABLED	PTA6	I2C0_SDA	UART0_RX	
11	C5	PTA7	DISABLED	PTA7	I2C0_SCL	UART0_TX	
12	C6	VSS2/ VSSA	VSS2/ VSSA	VSS2/ VSSA			
13	D6	VDD2/ VDDA	VDD2/ VDDA	VDD2/ VDDA			
14	E6	PTB0/ RESET_b/ NMI_b	RESET_b	PTB0	IRQ	NMI_b	RESET_b
15	D5	PTB1	DISABLED	PTB1	ADCAD0	SBAR_IN0	
16	E5	PTB2	DISABLED	PTB2	ADCAD1	ACMP0_IN0	
17	F6	PTB3	DISABLED	PTB3	ADCAD2	ACMP0_IN1	
18	E4	PTB4	DISABLED	PTB4	ADCAD3		
19	F5	VSS3	VSS3	VSS3			
20	F4	VREFH	VREFH	VREFH			
21	F3	VOUT_FB	VOUT_FB	VOUT_FB			
23	F2	VOUT	VOUT	VOUT			
24	F1	ISENS	ISENS	ISENS			
25	E2	CLC1	CLC1	CLC1			
26	D2	VLC	VLC	VLC			
27	E1	GD	GD	GD			
28	D1	VBOOT	VBOOT	VBOOT			
29	C2	AD_IN	AD_IN	AD_IN			
30	B2	AD_EN	AD_EN	AD_EN			
31	C1	AC1	AC1	AC1			
32	B1	AC2	AC2	AC2			

10.2.2 Device pin assignment

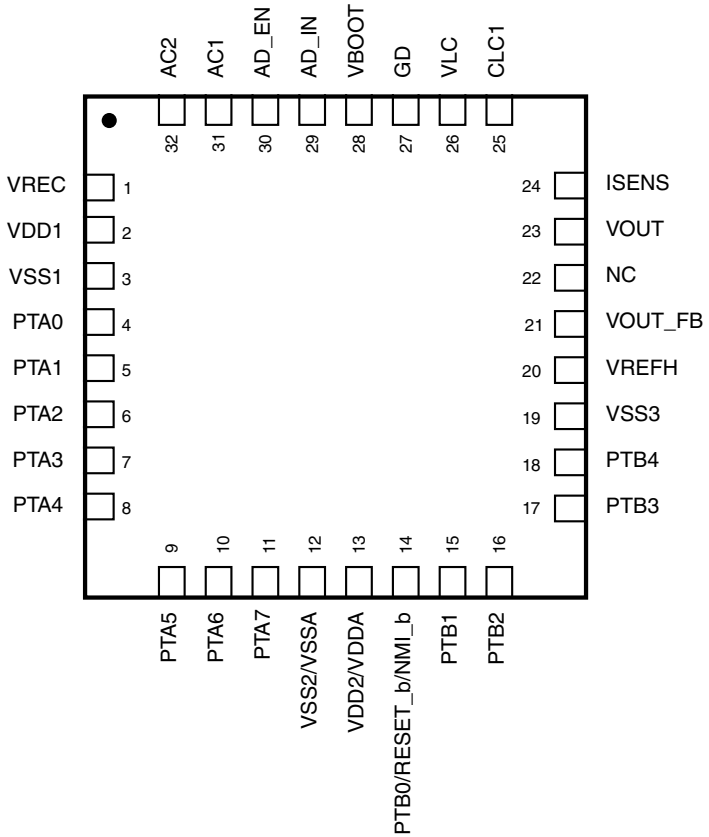


Figure 10-1. 32-pin QFN package

NOTE

The NC pin must be floating, and do not tie it to the VDD or VSS.

	1	2	3	4	5	6	
A	VREC	VDD1	VSS1	PTA2	PTA3	PTA4	A
B	AC2	AD_EN	PTA0	PTA1	PTA6	PTA5	B
C	AC1	AD_IN	VSS1	VSS1	PTA7	VSS2/ VSSA	C
D	VBOOT	VLC	VDD1	VDD1	PTB1	VDD2/ VDDA	D
E	GD	CLC1	CLC2	PTB4	PTB2	PTB0/ RESET_b/ NMI_b	E
F	ISENS	VOUT	VOUT_FB	VREFH	VSS3	PTB3	F
	1	2	3	4	5	6	

Figure 10-2. 36-pin WLCSP package

10.3 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.

10.3.1 Core modules

Table 10-1. SWD signal descriptions

Chip signal name	Module signal name	Description	I/O
SWDIO	SWD_DIO	Serial Wire Debug Data input/output. The SWD_DIO pin is used by an external debug tool for communication and device control. This pin is pulled up internally.	I/O
SWCLK	SWD_CLK	Serial Wire Clock. This pin is the clock for debug logic when in the Serial Wire Debug mode. [1]	I

1. The pad library of this device does not support on-chip pull down; it supports only pullup controlled by PTAPE0, external pulldown resistor is required to fully support SWD protocol.

10.3.2 System modules

Table 10-2. System signal descriptions

Chip signal name	Module signal name	Description	I/O
NMI_b	$\overline{\text{NMI}}$	Non-maskable interrupt **NOTE:** Driving the $\overline{\text{NMI}}$ signal low forces a non-maskable interrupt, if the $\overline{\text{NMI}}$ function is selected on the corresponding pin.	I
RESET_b	$\overline{\text{RESET}}$	Reset bidirectional signal	I/O
VDD1	—	MCU power 1	I/O
VDD2	—	MCU power 2	I
VDDA	—	MCU analog power	I
VREC	—	MCU power input	I
VSS1	—	MCU ground 1	I
VSS2	—	MCU ground 2	I
VSS3	—	MCU ground 3	I
VSSA	—	MCU analog ground	I

10.3.3 Clock modules

Table 10-3. OSC signal descriptions

Chip signal name	Module signal name	Description	I/O
EXTAL	EXTAL	External clock/oscillator input	Analog input
XTAL	XTAL	Oscillator output	Analog output

10.3.4 Memories and Memory Interfaces

10.3.5 Analog

Table 10-4. ADC signal descriptions

Chip signal name	Module signal name	Description	I/O
ADCAD3-ADCAD0	AN3-AN0	Analog channel inputs	I
VREFH/VDDA	VRH_0/VRH_1 and VDDA	High reference voltage for a ADC conversion	I
VSS3/VSSA	VRL_1/VRL_0 and VSSA	Low reference voltage for a ADC conversion	I

Table 10-5. ACMP0 signal descriptions

Chip signal name	Module signal name	Description	I/O
ACMP0_INn[1]	ACMPn[1]	Analog voltage inputs	I
ACMP0_OUT	External output	Comparator output	O

1. n=0, 1, 2. ACMP0_IN2 has no pinout, it is used for internal connection.

Table 10-6. LDO signal descriptions

Chip signal name	Module signal name	Description	I/O
VOUT	VOUT	LDO output power	I
VOUT_FB	VOUT_FB	VOUT feedback	I
ISENS	ISENS	Current sensing signal	I
VBOOT	VBOOT	Charge pump power	O
GD	GD	Gate drive signal for the external N-FET	O
VLC	VLC	Voltage loop compensation	I

Table continues on the next page...

Table 10-6. LDO signal descriptions (continued)

Chip signal name	Module signal name	Description	I/O
CLC1	CLC1	Current loop compensation 1	I

Table 10-7. CNC signal descriptions

Chip signal name	Module signal name	Description	I/O
VREC	V_{REC}	Rectifier voltage input	I
CLAMP	CLAMP	Clamp driver for rectifier overvoltage protection	O
VOUT	V_{OUT}	Wireless power receiver voltage output	I
AD_IN	AD_IN	Wired power voltage input	I
AD_EN	$\overline{AD\_EN}$	Wired power PMOS switch	O
AC1	AC1	AC input 1 from wireless power receiver coil	I
AC2	AC2	AC input 2 from wireless power receiver coil	I

Table 10-8. PMC signal descriptions

Chip signal name	Module signal name	Description	I/O
VREC	VREC	PMC voltage supply input	I
VDD1/VDD2	VDDX	5 V power voltage output	O
VDDA	VDDA	5 V analog supply	O
VREFH	VREFH	Accurate voltage reference for ADC	O
—	VDDF	2.8 V flash supply voltage	O
—	VDD	1.8 V core supply voltage	O

Table 10-9. PGA signal descriptions

Chip signal name	Module signal name	Description	I/O
VOUT	V_{INP}	Voltage input plus, across the current sensing resistor	I
ISENS	V_{INM}	Voltage input minus, across the current sensing resistor	I

10.3.6 Timer modules

Table 10-10. FTM0 signal descriptions

Chip signal name	Module signal name	Description	I/O
FTM0_CH[1:0]	CHn	FTM channel	I/O

Table 10-11. FTM1 signal descriptions

Chip signal name	Module signal name	Description	I/O
FTM1_CH[1:0]	CHn	FTM channel	I/O

Table 10-12. FSKDT signal descriptions

Chip signal name	Module signal name	Description	I/O
—	FSKDT_IN	The FSK signal input channel	O

10.3.7 Communication Interfaces

Table 10-13. I2C0 signal descriptions

Chip signal name	Module signal name	Description	I/O
I2C0_SCL	SCL	Bidirectional serial clock line of the I^2C system.	I/O
I2C0_SDA	SDA	Bidirectional serial data line of the I^2C system.	I/O

Table 10-14. UART0 signal descriptions

Chip signal name	Module signal name	Description	I/O
UART0_TX	TxD	Transmit data	I/O
UART0_RX	RxD	Receive data	I

10.3.8 Human-machine interfaces (HMI)

Table 10-15. GPIO signal descriptions

Chip signal name	Module signal name	Description	I/O
PTA[7:0][1]	PTA7-PTA0	General-purpose input/output	I/O
PTB[4:0][1]	PTB4–PTB0	General-purpose input/output	I/O

1. The available GPIO pins depend on the specific package. See the signal multiplexing section for which exact GPIO signals are available.

Chapter 11
Port Control (PORT)

11.1 Introduction

This device has two sets of I/O ports, which include up to 13 general-purpose I/O pins.

Not all pins are available on all devices.

Many of the I/O pins are shared with on-chip peripheral functions. The peripheral modules have priority over the I/O, so when a peripheral is enabled, the associated I/O functions are disabled.

After reset, the shared peripheral functions are disabled so that the pins are controlled by the parallel I/O except PTA4, PTA5 and PTB0 that are default to SWDIO, SWCLK and $\overline{\text{RESET}}$ function. All of the parallel I/O are configured as high-impedance (Hi-Z). The pin control functions for each pin are configured as follows:

- input disabled (GPIOx_PIDR[PID] = 1),
- output disabled (GPIOx_PDDR[PDD] = 0), and
- internal pullups disabled (PORT_PUE0[PTxPEn] = 0, x = A, B; n = 0-7).

The following figures show the structures of each I/O pin.

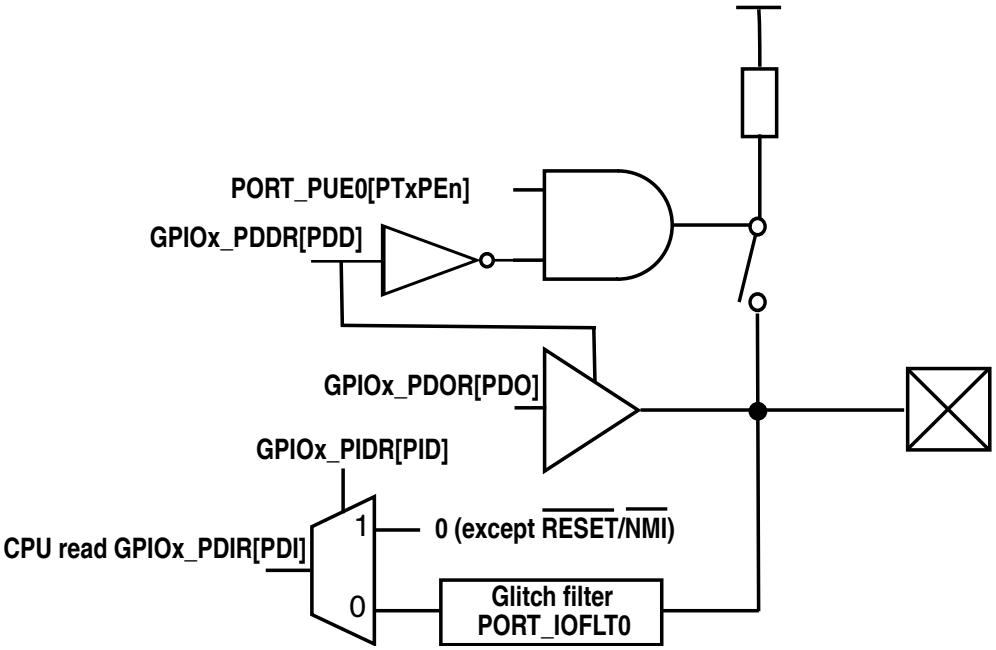


Figure 11-1. Normal I/O structure

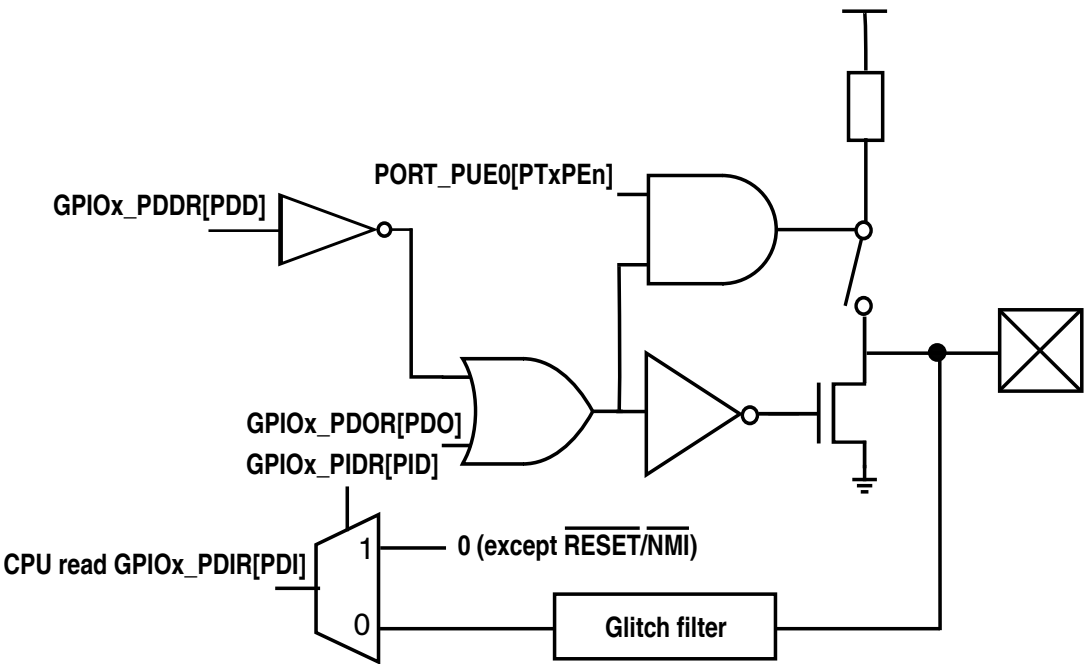


Figure 11-2. PTA6/PTA7 structure

11.2 Port data and data direction

Reading and writing of parallel I/O is accomplished through the port data registers (GPIOx_PDIR/PDOR). The direction, input or output, is controlled through the input disable or output enable registers.

After reset, all parallel I/O default to the Hi-Z state. The corresponding bit in port data direction register (GPIOx_PDDR) or input disable register (GPIOx_PIDR) must be configured for output or input operation. Each port pin has an input disable bit and an output enable bit. When GPIOx_PIDR[PID] = 0, a read from GPIOx_PDIR returns the input value of the associated pin; when GPIOx_PIDR[PID] = 1, a read from GPIOx_PDIR[PDI] returns 0 except for $\overline{\text{RESET}}/\overline{\text{NMI}}$.

NOTE

The GPIOx_PDDR must be clear when the corresponding pin is used as input function to avoid contention. If the corresponding GPIOx_PDDR and GPIOx_PIDR bits are set at same time, read from GPIOx_PDIR will always read the pin status.

When a peripheral module or system function is in control of a port pin, the GPIOx_PDDR register bit still controls what is returned for reads of the GPIOx_PDOR, even though the peripheral system has overriding control of the actual pin direction.

When a shared analog function other than ADC is enabled for a pin, all digital pin functions are disabled. A read of the GPIOx_PDOR returns a value of 0 for any bits that have shared analog functions enabled. In general, whenever a pin is shared with both an alternate digital function and an analog function, the analog function has priority such that if both of the digital and analog functions are enabled, the analog function controls the pin.

A write of valid data to a GPIOx_PDOR must occur before setting the GPIOx_PDDR for the associated port pin. This ensures that the pin will not be driven with an incorrect data value.

11.3 Internal pullup enable

An internal pullup device can be enabled for each port pin by setting the corresponding bit in one of the pullup enable registers (PORT_PUE0). The internal pullup device is disabled if the pin is configured as an output by the parallel I/O control logic, or by any shared peripheral function, regardless of the state of the corresponding pullup enable register bit. The internal pullup device is also disabled if the pin is controlled by an analog function.

NOTE

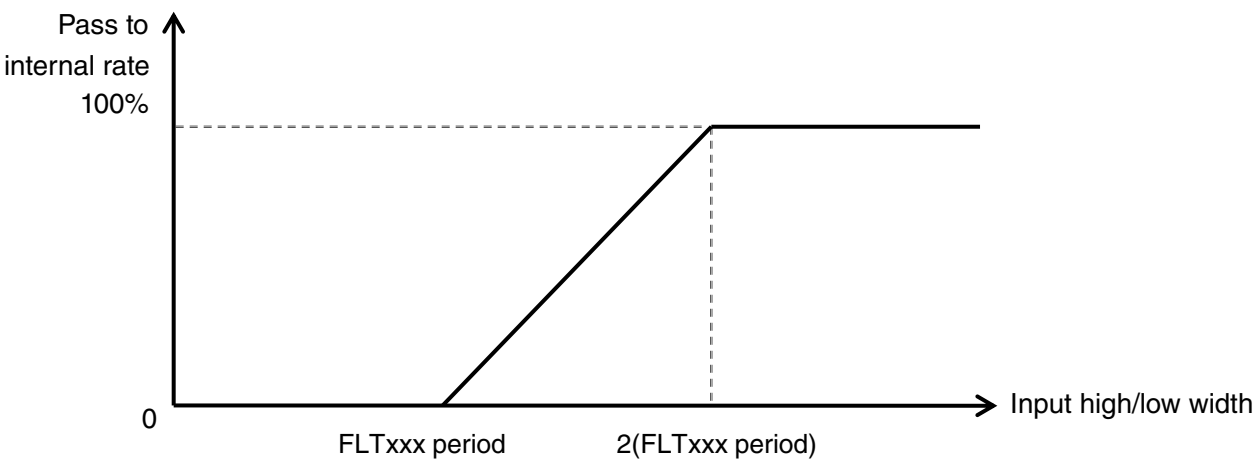
When configuring I2C0 to use "I2C0_SDA(PTA6) and I2C0_SCL(PTA7)" pins, and if an application uses internal pullups instead of external pullups, the internal pullups remain at present setting when the pins are configured as outputs, but

they are automatically disabled to save power when the output values are low.

11.4 Input glitch filter setting

A filter is implemented for PTB0 that is configured as a $\overline{\text{RESET}}$ input. It can be used as a simple low-pass filter to filter any glitch that is introduced from the pin (PTB0) $\overline{\text{RESET}}$. The glitch width threshold can be adjusted easily by setting PORT_IOFLT0[FLTDIVn] between 1~4096 BUSCLKs (or 1~128 LPOCLKs). This configurable glitch filter can take the place of an on board external analog filter, and greatly improve the EMC performance because any glitch will not be wrongly sampled or ignored.

Glitches that are shorter than the selected clock period will be filtered out; glitches that are more than twice the selected clock period will not be filtered out. It will pass to internal circuitry.



Note: FLTxxx is contents in register PORT_IOFLT0.

Figure 11-3. Input glitch filter

11.5 Pin behavior in Stop mode

In Stop mode, all I/O is maintained because internal logic circuitry stays powered up. Upon recovery, normal I/O function is available to the user.

11.6 Port data registers

PORT memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4004_9000	Port Filter Control Register 0 (PORT_IOFLT0)	32	R/W	0000_0000h	11.6.1/95
4004_9004	Port Pullup Enable Register 0 (PORT_PUE0)	32	R/W	0000_0020h	11.6.2/96

11.6.1 Port Filter Control Register 0 (PORT_IOFLT0)

This register sets the filters for input pins. Configure the high/low level glitch width threshold. Glitches that are shorter than the selected clock period will be filtered out; glitches that are more than twice the selected clock period will not be filtered out and will pass to internal circuitry.

Address: 4004_9000h base + 0h offset = 4004_9000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	FLTDIV3			FLTDIV2			FLTDIV1		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														FLTRST	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PORT_IOFLT0 field descriptions

Field	Description
31–29 FLTDIV3	Filter Division Set 3 Port Filter Division Set 3 000 LPOCLK 001 LPOCLK/2 010 LPOCLK/4 011 LPOCLK/8 100 LPOCLK/16 101 LPOCLK/32 110 LPOCLK/64 111 LPOCLK/128
28–26 FLTDIV2	Filter Division Set 2 Port Filter Division Set 2

Table continues on the next page...

PORT_IOFLT0 field descriptions (continued)

Field	Description
	000 BUSCLK/32 001 BUSCLK/64 010 BUSCLK/128 011 BUSCLK/256 100 BUSCLK/512 101 BUSCLK/1024 110 BUSCLK/2048 111 BUSCLK/4096
25–24 FLTDIV1	Filter Division Set 1 Port Filter Division Set 1 00 BUSCLK/2 01 BUSCLK/4 10 BUSCLK/8 11 BUSCLK/16
23–2 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
FLTRST	Filter Selection for Input from RESET 00 No filter. 01 Selects FLTDIV1, and will switch to FLTDIV3 in Stop mode automatically. 10 Selects FLTDIV2, and will switch to FLTDIV3 in Stop mode automatically. 11 FLTDIV3

11.6.2 Port Pullup Enable Register 0 (PORT_PUE0)

Address: 4004_9000h base + 4h offset = 4004_9004h

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			PTBPE4	PTBPE3	PTBPE2	PTBPE1	PTBPE0	PTAPE7	PTAPE6	PTAPE5	PTAPE4	PTAPE3	PTAPE2	PTAPE1	PTAPE0
W																
Reset	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0

PORT_PUE0 field descriptions

Field	Description
31–13 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
12 PTBPE4	Pull Enable for Port B Bit 4 This control field determines if the internal pullup device is enabled for the associated PTB pin. For port B pins that are configured as outputs or Hi-Z, this field has no effect. 0 Pullup is disabled for port B bit 4. 1 Pullup is enabled for port B bit 4.
11 PTBPE3	Pull Enable for Port B Bit 3 This control field determines if the internal pullup device is enabled for the associated PTB pin. For port B pins that are configured as outputs or Hi-Z, this field has no effect. 0 Pullup is disabled for port B bit 3. 1 Pullup is enabled for port B bit 3.
10 PTBPE2	Pull Enable for Port B Bit 2 This control field determines if the internal pullup device is enabled for the associated PTB pin. For port B pins that are configured as outputs or Hi-Z, this field has no effect. 0 Pullup is disabled for port B bit 2. 1 Pullup is enabled for port B bit 2.
9 PTBPE1	Pull Enable for Port B Bit 1 This control field determines if the internal pullup device is enabled for the associated PTB pin. For port B pins that are configured as outputs or Hi-Z, tthis field has no effect. 0 Pullup is disabled for port B bit 1. 1 Pullup is enabled for port B bit 1.
8 PTBPE0	Pull Enable for Port B Bit 0 This control field determines if the internal pullup device is enabled for the associated PTB pin. For port B pins that are configured as outputs or Hi-Z, this field has no effect. 0 Pullup is disabled for port B bit 0. 1 Pullup is enabled for port B bit 0.
7 PTAPE7	Pull Enable for Port A Bit 7 This control field determines if the internal pullup device is enabled for the associated PTA pin. For port A pins that are configured as outputs or Hi-Z, this field has no effect. 0 Pullup is disabled for port A bit 7. 1 Pullup is enabled for port A bit 7.
6 PTAPE6	Pull Enable for Port A Bit 6 This control field determines if the internal pullup device is enabled for the associated PTA pin. For port A pins that are configured as outputs or Hi-Z, this field has no effect. 0 Pullup is disabled for port A bit 6. 1 Pullup is enabled for port A bit 6.

Table continues on the next page...

PORT_PUE0 field descriptions (continued)

Field	Description
5 PTAPE5	Pull Enable for Port A Bit 5 This control field determines if the internal pullup device is enabled for the associated PTA pin. For port A pins that are configured as outputs or Hi-Z, this field has no effect. 0 Pullup is disabled for port A bit 5. 1 Pullup is enabled for port A bit 5.
4 PTAPE4	Pull Enable for Port A Bit 4 This control field determines if the internal pullup device is enabled for the associated PTA pin. For port A pins that are configured as outputs or Hi-Z, this field has no effect. 0 Pullup is disabled for port A bit 4. 1 Pullup is enabled for port A bit 4.
3 PTAPE3	Pull Enable for Port A Bit 3 This control field determines if the internal pullup device is enabled for the associated PTA pin. For port A pins that are configured as outputs or Hi-Z, this field has no effect. **NOTE:** When configuring to use this pin as output high for IIC, the internal pullup device remains active when PTAPE3 is set. It is automatically disabled to save power when output low. 0 Pullup is disabled for port A bit 3. 1 Pullup is enabled for port A bit 3.
2 PTAPE2	Pull Enable for Port A Bit 2 This control field determines if the internal pullup device is enabled for the associated PTA pin. For port A pins that are configured as outputs or Hi-Z, this field has no effect. **NOTE:** When configuring to use this pin as output high for IIC, the internal pullup device remains active when PTAPE2 is set. It is automatically disabled to save power when output low. 0 Pullup is disabled for port A bit 2. 1 Pullup is enabled for port A bit 2.
1 PTAPE1	Pull Enable for Port A Bit 1 This control field determines if the internal pullup device is enabled for the associated PTA pin. For port A pins that are configured as outputs or Hi-Z, this field has no effect. 0 Pullup is disabled for port A bit 1. 1 Pullup is enabled for port A bit 1.
0 PTAPE0	Pull Enable for Port A Bit 0 This control field determines if the internal pullup device is enabled for the associated PTA pin. For port A pins that are configured as outputs or Hi-Z, this field has no effect. 0 Pullup is disabled for port A bit 0. 1 Pullup is enabled for port A bit 0.

Chapter 12
System Integration Module (SIM)

12.1 Introduction

The system integration module (SIM) provides system control and chip configuration registers.

12.1.1 Features

The features of the SIM module are listed below.

- Reset status and device ID information
- System interconnection configuration and special pin enable
- Pin re-mapping control
- System clock gating control and bus clock division
- System peripherals reset control

12.2 Memory map and register definition

This section includes the module memory map and detailed descriptions of all registers.

SIM memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4004_8000	System Reset Status and ID Register (SIM_SRSID)	32	R	See section	12.2.1/101
4004_8004	System Options Register 0 (SIM_SOPT0)	32	R/W	See section	12.2.2/104
4004_8008	Trigger Crossbar Configuration Register 0 (SIM_TBARCFG0)	32	R/W	0000_0000h	12.2.3/107
4004_800C	Trigger Crossbar Configuration Register 1 (SIM_TBARCFG1)	32	R/W	0000_0000h	12.2.4/107

Table continues on the next page...

SIM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4004_8010	Signal Crossbar Configuration Register (SIM_SBARCFG)	32	R/W	0000_0000h	12.2.5/108
4004_8014	FSK Configuration Register (SIM_XBARCFG)	32	R/W	0000_0000h	12.2.6/109
4004_8018	System Clock Gating Control Register (SIM_SCGC)	32	R/W	0000_3000h	12.2.7/110
4004_801C	Reset control Register (SIM_RST)	32	R/W	0000_0000h	12.2.8/113
4004_8020	Clock Divider Register (SIM_CLKDIV)	32	R/W	See section	12.2.9/115
4004_8024	CNC external clamp and LDO external shutdown Flags Register (SIM_FLG)	32	R/W	0000_0000h	12.2.10/117
4004_8028	AOI function Configuration Register (SIM_AOICFG)	32	R/W	0000_0000h	12.2.11/118
4004_802C	ACMP channel 2 Configuration and SIM_SCGC[ADC] write enable Register (SIM_CR)	32	R/W	0000_0100h	12.2.12/121
4004_8030	Universally Unique Identifier Low Register (SIM_UUIDL)	32	R	Undefined	12.2.13/122
4004_8034	Universally Unique Identifier Middle Low Register (SIM_UUIDML)	32	R	Undefined	12.2.14/122
4004_8038	Universally Unique Identifier Middle High Register (SIM_UUIDMH)	32	R	Undefined	12.2.15/123
4004_803C	Peripherals IFR bits Register 0 (SIM_IFR0)	32	R	Undefined	12.2.16/123
4004_8040	Peripherals IFR bits Register 1 (SIM_IFR1)	32	R	Undefined	12.2.17/124
4004_8044	Peripherals IFR bits Register 2 (SIM_IFR2)	32	R	Undefined	12.2.18/124
4004_8048	Peripherals IFR bits Register 3 (SIM_IFR3)	32	R	Undefined	12.2.19/125

12.2.1 System Reset Status and ID Register (SIM_SRSID)

NOTE

The reset value of the field within bits 13–1 (except the bit7 POR and the bit1 LVD) is 1, if its corresponding reset source caused that reset.

Address: 4004_8000h base + 0h offset = 4004_8000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	FAMID				SUBFAMID				0							
W																
Reset	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
POR	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
LVD	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	0		SACKERR	0	MDMAP	SW	LOCKUP	0	POR	PIN	WDOG	0		LOC	LVD	0
W																
Reset	0	0	0	0	0	0	0	0	*	0	0	0	0	0	0	0
POR	0	0	0	0	0	0	0	0	1*	0	0	0	0	0	1	0
LVD	0	0	0	0	0	0	0	0	1*	0	0	0	0	0	1	0

* Notes:

- POR field: After POR reset, POR reset value = 1. LVD reset value = u (unaffected).

SIM_SRSID field descriptions

Field	Description
31–28 FAMID	Wireless power controller Family ID 0000 Wireless power receiver (WPR) 0001 Wireless charger receiver (WCR) 1000 Wireless charger transmitter (WCT) other Reserved
27–24 SUBFAMID	Wireless power controller Sub-family power ID 0000 1 W 0001 5 W 0010 15 W other Reserved
23–14 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
13 SACKERR	Stop Mode Acknowledge Error Reset Indicates that after an attempt to enter Stop mode, a reset has been caused by the failure of one or more I2Cs to acknowledge within approximately 50 ms(1024 clock cycles of the 20 kHz LPO) to enter Stop mode. 0 Reset is not caused by peripheral failure to acknowledge attempt to enter Stop mode. 1 Reset is caused by peripheral failure to acknowledge attempt to enter Stop mode.
12 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
11 MDMAP	MDM-AP System Reset Request Indicates a reset has been caused by the host debugger system setting of the System Reset Request field in the MDM-AP Control Register. 0 Reset is not caused by the host debugger system setting of the System Reset Request field. 1 Reset is caused by the host debugger system setting of the System Reset Request field.
10 SW	Software Indicates a reset has been caused by software setting of the SYSRESETREQ bit in Application Interrupt and Reset Control Register in the ARM core. 0 Reset is not caused by software setting of the SYSRESETREQ bit. 1 Reset caused by software setting of the SYSRESETREQ bit
9 LOCKUP	Core Lockup Indicates a reset has been caused by the ARM core indication of a LOCKUP event. 0 Reset is not caused by core LOCKUP event. 1 Reset is caused by core LOCKUP event.
8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7 POR	Power-on Reset Causes reset by the power-on detection logic. When the internal supply voltage is ramping up, the LVD reset status field is also set at that time, to indicate that the reset has occurred while the internal supply was below the LVD threshold.

Table continues on the next page...

SIM_SRSID field descriptions (continued)

Field	Description
	NOTE: This bit field is set to 1 by POR, unaffected by LVD reset, and set to 0 at any other reset condition. 0 Reset not caused by POR. 1 POR-caused reset.
6 PIN	External Reset Pin Causes reset by an active low-level on the external reset pin. 0 Reset is not caused by external reset pin. 1 Reset originates from external reset pin.
5 WDOG	Watchdog (WDOG) Causes reset by the WDOG timer timing out. This reset source may be blocked by WDOG_CS1[EN] = 0. 0 Reset is not caused by WDOG timeout. 1 Reset is caused by WDOG timeout.
4–3 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
2 LOC	Internal Clock Source (ICS) Module Reset Causes reset by an ICS module reset. 0 Reset is not caused by the ICS module. 1 Reset is caused by the ICS module.
1 LVD	Low Voltage Detection If the supply drops below the LVD trip voltage, an LVD reset occurs. **NOTE:** This bit field is set to 1 on POR and LVD reset, and set to 0 on any other reset. 0 Reset is not caused by LVD trip or POR. 1 Reset is caused by LVD trip or POR.
0 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

12.2.2 System Options Register 0 (SIM_SOPT0)

NOTE

RSTPE and NMIE are write-once only on each normal reset.

Address: 4004_8000h base + 4h offset = 4004_8004h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	DELAY								DLYACT	0			CLKOE	BUSREF		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
POR/ LVD	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	SBARIN1EN	SBARIN0EN	UARTTXEN	UARTRXEN	FTM1CHEN[1:0]		FTM0CHEN[1:0]		0				SWDE	RSTPE	NMIE	0
W																
Reset	0	0	1	1	1	1	1	1	0	0	0	0	1	u*	u*	0
POR/ LVD	0	0	1	1	1	1	1	1	0	0	0	0	1	1	0	0

* Notes:
- u = Unaffected by reset.

SIM_SOPT0 field descriptions

Field	Description
31–24 DELAY	ADC external Trigger Delay Specifies the delay from trigger source to ADC hardware trigger. The 8-bit modulo value allows the delay from 0 to 255 upon the BUSREF clock settings. This is a one-shot counter that starts ticking when the trigger arrives and stops ticking when the counter value reaches the modulo value that is defined.
23 DLYACT	ADC external Trigger Delay Active This read-only field specifies the status if the ADC initial or match delay is active. This field is set when an ADC trigger arrives and the delay counter is ticking. Otherwise, this field is cleared. 0 The delay is inactive. 1 The delay is active.
22–20 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
19 CLKOE	Bus Clock Output Enable Enables bus clock output on PTA2 0 Bus clock output is disabled on PTA2. 1 Bus clock output is enabled on PTA2.
18–16 BUSREF	BUS Clock Output select Enables bus clock output PTA2 via an optional prescaler. 000 Bus 001 Bus divided by 2 010 Bus divided by 4 011 Bus divided by 8 100 Bus divided by 16 101 Bus divided by 32 110 Bus divided by 64 111 Bus divided by 128
15 SBARIN1EN	SBAR_IN1 pin mapping enable on PTA0 0 SBAR_IN1 on PTA0 is disabled. 1 SBAR_IN1 on PTA0 is enabled.
14 SBARIN0EN	SBAR_IN0 pin mapping enable on PTB1 0 SBAR_IN0 on PTB1 is disabled. 1 SBAR_IN0 on PTB1 is enabled.
13 UARTTXEN	UART0_TX pin mapping enable on PTA7 **NOTE:** Both UARTTXEN and UARTRXEN must be set to 1, if using UART single wire mode. 0 UART0_TX on PTA7 is disabled. 1 UART0_TX on PTA7 is enabled.
12 UARTRXEN	UART0_RX pin mapping enable on PTA6 **NOTE:** Both UARTTXEN and UARTRXEN must be set to 1, if using UART single wire mode.

Table continues on the next page...

SIM_SOPT0 field descriptions (continued)

Field	Description
	0 UART0_RX on PTA6 is disabled. 1 UART0_RX on PTA6 is enabled.
11–10 FTM1CHEN[1:0]	FTM1 Channel mapping Enable FTM1CHEN[1] controls the enable of FTM1_CH1 mapping on PTA5, while FTM1CHEN[0] controls the enable of FTM1_CH0 mapping on PTA4. For example, when FTM1CHEN = 10, it enables FTM1_CH1 and disables FTM1_CH0.
9–8 FTM0CHEN[1:0]	FTM0 Channel mapping Enable FTM0CHEN[1] controls the enable of FTM0_CH1 mapping on PTA3, while FTM0CHEN[0] controls the enable of FTM0_CH0 mapping on PTA2. For example, when FTM0CHEN = 01, it disables FTM0_CH1 and enables FTM0_CH0.
7–4 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
3 SWDE	Single Wire Debug port pin mapping Enable Enables the PTA4 pin to function as SWDIO, and PTA5 pin function as SWCLK. When SWDE is cleared, the two pins function as PTA4 and PTA5. This pin defaults to the SWDIO and SWCLK function following any MCU reset. 0 SWDIO pin mapping on PTA4 is disabled, and SWCLK pin mapping on PTA5 is disabled. 1 PTA4 as SWDIO function, and PTA5 as SWCLK function.
2 RSTPE	RESET Pin mapping Enable This write-once field can be written after any reset. When RSTPE is set, the PTB0 pin functions as $\overline{\text{RESET}}$. When cleared, the pin functions as one of its alternative functions. This pin defaults to $\overline{\text{RESET}}$ following an MCU POR. Other resets do not affect this field. When RSTPE is set, an internal pullup device on $\overline{\text{RESET}}$ is enabled. 0 $\overline{\text{RESET}}$ pin mapping is disabled on PTB0. 1 PTB0 pin functions as $\overline{\text{RESET}}$.
1 NMIE	NMI pin mapping Enable This write-once field can be written after any reset. When NMIE is set (and RSTPE is cleared), the PTB0 pin functions as $\overline{\text{NMI}}$. When cleared, the pin functions as one of its alternative functions. This pin defaults to $\overline{\text{RESET}}$ following an MCU POR. Other resets do not affect this bit. When NMIE is set, an internal pullup device on $\overline{\text{NMI}}$ is enabled. **NOTE:** The RSTPE bit takes higher priority than NMIE and GPIO. 0 $\overline{\text{NMIE}}$ pin mapping is disabled on PTB0. 1 PTB0 pin functions as $\overline{\text{NMI}}$, when RSTPE = 0.
0 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

12.2.3 Trigger Crossbar Configuration Register 0 (SIM_TBARCFG0)

See Figure 12-20 for more detailed information.

Address: 4004_8000h base + 8h offset = 4004_8008h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	LDOSHUTDOWN2				LDOSHUTDOWN1				LDOSHUTDOWN0				CNCEXCLAMP			
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	ADCLOADOK				ADCABORT				ADCRESTART				ADCTRIG			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SIM_TBARCFG0 field descriptions

Field	Description
31–28 LDOSHUTDOWN2	LDO external Shutdown source 2 TBAR_INPUT*n* to be muxed as LDOSHUTDOWN2.
27–24 LDOSHUTDOWN1	LDO external Shutdown source 1 TBAR_INPUT*n* to be muxed as LDOSHUTDOWN1.
23–20 LDOSHUTDOWN0	LDO external Shutdown source 0 TBAR_INPUT*n* to be muxed as LDOSHUTDOWN0.
19–16 CNCEXCLAMP	CNC External Clamp source selection
15–12 ADCLOADOK	ADC LoadOK source selection If ADCLOADOK is not the same as ADCRESTART, then ADC_LoadOK is connected to 0.
11–8 ADCABORT	ADC sequence Abort source selection
7–4 ADCRESTART	ADC Restart source selection
ADCTRIG	ADC Trigger source selection

12.2.4 Trigger Crossbar Configuration Register 1 (SIM_TBARCFG1)

Address: 4004_8000h base + Ch offset = 4004_800Ch

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																												TBAROUT			
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_TBARCFG1 field descriptions

Field	Description
31–4 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
TBAROUT	TBAR_OUT source selection TBAR_INPUT*n* to be muxed as TBAR_OUT. TBAR_OUT is connected to SBAR_INPUT15.

12.2.5 Signal Crossbar Configuration Register (SIM_SBARCFG)

See Figure 12-21 for more detailed information.

Address: 4004_8000h base + 10h offset = 4004_8010h

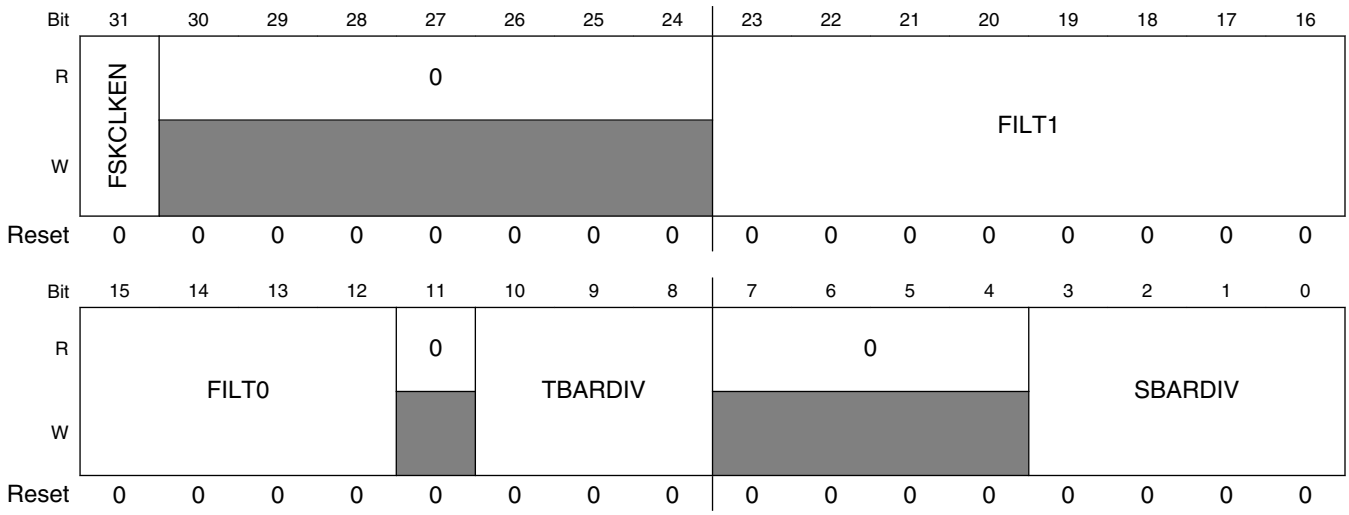
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SBAROUT2				SBAROUT1				SBAROUT0				UARTRX				FTM1CH1IN				FTM1CH0IN				IRQ				FSKDTIN			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SIM_SBARCFG field descriptions

Field	Description
31–28 SBAROUT2	SBAR_OUT2 source selection SBAR_INPUT*n* to be muxed as SBAR_OUT2.
27–24 SBAROUT1	SBAR_OUT1 source selection SBAR_INPUT*n* to be muxed as SBAR_OUT1. **NOTE:** If this field is set to a non-zero value, the SBAR_OUT1 pin mapping is automatically enabled (SBAR_OUT1 pin map on PTA2 for this case).
23–20 SBAROUT0	SBAR_OUT0 source selection SBAR_INPUT*n* to be muxed as SBAR_OUT0. **NOTE:** If this field is set to a non-zero value, the SBAR_OUT0 pin mapping is automatically enabled (SBAR_OUT0 pin map on PTA1 for this case).
19–16 UARTRX	UART_RxD source selection When setting this field to 0, it selects the source from pin PTA6.
15–12 FTM1CH1IN	FTM1_CH1 Input source selection When setting this field to 0, it selects the source from pin PTA5.
11–8 FTM1CH0IN	FTM1_CH0 Input source selection When setting this field to 0, it selects the source from pin PTA4.
7–4 IRQ	IRQ input source selection When setting this field to 0, it selects the source from pin PTB0.
FSKDTIN	FSKDT Input source selection

12.2.6 FSK Configuration Register (SIM_XBARCFG)

Address: 4004_8000h base + 14h offset = 4004_8014h

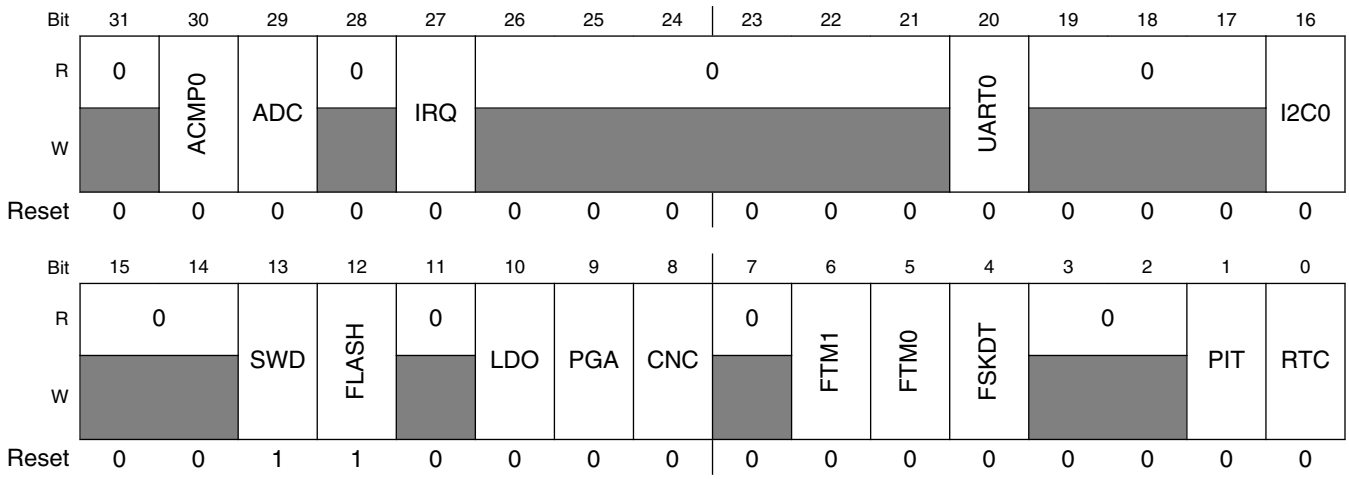


SIM_XBARCFG field descriptions

Field	Description
31 FSKCLKEN	FSK glitch filter and divider Clock Enable control This bit is used to control the glitch filter and divider related clock. 0 Disables the FSK filter/divider clock. 1 Enables the FSK filter/divider clock.
30–24 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
23–16 FILT1	The second glitch filter width configuration control
15–12 FILT0	The first glitch filter width configuration control
11 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
10–8 TBARDIV	TBAR FSK Divider source control N Divided by 2^N (if N > 5, then this division factor is not applicable).
7–4 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
SBARDIV	SBAR FSK Divider source control N Divided by 2^N (if N > 9, then this division factor is not applicable).

12.2.7 System Clock Gating Control Register (SIM_SCGC)

Address: 4004_8000h base + 18h offset = 4004_8018h



SIM_SCGC field descriptions

Field	Description
31 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
30 ACMP0	ACMP0 Clock Gating Control Controls the clock gating to the ACMP0 module. 0 Bus clock to the ACMP0 module is disabled. 1 Bus clock to the ACMP0 module is enabled.
29 ADC	ADC Clock Gating Control Controls the clock gating to the ADC module. 0 Bus clock to the ADC module is disabled. 1 Bus clock to the ADC module is enabled.
28 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
27 IRQ	IRQ Clock Gating Control Controls the clock gating to the IRQ module. 0 Bus clock to the IRQ module is disabled. 1 Bus clock to the IRQ module is enabled.
26–21 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
20 UART0	UART0 Clock Gating Control Controls the clock gating to the UART0 module.

Table continues on the next page...

SIM_SCGC field descriptions (continued)

Field	Description
	0 Bus clock to the UART0 module is disabled. 1 Bus clock to the UART0 module is enabled.
19–17 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
16 I2C0	I^2C0 Clock Gating Control Controls the clock gating to the I^2C0 module. 0 Bus clock to the I^2C0 module is disabled. 1 Bus clock to the I^2C0 module is enabled.
15–14 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
13 SWD	Single Wire Debugger (SWD) Clock Gating Control Controls the clock gating to the SWD module. 0 Bus clock to the SWD module is disabled. 1 Bus clock to the SWD module is enabled.
12 FLASH	Flash Clock Gating Control Controls the clock gating to the Flash module. 0 Bus clock to the Flash module is disabled. 1 Bus clock to the Flash module is enabled.
11 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
10 LDO	LDO Clock Gating Control Controls the clock gating to the LDO module. 0 Bus clock to the LDO module is disabled. 1 Bus clock to the LDO module is enabled.
9 PGA	PGA Clock Gating Control Controls the clock gating to the PGA module. 0 Bus clock to the PGA module is disabled. 1 Bus clock to the PGA module is enabled.
8 CNC	CNC Clock Gating Control Controls the clock gating to the CNC module. 0 Bus clock to the CNC module is disabled. 1 Bus clock to the CNC module is enabled.
7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
6 FTM1	FTM1 Clock Gating Control Controls the clock gating to the FTM1 module.

Table continues on the next page...

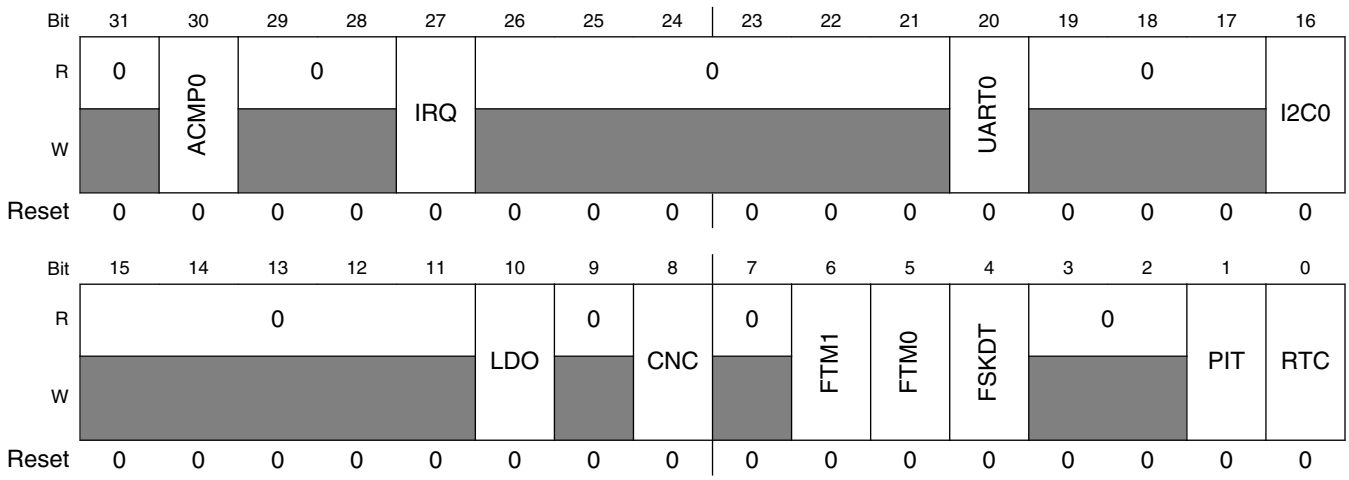
SIM_SCGC field descriptions (continued)

Field	Description
	0 Bus clock to the FTM1 module is disabled. 1 Bus clock to the FTM1 module is enabled.
5 FTM0	FTM0 Clock Gating Control Controls the clock gating to the FTM0 module. 0 Bus clock to the FTM0 module is disabled. 1 Bus clock to the FTM0 module is enabled.
4 FSKDT	FSKDT Clock Gating Control Controls the clock gating to the FSKDT module. 0 Bus clock to the FSKDT module is disabled. 1 Bus clock to the FSKDT module is enabled.
3–2 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
1 PIT	PIT Clock Gating Control Controls the clock gating to the PIT module. 0 Bus clock to the PIT module is disabled. 1 Bus clock to the PIT module is enabled.
0 RTC	RTC Clock Gating Control Controls the clock gating to the RTC module. 0 Bus clock to the RTC module is disabled. 1 Bus clock to the RTC module is enabled.

12.2.8 Reset control Register (SIM_RST)

This register is used to reset corresponding peripherals, when writing 1 to the bit fields. Reading this register always returns 0.

Address: 4004_8000h base + 1Ch offset = 4004_801Ch



SIM_RST field descriptions

Field	Description
31 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
30 ACMP0	ACMP0 Reset Control Controls the reset to the ACMP0 module. 0 Reset to the ACMP0 module is disabled. 1 Reset to the ACMP0 module is enabled.
29–28 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
27 IRQ	IRQ Reset Control Controls the reset to the IRQ module. 0 Reset to the IRQ module is disabled. 1 Reset to the IRQ module is enabled.
26–21 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
20 UART0	UART0 Reset Control Controls the reset to the UART0 module. 0 Reset to the UART0 module is disabled. 1 Reset to the UART0 module is enabled.

Table continues on the next page...

SIM_RST field descriptions (continued)

Field	Description
19–17 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
16 I2C0	I2C0 Reset Control Controls the reset to the I^2C0 module. 0 Reset to the I^2C0 module is disabled. 1 Reset to the I^2C0 module is enabled.
15–11 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
10 LDO	LDO Reset Control Controls the reset to the LDO module. 0 Reset to the LDO module is disabled. 1 Reset to the LDO module is enabled.
9 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
8 CNC	CNC Reset Control Controls the reset to the CNC module. 0 Reset to the CNC module is disabled. 1 Reset to the CNC module is enabled.
7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
6 FTM1	FTM1 Reset Control Controls the reset to the FTM1 module. 0 Reset to the FTM1 module is disabled. 1 Reset to the FTM1 module is enabled.
5 FTM0	FTM0 Reset Control Controls the reset to the FTM0 module. 0 Reset to the FTM0 module is disabled. 1 Reset to the FTM0 module is enabled.
4 FSKDT	FSKDT Reset Control Controls the reset to the FSKDT module. 0 Reset to the FSKDT module is disabled. 1 Reset to the FSKDT module is enabled.
3–2 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
1 PIT	PIT Reset Control Controls the reset to the PIT module.

Table continues on the next page...

SIM_RST field descriptions (continued)

Field	Description
	0 Reset to the PIT module is disabled. 1 Reset to the PIT module is enabled.
0 RTC	RTC Reset Control Controls the reset to the RTC module. 0 Reset to the RTC module is disabled. 1 Reset to the RTC module is enabled.

12.2.9 Clock Divider Register (SIM_CLKDIV)

This register sets the division factor value for the clock.

NOTE

Carefully configure the OUTDIV1 and OUTDIV2 fields to avoid bus clock frequency higher than 24 MHz.

Address: 4004_8000h base + 20h offset = 4004_8020h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0		OUTDIV1		0			OUTDIV2	0			OUTDIV3	0			
W																
Reset	0	0	u*	u*	0	0	0	u*	0	0	0	u*	0	0	0	0
POR/ LVD	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
POR/ LVD	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

* Notes:
- u = Unaffected by reset.

SIM_CLKDIV field descriptions

Field	Description
31–30 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

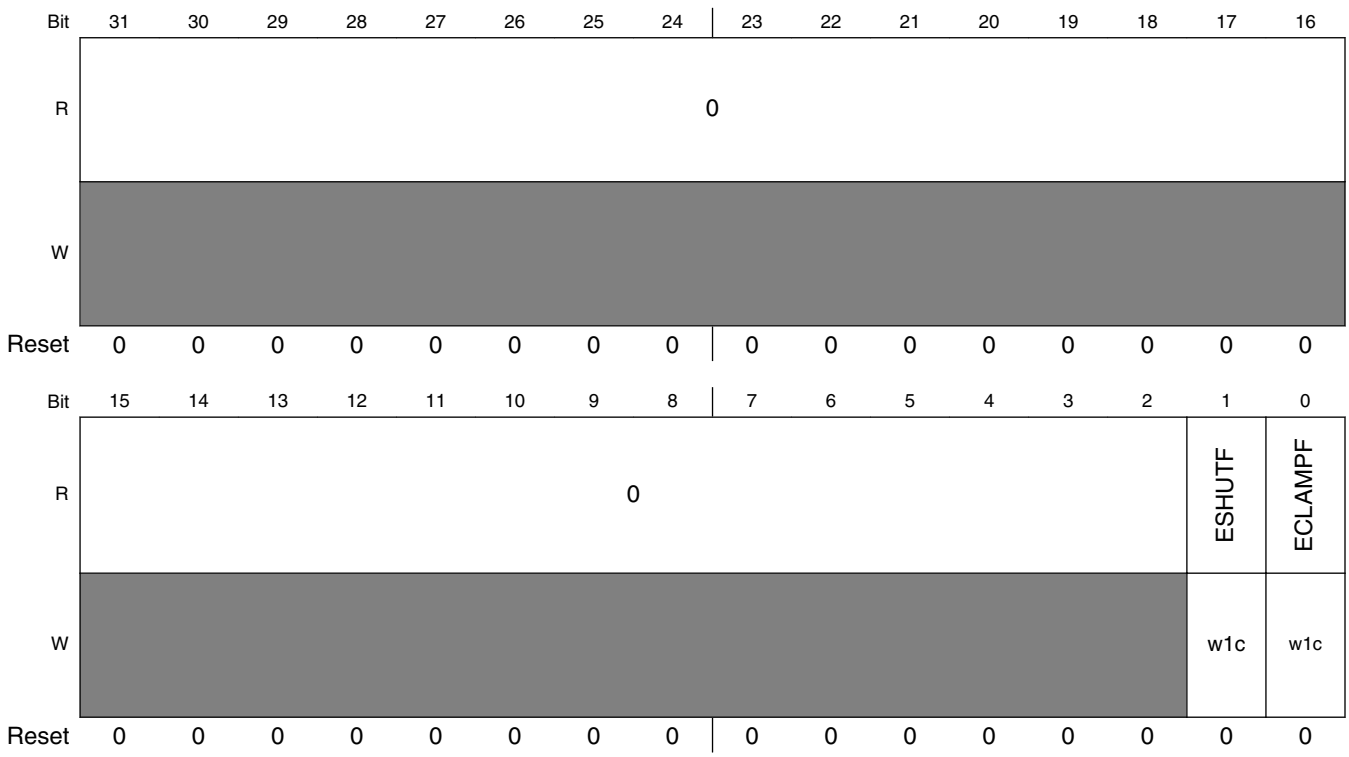
Table continues on the next page...

SIM_CLKDIV field descriptions (continued)

Field	Description
29–28 OUTDIV1	Clock 1 Output Divider value This field sets the divider value for the core/system clock. 00 Same as ICSOUTCLK. 01 ICSOUTCLK is divided by 2. 10 ICSOUTCLK is divided by 3. 11 ICSOUTCLK is divided by 4.
27–25 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
24 OUTDIV2	Clock 2 Output Divider value This field sets the divider value for the bus/FLASH, following OUTDIV1. 0 Not divided from Divider1. 1 Divided by 2 from Divider1.
23–21 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
20 OUTDIV3	Clock 3 Output Divider value This field sets the divider value for the timers (FTM0, FTM1, FSKDT). 0 Same as ICSOUTCLK. 1 ICSOUTCLK is divided by 2.
Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

12.2.10 CNC external clamp and LDO external shutdown Flags Register (SIM_FLG)

Address: 4004_8000h base + 24h offset = 4004_8024h



SIM_FLG field descriptions

Field	Description
31–2 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
1 ESHUTF	LDO External Shutdown status Flag This field indicates LDO external shutdown status, and it also drives the LDO external shutdown request input. 0 LDO external shutdown is de-asserted. 1 LDO external shutdown is asserted. Write 1 to clear this flag.
0 ECLAMPF	CNC External Clamp status Flag This field indicates CNC external clamp status, and it also drives the CNC external clamp input. 0 CNC external clamp is de-asserted. 1 CNC external clamp is asserted. Write 1 to clear this flag.

12.2.11 AOI function Configuration Register (SIM_AOICFG)

Address: 4004_8000h base + 28h offset = 4004_8028h

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	PT0_AC		PT0_BC		PT0_CC		PT0_DC		PT1_AC		PT1_BC		PT1_CC		PT1_DC		PT2_AC		PT2_BC		PT2_CC		PT2_DC		PT3_AC		PT3_BC		PT3_CC		PT3_DC	
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_AOICFG field descriptions

Field	Description
31–30 PT0_AC	Product Term 0, A input Configuration This 2-bit field defines the Boolean evaluation associated with the selected input A in product term 0. 00 Force the A input in this product term to a logic zero. 01 Pass the A input in this product term. 10 Complement the A input in this product term. 11 Force the A input in this product term to a logic one.
29–28 PT0_BC	Product Term 0, B input Configuration Defines the Boolean evaluation associated with the selected input B in product term 0. 00 Force the B input in this product term to a logic zero. 01 Pass the B input in this product term. 10 Complement the B input in this product term. 11 Force the B input in this product term to a logic one.
27–26 PT0_CC	Product Term 0, C input Configuration Defines the Boolean evaluation associated with the selected input C in product term 0. 00 Force the C input in this product term to a logic zero. 01 Pass the C input in this product term. 10 Complement the C input in this product term. 11 Force the C input in this product term to a logic one.
25–24 PT0_DC	Product Term 0, D input Configuration Defines the Boolean evaluation associated with the selected input D in product term 0. 00 Force the D input in this product term to a logic zero. 01 Pass the D input in this product term. 10 Complement the D input in this product term. 11 Force the D input in this product term to a logic one.
23–22 PT1_AC	Product Term 1, A input Configuration Defines the Boolean evaluation associated with the selected input A in product term 1. 00 Force the A input in this product term to a logic zero. 01 Pass the A input in this product term.

Table continues on the next page...

SIM_AOICFG field descriptions (continued)

Field	Description
	10 Complement the A input in this product term. 11 Force the A input in this product term to a logic one.
21–20 PT1_BC	Product Term 1, B input Configuration Defines the Boolean evaluation associated with the selected input B in product term 1. 00 Force the B input in this product term to a logic zero. 01 Pass the B input in this product term. 10 Complement the B input in this product term. 11 Force the B input in this product term to a logic one.
19–18 PT1_CC	Product Term 1, C input Configuration Defines the Boolean evaluation associated with the selected input C in product term 1. 00 Force the C input in this product term to a logic zero. 01 Pass the C input in this product term. 10 Complement the C input in this product term. 11 Force the C input in this product term to a logic one.
17–16 PT1_DC	Product Term 1, D input Configuration Defines the Boolean evaluation associated with the selected input D in product term 1. 00 Force the D input in this product term to a logic zero. 01 Pass the D input in this product term. 10 Complement the D input in this product term. 11 Force the D input in this product term to a logic one.
15–14 PT2_AC	Product Term 2, A input Configuration Defines the Boolean evaluation associated with the selected input A in product term 2. 00 Force the A input in this product term to a logic zero. 01 Pass the A input in this product term. 10 Complement the A input in this product term. 11 Force the A input in this product term to a logic one.
13–12 PT2_BC	Product Term 2, B input Configuration Defines the Boolean evaluation associated with the selected input B in product term 2. 00 Force the B input in this product term to a logic zero. 01 Pass the B input in this product term. 10 Complement the B input in this product term. 11 Force the B input in this product term to a logic one.
11–10 PT2_CC	Product Term 2, C input Configuration Defines the Boolean evaluation associated with the selected input C in product term 2. 00 Force the C input in this product term to a logic zero. 01 Pass the C input in this product term. 10 Complement the C input in this product term. 11 Force the C input in this product term to a logic one.

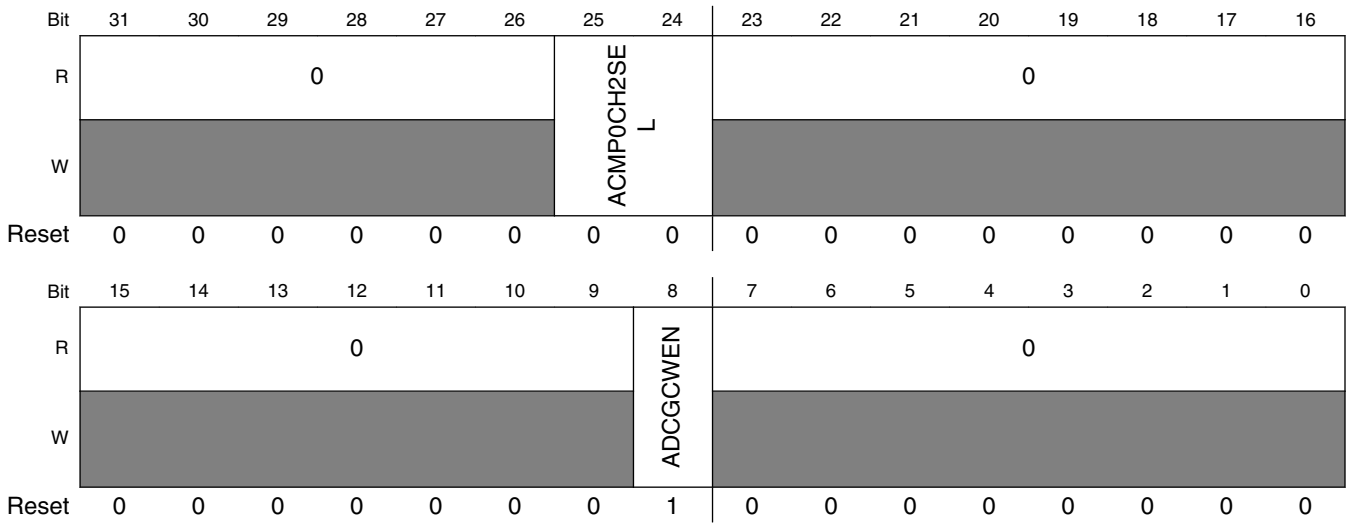
Table continues on the next page...

SIM_AOICFG field descriptions (continued)

Field	Description
9–8 PT2_DC	Product Term 2, D input Configuration Defines the Boolean evaluation associated with the selected input D in product term 2. 00 Force the D input in this product term to a logic zero. 01 Pass the D input in this product term. 10 Complement the D input in this product term. 11 Force the D input in this product term to a logic one.
7–6 PT3_AC	Product Term 3, A input Configuration Defines the Boolean evaluation associated with the selected input A in product term 3. 00 Force the A input in this product term to a logic zero. 01 Pass the A input in this product term. 10 Complement the A input in this product term. 11 Force the A input in this product term to a logic one.
5–4 PT3_BC	Product Term 3, B input Configuration Defines the Boolean evaluation associated with the selected input B in product term 3. 00 Force the B input in this product term to a logic zero. 01 Pass the B input in this product term. 10 Complement the B input in this product term. 11 Force the B input in this product term to a logic one.
3–2 PT3_CC	Product Term 3, C input Configuration Defines the Boolean evaluation associated with the selected input C in product term 3. 00 Force the C input in this product term to a logic zero. 01 Pass the C input in this product term. 10 Complement the C input in this product term. 11 Force the C input in this product term to a logic one.
PT3_DC	Product Term 3, D input Configuration Defines the Boolean evaluation associated with the selected input D in product term 3. 00 Force the D input in this product term to a logic zero. 01 Pass the D input in this product term. 10 Complement the D input in this product term. 11 Force the D input in this product term to a logic one.

12.2.12 ACMP channel 2 Configuration and SIM_SCGC[ADC] write enable Register (SIM_CR)

Address: 4004_8000h base + 2Ch offset = 4004_802Ch



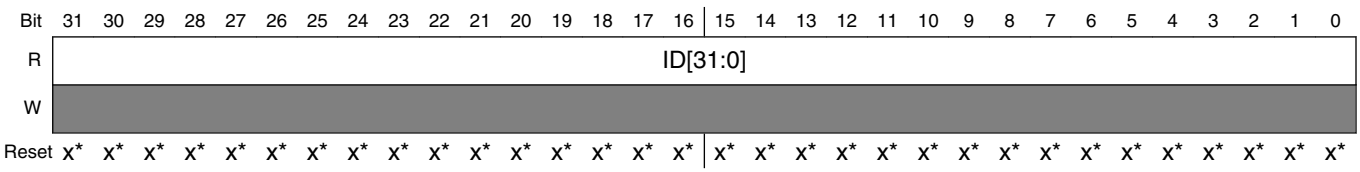
SIM_CR field descriptions

Field	Description
31–26 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
25–24 ACMP0CH2SEL	ACMP0_IN[2] Selection This bit selects the ACMP0 input channel 2 source. 00 AC1 divider output from CNC, the divider value depends on the CNC_ANACFG2[ACDIV], and output enable is controlled by CNC_ANACFG1[AC1DIVOE]. 01 1/5 AD_IN output from CNC, the output enable is controlled by CNC_ANACFG1[VADDIVOE]. 10 1/8 V_{REC} output. 11 7/10 V_{OUT} output from LDO, the output control is enabled by setting LDO_CR[LDOEN], LDO_CR[CPEN], LDO_CR[LDOREGEN], and LDO_CR[CPCLKPS] first and waiting LDO_SR[CPOKF] flag to be set.
23–9 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
8 ADCGCWEN	ADC Clock Gating control bit Write Enable This bit enables user to control the ADC clock gating through SIM_SCGC[ADC]. 0 SIM_SCGC[ADC] cannot be changed. 1 SIM_SCGC[ADC] can be changed.
Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

12.2.13 Universally Unique Identifier Low Register (SIM_UUIDL)

The read-only SIM_UUIDL register contains a serie of numbers to identify the unique device in the family.

Address: 4004_8000h base + 30h offset = 4004_8030h



* Notes:
- x = Undefined at reset.

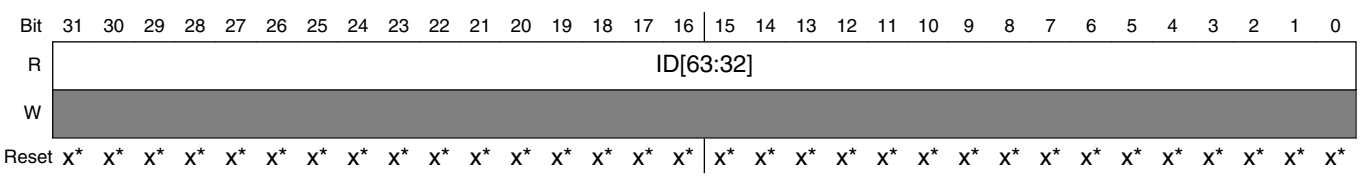
SIM_UUIDL field descriptions

Field	Description
ID[31:0]	Universally Unique Identifier

12.2.14 Universally Unique Identifier Middle Low Register (SIM_UUIDML)

The read-only SIM_UUIDML register contains a serie of numbers to identify the unique device in the family.

Address: 4004_8000h base + 34h offset = 4004_8034h



* Notes:
- x = Undefined at reset.

SIM_UUIDML field descriptions

Field	Description
ID[63:32]	Universally Unique Identifier

12.2.15 Universally Unique Identifier Middle High Register (SIM_UUIDMH)

The read-only SIM_UUIDMH register contains a serie of numbers to identify the unique device in the family.

Address: 4004_8000h base + 38h offset = 4004_8038h

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																ID[80:64]																
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.

SIM_UUIDMH field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
ID[80:64]	Universally Unique Identifier

12.2.16 Peripherals IFR bits Register 0 (SIM_IFR0)

Address: 4004_8000h base + 3Ch offset = 4004_803Ch

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	IFR0_31_21											IFR0_20_10											0									
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.

SIM_IFR0 field descriptions

Field	Description
31–21 IFR0_31_21	PGA gain = 10 offset. The bit 31 indicates the value is positive or negative.
20–10 IFR0_20_10	PGA gain = 8 offset. The bit 20 indicates the value is positive or negative.

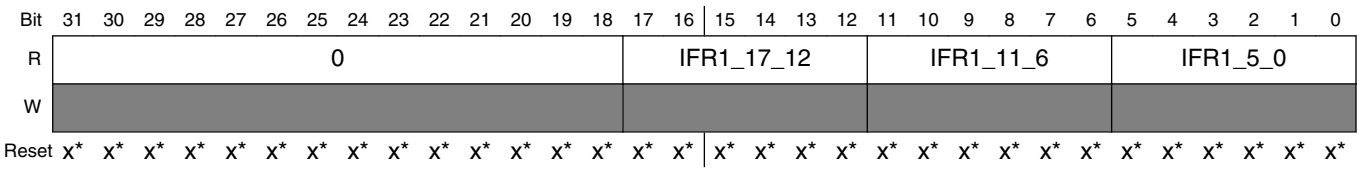
Table continues on the next page...

SIM_IFR0 field descriptions (continued)

Field	Description
Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

12.2.17 Peripherals IFR bits Register 1 (SIM_IFR1)

Address: 4004_8000h base + 40h offset = 4004_8040h



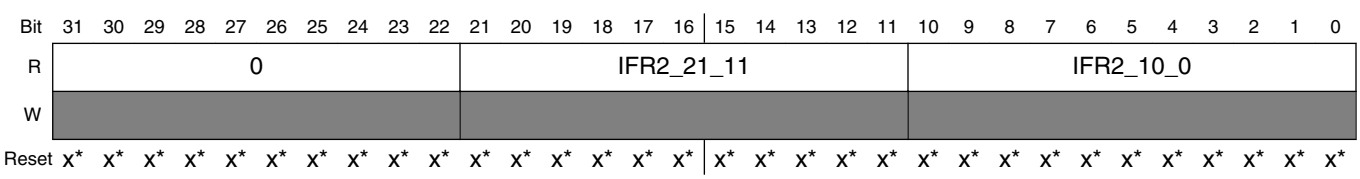
* Notes:
- x = Undefined at reset.

SIM_IFR1 field descriptions

Field	Description
31–18 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
17–12 IFR1_17_12	PMC V_{REFH5V} Trim (3.8 V Trim case).
11–6 IFR1_11_6	PMC V_{REFH5V} Trim (4.2 V Trim case).
IFR1_5_0	PMC V_{REFH5V} Trim (4.6 V Trim case).

12.2.18 Peripherals IFR bits Register 2 (SIM_IFR2)

Address: 4004_8000h base + 44h offset = 4004_8044h



* Notes:
- x = Undefined at reset.

SIM_IFR2 field descriptions

Field	Description
31–22 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
21–11 IFR2_21_11	PGA gain = 15 offset. The bit 21 indicates the value is positive or negative.
IFR2_10_0	PGA gain = 20 offset. The bit 10 indicates the value is positive or negative.

12.2.19 Peripherals IFR bits Register 3 (SIM_IFR3)

Address: 4004_8000h base + 48h offset = 4004_8048h

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									IFR3_22_12											IFR3_11_0												
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.

SIM_IFR3 field descriptions

Field	Description
31–23 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
22–12 IFR3_22_12	V_{BG} conversion value. Normal BG, only 11 bit as BG is only about 1.2 V.
IFR3_11_0	V_{OUT} divider (0.7 V_{OUT}) conversion value.

12.3 Functional description

The following sections describe functional details of the module.

12.3.1 System clock gating and reset control

This device includes clock gating control for each peripheral, that is, the clock to each peripheral can be explicitly gated ON or OFF, using clock gating control fields in the SIM_SCGC register.

Gating the ADC clock during an on-going conversion may cause errors. To avoid such a condition, SIM module offers the ADC clock gating protection. When SIM_CR[ADCGCWEN] is cleared, SIM_SCGC[ADC] is protected from writing.

The device also contains reset control for each peripheral, which means peripherals could be individually reset through software. By setting the control fields in the SIM peripheral reset control register (SIM_RST), the correspondent peripheral could be reset accordingly.

12.3.2 TBAR

The module interconnections are managed by two inter-crossbars: TBAR (trigger crossbar) and SBAR (signal crossbar).

The following figure shows the TBAR configuration.

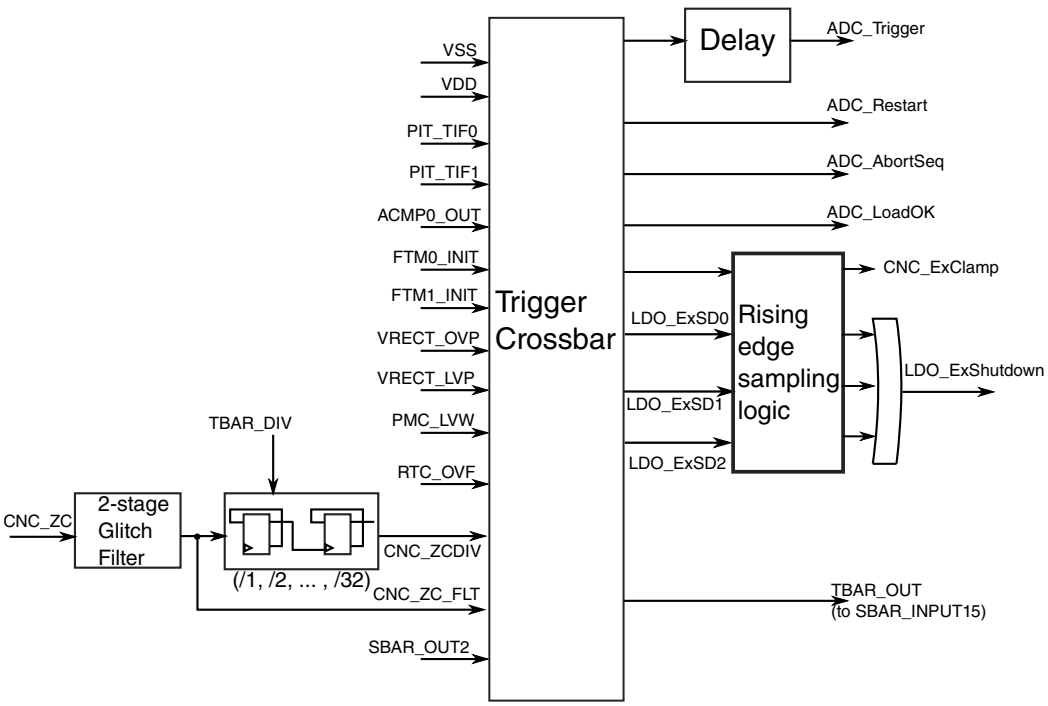


Figure 12-20. TBAR configuration diagram

The following table shows the TBAR inputs.

Table 12-21. TBAR inputs

No.	SEL[3:0]	Name	Description
TBAR_INPUT0	0000b	VSS	V_{SS} (reset value)
TBAR_INPUT1	0001b	VDD	V_{DD}
TBAR_INPUT2	0010b	PIT_TIF0	PIT CH0 timer interrupt flag

Table continues on the next page...

Table 12-21. TBAR inputs (continued)

No.	SEL[3:0]	Name	Description
TBAR_INPUT3	0011b	PIT_TIF1	PIT CH1 timer interrupt flag
TBAR_INPUT4	0100b	ACMP0_OUT	ACMP0 output
TBAR_INPUT5	0101b	FTM0_INIT	Flex Timer 0 initialization trigger
TBAR_INPUT6	0110b	FTM1_INIT	Flex Timer 1 initialization trigger
TBAR_INPUT7	0111b	VRECT_OVP	CNC rectifier overvoltage protection
TBAR_INPUT8	1000b	VRECT_LVP	CNC rectifier low voltage protection
TBAR_INPUT9	1001b	PMC_LVW	PMC V_{DD} low voltage warning
TBAR_INPUT10	1010b	RTC_OVF	RTC overflow flag
TBAR_INPUT11	1011b	CNC_ZCDIV	Divided CNC AC1/AC2 voltage zero-crossing
TBAR_INPUT12	1100b	CNC_ZC_FLT	CNC AC1/AC2 voltage zero-crossing, after the glitch filter
TBAR_INPUT13	1101b	SBAR_OUT2	Signal crossbar output 2
TBAR_INPUT14	1110b	Reserved	Reserved
TBAR_INPUT15	1111b	Reserved	Reserved

The following table shows the TBAR outputs.

Table 12-22. TBAR outputs

No.	Control bits	Name	Description
0	SIM_TBRACFG0[ADCTRIG]	ADC_Trigger	ADC trigger
1	SIM_TBRACFG0[ADCRESTART]	ADC_Restart	ADC restart
2	SIM_TBRACFG0[ADCABORT]	ADC_AbortSeq	ADC abort sequence
3	SIM_TBRACFG0[ADCLOADOK]	ADC_LoadOK	ADC load OK
4	SIM_TBRACFG0[CNCEXCLAMP]	CNC_ExClamp	CNC clamp driver external trigger
5	SIM_TBRACFG0[LDOSHUTDOWN0]	LDO_ExSD0	LDO external shutdown trigger 0
6	SIM_TBRACFG0[LDOSHUTDOWN1]	LDO_ExSD1	LDO external shutdown trigger 1
7	SIM_TBRACFG0[LDOSHUTDOWN2]	LDO_ExSD2	LDO external shutdown trigger 2
8	SIM_TBRACFG1[TBAROUT]	TBAR_OUT	Trigger crossbar output (connected to SBAR input)

Each TBAR output can select any input as its source through SIM_TBARCFG0 and SIM_TBARCFG1. For example, to configure FTM0_INIT as the ADC_Trigger source, user needs to set SIM_TBARCFG0[ADCTRG] as 0101b.

12.3.2.1 CNC_ZC 2-stage glitch filter and divider

A 2-stage glitch filter is implemented for the CNC_ZC (CNC AC1/AC2 voltage zero-crossing) signal.

User can set SIM_XBARCFG[FSKCLKEN] to enable this filter. The TIMER_CLK is used as the filter clock.

The first stage filter is a 4-bit configurable digital glitch filter. To enable this first stage, set SIM_XBARCFG[FILT0] to any non-zero value.

The second stage filter is a 8-bit configurable digital glitch filter. To enable the second stage, set SIM_XBARCFG[FILT1] to any non-zero value.

A digital divider is also implemented on the CNC_ZC signal. User can configure SIM_XBARCFG[TBARDIV] to set the CNC_ZC division factor. The division factor is calculated as $2^{TBARDIV}$. The allowed maximum value of division factor is 32. If the TBARDIV field value is configured to be higher than or equal to 110b, it is then ignored.

12.3.2.2 ADC trigger delay module

A delay module is implemented for ADC trigger signal. This module enables the ADC trigger signal to be synchronized to any trigger source selected through the TBAR.

Configure the SIM_SOPT0[DELAY] bits to define the delay cycle counts. BUS_CLK is used as the delay module clock.

12.3.3 SBAR

The module interconnections are managed by two inter-crossbars: TBAR (trigger crossbar) and SBAR (signal crossbar).

The following figure shows the SBAR configuration.

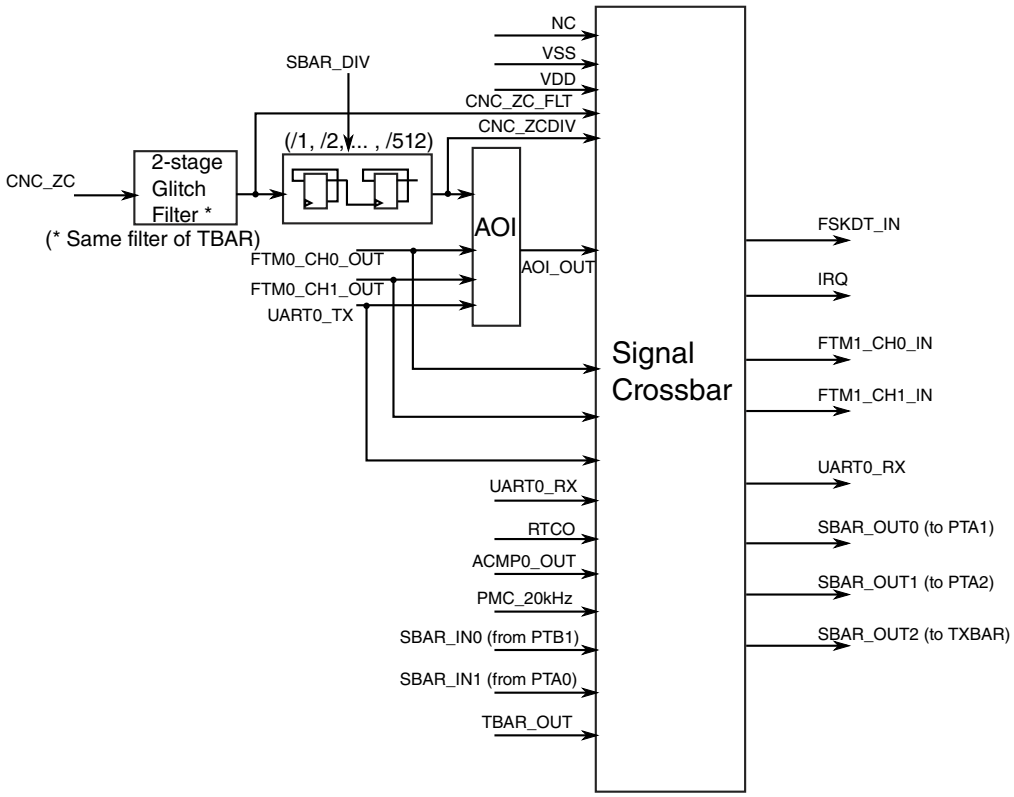


Figure 12-21. SBAR configuration diagram

The following table shows the SBAR inputs.

Table 12-23. SBAR inputs

No.	SEL[3:0]	Name	Description
SBAR_INPUT0	0000b	NC	No connection (tie to 0 for functions except IRQ, UART0_RX, FTM1_CH0_IN, and FTM1_CH1_IN)
SBAR_INPUT1	0001b	VSS	V_{SS}
SBAR_INPUT2	0010b	VDD	V_{DD}
SBAR_INPUT3	0011b	CNC_ZC_FLT	CNC AC1/AC2 voltage zero-crossing, after the glitch filter
SBAR_INPUT4	0100b	CNC_ZCDIV	Divided CNC AC1/AC2 voltage zero-crossing
SBAR_INPUT5	0101b	AOI_OUT	And-Or-Invert (AOI) module output
SBAR_INPUT6	0110b	FTM0_CH0_OUT	Flex Timer 0 channel 0 output
SBAR_INPUT7	0111b	FTM0_CH1_OUT	Flex Timer 0 channel 1 output
SBAR_INPUT8	1000b	UART0_TX	UART0 Tx
SBAR_INPUT9	1001b	UART0_RX	UART0 Rx
SBAR_INPUT10	1010b	RTCO	RTC output
SBAR_INPUT11	1011b	ACMP0_OUT	ACMP0 output
SBAR_INPUT12	1100b	PMC_20kHz	PMC 20 kHz RC oscillator output
SBAR_INPUT13	1101b	SBAR_IN0	SBAR input 0 (mapped on PTB1)

Table continues on the next page...

Table 12-23. SBAR inputs (continued)

No.	SEL[3:0]	Name	Description
SBAR_INPUT14	1110b	SBAR_IN1	SBAR input 1 (mapped on PTA0)
SBAR_INPUT15	1111b	TBAR_OUT	Trigger crossbar output

The following table shows the SBAR outputs.

Table 12-24. SBAR outputs

No.	Control bits	Name	Description
0	SIM_SBARCFG[FSKDTIN]	FSKDT_IN	FSKDT input
1	SIM_SBARCFG[IRQ]	IRQ	IRQ input
2	SIM_SBARCFG[FTM1CH0IN]	FTM1_CH0_IN	Flex Timer 1 channel 0 input
3	SIM_SBARCFG[FTM1CH1IN]	FTM1_CH1_IN	Flex Timer 1 channel 1 input
4	SIM_SBARCFG[UARTRX]	UART0_RX	UART Rx
5	SIM_SBARCFG[SBAROUT0]	SBAR_OUT0	SBAR output to PTA1
6	SIM_SBARCFG[SBAROUT1]	SBAR_OUT1	SBAR output to PTA2
7	SIM_SBARCFG[SBAROUT2]	SBAR_OUT2	SBAR output to TBAR input

Each SBAR output can select any input as its source through the SIM_SBARCFG register. For example, to configure CNC_ZC as the FSKDT input source, user needs to set SIM_SBARCFG[FSKDTIN] as 0011b.

12.3.3.1 CNC_ZC 2-stage glitch filter and divider

On SBAR, the CNC_ZC (CNC AC1/AC2 voltage zero-crossing) signal shares the same 2-stage digital glitch filter as that on TBAR.

User can set SIM_XBARCFG[FSKCLKEN] to enable this filter. The TIMER_CLK is used as the filter clock.

The first stage filter is a 4-bit configurable digital glitch filter. To enable the first stage, set SIM_XBARCFG[FILT0] to any non-zero value.

The second stage filter is a 8-bit configurable digital glitch filter. To enable the second stage, set SIM_XBARCFG[FILT1] to any non-zero value.

A digital divider is also implemented on the CNC_ZC signal. User can configure SIM_XBARCFG[SBARDIV] to set CNC_ZC division factor. The division factor is calculated as $2^{SBARDIV}$. The allowed maximum value of division factor is 512. If the SBARDIV field value is configured to be higher than or equal to 1010b, it is then ignored.

12.3.3.2 And-Or-Invert (AOI) module

The AOI module supports one event output, and this event output represents a user-programmed combinational Boolean function based on four event inputs. The AOI is a highly programmable module for creating combinational Boolean outputs for use as hardware triggers. Each AOI output channel, as shown in the following figure, has one logic function: evaluation of a combinational Boolean expression as a sum of four products, where each product term includes all four selected input sources available as true or complement values.

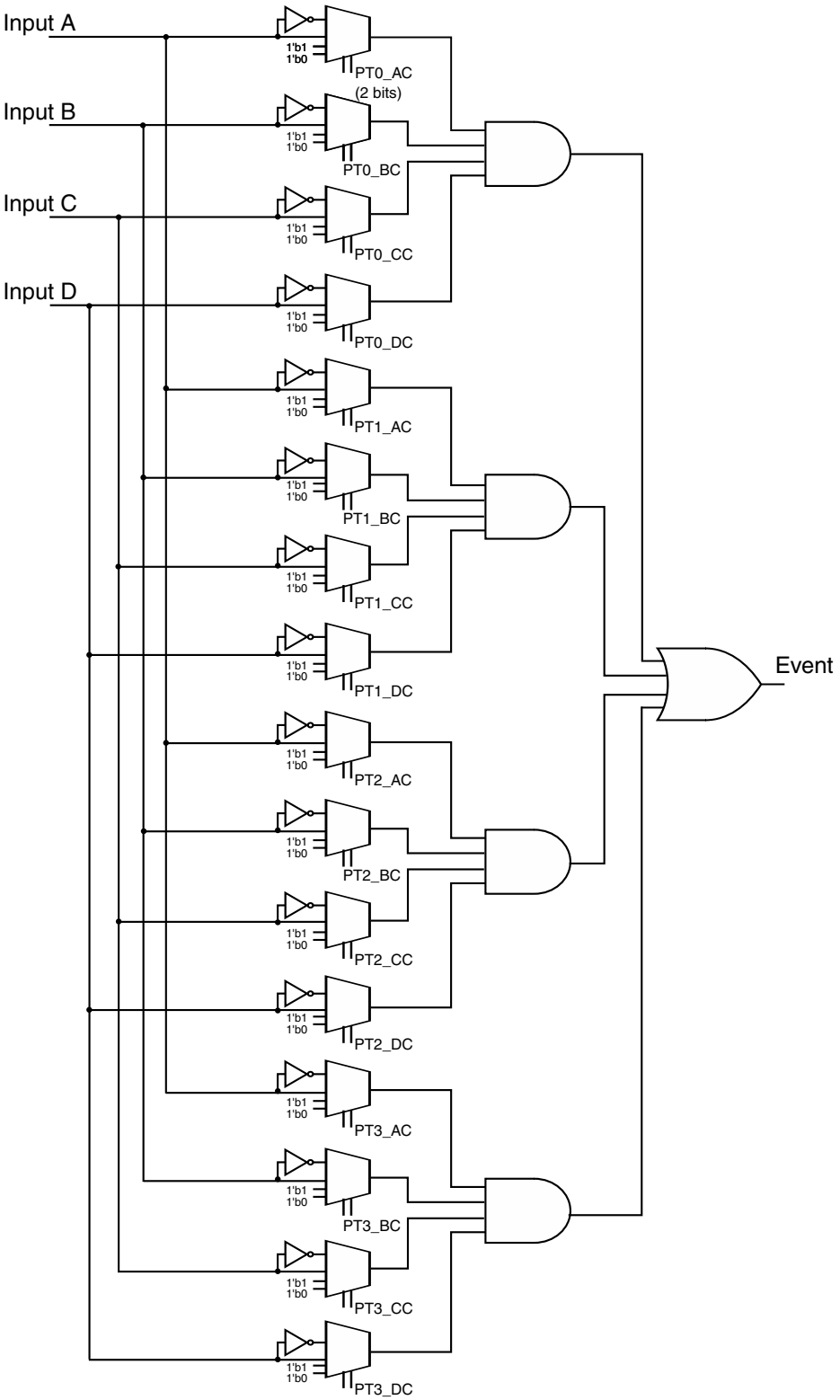


Figure 12-22. Simplified AOI diagram

The AOI module supports access to its programming model via the SIM_AOICFG register.

The AOI module provides a universal Boolean function generator using a 4-term sum of products expression, with each product term containing true or complement values of the four selected event inputs (A, B, C, D). Specifically, the output is defined by the following 4 × 4 Boolean expression:

```
AOI output
= (0,A,Ā,1) & (0,B,B̄,1) & (0,C,C̄,1) & (0,D,D̄,1)    //product term 0
  |(0,A,Ā,1) & (0,B,B̄,1) & (0,C,C̄,1) & (0,D,D̄,1)    //product term 1
  |(0,A,Ā,1) & (0,B,B̄,1) & (0,C,C̄,1) & (0,D,D̄,1)    //product term 2
  |(0,A,Ā,1) & (0,B,B̄,1) & (0,C,C̄,1) & (0,D,D̄,1)    //product term 3
```

Each selected input (A, B, C or D) can be configured to produce a logic 0 or 1 or pass the true or complement of the selected event input, depending on the SIM_AOICFG register value. Each product term uses 8 bits of configuration information, 2 bits for each of the four selected event inputs. The resulting logic provides a simple yet powerful Boolean function evaluation for defining an event output.

Table 12-25. SIM_AOICFG values for simple Boolean expressions

Output	PT0	PT1	PT2	PT3	SIM_AOICFG
A & B	A & B	0	0	0	01011111_00000000_00000000_00000000
A & B & C	A & B & C	0	0	0	01010111_00000000_00000000_00000000
(A & B & C) + D	A & B & C	D	0	0	01010111_11111101_00000000_00000000
A + B + C + D	A	B	C	D	01111111_11011111_11110111_11111101
(A & $\overline{B}$) + ($\overline{A}$ & B)	A & $\overline{B}$	$\overline{A}$ & B	0	0	01101111_10011111_00000000_00000000

The following table shows the AOI module inputs.

Table 12-26. AOI module inputs assignment

No.	Signal	Description
A	CNC_ZCDIV	Divided CNC AC1/AC2 voltage zero crossing
B	FTM0_CH0_OUT	Flex Timer 0 channel 0 output
C	FTM0_CH1_OUT	Flex Timer 0 channel 1 output
D	UART0_TX	UART0 Tx

The AOI output is connected to one of the SBAR inputs.

12.3.3.3 SBAR related pin management

Some of the signal crossbar (SBAR) inputs and outputs share the other on-chip modules' signals which are also mapped to GPIO, like FTM channels, UART0_TX and UART0_RX.

Whether these signals are mapped to/from on-chip modules or to/from GPIO inputs/outputs are controlled by the configuration of the following bits.

- SIM_SOPT0[SBARIN1EN]
- SIM_SOPT0[SBARIN0EN]
- SIM_SOPT0[UARTTXEN]
- SIM_SOPT0[UARTRXEN]
- SIM_SOPT0[FTM1CHEN]
- SIM_SOPT0[FTM0CHEN]

For example, when the bit SIM_SOPT0[SBARIN1EN] is set to 1, SBAR_IN1 input from PTA0 is enabled, otherwise the input is disabled. Similarly for the case of other bits, see the register field descriptions of SIM_SOPT0 for more details.

Chapter 13
Power Management Controller (PMC)

13.1 Chip-specific PMC information

13.1.1 LPO measurement and trim

The low-power oscillator (LPO), 20 kHz RC OSC from PMC, and the LVW on VDDX could be connected to other on-chip timers through the signal crossbar (SBAR) in the SIM module for measurement.

Based on the measurement result, user is allowed to configure this RC OSC frequency by setting PMC_RC20KTRM[OSCOT], after setting the general write protection bit PMC_CTRL[GWREN].

13.2 Introduction

The power management controller (PMC) module contains four main voltage regulators, which provide internal power to other analog modules and the MCU from an external DC source. The nominal output voltage remains steady over the input voltage range of 5.3 V to 20 V.

PMC also contains the power-on reset (POR), the low voltage detection system (low voltage reset and low voltage warning), a high-accuracy reference voltage output, a 20 kHz low-power oscillator and an internal temperature monitor.

13.3 Features

PMC includes the following features:

- Four integrated voltage regulators: $VREG_{VDDX}$, $VREG_{VDDF}$, $VREG_{VDD}$ and $VREG_{VREFH}$
  - 5 volt ($VREG_{VDDX}$ for analog modules)
  - 2.8 volt ($VREG_{VDDF}$ for flash memory and oscillator)
  - 1.8 volt ($VREG_{VDD}$ for digital logics)
  - 3.7~4.9 volt output with 6-bit trim ($VREG_{VREFH}$ for ADC voltage reference)
  - Output supply decoupling capacitors of 4.7~10 µF for $VREG_{VDDX}$ and 10 µF for $VREG_{VREFH}$ required
  - No output supply decoupling capacitors for $VREG_{VDDF}$ and $VREG_{VDD}$ required
  - Reduced performance mode (RPM), full performance mode (FPM), and wake-up from the RPM state via external input
- Integrated power-on reset (POR)
- Low voltage detection system
  - Integrated low voltage reset (LVR) with reset capability in $VREG_{VDDX}$, $VREG_{VDDF}$ and $VREG_{VDD}$
  - Integrated low voltage warning (LVW) indicator in $VREG_{VDDX}$
  - Programmable LVW indicator for VREFH in $VREG_{VREFH}$
- Buffered high-accuracy reference voltage output
  - Factory programmed trim for high-accuracy reference
- 20 kHz low-power oscillator (LPO) clock source
- Integrated temperature sensor allowing both internal and external monitoring

13.4 Overview

This section presents an overview of the PMC module. The following figure illustrates the simplified PMC block diagram.

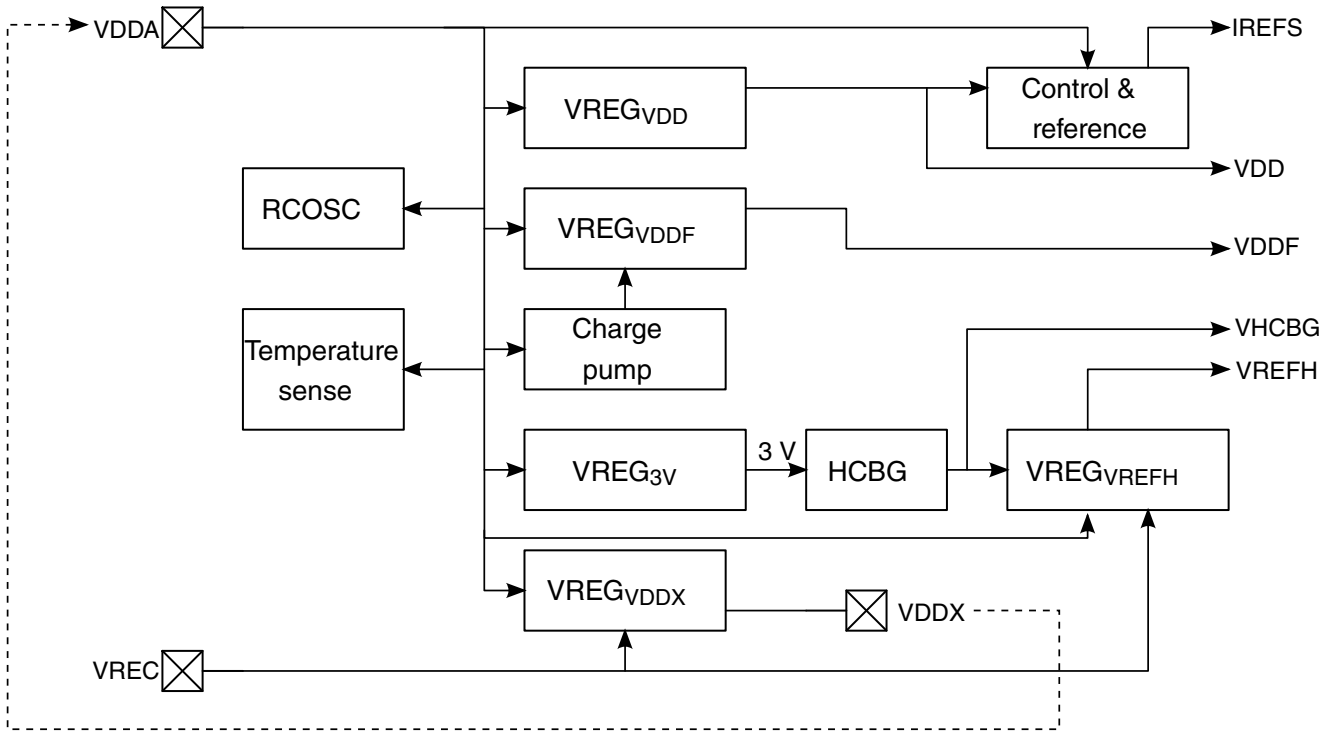


Figure 13-1. Simplified PMC block diagram

13.5 Modes of operation

PMC supports two operating modes: the reduced performance mode (RPM) and the full performance mode (FPM).

13.5.1 Reduced performance mode

When the MCU enters the Stop mode, PMC enters the reduced performance mode (RPM). In this state, the performance of four regulators is reduced but it allows an external logic input to wake up the PMC. When in RPM, the four internal regulators have some limitations as follows:

- $VREG_{VDDX}$ voltage accuracy is not guaranteed to be 5 V and its drive capability is reduced.
- $VREG_{VDDF}$ output voltage drops and its drive capability is very limited.
- $VREG_{VDD}$ output voltage drops and its drive capability is very limited.
- $VREG_{VREFH}$ is off.

When in RPM, the low voltage detection system, the high-accuracy reference voltage and the temperature sensor are off.

13.5.2 Full performance mode

In the full performance mode (FPM) state, the four regulators of PMC perform full regulation of the input supply, to produce the internal regulated supplies in the chip.

13.6 External signal description

PMC has one input signal channel. The following table itemizes all the PMC external pins.

Signal	I/O	Detailed description	Connect from/to
VREC	I	PMC voltage supply input	From pin VREC
VDDX	O	5 V power voltage output	To pin VDD1
VDDA	I	5 V analog supply	From pin VDDA
VREFH	O	Accurate voltage reference for ADC	To pin VREFH
VDDF	O	2.8 V flash supply voltage	To flash module
VDD1.8	O	1.8 V core supply voltage	To MCU digital logics

13.6.1 VREC

VREC is the PMC integrated regulator power supply voltage pin. This is the voltage supply input from which the voltage regulator generates the on-chip voltage supplies. An external decoupling capacitor is required on this pin.

13.6.2 VDDX

VDDX is the 5 V voltage regulator ($VREG_{VDDX}$) output to supply power for the digital I/O drivers. VDDX and VDDA are connected together with an external decoupling capacitor.

13.6.3 VDDA

VDDA is the PMC analog power supply pin. It acts as the 5 V power supply for integrated analog modules such as ADC. VDDX and VDDA are connected together with an external decoupling capacitor.

13.6.4 VREFH

VREFH is the accurate voltage reference ($VREG_{VREFH}$) output. It can be configured from 3.7 V to 4.9 V. An external decoupling capacitor is required on this pin.

13.6.5 VDDF

VDDF is the 2.8 V regulator ($VREG_{VDDF}$) output. It is the power supply of the on-chip NVM block. VDDF is not pinned out because no external decoupling capacitor is required.

13.6.6 VDD1.8

VDD1.8 is the 1.8 V regulator ($VREG_{VDD}$) output. It is the power supply of the on-chip digital logics including CPU and RAM. VDD1.8 is not pinned out because no external decoupling capacitor is required.

13.7 Memory map and register definition

This section includes the module memory map and detailed descriptions of all registers.

PMC memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4007_D000	Control Register (PMC_CTRL)	8	R/W	See section	13.7.1/140
4007_D001	Reset Flags Register (PMC_RST)	8	R/W	See section	13.7.2/141
4007_D002	Temperature Control and Status Register (PMC_TPCTRLSTAT)	8	R/W	00h	13.7.3/142
4007_D003	Temperature Offset Step Trim Register (PMC_TPTM)	8	R/W	See section	13.7.4/143

Table continues on the next page...

PMC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4007_D004	RC Oscillator Offset Step Trim Register (PMC_RC20KTRM)	8	R/W	1Fh	13.7.5/143
4007_D005	Low Voltage Control and Status Register 1 (system 5 V) (PMC_LVCTLSTAT1)	8	R/W	See section	13.7.6/144
4007_D006	Low Voltage Control and Status Register 2 (VREFH) (PMC_LVCTLSTAT2)	8	R/W	See section	13.7.7/145
4007_D007	VREFH Configuration Register (PMC_VREFHCFG)	8	R/W	See section	13.7.8/146
4007_D008	VREFH Low Voltage Warning (LVW) Configuration Register (PMC_VREFHLVW)	8	R/W	02h	13.7.9/146
4007_D009	Status Register (PMC_STAT)	8	R/W	05h	13.7.10/147

13.7.1 Control Register (PMC_CTRL)

NOTE

The VREFDN and RC20KENSTP can be cleared only by POR.

Address: 4007_D000h base + 0h offset = 4007_D000h

Bit	7	6	5	4	3	2	1	0
Read	GWREN	0				VREFDN	RC20KENSTP	0
Write								
Reset	0	0	0	0	0	u*	u*	0
POR	0	0	0	0	0	0	1	0

* Notes:
- u = Unaffected by reset.

PMC_CTRL field descriptions

Field	Description
7 GWREN	General Write protection Enable General write protection enable by sharing with other trim write enable. This bit is the general write enable control for the following registers: • Temperature Offset Step Trim Register (PMC_TPTM) • RC Oscillator Offset Step Trim Register (PMC_RC20KTRM) • VREFH Configuration Register (PMC_VREFHCFG) 0 The general write protection is enabled. 1 The general write protection is disabled. Therefore, the related registers are writable.
6–3 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

Table continues on the next page...

PMC_CTRL field descriptions (continued)

Field	Description
2 VREFDN	VREFH Down This bit controls whether to disable the VREFH regulator. 0 Enables the VREFH regulator. 1 Disables the VREFH regulator.
1 RC20KENSTP	20 kHz RC oscillator Enable in Stop mode The default reset value is 1. In the Run mode this bit is always ON, and in the Stop mode it selects whether to enable the 20 kHz RC oscillator. 0 Disables 20 kHz RC oscillator in the Stop mode. 1 Enables 20 kHz RC oscillator in the Stop mode.
0 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

13.7.2 Reset Flags Register (PMC_RST)

Address: 4007_D000h base + 1h offset = 4007_D001h

Bit	7	6	5	4	3	2	1	0
Read	0	PORF	LVRF	0				
Write								
Reset	0	u*	u*	0	0	0	0	0
POR	0	1	1	0	0	0	0	0
LVR	0	1	1	0	0	0	0	0

* Notes:
- u = Unaffected by reset.

PMC_RST field descriptions

Field	Description
7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
6 PORF	Power-on Reset Flag This bit is set to 1 when a power-on reset (POR) occurs. This flag can only be cleared by writing a 1. Writing a 0 takes no effect. 0 POR does not occur. 1 POR occurs.
5 LVRF	Low Voltage Reset (LVR) Flag This bit is set to 1 when a low voltage reset occurs in the full performance mode (FPM). This flag can only be cleared by writing a 1. Writing a 0 takes no effect.

Table continues on the next page...

PMC_RST field descriptions (continued)

Field	Description
	0 Low voltage reset does not occur. 1 Low voltage reset occurs.
Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

13.7.3 Temperature Control and Status Register (PMC_TPCTRLSTAT)

Address: 4007_D000h base + 2h offset = 4007_D002h

Bit	7	6	5	4	3	2	1	0
Read	0			SWON	TEMPEN	HTDS	HTIE	HTIF
Write								
Reset	0	0	0	0	0	0	0	0

PMC_TPCTRLSTAT field descriptions

Field	Description
7–5 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
4 SWON	Switch On This bit selects bandgap or temperature sensor output to the ADC channel, when the bit TEMPEN is enabled. 0 Selects the temperature sensor output. 1 Selects the bandgap output.
3 TEMPEN	Temperature sensor Enable This bit is the enable control to the temperature sensor. 0 Disables the temperature sensor. 1 Enables the temperature sensor.
2 HTDS	High Temperature Detection Status This read-only bit indicates the temperature status. Writing a value takes no effect. 0 Junction temperature is below the alert level. 1 Junction temperature is above the alert level.
1 HTIE	High Temperature Interrupt Enable 0 Disables the high temperature interrupt. 1 Enables the high temperature interrupt.
0 HTIF	High Temperature Interrupt Flag This bit is set to 1 when the HTDS status bit changes. This flag can only be cleared by writing a 1. Writing a 0 takes no effect. If enabled (HTIE = 1), HTIF causes an interrupt request.

Table continues on the next page...

PMC_TPCTRLSTAT field descriptions (continued)

Field	Description
	0 No change in the HTDS bit. 1 The HTDS bit changes.

13.7.4 Temperature Offset Step Trim Register (PMC_TPTM)

Address: 4007_D000h base + 3h offset = 4007_D003h

Bit	7	6	5	4	3	2	1	0
Read	TRMTPEN	0			TOT[3:0]			
Write								
Reset	0	0	0	0	f	f	f	f

PMC_TPTM field descriptions

Field	Description
7 TRMTPEN	Temperature offset Trim Enable If the bit is set, the temperature sensor offset is enabled. 0 The temperature sensor offset is disabled. TOT[3:0] takes no effect. 1 The temperature sense offset is enabled. TOT[3:0] selects the temperature offset value. See TOT[3:0] for more details.
6–4 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
TOT[3:0]	Temperature Offset step Trim These bits are used to trim the temperature offset. TOT[3:0] determines the trim value for assert or de-assert of system interrupt. Refer to the electrical specifications information in the chip datasheet to get the detailed values. **NOTE:** After de-assert of system reset, a trim value is automatically loaded from the flash memory. Normal IPS writable only after PMC_CTRL[GWREN] is set. **NOTE:** TOT[3:0] can be cleared only by POR.

13.7.5 RC Oscillator Offset Step Trim Register (PMC_RC20KTRM)

Address: 4007_D000h base + 4h offset = 4007_D004h

Bit	7	6	5	4	3	2	1	0
Read	0		OSCOT[5:0]					
Write								
Reset	0	0	0	1	1	1	1	1

PMC_RC20KTRM field descriptions

Field	Description
7–6 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
OSCOT[5:0]	RC Oscillator Offset step Trim These bits are defined as the step trim values for the RC oscillator offset. Refer to the electrical specifications information in the chip datasheet to get the detailed values. **NOTE:** After de-assert of system reset, a trim value is automatically loaded from the flash memory. Normal IPS writable only after PMC_CTRL[GWREN] is set.

13.7.6 Low Voltage Control and Status Register 1 (system 5 V) (PMC_LVCTLSTAT1)

Address: 4007_D000h base + 5h offset = 4007_D005h

Bit	7	6	5	4	3	2	1	0
Read	0			SLVWF		SLVWIE	SLVWSEL	0
Write					SLVWACK			
Reset	0	0	0	0	0	0	0	0
POR	0	0	0	1	0	0	0	0

PMC_LVCTLSTAT1 field descriptions

Field	Description
7–5 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
4 SLVWF	Low Voltage Warning (LVW) Flag for VDDX This read-only status bit indicates a low voltage warning event. SLVWF is set when V_{Supply} transitions below the trip point, or after reset and V_{Supply} is already below V_{LVW}. SLVWF is set to 1 after power-on reset. Therefore, to use the LVW interrupt function, before enabling SLVWIE, SLVWF must be cleared by writing SLVWACK first. 0 Low voltage warning event is not detected. 1 Low voltage warning event is detected.
3 SLVWACK	Low Voltage Warning Acknowledge This write-only bit is used to acknowledge low voltage warning errors. Write 1 to clear SLVWF. Reading always returns 0.
2 SLVWIE	Low Voltage Warning Interrupt Enable Enables hardware interrupt requests for SLVWF. 0 Hardware interrupt disabled (use polling). 1 Request a hardware interrupt when SLVWF = 1.
1 SLVWSEL	Low Voltage Warning Selection

Table continues on the next page...

PMC_LVCTLSTAT1 field descriptions (continued)

Field	Description
	0 4.2 V LVW threshold selected. 1 3.7 V LVW threshold selected.
0 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

13.7.7 Low Voltage Control and Status Register 2 (V_{REFH}) (PMC_LVCTLSTAT2)

Address: 4007_D000h base + 6h offset = 4007_D006h

Bit	7	6	5	4	3	2	1	0
Read	0			RLVWF		RLVWIE	0	
Write					RLVWACK			
Reset	0	0	0	0	0	0	0	0
POR	0	0	0	1	0	0	0	0

PMC_LVCTLSTAT2 field descriptions

Field	Description
7–5 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
4 RLVWF	Low Voltage Warning Flag for VREFH This read-only status bit indicates a low voltage warning event. RLVWF is set when V_{Supply} transitions below the trip point, or after reset and V_{Supply} is already below V_{LVW}. RLVWF is set to 1 after power-on reset. Therefore, to use LVW interrupt function, before enabling RLVWIE, RLVWF must be cleared by writing RLVWACK first. 0 Low voltage warning event is not detected. 1 Low voltage warning event is detected.
3 RLVWACK	Low Voltage Warning Acknowledge This write-only bit is used to acknowledge low voltage warning errors. Write 1 to clear RLVWF. Reading always returns 0.
2 RLVWIE	Low Voltage Warning Interrupt Enable Enables hardware interrupt requests for RLVWF. 0 Hardware interrupt disabled (use polling). 1 Request a hardware interrupt when RLVWF = 1.
Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

13.7.8 V_{REFH} Configuration Register (PMC_VREFHCFG)

Address: 4007_D000h base + 7h offset = 4007_D007h

Bit	7	6	5	4	3	2	1	0
Read	0		T5V					
Write								
Reset	0	0	f	f	f	f	f	f

PMC_VREFHCFG field descriptions

Field	Description
7–6 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
T5V	Trim 5 V reference voltage for ADC Auto-loaded from IFR. After de-assert of system reset, a trim value is automatically loaded from the flash memory. The default is 3.8 V. Refer to SIM_IFR1 for other setting values. **NOTE:** Normal IPS writable only after PMC_CTRL[GWREN] is set.

13.7.9 VREFH Low Voltage Warning (LVW) Configuration Register (PMC_VREFHLVW)

Address: 4007_D000h base + 8h offset = 4007_D008h

Bit	7	6	5	4	3	2	1	0
Read	0						LVWCFG[1:0]	
Write								
Reset	0	0	0	0	0	0	1	0

PMC_VREFHLVW field descriptions

Field	Description
7–2 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
LVWCFG[1:0]	VREFH LVW reference voltage Configuration These bits are used to configure the VREFH low voltage warning threshold value. 00 3.6 V LVW threshold. 01 3.7 V LVW threshold. 10 4.1 V LVW threshold. 11 4.4 V LVW threshold.

13.7.10 Status Register (PMC_STAT)

Address: 4007_D000h base + 9h offset = 4007_D009h

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

PMC_STAT field descriptions

Field	Description
7–3 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
2 HBGRDY	High-accuracy Bandgap Ready flag 0 Below 0.8 V, this bit is cleared. 1 Above 1 V, this bit is asserted.
1 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
0 VREFRDY	VREFH Ready flag **NOTE:** Even after VREFRDY is set, it does not necessarily mean that VREF is okay for use instantly. It needs to wait about 3 ms, in the chip power-on or stop mode, to wake up before using the VREF for ADC. 0 Below 3.0 V, this bit is cleared. 1 Above 3.45 V, this bit is asserted.

13.8 Functional description

The following sections describe functional details of the module.

13.8.1 Voltage regulators

PMC integrates four voltage regulators: $VREG_{VDDX}$, $VREG_{VDDF}$, $VREG_{VDD}$ and $VREG_{VREFH}$. These four voltage regulators are the power supply for all on-chip modules.

13.8.1.1 $VREG_{VDDX}$

$VREG_{VDDX}$ is a 5 V output regulator. It is the power supply for the digital I/Os domain (VDDX) and the analog modules domain (VDDA).

13.8.1.2 $VREG_{VDDF}$

$VREG_{VDDF}$ is a 2.8 V output regulator. It is the power supply for the on-chip NVM module (VDDF).

13.8.1.3 $VREG_{VDD}$

$VREG_{VDD}$ is a 1.8 V output regulator. It is the power supply for the digital logics such as CPU and RAM.

13.8.1.4 $VREG_{VREFH}$

$VREG_{VREFH}$ is a 3.7~4.9 V configurable output regulator. It is the reference voltage output for on-chip analog modules such as ADC.

This regulator can be enabled or disabled by configuring PMC_CTRL[VREFDN]. After POR or enabled, the flag PMC_STAT[VREFRDY] is set, once the regulator output is ready.

The regulator output voltage can be trimmed from 3.7 V to 4.9 V through the PMC_VREFHCFG[T5V]. After reset, a factory trimmed value is automatically loaded to the PMC_VREFHCFG register so that $VREG_{VREFH}$ has a default output voltage. After the write protection enable bit PMC_CTRL[GWREN] is set to 1, user can configure this voltage to other levels according to the application need.

13.8.2 Power-on reset

The POR circuit monitors the $VREG_{VDD}$ outputs (VDD1.8), which represent the MCU VDD domain. When power is initially applied to the MCU, or when the supply voltage drops below the V_{POR} level, the POR circuit causes a reset condition.

When POR reset occurs, the flag bit PMC_RST[PORF] is set to 1. This flag is not affected by other system resets.

13.8.3 Low voltage reset (LVR)

The LVR circuit monitors the 5 V $VREG_{VDDX}$ output (VDDX), the 2.8 V $VREG_{VDDF}$ output (VDDF), and the 1.8 V $VREG_{VDD}$ output (VDD1.8). The LVR circuit can generate a reset when detecting a low voltage condition on VDDX, VDDF or VDD1.8. After an LVR reset occurs, the LVR system holds the MCU in reset status until the supply voltage rises above the low voltage level.

In the FPM mode, when LVR reset occurs, the flag bit PMC_RST[LVRF] is set to 1. The flag LVRF is also set to 1 when a POR occurs, but is not affected by other system resets.

13.8.3.1 LVR in low power mode

The LVR circuit is disabled when PMC enters the reduced performance mode (RPM).

13.8.4 Low voltage warning (LVW)

The LVW circuit monitors the 5 V $VREG_{VDDX}$ outputs (VDDX/VDDA) and the $VREG_{VREFH}$ output (VREFH). The LVW circuit can generate an interrupt when detecting a low voltage condition on VDDX/VDDA or VREFH.

13.8.4.1 LVW on VDDX/VDDA

Low voltage on VDDX/VDDA sets the flag bit PMC_LVCTLSTAT1[SLVWF] to 1. When PMC_LVCTLSTAT1[SLVWIE] is set, the low voltage condition incurs an interrupt.

To clear the SLVWF flag, user should write 1 to PMC_LVCTLSTAT1[SLVWACK].

13.8.4.2 LVW on VREFH

Low voltage on VREFH sets the flag PMC_LVCTLSTAT2[RLVWF] to 1. When PMC_LVCTLSTAT2[RLVWIE] is set, the low voltage condition incurs an interrupt.

To clear the RLVWF flag, user should write 1 to PMC_LVCTLSTAT2[RLVWACK].

The VREFH low voltage level can be configured through the LVWCFG[1:0] bits in PMC_VREFHLVW register.

13.8.4.3 LVW in low power mode

The LVW circuit is disabled when PMC enters the reduced performance mode (RPM).

13.8.5 High-accuracy reference voltage

PMC includes a high-accuracy reference voltage. This voltage is available for other on-chip modules.

After POR, the flag PMC_STAT[HBGRDY] is set to 1 when the high-accuracy output is ready.

The high-accurary reference voltage is always enabled except in RPM mode.

13.8.6 Temperature sensor

PMC includes an internal temperature sensing voltage generator and a high temperature warning comparator. The generator is always enabled in FPM mode and can provide the sensing voltage to ADC to measure the temperature.

To enable the PMC temperature sensor, PMC_TPCTRLSTAT[TEMPEN] should be set to 1.

After enabled, the temperature sensor voltage output can be provided to external modules such as ADC. Depending on the PMC_TPCTRLSTAT[SWON] bit value, the voltage output is selected as follows:

- If SWON = 0, the temperature sensor voltage is as the output.
- If SWON = 1, the bandgap is as the output. Refer to the chip datasheet for the detailed voltage value.

The temperature sensor is disabled in RPM mode.

13.8.6.1 High temperature warning

The temperature sensor integrates a high temperature warning circuit with configurable temperature threshold.

The flag PMC_TPCTRLSTAT[HTDS] shows the temperature status:

- HTDS bit is set when the temperature rises above the HTDS assert threshold.
- HTDS bit is cleared when the temperature falls below the HTDS de-assert threshold.

If PMC_TPCTRLSTAT[HTIE] is set to 1, PMC sets the HTIF flag in the same register and generates an interrupt when the HTDS status changes. Writing 1 to HTIF can clear this HTIF flag.

When PMC_TPTM[TRMTPEN] is set, the HTDS flag assert/de-assert threshold is defined by the TOT[3:0] bits in the same register. For the detailed HTDS flag assert/de-assert thresholds, see the reference in TOT[3:0].

Before configuring the TOT[3:0] bits, user should set PMC_CTRL[GWREN] to unlock the write protection.

13.8.7 Low-power RC oscillator

PMC integrates a low-power RC oscillator (LPO) which provides a typical 20 kHz output. This LPO can serve as an independent clock source for the MCU on-chip modules such as watchdog and RTC.

This RC oscillator is set to ON by default in the FPM mode, and can be controlled to OFF in the RPM mode by configuring PMC_CTRL[RC20KENSTP].

The RC oscillator out frequency can be configured by user through the OSCOT[5:0] bits in PMC_RC20KTRM register. Before writing to OSCOT[5:0], user should set PMC_CTRL[GWREN].

13.9 Application information

1. VREFH readiness

   VREFH is a high-accuracy voltage reference. It needs 3 ms to be stable after PMC_STAT[VREFRDY] is asserted. When entering the Stop mode, VREFH is disabled automatically. So after exiting from the Stop mode, it requires to wait enough settling time. After the PMC_VREFHCFG setting is changed, it takes some time to get it settled. ADC or other functions using VREFH cannot work correctly during this transition period.

2. 20 kHz LPO calibration

LPO has to be calibrated after the PMC powers up, in order to get the ±5% precision. The LPO clock is connected to SBAR (in the SIM module), and the calibration can be achieved by using FTM1 with on-chip clock. Refer to the SIM chapter for more detailed setting information.

3. Special write enable register handling

   PMC_TPTM, PMC_RC20KTRM and PMC_VREFHCFG are protected by a special write enable register handling. They cannot be written unless PMC_CTL[GWREN] is 1.

Chapter 14
Miscellaneous Control Module (MCM)

14.1 Introduction

The Miscellaneous Control Module (MCM) provides a myriad of miscellaneous control functions.

14.1.1 Features

The MCM includes the following features:

- Program-visible information on the platform configuration
- Flash controller speculation buffer configurations

14.2 Memory map/register descriptions

The memory map and register descriptions found here describe the registers using byte addresses. The registers can be written only when in supervisor mode.

MCM memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
F000_3008	Crossbar Switch (AXBS) Slave Configuration (MCM_PLASC)	16	R	0007h	14.2.1/154
F000_300A	Crossbar Switch (AXBS) Master Configuration (MCM_PLAMC)	16	R	0009h	14.2.2/154
F000_300C	Platform Control Register (MCM_PLACR)	32	R/W	0000_0050h	14.2.3/155

14.2.1 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: F000_3000h base + 8h offset = F000_3008h

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	0	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.

14.2.2 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: F000_3000h base + Ah offset = F000_300Ah

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	0	1	0	0	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

14.2.3 Platform Control Register (MCM_PLACR)

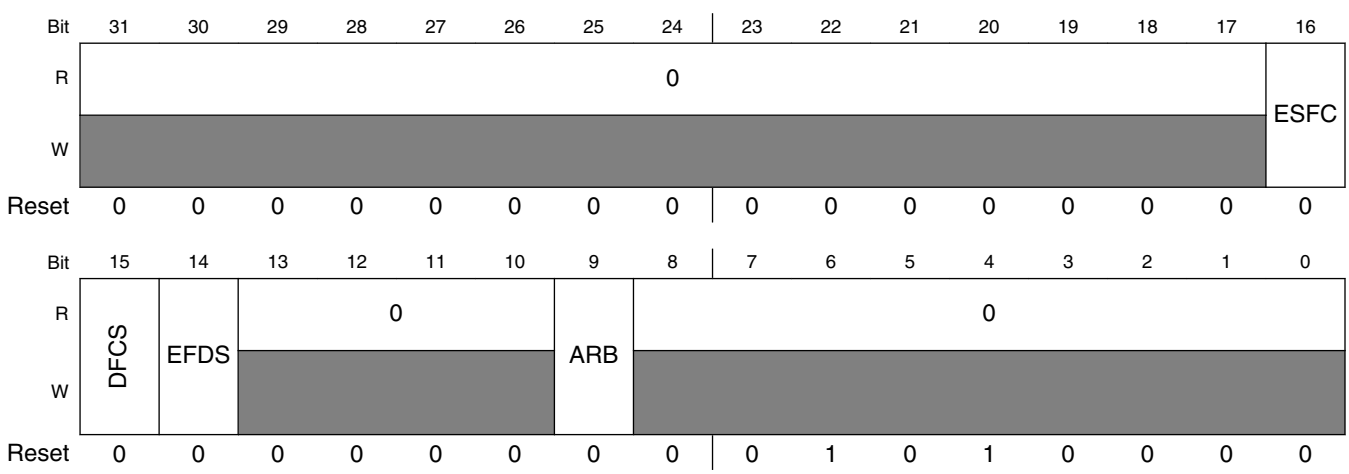
The PLACR register selects the arbitration policy for the crossbar masters and configures the flash memory controller.

The speculation buffer in the flash memory controller is configurable via PLACR[15: 14].

The speculation buffer is enabled only for instructions after reset. It is possible to have these states for the speculation buffer:

DFCS	EFDS	Description
0	0	Speculation buffer is on for instruction and off for data.
0	1	Speculation buffer is on for instruction and on for data.
1	X	Speculation buffer is off.

Address: F000_3000h base + Ch offset = F000_300Ch



MCM_PLACR field descriptions

Field	Description
31–17 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

Table continues on the next page...

MCM_PLACR field descriptions (continued)

Field	Description
16 ESFC	Enable Stalling Flash Controller Enables stalling flash controller when flash is busy. When software needs to access the flash memory while a flash memory resource is being manipulated by a flash command, software can enable a stall mechanism to avoid a read collision. The stall mechanism allows software to execute code from the same block on which flash operations are being performed. However, software must ensure the sector the flash operations are being performed on is not the same sector from which the code is executing. ESFC enables the stall mechanism. This bit must be set only just before the flash operation is executed and must be cleared when the operation completes. 0 Disable stalling flash controller when flash is busy. 1 Enable stalling flash controller when flash is busy.
15 DFCS	Disable Flash Controller Speculation Disables flash controller speculation. 0 Enable flash controller speculation. 1 Disable flash controller speculation.
14 EFDS	Enable Flash Data Speculation Enables flash data speculation. 0 Disable flash data speculation. 1 Enable flash data speculation.
13–10 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. This read-only field is reserved and always has the value 0.

Chapter 15
Peripheral Bridge (AIPS-Lite)

15.1 Introduction

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.

15.1.1 Features

Key features of the peripheral bridge are:

- Supports peripheral slots with 8-, 16-, and 32-bit datapath width

15.1.2 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 peripheral bridge performs a bus protocol conversion of the master transactions and generates the following as inputs to the peripherals:

- Module enables
- Module addresses
- Transfer attributes
- Byte enables
- Write data

The peripheral bridge selects and captures read data from the peripheral interface and returns it to the crossbar switch.

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.

The AIPS-Lite module uses the data width of accessed peripheral to perform proper data byte lane routing; bus decomposition (bus sizing) is performed when the access size is larger than the peripheral's data width.

15.2 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.

15.2.1 Access support

All combinations of access size and peripheral data port width are supported. An access that is larger than the target peripheral's data width will be decomposed to multiple, smaller accesses. Bus decomposition is terminated by a transfer error caused by an access to an empty register area.

Chapter 16
Crossbar Switch Lite (AXBS-Lite)

16.1 Chip-specific AXBS-Lite information

16.1.1 Crossbar-Light Switch Master Assignments

The masters connected to the crossbar switch are assigned as follows:

Master module	Master port number
ARM core unified bus	0
ADC	3

16.1.2 Crossbar Switch Slave Assignments

This device contains 3 slaves connected to the crossbar switch.

The slave assignment is as follows:

Slave module	Slave port number
Flash memory controller	0
SRAM controller	1
Peripheral bridge	2

16.2 Introduction

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.

16.2.1 Features

The crossbar switch includes these features:

- Symmetric crossbar bus switch implementation
  - Allows concurrent accesses from different masters to different slaves
- 32-bit data bus
- Operation at a 1-to-1 clock frequency with the bus masters
- Programmable configuration for fixed-priority or round-robin slave port arbitration

16.3 Memory Map / Register Definition

This crossbar switch is designed for minimal gate count. It, therefore, has no memory-mapped configuration registers.

16.4 Functional Description

16.4.1 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.

A master is given control of the targeted slave port only after a previous access to a different slave port completes, regardless of its priority on the newly targeted slave port. This prevents deadlock from occurring when:

- A higher priority master has:
  - An outstanding request to one slave port that has a long response time and
  - A pending access to a different slave port, and
- A lower priority master is also making a request to the same slave port as the pending access of the higher priority master.

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 master had to arbitrate to gain control of the slave port.

16.4.2 Arbitration

The crossbar switch supports two arbitration algorithms:

- Fixed priority
- Round-robin

The selection of the global slave port arbitration is controlled by MCM_PLACR[ARB]. For fixed priority, set MCM_PLACR[ARB] to 0. For round robin, set MCM_PLACR[ARB] to 1. This arbitration setting applies to all slave ports.

16.4.2.1 Arbitration during undefined length bursts

All lengths of burst accesses lock out arbitration until the last beat of the burst.

16.4.2.2 Fixed-priority operation

When operating in fixed-priority mode, each master is assigned a unique priority level with the highest numbered master having the highest priority (for example, in a system with 5 masters, master 1 has lower priority than master 3). If two masters request access to the same slave port, the master with the highest priority gains control over the slave port.

NOTE

In this arbitration mode, a higher-priority master can monopolize a slave port, preventing accesses from any lower-priority master to the port.

When a master makes a request to a slave port, the slave port checks whether the new requesting master's priority level is higher than that of the master that currently has control over the slave port, unless the slave port is in a parked state. The slave port performs an arbitration check at every clock edge to ensure that the proper master, if any, has control of the slave port.

The following table describes possible scenarios based on the requesting master port:

Table 16-1. How the Crossbar Switch grants control of a slave port to a master

When	Then the Crossbar Switch grants control to the requesting master
Both of the following are true: • The current master is not running a transfer. • The new requesting master's priority level is higher than that of the current master.	At the next clock edge
The requesting master's priority level is lower than the current master.	At the conclusion of one of the following cycles: • An IDLE cycle • A non-IDLE cycle to a location other than the current slave port

16.4.2.3 Round-robin priority operation

When operating in round-robin mode, each master is assigned a relative priority based on the master port number. This relative priority is compared to the master port number (ID) of the last master to perform a transfer on the slave bus. The highest priority requesting master becomes owner of the slave bus at the next transfer boundary. Priority is based on how far ahead the ID of the requesting master is to the ID of the last master.

After granted access to a slave port, a master may perform as many transfers as desired to that port until another master makes a request to the same slave port. The next master in line is granted access to the slave port at the next transfer boundary, or possibly on the next clock cycle if the current master has no pending access request.

As an example of arbitration in round-robin mode, assume the crossbar is implemented with master ports 0, 1, 4, and 5. If the last master of the slave port was master 1, and master 0, 4, and 5 make simultaneous requests, they are serviced in the order: 4 then 5 then 0.

The round-robin arbitration mode generally provides a more fair allocation of the available slave-port bandwidth (compared to fixed priority) as the fixed master priority does not affect the master selection.

16.5 Initialization/application information

No initialization is required for the crossbar switch.

See the AXBS section of the configuration chapter for the reset state of the arbitration scheme.

Chapter 17
Watchdog timer (WDOG)

17.1 Chip-specific WDOG information

17.1.1 WDOG clocks

The watchdog has four selectable clock sources:

- 20 kHz internal low power oscillator (LPOCLK)
- Internal 32 kHz reference clock (ICSIRCLK)
- External clock (OSCERCLK)
- Bus clock

17.2 Introduction

The Watchdog Timer (WDOG) module is an independent timer that is available for system use. It provides a safety feature to ensure that software is executing as planned and that the CPU is not stuck in an infinite loop or executing unintended code. If the WDOG module is not serviced (refreshed) within a certain period, it resets the MCU.

17.2.1 Features

Features of the WDOG module include:

- Configurable clock source inputs independent from the:
    - bus clock
    - Internal 32 kHz RC oscillator
    - Internal 20 kHz RC oscillator
    - External clock source

- Programmable timeout period
  - Programmable 16-bit timeout value
  - Optional fixed 256 clock prescaler when longer timeout periods are needed
- Robust write sequence for counter refresh
  - Refresh sequence of writing 0x02A6 and then 0x80B4 within 16 bus clocks
- Window mode option for the refresh mechanism
  - Programmable 16-bit window value
  - Provides robust check that program flow is faster than expected
  - Early refresh attempts trigger a reset.
- Optional timeout interrupt to allow post-processing diagnostics
  - Interrupt request to CPU with interrupt vector for an interrupt service routine (ISR)
  - Forced reset occurs 128 bus clocks after the interrupt vector fetch.
- Configuration bits are write-once-after-reset to ensure watchdog configuration cannot be mistakenly altered.
- Robust write sequence for unlocking write-once configuration bits
  - Unlock sequence of writing 0x20C5 and then 0x28D9 within 16 bus clocks for allowing updates to write-once configuration bits
  - Software must make updates within 128 bus clocks after unlocking and before WDOG closing unlock window.

17.2.2 Block diagram

The following figure provides a block diagram of the WDOG module.

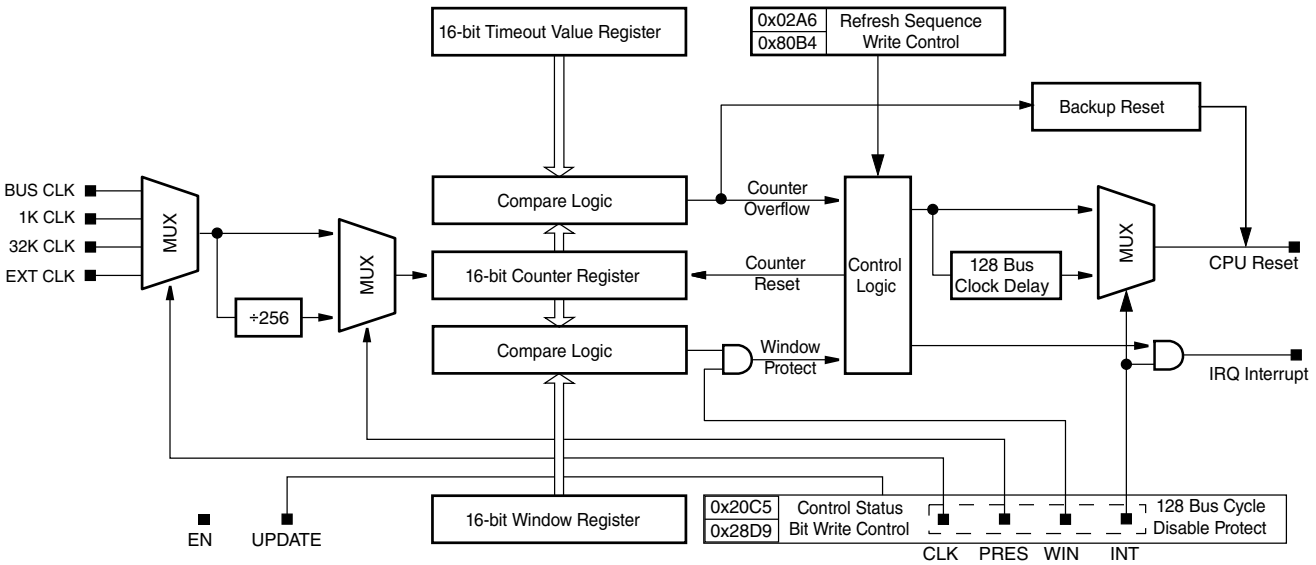


Figure 17-1. WDOG block diagram

17.3 Memory map and register definition

NOTE

If the device uses half-word to access WDOG_CNT, WDOG_TOVAL and WDOG_WIN, the transposed 16-bit bytes must follow the format of LowByte:HighByte. So 8-bit R/W is preferred.

WDOG memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4005_2000	Watchdog Control and Status Register 1 (WDOG_CS1)	8	R/W	80h	17.3.1/167
4005_2001	Watchdog Control and Status Register 2 (WDOG_CS2)	8	R/W	01h	17.3.2/169
4005_2002	Watchdog Counter Register: High (WDOG_CNTH)	8	R	00h	17.3.3/170
4005_2003	Watchdog Counter Register: Low (WDOG_CNTL)	8	R	00h	17.3.4/171
4005_2004	Watchdog Timeout Value Register: High (WDOG_TOVALH)	8	R/W	00h	17.3.5/171
4005_2005	Watchdog Timeout Value Register: Low (WDOG_TOVALL)	8	R/W	FFh	17.3.6/171
4005_2006	Watchdog Window Register: High (WDOG_WINH)	8	R/W	00h	17.3.7/172
4005_2007	Watchdog Window Register: Low (WDOG_WINL)	8	R/W	00h	17.3.8/172

17.3.1 Watchdog Control and Status Register 1 (WDOG_CS1)

This section describes the function of Watchdog Control and Status Register 1.

NOTE

TST is cleared (0:0) on POR only. Any other reset does not affect the value of this field.

Address: 4005_2000h base + 0h offset = 4005_2000h

Bit	7	6	5	4	3	2	1	0
Read Write	EN	INT	UPDATE	TST		DBG	WAIT	STOP
Reset	1	0	0	0	0	0	0	0

WDOG_CS1 field descriptions

Field	Description
7 EN	Watchdog Enable This write-once bit enables the watchdog counter to start counting. 0 Watchdog disabled. 1 Watchdog enabled.
6 INT	Watchdog Interrupt This write-once bit configures the watchdog to generate an interrupt request upon a reset-triggering event (timeout or illegal write to the watchdog), prior to forcing a reset. After the interrupt vector fetch, the reset occurs after a delay of 128 bus clocks. 0 Watchdog interrupts are disabled. Watchdog resets are not delayed. 1 Watchdog interrupts are enabled. Watchdog resets are delayed by 128 bus clocks.
5 UPDATE	Allow updates This write-once bit allows software to reconfigure the watchdog without a reset. 0 Updates not allowed. After the initial configuration, the watchdog cannot be later modified without forcing a reset. 1 Updates allowed. Software can modify the watchdog configuration registers within 128 bus clocks after performing the unlock write sequence.
4–3 TST	Watchdog Test Enables the fast test mode. The test mode allows software to exercise all bits of the counter to demonstrate that the watchdog is functioning properly. See the Fast testing of the watchdog section. This write-once field is cleared (0:0) on POR only. Any other reset does not affect the value of this field. 00 Watchdog test mode disabled. 01 Watchdog user mode enabled. (Watchdog test mode disabled.) After testing the watchdog, software should use this setting to indicate that the watchdog is functioning normally in user mode. 10 Watchdog test mode enabled, only the low byte is used. WDOG_CNTL is compared with WDOG_TOVALL. 11 Watchdog test mode enabled, only the high byte is used. WDOG_CNTH is compared with WDOG_TOVALH.
2 DBG	Debug Enable This write-once bit enables the watchdog to operate when the chip is in debug mode. 0 Watchdog disabled in chip debug mode. 1 Watchdog enabled in chip debug mode.

Table continues on the next page...

WDOG_CS1 field descriptions (continued)

Field	Description
1 WAIT	Wait Enable This write-once bit enables the watchdog to operate when the chip is in wait mode. 0 Watchdog disabled in chip wait mode. 1 Watchdog enabled in chip wait mode.
0 STOP	Stop Enable This write-once bit enables the watchdog to operate when the chip is in stop mode. 0 Watchdog disabled in chip stop mode. 1 Watchdog enabled in chip stop mode.

17.3.2 Watchdog Control and Status Register 2 (WDOG_CS2)

This section describes the function of the watchdog control and status register 2.

Address: 4005_2000h base + 1h offset = 4005_2001h

Bit	7	6	5	4	3	2	1	0
Read	WIN	FLG	0	PRES	0		CLK	
Write		w1c						
Reset	0	0	0	0	0	0	0	1

WDOG_CS2 field descriptions

Field	Description
7 WIN	Watchdog Window This write-once bit enables window mode. See the Window mode section. 0 Window mode disabled. 1 Window mode enabled.
6 FLG	Watchdog Interrupt Flag This bit is an interrupt indicator when INT is set in control and status register 1. Write 1 to clear it. 0 No interrupt occurred. 1 An interrupt occurred.
5 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
4 PRES	Watchdog Prescalar This write-once bit enables a fixed 256 pre-scaling of watchdog counter reference clock. (The block diagram shows this clock divider option.) 0 256 prescalar disabled. 1 256 prescalar enabled.

Table continues on the next page...

WDOG_CS2 field descriptions (continued)

Field	Description
3–2 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
CLK	Watchdog Clock This write-once field indicates the clock source that feeds the watchdog counter. See the Clock source section. 00 Bus clock. 01 20 kHz internal low-power oscillator (LPOCLK). 10 32 kHz internal oscillator (ICSIRCLK). 11 External clock source.

17.3.3 Watchdog Counter Register: High (WDOG_CNTH)

This section describes the watchdog counter registers: high (CNTH) and low (CNTL) combined.

The watchdog counter registers CNTH and CNTL provide access to the value of the free-running watchdog counter. Software can read the counter registers at any time.

Software cannot write directly to the watchdog counter; however, two write sequences to these registers have special functions:

1. The *refresh sequence* resets the watchdog counter to 0x0000. See the Refreshing the Watchdog section.
2. The *unlock sequence* allows the watchdog to be reconfigured without forcing a reset (when WDOG_CS1[UPDATE] = 1). See the Example code: Reconfiguring the Watchdog section.

NOTE

All other writes to these registers are illegal and force a reset.

Address: 4005_2000h base + 2h offset = 4005_2002h

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

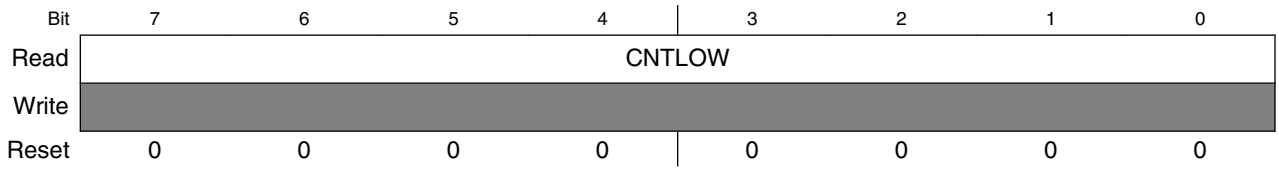
WDOG_CNTH field descriptions

Field	Description
CNTHIGH	High byte of the Watchdog Counter

17.3.4 Watchdog Counter Register: Low (WDOG_CNTL)

See the description of the WDOG_CNTH register.

Address: 4005_2000h base + 3h offset = 4005_2003h



WDOG_CNTL field descriptions

Field	Description
CNTLOW	Low byte of the Watchdog Counter

17.3.5 Watchdog Timeout Value Register: High (WDOG_TOVALH)

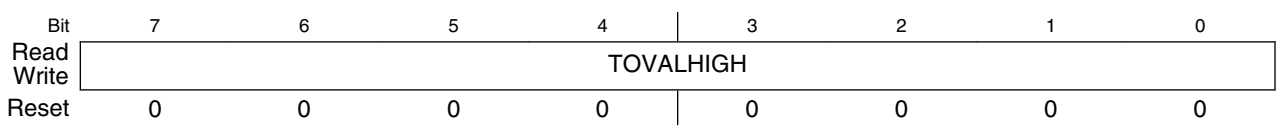
This section describes the watchdog timeout value registers: high (WDOG_TOVALH) and low (WDOG_TOVALL) combined. WDOG_TOVALH and WDOG_TOVALL contains the 16-bit value used to set the timeout period of the watchdog.

The watchdog counter (WDOG_CNTH and WDOG_CNTL) is continuously compared with the timeout value (WDOG_TOVALH and WDOG_TOVALL). If the counter reaches the timeout value, the watchdog forces a reset.

NOTE

Do not write 0 to the Watchdog Timeout Value Register, otherwise, the watchdog always generates a reset.

Address: 4005_2000h base + 4h offset = 4005_2004h



WDOG_TOVALH field descriptions

Field	Description
TOVALHIGH	High byte of the timeout value

17.3.6 Watchdog Timeout Value Register: Low (WDOG_TOVALL)

See the description of the WDOG_TOVALH register.

NOTE

All the bits reset to 0 in read.

Address: 4005_2000h base + 5h offset = 4005_2005h

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

WDOG_TOVALL field descriptions

Field	Description
TOVALLOW	Low byte of the timeout value

17.3.7 Watchdog Window Register: High (WDOG_WINH)

This section describes the watchdog window registers: high (WDOG_WINH) and low (WDOG_WINL) combined. When window mode is enabled (WDOG_CS2[WIN] is set), WDOG_WINH and WDOG_WINL determine the earliest time that a refresh sequence is considered valid. See the Watchdog refresh mechanism section.

WDOG_WINH and WDOG_WINL must be less than WDOG_TOVALH and WDOG_TOVALL.

Address: 4005_2000h base + 6h offset = 4005_2006h

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

WDOG_WINH field descriptions

Field	Description
WINHIGH	High byte of Watchdog Window

17.3.8 Watchdog Window Register: Low (WDOG_WINL)

See the description of the WDOG_WINH register.

Address: 4005_2000h base + 7h offset = 4005_2007h

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

WDOG_WINL field descriptions

Field	Description
WINLOW	Low byte of Watchdog Window

17.4 Functional description

The WDOG module provides a fail safe mechanism to ensure the system can be reset to a known state of operation in case of system failure, such as the CPU clock stopping or there being a run away condition in the software code. The watchdog counter runs continuously off a selectable clock source and expects to be serviced (refreshed) periodically. If it is not, it resets the system.

The timeout period, window mode, and clock source are all programmable but must be configured within 128 bus clocks after a reset.

17.4.1 Watchdog refresh mechanism

The watchdog resets the MCU if the watchdog counter is not refreshed. A robust refresh mechanism makes it very unlikely that the watchdog can be refreshed by runaway code.

To refresh the watchdog counter, software must execute a refresh write sequence before the timeout period expires. In addition, if window mode is used, software must not start the refresh sequence until after the time value set in the WDOG_WINH and WDOG_WINL registers. See the following figure.

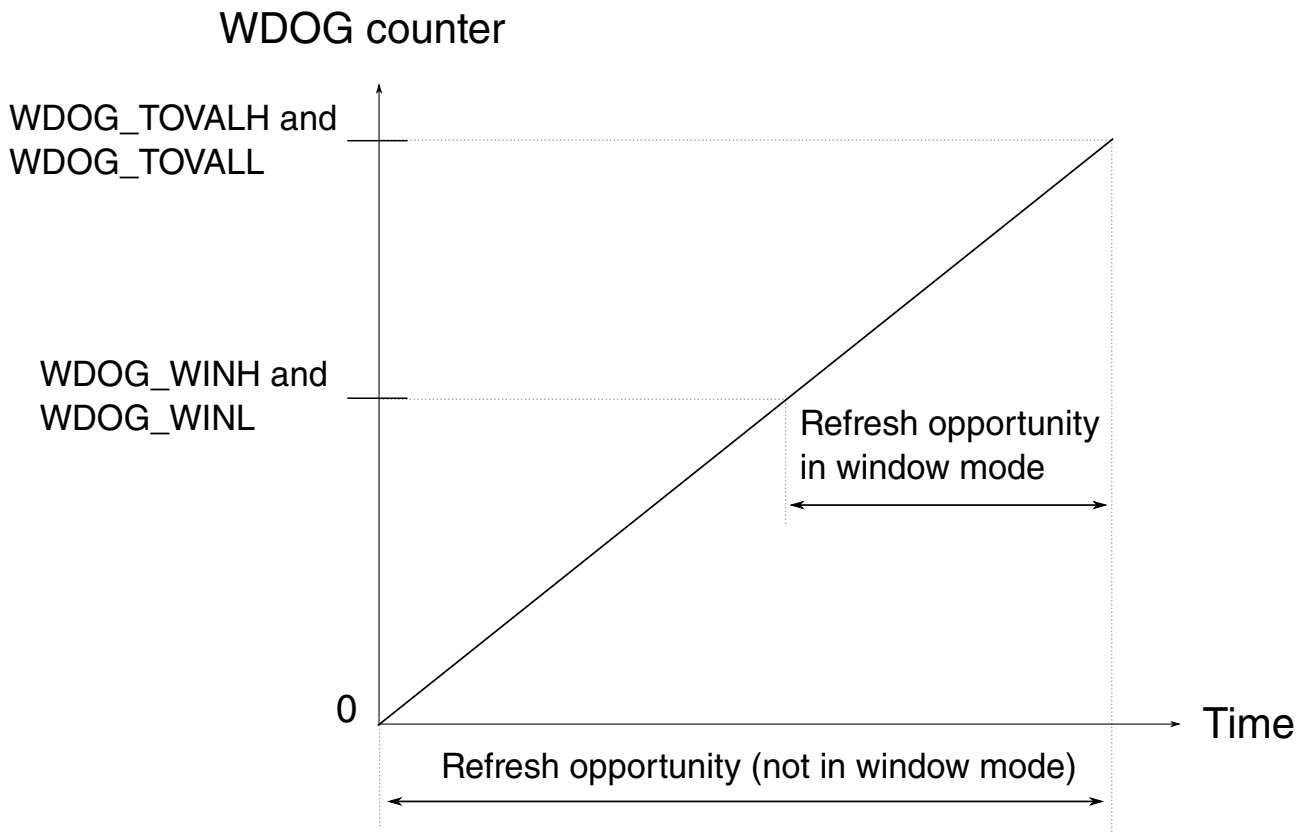


Figure 17-10. Refresh opportunity for the Watchdog counter

17.4.1.1 Window mode

Software finishing its main control loop faster than expected could be an indication of a problem. Depending on the requirements of the application, the WDOG can be programmed to force a reset when refresh attempts are early.

When Window mode is enabled, the watchdog must be refreshed after the counter has reached a minimum expected time value; otherwise, the watchdog resets the MCU. The minimum expected time value is specified in the WDOG_WINH:L registers. Setting CS1[WIN] enables Window mode.

17.4.1.2 Refreshing the Watchdog

The refresh write sequence is a write of 0x02A6 followed by a write of 0x80B4 to the WDOG_CNTH and WDOG_CNTL registers. The write of the 0x80B4 must occur within 16 bus clocks after the write of 0x02A6; otherwise, the watchdog resets the MCU.

Note

Before starting the refresh sequence, disable global interrupts. Otherwise, an interrupt could effectively invalidate the refresh sequence if writing the four bytes takes more than 16 bus clocks. Re-enable interrupts when the sequence is finished.

17.4.1.3 Example code: Refreshing the Watchdog

The following code segment shows the refresh write sequence of the WDOG module.

```
/* Refresh watchdog */

for (;;) // main loop
{
   ...

   DisableInterrupts; // disable global interrupt

   WDOG_CNT = 0x02A6; // write the 1st refresh word

   WDOG_CNT = 0x80B4; // write the 2nd refresh word to refresh counter

   EnableInterrupts; // enable global interrupt

   ...
}
```

17.4.2 Configuring the Watchdog

All watchdog control bits, timeout value, and window value are write-once after reset. This means that after a write has occurred they cannot be changed unless a reset occurs. This provides a robust mechanism to configure the watchdog and ensure that a runaway condition cannot mistakenly disable or modify the watchdog configuration after configured.

This is guaranteed by the user configuring the window and timeout value first, followed by the other control bits, and ensuring that CS1[UPDATE] is also set to 0. The new configuration takes effect only after all registers except WDOG_CNTH:L are written once after reset. Otherwise, the WDOG uses the reset values by default. If window mode is not used (CS2[WIN] is 0), writing to WDOG_WINH:L is not required to make the new configuration take effect.

17.4.2.1 Reconfiguring the Watchdog

In some cases (such as when supporting a bootloader function), users may want to reconfigure or disable the watchdog without forcing a reset first. By setting CS1[UPDATE] to a 1 on the initial configuration of the watchdog after a reset, users can reconfigure the watchdog at any time by executing an unlock sequence. (Conversely, if CS1[UPDATE] remains 0, the only way to reconfigure the watchdog is by initiating a reset.) The unlock sequence is similar to the refresh sequence but uses different values.

17.4.2.2 Unlocking the Watchdog

The unlock sequence is a write to the WDOG_CNTH:L registers of 0x20C5 followed by 0x28D9 within 16 bus clocks at any time after the watchdog has been configured. On completing the unlock sequence, the user must reconfigure the watchdog within 128 bus clocks; .

NOTE

Due to 128 bus clocks requirement for reconfiguring the watchdog, some delays must be inserted before executing STOP or WAIT instructions after reconfiguring the watchdog. This ensures that the watchdog's new configuration takes effect before MCU enters low power mode. Otherwise, the MCU may not be waken up from low power mode.

17.4.2.3 Example code: Reconfiguring the Watchdog

The following code segment shows an example reconfiguration of the WDOG module.

```
/* Initialize watchdog with ~20kHz clock source */

DisableInterrupts; // disable global interrupt

WDOG_CNT = 0x20C5; // write the 1st unlock word

WDOG_CNT = 0x28D9; // write the 2nd unlock word

WDOG_TOVAL = 1000; // setting timeout value

WDOG_CS2 = WDOG_CS2_CLK_MASK; // setting 20-kHz clock source

WDOG_CS1 = WDOG_CS1_EN_MASK; // enable counter running

EnableInterrupts; // enable global interrupt
```

17.4.3 Clock source

The watchdog counter has four clock source options selected by programming CS2[CLK]:

- bus clock
- internal Low-Power Oscillator (LPO) running at approximately 20 kHz (This is the default source.)
- internal 32 kHz clock
- external clock

The options allow software to select a clock source independent of the bus clock for applications that need to meet more robust safety requirements. Using a clock source other than the bus clock ensures that the watchdog counter continues to run if the bus clock is somehow halted; see Backup reset.

An optional fixed prescaler for all clock sources allows for longer timeout periods. When CS2[PRES] is set, the clock source is prescaled by 256 before clocking the watchdog counter.

The following table summarizes the different watchdog timeout periods available.

Table 17-10. Watchdog timeout availability

Reference clock	Prescaler	Watchdog time-out availability
Internal ~20 kHz (LPO)	Pass through	~1 ms–65.5 s[1]
	÷256	~256 ms–16,777 s
Internal ~32 kHz	Pass through	~31.25 µs–2.048 s
	÷256	~8 ms–524.3 s
1 MHz (from bus or external)	Pass through	1 µs–65.54 ms
	÷256	256 µs–16.777 s
20 MHz (from bus or external)	Pass through	50 ns–3.277 ms
	÷256	12.8 µs–838.8 ms

1. The default timeout value after reset is approximately 4 ms.

NOTE

When the programmer switches clock sources during reconfiguration, the watchdog hardware holds the counter at zero for 2.5 periods of the previous clock source and 2.5 periods of the new clock source after the configuration time period (128 bus clocks) ends. This delay ensures a smooth transition before restarting the counter with the new configuration.

17.4.4 Using interrupts to delay resets

When interrupts are enabled (CS1[INT] = 1), the watchdog first generates an interrupt request upon a reset triggering event (such as a counter timeout or invalid refresh attempt). The watchdog delays forcing a reset for 128 bus clocks to allow the interrupt service routine (ISR) to perform tasks, such as analyzing the stack to debug code.

When interrupts are disabled (CS1[INT] = 0), the watchdog does not delay forcing a reset.

17.4.5 Backup reset

NOTE

A clock source other than the bus clock must be used as the reference clock for the counter; otherwise, the backup reset function is not available.

The backup reset function is a safeguard feature that independently generates a reset in case the main WDOG logic loses its clock (the bus clock) and can no longer monitor the counter. If the watchdog counter overflows twice in succession (without an intervening reset), the backup reset function takes effect and generates a reset.

17.4.6 Functionality in debug and low-power modes

By default, the watchdog is not functional in Active Background mode, Wait mode, or Stop mode. However, the watchdog can remain functional in these modes as follows:

- For Active Background mode, set CS1[DBG]. (This way the watchdog is functional in Active Background mode even when the CPU is held by the Debug module.)
- For Wait mode, set CS1[WAIT].
- For Stop mode, set CS1[STOP].

NOTE

The watchdog can not generate interrupt in Stop mode even if CS1[STOP] is set and will not wake the MCU from Stop mode. It can generate reset during Stop mode.

For Active Background mode and Stop mode, in addition to the above configurations, a clock source other than the bus clock must be used as the reference clock for the counter; otherwise, the watchdog cannot function.

17.4.7 Fast testing of the watchdog

Before executing application code in safety critical applications, users are required to test that the watchdog works as expected and resets the MCU. Testing every bit of a 16-bit counter by letting it run to the overflow value takes a relatively long time (64 kHz clocks).

To help minimize the startup delay for application code after reset, the watchdog has a feature to test the watchdog more quickly by splitting the counter into its constituent byte-wide stages. The low and high bytes are run independently and tested for timeout against the corresponding byte of the timeout value register. (For complete coverage when testing the high byte of the counter, the test feature feeds the input clock via the 8th bit of the low byte, thus ensuring that the overflow connection from the low byte to the high byte is tested.)

Using this test feature reduces the test time to 512 clocks (not including overhead, such as user configuration and reset vector fetches). To further speed testing, use a faster clock (such as the bus clock) for the counter reference.

On a power-on reset, the POR bit in the system reset register is set, indicating the user should perform the WDOG fast test.

17.4.7.1 Testing each byte of the counter

The test procedure follows these steps:

1. Program the preferred watchdog timeout value in the WDOG_TOVALH and WDOG_TOVALL registers during the watchdog configuration time period.
2. Select a byte of the counter to test using the WDOG_CS1[TST] = 10b for the low byte; WDOG_CS1[TST] = 11b for the high byte.
3. Wait for the watchdog to timeout. Optionally, in the idle loop, increment RAM locations as a parallel software counter for later comparison. Because the RAM is not affected by a watchdog reset, the timeout period of the watchdog counter can be compared with the software counter to verify the timeout period has occurred as expected.
4. The watchdog counter times out and forces a reset.

5. Confirm the WDOG flag in the system reset register is set, indicating that the watchdog caused the reset. (The POR flag remains clear.)
6. Confirm that WDOG_CS1[TST] shows a test (10b or 11b) was performed.

If confirmed, the count and compare functions work for the selected byte. Repeat the procedure, selecting the other byte in step 2.

NOTE

WDOG_CS1[TST] is cleared by a POR only and not affected by other resets.

17.4.7.2 Entering user mode

After successfully testing the low and high bytes of the watchdog counter, the user can configure WDOG_CS1[TST] to 01b to indicate the watchdog is ready for use in application user mode. Thus if a reset occurs again, software can recognize the reset trigger as a real watchdog reset caused by runaway or faulty application code.

As an ongoing test when using the default 20 kHz clock source, software can periodically read the WDOG_CNTH and WDOG_CNTL registers to ensure the counter is being incremented.

Chapter 18
Flash Memory Module (FTMRE)

18.1 Introduction

The FTMRE module implements the following:

- Program flash (flash) memory

The flash memory is ideal for single-supply applications allowing for field reprogramming without requiring external high voltage sources for program or erase operations. The flash module includes a memory controller that executes commands to modify flash memory contents. The user interface to the memory controller consists of the indexed Flash Common Command Object (FCCOB) register which is written to with the command, global address, data, and any required command parameters. The memory controller must complete the execution of a command before the FCCOB register can be written to with a new command.

CAUTION

A flash byte or longword must be in the erased state before being programmed. Cumulative programming of bits within a flash byte or longword is not allowed.

The flash memory is read as longwords. Read access time is one bus cycle for longwords. For flash memory, an erased bit reads 1 and a programmed bit reads 0.

18.2 Feature

18.2.1 Flash memory features

The flash memory has the following features:

- 16 KB of flash memory composed of one 16 KB flash block divided into 32 sectors of 512 bytes
- Automated program and erase algorithm with verify

- Fast sector erase and longword program operation
- Flexible protection scheme to prevent accidental programming or erasing of flash memory

18.2.2 Other flash module features

The flash memory module has the following other features:

- No external high-voltage power supply required for flash memory program and erase operations
- Interrupt generation on flash command completion
- Security mechanism to prevent unauthorized access to the flash memory

18.3 Functional description

18.3.1 Modes of operation

The flash memory module provides the normal user mode of operation. The operating mode is determined by module-level inputs and affects the FPROT, FCNFG, and FCLKDIV registers.

18.3.1.1 Wait mode

The flash memory module is not affected if the MCU enters Wait mode. The flash module can recover the MCU from Wait via the CCIF interrupt. See Flash interrupts.

18.3.1.2 Stop mode

If a flash command is active, that is, FSTAT[CCIF] = 0, when the MCU requests Stop mode, the current NVM operation will be completed before the MCU is allowed to enter Stop mode.

18.3.2 Flash block read access

If a flash block is read during execution of a command (while FSTAT[CCIF] = 0), the read operation will return invalid data and it will trigger a illegal access exception in the MCU.

18.3.3 Flash memory map

The MCU places the flash memory as shown in the following table.

Table 18-1. Flash memory addressing

Global address	Flash size	Description
0x0000_0000–0x0000_3FFF	16 KB	Flash block contains flash configuration field.

18.3.4 Flash initialization after system reset

On each system reset, the flash module executes an initialization sequence that establishes initial values for the flash block configuration parameters, the FPROT protection register, and the FOPT and FSEC registers. The initialization routine reverts to built-in default values that leave the module in a fully protected and secured state if errors are encountered during execution of the reset sequence. If an error is detected during the reset sequence, both FSTAT[MGSTAT] bits will be set.

FSTAT[CCIF] is cleared throughout the initialization sequence. The NVM module holds off all CPU access for a portion of the initialization sequence. Flash reads are allowed after the hold is removed. Completion of the initialization sequence is marked by setting FSTAT[CCIF] high, which enables user commands. While FSTAT[CCIF] remains cleared, it is not possible to write on registers FCCOBIX or FCCOB.

If a reset occurs while any flash command is in progress, that command will be immediately aborted. The state of the word being programmed or the sector/block being erased is not guaranteed.

18.3.5 Flash command operations

Flash command operations are used to modify flash memory contents.

The command operations contain three steps:

1. Configure the clock for flash program and erase command operations.
2. Use command write sequence to set flash command parameters and launch execution.
3. Execute valid flash commands according to MCU functional mode and MCU security state.

The figure below shows a general flowchart of the flash command write sequence.

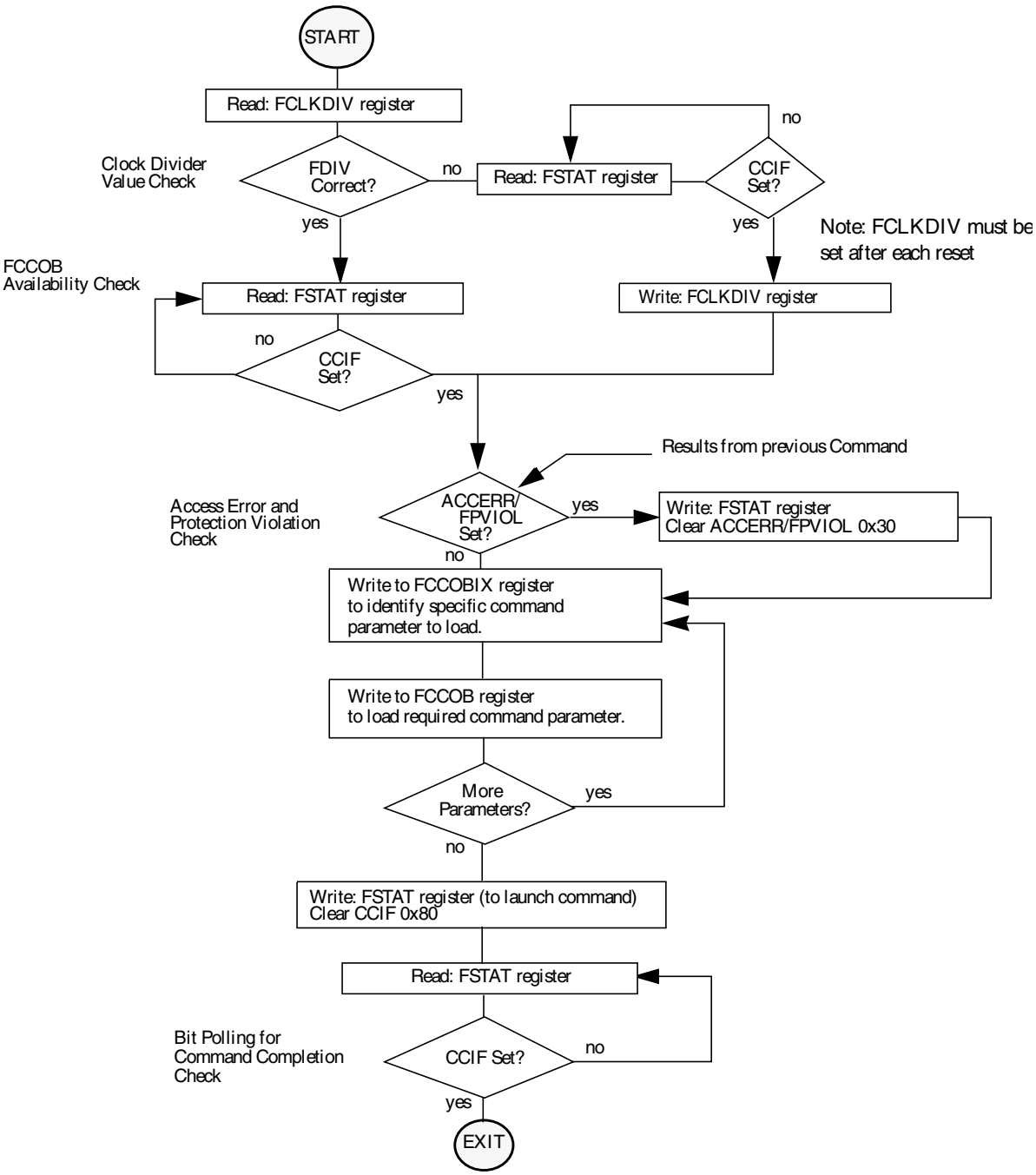


Figure 18-1. Generic flash command write sequence flowchart

18.3.5.1 Writing the FCLKDIV register

Prior to issuing any flash program or erase command after a reset, the user is required to write the FCLKDIV register to divide BUSCLK down to a target FCLK of 1 MHz. The following table shows recommended values for FCLKDIV[FDIV] based on BUSCLK frequency.

Table 18-2. FDIV values for various BUSCLK frequencies

BUSCLK frequency (MHz)		FDIV[5:0]
MIN[1]	MAX[2]	
1.0	1.6	0x00
1.6	2.6	0x01
2.6	3.6	0x02
3.6	4.6	0x03
4.6	5.6	0x04
5.6	6.6	0x05
6.6	7.6	0x06
7.6	8.6	0x07
8.6	9.6	0x08
9.6	10.6	0x09
10.6	11.6	0x0A
11.6	12.6	0x0B
12.6	13.6	0x0C
13.6	14.6	0x0D
14.6	15.6	0x0E
15.6	16.6	0x0F
16.6	17.6	0x10
17.6	18.6	0x11
18.6	19.6	0x12
19.6	20.6	0x13
20.6	21.6	0x14
21.6	22.6	0x15
22.6	23.6	0x16
23.6	24.6	0x17
24.6	25.6	0x18

1. BUSCLK is greater than this value.
2. BUSCLK is less than or equal to this value.

CAUTION

Programming or erasing the flash memory cannot be performed if the bus clock runs at less than 0.8 MHz. Setting FCLKDIV[FDIV] too high can destroy the flash memory due to overstress. Setting FCLKDIV[FDIV] too low can result in incomplete programming or erasure of the flash memory cells.

When the FCLKDIV register is written, FCLKDIV[FDIVLD] is set automatically. If FCLKDIV[FDIVLD] is 0, the FCLKDIV register has not been written since the last reset. If the FCLKDIV register has not been written, any flash program or erase command loaded during a command write sequence will not execute and FSTAT[ACCERR] will be set.

18.3.5.2 Command write sequence

The memory controller will launch all valid flash commands entered using a command write sequence.

Before launching a command, FSTAT[ACCERR] and FSTAT[FPVIOL] must be cleared and the FSTAT[CCIF] flag will be tested to determine the status of the current command write sequence. If FSTAT[CCIF] is 0, indicating that the previous command write sequence is still active, a new command write sequence cannot be started and all writes to the FCCOB register are ignored.

The FCCOB parameter fields must be loaded with all required parameters for the flash command being executed. Access to the FCCOB parameter fields is controlled via FCCOBIX[CCOBIX].

Flash command mode uses the indexed FCCOB register to provide a command code and its relevant parameters to the memory controller. First, the user must set up all required FCCOB fields. Then they can initiate the command's execution by writing a 1 to FSTAT[CCIF]. This action clears the CCIF command completion flag to 0. When the user clears FSTAT[CCIF], all FCCOB parameter fields are locked and cannot be changed by the user until the command completes (evidenced by the memory controller returning FSTAT[CCIF] to1). Some commands return information to the FCCOB register array.

The generic format for the FCCOB parameter fields in flash command mode is shown in the following table. The return values are available for reading after the FSTAT[CCIF] flag has been returned to 1 by the memory controller. Writes to the unimplemented parameter fields, FCCOBIX[CCOBIX] =110b and FCCOBIX[CCOBIX] = 111b, are ignored with read from these fields returning 0x0000.

Table 18-3 shows the generic flash command format. The high byte of the first word in the CCOB array contains the command code, followed by the parameters for this specific flash command. For details on the FCCOB settings required by each command, see the flash command descriptions in Flash command summary .

Table 18-3. FCCOB – flash command mode typical usage

CCOBIX[2:0]	Byte	FCCOB parameter fields in flash command mode
000	HI	FCMD[7:0] defining flash command
	LO	Global address [23:16]
001	HI	Global address [15:8]
	LO	Global address [7:0]
010	HI	Data 0 [15:8]
	LO	Data 0 [7:0]
011	HI	Data 1 [15:8]
	LO	Data 1 [7:0]
100	HI	Data 2 [15:8]
	LO	Data 2 [7:0]
101	HI	Data 3 [15:8]
	LO	Data 3 [7:0]

The contents of the FCCOB parameter fields are transferred to the memory controller when the user clears the FSTAT[CCIF] command completion flag by writing 1. The CCIF flag will remain clear until the flash command has completed. Upon completion, the memory controller will return FSTAT[CCIF] to 1 and the FCCOB register will be used to communicate any results.

The following table presents the valid flash commands, as enabled by the combination of the functional MCU mode with the MCU security state of unsecured or secured.

MCU secured state is selected by FSEC[SEC].

Table 18-4. Flash commands by mode and security state

FCMD	Command	Unsecured	Secured
		U[1]	U[2]
0x01	Erase verify all blocks	*	*
0x02	Erase verify block	*	*
0x03	Erase verify flash section	*	*
0x04	Read once	*	*
0x06	Program flash	*	*
0x07	Program once	*	*
0x08	Erase all block	*	*
0x09	Erase flash block	*	*

Table continues on the next page...

Table 18-4. Flash commands by mode and security state (continued)

FCMD	Command	Unsecured	Secured
		U[1]	U[2]
0x0A	Erase flash sector	*	*
0x0B	Unsecure flash	*	*
0x0C	Verify backdoor access key	*	*
0x0D	Set user margin level	*	*
0x0E	Set factory margin level	*	*
0x0F	Configure NVM	*	*

1. Unsecured user mode
2. Secured user mode

18.3.6 Flash interrupts

The flash module can generate an interrupt when a flash command operation has completed.

Table 18-5. Flash interrupt source

Interrupt source	Interrupt flag	Local enable	Global (CCR) mask
Flash command complete	CCIF (FSTAT register)	CCIE (FCNFG register)	I Bit

18.3.6.1 Description of flash interrupt operation

The flash module uses the FSTAT[CCIF] flag in combination with the FCNFG[CCIE] interrupt enable bit to generate the flash command interrupt request.

The logic used for generating the flash module interrupts is shown in the following figure.

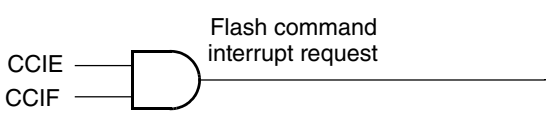


Figure 18-2. Flash module interrupts implementation

18.3.7 Protection

The FPROT register can be set to protect regions in the flash memory from accidental programing or erasing. Two separate memory regions, one growing upward from global address 0x0000 in the flash memory, called the lower region, and the remaining addresses in the flash memory, can be activated for protection. The flash memory addresses covered by these protectable regions are shown in the flash memory map.

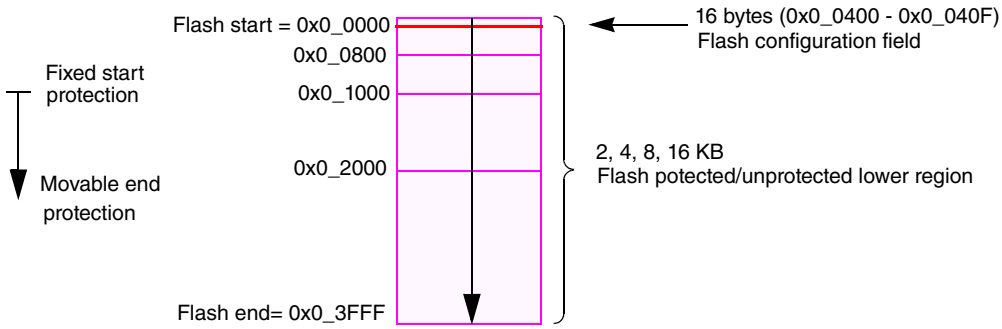


Figure 18-3. 16 KB flash protection memory map

Default protection settings as well as security information that allows the MCU to restrict access to the flash module are stored in the flash configuration field as described in the table below.

Table 18-6. Flash configuration field

Global address	Size (Bytes)	Description
0x0400–0x0407	8	Backdoor comparison key. See Verify backdoor access key command and Unsecuring the MCU using backdoor key access.
0x0408–0x040B [1]	4	Reserved
0x040C–0x040F[1]	1	Flash nonvolatile byte - data[31:24]
	1	Flash security byte - data[23:16]
	1	Flash protection byte - data[15:8]
	1	Reserved - data [7:0]

1. 0x0_0408–0x040B and 0x040C–0x0_040F form a flash longword in each address range and must be programmed in a single command write sequence. Each byte in these longwords that are marked as reserved must be programmed to 0xFF. Alternatively, the Flash phrase 0x0408-0x040F can also be programmed in a single command write sequence.

The flash module provides protection to the MCU. During the reset sequence, the FPROT register is loaded with the contents of the flash protection byte in the flash configuration field at global address 0x040D in flash memory. The protection functions depend on the configuration of bit settings in FPROT register.

The flash protection scheme can be used by applications requiring reprogramming in single chip mode while providing as much protection as possible if reprogramming is not required.

All possible flash protection scenarios are shown in Figure 18-4. Although the protection scheme is loaded from the flash memory at global address 0x040D during the reset sequence, it can be changed by the user.

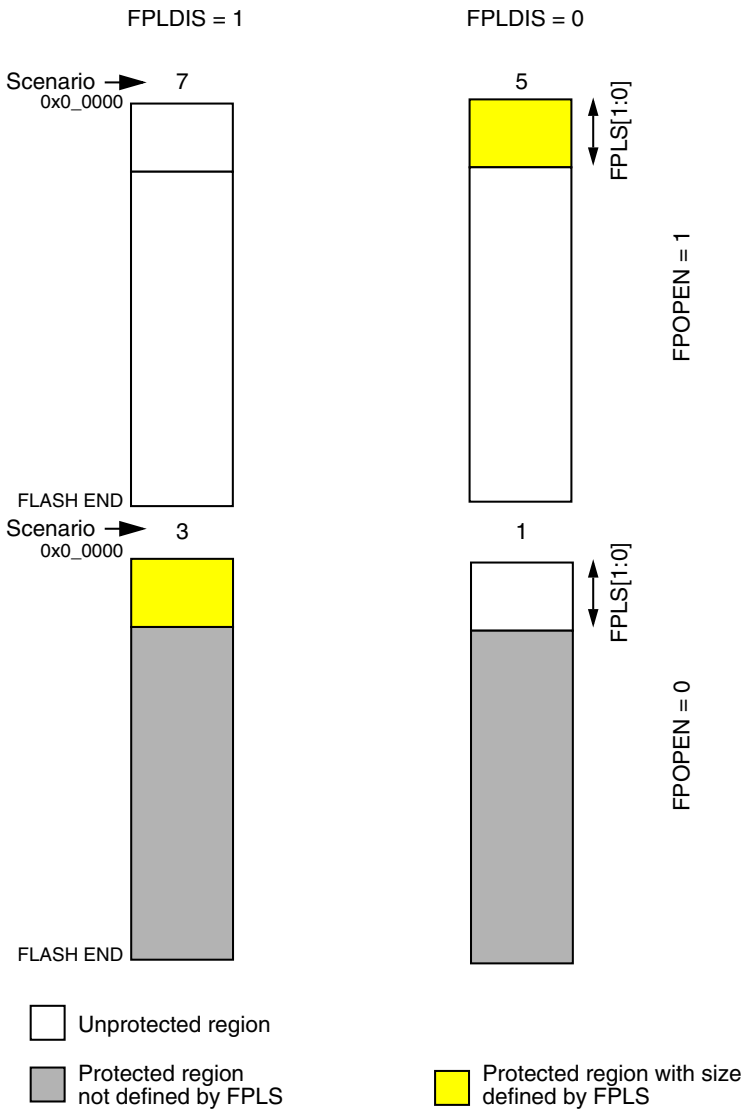


Figure 18-4. Flash protection scenarios

The general guideline is that flash protection can only be added and not removed. The following table specifies all valid transitions between flash protection scenarios. Any attempt to write an invalid scenario to the FPROT register will be ignored. The contents of the FPROT register reflect the active protection scenario. See the FPROT[FPLS] field descriptions for additional restrictions.

Table 18-7. Flash protection scenario transitions

From protection scenario	To protection scenario[1]			
	1	3	5	7
1	×	×		
3		×		
5		×	×	
7	×	×	×	×

1. Allowed transitions marked with X.

The flash protection address range is listed in the following two tables regarding the scenarios in the table above.

18.3.8 Security

The flash module provides security information to the MCU. The flash security state is defined by FSEC[SEC]. During reset, the flash module initializes the FSEC register using data read from the security byte of the flash configuration field. The security state out of reset can be permanently changed by programming the security byte, assuming that the MCU is starting from a mode where the necessary flash erase and program commands are available and that the upper region of the flash is unprotected. If the flash security byte is successfully programmed, its new value will take effect after the next MCU reset.

The following subsections describe these security-related subjects:

- Unsecuring the MCU using backdoor key access
- Unsecuring the MCU using SWD
- Mode and security effects on flash command availability

18.3.8.1 Unsecuring the MCU using backdoor key access

The MCU may be unsecured by using the backdoor key access feature which requires knowledge of the contents of the backdoor keys, which are four 16-bit words programmed at addresses 0x400–0x407. If the KEYEN[1:0] bits are in the enabled state, the verify backdoor access key command – see Verify backdoor access key command, allows the user to present four prospective keys for comparison to the keys stored in the flash memory via the memory controller. If the keys presented in the verify backdoor access key command match the backdoor keys stored in the flash memory, FSEC[SEC]

will be changed to unsecure the MCU. Key values of 0x0000 and 0xFFFF are not permitted as backdoor keys. While the Verify Backdoor Access Key command is active, flash memory will not be available for read access and will return invalid data.

The user code stored in the 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[1:0] bits are in the enabled state, the MCU can be unsecured by the backdoor key access sequence described below:

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 MCU is unsecured and FSEC[SEC] is forced to the unsecure state of 10.

The verify backdoor access key command is monitored by the memory controller and an illegal key will prohibit future use of the verify backdoor access key command. A reset of the MCU is the only method to re-enable the verify backdoor access key command. The security as defined in the flash security byte is not changed by using the verify backdoor access key command sequence. The backdoor keys stored in addresses 0x400–0x407 are unaffected by the verify backdoor access key command sequence. The verify backdoor access key command sequence has no effect on the program and erase protections defined in the flash protection register, FPROT.

After the backdoor keys have been correctly matched, the MCU will be unsecured. After the MCU is unsecured, the sector containing the flash security byte can be erased and the flash security byte can be reprogrammed to the unsecure state, if desired. In the unsecure state, the user has full control of the contents of the backdoor keys by programming addresses 0x400–0x407 in the flash configuration field.

18.3.8.2 Unsecuring the MCU using SWD

A secured MCU can be unsecured by using the following method to erase the flash memory:

1. Reset the device by asserting $\overline{\text{RESET}}$ pin or DAP_CTRL[3].
2. Set DAP_CTRL[0] bit to invoke debug mass erase via SWD
3. Release reset by deasserting $\overline{\text{RESET}}$ pin or DAP_CTRL[3] bit via SWD.
4. Wait till DAP_CTRL[0] bit is cleared (After mass erase completes, DAP_CTRL[0] bit is cleared automatically). At this time, CPU will be in hold state, MASS erase is

completed, and the device is in unsecure state (flash security byte in flash configuration field is programmed with 0xFE) .

5. Reset the device.

18.3.8.3 Mode and security effects on flash command availability

The availability of flash module commands depends on the MCU operating mode and security state as shown in Table 18-4.

18.3.9 Flash commands

18.3.9.1 Flash commands

The following table summarizes the valid flash commands as well as the effects of the commands on the flash block and other resources within the flash module.

NOTE

All commands in the following table, regardless of MCU mode or security state, cannot be launched while the flash array is being read (i.e. the commands must not be executed if the core is fetching code from the flash). If the core attempts to read the flash while any command is running that may result in an illegal access. Refer to Flash block read access for details.

Table 18-8. Flash commands

FCMD	Command	Function on flash memory
0x01	Erase Verify All Blocks	Verifies that all flash blocks are erased
0x02	Erase Verify Block	Verifies that a flash block is erased
0x03	Erase Verify Flash Section	Verifies that a given number of words starting at the address provided are erased
0x04	Read Once	Reads a dedicated 64-byte field in the nonvolatile information register in flash block that was previously programmed using the program once command
0x06	Program Flash	Programs up to two longwords in a flash block
0x07	Program Once	Programs a dedicated 64 byte field in the nonvolatile information register in flash block that is allowed to be programmed only once
0x08	Erase All Block	Erases all flash blocks An erase of all flash blocks is possible only when the FPROT[FPHDIS], FPROT[FPLDIS] and FPROT[FPOEN] and the bit are set prior to launching the command
0x09	Erase Flash Block	Erases a flash block

Table continues on the next page...

Table 18-8. Flash commands (continued)

FCMD	Command	Function on flash memory
		An erase of the full flash block is possible only when FPROT[FPLDIS], FPROT[FPHDIS], and FPROT[FPOEN] are set prior to launching the command.
0x0A	Erase Flash Sector	Erases all bytes in a flash sector
0x0B	Unsecure Flash	Supports a method of releasing MCU security by erasing all flash blocks and verifying that all flash blocks are erased
0x0C	Verify Backdoor Access key	Supports a method of releasing MCU security by verifying a set of security keys
0x0D	Set User Margin Level	Specifies a user margin read level for all flash blocks
0x0E	Set Factory Margin Level	Specifies a factory margin read level for all flash blocks
0x0F	Configure NVM	Configure NVM parameters to enable or disable some features in the NVM array, allowing to save current/power under certain circumstances.

18.3.10 Flash command summary

This section provides details of all available flash commands launched by a command write sequence. The FSTAT[ACCERR] will be set during the command write sequence if any of the following illegal steps are performed, causing the command not to be processed by the memory controller:

- Starting any command write sequence that programs or erases flash memory before initializing the FLCKDIV register.
- Writing an invalid command as part of the command write sequence.
- For additional possible errors, refer to the error handling table provided for each command.

If a flash block is read during the execution of an algorithm (FSTAT[CCIF] = 0) on that same block, the read operation will return invalid data. It will also trigger an illegal access exception.

If FSTAT[ACCERR] or FSTAT[FPVIOL] are set, the user must clear these fields before starting any command write sequence.

CAUTION

An flash longword must be in the erased state before being programmed. Cumulative programming of bits within an flash longword is not allowed.

18.3.10.1 Erase Verify All Blocks command

The Erase Verify All Blocks command will verify that all flash blocks have been erased.

Table 18-9. Erase Verify All Blocks command FCCOB requirements

CCOBIX[2:0]	FCCOBHI parameters	FCCOBLO parameters
000	0x01	Not required

Upon clearing FSTAT[CCIF] to launch the Erase Verify All Blocks command, the memory controller will verify that the entire flash memory space is erased. The FSTAT[CCIF] flag will set after the erase verify all blocks operation has completed. If all blocks are not erased, it means blank check failed and both FSTAT[MGSTAT] bits will be set.

Table 18-10. Erase verify all blocks command error handling

Register	Error bit	Error condition
FSTAT	ACCERR	Set if CCOBIX[2:0] != 000 at command launch
	FPVIOL	None
	MGSTAT1	Set if any errors have been encountered during the read[1] or if blank check failed
	MGSTAT0	Set if any errors have been encountered during the read or if blank check failed

1. As found in the memory map for NVM

18.3.10.2 Erase Verify Block command

The Erase Verify Block command allows the user to verify that an entire flash block has been erased. The FCCOB global address [23:0] bits determine which block must be verified.

Table 18-11. Erase Verify Block Command FCCOB requirements

CCOBIX[2:0]	FCCOBHI parameters	FCCOBLO parameters
000	0x02	Global address [23:16] to identify flash block
001	Global address [15:0] in flash block to be verified	

Upon clearing FSTAT[CCIF] to launch the erase verify block command, the memory controller will verify that the selected flash block is erased. The FSTAT[CCIF] flag will set after the erase verify block operation has completed. If the block is not erased, it means blank check failed and both FSTAT[MGSTAT] bits will be set.

Table 18-12. Erase Verify Block command error handling

Register	Error bit	Error condition
FSTAT	ACCERR	Set if CCOBIX[2:0] != 001 at command launch
		Set if an invalid global address [23:0] is supplied[1]
	FPVIOL	None
	MGSTAT1	Set if any errors have been encountered during the read or if blank check failed
	MGSTAT0	Set if any errors have been encountered during the read or if blank check failed

1. As found in the memory map for NVM

18.3.10.3 Erase Verify Flash Section command

The Erase Verify Flash Section command will verify that a section of code in the flash memory is erased. The Erase Verify Flash Section command defines the starting point of the code to be verified and the number of longwords.

Table 18-13. Erase verify flash section command FCCOB requirements

CCOBIX[2:0]	FCCOBHI parameters	FCCOBLO parameters
000	0x03	Global address [23:16] of flash block
001	Global address [15:0] of the first longwords to be verified	
010	Number of long words to be verified	

Upon clearing FSTAT[CCIF] to launch the erase verify flash section command, the memory controller will verify that the selected section of flash memory is erased. The FSTAT[CCIF] flag will set after the erase verify flash section operation has completed. If the section is not erased, it means blank check failed and both FSTAT[MGSTAT] bits will be set.

Table 18-14. Erase Verify Flash Section command error handling

Register	Error bit	Error condition
FSTAT	ACCERR	Set if CCOBIX[2:0] != 010 at command launch
		Set if command not available in current mode (see Table 18-4)
		Set if an invalid global address [23:0] is supplied (see Table 18-1)[1]
		Set if a misaligned long words address is supplied (global address[1:0] != 00)
		Set if the requested section crosses flash address boundary
	FPVIOL	None
	MGSTAT1	Set if any errors have been encountered during the read[2] or if blank check failed

Table continues on the next page...

Table 18-14. Erase Verify Flash Section command error handling (continued)

Register	Error bit	Error condition
	MGSTAT0	Set if any errors have been encountered during the read[2] or if blank check failed

1. As defined by the memory map for NVM
2. As found in the memory map for NVM

18.3.10.4 Read once command

The read once command provides read access to a reserved 64-byte field (8 phrase) located in the nonvolatile information register of flash. The read once field can only be programmed once and can not be erased. It can be used to store the product ID or any other information that can be written only once. It is programmed using the program once command described in Program Once command. To avoid code runaway, the read once command must not be executed from the flash block containing the program once reserved field.

Table 18-15. Read Once command FCCOB requirements

CCOBIX[2:0]	FCCOB parameters	
000	0x04	Not required
001	Read once phrase index (0x0000 – 0x0007)	
010	Read once word 0 value	
011	Read once word 1 value	
100	Read once word 2 value	
101	Read once word 3 value	

Upon clearing FSTAT[CCIF] to launch the read once command, a read once phrase is fetched and stored in the FCCOB indexed register. The FSTAT[CCIF] flag will set after the read once operation has completed. Valid phrase index values for the read once command range from 0x0000 to 0x0007. During execution of the read once command, any attempt to read addresses within flash block will return invalid data.

Table 18-16. Read Once command error handling

Register	Error bit	Error condition
FSTAT	ACCERR	Set if CCOBIX[2:0] != 001 at command launch
		Set if command is not available in current mode (see Table 18-4)
		Set if an invalid phrase index is supplied
	FPVIOL	None
	MGSTAT1	Set if any errors have been encountered during the read
	MGSTAT0	Set if any errors have been encountered during the read

18.3.10.5 Program Flash command

The program flash operation will program up to two previously erased longwords in the flash memory using an embedded algorithm.

Note

A flash longword must be in the erased state before being programmed. Cumulative programming of bits within a flash longword is not allowed.

Table 18-17. Program Flash command FCCOB requirements

CCOBIX[2:0]	FCCOBHI parameters	FCCOBLO parameters
000	0x06	Global address [23:16] to identify flash block
001	Global address [15:0] of longwords location to be programmed[1]	
010	Word 0 (longword 0) program value	
011	Word 1 (longword 0) program value	
100	Word 2 (longword 1) program value	
101	Word 3 (longword 1) program value	

1. Global address [1:0] must be 00.

Upon clearing FSTAT[CCIF] to launch the Program Flash command, the memory controller will program the data longwords to the supplied global address and will then proceed to verify the data longwords read back as expected. The FSTAT[CCIF] flag will set after the program flash operation has completed.

Table 18-18. Program Flash command error handling

Register	Error bit	Error condition
FSTAT	ACCERR	Set if CCOBIX[2:0] ≠ 011 or 101 at command launch
		Set if command not available in current mode (see Table 18-4)
		Set if an invalid global address [23:0] is supplied (see Table 18-1.[1]
		Set if a misaligned longword address is supplied (global address [1:0] != 00)
		Set if the requested group of words breaches the end of the flash block.
	FPVIOL	Set if the global address [23:0] points to a protected data
	MGSTAT1	Set if any errors have been encountered during the verify operation
	MGSTAT0	Set if any errors have been encountered during the verify operation

1. As defined by the memory map of NVM.

18.3.10.6 Program Once command

The Program Once command restricts programming to a reserved 64-byte field (8 phrases) in the nonvolatile information register located in flash. The program once reserved field can be read using the read once command as described in Read once command. The program once command must be issued only once because the nonvolatile information register in flash cannot be erased. To avoid code runaway, the program once command must not be executed from the flash block containing the program once reserved field.

Table 18-19. Program Once command FCCOB requirements

CCOBIX[2:0]	FCCOB parameters	
000	0x07	Not required
001	Program Once phrase index (0x000 – 0x0007)	
010	Program once Word 0 value	
011	Program once Word 1value	
100	Program once Word 2 value	
101	Program once Word 3 value	

Upon clearing FSTAT[CCIF] to launch the program once command, the memory controller first verifies that the selected phrase is erased. If erased, then the selected phrase will be programmed and then verified with read back. The FSTAT[CCIF] flag will remain clear, setting only after the program once operation has completed.

The reserved nonvolatile information register accessed by the Program Once command cannot be erased, and any attempt to program one of these phrases a second time will not be allowed. Valid phrase index values for the program once command range from 0x0000 to 0x0007. During execution of the program once command, any attempt to read addresses within flash will return invalid data.

Table 18-20. Program Once ommand error handling

Register	Error bit	Error condition
FSTAT	ACCERR	Set if CCOBIX[2:0] != 101 at command launch
		Set if command not available in current mode (see Table 18-4)
		Set if an invalid phrase index is supplied
		Set if the requested phrase has already been programmed[1]
	FPVIOL	None
	MGSTAT1	Set if any errors have been encountered during the verify operation
	MGSTAT0	Set if any errors have been encountered during the verify operation

1. If a program once phrase is initially programmed to 0xFFFF_FFFF_FFFF_FFFF, the program once command will be allowed to execute again on that same phrase.

18.3.10.7 Erase All Blocks command

The Erase All Blocks operation will erase the entire flash memory space.

Table 18-21. Erase All Blocks command FCCOB requirements

CCOBIX[2:0]	FCCOBHI parameters	FCCOBLO parameters
000	0x08	Not required

Upon clearing FSTAT[CCIF] to launch the Erase All Blocks command, the memory controller will erase the entire NVM memory space and verify that it is erased. If the memory controller verifies that the entire NVM memory space was properly erased, security will be released. Therefore, the device is in unsecured state. During the execution of this command (FSTAT[CCIF] = 0) the user must not write to any NVM module register. The FSTAT[CCIF] flag will set after the erase all blocks operation has completed.

Table 18-22. Erase All Blocks command error handling

Register	Error bit	Error condition
FSTAT	ACCERR	Set if CCOBIX[2:0] ≠ 000 at command launch
		Set if command not available in current mode (see Table 18-4)
	FPVIOL	Set if any area of the flash memory is protected
	MGSTAT1	Set if any errors have been encountered during the verify operation[1]
	MGSTAT0	Set if any errors have been encountered during the verify operation[1]

1. As found in the memory map for NVM.

18.3.10.8 Debugger mass erase request

The functionality of the Erase All Blocks command is also available in an uncommanded fashion from the Debugger Mass Erase Request feature.

The Debugger Mass Erase request requires the clock divider register FCLKDIV to be loaded before invoking this function. Please look into the Reference Manual for information about the default value of FCLKDIV in case direct writes to register FCLKDIV are not allowed by the time this feature is invoked. If FCLKDIV is not set the Debugger Mass Erase request will not execute and the FSTAT[ACCERR] flag will set. After the execution of the Mass Erase function, the FCLKDIV register will be reset and the value of register FCLKDIV must be loaded before launching any other command afterwards.

Before invoking the erase-all function, the FSTAT[ACCERR]and FSTAT[FPVIOL] flags must be clear. When invoked the Debugger Mass Erase request will erase all flash memory space regardless of the protection settings. If the post-erase verify passes, the routine will then release security by setting the FSEC[SEC] to the unsecure state. The security byte in the Flash Configuration Field will be programmed to the unsecure state. The status of the Debugger Mass Erase request is reflected in the FCNFG[ERSAREQ]. The FCNFG[ERSAREQ] will be cleared once the operation has completed and the normal FSTAT error reporting will be available as described in the following table.

At the end of the Mass Erase sequence Protection will remain configured as it was before executing the Mass Erase function. If the application requires programming P-Flash after the Mass Erase function completes, the existing protection limits must be taken into account. If protection needs to be disabled the user may need to reset the system right after completing the Mass Erase function.

Table 18-23. Debugger mass erase request error handling

Register	Error Bit	Error Condition
FSTAT	ACCERR	Set if command not available in current mode.
	MGSTAT1	Set if any errors have been encountered during the erase verify operation, or during the program verify operation.
	MGSTAT0	Set if any non-correctable errors have been encountered during the erase verify operation, or during the program verify operation.

18.3.10.9 Erase flash block command

The erase flash block operation will erase all addresses in a flash block.

Table 18-24. Erase flash block command FCCOB requirements

CCOBIX[2:0]	FCCOB parameters	
000	0x09	Global address [23:16] to identify flash block
001	Global address[15:0] in flash block to be erased	

Upon clearing FSTAT[CCIF] to launch the erase flash block command, the memory controller will erase the selected flash block and verify that it is erased. The FSTAT[CCIF] flag will set after the erase flash block operation has completed.

Table 18-25. Erase flash block command error handling

Register	Error Bit	Error Condition
FSTAT	ACCERR	Set if CCOBIX[2:0] != 001 at command launch
		Set if command not available in current mode (see Table 18-4)

Table continues on the next page...

Table 18-25. Erase flash block command error handling (continued)

Register	Error Bit	Error Condition
		Set if an invalid global address [23:0] is supplied[1]
	FPVIOL	Set if an area of the selected flash block is protected
	MGSTAT1	Set if any errors have been encountered during the verify operation[2]
	MGSTAT0	Set if any errors have been encountered during the verify operation[2]

1. As defined by the memory map for NVM.
2. As found in the memory map for NVM.

18.3.10.10 Erase flash sector command

The erase flash sector operation will erase all addresses in a flash sector.

Table 18-26. Erase flash sector command FCCOB requirements

CCOBIX[2:0]	FCCOB parameters	
000	0x0A	Global address [23:16] to identify flash block to be erased
001	Global address [15:0] anywhere within the sector to be erased. Refer to Overview for the flash sector size	

Upon clearing FSTAT[CCIF] to launch the erase flash sector command, the memory controller will erase the selected flash sector and then verify that it is erased. The FSTAT[CCIF] flag will be set after the erase flash sector operation has completed.

Table 18-27. Erase flash sector command error handling

Register	Error bit	Error condition
FSTAT	ACCERR	Set if CCOBIX[2:0] != 001 at command launch
		Set if command not available in current mode (see Table 18-4)
		Set if an invalid global address [23:0] is supplied.[1] (see Table 18-1)
		Set if a misaligned longword address is supplied (global address [1:0] != 00)
	FPVIOL	Set if the selected flash sector is protected
	MGSTAT1	Set if any errors have been encountered during the verify operation
	MGSTAT0	Set if any errors have been encountered during the verify operation

1. As defined by the memory map for NVM

18.3.10.11 Unsecure flash command

The unsecure flash command will erase the entire flash memory space, and if the erase is successful, will release security.

Table 18-28. Unsecure flash command FCCOB requirements

CCOBIX[2:0]	FCCOB parameters	
000	0x0B	Not required

Upon clearing FSTAT[CCIF] to launch the unsecure flash command, the memory controller will erase the entire flash memory space and verify that it is erased. If the memory controller verifies that the entire flash memory space was properly erased, security will be released. If the erase verify is not successful, the unsecure flash operation sets FSTAT[MGSTAT1] and terminates without changing the security state. During the execution of this command (FSTAT[CCIF] = 0), the user must not write to any flash module register. The FSTAT[CCIF] flag is set after the unsecure flash operation has completed.

Table 18-29. Unsecure flash command error handling

Register	Error bit	Error condition
FSTAT	ACCERR	Set if CCOBIX[2:0] != 000 at command launch
		Set if command is not available in current mode (see Table 18-4)
	FPVIOL	Set if any area of the flash memory is protected
	MGSTAT1	Set if any errors have been encountered during the verify operation[1]
	MGSTAT0	Set if any errors have been encountered during the verify operation[1]

1. As found in the memory map for NVM

18.3.10.12 Verify backdoor access key command

The verify backdoor access key command will execute only if it is enabled by the FSEC[KEYEN] bits. The verify backdoor access key command releases security if user-supplied keys match those stored in the flash security bytes of the flash configuration field. See Table 18-1 for details. The code that performs verifying backdoor access command must be running from RAM.

Table 18-30. Verify backdoor access key command FCCOB requirements

CCOBIX[2:0]	FCCOBHI parameters	FCCOBLO parameters
000	0x0C	Not required
001	Key 0	
010	Key 1	
011	Key 2	
100	Key 3	

Upon clearing FSTAT[CCIF] to launch the verify backdoor access key command, the memory controller will check the FSEC[KEYEN] bits to verify that this command is enabled. If not enabled, the memory controller sets the FSTAT[ACCERR] bit. If the command is enabled, the memory controller compares the key provided in FCCOB to the backdoor comparison key in the flash configuration field with Key 0 compared to 0x0400, and so on. If the backdoor keys match, security will be 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 aborted (set FSTAT[ACCERR]) until a reset occurs. The FSTAT[CCIF] flag is set after the verify backdoor access key operation has completed.

Table 18-31. Verify backdoor access key command error handling

Register	Error bit	Error condition
FSTAT	ACCERR	Set if CCOBIX[2:0] ≠ 100 at command launch
		Set if an incorrect backdoor key is supplied
		Set if backdoor key access has not been enabled (KEYEN[1:0] ≠ 10
		Set if the backdoor key has mismatched since the last reset
	FPVIOL	None
	MGSTAT1	None
	MGSTAT0	None

18.3.10.13 Set user margin level command

The user margin is a small delta to the normal read reference level and, in effect, is a minimum safety margin. That is, if the reads pass at the tighter tolerances of the user margins, the normal reads have at least that much safety margin before users experience data loss.

The set user margin level command causes the memory controller to set the margin level for future read operations of the flash block.

Table 18-32. Set user margin level command FCCOB requirements

CCOBIX[2:0]	FCCOBHI parameters	FCCOBLO parameters
000	0x0D	Global address [23:16] to identify flash block
001	Global address [15:0] to identify flash block	
010	Margin level setting	

Upon clearing FSTAT[CCIF] to launch the set user margin level command, the memory controller will set the user margin level for the targeted block and then set the FSTAT[CCIF] flag.

Note

Valid margin level settings for the set user margin level command are defined in the following tables.

Table 18-33. Valid set user margin level settings

CCOB (CCOBIX = 010)	Level description
0x0000	Return to normal level
0x0001	User margin-1 level[1]
0x0002	User margin-0 level[2]

1. Read margin to the erased state
2. Read margin to the programmed state

Table 18-34. Set user margin level command error handling

Register	Error bit	Error condition
FSTAT	ACCERR	Set if CCOBIX[2:0] != 010 at command launch
		Set if command is not available in current mode (see Table 18-4)
		Set if an invalid global address [23:0] is supplied
		Set if an invalid margin level setting is supplied
	FPVIOL	None
	MGSTAT1	None
	MGSTAT0	None

Note

User margin levels can be used to check that NVM memory contents have adequate margin for normal level read operations. If unexpected results are encountered when checking NVM memory contents at user margin levels, a potential loss of information has been detected.

18.3.10.14 Set factory margin level command

The set factory margin Level command causes the memory controller to set the margin level specified for future read operations of the flash block.

Table 18-35. Set factory margin level command FCCOB requirements

CCOBIX[2:0]	FCCOBHI parameters	FCCOBLO parameters
000	0x0E	Global address [23:16] to identify flash block

Table continues on the next page...

Table 18-35. Set factory margin level command FCCOB requirements (continued)

CCOBIX[2:0]	FCCOBHI parameters	FCCOBLO parameters
001	Global address [15:0] to identify flash block	
010	Margin level setting	

Upon clearing FSTAT[CCIF] to launch the set factory margin level command, the memory controller will set the factory margin level for the targeted block and then set the FSTAT[CCIF] flag.

Note

Valid margin level settings for the set factory margin level command are defined in the following tables.

Table 18-36. Valid set factory margin level settings

CCOB (CCOBIX = 010)	Level description
0x0000	Return to normal level
0x0001	User margin-1 level[1]
0x0002	User margin-0 level[2]
0x0003	Factory margin-1 level[1]
0x0004	Factory margin-0 level[2]

1. Read margin to the erased state
2. Read margin to the programmed state

Table 18-37. Set factory margin level command error handling

Register	Error bit	Error condition
FSTAT	ACCERR	Set if CCOBIX[2:0] != 010 at command launch
		Set if command is not available in current mode (see Table 18-4)
		Set if an invalid global address [23:0] is supplied
		Set if an invalid margin level setting is supplied
	FPVIOL	None
	MGSTAT1	None
	MGSTAT0	None

CAUTION

Factory margin levels must only be used during verify of the initial factory programming.

Note

Factory margin levels can be used to check that Flash memory contents have adequate margin for data retention at the normal level setting. If unexpected results are encountered when checking flash memory contents at factory margin levels, the fash memory contents must be erased and reprogrammed.

18.3.10.15 Configure NVM command

The Configure NVM command allows the user to control some features in the NVM array, enabling or disabling them with the purpose to save current/power under certain circumstances. The settings modified with Configure NVM command are not permanent, therefore the command must be invoked after each reset if changing these settings is required in the user application.

Table 18-38. Configure NVM command FCCOB requirements

CCOBIX[2:0]	FCCOBHI parameters	FCCOBLO parameters
000	0x0F	Configure NVM control byte as the following table.

Table 18-39. NVM Control Byte - field descriptions

Field	Description
7 QUERY	**Query / Drive Mode** - This bit controls if Configure NVM command will be used to drive or to retrieve current status of FLPHV and FLPLF bits. 0 Drive option on bits FLPHV and FLPLF into the NVM array 1 Query current status of FLPHV and FLPLF. FCCOB[0] can be read after command execution to retrieve current status of FLPHV and FLPLV.
6-2 RNV	**Reserved Bits** - The RNV bits should be 1'b0, reserved for future enhancements.
1 FLPHV	**Flash Low Power Control With High Supply Voltage** - The FLPHV bit controls a circuit in the NVM Array that generates an internal voltage reference to be used in Flash reads when the device is operating in the lower range of the external supply voltage specification. Under specific circumstances this circuit can be disabled to save power. 0 Circuit enabled (recommended, default value) - this ensures that the NVM Array will work correctly for reads across the full range of the external supply voltage. 1 Circuit disabled - to be set only if the external voltage supply is guaranteed to drive voltage above a minimum level.
0 FLPLF	**Flash Low Power Control In Low Frequency** - The FLPLF bit controls a circuit in the NVM Array that limits current consumption during Flash reads if the device is running in low frequencies. The circuit itself consumes some power and is recommended to be disabled for typical frequency settings. Under specific circumstances (low bus frequency) this circuit can be enabled to save power. 0 Circuit disabled (recommended, default value[1]) - for typical bus frequency values this circuit can be left disabled to minimize power consumption.

Table 18-39. NVM Control Byte - field descriptions

Field	Description
	1 Circuit enabled - to be set if the bus frequency is setup to be below a minimum threshold, so the Flash will save power while operating in low frequency.

1. Freescale might deliver parts with a different initial setting for this control in the NVM Array, eventually under a different part number, if the part is to be used specifically in a low frequency range of operation. Therefore, if the user is not sure about the initial setting of this control it is advisable to first run the Configure NVM command in Query mode to retrieve the initial status of FLPLF.

LPHV controls an internal circuit that guarantees correct Flash reads across the full range of the external voltage supply, more specifically in the lower levels of the voltage specification. FLPHV can be used to disable this circuit if the external supply source drives the voltage above a minimum level specified in the NVM Electrical Parameters - please refer to the proper section in the Reference Manual to identify this minimum required voltage level and the current savings in case this circuit can be disabled.

CAUTION

There is a risk if the user incorrectly disables the circuit controlled by FLPHV and the operating voltage goes below the minimum supply voltage level as specified. Under this condition the Flash will not read meaningful data, with unpredictable results. The only way to restore the Flash functionality may be to reset the part, so the NVM Array will be restored to default conditions (i.e. the circuit controlled by FLPHV is enabled)

FLPLF controls an internal circuit that limits current consumption during Flash reads if the Flash is running at lower bus frequencies (BUSCLK). Under typical frequency values this circuit is recommended to be disabled to save power, but for applications that run at low frequencies this circuit can be enabled to minimize current consumption. For details about the threshold bus frequency value under which there can be some power savings by enabling this circuit.

NOTE

Bit FLPLF relates to Read operations, that are affected by the bus frequency (BUSCLK), whereas program and erase operations are affected by the FCLK time base that is unrelated to FLPLF.

Upon clearing CCIF to launch the Configure NVM command the MGATE will set the NVM Array parameters accordingly (if the command is launched in Drive mode - QUERY=0) and then set the CCIF flag. The current status of FLPHV and FLPLV can

always be read in FCCOB[0] after the command execution. The Configure NVM command can be executed in Query Mode (QUERY=1) to retrieve the status of these bits without driving them into the NVM array.

The user is not required to run the Configure NVM command for correct operation of the Flash array. The default value after reset are the recommended / safe states for operation of the Flash. The Configure NVM command is intended to be invoked only in conditions where it can be used to save some power, as explained above.

Table 18-40. Configure NVM command error handling

Register	Error bit	Error condition
FSTAT	ACCERR	Set if CCOBIX[2:0] != 000 at command launch
		Set if command is not available in current mode (see Table 18-4)
		Set if an invalid global address [23:0] is supplied
		Set if an invalid margin level setting is supplied
	FPVIOL	None
	MGSTAT1	None
	MGSTAT0	None

18.4 Memory map and register definition

This section presents a high-level summary of the registers and how they are mapped. The registers can be accessed in 32-bits, 16-bits (aligned on data[31:16] or on data[15:0]) or 8-bits. In the case of the writable registers, the write accesses are forbidden during flash command execution. For more details, see Caution note in Flash memory map.

FTMRE memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4002_0001	Flash CCOB Index Register (FTMRE_FCCOBIX)	8	R/W	00h	18.4.1/210
4002_0002	Flash Security Register (FTMRE_FSEC)	8	R	Undefined	18.4.2/210
4002_0003	Flash Clock Divider Register (FTMRE_FCLKDIV)	8	R/W	00h	18.4.3/211
4002_0005	Flash Status Register (FTMRE_FSTAT)	8	R/W	80h	18.4.4/212
4002_0007	Flash Configuration Register (FTMRE_FCNFG)	8	R/W	00h	18.4.5/213
4002_0008	Flash Common Command Object Register: Low (FTMRE_FCCOBLO)	8	R/W	00h	18.4.6/214
4002_0009	Flash Common Command Object Register:High (FTMRE_FCCOBHI)	8	R/W	00h	18.4.7/214
4002_000B	Flash Protection Register (FTMRE_FPROT)	8	R	See section	18.4.8/215
4002_000F	Flash Option Register (FTMRE_FOPT)	8	R	Undefined	18.4.9/216

18.4.1 Flash CCOB Index Register (FTMRE_FCCOBIX)

The FCCOBIX register is used to index the FCCOB register for NVM memory operations.

Address: 4002_0000h base + 1h offset = 4002_0001h

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

FTMRE_FCCOBIX field descriptions

Field	Description
7–3 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
CCOBIX	Common Command Register Index Selects which word of the FCCOB register array is being read or written to.

18.4.2 Flash Security Register (FTMRE_FSEC)

The FSEC register holds all bits associated with the security of the MCU and NVM module. All fields in the FSEC register are readable but not writable. During the reset sequence, the FSEC register is loaded with the contents of the flash security byte in the flash configuration field located in flash memory.

See Security for security function.

Address: 4002_0000h base + 2h offset = 4002_0002h

Bit	7	6	5	4	3	2	1	0
Read	KEYEN		Reserved				SEC	
Write								
Reset	x*	x*	x*	x*	x*	x*	x*	x*

* Notes:
- x = Undefined at reset.

FTMRE_FSEC field descriptions

Field	Description
7–6 KEYEN	Backdoor Key Security Enable Bits The KEYEN[1:0] bits define the enabling of backdoor key access to the flash module. **NOTE:** 01 is the preferred KEYEN state to disable backdoor key access.

Table continues on the next page...

FTMRE_FSEC field descriptions (continued)

Field	Description
	00 Disabled 01 Disabled 10 Enabled 11 Disabled
5–2 Reserved	This field is reserved.
SEC	Flash Security Bits Defines the security state of the MCU. If the flash module is unsecured using backdoor key access, the SEC field is forced to 10. **NOTE:** 00 is the preferred SEC state to set MCU to secured state. 00 Secured 01 Secured 10 Unsecured 11 Secured

18.4.3 Flash Clock Divider Register (FTMRE_FCLKDIV)

The FCLKDIV register is used to control timed events in program and erase algorithms.

All bits in the FCLKDIV register are readable, bit 7 is not writable, bit 6 is write-once-high and controls the writability of the FDIV field in user mode. In debug mode, bits 6-0 are writable any number of times but bit 7 remains unwritable.

NOTE

The FCLKDIV register must not be written while a flash command is executing (FSTAT[CCIF] = 0)

Address: 4002_0000h base + 3h offset = 4002_0003h

Bit	7	6	5	4	3	2	1	0
Read	FDIVLD	FDIVLCK	FDIV					
Write								
Reset	0	0	0	0	0	0	0	0

FTMRE_FCLKDIV field descriptions

Field	Description
7 FDIVLD	Clock Divider Loaded 0 FCLKDIV register has not been written since the last reset. 1 FCLKDIV register has been written since the last reset.

Table continues on the next page...

FTMRE_FCLKDIV field descriptions (continued)

Field	Description
6 FDIVLCK	Clock Divider Locked 0 FDIV field is open for writing. 1 FDIV value is locked and cannot be changed. After the lock bit is set high, only reset can clear this bit and restore writability to the FDIV field in user mode.
FDIV	Clock Divider Bits FDIV[5:0] must be set to effectively divide BUSCLK down to 1MHz to control timed events during flash program and erase algorithms. Refer to the table in the Writing the FCLKDIV register for the recommended values of FDIV based on the BUSCLK frequency.

18.4.4 Flash Status Register (FTMRE_FSTAT)

The FSTAT register reports the operational status of the flash module.

Address: 4002_0000h base + 5h offset = 4002_0005h

Bit	7	6	5	4	3	2	1	0
Read	CCIF	0	ACCERR	FPVIOL	MGBUSY	0	MGSTAT	
Write								
Reset	1	0	0	0	0	0	0	0

FTMRE_FSTAT field descriptions

Field	Description
7 CCIF	Command Complete Interrupt Flag Indicates that a flash command has completed. The CCIF flag is cleared by writing a 1 to CCIF to launch a command and CCIF will stay low until command completion or command violation. 0 Flash command is in progress. 1 Flash command has completed.
6 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
5 ACCERR	Flash Access Error Flag Indicates an illegal access has occurred to the flash memory caused by either 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. Writing 1 to this field clears it while writing a 0 to this field has no effect. 0 No access error is detected. 1 Access error is detected.
4 FPVIOL	Flash Protection Violation Flag

Table continues on the next page...

FTMRE_FSTAT field descriptions (continued)

Field	Description
	Indicates an attempt was made to program or erase an address in a protected area of flash memory during a command write sequence. Writing 1 to FPVIOL clears this field while writing 0 to this field has no effect. While FPIOL is set, it is not possible to launch a command or start a command write sequence. 0 No protection violation is detected. 1 Protection violation is detected.
3 MGBUSY	Memory Controller Busy Flag Reflects the active state of the memory controller. 0 Memory controller is idle. 1 Memory controller is busy executing a flash command (CCIF = 0).
2 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
MGSTAT	Memory Controller Command Completion Status Flag One or more MGSTAT flag bits are set if an error is detected during execution of a flash command or during the flash reset sequence. **NOTE:** Reset value can deviate from the value shown if a double bit fault is detected during the reset sequence.

18.4.5 Flash Configuration Register (FTMRE_FCNFG)

The FCNFG register enables the flash command complete interrupt.

Address: 4002_0000h base + 7h offset = 4002_0007h

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

FTMRE_FCNFG field descriptions

Field	Description
7 CCIE	Command Complete Interrupt Enable Controls interrupt generation when a flash command has completed. 0 Command complete interrupt is disabled. 1 An interrupt will be requested whenever the CCIF flag in the FSTAT register is set.
6 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
5 ERSAREQ	Debugger Mass Erase Request

Table continues on the next page...

FTMRE_FCNFG field descriptions (continued)

Field	Description
	Requests the MGATE to execute the Erase All Blocks command and release security. ERSAREQ is not directly writable but is under indirect user control. The ERSAREQ field sets to 1 when the MGATE starts executing the sequence. ERSAREQ will be reset to 0 by the MGATE when the operation is completed 0 No request or request complete 1 Request to • run the Erase All Blocks command • verify the erased state • program the security byte in the Flash Configuration Field to the unsecure state • release MCU security by setting FSEC[SEC] to the unsecure state
Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

18.4.6 Flash Common Command Object Register: Low (FTMRE_FCCOBLO)

The FCCOB is an array of six words addressed via the CCOBIX index found in the FCCOBIX register. Byte-wide reads and writes are allowed to the FCCOB register.

Address: 4002_0000h base + 8h offset = 4002_0008h

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

FTMRE_FCCOBLO field descriptions

Field	Description
CCOB	Common Command Object Bit 7:0 Low 8 bits of Common Command Object register

18.4.7 Flash Common Command Object Register:High (FTMRE_FCCOBHI)

The FCCOB is an array of six words addressed via the CCOBIX index found in the FCCOBIX register. Byte-wide reads and writes are allowed to the FCCOB register.

Address: 4002_0000h base + 9h offset = 4002_0009h

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

FTMRE_FCCOBHI field descriptions

Field	Description
CCOB	Common Command Object Bit 15:8 High 8 bits of Common Command Object register

18.4.8 Flash Protection Register (FTMRE_FPROT)

The FPROT register defines which flash sectors are protected against program and erase operations.

The unreserved bits of the FPROT register are writable with the restriction that the size of the protected region can only be increased (see Protection). All the unreserved bits of the FPROT register are writable without restriction in debug mode.

During the reset sequence, the FPROT register is loaded with the contents of the flash protection byte in the flash configuration field at global address 0x40D located in flash memory. To change the flash protection that will be loaded during the reset sequence, the upper sector of the flash memory must be unprotected, then the flash protection byte must be reprogrammed.

Trying to alter data in any protected area in the flash memory will result in a protection violation error and the FPVIOL bit will be set in the FSTAT register. The block erase of a flash block is not possible if any of the flash sectors contained in the same flash block are protected.

Address: 4002_0000h base + Bh offset = 4002_000Bh

Bit	7	6	5	4	3	2	1	0
Read	FPOPEN	RNV6	RNV			FPLDIS	FPLS	
Write								
Reset	x*	x*	0	0	0	x*	x*	x*

* Notes:
- x = Undefined at reset.

FTMRE_FPROT field descriptions

Field	Description
7 FPOPEN	Flash Protection Operation Enable The FPOPEN bit determines the protection function for program or erase operations.

Table continues on the next page...

FTMRE_FPROT field descriptions (continued)

Field	Description
	0 When FPOPEN is clear, the FPLDIS field defines unprotected address ranges as specified by the corresponding FPLS field. 1 When FPOPEN is set, the FPLDIS field enables protection for the address range specified by the corresponding FPLS field.
6 RNV6	Reserved Nonvolatile Bit The RNV bit must remain in the erased state.
5–3 RNV	Reserved Nonvolatile Bit The RNV bit must remain in the erased state.
2 FPLDIS	Flash Protection Lower Address Range Disable The FPLDIS bit determines whether there is a protected/unprotected area in a specific region of the flash memory beginning with global address 0x0_0000. 0 Protection/Unprotection enabled. 1 Protection/Unprotection disabled.
FPLS	Flash Protection Lower Address Size The FPLS bits determine the size of the protected/unprotected area in flash memory. The FPLS bits can only be written to while the FPLDIS bit is set.

18.4.9 Flash Option Register (FTMRE_FOPT)

The FOPT register is the flash option register.

All bits in the FOPT register are readable but are not writable. In debug mode, all bits in the FOPT register are readable and writable .

During the reset sequence, the FOPT register is loaded from the flash nonvolatile byte in the flash configuration field at global address 0x040F located in flash memory as indicated by reset condition.

Address: 4002_0000h base + Fh offset = 4002_000Fh

Bit	7	6	5	4	3	2	1	0
Read	NV							
Write								
Reset	x*	x*	x*	x*	x*	x*	x*	x*

* Notes:
- x = Undefined at reset.

FTMRE_FOPT field descriptions

Field	Description
NV	Nonvolatile Bits

FTMRE_FOPT field descriptions (continued)

Field	Description
	The NV[7:0] bits are available as nonvolatile bits. During the reset sequence, the FOPT register is loaded from the flash nonvolatile byte in the flash configuration field at global address 0x40F located in flash memory.

Chapter 19
Internal Clock Source (ICS)

19.1 Introduction

The internal clock source (ICS) module provides clock source choices for the MCU. The module contains a frequency-locked loop (FLL) as a clock source that is controllable by either an internal or an external reference clock. The module can provide this FLL clock or either of the internal or external reference clocks as a source for the MCU system clock. There are also signals provided to control a low-power oscillator (OSC) module. These signals configure and enable the OSC module to generate its external crystal/ resonator clock (OSC_OUT) used by peripheral modules and as the ICS external reference clock source. The ICS external reference clock can be the external crystal/ resonator (OSC_OUT) supplied by an OSC, or it can be another external clock source.

The ICS clock source chosen is passed through a reduced bus divider (BDIV) which allows a lower final output clock frequency to be derived.

19.1.1 Features

The key features of the ICS module are given below:

- Internal reference clock has 9 trim bits for accuracy
- Internal or external reference clocks can be used to control the FLL.
- Selectable dividers for external reference clock to ensure proper input frequency to FLL.
- Internal or external reference clocks can be selected as the clock source for the MCU.
- FLL Engaged Internal mode is automatically selected out of reset.
- FLL lock detector and external clock monitor
  - FLL lock detector with interrupt capability
  - External reference clock monitor with reset capability
- Digitally controlled oscillator optimized for 40-48 MHz frequency range

19.1.2 Block diagram

The following figure is the ICS block diagram.

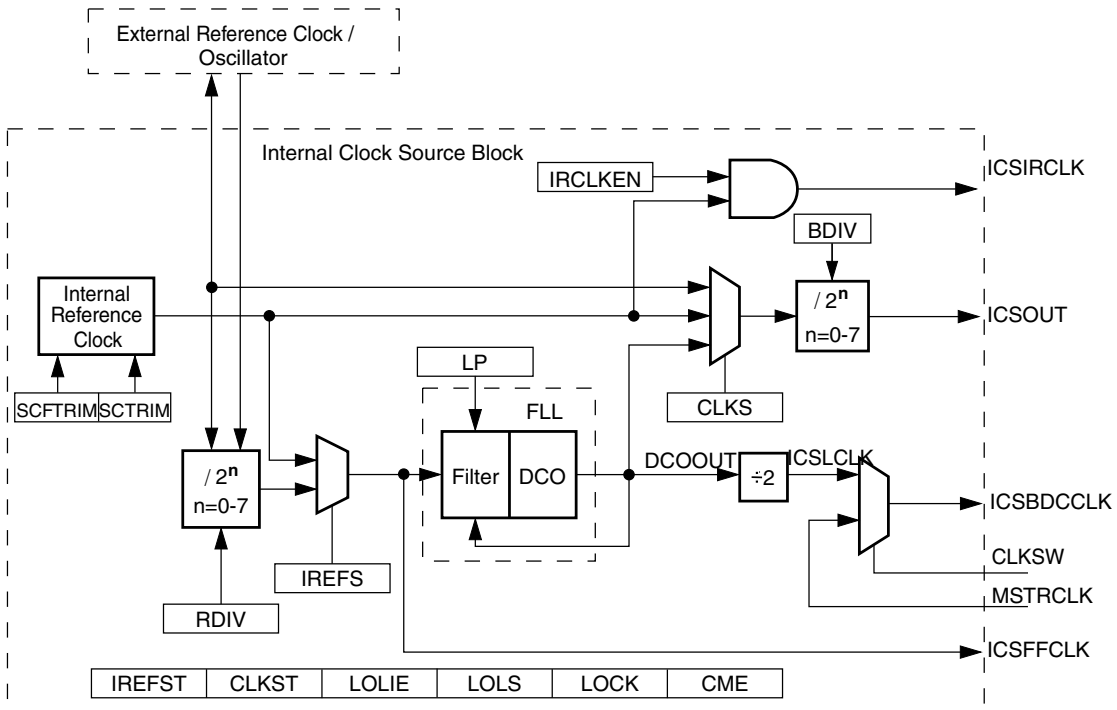


Figure 19-1. Internal clock source (ICS) block diagram

19.1.3 Modes of operation

There are seven modes of operation for the ICS: FEI, FEE, FBI, FBILP, FBE, FBELP, and STOP. Each of these modes is explained briefly in the following subsections.

19.1.3.1 FLL engaged internal (FEI)

In FLL engaged internal mode, which is the default mode, the ICS supplies a clock derived from the FLL which is controlled by the internal reference clock.

19.1.3.2 FLL engaged external (FEE)

In FLL engaged external mode, the ICS supplies a clock derived from the FLL which is controlled by an external reference clock source.

19.1.3.3 FLL bypassed internal (FBI)

In FLL bypassed internal mode, the FLL is enabled and controlled by the internal reference clock, but is bypassed. The ICS supplies a clock derived from the internal reference clock.

19.1.3.4 FLL bypassed internal low power (FBILP)

In FLL bypassed internal low power mode, the FLL is disabled and bypassed, and the ICS supplies a clock derived from the internal reference clock.

19.1.3.5 FLL bypassed external (FBE)

In FLL bypassed external mode, the FLL is enabled and controlled by an external reference clock, but is bypassed. The ICS supplies a clock derived from the external reference clock source.

19.1.3.6 FLL bypassed external low power (FBELP)

In FLL bypassed external low power mode, the FLL is disabled and bypassed, and the ICS supplies a clock derived from the external reference clock.

19.1.3.7 Stop (STOP)

In Stop mode, the FLL is disabled and the internal or the ICS external reference clocks source (OSC_OUT) can be selected to be enabled or disabled. The ICS does not provide any MCU clock sources.

NOTE

The DCO frequency changes from the pre-stop value to its reset value and the FLL needs to reacquire the lock before the frequency is stable. Timing sensitive operations must wait for the FLL acquisition time, $t_{Acquire}$, before executing.

19.2 External signal description

There are no ICS signals that connect off chip.

19.3 Register definition

ICS memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4006_4000	ICS Control Register 1 (ICS_C1)	8	R/W	04h	19.3.1/222
4006_4001	ICS Control Register 2 (ICS_C2)	8	R/W	20h	19.3.2/223
4006_4002	ICS Control Register 3 (ICS_C3)	8	R/W	Undefined	19.3.3/224
4006_4003	ICS Control Register 4 (ICS_C4)	8	R/W	See section	19.3.4/225
4006_4004	ICS Status Register (ICS_S)	8	R	10h	19.3.5/226

19.3.1 ICS Control Register 1 (ICS_C1)

Address: 4006_4000h base + 0h offset = 4006_4000h

Bit	7	6	5	4	3	2	1	0
Read Write	CLKS		RDIV			IREFS	IRCLKEN	IREFSTEN
Reset	0	0	0	0	0	1	0	0

ICS_C1 field descriptions

Field	Description
7–6 CLKS	Clock Source Select Selects the clock source that controls the bus frequency. The actual bus frequency depends on the value of ICS_C2[BDIV]. 00 Output of FLL is selected. 01 Internal reference clock is selected. 10 External reference clock is selected. 11 Reserved, defaults to 00.
5–3 RDIV	Reference Divider Changing RDIV will cause the change of reference clock frequency of FLL, RDIV is not allowed to be changed in FEE/FBE mode. Selects the amount to divide down the FLL reference clock selected by the IREFS bits. Resulting frequency must be in the range 31.25 kHz to 39.0625 kHz.

Table continues on the next page...

ICS_C1 field descriptions (continued)

Field	Description
	RDIV / OSC_CR[RANGE]= 0 / OSC_CR[RANGE]= 1
	000 / 1[1] / 32
	001 / 2 / 64
	010 / 4 / 128
	011 / 8 / 256
	100 / 16 / 512
	101 / 32 / 1024
	110 / 64 / Reserved
	111 / 128 / Reserved
	1. Reset default
2 IREFS	Internal Reference Select Selects the reference clock source for the FLL. 0 External reference clock is selected. 1 Internal reference clock is selected.
1 IRCLKEN	Internal Reference Clock Enable Enables the internal reference clock for use as ICSIRCLK. 0 ICSIRCLK is inactive. 1 ICSIRCLK is active.
0 IREFSTEN	Internal Reference Stop Enable Controls whether or not the internal reference clock remains enabled when the ICS enters Stop mode. 0 Internal reference clock is disabled in Stop mode. 1 Internal reference clock stays enabled in Stop mode if IRCLKEN is set, or if ICS is in FEI, FBI, or FBILP mode before entering Stop.

1. Reset default

19.3.2 ICS Control Register 2 (ICS_C2)

Address: 4006_4000h base + 1h offset = 4006_4001h

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

ICS_C2 field descriptions

Field	Description
7–5 BDIV	Bus Frequency Divider

Table continues on the next page...

ICS_C2 field descriptions (continued)

Field	Description
	Selects the amount to divide down the clock source selected by ICS_C1[CLKS]. This controls the bus frequency. 000 Encoding 0—Divides the selected clock by 1. 001 Encoding 1—Divides the selected clock by 2 (reset default). 010 Encoding 2—Divides the selected clock by 4. 011 Encoding 3—Divides the selected clock by 8. 100 Encoding 4—Divides the selected clock by 16. 101 Encoding 5—Divides the selected clock by 32. 110 Encoding 6—Divides the selected clock by 64. 111 Encoding 7—Divides the selected clock by 128.
4 LP	Low Power Select Controls whether the FLL is disabled in FLL bypassed modes. 0 FLL is not disabled in bypass mode. 1 FLL is disabled in bypass modes unless debug is active.
Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

19.3.3 ICS Control Register 3 (ICS_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.

ICS_C3 field descriptions

Field	Description
SCTRIM	Slow Internal Reference Clock Trim Setting Controls the slow internal reference clock frequency by controlling the internal reference clock period. The bits are binary weighted. In other words, bit 1 adjusts twice as much as bit 0. Increasing the binary value of SCTRIM will increase the period, and decreasing the value will decrease the period. An additional fine trim bit is available as the ICS_C4[SCFTRIM]. ICS_C3 is automatically loaded during reset from a factory programmed location when not in a debug mode. The factory programmed trim value adjusts the internal oscillator frequency to fint_ft as specified in the datasheet. The user can provide a custom trim value to attain other internal reference clock frequencies within the fint_t range. The custom trim value must be programmed into reserved flash location 0x0000_03FF and copied to ICS_C3 during code initialization.

19.3.4 ICS Control Register 4 (ICS_C4)

Address: 4006_4000h base + 3h offset = 4006_4003h

Bit	7	6	5	4	3	2	1	0
Read	LOLIE	0	CME	0				SCFTRIM
Write								
Reset	0	0	0	0	0	0	0	x*

* Notes:
- x = Undefined at reset.

ICS_C4 field descriptions

Field	Description
7 LOLIE	Loss of Lock Interrupt Determines if an interrupt request is made following a loss of lock indication. This field has an effect only when ICS_S[LOLS] is set. 0 No request on loss of lock. 1 Generates an interrupt request on loss of lock.
6 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
5 CME	Clock Monitor Enable Determines if a reset request is made following a loss of external clock indication. This field must be set to a logic 1 only when the ICS is in an operational mode that uses the external clock (FEE, FBE, or FBELP). 0 Clock monitor is disabled. 1 Generates a reset request on loss of external clock.
4–1 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
0 SCFTRIM	Slow Internal Reference Clock Fine Trim Controls the smallest adjustment of the internal reference clock frequency. Setting SCFTRIM will increase the period and clearing SCFTRIM will decrease the period by the smallest amount possible. ICS_C4[SCFTRIM] is automatically loaded during reset from a factory programmed location when not in a debug mode. The factory programmed trim value adjusts the internal oscillator frequency to fint_ft as specified in the datasheet. The user can provide a custom trim value to attain other internal reference clock frequencies within the fint_t range. The custom fine trim bit value must be programmed into reserved flash location 0x0000_03FE and copied to ICS_C4 during code initialization.

19.3.5 ICS Status Register (ICS_S)

Address: 4006_4000h base + 4h offset = 4006_4004h

Bit	7	6	5	4	3	2	1	0
Read	LOLS	LOCK	0	IREFST	CLKST		0	
Write	w1c							
Reset	0	0	0	1	0	0	0	0

ICS_S field descriptions

Field	Description
7 LOLS	Loss of Lock Status Indicates the lock status for the FLL. LOLS is set when lock detection is enabled and after acquiring lock, the FLL output frequency has fallen outside the lock exit frequency tolerance, from ±4.7% to ±5.97%. ICS_C4[LOLIE] determines whether an interrupt request is made when set. LOLS is cleared by reset or by writing a logic 1 to LOLS when LOLS is set. Writing a logic 0 to LOLS has no effect. 0 FLL has not lost lock since LOLS was last cleared. 1 FLL has lost lock since LOLS was last cleared.
6 LOCK	Lock Status Indicates whether the FLL has acquired lock. Lock detection is disabled when FLL is disabled. If the lock status bit is set then changing the value of any of the following fields IREFS, RDIV[2:0], or, if in FEI or FBI modes, SCTRIM[7:0] will cause the lock status bit to clear and stay cleared until the FLL has reacquired lock. Stop mode entry will also cause the lock status bit to clear and stay cleared until the FLL has reacquired lock. 0 FLL is currently unlocked. 1 FLL is currently locked.
5 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
4 IREFST	Internal Reference Status Indicates the current source for the reference clock. This field does not update immediately after a write to ICS_C1[IREFS] due to internal synchronization between clock domains. 0 Source of reference clock is external clock. 1 Source of reference clock is internal clock.
3–2 CLKST	Clock Mode Status Indicates the current clock mode. This field doesn't update immediately after a write to ICS_C1[CLKS] due to internal synchronization between clock domains. 00 Output of FLL is selected. 01 FLL Bypassed, internal reference clock is selected. 10 FLL Bypassed, external reference clock is selected. 11 Reserved.
Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

19.4 Functional description

19.4.1 Operational modes

The seven states of the ICS are shown as a state diagram and are described below. The arrows indicate the allowed movements among the states.

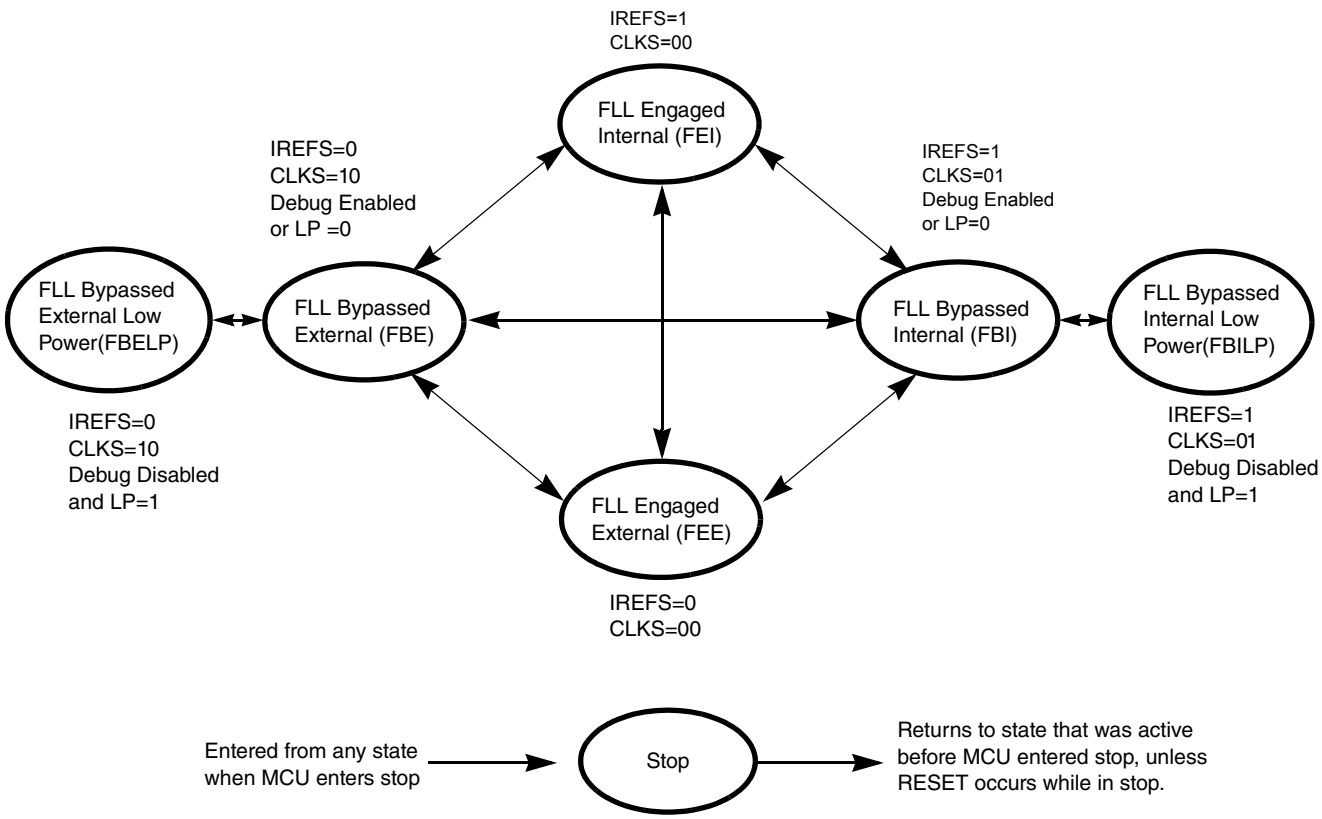


Figure 19-7. Clock switching modes

19.4.1.1 FLL engaged internal (FEI)

FLL engaged internal (FEI) is the default mode of operation and is entered when all the following conditions occur:

- 00b is written to ICS_C1[CLKS].
- 1b is written to ICS_C1[IREFS].

In FLL engaged internal mode, the ICSOUT clock is derived from the FLL clock, which is controlled by the internal reference clock. The FLL loop locks the frequency to 1280 times the internal reference frequency. The internal reference clock is enabled.

19.4.1.2 FLL engaged external (FEE)

The FLL engaged external (FEE) mode is entered when all the following conditions occur:

- 00b is written to ICS_C1[CLKS].
- 0b is written to ICS_C1[IREFS].
- ICS_C1[RDIV] and OSC_CR[RANGE] are written to divide external reference clock to be within the range of 31.25 kHz to 39.0625 kHz.

In FLL engaged external mode, the ICSOUT clock is derived from the FLL clock which is controlled by the external reference clock source.The FLL loop locks the frequency to 1280 times the external reference frequency, as selected by ICS_C1[RDIV] and OSC_CR[RANGE]. The external reference clock is enabled.

19.4.1.3 FLL bypassed internal (FBI)

The FLL bypassed internal (FBI) mode is entered when all the following conditions occur:

- 01b is written to ICS_C1[CLKS].
- 1b is written to ICS_C1[IREFS].
- BDM mode is active or ICS_C2[LP] bit is written to 0.

In FLL bypassed internal mode, the ICSOUT clock is derived from the internal reference clock. The FLL clock is controlled by the internal reference clock, and the FLL loop locks the FLL frequency to 1280 times the internal reference frequency. The internal reference clock is enabled.

19.4.1.4 FLL bypassed internal low power (FBILP)

The FLL bypassed internal low power (FBILP) mode is entered when all the following conditions occur:

- 01b is written to ICS_C1[CLKS].
- 1b is written to ICS_C1[IREFS].
- BDM mode is not active and ICS_C2[LP] bit is written to 1b.

In FLL bypassed internal low-power mode, the ICSOUT clock is derived from the internal reference clock and the FLL is disabled. The internal reference clock is enabled.

19.4.1.5 FLL bypassed external (FBE)

The FLL bypassed external (FBE) mode is entered when all the following conditions occur:

- 10b is written to ICS_C1[CLKS].
- 0b is written to ICS_C1[IREFS].
- ICS_C1[RDIV] and OSC_CR[RANGE] fields are written to divide external reference clock to be within the range of 31.25 kHz to 39.0625 kHz.
- BDM mode is active or 0b is written to C2[LP].

In FLL bypassed external mode, the ICSOUT clock is derived from the external reference clock source. The FLL clock is controlled by the external reference clock, and the FLL loop locks the FLL frequency to 1280 times the external reference frequency, as selected by ICS_C1[RDIV] and OSC_CR[RANGE], the external reference clock is enabled.

19.4.1.6 FLL bypassed external low power (FBELP)

The FLL bypassed external low-power (FBELP) mode is entered when all the following conditions occur:

- 10b is written to ICS_C1[CLKS].
- 0b is written to ICS_C1[IREFS].
- BDM mode is not active and ICS_C2[LP] bit is written to 1b.

In FLL bypassed external low-power mode, the ICSOUT clock is derived from the external reference clock source and the FLL is disabled. The external reference clock source is enabled.

19.4.1.7 Stop

NOTE

The DCO frequency changes from the pre-stop value to its reset value and the FLL need to re-acquire the lock before the frequency is stable. Timing sensitive operations must wait for the FLL acquisition time, $t_{Acquire}$, before executing.

Stop mode is entered whenever the MCU enters a STOP state. In this mode, all ICS clock signals are static except in the following cases:

ICSIRCLK will be active in Stop mode when all the following conditions occur:

- 1b is written to ICS_C1[IRCLKEN].
- 1b is written to ICS_C1[IREFSTEN].

19.4.2 Mode switching

ICS_C1[IREFS] can be changed at anytime, but the actual switch to the newly selected clock is shown by ICS_S[IREFST]. When switching between FLL engaged internal (FEI) and FLL engaged external (FEE) modes, the FLL begins locking again after the switch is completed.

ICS_C1[CLKS] can also be changed at anytime, but the actual switch to the newly selected clock is shown by ICS_S[CLKST]. If the newly selected clock is not available, the previous clock remains selected.

NOTE

When mode switching is from FEE, FBE or FBELP to FEI, it is suggested to wait IREFST switch completion, then change ICS_C1[CLKS].

19.4.3 Bus frequency divider

ICS_C2[BDIV] can be changed anytime and the actual switch to the new frequency occurs immediately.

19.4.4 Low-power field usage

The Low-Power (LP) field in the ICS_C2 register is provided to allow the FLL to be disabled and thus conserve power when it is not being used.

However, in some applications it may be desirable to allow the FLL to be enabled and to lock for maximum accuracy before switching to an FLL engaged mode. To do this, write 0b to ICS_C2[LP].

19.4.5 Internal reference clock

When ICS_C1[IRCLKEN] is set, the internal reference clock signal is presented as ICSIRCLK, which can be used as an additional clock source. To re-target the ICSIRCLK frequency, write a new value to the ICS_C3[SCTRIM] bits to trim the period of the internal reference clock:

- Writing a larger value slows down the ICSIRCLK frequency.
- Writing a smaller value to the ICS_C3 register speeds up the ICSIRCLK frequency.

The trim bits affect the ICSOUT frequency if the ICS is in FLL engaged internal (FEI), FLL bypassed internal (FBI), or FLL bypassed internal low power (FBILP) mode.

Until ICSIRCLK is trimmed, programming low bus divider (ICS_C2[BDIV]) factors may result in ICSOUT frequencies that exceed the maximum chip-level frequency and violate the chip-level clock timing specifications.

If ICS_C1[IREFSTEN] is set and 1b is written to ICS_C1[IRCLKEN], the internal reference clock keeps running during Stop mode in order to provide a fast recovery upon exiting Stop mode.

All MCU devices are factory programmed with a trim value in a reserved memory location. This value is uploaded to the ICS_C3 register and ICS_C4[SCFTRIM] during any reset initialization. For finer precision, trim the internal oscillator in the application and set ICS_C4[SCFTRIM] accordingly.

19.4.6 Fixed frequency clock

The ICS presents the divided FLL reference clock as ICSFFCLK for use as an additional clock source. ICSFFCLK frequency must be no more than 1/4 of the ICSOUT frequency to be valid. Because of this requirement, in bypass modes, the ICSFFCLK is valid only in bypass external modes (FBE and FBELP) for the following conditions of ICS_C2[BDIV], and divider factor of ICS_C1[RDIV] and OSC_CR[RANGE] values:

if OSC_CR[RANGE] is high,

- ICS_C2[BDIV] = 000, ICS_C2[RDIV] ≥ 010
- ICS_C2[BDIV] = 001 (divide by 2), ICS_C2[RDIV] ≥ 011
- ICS_C2[BDIV] = 010 (divide by 4), ICS_C2[RDIV] ≥ 100
- ICS_C2[BDIV] = 011 (divide by 8), ICS_C2[RDIV] ≥ 101

19.4.7 FLL lock and clock monitor

19.4.7.1 FLL clock lock

In FBE and FEE modes, the clock detector source uses the external reference as the reference. When FLL is detected from lock to unlock, ICS_S[LOLS] is set. An interrupt will be generated if ICS_C4[LOLIE] is set. ICS_S[LOLS] is cleared by reset or by writing a logic 1 to ICS_S[LOLS] when ICS_S[LOLS] is set. Writing a logic 0 to ICS_S[LOLS] has no effect.

In FBI and FEI modes, the lock detector source uses the internal reference as the reference. When FLL is detected from lock to unlock, ICS_S[LOLS] is set. An interrupt will be generated if ICS_C4[LOLIE] is set. ICS_S[LOLS] is cleared by reset or by writing a logic 1 to ICS_S[LOLS] when ICS_S[LOLS] is set. Writing a logic 0 to ICS_S[LOLS] has no effect.

In FBELP and FBILP modes, the FLL is not on so that lock detect function is not applicable.

19.4.7.2 External reference clock monitor

In FBE, FEE, or FBELP modes, if 1 is written to ICS_C4[CME], the clock monitor is enabled. If the external reference falls below a certain frequency, the MCU will reset. The will be set to indicate the error.

19.5 Initialization/application information

This section provides example code to give some basic direction to a user on how to initialize and configure the ICS module. The example software is implemented in C language.

19.5.1 Initializing FEI mode

The following code segment demonstrates setting ICS to FEI mode.

Example: 19.5.1.1 FEI mode initialization routine

```
/* the following code segment demonstrates setting the ICS to FEI mode using the factory
trim value. The resulting ICSOUT frequency is fint_ft*1280/BDIV. */
ICS_C2 = 0x20; // BDIV=divide by 2 - use default until clock dividers configured
ICS_C1 = 0x04; // internal reference clock as source to FLL
while ((ICS_S & ICS_S_LOCK_MASK) == 0); // wait for FLL to lock
SIM_CLKDIV = 0x01100000; // core clock = ICSOUT/1 and bus clock = core clock/2
ICS_C2 = 0x00; // BDIV=divide by 1 - allows max core and bus clock frequencies

/* the following code segment demonstrates setting the ICS to FEI mode using a custom trim
value provided by a programming tool. The resulting ICSOUT frequency is fint_t*1280/BDIV. */
ICS_C3 = *((uint8_t*) 0x03FF); // trim internal reference clock
ICS_C4 = *((uint8_t*) 0x03FE); // fine trim internal reference clock
ICS_C2 = 0x20; // BDIV=divide by 2 - use default until clock dividers configured
ICS_C1 = 0x04; // internal reference clock as source to FLL
while ((ICS_S & ICS_S_LOCK_MASK) == 0); // wait for FLL to lock
SIM_CLKDIV = 0x01100000; // core clock = ICSOUT/1 and bus clock = core clock/2
ICS_C2 = 0x00; // BDIV=divide by 1 - allows max core and bus clock frequencies
```

19.5.2 Initializing FBI mode

The following code segment demonstrates setting ICS to FBI mode.

Example: 19.5.2.1 FBI mode initialization routine

```
/* the following code segment demonstrates setting the ICS to FBI mode using the factory
trim value. The resulting ICSOUT frequency is fint_ft/BDIV. Note that the FLL will be
running at a frequency of fint_ft*1280/BDIV even though the FLL is bypassed. */
ICS_C2 = 0x20; // BDIV=divide by 2 - use default until clock dividers configured
ICS_C1 = 0x44; // internal reference clock as source for ICSOUT
while ((ICS_S & 0x0C) != 0x04); // wait until internal reference is selected
SIM_CLKDIV = 0x00000000; // core clock = ICSOUT/1; bus clock = core clock/1
ICS_C2 = 0x00; // BDIV=divide by 1 - allows max core and bus clock frequencies
```

19.5.3 Initializing FEE mode

The following code segment demonstrates setting ICS to FEE mode.

Example: 19.5.3.1 FEE mode initialization routine

```
/* the following code segment demonstrates setting the ICS to FEE mode generating a 40MHZ
core clock frequency using an external 8MHz crystal */
OSC_CR = 0x96; // high-range, high-gain oscillator selected
while ((OSC_CR & OSC_CR_OSCINIT_MASK) == 0); // wait until oscillator is ready
ICS_C2 = 0x20; // BDIV=divide by 2 - use default until clock dividers configured
ICS_C1 = 0x18; // 8MHz external reference clock/256 as source to FLL
while ((ICS_S & ICS_S_IREFST_MASK)); // wait for external source selected
while ((ICS_S & ICS_S_LOCK_MASK) == 0); // wait for FLL to lock
SIM_CLKDIV = 0x01100000; // core clock = ICSOUT/1 and bus clock = core clock/2
ICS_C2 = 0x00; // BDIV=divide by 1 - allows max core and bus clock frequencies
```

19.5.4 Initializing FBE mode

The following code segment demonstrates setting ICS to FBE mode.

Example: 19.5.4.1 FBE mode initialization routine

```
/* the following code segment demonstrates setting the ICS to FBE mode generating 20MHZ core
clock frequency using an external 20MHz crystal */
OSC_CR = 0x96; // high-range, high-gain oscillator selected
while ((OSC_CR & OSC_CR_OSCINIT_MASK) == 0); // wait until oscillator is ready
ICS_C2 = 0x20; // BDIV=divide by 2 - use default until clock dividers configured
ICS_C1 = 0xA0; // 20MHz external clock as ICSOUT source; FLL source = 20MHz/512
while ((ICS_S & ICS_S_IREFST_MASK)); // wait for external source selected
while ((ICS_S & 0x0C) != 0x08); // wait until FBE mode is selected
SIM_CLKDIV = 0x00000000; // core clock = ICSOUT/1 and bus clock = core clock/1
ICS_C2 = 0x00; // BDIV=divide by 1 - allows max core and bus clock frequencies
```

Chapter 20
Oscillator (OSC)

20.1 Introduction

20.1.1 Overview

The OSC module provides the clock source for the MCU. The OSC module, in conjunction with an external crystal or resonator, generates a clock for the MCU that can be used as reference clock or bus clock.

20.1.2 Features and modes

Key features of the OSC module are:

- Supports 32 kHz crystals (low range mode)
- Supports 4–24 MHz crystals and resonators (high range mode)
- Automatic gain control (AGC) to optimize power consumption in both frequency ranges using low-power mode (low gain mode)
- High gain option in both frequency ranges: 32 kHz, 4–24 MHz
- Voltage and frequency filtering to guarantee clock frequency and stability
- Supports to be enabled by ICS.

20.1.3 Block diagram

See the following figure for OSC module block diagram.

The OSC module uses a crystal or resonator to generate three filtered oscillator clock signals(XTL_CLK). The XTL_CLK can work in Stop mode since they come from Hard block which always has power.

The OSCOS decides whether OSC_OUT comes from internal oscillators(XTL_CLK) or directly from external clock driven on EXTAL pin. The OSCOS signal allows the XTAL pad to be used as I/O or test clock.

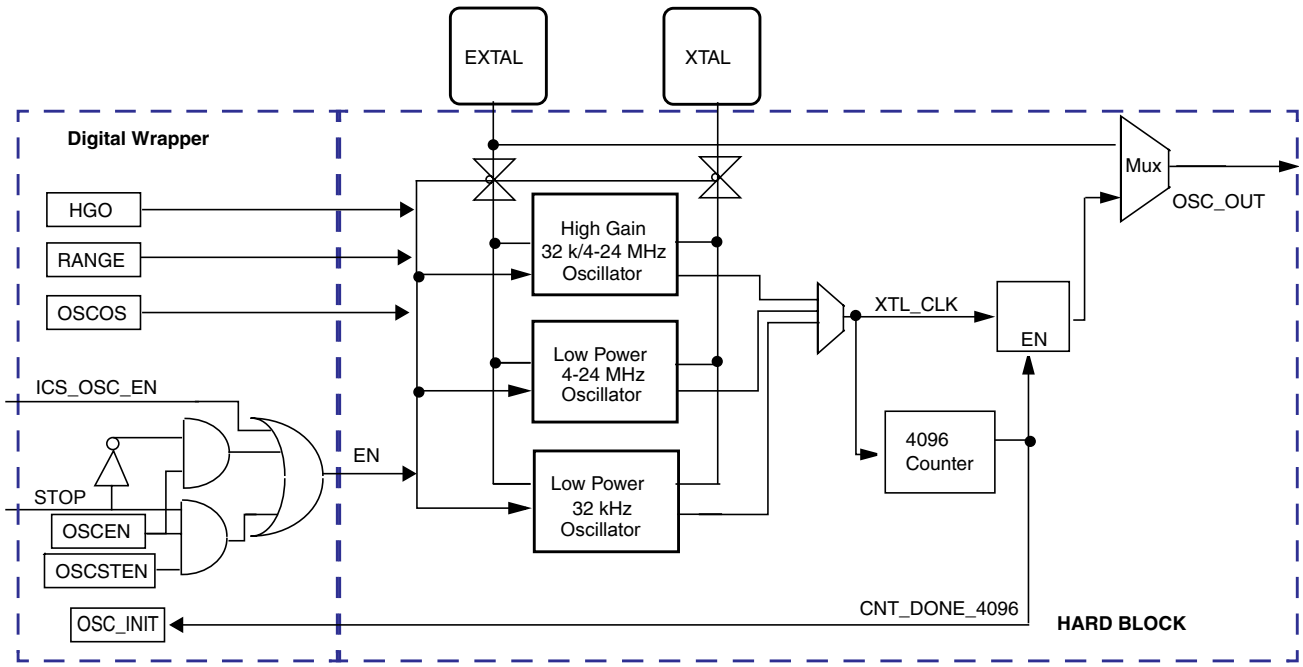


Figure 20-1. OSC module block diagram

20.2 Signal description

The following table shows the user-accessible signals available for the OSC module. See the chip-level specification to find out which signals are actually connected to external pins.

Table 20-1. OSC signal descriptions

Signal	Description	I/O
EXTAL	External clock/oscillator input	Analog input
XTAL	Oscillator output	Analog output

20.3 External crystal / resonator connections

The connections for a crystal/resonator frequency reference are shown in Figure 20-2 and Figure 20-3. When using low-frequency, low-power mode, the only external component is the crystal or resonator itself. In the other oscillator modes, load capacitors (C_x, C_y) and feedback resistor (R_F) are required. In addition, a series resistor (R_S) may be used in high-gain modes. Recommended component values are listed in the data sheet.

Table 20-2. External crystal/resonator connections

Oscillator mode	Connections
Low frequency, high gain	Connection2
Low frequency, low-power	Connection1
High frequency, high gain (4–24 MHz)	Connection2
High frequency, low-power (4–24 MHz)	Connection2

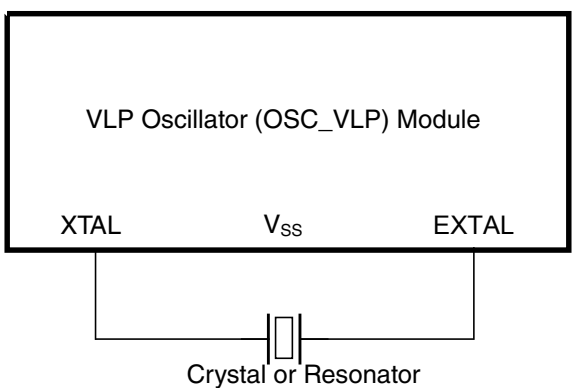


Figure 20-2. Crystal/resonator connections - connection 1

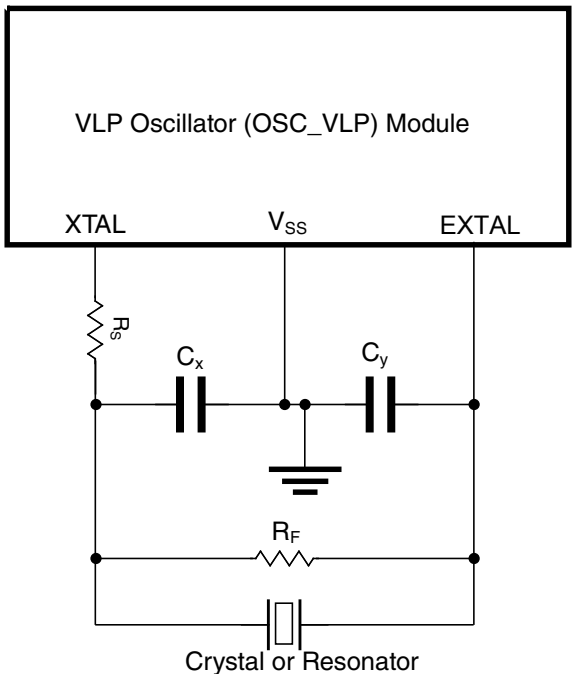


Figure 20-3. Crystal/resonator connections - connection 2

20.4 External clock connections

In external clock mode (OSC_CR[OSCOS] = 0), the pins can be connected as shown in the following figure.

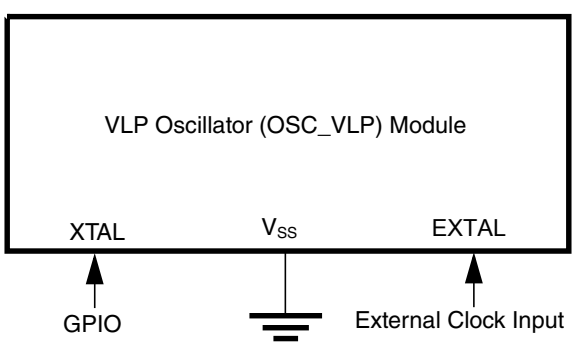


Figure 20-4. External clock connections

20.5 Memory map and register descriptions

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	20.5.1/239

20.5.1 OSC Control Register (OSC_CR)

Address: 4006_5000h base + 0h offset = 4006_5000h

Bit	7	6	5	4	3	2	1	0
Read	OSCEN	0	OSCSTEN	OSCOS	0	RANGE	HGO	OSCINIT
Write								
Reset	0	0	0	0	0	0	0	0

OSC_CR field descriptions

Field	Description
7 OSCEN	OSC Enable Enables the OSC module. The OSC module can also be enabled by the ICS module. 0 OSC module is disabled. 1 OSC module is enabled.
6 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
5 OSCSTEN	OSC Enable in Stop mode Controls whether or not the OSC clock remains enabled when MCU enters Stop mode and OSCEN is set. OSCSTEN has no effect if ICS requests OSC enable. 0 OSC clock is disabled in Stop mode. 1 OSC clock stays enabled in Stop mode.
4 OSCOS	OSC Output Select Selects the output clock of the OSC module. 0 External clock source from EXTAL pin is selected. 1 Oscillator clock source is selected.
3 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
2 RANGE	Frequency Range Select Selects the frequency range for the OSC module. This bit must be configured before the OSC is enabled and DO NOT change it after the OSC is enabled

Table continues on the next page...

OSC_CR field descriptions (continued)

Field	Description
	0 Low frequency range of 32 kHz. 1 High frequency range of 4–24 MHz.
1 HGO	High Gain Oscillator Select Controls the OSC mode of operation. 0 Low-power mode 1 High-gain mode
0 OSCINIT	OSC Initialization This field is set after the initialization cycles of oscillator are completed. 0 Oscillator initialization is not complete. 1 Oscillator initialization is completed.

20.6 Functional description

20.6.1 OSC module states

There are three states of the OSC module. A state diagram is shown in Figure 20-6. The states and the transitions among each other are described in this section.

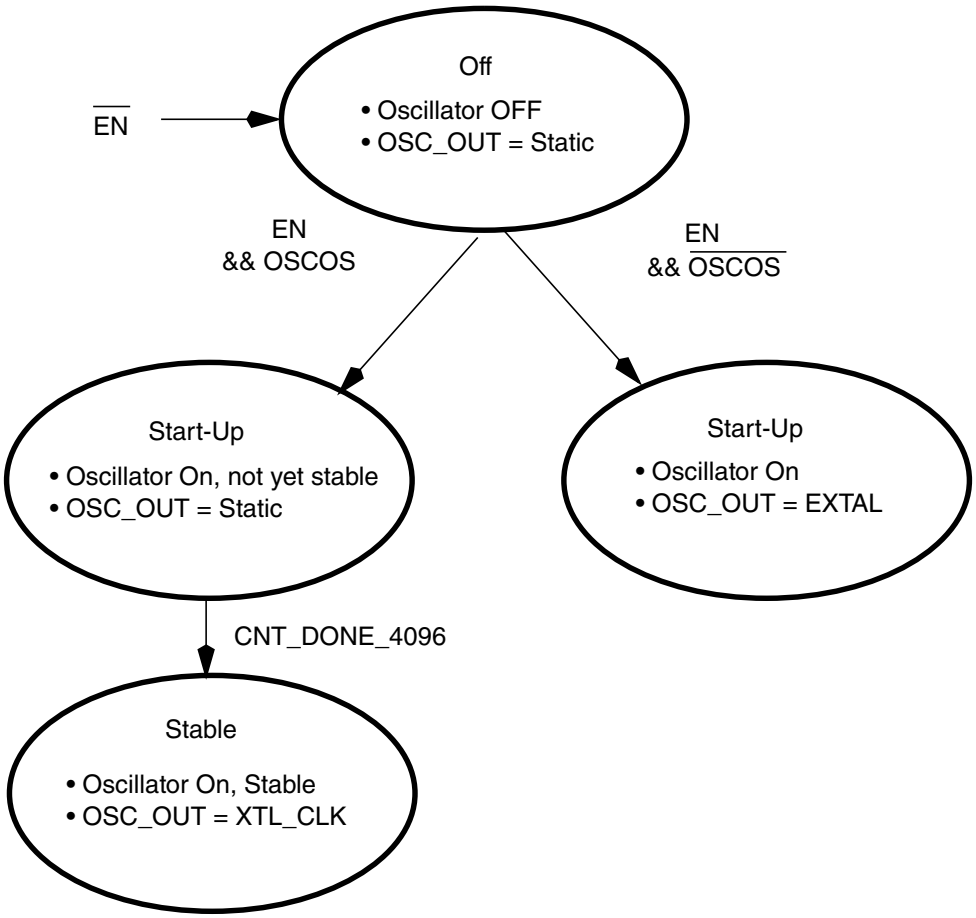


Figure 20-6. OSC module state diagram

EN is decided by OSC_CR[OSCEN], Stop, OSC_CR[OSCSTEN], and external request (ICS_OSC_EN). See the following table for details.

Table 20-5. EN status

EN	ICS_OSC_EN	OSC_CR[OSCEN]	OSC_CR[OSCSTEN]	Stop
1	1	-	-	-
1	0	1	-	0
1	0	1	1	1
0	0	1	0	1
0	0	0	-	-

20.6.1.1 Off

The off state is entered whenever the EN signal is negated. Upon entering this state, XTL_CLK and OSC_OUT is static. 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.

20.6.1.2 Oscillator startup

The oscillator startup state is entered whenever the oscillator is first enabled (EN transitions high) and OSC_CR[OSCOS] is high. 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 has seen 4096 cycles of XTL_CLK, the oscillator is considered stable and XTL_CLK is passed to the output clock OSC_OUT.

20.6.1.3 Oscillator stable

The oscillator stable state is entered whenever the oscillator is enabled (EN is high), OSC_CR[OSCOS] is high, and the counter has seen 4096 cycles of XTL_CLK (CNT_DONE_4096 is high). In this state, the OSC module is producing a stable output clock on OSC_OUT. Its frequency is determined by the external components being used.

20.6.1.4 External clock mode

The external clock state is entered when the oscillator is enabled(EN is high) and OSC_CR[OSCOS] is low. In this state, the OSC module is set up to buffer (with hysteresis) a clock from EXTAL onto the OSC_OUT. Its frequency is determined by the external clock being supplied.

20.6.2 OSC module modes

The oscillator is a Pierce-type oscillator that supports external crystals or resonators operating over the frequency ranges shown in the following table. These modes assume EN = 1, OSC_CR[OSCOS] = 1.

Table 20-6. Oscillator modes

RANGE	HGO	Mode	Frequency range
0	1	Low-frequency, high-gain	f_{lo}(min) up to f_{lo}(max)
0	0	Low-frequency, low-power (VLP)	
1	1	High-frequency mode1, high-gain	f_{hi}(min) up to f_{hi}(max)
1	0	High-frequency mode1, low-power	

20.6.2.1 Low-frequency, high-gain mode

In low-frequency, high-gain mode (OSC_CR[RANGE] = 0, OSC_CR[HGO] = 1) 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.

20.6.2.2 Low-frequency, low-power mode

In low-frequency, low-power mode (OSC_CR[RANGE] = 0, OSC_CR[HGO] = 0), 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.

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 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 which biases EXTAL.

20.6.2.3 High-frequency, high-gain mode

In high-frequency, high-gain Mode (OSC_CR[RANGE] = 1, OSC_CR[HGO] = 1), 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.

20.6.2.4 High-frequency, low-power mode

In high-frequency, low-power mode (OSC_CR[RANGE] = 1, OSC_CR[HGO] = 0) 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.

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.

20.6.3 Counter

The oscillator output clock (OSC_OUT) is gated off until the counter has detected 4096 cycles of its input clock (XTL_CLK). Once 4096 cycles are complete, the counter passes XTL_CLK onto OSC_OUT. This counting timeout is used to guarantee output clock stability.

20.6.4 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 can be used for I/O or test clock purposes as long as the specifications listed in the data sheet are met.

Chapter 21
Interrupt (IRQ)

21.1 Chip-specific IRQ information

21.1.1 IRQ Overview

This device has one IRQ modules to support external pin interrupt request.

21.1.2 IRQ assignments

The IRQ is by default assigned to pin PTB0.

The IRQ could also be assigned to other on-chip signal through the SBAR.

21.2 Introduction

The external interrupt (IRQ) module provides a maskable interrupt input.

21.3 Features

Features of the IRQ module include:

- A dedicated external interrupt pin IRQ
- IRQ Interrupt Control bits
- Programmable edge-only or edge and level interrupt sensitivity

- Automatic interrupt acknowledge
- Internal pullup device

A low level applied to the external interrupt request (IRQ) pin can latch a CPU interrupt request. The following figure shows the structure of the IRQ module:

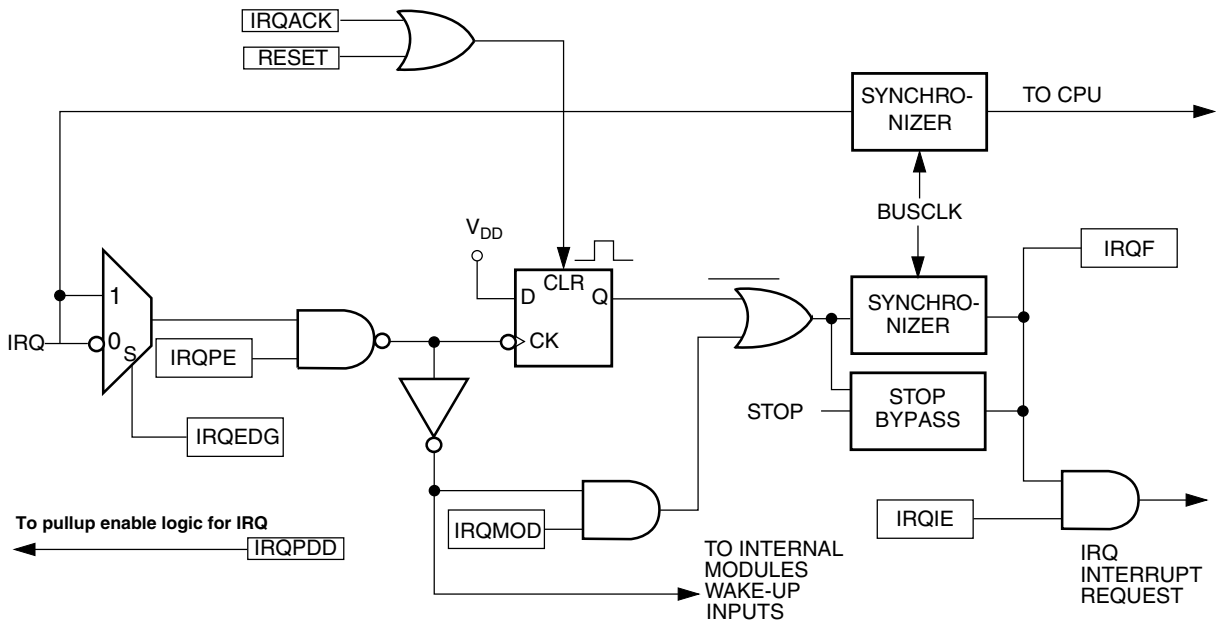


Figure 21-1. IRQ module block diagram

External interrupts are managed by the IRQSC status and control register. When the IRQ function is enabled, synchronous logic monitors the pin for edge-only or edge-and-level events. When the MCU is in Stop mode and system clocks are shut down, a separate asynchronous path is used so that the IRQ, if enabled, can wake the MCU.

21.3.1 Pin configuration options

The IRQ Pin Enable (IRQSC[IRQPE]) control field must be 1 for the IRQ pin to act as the IRQ input. The user can choose the polarity of edges or levels detected (IRQEDG), whether the pin detects edges-only or edges and levels (IRQMOD), or whether an event causes an interrupt or only sets the IRQSC[IRQF] flag, which can be polled by software.

When enabled, the IRQ pin defaults to use an internal pullup device (IRQSC[IRQPDD] = 0). If the user uses an external pullup or pulldown, the IRQSC[IRQPDD] can be written to a 1 to turn off the internal device.

BIH and BIL instructions may be used to detect the level on the IRQ pin when it is configured to act as the IRQ input.

Note

This pin does not contain a clamp diode to V_{DD} and must not be driven above V_{DD}. The voltage measured on the internally pullup IRQ pin may be as low as V_{DD} – 0.7 V. The internal gates connected to this pin are pulled all the way to V_{DD}.

When enabling the IRQ pin for use, IRQSC[IRQF] will be set, and must be cleared prior to enabling the interrupt. When configuring the pin for falling edge and level sensitivity in a 3 V system, it is necessary to wait at least cycles between clearing the flag and enabling the interrupt.

21.3.2 Edge and level sensitivity

The IRQSC[IRQMOD] control field reconfigures the detection logic so that it can detect edge events and pin levels. In this detection mode, IRQSC[IRQF] status flag is set when an edge is detected, if the IRQ pin changes from the deasserted to the asserted level, but the flag is continuously set and cannot be cleared as long as the IRQ pin remains at the asserted level.

21.4 Interrupt pin request register

IRQ memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4003_1000	Interrupt Pin Request Status and Control Register (IRQ_SC)	8	R/W	00h	21.4.1/247

21.4.1 Interrupt Pin Request Status and Control Register (IRQ_SC)

This direct page register includes status and control bits, which are used to configure the IRQ function, report status, and acknowledge IRQ events.

Address: 4003_1000h base + 0h offset = 4003_1000h

Bit	7	6	5	4	3	2	1	0
Read	0	IRQPDD	IRQEDG	IRQPE	IRQF	0	IRQIE	IRQMOD
Write						IRQACK		
Reset	0	0	0	0	0	0	0	0

IRQ_SC field descriptions

Field	Description
7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
6 IRQPDD	Interrupt Request (IRQ) Pull Device Disable This read/write control bit is used to disable the internal pullup device when the IRQ pin is enabled (IRQPE = 1) allowing for an external device to be used. 0 IRQ pull device enabled if IRQPE = 1. 1 IRQ pull device disabled if IRQPE = 1.
5 IRQEDG	Interrupt Request (IRQ) Edge Select This read/write control field is used to select the polarity of edges or levels on the IRQ pin that cause IRQF to be set. The IRQMOD control field determines whether the IRQ pin is sensitive to both edges and levels or only edges. When the IRQ pin is enabled as the IRQ input and is configured to detect rising edges, the optional pullup resistor is disabled. 0 IRQ is falling-edge or falling-edge/low-level sensitive. 1 IRQ is rising-edge or rising-edge/high-level sensitive.
4 IRQPE	IRQ Pin Enable This read/write control field enables the IRQ pin function. When this field is set, the IRQ pin can be used as an interrupt request. 0 IRQ pin function is disabled. 1 IRQ pin function is enabled.
3 IRQF	IRQ Flag This read-only status field indicates when an interrupt request event has occurred. 0 No IRQ request 1 IRQ event is detected.
2 IRQACK	IRQ Acknowledge This write-only field is used to acknowledge interrupt request events (write 1 to clear IRQF). Writing 0 has no meaning or effect. Reads always return 0. If edge-and-level detection is selected (IRQMOD = 1), IRQF cannot be cleared while the IRQ pin remains at its asserted level.
1 IRQIE	IRQ Interrupt Enable This read/write control field determines whether IRQ events generate an interrupt request. 0 Interrupt request when IRQF set is disabled (use polling). 1 Interrupt requested whenever IRQF = 1.
0 IRQMOD	IRQ Detection Mode This read/write control field selects either edge-only detection or edge-and-level detection. 0 IRQ event is detected only on falling/rising edges. 1 IRQ event is detected on falling/rising edges and low/high levels.

Chapter 22
Analog-to-Digital Converter (ADC)

22.1 Chip-specific ADC information

22.1.1 ADC instantiation information

This device contains one 12-bit successive approximation ADC with up to 4 external channels and 8 internal channels.

The ADC supports both software and hardware conversion flow control events (Trigger, Restart, LoadOK and Seq_Abort). These ADC hardware events interfaces sources are selectable from the SIM trigger crossbar.

See TBAR for details on hardware flow control events sources.

22.1.2 ADC0 connections/channel assignment

The ADC channel assignments for the device are shown in following table. The internal resistor divider is required for pin Vrect, Vout and Vad_in before they are connected to ADC input channels.

Table 22-1. ADC internal channel assigement

CHSEL[5:0]	Channel	Input
001000	Internal_0	ADC0 temperature sensor
001001	Internal_1	PMC temperature sensor / bandgap
001010	Internal_2	PMC VDDF
001011	Internal_3	1/8 scaled Vrect
001100	Internal_4	7/10 scaled Vout
001101	Internal_5	1/5 scaled Vad_in
001110	Internal_6	PGA VOP
001111	Internal_7	PGA VON

Table 22-2. ADC external channel assigement

CHSEL[5:0]	Channel	Input
010000	AN0	PTB1/ADPC[0]
010001	AN1	PTB2/ADPC[1]
010010	AN2	PTB3/ADPC[2]
010011	AN3	PTB4/ADPC[3]

22.1.3 ADC analog supply and reference connections

ADC VRH_0 is connected to VREFH.

ADC VRL_0 is connected to VSSA.

ADC VRH_1 is connected to VDDA.

ADC VRL_1 is connected to VSSA.

NOTE

VREFH can be generated by PMC, and an external 10 μF capacitor is required to be connected VREFH pin. The PMC VREFH is not available in STOP mode. When PMC VREFH is disabled (PMC_CTRL[VREFDN_EN] being cleared), VREFH can be external reference input from VREFH pin.

22.1.4 Temperature sensor

The ADC internal channel 2 is connected to the PMC temperature sensor output. As the PMC integrates the on-chip LDO which supply power to all on-chip components, software could monitor the chip thermal condition by sampling this sensor by ADC.

Besides, the ADC module integrates an on-chip temperature sensor, which is connected to ADC internal channel 0.

22.1.5 Summary of Conversion Flow Control Bit Scenarios

Table 22-3. Summary of Conversion Flow Control Bit Scenarios

RSTA	TRIG	SEQA	LDOK	Conversion Flow Control Mode	Conversion Flow Control Scenario
0	0	0	0	Both Modes	Valid

Table continues on the next page...

Table 22-3. Summary of Conversion Flow Control Bit Scenarios (continued)

RSTA	TRIG	SEQA	LDOK	Conversion Flow Control Mode	Conversion Flow Control Scenario
0	0	0	1	Both Modes	Can Not Occur
0	0	1	0	Both Modes	Valid[1]
0	0	1	1	Both Modes	Can Not Occur
0	1	0	0	Both Modes	Valid[2]
0	1	0	1	Both Modes	Can Not Occur
0	1	1	0	Both Modes	Can Not Occur
0	1	1	1	Both Modes	Can Not Occur
1	0	0	0	Both Modes	Valid[3]
1	0	0	1	Both Modes	Valid[4,3]
1	0	1	0	Both Modes	Valid[1,3,5]
1	0	1	1	Both Modes	Valid[1,3,4,5]
1	1	0	0	Restart Mode	Error flag TRIG_EIF set
				Trigger Mode	Valid[2,3,6]
1	1	0	1	Restart Mode	Error flag TRIG_EIF set
				Trigger Mode	Valid[2,3,4,6]
1	1	1	0	Restart Mode	Error flag TRIG_EIF set
				Trigger Mode	Valid[1,2,3,5,6]
1	1	1	1	Restart Mode	Error flag TRIG_EIF set
				Trigger Mode	Valid[1,2,3,4,5,6]

1. Abort any ongoing conversion, conversion sequence and CSL.
2. Start conversion sequence.
3. Load conversion command from top of CSL.
4. Swap CSL buffer.
5. Prevent RSTA_EIF and LDOK_EIF.
6. Bit TRIG set automatically in Trigger Mode

22.2 Introduction

The ADC is an n-channel multiplexed input successive approximation analog-to-digital converter. Refer to device electrical specifications for ADC parameters and accuracy.

The List Based Architecture (LBA) provides flexible conversion sequence definition as well as flexible oversampling. The order of channels to be converted can be freely defined. Also, multiple instantiations of the module can be triggered simultaneously (matching sampling point across multiple module instantiations).

There are register bits which control the conversion flow (please refer to the description of register ADC_FLWCTL).

The conversion flow control bits of register ADC_FLWCTL can be modified in two different ways:

- Via data bus accesses
- Via internal interface signals (Trigger, Restart, LoadOK, and Seq_Abort; see also Figure 22-2). Each interface signal is associated with one conversion flow control bit.

The ADC_FLWCTL register can be controlled via internal interface only or via data bus only or by both depending on the register access configuration bits ADC_CTL0[ACC_CFG].

The bits of register ADC_FLWCTL reflect the captured request and status of the internal interface signals (LoadOK, Trigger, Restart, and Seq_abort; see also Figure 22-2) if access configuration is set accordingly and indicate event progress (when an event is processed and when it is finished).

Conversion flow error situations are captured by corresponding interrupt flags in the ADC_EIF register.

There are two conversion flow control modes (Restart mode, Trigger mode). Each mode causes a certain behavior of the conversion flow control bits which can be selected according to the application needs.

Because internal components of the ADC are turned on/off with ADC_CTL0[ADC_EN], the ADC requires a recovery time period (t_{REC}) after ADC is enabled until the first conversion can be launched via a trigger.

When ADC_CTL0[ADC_EN] gets cleared (transition from 1b to 0b) any ongoing conversion sequence will be aborted and pending results, or the result of current conversion, gets discarded (not stored). The ADC cannot be re-enabled before any pending action or action in process is finished respectively aborted, which could take up to a maximum latency time of $t_{DISABLE}$.

22.3 Key features

This ADC has the following features:

- Programmer's model with list based architecture for conversion command and result value organization
- Selectable resolution of 8-bit, 10-bit, or 12-bit
- Channel select control for n external analog input channels
- Provides up to eight device internal channels
- Programmable sample time

- A sample buffer amplifier for channel sampling (improved performance in view to influence of channel input path resistance versus conversion accuracy)
- Left/right justified result data
- Individual selectable VRH_0/1 and VRL_0/1 inputs on a conversion command basis
- Special conversions for selected VRH_0/1, VRL_0/1, (VRL_0/1 + VRH_0/1) / 2
- 15 conversion interrupts with flexible interrupt organization per conversion result
- One dedicated interrupt for "End Of List" type commands
- Command sequence list (CSL) with a maximum number of 64 command entries
- Provides conversion sequence abort
- Restart from top of active command sequence list (CSL)
- The command sequence list and result value list are implemented in double buffered manner (two lists in parallel for each function)
- Conversion command (CSL) loading possible from system RAM or NVM
- Single conversion flow control register with software selectable access path
- Two conversion flow control modes optimized to different application use cases

22.4 Mode of operation

22.4.1 Conversion modes

This architecture provides single, multiple, or continuous conversion on a single channel or on multiple channels based on the command sequence list.

22.4.2 MCU operating modes

- MCU Stop mode

  Before issuing an MCU Stop mode request, the ADC must be idle (no conversion or conversion sequence or command sequence list ongoing).

  If a conversion, conversion sequence, or CSL is in progress when an MCU Stop mode request is issued, a sequence abort event occurs automatically and any ongoing conversion finish. After the sequence abort event finishes, if the ADC_CTL0[STR_SEQA] is set (ADC_CTL0[STR_SEQA]=1), then the conversion result is stored and the corresponding flags are set. If the ADC_CTL0[STR_SEQA] is cleared (ADC_CTL0[STR_SEQA]=0), then the conversion result is not stored and the corresponding flags are not set. The microcontroller then enters MCU Stop mode without ADC_IF[SEQAD_IF] being set.

Alternatively, the sequence abort event can be issued by software before an MCU Stop mode request. As soon as flag ADC_IF[SEQAD_IF] is set, the MCU Stop mode request can be is issued.

With the occurrence of the MCU Stop mode request until exit from Stop mode, all flow control signals (ADC_FLWCTL[RSTA], ADC_FLWCTL[SEQA], ADC_FLWCTL[LDOK], ADC_FLWCTL[TRIG]) are cleared.

After exiting MCU Stop mode, the following happens in the order given with expected event(s) depending on the conversion flow control mode:

- In ADC conversion flow control mode "Trigger Mode", a Restart event is expected to simultaneously set ADC_FLWCTL[TRIG] and ADC_FLWCTL[RSTA], causing the ADC to execute the Restart event (ADC_CIDX[CMD_IDX] and ADC_RIDX[RES_IDX] cleared) followed by the Trigger event. The Restart event can be generated automatically after exit from MCU Stop mode if ADC_CTL1[AUT_RSTA] is set and the chip does not use internal VREFH.
- In ADC conversion flow control mode "Restart Mode", a Restart event is expected to set bit ADC_FLWCTL[RSTA] only (ADC already aborted at MCU Stop mode entry, hence bit ADC_FLWCTL[SEQA] must not be set simultaneously) causing the ADC to execute the Restart event (ADC_CIDX[CMD_IDX] and ADC_RIDX[RES_IDX] cleared). The Restart event can be generated automatically after exit from MCU Stop mode if ADC_CTL1[AUT_RSTA] is set and the chip does not use internal VREFH.
- The RVL buffer select (ADC_STS[RVL_SEL]) is not changed if a CSL is in process at MCU Stop Mode request. Hence the same buffer will be used after exit from Stop Mode that was used when the Stop Mode request occurred.

- MCU Wait mode

Depending on the ADC Wait Mode configuration bit ADC_CTL0[SWAI], the ADC either continues conversion in MCU Wait Mode or freezes conversion at the next conversion boundary before MCU Wait Mode is entered.

ADC behavior for configuration ADC_CTL0[SWAI] =0b:

The ADC continues conversion during Wait Mode according to the conversion flow control sequence. It is assumed that the conversion flow control sequence is continued (conversion flow control bits ADC_FLWCTL[TRIG], ADC_FLWCTL[RSTA], ADC_FLWCTL[SEQA], and ADC_FLWCTL[LDOK] are serviced accordingly).

ADC behavior for configuration ADC_CTL0[SWAI] = 1b:

At MCU Wait Mode request the ADC should be idle (no conversion or conversion sequence or Command Sequence List ongoing).

If a conversion, conversion sequence, or CSL is in progress when an MCU Wait Mode request is issued, a Sequence Abort Event occurs automatically and any ongoing conversion finish. After the Sequence Abort Event finishes, if the ADC_CTL0[STR_SEQA] is set (ADC_CTL0[STR_SEQA]=1), then the conversion result is stored and the corresponding flags are set. If the ADC_CTL0[STR_SEQA] bit is cleared (ADC_CTL0[STR_SEQA]=0), then the conversion result is not stored and the corresponding flags are not set.

Alternatively the Sequence Abort Event can be issued by software before MCU Wait Mode request. As soon as flag ADC_IF[SEQAD_IF] is set, the MCU Wait Mode request can be issued.

With the occurrence of the MCU Wait Mode request until exit from Wait Mode all flow control signals (ADC_FLWCTL[TRIG], ADC_FLWCTL[RSTA], ADC_FLWCTL[SEQA], and ADC_FLWCTL[LDOK]) are cleared.

After exiting MCU Wait Mode, the following happens in the order given with expected event(s) depending on the conversion flow control mode:

- In ADC conversion flow control mode "Trigger Mode", a Restart Event is expected to occur. This simultaneously sets bit TRIG and RSTA causing the ADC to execute the Restart Event (ADC_CIDX[CMD_IDX] and ADC_RIDX[RES_IDX] cleared) followed by the Trigger Event. The Restart Event can be generated automatically after exit from MCU Wait Mode if bit ADC_CTL1[AUT_RSTA] is set.
- In ADC conversion flow control mode "Restart Mode", a Restart Event is expected to set ADC_CTL1[AUT_RSTA] only (ADC already aborted at MCU Wait Mode entry hence ADC_FLWCTL[SEQA]must not be set simultaneously) causing the ADC to execute the Restart Event (ADC_CIDX[CMD_IDX] and ADC_RIDX[RES_IDX] cleared). The Restart Event can be generated automatically after exit from MCU Wait Mode if ADC_CTL1[AUT_RSTA] is set.
- The RVL buffer select (ADC_STS[RVL_SEL]) is not changed if a CSL is in process at MCU Wait Mode request. Hence the same RVL buffer will be used after exit from Wait Mode that was used when Wait Mode request occurred.

NOTE

In principle, the MCU could stay in Wait Mode for a shorter period of time than the ADC needs to abort an ongoing conversion (range of µs). Therefore in case a Sequence Abort Event is issued automatically due to MCU Wait Mode request a following Restart Event after exit from MCU Wait Mode can not be executed before ADC has finished this Sequence Abort Event. The Restart Event

is detected but it is pending. This applies in case MCU Wait Mode is exited before ADC has finished the Sequence Abort Event and a Restart Event is issued immediately after exit from MCU Wait Mode. ADC_STS[READY] can be used by software to detect when the Restart Event can be issued without latency time in processing the event (see also the following figure)

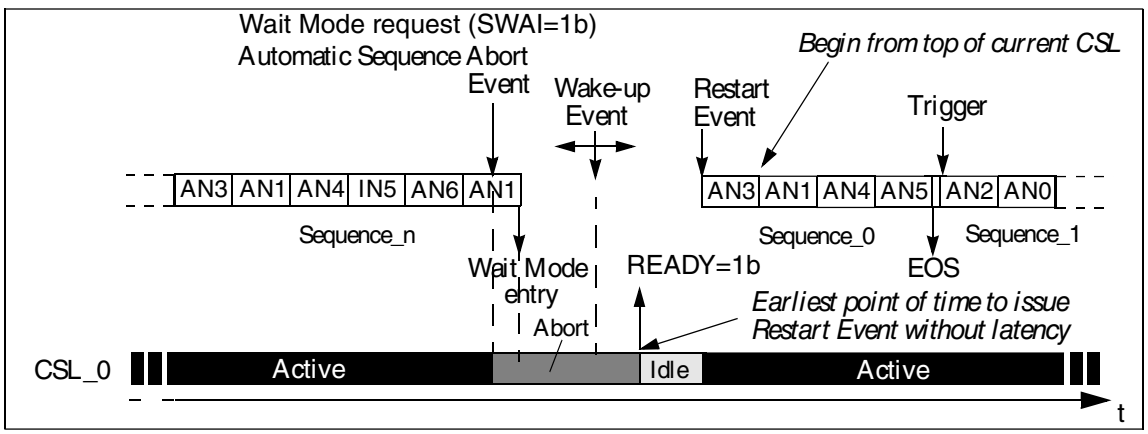


Figure 22-1. Conversion flow control diagram - Wait mode (ADC_CTL0[SWAI]=1b, ADC_CTL1[AUT_RSTA]=0b)

- MCU Freeze mode

  Depending on the ADC Freeze mode configuration bit ADC_CTL0[FRZ_MOD], the ADC either continues conversion in Freeze Mode or freezes conversion at next conversion boundary before the MCU Freeze mode is entered. After exit from MCU Freeze mode with previously frozen conversion sequence the ADC continues the conversion with the next conversion command and all ADC interrupt flags are unchanged during MCU Freeze mode.

22.4.3 Block diagram

The block diagram of ADC is as follows.

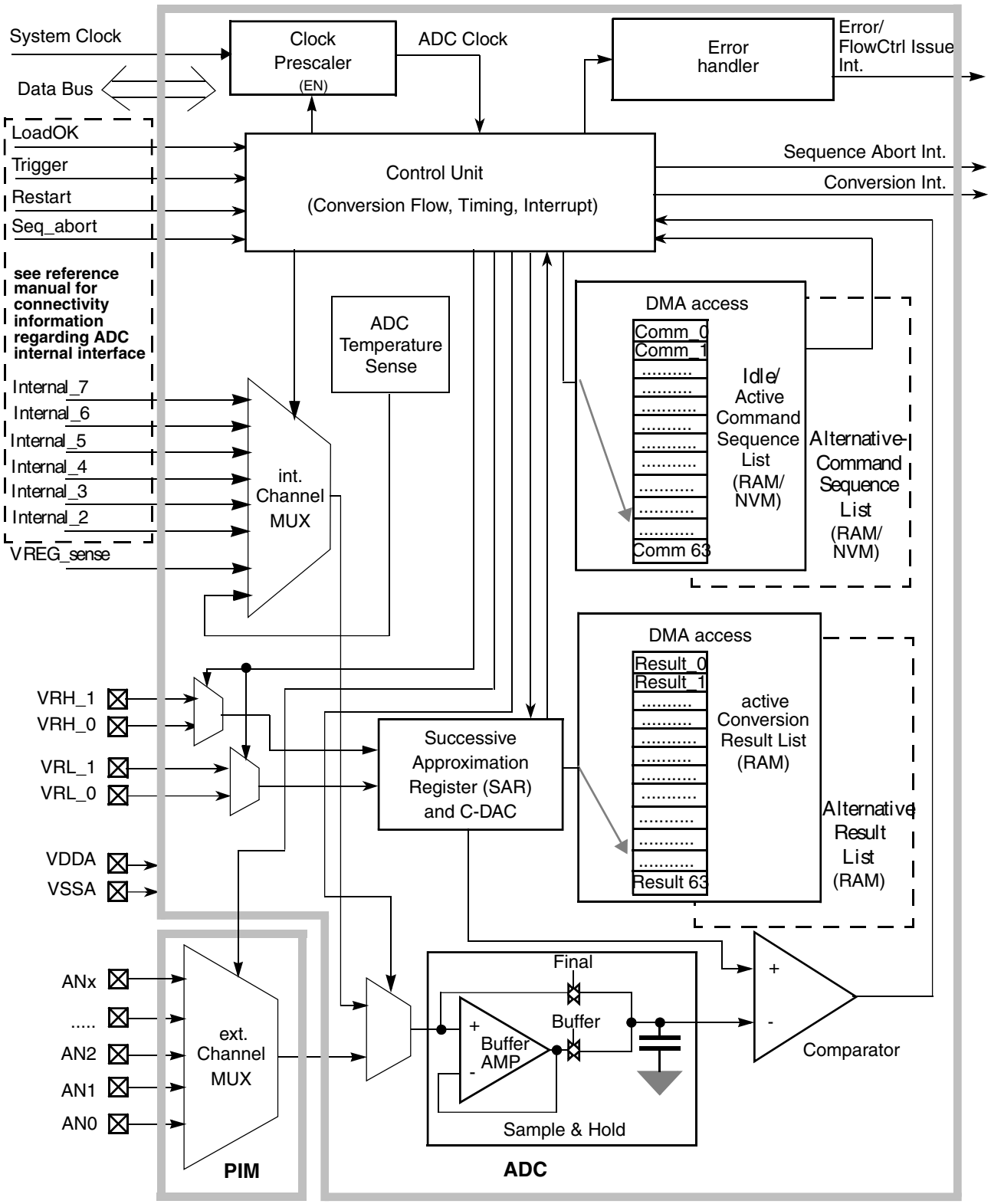


Figure 22-2. ADC block diagram

22.5 Signal description

This section lists all inputs to the ADC block.

Table 22-4. ADC signal descriptions

Signal	Description	I/O
ANx (x=n,...2,1,0)	Analog input channel	I
VRH_0	High reference voltage for a ADC conversion selectable on a conversion command basis	I
VRH_1	High reference voltage for a ADC conversion selectable on a conversion command basis	I
VRL_0	Low reference voltage for a ADC conversion selectable on a conversion command basis	I
VRL_1	Low reference voltage for a ADC conversion selectable on a conversion command basis	I
VDDA	Analog power supply	I
VSSA	Analog ground	I

22.6 Memory Map and Register Descriptions

ADC memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4003_B000	ADC Timing Register (ADC_TIM)	8	R/W	05h	22.6.1/259
4003_B001	ADC Status Register (ADC_STS)	8	R/W	08h	22.6.2/260
4003_B002	ADC Control Register 1 (ADC_CTL1)	8	R/W	00h	22.6.3/261
4003_B003	ADC Control Register 0 (ADC_CTL0)	8	R/W	00h	22.6.4/262
4003_B004	ADC Interrupt Enable Register (ADC_IE)	8	R/W	00h	22.6.5/264
4003_B005	ADC Error Interrupt Enable Register (ADC_EIE)	8	R/W	00h	22.6.6/265
4003_B006	ADC Conversion Flow Control Register (ADC_FLWCTL)	8	R/W	00h	22.6.7/266
4003_B007	ADC Format Register (ADC_FMT)	8	R/W	08h	22.6.8/269
4003_B008	ADC Conversion Interrupt Enable Register 1 (ADC_CONIE1)	8	R/W	00h	22.6.9/270
4003_B009	ADC Conversion Interrupt Enable Register 0 (ADC_CONIE0)	8	R/W	00h	22.6.10/270
4003_B00A	ADC Interrupt Flag Register (ADC_IF)	8	R/W	00h	22.6.11/271
4003_B00B	ADC Error Interrupt Flag Register (ADC_EIF)	8	R/W	00h	22.6.12/272
4003_B00C	ADC Intermediate Result Information Register 1 (ADC_IMDRI1)	8	R	00h	22.6.13/274

Table continues on the next page...

ADC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4003_B00D	ADC Intermediate Result Information Register 0 (ADC_IMDRI0)	8	R	00h	22.6.14/275
4003_B00E	ADC Conversion Interrupt Flag Register 1 (ADC_CONIF1)	8	R/W	00h	22.6.15/276
4003_B00F	ADC Conversion Interrupt Flag Register 0 (ADC_CONIF0)	8	R/W	00h	22.6.16/276
4003_B013	ADC End Of List Result Information Register (ADC_EOLRI)	8	R	00h	22.6.17/277
4003_B015	ADC Command Register 2 (ADC_CMD2)	8	R/W	00h	22.6.18/277
4003_B016	ADC Command Register 1 (ADC_CMD1)	8	R/W	00h	22.6.19/279
4003_B017	ADC Command Register 0 (ADC_CMD0)	8	R/W	00h	22.6.20/280
4003_B01C	ADC Command Base Pointer Register 2 (ADC_CBP2)	8	R/W	00h	22.6.21/281
4003_B01D	ADC Command Base Pointer Register 1 (ADC_CBP1)	8	R/W	00h	22.6.22/282
4003_B01E	ADC Command Base Pointer Register 0 (ADC_CBP0)	8	R/W	00h	22.6.23/282
4003_B01F	ADC Command Index Register (ADC_CIDX)	8	R	00h	22.6.24/283
4003_B020	ADC Result Base Pointer Register 2 (ADC_RBP2)	8	R/W	00h	22.6.25/283
4003_B021	ADC Result Base Pointer Register 1 (ADC_RBP1)	8	R/W	00h	22.6.26/284
4003_B023	ADC Result Index Register (ADC_RIDX)	8	R	00h	22.6.27/284
4003_B026	ADC Command and Result Offset Register 1 (ADC_CROFF1)	8	R/W	00h	22.6.28/285
4003_B027	ADC Command and Result Offset Register 0 (ADC_CROFF0)	8	R	00h	22.6.29/286

22.6.1 ADC Timing Register (ADC_TIM)

NOTE

These bits are writable if bit ADC_EN is clear or bit SMOD_ACC is set.

Address: 4003_B000h base + 0h offset = 4003_B000h

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

ADC_TIM field descriptions

Field	Description
7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
PRS	ADC Clock Prescaler

Table continues on the next page...

ADC_TIM field descriptions (continued)

Field	Description
	These 7 bits are the binary prescaler value PRS. The ADC conversion clock frequency is calculated as follows $f_{ATDCLK}=f_{BUS}/(2x(PRS+1))$ See Device Specification for allowed frequency range of f_{ATDCLK}.

22.6.2 ADC Status Register (ADC_STS)

It is important to note that if flag DBECC_ERR is set, the ADC ceases operation. An ADC Soft-Reset must be issued to make the ADC operational again. An ADC Soft-Reset clears bits CSL_SEL and RVL_SEL.

NOTE

Bits CSL_SEL and RVL_SEL are writable at anytime if bit ADC_EN is clear or bit SMOD_ACC is set.

Address: 4003_B000h base + 1h offset = 4003_B001h

Bit	7	6	5	4	3	2	1	0
Read	CSL_SEL	RVL_SEL	0	0	READY	0		
Write								
Reset	0	0	0	0	1	0	0	0

ADC_STS field descriptions

Field	Description
7 CSL_SEL	Command Sequence List Select bit This bit controls and indicates which ADC command list is active. This bit can only be written if ADC_EN bit is clear. This bit toggles in CSL double buffer mode when no conversion or conversion sequence is ongoing and bit LDOK is set and bit RSTA is set. In CSL single buffer mode, this bit is forced to 0b by bit CSL_BMOD. 0 ADC Command List 0 is active. 1 ADC Command List 1 is active.
6 RVL_SEL	Result Value List Select Bit This bit controls and indicates which ADC result list is active. This bit can only be written if bit ADC_EN is clear. After storage of the initial result value list, this bit toggles in RVL double buffer mode whenever the conversion result of the first conversion of the current CSL is stored or a CSL got aborted. In RVL single buffer mode, this bit is forced to 0b by bit RVL_BMOD. 0 ADC Result List 0 is active. 1 ADC Result List 1 is active.
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...

ADC_STS field descriptions (continued)

Field	Description
3 READY	Ready For Restart Event Flag This flag indicates that ADC is in its idle state and ready for a restart event. It can be used to verify after exit from Wait mode if a restart event can be issued and processed immediately without any latency time due to an ongoing sequence abort event after exit from MCU Wait mode. 0 ADC not in idle state. 1 ADC is in idle state.
Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

22.6.3 ADC Control Register 1 (ADC_CTL1)

NOTE

Bit CSL_BMOD and RVL_BMOD are writable if bit ADC_EN is clear or bit SMOD_ACC is set

Bit SMOD_ACC is writable only in MCU Special mode

Bit AUT_RSTA is writable anytime

Address: 4003_B000h base + 2h offset = 4003_B002h

Bit	7	6	5	4	3	2	1	0
Read	CSL_BMOD	RVL_BMOD	SMOD_ACC	AUT_RSTA	0			
Write								
Reset	0	0	0	0	0	0	0	0

ADC_CTL1 field descriptions

Field	Description
7 CSL_BMOD	CSL Buffer Mode Select Bit This bit defines the CSL buffer mode. This bit is writable only if ADC_EN is clear. 0 CSL single buffer mode. 1 CSL double buffer mode.
6 RVL_BMOD	RVL Buffer Mode Select Bit This bit defines the RVL buffer mode. 0 RVL single buffer mode. 1 RVL double buffer mode.
5 SMOD_ACC	Special Mode Access Control Bit

Table continues on the next page...

ADC_CTL1 field descriptions (continued)

Field	Description
	This bit controls register access rights in MCU Debug mode. This bit is automatically cleared when leaving MCU Debug mode. **NOTE:** When this bit is set, also the ADC_CMD0 and ADC_CMD1 registers are writeable via the data bus to allow modification of the current command for debugging purpose. But this is possible only if the current command is not already processed (conversion not started). Care must be taken when modifying ADC registers while bit SMOD_ACC is set to not corrupt a possible ongoing conversion. 0 Normal user access - Register write restrictions exist as specified for each bit. 1 Special access - Register write restrictions are lifted.
4 AUT_RSTA	Automatic Restart Event after exit from MCU Stop and Wait Mode (SWAI set) This bit controls whether a restart event is automatically generated after exit from MCU Stop mode or Wait mode with bit SWAI set. It can be configured for ADC conversion flow control mode "Trigger mode" and "Restart mode" (anytime during application runtime). 0 No automatic restart event after exit from MCU Stop mode. 1 Automatic restart event occurs after exit from MCU Stop mode.
Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

22.6.4 ADC Control Register 0 (ADC_CTL0)

NOTE

Bits ADC_EN, ADC_SR, FRZ_MOD and SWAI are writable anytime

Bits MOD_CFG, STR_SEQA and ACC_CFG[1:0] are writable if bit ADC_EN is clear or bit SMOD_ACC is set

Each conversion flow control bit (SEQA, RSTA, TRIG, LDOK) must be controlled by software or internal interface according to the requirements.

Address: 4003_B000h base + 3h offset = 4003_B003h

Bit	7	6	5	4	3	2	1	0
Read Write	ADC_EN	ADC_SR	FRZ_MOD	SWAI	ACC_CFG		STR_SEQA	MOD_CFG
Reset	0	0	0	0	0	0	0	0

ADC_CTL0 field descriptions

Field	Description
7 ADC_EN	ADC Enable Bit This bit enables the ADC (e.g. sample buffer amplifier etc.) and controls accessibility of ADC register bits. When this bit gets cleared, any ongoing conversion sequence will be aborted and pending results or the result of current conversion gets discarded (not stored). The ADC cannot be re-enabled before any pending action or action in process is finished or aborted, which could take up to a maximum latency time of $t_{DISABLE}$. Because internal components of the ADC are turned on/off with this bit, the ADC requires a recovery time period (t_{REC}) after ADC is enabled until the first conversion can be launched via a trigger. 0 ADC disabled. 1 ADC enabled.
6 ADC_SR	ADC Soft-Reset This bit causes an ADC Soft-Reset if set after a severe error occurred (see list of severe errors in ADC_EIF that causes the ADC to cease operation). It clears all overrun flags and error flags and forces the ADC state machine to its idle state. It also clears the ADC_CIDX register, the ADC_RIDX register, and the ADC_STS[CSL_SEL] and ADC_STS[RVL_SEL] bits (to be ready for a new control sequence to load new command and start execution again from top of selected CSL). A severe error occurs if an error flag is set which cause the ADC to cease operation. In order to make the ADC operational again, an ADC Soft-Reset must be issued. If this bit is set, it can not be cleared by writing any value. It is cleared only by ADC hardware after the Soft-Reset has been executed. 0 No ADC Soft-Reset issued. 1 Issue ADC Soft-Reset.
5 FRZ_MOD	Freeze Mode Configuration This bit influences conversion flow during Freeze mode. 0 ADC continues conversion in Freeze mode. 1 ADC freezes the conversion at next conversion boundary at Freeze mode entry.
4 SWAI	Wait Mode Configuration This bit influences conversion flow during Wait mode. 0 ADC continues conversion in Wait mode. 1 ADC halts the conversion at next conversion boundary at Wait mode entry.
3–2 ACC_CFG	ADC_FLWCTL Register Access Configuration These bits define if the register ADC_FLWCTL is controlled via internal interface only or data bus only or both. 00 None of the access paths is enabled (default / reset configuration) 01 Single access mode - internal interface(ADC_FLWCTL access via internal interface only) 10 Single access mode - data bus(ADC_FLWCTL access via data bus only) 11 Dual access mode(ADC_FLWCTL register access via internal interface and data bus)
1 STR_SEQA	Control Of Conversion Result Storage and RSTAR_EIF flag setting at Sequence Abort or Restart Event This bit controls conversion result storage and RSTAR_EIF flag setting when a Sequence Abort Event or Restart Event occurs as follows: If STR_SEQA = 0b and if a:

Table continues on the next page...

ADC_CTL0 field descriptions (continued)

Field	Description
	• Sequence Abort Event or Restart Event is issued during a conversion the data of this conversion is not stored and the respective conversion complete flag is not set • Restart Event only is issued before the last conversion of a CSL is finished and no Sequence Abort Event is in process (SEQA clear) causes the RSTA_EIF error flag to be asserted and bit SEQA gets set by hardware. If STR_SEQA = 1b and if a: • Sequence Abort Event or Restart Event is issued during a conversion the data of this conversion is stored and the respective conversion complete flag is set and Intermediate Result Information Register is updated. • Restart Event only occurs during the last conversion of a CSL and no Sequence Abort Event is in process (SEQA clear) does not set the RSTA_EIF error flag • Restart Event only is issued before the CSL is finished and no Sequence Abort Event is in process (SEQA clear) causes the RSTA_EIF error flag to be asserted and bit SEQA gets set by hardware
0 MOD_CFG	(Conversion Flow Control) Mode Configuration This bit defines the conversion flow control after a Restart Event and after execution of the "End Of List" command type. 0 "Restart Mode" selected. 1 "Trigger Mode" selected.

22.6.5 ADC Interrupt Enable Register (ADC_IE)

Address: 4003_B000h base + 4h offset = 4003_B004h

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

ADC_IE field descriptions

Field	Description
7 SEQAD_IE	Conversion Sequence Abort Done Interrupt Enable Bit This bit enables the conversion sequence abort event done interrupt. 0 Conversion sequence abort event done interrupt is disabled. 1 Conversion sequence abort event done interrupt is enabled.
6 CONIF_OIE	ADCCONIF Register Flags Overrun Interrupt Enable This bit enables the flag which indicates if an overrun situation occurred for one of the CON_IF[15:1] flags or for the EOL_IF flag. 0 No ADC_CONIF Register Flag overrun is occurred. 1 ADC_CONIF Register Flag overrun is occurred.
Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

22.6.6 ADC Error Interrupt Enable Register (ADC_EIE)

Address: 4003_B000h base + 5h offset = 4003_B005h

Bit	7	6	5	4	3	2	1	0
Read	WA_EIE	CMD_EIE	EOL_EIE	0	TRIG_EIE	RSTAR_EIE	LDOK_EIE	RA_EIE
Write								
Reset	0	0	0	0	0	0	0	0

ADC_EIE field descriptions

Field	Description
7 WA_EIE	Write Access Error Interrupt Enable Bit This bit enables the write access error interrupt. 0 Write access error interrupt is disabled. 1 Write access error interrupt is enabled.
6 CMD_EIE	Command Value Error Interrupt Enable Bit This bit enables the command value error interrupt. 0 Command value interrupt is disabled. 1 Command value interrupt is enabled.
5 EOL_EIE	"End Of List" Error Interrupt Enable Bit This bit enables the "End Of List" error interrupt. 0 "End Of List" error interrupt is disabled. 1 "End Of List" error interrupt is enabled.
4 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
3 TRIG_EIE	Conversion Sequence Trigger Error Interrupt Enable Bit This bit enables the conversion sequence trigger error interrupt. 0 Conversion sequence trigger error interrupt is disabled. 1 Conversion sequence trigger error interrupt is enabled.
2 RSTAR_EIE	Restart Request Error Interrupt Enable Bit This bit enables the restart request error interrupt. 0 Restart Request error interrupt is disabled. 1 Restart Request error interrupt is enabled.
1 LDOK_EIE	Load OK Error Interrupt Enable Bit This bit enables the Load OK error interrupt. 0 Load OK error interrupt is disabled. 1 Load OK error interrupt is enabled.
0 RA_EIE	Read Access Error Interrupt Enable Bit

Table continues on the next page...

ADC_EIE field descriptions (continued)

Field	Description
	This bit enables the read access error interrupt. 0 Read access error interrupt is disabled. 1 Read access error interrupt is enabled.

22.6.7 ADC Conversion Flow Control Register (ADC_FLWCTL)

Bit set and bit clear instructions must not be used to access this register.

When the ADC is enabled, the bits of FLWCTL register can be modified after a latency time of three bus clock cycles.

All bits are cleared if bit ADC_EN is clear or via ADC soft-reset.

Timing considerations (Trigger Event - channel sample start) depending on ADC mode configuration:

- Restart mode

  When the restart event has been processed (initial command of current CSL is loaded), it takes two bus clock cycles plus two ADC conversion clock cycles (pump phase) from the trigger event (bit TRIG set) until the select channel starts to sample.

  During a conversion sequence (back to back conversions), it takes five bus clock cycles plus two ADC conversion clock cycles (pump phase) from current conversion period end until the newly selected channel is sampled in the following conversion period.

- Trigger mode

  When a restart event occurs, a trigger event is issued simultaneously. The time required to process the restart event is mainly defined by the internal read data bus availability and therefore can vary. In this mode, the trigger event is processed immediately after the restart event is finished and both conversion flow control bits are cleared simultaneously. From de-assert of bit TRIG until sampling begins five bus clock cycles are required. Hence from occurrence of a restart event until channel sampling, it takes five bus clock cycles plus an uncertainty of a few bus clock cycles.

NOTE

Bits SEQA, TRIG, RSTA, LDOK can be set only if bit ADC_EN is set.

Writing 0b to any of these bits does not have an effect.

See Summary of Conversion Flow Control Bit Scenarios for details.

Address: 4003_B000h base + 6h offset = 4003_B006h

Bit	7	6	5	4	3	2	1	0
Read	SEQA	TRIG	RSTA	LDOK	0			
Write								
Reset	0	0	0	0	0	0	0	0

ADC_FLWCTL field descriptions

Field	Description
7 SEQA	Conversion Sequence Abort Event This bit indicates that a conversion sequence abort event is in progress. When this bit is set, the ongoing conversion sequence and current CSL will be aborted at the next conversion boundary. This bit gets cleared when the ongoing conversion sequence is aborted and ADC is idle. This bit can only be set if bit ADC_EN is set. This bit is cleared if bit ADC_EN is clear. Data bus control: This bit can be controlled via the data bus if access control is configured accordingly via ACC_CFG[1:0]. Writing a value of 0b does not clear the flag. Writing a 1 to this bit does not clear it but causes an overrun if the bit has already been set. Internal interface control: This bit can be controlled via the internal interface signal "Seq_Abort" if access control is configured accordingly via ACC_CFG[1:0]. After being set, an additional request via the internal interface signal "Seq_Abort" causes an overrun. See also conversion flow control in case of overrun situations. General: In both conversion flow control modes, (Restart mode and Trigger mode) when bit RSTA gets set automatically, bit SEQA gets set when the ADC has not reached one of the following scenarios: • A sequence abort request is about to be executed or has been executed. • End Of List" command type has been executed or is about to be executed. In case bit SEQA is set, automatically the restart error flag RSTA_EIF is set to indicate an unexpected Restart Request. 0 No conversion sequence abort request. 1 Conversion sequence abort request.
6 TRIG	Conversion Sequence Trigger Bit This bit starts a conversion sequence if set and no conversion or conversion sequence is ongoing. This bit is cleared when the first conversion of a sequence starts to sample. This bit can only be set if bit ADC_EN is set. This bit is cleared if bit ADC_EN is clear. Data bus control: This bit can be controlled via the data bus if access control is configured accordingly via ACC_CFG[1:0]. Writing a value of 0b does not clear the flag. After being set, this bit can not be cleared by writing a value of 1b instead the error flag TRIG_EIF is set. Internal interface control:

Table continues on the next page...

ADC_FLWCTL field descriptions (continued)

Field	Description
	This bit can be controlled via the internal interface signal "Trigger" if access control is configured accordingly via ACC_CFG[1:0]. After being set, an additional request via internal interface signal "Trigger" causes the flag TRIG_EIF to be set. 0 No conversion sequence trigger. 1 Trigger to start conversion sequence.
5 RSTA	Restart Event (Restart from Top of Command Sequence List) This bit indicates that a restart event is executed. The ADC loads the conversion command from top of the active sequence command list when no conversion or conversion sequence is ongoing. This bit is cleared when the first conversion command of the sequence from top of active sequence command list has been loaded into the ADC_CMD0 and ADC_CMD1 registers. This bit can only be set if bit ADC_EN is set. This bit is cleared if bit ADC_EN is clear. Data bus control: This bit can be controlled via the data bus if access control is configured accordingly via ACC_CFG[1:0]. Writing a value of 0b does not clear the flag. Writing a 1b to this bit does not clear it but causes an overrun if the bit has already been set. Internal interface control: This bit can be controlled via the internal interface signal "Restart" if access control is configured accordingly via ACC_CFG[1:0]. After being set, an additional request via internal interface signal "Restart" causes an overrun. See conversion flow control in case of overrun situations for more details. General: In conversion flow control mode "Trigger mode", when bit RSTA gets set, bit TRIG is set simultaneously if one of the following has been executed: • "End Of List" command type has been executed or is about to be executed. • Sequence abort event 0 Continue with commands from active Sequence Command List. 1 Restart from top of active Sequence Command List.
4 LDOK	Load OK for alternative Command Sequence List This bit indicates if the preparation of the alternative sequence command list is done and command sequence list must be swapped with the Restart event. This bit is cleared when bit RSTA is set (Restart Event executed) and the command sequence list got swapped. This bit can only be set if bit ADC_EN is set. This bit is cleared if bit ADC_EN is clear. This bit is forced to zero if bit CSL_BMOD is clear. Data bus control: This bit can be controlled via the data bus if access control is configured accordingly via ACC_CFG[1:0]. Writing a value of 0b does not clear the flag. To set bit LDOK, the bits LDOK and RSTA must be written simultaneously. After being set, this bit can not be cleared by writing a value of 1b. Internal interface control:

Table continues on the next page...

ADC_FLWCTL field descriptions (continued)

Field	Description
	This bit can be controlled via the internal interface signal "LoadOK" and "Restart" if access control is configured accordingly via ACC_CFG[1:0]. With the assertion of Interface signal "Restart", the interface Signal "LoadOK" is evaluated and bit LDOK set accordingly (bit LDOK set if Interface signal "LoadOK" asserted when interface signal "Restart" asserts). General: Only in "Restart mode" if a Restart Event occurs without bit LDOK being set the error flag LDOK_EIF is set except when the respective restart request occurred after or simultaneously with a sequence abort request. The LDOK_EIF error flag is also not set in "Restart mode", if the first restart event occurs after: • ADC got enabled • Exit from Stop mode • ADC soft-reset 0 Load of alternative list done. 1 Load alternative list.
Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

22.6.8 ADC Format Register (ADC_FMT)

NOTE

Bits DJM and SRES[2:0] are writable if bit ADC_EN is clear or bit SMOD_ACC is set.

Address: 4003_B000h base + 7h offset = 4003_B007h

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

ADC_FMT field descriptions

Field	Description
7 DJM	Result Register Data Justification Conversion result data format is always unsigned. This bit controls justification of conversion result data in the conversion result list. 0 Left justified data in the conversion result list. 1 Right justified data in the conversion result list.
6–3 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
SRES	ADC Resolution Select

Table continues on the next page...

ADC_FMT field descriptions (continued)

Field	Description
	These bits select the resolution of conversion results. **NOTE:** Reserved settings cause a severe error at ADC conversion start whereby the CMD_EIF flag is set and ADC ceases operation. 000 8-bit data 001 Reserved 010 10-bit data 011 Reserved 100 12-bit data 1xx Reserved

22.6.9 ADC Conversion Interrupt Enable Register 1 (ADC_CONIE1)

Address: 4003_B000h base + 8h offset = 4003_B008h

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

ADC_CONIE1 field descriptions

Field	Description
7–1 CON_IE	Conversion Interrupt Enable Bits 7:1 These bits enable the individual interrupts which can be triggered via interrupt flags CON_IF[7:1]. 0 ADC conversion interrupt is disabled. 1 ADC conversion interrupt is enabled.
0 EOL_IE	End Of List Interrupt Enable Bit This bit enables the end of conversion sequence list interrupt. 0 End of list interrupt is disabled. 1 End of list interrupt is enabled.

22.6.10 ADC Conversion Interrupt Enable Register 0 (ADC_CONIE0)

Address: 4003_B000h base + 9h offset = 4003_B009h

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

ADC_CONIE0 field descriptions

Field	Description
CON_IE	Conversion Interrupt Enable Bits 15:8 These bits enable the individual interrupts which can be triggered via interrupt flags CON_IF[15:8]. 0 ADC conversion interrupt is disabled. 1 ADC conversion interrupt is enabled.

22.6.11 ADC Interrupt Flag Register (ADC_IF)

After being set, any of these bits can be cleared by writing a value of 1b or via ADC soft-reset (bit ADC_SR). All bits are cleared if bit ADC_EN is clear. Writing any flag with value 0b does not clear the flag. Writing any flag with value 1b does not set the flag.

Address: 4003_B000h base + Ah offset = 4003_B00Ah

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

ADC_IF field descriptions

Field	Description
7 SEQAD_IF	Conversion Sequence Abort Done Interrupt Flag This flag is set when the sequence abort event has been executed except the sequence abort event occurred by hardware to enter MCU Stop mode or Wait mode with bit SWAI set.This flag is also not set if the sequence abort request occurs during execution of the last conversion command of a CSL and bit STR_SEQA being set. 0 No conversion sequence abort request occurred. 1 A conversion sequence abort request occurred.
6 CONIF_OIF	ADCCONIF Register Flags Overrun Interrupt Flag This flag indicates if an overrun situation occurred for one of the CON_IF[15:1] flags or for the EOL_IF flag. In RVL single buffer mode (RVL_BMOD clear), an overrun of the EOL_IF flag is not indicated. **NOTE:** In RVL double buffer mode, a conversion interrupt flag (CON_IF[15:1]) or End Of List interrupt flag (EOL_IF) overrun is detected if one of these bits is set when it must be set again due to conversion command execution. In RVL single buffer mode, a conversion interrupt flag (CON_IF[15:1]) overrun is detected only. The overrun is detected if any of the conversion interrupt flags (CON_IF[15:1]) is set while the first conversion result of a CSL is stored (result of first conversion from top of CSL is stored). 0 No ADCCONIF register flag overrun occurred. 1 ADCCONIF register flag overrun occurred.
Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

22.6.12 ADC Error Interrupt Flag Register (ADC_EIF)

The ADC ceases operation, when one of the following error flags is set to 1b:

- IA_EIF
- CMD_EIF
- EOL_EIF
- TRIG_EIF

To make the ADC operational again, an ADC Soft-Reset must be issued which clears above listed error interrupt flags.

The error interrupt flags RSTAR_EIF and LDOK_EIF do not cause the ADC to cease operation. If set, the ADC continues operation. Each of the two bits can be cleared by writing a value of 1b. Both bits are also cleared if an ADC soft-reset is issued.

All bits are cleared if bit ADC_EN is clear. Writing any flag with value 0b does not clear a flag. Writing any flag with value 1b does not set the flag.

Address: 4003_B000h base + Bh offset = 4003_B00Bh

Bit	7	6	5	4	3	2	1	0
Read	WA_EIF	CMD_EIF	EOL_EIF	0	TRIG_EIF	RSTAR_EIF	LDOK_EIF	RA_EIF
Write								
Reset	0	0	0	0	0	0	0	0

ADC_EIF field descriptions

Field	Description
7 WA_EIF	Write Access Error Interrupt Flag This flag indicates that storing the conversion result caused an illegal access error occurred. The ADC ceases operation if this error flag is set (issue of type severe). 0 No write access error occurred. 1 A write access error occurred.
6 CMD_EIF	Command Value Error Interrupt Flag This flag indicates that an invalid command is loaded (any command that contains reserved bit settings) or illegal format setting selected (reserved SRES[2:0] bit settings). The ADC ceases operation if this error flag is set (issue of type severe). 0 Valid conversion command is loaded. 1 Invalid conversion command is loaded.
5 EOL_EIF	"End Of List" Error Interrupt Flag This flag indicates a missing "End Of List" command type in current executed CSL. The ADC ceases operation if this error flag is set (issue of type severe). 0 No "End Of List" error. 1 "End Of List" command type 'is missing in current executed CSL.

Table continues on the next page...

ADC_EIF field descriptions (continued)

Field	Description
4 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
3 TRIG_EIF	Trigger Error Interrupt Flag This flag indicates that a trigger error occurred. This flag is set in "Restart" mode when a conversion sequence got aborted and no restart event occurred before the trigger event or if the trigger event occurred before the restart event was finished (conversion command has been loaded). This flag is set in "Trigger" mode when a trigger event occurs before the restart event is issued to start conversion of the initial command sequence list. In "Trigger" mode, only a restart event is required to start conversion of the initial command sequence list. This flag is set when a trigger event occurs before a conversion sequence got finished. This flag is also set if a trigger occurs while a trigger event is just processed - first conversion command of a sequence is beginning to sample. This flag is also set if the trigger event occurs automatically generated by hardware in "Trigger mode" due to a restart event and simultaneously a trigger event is generated via data bus or internal interface. The ADC ceases operation if this error flag is set (issue of type severe). 0 No trigger error occurred. 1 A trigger error occurred.
2 RSTAR_EIF	Restart Request Error Interrupt Flag This flag indicates a flow control issue. It is set when a restart request occurs after a trigger event and before one of the following conditions was reached: • The "End Of List" command type has been executed • Depending on bit STR_SEQA if the "End Of List" command type is about to be executed • The current CSL has been aborted or is about to be aborted due to a sequence abort request. The ADC continues operation if this error flag is set. This flag is not set for Restart Request overrun scenarios. 0 No Restart request error situation occurred. 1 Restart request error situation occurred.
1 LDOK_EIF	Load OK Error Interrupt Flag This flag can only be set in "Restart mode". It indicates that a restart request occurred without LDOK. This flag is not set if a sequence abort event is already in process (bit SEQA set) when the restart request occurs or a sequence abort request occurs simultaneously with the restart request. The LDOK_EIF error flag is also not set in "Restart mode" if the first restart event occurs after: • ADC got enabled • Exit from Stop mode • ADC soft-reset • ADC used in CSL single buffer mode The ADC continues operation if this error flag is set. 0 No Load OK error situation occurred. 1 Load OK error situation occurred.
0 RA_EIF	Read Access Error Interrupt Flag This flag indicates that conversion command loading from outside system RAM or NVM area caused an illegal access error occurred. The ADC ceases operation if this error flag is set (issue of type severe).

Table continues on the next page...

ADC_EIF field descriptions (continued)

Field	Description
	0 No read access error occurred. 1 A read access error occurred.

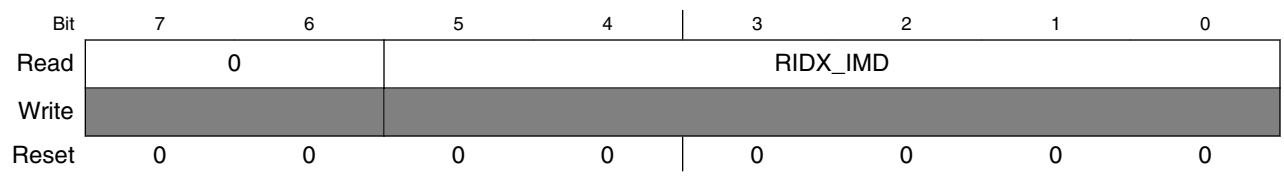
22.6.13 ADC Intermediate Result Information Register 1 (ADC_IMDRI1)

This register is cleared when bit ADC_SR is set or bit ADC_EN is clear.

NOTE

This register is updated and simultaneously a conversion interrupt flag CON_IF[15:1] occurs when the corresponding conversion command (conversion command with INTFLG_SEL[3:0] set) has been processed and related data has been stored to RAM.

Address: 4003_B000h base + Ch offset = 4003_B00Ch



ADC_IMDRI1 field descriptions

Field	Description
7–6 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
RIDX_IMD	RES_IDX Value At Intermediate Event These bits indicate the result index (RES_IDX) value at the occurrence of a conversion interrupt flag (CON_IF[15:1]) (occurrence of an intermediate result buffer fill event) or occurrence of EOL_IF flag or when a sequence abort event gets executed to abort an ongoing conversion (the result index RES_IDX is captured at the occurrence of a result data store). When a sequence abort event has been processed, flag SEQAD_IF is set and the RES_IDX value of the last stored result is provided. Hence in case an ongoing conversion is aborted, the RES_IDX value captured in RIDX_IMD bits depends on bit STORE_SEQA: • STORE_SEQA=1: The result index of the aborted conversion is provided • STORE_SEQA=0: The result index of the last stored result at abort execution time is provided In case a CSL is aborted while no conversion is ongoing (ADC waiting for a Trigger Event), the last captured result index is provided. In case a sequence abort event was initiated by hardware due to MCU entering Stop mode or Wait mode with bit SWAI set, the result index of the last stored result is captured by bits RIDX_IMD but flag SEQAD_IF is not set.

22.6.14 ADC Intermediate Result Information Register 0 (ADC_IMDRI0)

This register is cleared when bit ADC_SR is set or bit ADC_EN is clear.

NOTE

This register and IMDRI1 are updated and simultaneously a conversion interrupt flag CON_IF[15:1] occurs when the corresponding conversion command (conversion command with INTFLG_SEL[3:0] set) has been processed and related data has been stored to RAM.

Address: 4003_B000h base + Dh offset = 4003_B00Dh

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

ADC_IMDRI0 field descriptions

Field	Description
7 CSL_IMD	Active CSL At Intermediate Event This bit indicates the active (used) CSL at the occurrence of a conversion interrupt flag (CON_IF[15:1]) (occurrence of an intermediate result buffer fill event) or when a sequence abort event gets executed. 0 CSL_0 is active (used) when a conversion interrupt flag (CON_IF[15:1]) got set. 1 CSL_1 is active (used) when a conversion interrupt flag (CON_IF[15:1]) got set.
6 RVL_IMD	Active RVL At Intermediate Event This bit indicates the active (used) RVL buffer at the occurrence of a conversion interrupt flag (CON_IF[15:1]) (occurrence of an intermediate result buffer fill event) or when a sequence abort event gets executed. 0 RVL_0 is active (used) when a conversion interrupt flag (CON_IF[15:1]) got set. 1 RVL_1 is active (used) when a conversion interrupt flag (CON_IF[15:1]) got set.
Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

22.6.15 ADC Conversion Interrupt Flag Register 1 (ADC_CONIF1)

After being set, any of these bits can be cleared by writing a value of 1b. All bits are cleared if bit ADC_EN is clear or via ADC soft-reset (bit ADC_SR set). Writing any flag with value 0b does not clear the flag. Writing any flag with value 1b does not set the flag.

Address: 4003_B000h base + Eh offset = 4003_B00Eh

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

ADC_CONIF1 field descriptions

Field	Description
7–1 CON_IF	Conversion Interrupt Flags 7:1 These bits could be set by the binary coded interrupt select bits INTFLG_SEL[3:0] when the corresponding conversion command has been processed and related data has been stored to RAM. **NOTE:** These bits can be used to indicate if a certain packet of conversion results is available. Clearing a flag indicates that conversion results have been retrieved by software and the flag can be used again.
0 EOL_IF	End Of List Interrupt Flag This bit is set by the binary coded conversion command type select bits CMD_SEL[1:0] for "end of list" type of commands and after such a command has been processed and the related data has been stored RAM. **NOTE:** Overrun situation of a flag CON_IF[15:1] and EOL_IF are indicated by flag CONIF_OIF.

22.6.16 ADC Conversion Interrupt Flag Register 0 (ADC_CONIF0)

After being set, any of these bits can be cleared by writing a value of 1b. All bits are cleared if bit ADC_EN is clear or via ADC soft-reset (bit ADC_SR set). Writing any flag with value 0b does not clear the flag. Writing any flag with value 1b does not set the flag.

Address: 4003_B000h base + Fh offset = 4003_B00Fh

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

ADC_CONIF0 field descriptions

Field	Description
CON_IF	Conversion Interrupt Flags 15:8

ADC_CONIF0 field descriptions (continued)

Field	Description
	These bits could be set by the binary coded interrupt select bits INTFLG_SEL[3:0] when the corresponding conversion command has been processed and related data has been stored to RAM. **NOTE:** These bits can be used to indicate whether a certain packet of conversion results is available. Clearing a flag indicates that conversion results have been retrieved by software and the flag can be used again.

22.6.17 ADC End Of List Result Information Register (ADC_EOLRI)

This register is cleared when bit ADC_SR is set or bit ADC_EN is clear.

Address: 4003_B000h base + 13h offset = 4003_B013h

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

ADC_EOLRI field descriptions

Field	Description
7 CSL_EOL	Active CSL When "End Of List" Command Type Executed This bit indicates the active (used) CSL when a "End Of List" command type has been executed and related data has been stored to RAM. 0 CSL_0 is active when "End Of List" command type executed. 1 CSL_1 is active when "End Of List" command type executed.
6 RVL_EOL	Active RVL When "End Of List" Command Type Executed This bit indicates the active (used) RVL when a "End Of List" command type has been executed and related data has been stored to RAM. **NOTE:** The conversion interrupt EOL_IF occurs and simultaneously the register ADC_EOLRI is updated when the "End Of List" conversion command type has been processed and related data has been stored to RAM. 0 RVL_0 is active when "End Of List" command type executed. 1 RVL_1 is active when "End Of List" command type executed.
Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

22.6.18 ADC Command Register 2 (ADC_CMD2)

This register is writable only if SMOD_ACC is 1b.

If bit SMOD_ACC is set, modifying this register only when no conversion and conversion sequence is ongoing.

A command which contains reserved bit settings causes the error flag CMD_EIF being set and ADC cease operation.

Address: 4003_B000h base + 15h offset = 4003_B015h

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

ADC_CMD2 field descriptions

Field	Description
7–3 SMP	Sample Time Select Bits These bits select the length of the sample time in units of ADC conversion clock cycles. Note that the ADC conversion clock period is itself a function of the prescaler value (bits PRS[6:0]). 00000 Sample time in 4 ADC clock cycles. 00001 Sample time in 5 ADC clock cycles. 00010 Sample time in 6 ADC clock cycles. 00011 Sample time in 7 ADC clock cycles. 00100 Sample time in 8 ADC clock cycles. 00101 Sample time in 9 ADC clock cycles. 00110 Sample time in 10 ADC clock cycles. 00111 Sample time in 11 ADC clock cycles. 01000 Sample time in 12 ADC clock cycles. 01001 Sample time in 13 ADC clock cycles. 01010 Sample time in 14 ADC clock cycles. 01011 Sample time in 15 ADC clock cycles. 01100 Sample time in 16 ADC clock cycles. 01101 Sample time in 17 ADC clock cycles. 01110 Sample time in 18 ADC clock cycles. 01111 Sample time in 19 ADC clock cycles. 10000 Sample time in 20 ADC clock cycles. 10001 Sample time in 21 ADC clock cycles. 10010 Sample time in 22 ADC clock cycles. 10011 Sample time in 23 ADC clock cycles. 10100 Sample time in 24 ADC clock cycles. 10101 Reserved. 10110 Reserved. 10111 Reserved. 11xxx Reserved.
Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

22.6.19 ADC Command Register 1 (ADC_CMD1)

This register is writable only if SMOD_ACC is 1b.

If bit SMOD_ACC is set, modifying this register only when no conversion and conversion sequence is ongoing.

A command which contains reserved bit settings causes the error flag CMD_EIF being set and ADC cease operation.

Address: 4003_B000h base + 16h offset = 4003_B016h

Bit	7	6	5	4	3	2	1	0
Read Write	VRH_SEL	INTFLG_SEL	CH_SEL					
Reset	0	0	0	0	0	0	0	0

ADC_CMD1 field descriptions

Field	Description
7 VRH_SEL	Reference High Voltage Select Bit This bit selects the high voltage reference for current conversion. 0 VRH_0 input is selected as high voltage reference. 1 VRH_1 input is selected as high voltage reference.
6 INTFLG_SEL	Reference Low Voltage Select Bit This bit selects the voltage reference for current conversion. 0 VRL_0 input is selected as low voltage reference. 1 VRL_1 input is selected as low voltage reference.
CH_SEL	ADC Input Channel Select Bits These bits select the input channel for the current conversion. **NOTE:** ANx is the maximum number of implemented analog input channels on the device. 000000 VRL_0/1. 000001 VRH_0/1. 000010 (VRH_0/1 + VRL_0/1) / 2 000011 Reserved 000100 Reserved 000101 Reserved. 000110 Reserved. 000111 Reserved. 001000 Internal_0 (ADC temperature sense). 001001 Internal_1. 001010 Internal_2. 001011 Internal_3. 001100 Internal_4.

Table continues on the next page...

ADC_CMD1 field descriptions (continued)

Field	Description
	001101 Internal_5. 001110 Internal_6. 001111 Internal_7. 01xxxx ANx. CH_SEL=010000, x=0; CH_SEL=010001, x=1; CH_SEL=011011, x=15, etc 1xxxxx Reserved.

22.6.20 ADC Command Register 0 (ADC_CMD0)

This register is writable only if SMOD_ACC is 1b.

If bit SMOD_ACC is set, modifying this register only when no conversion and conversion sequence is ongoing.

Address: 4003_B000h base + 17h offset = 4003_B017h

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

ADC_CMD0 field descriptions

Field	Description
7–6 CMD_SEL	Conversion Command Select Bits These bits define the type of current conversion. 00 Normal conversion. 01 End of sequence (wait for trigger to execute next sequence or for a restart) 10 End of list (automatic wrap to top of CSL and continue conversion). 11 End of list (wrap to top of CSL and: in "Restart mode", wait for restart event followed by a trigger; in "Trigger mode", wait for trigger or restart event)
5–4 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
INTFLG_SEL	Conversion Interrupt Flag Select Bits These bits define which interrupt flag is set in the ADC_IFH/L register at the end of current conversion.The interrupt flags ADCIF[15:1] are selected via binary coded bits INTFLG_SEL[3:0]. 0000 No flag set. 0001 CON_IF[0x0001] is selected. 0010 CON_IF[0x0002] is selected. 0011 CON_IF[0x0004] is selected. 0100 CON_IF[0x0008] is selected. 0101 CON_IF[0x0010] is selected. 0110 CON_IF[0x0020] is selected. 0111 CON_IF[0x0040] is selected.

Table continues on the next page...

ADC_CMD0 field descriptions (continued)

Field	Description
	1000 CON_IF[0x0080] is selected. 1001 CON_IF[0x0100] is selected. 1010 CON_IF[0x0200] is selected. 1011 CON_IF[0x0400] is selected. 1100 CON_IF[0x0800] is selected. 1101 CON_IF[0x1000] is selected. 1110 CON_IF[0x2000] is selected. 1111 CON_IF[0x4000] is selected.

22.6.21 ADC Command Base Pointer Register 2 (ADC_CBP2)

NOTE

Bits CMD_PTR is writable only if bit ADC_EN clear or bit SMOD_ACC set.

Address: 4003_B000h base + 1Ch offset = 4003_B01Ch

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

ADC_CBP2 field descriptions

Field	Description
7–2 CMD_PTR	ADC Command Base Pointer Address [7:2] These bits define the base address of the two CSL areas inside the system RAM or NVM of the memory map. They are used to calculate the final address from which the conversion commands will be loaded depending on which list is active.
Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

22.6.22 ADC Command Base Pointer Register 1 (ADC_CBP1)

NOTE

Bits CMD_PTR is writable only if bit ADC_EN clear or bit SMOD_ACC set.

Address: 4003_B000h base + 1Dh offset = 4003_B01Dh

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

ADC_CBP1 field descriptions

Field	Description
CMD_PTR	ADC Command Base Pointer Address [15:8] These bits define the base address of the two CSL areas inside the system RAM or NVM of the memory map. They are used to calculate the final address from which the conversion commands will be loaded depending on which list is active.

22.6.23 ADC Command Base Pointer Register 0 (ADC_CBP0)

NOTE

Bits CMD_PTR is writable only if bit ADC_EN clear or bit SMOD_ACC set.

Address: 4003_B000h base + 1Eh offset = 4003_B01Eh

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

ADC_CBP0 field descriptions

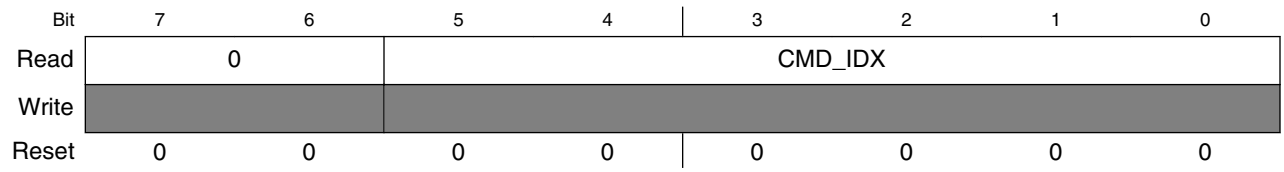
Field	Description
7 CMD_PTR	ADC Command Base Pointer Address [23] If bit 23 is 1, the CSL list start address will be 0x0000_0000 + CMD_PTR[15:0], if bit 23 is 0, the CSL list start address will be 0x2000_0000 + CMD_PTR[15:0]
Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

22.6.24 ADC Command Index Register (ADC_CIDX)

NOTE

These bits do not represent absolute addresses, instead it is a sample index (object size 32 bits)

Address: 4003_B000h base + 1Fh offset = 4003_B01Fh



ADC_CIDX field descriptions

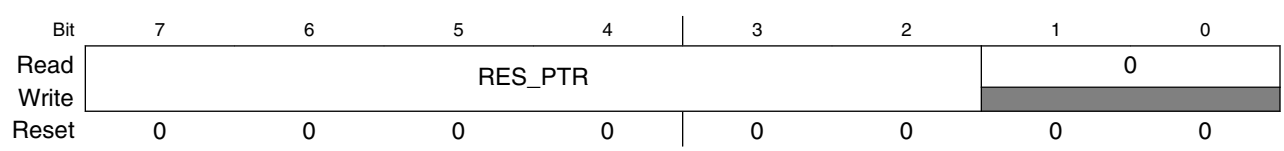
Field	Description
7–6 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
CMD_IDX	ADC Command Index Bits These bits represent the command index value for the conversion commands relative to the two CSL start addresses in the memory map. These bits do not represent absolute addresses instead it is a sample index (object size 32 bits).

22.6.25 ADC Result Base Pointer Register 2 (ADC_RBP2)

NOTE

Bits RES_PTR is writable only if bit ADC_EN clear or bit SMOD_ACC set.

Address: 4003_B000h base + 20h offset = 4003_B020h



ADC_RBP2 field descriptions

Field	Description
7–2 RES_PTR	ADC Result Base Pointer Address [7:2]

Table continues on the next page...

ADC_RBP2 field descriptions (continued)

Field	Description
	These bits define the base address of the list areas inside the system RAM of the memory map to which conversion results will be stored to at the end of a conversion. These bits can only be written if bit ADC_EN is clear.
Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

22.6.26 ADC Result Base Pointer Register 1 (ADC_RBP1)

NOTE

Bits RES_PTR is writable only if bit ADC_EN clear or bit SMOD_ACC set.

Address: 4003_B000h base + 21h offset = 4003_B021h

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

ADC_RBP1 field descriptions

Field	Description
RES_PTR	ADC Result Base Pointer Address [15:8] These bits define the base address of the list areas inside the system RAM of the memory map to which conversion results will be stored to at the end of a conversion. The conversion result list start address will be 0x20000_0000 + RESPTR[15:0]. These bits can only be written if bit ADC_EN is clear.

22.6.27 ADC Result Index Register (ADC_RIDX)

NOTE

These bits do not represent absolute addresses, instead it is a sample index (object size 32 bits)

Address: 4003_B000h base + 23h offset = 4003_B023h

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

ADC_RIDX field descriptions

Field	Description
7–6 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
RES_IDX	ADC Result Index Bits These read only bits represent the index value for the conversion results relative to the two RVL start addresses in the memory map. These bits do not represent absolute addresses, instead it is a sample index (object size 16 bits).

22.6.28 ADC Command and Result Offset Register 1 (ADC_CROFF1)

NOTE

These bits do not represent absolute addresses, instead it is an sample offset (object size 16 bits for RVL, object size 32 bits for CSL).

These bits are writable if bit ADC_EN clear or bit SMOD_ACC set.

Address: 4003_B000h base + 26h offset = 4003_B026h

Bit	7	6	5	4	3	2	1	0
Read	0	CMDRES_OFF1						
Write								
Reset	0	0	0	0	0	0	0	0

ADC_CROFF1 field descriptions

Field	Description
7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
CMDRES_OFF1	ADC Command and Result Offset Value These bits represent the conversion command and result offset value relative to the conversion command base pointer address and result base pointer address in the memory map to refer to CSL_1 and RVL_1. It is used to calculate the address inside the system RAM to which the result at the end of the current conversion is stored to and the area (RAM or NVM) from which the conversion commands are loaded from. These bits do not represent absolute addresses, instead it is an sample offset (object size 16 bits for RVL, object size 32 bits for CSL).,These bits can only be modified if bit ADC_EN is clear.

22.6.29 ADC Command and Result Offset Register 0 (ADC_CROFF0)

Address: 4003_B000h base + 27h offset = 4003_B027h

Bit	7	6	5	4	3	2	1	0
Read	0	CMDRES_OFF0						
Write								
Reset	0	0	0	0	0	0	0	0

ADC_CROFF0 field descriptions

Field	Description
7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
CMDRES_OFF0	ADC Command and Result Offset Value These read only bits represent the conversion command and result offset value relative to the conversion command base pointer address and result base pointer address in the memory map to refer to CSL_0 and RVL_0. It is used to calculate the address inside the system RAM to which the result at the end of the current conversion is stored to and the area (RAM or NVM) from which the conversion commands are loaded from. This is a zero offset (null offset) which can not be modified. These bits do not represent absolute addresses instead it is a sample offset (object size 16 bits for RVL, object size 32 bits for CSL).

22.7 Functional Description

22.7.1 Overview

The ADC block consists of an analog sub-block and a digital sub-block. It is a successive approximation analog-to-digital converter including a sample-and-hold mechanism and an internal charge scaled C-DAC (switched capacitor scaled digital-to-analog converter) with a comparator to realize the successive approximation algorithm.

22.7.2 Analog sub-block

The analog sub-block contains all analog circuits (sample and hold, C-DAC, analog comparator, and so on) required to perform a single conversion. Separate power supplies VDDA and VSSA allow noise from the MCU circuitry to be isolated from the analog sub-block for improved accuracy.

22.7.2.1 Analog input multiplexer

The analog input multiplexers connect one of the external or internal analog input channels to the sample and hold storage node.

22.7.2.2 Sample and hold machine with sample buffer amplifier

The sample and hold machine controls the storage and charge of the storage node (sample capacitor) to the voltage level of the analog signal at the selected ADC input channel. This architecture employs the advantage of reduced crosstalk between channels.

The sample buffer amplifier is used to raise the effective input impedance of the A/D machine, so that external components (higher bandwidth or higher impedance connected as specified) are less significant to accuracy degradation.

During the sample phase, the analog input connects first via a sample buffer amplifier with the storage node always for two ADC clock cycles ("Buffer" sample time). For the remaining sample time ("Final" sample time) the storage node is directly connected to the analog input source. Please see also the following figure for illustration and the Appendix of the device reference manual for more details.

The input analog signals are unipolar and must be within the potential range of VSSA to VDDA. During the hold process, the analog input is disconnected from the storage node.

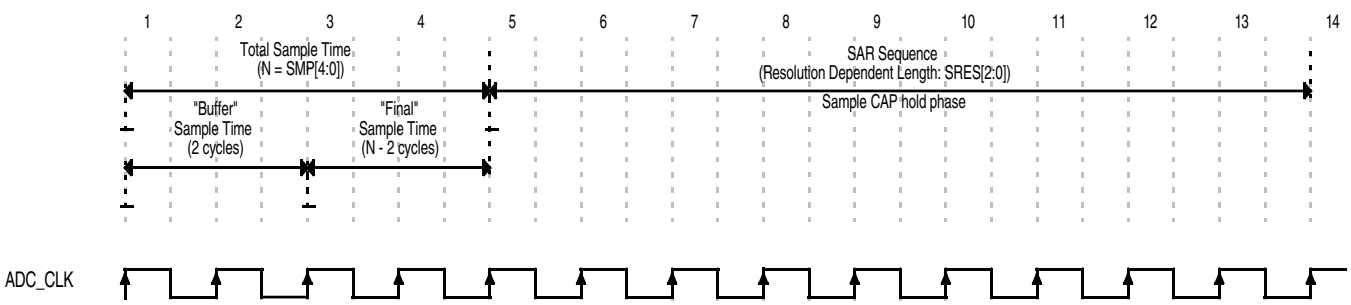


Figure 22-32. Sampling and conversion timing example (8-bit resolution, 4 cycle sampling)

Please note that there is always a pump phase of two ADC_CLK cycles before the sample phase begins. During this pump phase, a dynamic and high resistive connection from the selected channel source to the sample node already exists. It is the phase during which the multiplexers (transfer gates) are switched, hence high energetic glitches during the pump phase could impact the conversion accuracy for short sample times.

22.7.3 Digital sub-block

The digital sub-block contains a list-based programmer's model and the control logic for the analog sub-block circuits.

22.7.3.1 Analog-to-Digital (AD) machine

The A/D machine performs the analog-to-digital conversion. The resolution is program selectable to be either 8- or 10- or 12-bit. The A/D machine uses a successive approximation architecture. It functions by comparing the sampled and stored analog voltage with a series of binary coded discrete voltages.

By following a binary search algorithm, the A/D machine identifies the discrete voltage that is nearest to the sampled and stored voltage.

Only analog input signals within the potential range of VRL_0/1 to VRH_0/1 (A/D reference potentials) will result in a non-railed digital output code.

22.7.3.2 Introduction of the programmer's model

The ADC provides a programmer's model that uses a system memory list-based architecture for definition of the conversion command sequence and conversion result handling.

The command sequence list (CSL) and result value list (RVL) are implemented in double buffered manner and the buffer mode is user selectable for each list (bits CSL_BMOD, RVL_BMOD). The 32-bit wide conversion command is double buffered and the currently active command is visible in the ADC register map at ADC_CMD0, ADC_CMD1, ADC_CMD2, and ADC_CMD3 registers space.

22.7.3.2.1 Introduction of the command sequence list format

A command sequence list (CSL) contains up to 64 conversion commands. A user selectable number of successive conversion commands in the CSL can be grouped as a command sequence. This sequence of conversion commands is successively executed by the ADC at the occurrence of a trigger event. The commands of a sequence are successively executed until an "End Of Sequence" or "End Of List" command type identifier in a command is detected (command type is coded via bits ADC_CMD0[CMD_SEL]). The number of successive conversion commands that belong to a command sequence and the number of command sequences inside the CSL can be

freely defined by the user and is limited by the 64 conversion commands a CSL can contain. A CSL must contain at least one conversion command and one "end of list" command type identifier. The minimum number of command sequences inside a CSL is zero and the maximum number of command sequences is 63. A command sequence is defined with bits ADC_CMD0[CMD_SEL] by defining the end of a conversion sequence. The following two figures provide examples of a CSL.

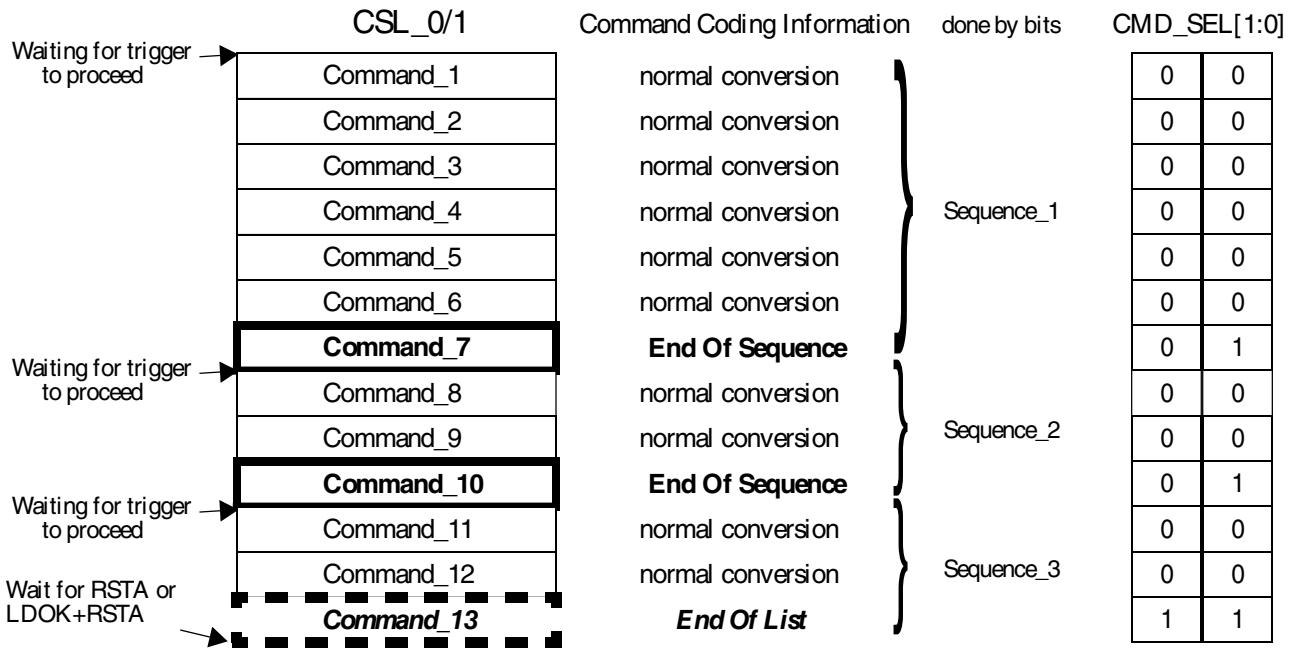


Figure 22-33. Example CSL with sequences and an "End Of List" command type identifier

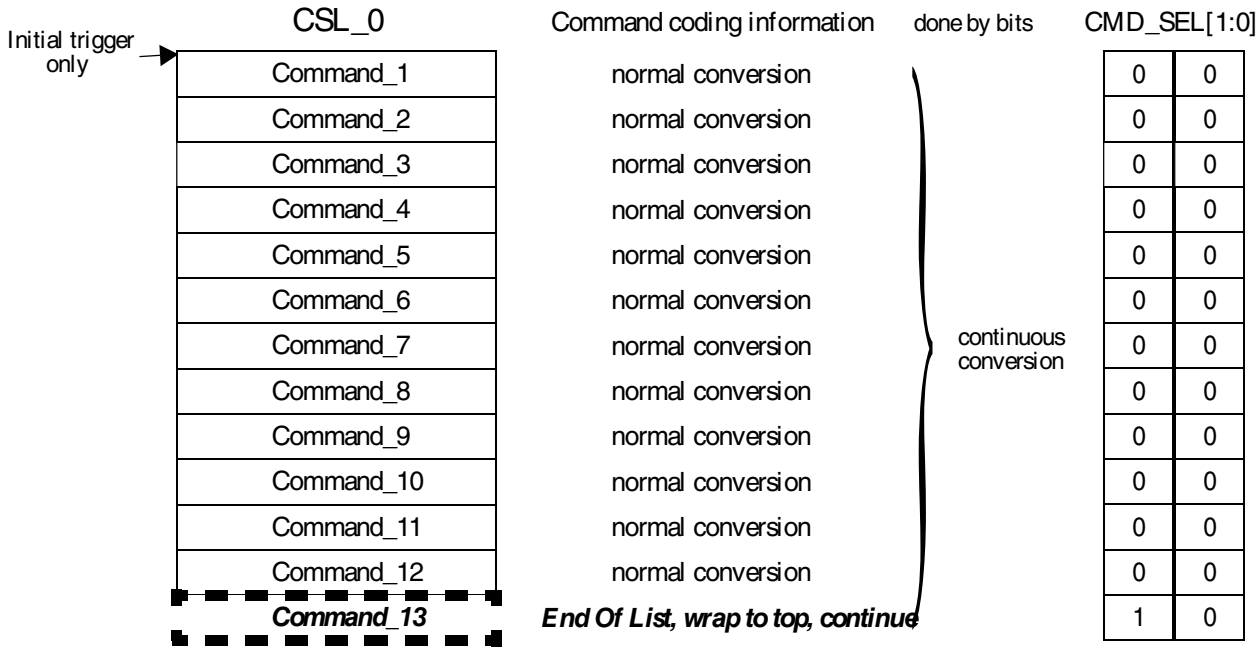


Figure 22-34. Example CSL for continues conversion

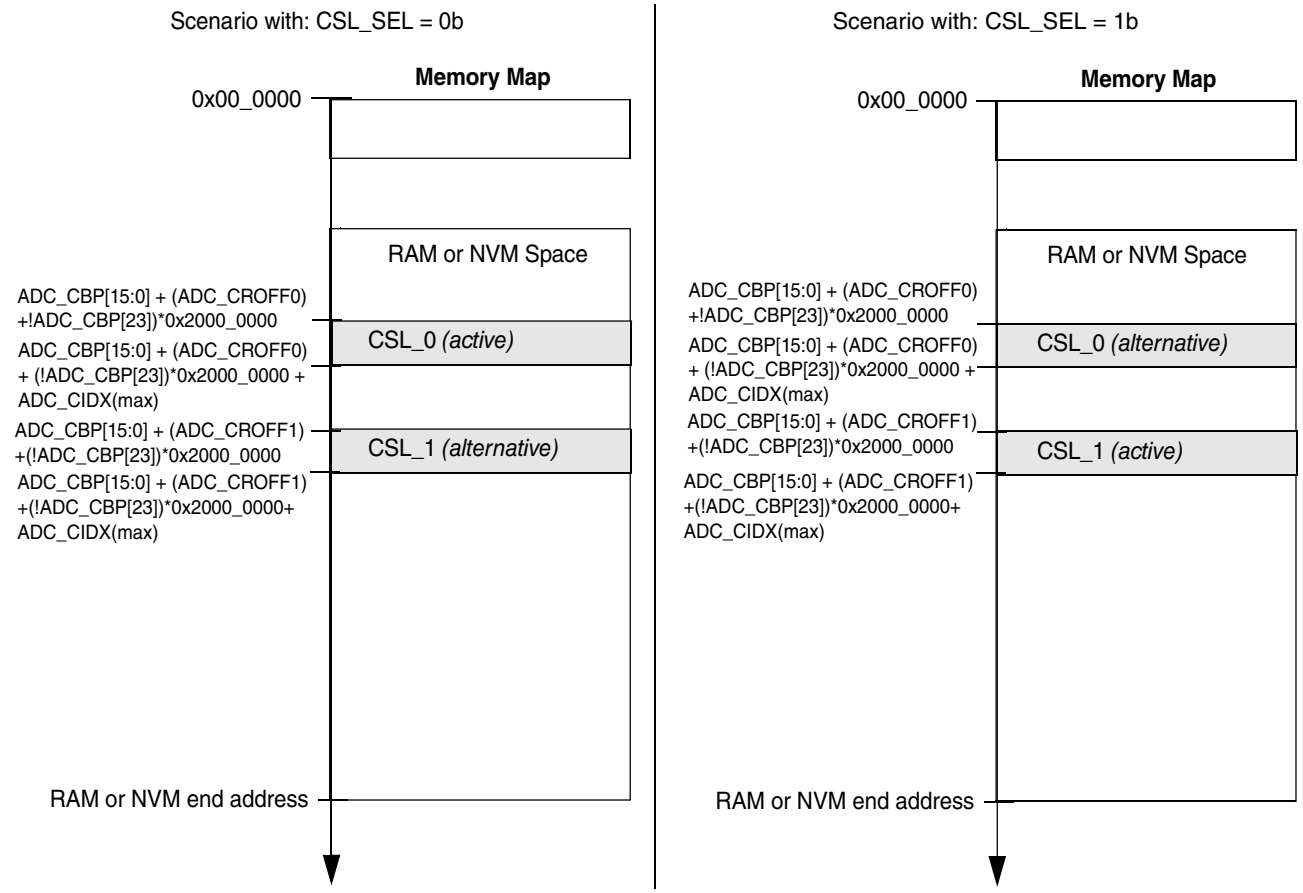
22.7.3.2.2 Introduction of the two command sequence lists

The two command sequence lists (CSLs) can be referred to via the Command Base Pointer Register plus the Command and Result Offset Registers plus the Command Index Register (ADC_CBP, ADC_CROFF0, ADC_CROFF1, ADC_CIDX).

The final address for conversion command loading is calculated by the sum of these registers (e.g.: ADC_CBP[15:0] + ADC_CROFF0/1 + ADC_CIDX + (!ADC_CBP[23]) x 0x2000_0000).

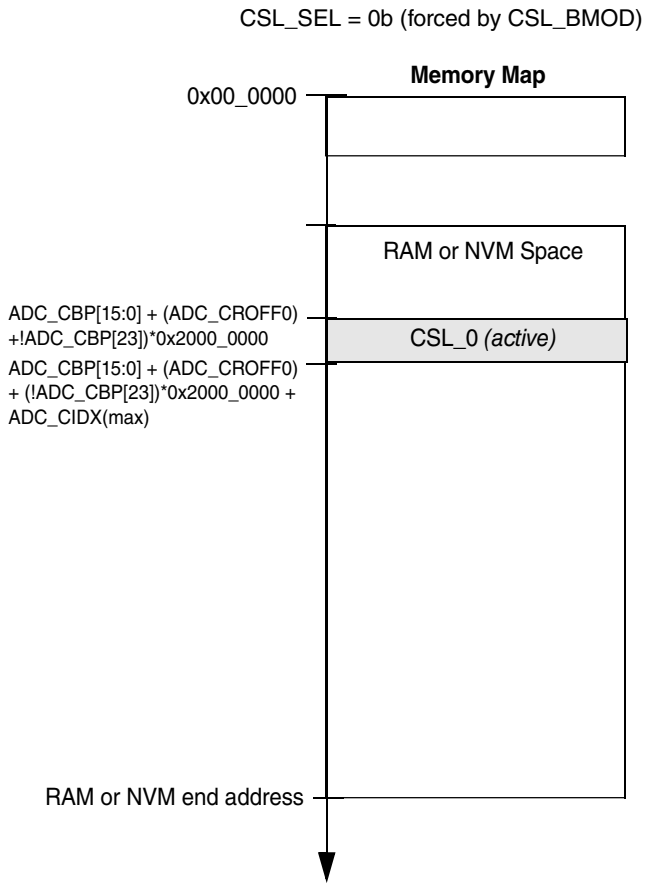
Bit ADC_CTL1[CSL_BMOD] selects if the CSL is used in double buffer or single buffer mode. In double buffer mode, the CSL can be swapped by ADC_FLWCTL[LDOK] and ADC_FLWCTL[RSTA]. For detailed information about when and how the CSL is swapped, see The four ADC conversion flow control bits - description of Restart Event + CSL Swap, Initial start of a command sequence list and Restart CSL execution with newother CSL (alternative CSL becomes active CSL) - CSL swapping .

Which list is actively used for ADC command loading is indicated by ADC_STS[CSL_SEL]. The register to define the CSL start addresses (ADC_CBP) can be set to any even location of the system RAM or NVM area. The different ADC lists must not overlap or exceed the system RAM or the NVM area, respectively. The error flag ADC_EIE[IA_EIF] will be set for accesses to ranges outside system RAM area and cause an error interrupt if enabled.



Note: Address register names in () are not absolute addresses instead they are a sample offset or sample index

Figure 22-35. Command sequence list Schema in double buffer mode



Note: Address register names in () are not absolute addresses instead they are a sample offset or sample index

Figure 22-36. Command sequence list Schema in single buffer mode

While the ADC is enabled, one CSL is active (indicated by ADC_STS[CSL_SEL]) and the corresponding list must not be modified anymore. At the same time, the alternative CSL can be modified to prepare the ADC for new conversion sequences in CSL double buffered mode. When the ADC is enabled, the command address registers (ADC_CBP, ADC_CROFF0, ADC_CROFF1, and ADC_CIDX) are read only and register ADC_CIDX is under control of the ADC.

22.7.3.2.3 Introduction of the two result value lists (RVLs)

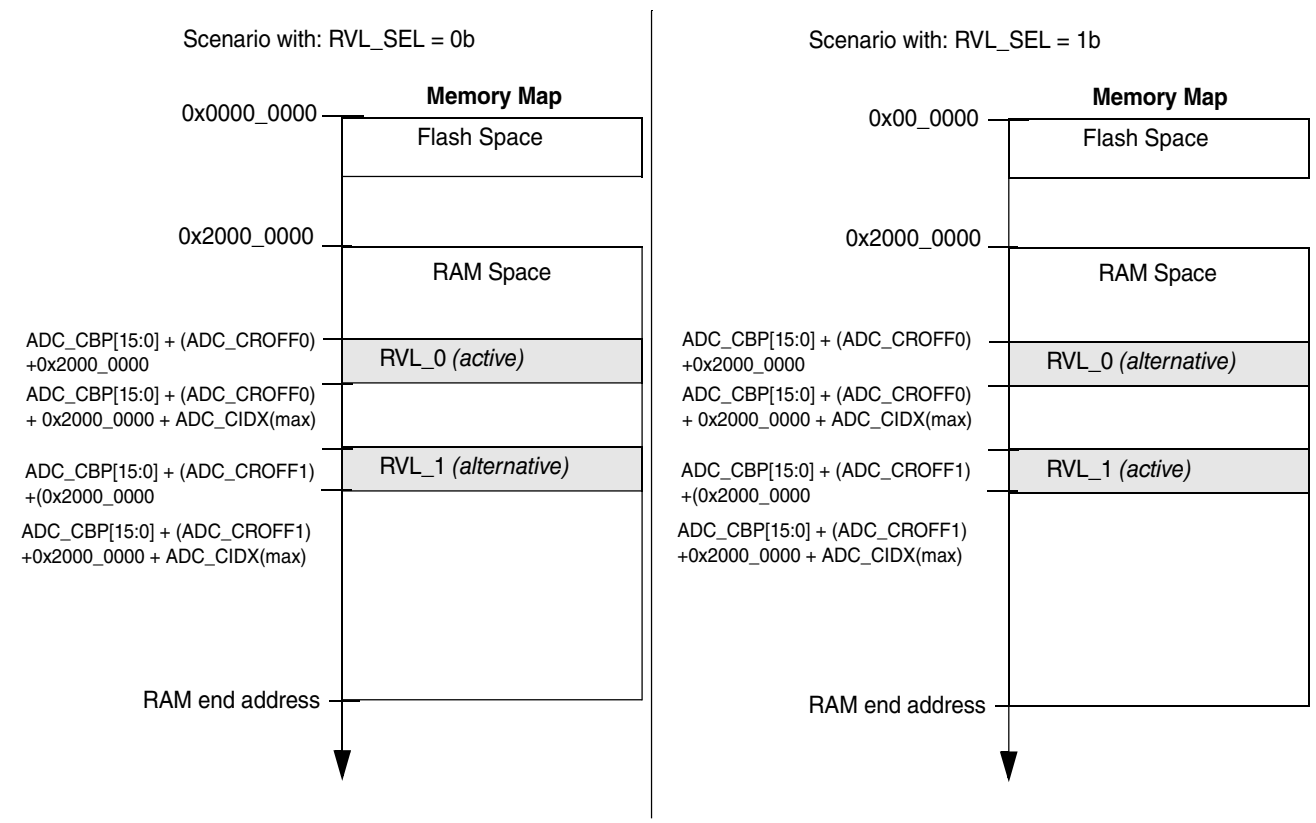
The same list-based architecture as described above for the CSL has been implemented for the result value list (RVL) with corresponding address registers (ADC_RBP, ADC_CROFF0, ADC_CROFF1, and ADC_RIDX).

The final address for conversion result storage is calculated by the sum of these registers (e.g.: 0x2000_0000 + ADC_RBP[15:0] +ADC_CROFF0+ADC_RIDX or 0x2000_0000 + ADC_RBP[15:0] ADC_CROFF1 + ADC_RIDX).

The ADC_EIF[RVL_BMOD] selects if the RVL is used in double buffer or single buffer mode. In double buffer mode, the RVL is swapped:

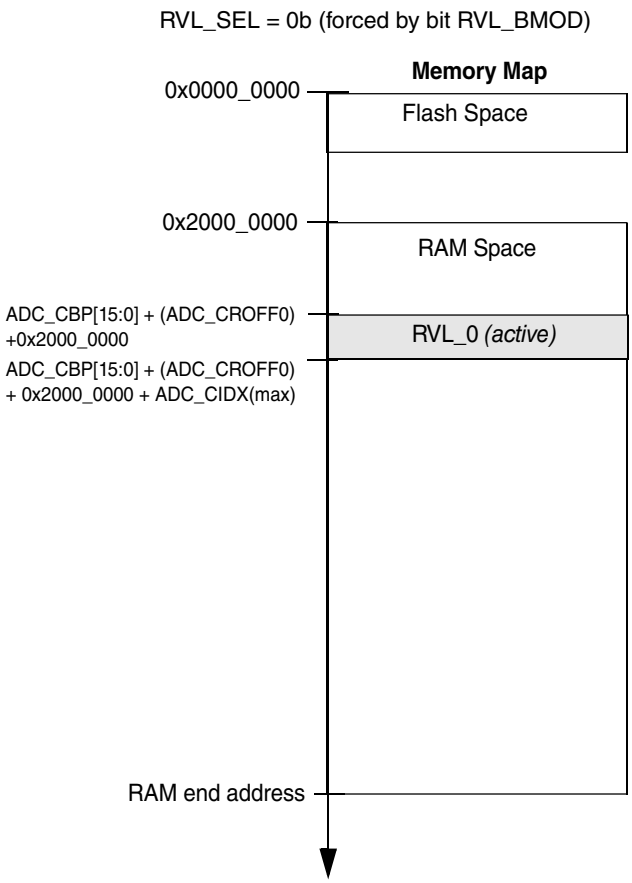
- Each time an "End Of List" command type got executed followed by the first conversion from top of the next CSL and related (first) result is about to be stored
- A CSL got aborted (ADC_FLWCTL[SEQA]=1b) and ADC enters idle state (becomes ready for new flow control events)

Using the RVL in double buffer mode, the RVL is not swapped after exit from Stop mode or Wait mode with ADC_CTL0[SWAI] set. Hence the RVL used before entry of Stop or Wait mode with ADC_CTL0[SWAI] set is overwritten after exit from the MCU Operating mode (see MCU operating modes). Which list is actively used for the ADC conversion result storage is indicated by ADC_STS[RVL_SEL]. The register to define the RVL start addresses (ADC_RBP) can be set to any even location of the system RAM area. The different ADC lists must not overlap or exceed the system RAM area. The error flag ADC_EIF[IA_EIF] will be set for accesses to ranges outside system RAM area and cause an error interrupt if enabled.



Note: Address register names in () are not absolute addresses instead they are a sample offset or sample index

Figure 22-37. Result value list schema in double buffer mode



Note: Address register names in () are not absolute addresses instead they are a sample offset or sample index

Figure 22-38. Result value list schema in single buffer mode

While ADC is enabled, one result value list is active (indicated by ADC_STS[RVL_SEL]). The conversion result value list can be read anytime. When the ADC is enabled, the conversion result address registers (ADC_RBP, ADC_CROFF0, ADC_CROFF1, and ADC_RIDX) are read only and register ADC_RIDX is under control of the ADC.

A conversion result is always stored as 16-bit entity in unsigned data representation. Left and right justification inside the entity is selected via the ADC_FMT[DJM]. Unused bits inside an entity are stored zero.

Table 22-35. Conversion result justification overview

Conversion resolution (ADC_FMT[SRES])	Left justified result (ADC_FMT[DJM]=0b)	Right justified result (ADC_FMT[DJM]=1b)
8-bit	{Result[7:0], 00000000b}	{00000000b,Result[7:0]}
10-bit	{Result[9:0], 000000b}	{000000b,Result[9:0]}
12-bit	{Result[11:0], 0000b}	{0000b,Result[11:0]}

22.7.3.2.4 The two conversion flow control mode configurations

The ADC provides two modes, "Trigger mode" and "Restart mode", which are different in the conversion control flow.

- The "Restart mode" provides precise timing control about the sample start point but is more complex from the flow control perspective.
- The "Trigger mode" is more simple from flow control point of view but is less controllable regarding conversion sample start.

Following are the key differences:

In "Trigger mode" configuration, when ADC_FLWCTL[RSTA] gets set, the ADC_FLWCTL[TRIG] gets set automatically. Hence in "Trigger mode", the applications must not set the ADC_FLWCTL[TRIG] and ADC_FLWCTL[RSTA] simultaneously (via data bus or internal interface), because it is a flow control failure and the ADC will cease operation.

In "Trigger Mode" configuration, after the execution of the initial restart event, the current CSL can be executed and controlled via trigger events only. Hence, if the "End Of List" command is reached, a restart of conversion flow from top of current CSL does not require to set ADC_FLWCTL[RSTA], because returning to the top of current CSL is done automatically. Therefore, the current CSL can be executed again after the "End Of List" command type is executed by a trigger event only.

In "Restart mode" configuration, the execution of a CSL is controlled via trigger events and restart events. After execution of the "End Of List" command, the conversion flow must be continued by a restart event followed by a trigger event and the trigger event must not occur before the restart event has finished.

For more details and examples regarding flow control and application use cases, see The four ADC conversion flow control bits and Conversion flow control application information .

22.7.3.2.5 The four ADC conversion flow control bits

There are four bits to control conversion flow (execution of a CSL and CSL exchange in double buffer mode). Each bit is controllable via the data bus and internal interface depending on the setting of ADC_CTL0[ACC_CFG]. In the following, the conversion control event to control the conversion flow is given with the related internal interface signal and corresponding register bit name together with information regarding:

- Function of the conversion control event
- How to request the event
- When is the event finished
- Mandatory requirements to executed the event

Trig event

Internal interface signal: trigger

Corresponding bit name: ADC_FLWCTL[TRIG]

- Function:

  Start the first conversion of a conversion sequence which is defined in the active command sequence list

- Requested by:
  - Positive edge of internal interface signal trigger
  - Write access via data bus to set control bit ADC_FLWCTL[TRIG]
- When finished:

  This bit is cleared by the ADC when the first conversion of the sequence is beginning to sample

- Mandatory requirements:
  - In all ADC conversion flow control modes, ADC_FLWCTL[TRIG] is set (trigger event executed) only if the trigger event occurs while no conversion or conversion sequence is ongoing (ADC idle)
  - In ADC conversion flow control mode "Restart mode" with a restart event in progress, it is not allowed that a trigger event occurs before the background command load phase has finished (restart event has been executed) else the error flag ADC_EIF[TRIG_EIF] is set
  - In ADC conversion flow control mode "Trigger mode", a restart event causes ADC_FLWCTL[TRIG] being set automatically. Bit ADC_FLWCTL[TRIG] is set when no conversion or conversion sequence is ongoing (ADC idle) and the RVL done condition is reached by one of the following:
    - A "End Of List" command type has been executed
    - A sequence abort event is in progress or has been executed

    The ADC executes the restart event followed by the trigger event.

  - In ADC conversion flow control mode "Trigger mode", a restart event and a simultaneous trigger event via internal interface or data bus causes the ADC_EIF[TRIG_EIF] being set and ADC cease operation.

Restart event (with current active CSL)

Internal interface signal: restart

Corresponding bit name: ADC_FLWCTL[RSTA]

- Function:
  - Go to top of active CSL (clear index register for CSL)

- Load one background command register and wait for trigger (CSL offset register is not switched independent of ADC_CTL1[CSL_BMOD])
- Set error flag ADC_EIF[RSTAR_EIF] when a restart request occurs before one of the following conditions was reached:
  - The "End Of List" command type has been executed
  - Depending on ADC_CTL0[STR_SEQA] if the "End Of List" command type is about to be executed
  - The current CSL has been aborted or is about to be aborted due to a sequence abort request.

- Requested by:
  - Positive edge of internal interface signal Restart
  - Write Access via data bus to set control bit ADC_FLWCTL[RSTA]
- When finished:

  This bit is cleared when the first conversion command of the sequence from top of active sequence command list is loaded.

- Mandatory requirement:
  - In all ADC conversion flow control modes, a restart event causes ADC_FLWCTL[RSTA] to be set. ADC_FLWCTL[SEQA] is set simultaneously by ADC hardware if:
    - ADC not idle (a conversion or conversion sequence is ongoing and current CSL not finished) and no sequence abort event in progress (ADC_FLWCTL[SEQA] not already set or set simultaneously via internal interface or data bus)
    - ADC idle but RVL done condition not reached

    The RVL done condition is reached by one of the following:
    - A "End Of List" command type has been executed
    - A sequence abort event is in progress or has been executed (ADC_FLWCTL[SEQA] already set or set simultaneously via internal interface or data bus)

    The ADC executes the sequence abort event followed by the restart event for the conditions described before or only a restart event.

  - In ADC conversion flow control mode "Trigger mode", a restart event causes ADC_FLWCTL[TRIG] being set automatically. ADC_FLWCTL[TRIG] is set when no conversion or conversion sequence is ongoing (ADC idle) and the RVL done condition is reached by one of the following:
    - A "End Of List" command type has been executed
    - A sequence abort event is in progress or has been executed

The ADC executes the restart event followed by the trigger event.

- In ADC conversion flow control mode "Trigger mode", a restart event and a simultaneous trigger event via internal interface or data bus causes the ADC_EIF[TRIG_EIF] being set and ADC cease operation.

Restart event + CSL exchange (swap)

Internal interface signals: restart + LoadOK

Corresponding bit names: ADC_FLWCTL[RSTA] + ADC_FLWCTL[LDOK]

- Function:
  Go to top of active CSL (clear index register for CSL) and switch to other offset register for address calculation if configured for double buffer mode (exchange the CSL list) requested by:
  - Internal interface with the assertion of interface signal restart the interface signal LoadOK is evaluated and ADC_FLWCTL[LDOK] is set accordingly (ADC_FLWCTL[LDOK] set if interface signal LoadOK asserted when interface signal restart asserts).
  - Write access via data bus to set control bit ADC_FLWCTL[RSTA] simultaneously with ADC_FLWCTL[LDOK].
- When finished:

  ADC_FLWCTL[LDOK] can only be cleared if it was set as described before and both bits (ADC_FLWCTL[LDOK], ADC_FLWCTL[RSTA]) are cleared when the first conversion command from top of active sequence command list is loaded

- Mandatory requirement:

  No ongoing conversion or conversion sequence

  Details if using the internal interface:

  If signal restart is asserted before signal LoadOK is set, the conversion starts from top of currently active CSL at the next trigger event (no exchange of CSL list).

  If signal restart is asserted after or simultaneously with signal LoadOK, the conversion starts from top of the other CSL at the next trigger event (CSL is switched) if CSL is configured for double buffer mode.

Sequence abort event

Internal interface signal: Seq_Abort

Corresponding bit name: ADC_FLWCTL[SEQA]

- Function:

Abort any possible ongoing conversion at next conversion boundary and abort current conversion sequence and active CSL

- Requested by:
  - Positive edge of internal interface signal Seq_Abort
  - Write access via data bus to set control bit ADC_FLWCTL[SEQA]
- When finished:

  This bit gets cleared when an ongoing conversion is finished and the result is stored and/or an ongoing conversion sequence is aborted and current active CSL is aborted (ADC idle, RVL done)

- Mandatory requirement:
  - In all ADC conversion flow control modes, ADC_FLWCTL[SEQA] can only be set if:
    - ADC not idle (a conversion or conversion sequence is ongoing)
    - ADC idle but RVL done condition not reached

    The RVL done condition is not reached if:
    - An "End Of List" command type has not been executed
    - A sequence abort event has not been executed (ADC_FLWCTL[SEQA] not already set)
  - In all ADC conversion flow control modes, a sequence abort event can be issued at any time
  - In ADC conversion flow control mode "Restart mode", after a conversion sequence abort request has been executed, it is mandatory to set ADC_FLWCTL[RSTA]. If a trigger event occurs before a restart event is executed (ADC_FLWCTL[RSTA] set and cleared by hardware), ADC_FLWCTL[TRIG] is set, error flag ADC_EIF[TRIG_EIF] is set, and the ADC can only be continued by a soft-reset. After the restart event, the ADC accepts new trigger events (ADC_FLWCTL[TRIG] set) and begins conversion from top of the currently active CSL.
  - In ADC conversion flow control mode "Restart mode", after a sequence abort event has been executed, a restart event causes only the ADC_FLWCTL[RSTA] being set. The ADC executes a restart event only.
  - In both conversion flow control modes, "Restart mode" and "Trigger mode", when conversion flow control bit ADC_FLWCTL[RSTA] gets set, automatically ADC_FLWCTL[SEQA] gets set when the ADC has not reached one of the following scenarios:
    - An "End Of List" command type has been executed or is about to be executed
    - A sequence abort request is about to be executed or has been executed.

In case ADC_FLWCTL[SEQA] is set, automatically the Restart error flag ADC_EIF[RSTAR_EIF] is set to indicate an unexpected restart request.

22.7.3.2.6 Conversion flow control in case of conversion sequence control bit overrun scenarios

- Restart request overrun:

  If a legal restart request is detected and no restart event is in progress, the ADC_FLWCTL[RSTA] is set due to the request. The set ADC_FLWCTL[RSTA] indicates that a restart request was detected and the restart event is in process. In case further restart requests occur while the ADC_FLWCTL[RSTA] is set, this is defined a overrun situation. This scenario is likely to occur when ADC_CTL0[STR_SEQA] is set or when a restart event causes a sequence abort event. The request overrun is captured in a background register that always stores the last detected overrun request. Hence if the overrun situation occurs more than once while a restart event is in progress, only the latest overrun request is pending. When the ADC_FLWCTL[RSTA] is cleared, the latest overrun request is processed and ADC_FLWCTL[RSTA] is set again one cycle later.

- LoadOK overrun:

  Simultaneously at any restart request overrun situation, the LoadOK input is evaluated and the status is captured in a background register which is alternated anytime a restart request overrun occurs while Load OK request is asserted. The Load OK background register is cleared as soon as the pending Restart Request gets processed.

- Trigger overrun:

  If a trigger occurs while ADC_FLWCTL[TRIG] is already set, this is defined as a trigger overrun situation and causes the ADC to cease conversion at the next conversion boundary and to set ADC_EIF[TRIG_EIF]. A overrun is also detected if the trigger event occurs automatically generated by hardware in "Trigger mode" due to a restart event and simultaneously a trigger event is generated via data bus or internal interface. In this case, the ADC ceases operation before conversion begins to sample. In "Trigger mode", a restart request overrun does not cause a trigger overrun (ADC_EIF[TRIG_EIF] not set).

- Sequence abort request overrun:

  If a sequence abort request occurs while ADC_FLWCTL[SEQA] is already set, this is defined as a sequence abort request overrun situation and the overrun request is ignored.

22.7.3.3 ADC list usage and conversion/conversion sequence flow description

The different lists must not overlap or exceed the system RAM area respectively. The CSL must not exceed the NVM area if located in the NVM. The error flag ADC_EIF[IA_EIF] will be set for accesses done outside the system RAM area and will cause an error interrupt if enabled for lists that are located in the system RAM.

Generic flow for ADC register load at conversion sequence start/restart:

- It is mandatory that the ADC is idle (no ongoing conversion or conversion sequence).
- It is mandatory to have at least one CSL with valid entries. See also Restart CSL execution with currently active CSL or Restart CSL execution with newother CSL (alternative CSL becomes active CSL) - CSL swapping for more details on possible scenarios.
- A restart event occurs, which causes the index registers to be cleared (register ADC_CIDX and ADC_RIDX are cleared) and to point to the top of the corresponding lists (top of active RVL and CSL).
- Load conversion command to background conversion command register 1.
- The control bit(s) ADC_FLWCTL[RSTA] (and ADC_FLWCTL[LDOK] if set) are cleared.
- Wait for trigger event to start conversion.

Generic flow for ADC register load during conversion:

- The index registers ADC_CIDX is incremented.
- The inactive background command register is loaded with a new conversion command.

Generic flow for ADC result storage at end of conversion:

- Index register ADC_RIDX is incremented and the conversion result is stored in system RAM. As soon as the result is successfully stored, any conversion interrupt flags are set accordingly.
- At the conversion boundary, the other background command register becomes active and visible in the ADC register map.
- If the last executed conversion command was of type "End Of Sequence", the ADC waits for the trigger event.
- If the last executed conversion command was of type "End Of List" and the ADC is configured in "Restart mode", the ADC sets all related flags and stays idle awaiting a restart event to continue.

- If the last executed conversion command was of type "End Of List" and the ADC is configured in "Trigger mode", the ADC sets all related flags and automatically returns to top of current CSL and is awaiting a trigger event to continue.
- If the last executed conversion command was of type "Normal Conversion", the ADC continues command execution in the order of the current CSL (continues conversion).

22.8 Resets

At reset, the ADC is disabled and in a power down state.

22.9 Interrupts

The ADC supports the following types of interrupts:

- Conversion interrupt
- Sequence abort interrupt
- Error and conversion flow control issue interrupt

Each of the interrupt types is associated with individual interrupt enable bits and interrupt flags.

22.9.1 ADC conversion interrupt

The ADC provides one conversion interrupt associated to 16 interrupt enable bits with dedicated interrupt flags. The 16 interrupt flags consist of:

- 15 conversion interrupt flags which can be associated to any conversion completion.
- One additional interrupt flag which is fixed to the "End Of List" conversion command type within the active CSL.

The association of the conversion number with the interrupt flag number is done in the conversion command.

22.9.2 ADC sequence abort done interrupt

The ADC provides one sequence abort done interrupt associated with the sequence abort request for conversion flow control. Hence, there is only one dedicated interrupt flag and interrupt enable bit for conversion sequence abort and it occurs when the sequence abort is done.

22.9.3 ADC error and conversion flow control issue interrupt

The ADC provides one error interrupt for the following error classes related to conversion interrupt overflow, command validness, DMA access status and conversion flow control issues, and CSL failure. The following error interrupt flags belong to the group of severe issues which cause an error interrupt if enabled and cease ADC operation:

- WA_EIF
- RA_EIF
- CMD_EIF
- EOL_EIF
- TRIG_EIF

In order to make the ADC operational again, an ADC soft-reset must be issued which clears the above listed error interrupt flags.

Remaining error interrupt flags cause an error interrupt if enabled, but ADC continues operation. The related interrupt flags are:

- RSTAR_EIF
- LDOK_EIF
- CONIF_OIF

22.10 Use cases and application information

22.10.1 List Usage - CSL single buffer mode and RVL single buffer mode

In this use case, both list types are configured for single buffer mode (ADC_CTL1[CSL_BMOD]=0b and ADC_CTL1[RVL_BMOD]=0b, ADC_STS[CSL_SEL] and ADC_STS[RVL_SEL] are forced to 0b). The index register for the CSL and RVL are cleared to start from the top of the list with next conversion command and result storage in the following cases:

- The conversion flow reaches the command containing the "End-of-List" command type identifier
- A restart request occurs at a sequence boundary
- After an aborted conversion or conversion sequence

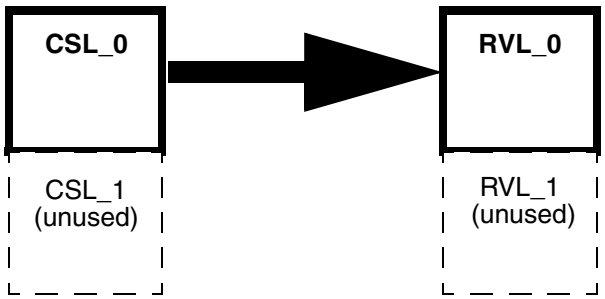


Figure 22-39. CSL single buffer mode - RVL single buffer mode diagram

22.10.2 List usage - CSL single buffer mode and RVL double buffer mode

In this use case, the CSL is configured for single buffer mode (ADC_CTL1[CSL_BMOD]=0b) and the RVL is configured for double buffer mode (ADC_CTL1[RVL_BMOD]=1b). In this buffer configuration, only the result list RVL is switched when the first conversion result of a CSL is stored after a CSL was successfully finished or a CSL got aborted.

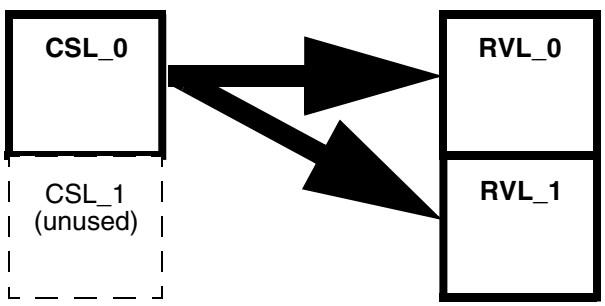


Figure 22-40. CSL single buffer mode - RVL double buffer mode diagram

The last entirely filled RVL (an RVL where the corresponding CSL has been executed including the "End Of List " command type) is shown by register ADC_EOLRI.

The CSL is used in single buffer mode and bit ADC_STS[CSL_SEL] is forced to 0b.

22.10.3 List usage - CSL double buffer mode and RVL double buffer mode

In this use case, both list types are configured for double buffer mode (ADC_CTL1[CSL_BMOD]=1b and ADC_CTL1[RVL_BMOD]=1b) and whenever a command sequence list (CSL) is finished or aborted the command sequence list is swapped by the simultaneous assertion of ADC_FLWCTL[LDOK] and ADC_FLWCTL[RSTA].

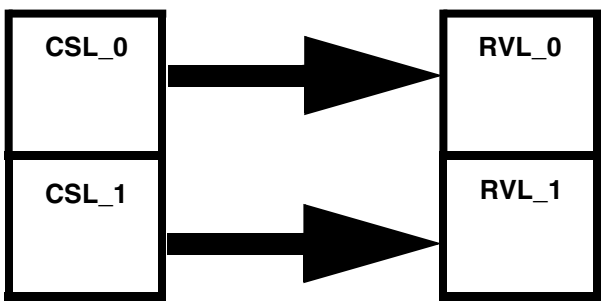


Figure 22-41. CSL double buffer mode - RVL double buffer mode diagram

This use case can be used if the channel order or CSL length varies very frequently in an application.

22.10.4 List usage - CSL double buffer mode and RVL single buffer mode

In this use case, the CSL is configured for double buffer mode (ADC_CTL1[CSL_BMOD]=1b) and the RVL is configured for single buffer mode (ADC_CTL1[RVL_BMOD]=0b).

The two command lists can be different sizes and the allocated result list memory area in the RAM must be able to hold as many entries as the larger of the two command lists. Each time when the end of a Command Sequence List is reached, if ADC_FLWCTL[LDOK] and ADC_FLWCTL[RSTA] are set, the commands list is swapped.

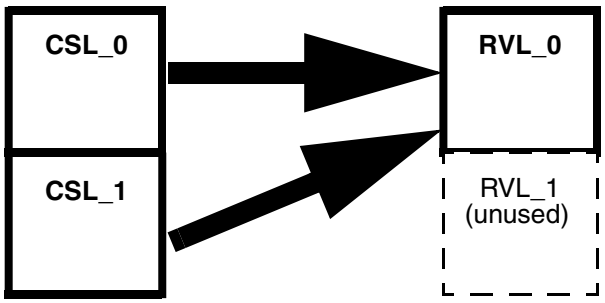


Figure 22-42. CSL double buffer mode - RVL single buffer mode diagram

22.10.5 List usage - CSL double buffer mode and RVL double buffer mode

In this use case, both list types are configured for double buffer mode (ADC_CTL1[CSL_BMOD]=1b) and ADC_CTL1[RVL_BMOD]=1b).

This setup is the same asList usage - CSL double buffer mode and RVL double buffer mode , but at the end of a CSL the CSL is not always swapped (ADC_FLWCTL[LDOK] not always set with ADC_FLWCTL[RSTA]). The result value list is swapped whenever a CSL is finished or a CSL got aborted

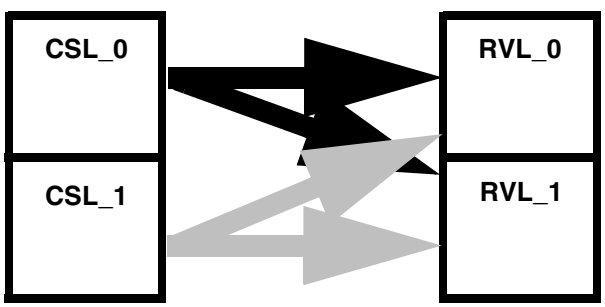
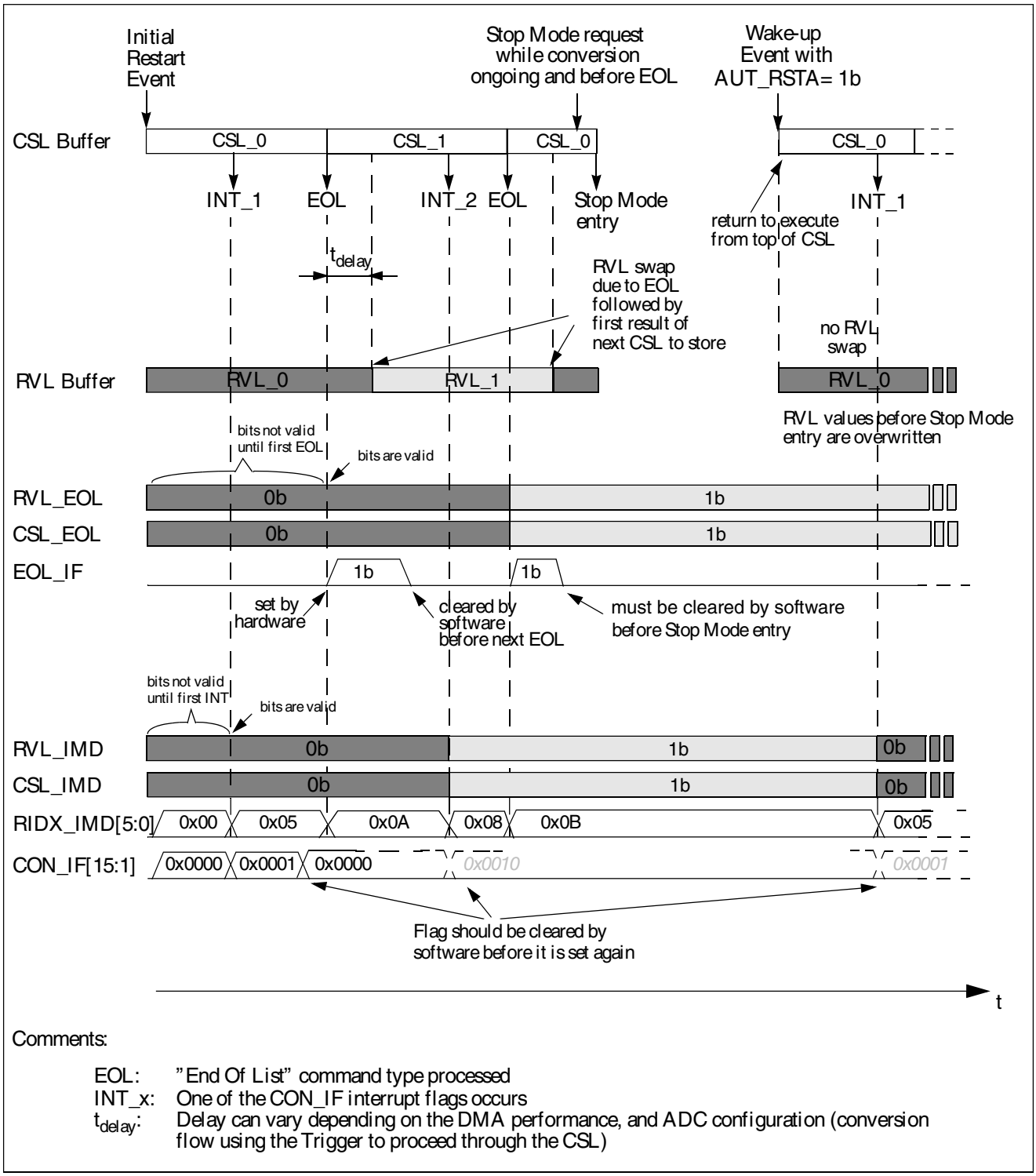


Figure 22-43. CSL double buffer mode - RVL double buffer mode diagram

22.10.6 RVL swapping in RVL double buffer mode and related registers ADC_IMDRI and ADC_EOLRI

When using the RVL in double buffer mode, the registers ADC_IMDRI and ADC_EOLRI can be used by the application software to identify which RVL holds relevant and latest data and which CSL is related to this data. These registers are updated at the setting of one of the ADC_CONIF0[CON_IF], ADC_CONIF1[CON_IF] or the ADC_CONIF1[EOL_IF] interrupt flags. The register ADC_IMDRI, for instance, is always updated at the occurrence of a ADC_CONIF0[CON_IF] and

ADC_CONIF1[CON_IF] interrupt flag amongst other cases. Also each time the last conversion command of a CSL is finished and the corresponding result is stored, the related ADC_CONIF1[EOL_IF] flag is set and register ADC_EOLRI is updated. Hence application software can pick up conversion results, or groups of results, or an entire result list driven fully by interrupts. A use case example diagram is shown in the following figure.



Comments:

EOL: "End Of List" command type processed
INT_x: One of the CON_IF interrupt flags occurs
t_{delay}: Delay can vary depending on the DMA performance, and ADC configuration (conversion flow using the Trigger to proceed through the CSL)

Figure 22-44. RVL swapping - use case diagram

22.10.7 Conversion flow control application information

The ADC provides various conversion control scenarios to the user accomplished by the following features.

The ADC conversion flow control can be realized via the data bus only, the internal interface only, or by both access methods. The method used is software configurable via ADCCTL0[ACC_CFG].

The conversion flow is controlled via the four conversion flow control bits: ADC_FLWCTL[SEQA], ADC_FLWCTL[TRIG], ADC_FLWCTL[RSTA], and ADC_FLWCTL[LDOK].

Two different conversion flow control modes can be configured: Trigger mode or Restart mode

Single or double buffer configuration of CSL and RVL.

22.10.7.1 Initial start of a command sequence list

At the initial start of a command sequence list after device reset, all entries for at least one of the two CSL must have been completed and data must be valid. Depending on if the CSL_0 or the CSL_1 is executed at the initial start of a command sequence list, the following conversion control sequence must be applied:

- If CSL_0 is be executed at the initial conversion start after device reset:

  A restart event and a trigger event must occur (depending to the selected conversion flow control mode, the events must occur one after the other or simultaneously) which causes the ADC to start conversion with commands loaded from CSL_0.

- If CSL_1 is be executed at the initial conversion start after device reset:

  ADC_FLWCTL[LDOK] must be set simultaneously with the restart event followed by a trigger event (depending on the selected conversion flow control mode, the trigger events must occur simultaneously or after the restart event is finished). As soon as the trigger event gets executed, the ADC starts conversion with commands loaded from CSL_1.

As soon as a new valid restart event occurs, the flow for ADC register load at conversion sequence start as described in ADC list usage and conversion/conversion sequence flow description applies.

22.10.7.2 Restart CSL execution with currently active CSL

To restart a command sequence list execution, it is mandatory that the ADC is idle (no conversion or conversion sequence is ongoing).

If necessary, a possible ongoing conversion sequence can be aborted by the sequence abort event (setting bit ADC_FLWCTL[SEQA]). As soon as ADC_FLWCTL[SEQA] is cleared by the ADC, the current conversion sequence has been aborted and the ADC is idle (no conversion sequence or conversion ongoing).

After a conversion sequence abort is executed, it is mandatory to request a restart event (ADC_FLWCTL[RSTA] is set). After the restart event is finished (ADC_FLWCTL[RSTA] is cleared), the ADC accepts a new trigger event (ADC_FLWCTL[TRIG] can be set) and begins conversion from the top of the currently active CSL. In conversion flow control mode "Trigger mode", only a restart event is necessary if ADC is idle to restart conversion sequence list execution (the trigger event occurs automatically).

It is possible to set ADC_FLWCTL[RSTA] and ADC_FLWCTL[SEQA] simultaneously, causing a sequence abort event followed by a restart event. In this case, the error flags behave differently depending on the selected conversion flow control mode:

- Setting both flow control bits simultaneously in conversion flow control mode "Restart mode" prevents the error flags ADC_EIF[RSTAR_EIF] and ADC_EIF[LDOK_EIF] from occurring.
- Setting both flow control bits simultaneously in conversion flow control mode "Trigger mode" prevents the error flag ADC_EIF[RSTAR_EIF] from occurring.

If only a restart event occurs while ADC is not idle and bit ADC_FLWCTL[SEQA] is not set already (sequence abort event in progress) a sequence abort event is issued automatically and ADC_EIF[RSTAR_EIF] is set.

See also the detailed conversion flow control bit mandatory requirements and execution information for ADC_FLWCTL[RSTA] and ADC_FLWCTL[SEQA] described in The four ADC conversion flow control bits .

22.10.7.3 Restart CSL execution with newother CSL (alternative CSL becomes active CSL) - CSL swapping

After all alternative conversion command list entries are finished the bit LDOK can be set simultaneously with the next Restart Event to swap command buffers.

To start conversion command list execution it is mandatory that the ADC is idle (no conversion or conversion sequence is ongoing).

If necessary, a possible ongoing conversion sequence can be aborted by the Sequence Abort Event (setting bit SEQA). As soon as bit SEQA is cleared by the ADC, the current conversion sequence has been aborted and the ADC is idle (no conversion sequence or conversion ongoing).

After a conversion sequence abort is executed it is mandatory to request a Restart Event (bit RSTA set) and simultaneously set bit LDOK to swap the CSL buffer. After the Restart Event is finished (bit RSTA and LDOK are cleared), the ADC accepts a new Trigger Event (bit TRIG can be set) and begins conversion from the top of the newly selected CSL buffer. In conversion flow control mode "Trigger Mode" only a Restart Event (simultaneously with bit LDOK being set) is necessary to restart conversion command list execution with the newly selected CSL buffer (the Trigger Event occurs automatically).

It is possible to set bits RSTA, LDOK and SEQA simultaneously, causing a Sequence Abort Event followed by a Restart Event. In this case the error flags behave differently depending on the selected conversion flow control mode:

- Setting these three flow control bits simultaneously in "Restart Mode" prevents the error flags RSTA_EIF and LDOK_EIF from occurring.
- Setting these three flow control bits simultaneously in "Trigger Mode" prevents the error flag RSTA_EIF from occurring.

If only a Restart Event occurs while ADC is not idle and bit SEQA is not set already (Sequence Abort Event in progress) a Sequence Abort Event is issued automatically and bit RSTAR_EIF is set.

Please see also the detailed conversion flow control bit mandatory requirements and execution information for bit RSTA and SEQA described in The four ADC conversion flow control bits .

22.10.8 Continuous conversion

Applications that only need to continuously convert a list of channels, without the need for timing control or the ability to perform different sequences of conversions (grouped number of different channels to convert) can make use of the following simple setup:

- "Trigger mode" configuration
- Single buffer CSL
- Depending on data transfer rate either use single or double buffer RVL configuration
- Define a list of conversion commands which only contains the "End Of List" command with automatic wrap to top of CSL

After finishing the configuration and enabling the ADC, an initial restart event is sufficient to launch the continuous conversion until next device reset or low power mode.

In case a low power mode is used:

If ADC_CTL1[AUT_RSTA] is set before low power mode is entered, the conversion continues automatically as soon as a low power mode (Stop mode or Wait mode with bit ADC_CTL0[SWAI] set) is exited.

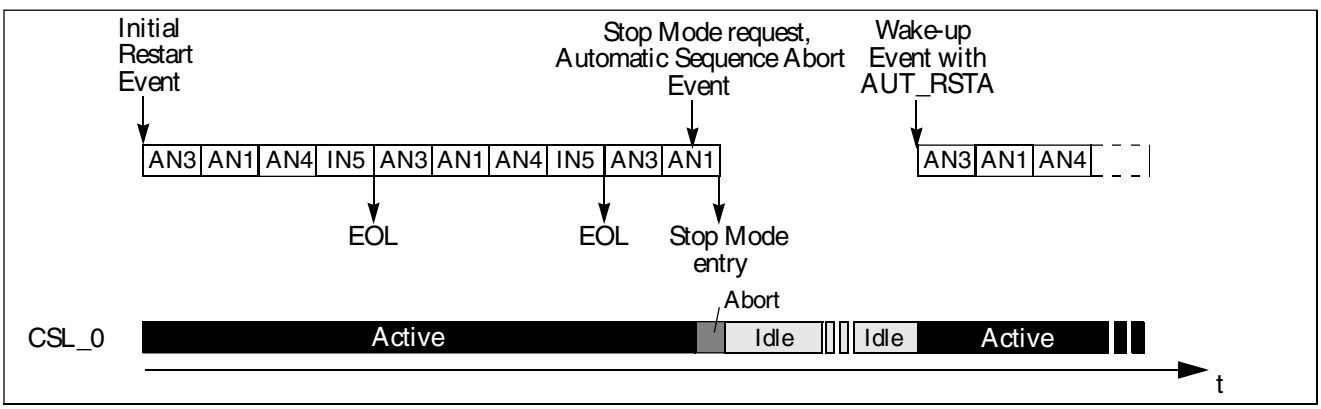


Figure 22-45. Conversion flow control diagram - continuous conversion (with Stop mode)

22.10.9 Triggered conversion - single CSL

Applications that require the conversion of one or more groups of different channels in a periodic and timed manner can make use of a configuration in "Trigger mode" with a single CSL containing a list of sequences. This means the CSL consists of several sequences, each separated by an "End of Sequence" command. The last command of the CSL uses the "End Of List" command with wrap to top of CSL and waiting for a trigger (ADC_CMD0[CMD_SEL] =11b). Hence after the initial restart event, each sequence can be launched via a trigger event and repetition of the CSL can be launched via a trigger after execution of the "End Of List" command.

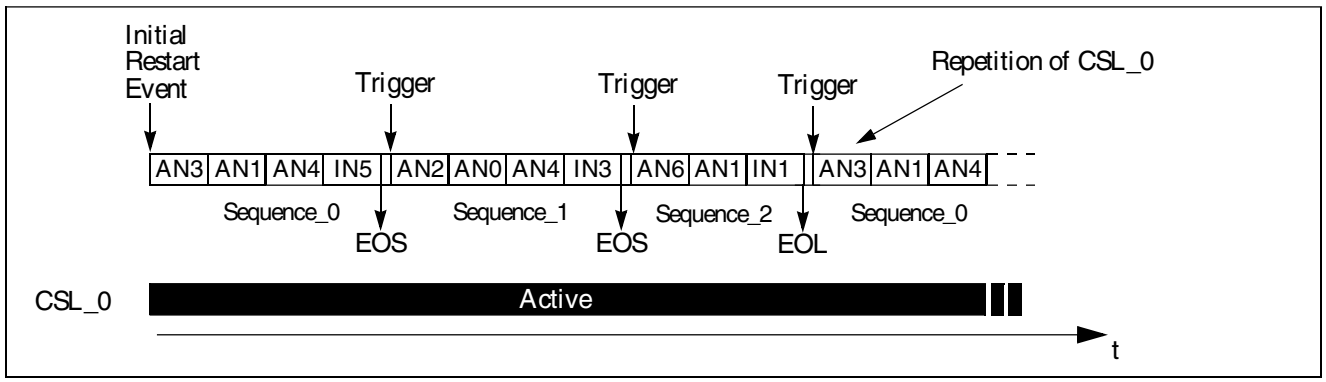


Figure 22-46. Conversion flow control diagram - triggered conversion (CSL repetition)

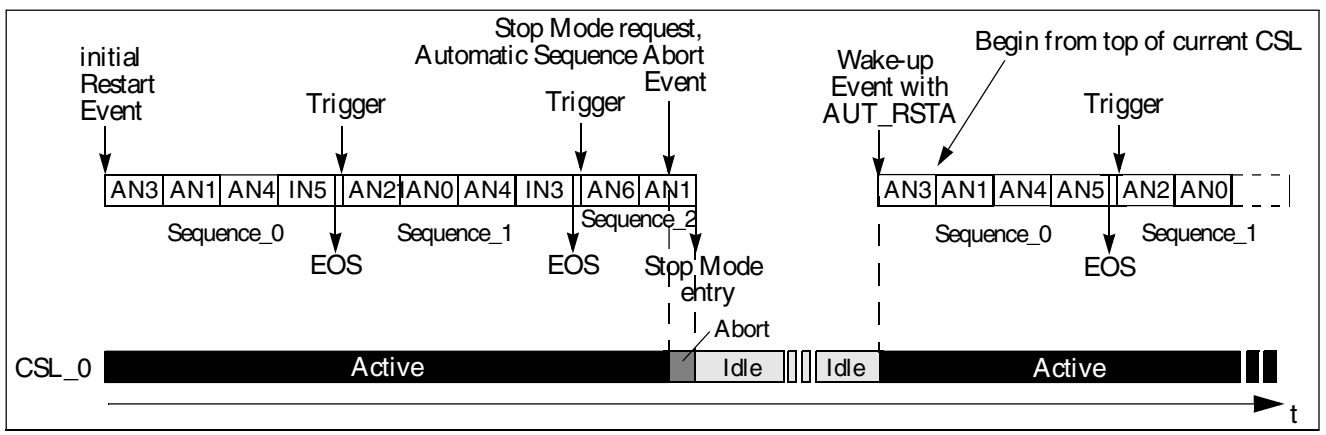


Figure 22-47. Conversion flow control diagram - triggered conversion (with Stop Mode)

In case a low power mode is used:

If ADC_CTL1[AUT_RSTA] is set before low power mode is entered, the conversion continues automatically as soon as a low power mode (Stop mode or Wait mode with ADC_CTL0[SWAI] set) is exited.

22.10.10 Fully timing controlled conversion

As described previously, in "Trigger mode" a restart event automatically causes a trigger. To have full and precise timing control of the beginning of any conversion/sequence, the "Restart mode" is available. In "Restart Mode", a restart event does not cause a trigger automatically; instead, the trigger must be issued separately and with correct timing, which means the trigger is not allowed before the restart event (conversion command loading) is finished (ADC_FLWCTL[RSTA]=0b again). The time required from trigger until sampling phase starts is given (see ADC_FLWCTL, timing considerations) and

hence timing is fully controllable by the application. Additionally, if a trigger occurs before a restart event is finished, this causes the ADC_EIF[TRIG_EIF] flag being set. This allows detection of false flow control sequences.

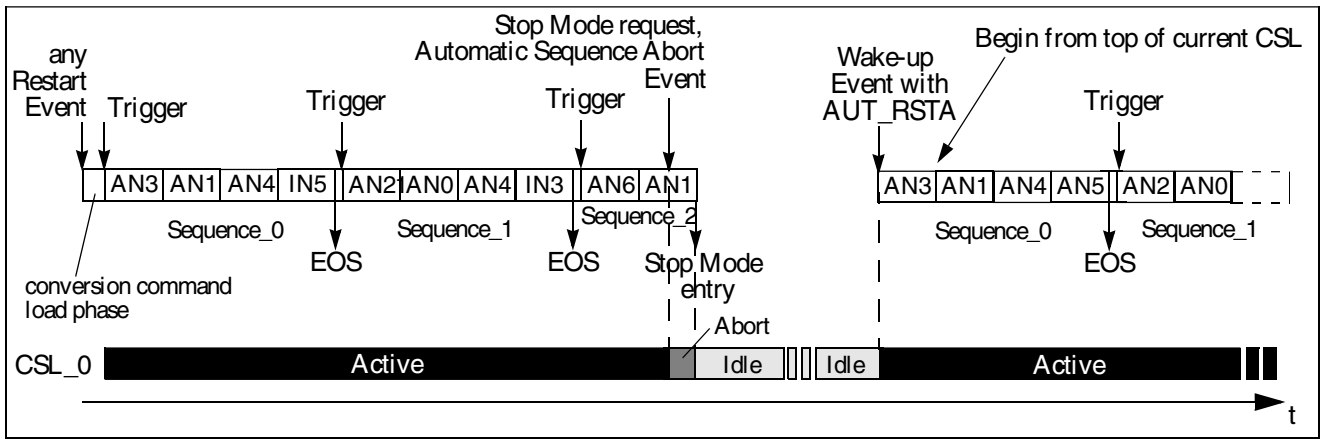


Figure 22-48. Conversion flow control diagram - fully timing controlled conversion (with Stop mode)

Unlike the Stop mode entry shown in Figure 22-47 and Figure 22-48, it is recommended to issue the Stop mode at sequence boundaries (when ADC is idle and no conversion/ conversion sequence is ongoing).

Any of the conversion flow control application use cases described above (continuous, triggered, or fully timing controlled conversion) can be used with CSL single buffer mode or with CSL double buffer mode. If using CSL double buffer mode, CSL swapping is performed by issuing a restart event with ADC_FLWCTL[LDOK] set.

Chapter 23
Analog comparator (ACMP)

23.1 Chip-specific ACMP information

23.1.1 ACMP interconnections

The ACMP0 output can be configured to connect to other on-chip modules through crossbar

The following table shows the input connections to the ACMP0:

Table 23-1. ACMP0 input connections

ACMP0 channel	Connection
0	PTB2
1	PTB3
2	7/10 scaled VOUT, 1/8 Scaled VRECT, 1/5 scaled Vad_in, 1/10 scaled AC1
3	DAC output

User can select the ACMP0 channel 2 source by configuring the SIM_CR[ACMP0CH2SEL].

23.2 Introduction

The analog comparator module (ACMP) provides a circuit for comparing two analog input voltages. The comparator circuit is designed to operate across the full range of the supply voltage (rail-to-rail operation).

The analog mux provides a circuit for selecting an analog input signal from four channels. One signal provided by the 6-bit 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 level. A 6-bit digital signal input selects output voltage level, which varies from V_{in} to $V_{in}/64$. V_{in} can be selected from two voltage sources.

23.2.1 Features

ACMP features include:

- Operational over the whole supply range of 2.7 V to 5.5 V
- On-chip 6-bit resolution DAC with selectable reference voltage from V_{DD} or internal VREFH
- Configurable hysteresis
- Selectable interrupt on rising edge, falling edge, or both rising or falling edges of comparator output
- Selectable inversion on comparator output
- Up to four selectable comparator inputs

23.2.2 Modes of operation

This section defines the ACMP operation in Wait, Stop, and Background Debug modes.

23.2.2.1 Operation in Wait mode

The ACMP continues to operate in Wait mode, if enabled. The interrupt can wake the MCU if enabled.

23.2.2.2 Operation in Debug mode

When the MCU is in Debug mode, the ACMP continues operating normally.

23.2.3 Block diagram

The block diagram of the ACMP module is shown in the following figure.

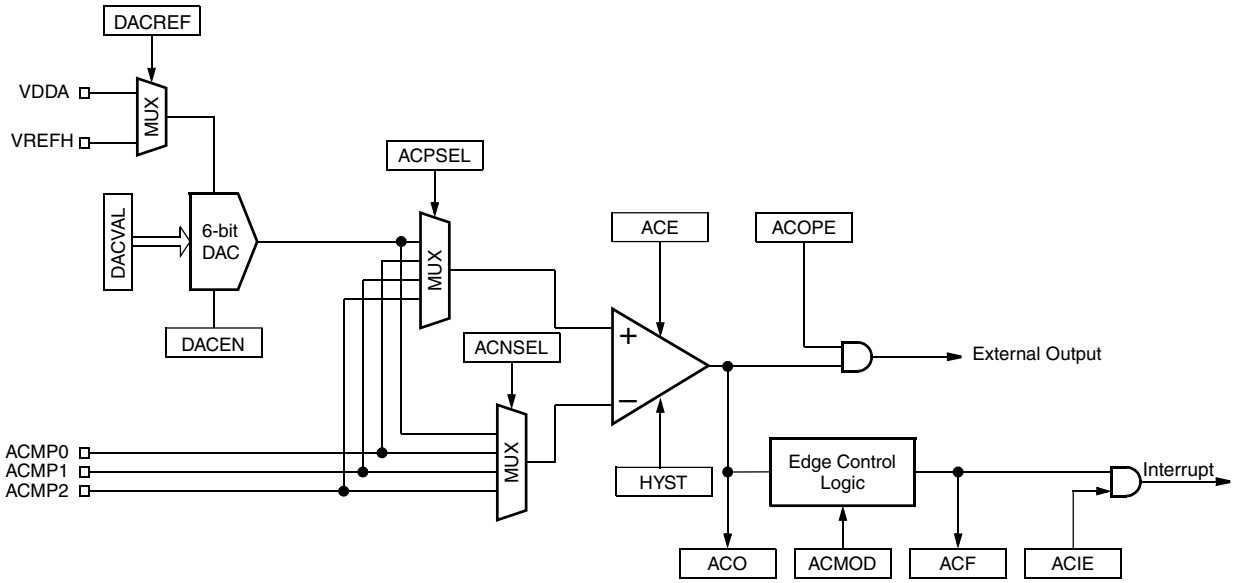


Figure 23-1. ACMP block diagram

23.3 External signal description

The output of ACMP can also be mapped to an external pin. When the output is mapped to an external pin, ACMP_CS[ACOPE] controls the pin to enable/disable the ACMP output function.

23.4 Memory map and register definition

ACMP memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4007_3000	ACMP Control and Status Register (ACMP0_CS)	8	R/W	00h	23.4.1/318
4007_3001	ACMP Control Register 0 (ACMP0_C0)	8	R/W	00h	23.4.2/319
4007_3002	ACMP Control Register 1 (ACMP0_C1)	8	R/W	00h	23.4.3/319
4007_3003	ACMP Control Register 2 (ACMP0_C2)	8	R/W	00h	23.4.4/320

23.4.1 ACMP Control and Status Register (ACMPx_CS)

Address: 4007_3000h base + 0h offset = 4007_3000h

Bit	7	6	5	4	3	2	1	0
Read	ACE	HYST	ACF	ACIE	ACO	ACOPE	ACMOD	
Write								
Reset	0	0	0	0	0	0	0	0

ACMPx_CS field descriptions

Field	Description
7 ACE	Analog Comparator Enable Enables the ACMP module. 0 The ACMP is disabled. 1 The ACMP is enabled.
6 HYST	Analog Comparator Hysterisis Selection Selects ACMP hysterisis. 0 20 mV. 1 30 mV.
5 ACF	ACMP Interrupt Flag Bit Synchronously set by hardware when ACMP output has a valid edge defined by ACMOD. The setting of this bit lags the ACMPO to bus clocks. Clear ACF bit by writing a 0 to this bit. Writing a 1 to this bit has no effect.
4 ACIE	ACMP Interrupt Enable Enables an ACMP CPU interrupt. 0 Disable the ACMP Interrupt. 1 Enable the ACMP Interrupt.
3 ACO	ACMP Output Reading ACO will return the current value of the analog comparator output. ACO is reset to a 0 and will read as a 0 when the ACMP is disabled (ACE = 0)
2 ACOPE	ACMP Output Pin Enable ACOPE enables the pad logic so that the output can be placed onto an external pin. 0 ACMP output cannot be placed onto external pin. 1 ACMP output can be placed onto external pin.
ACMOD	ACMP MOD Determines the sensitivity modes of the interrupt trigger. 00 ACMP interrupt on output falling edge. 01 ACMP interrupt on output rising edge.

Table continues on the next page...

ACMPx_CS field descriptions (continued)

Field	Description
	10 ACMP interrupt on output falling edge. 11 ACMP interrupt on output falling or rising edge.

23.4.2 ACMP Control Register 0 (ACMPx_C0)

Address: 4007_3000h base + 1h offset = 4007_3001h

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

ACMPx_C0 field descriptions

Field	Description
7–6 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
5–4 ACPSEL	ACMP Positive Input Select 00 External reference 0 01 External reference 1 10 External reference 2 11 DAC output
3–2 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
ACNSEL	ACMP Negative Input Select 00 External reference 0 01 External reference 1 10 External reference 2 11 DAC output

23.4.3 ACMP Control Register 1 (ACMPx_C1)

Address: 4007_3000h base + 2h offset = 4007_3002h

Bit	7	6	5	4	3	2	1	0
Read Write	DACEN	DACREF	DACVAL					
Reset	0	0	0	0	0	0	0	0

ACMPx_C1 field descriptions

Field	Description
7 DACEN	DAC Enable Enables the output of 6-bit DAC. 0 The DAC is disabled. 1 The DAC is enabled.
6 DACREF	DAC Reference Select 0 The DAC selects VREFH as the reference. 1 The DAC selects V_{DDA} as the reference.
DACVAL	DAC Output Level Selection Selects the output voltage using the given formula: V_{output}= (V_{in}/64)x(DACVAL[5:0]+1) The V_{output} range is from V_{in}/64 to V_{in}, the step is V_{in}/64

23.4.4 ACMP Control Register 2 (ACMPx_C2)

Address: 4007_3000h base + 3h offset = 4007_3003h

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

ACMPx_C2 field descriptions

Field	Description
7–3 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
ACIPE	ACMP Input Pin Enable This 3-bit field controls if the corresponding ACMP external pin can be driven by an analog input. 0 The corresponding external analog input is not allowed. 1 The corresponding external analog input is allowed.

23.5 Functional description

The ACMP module is functionally composed of two parts: digital-to-analog (DAC) and comparator (CMP).

The DAC includes a 64-level DAC (digital to analog converter) and relevant control logic. DAC can select one of two reference inputs, V_{DD} or on-chip VREFH, as the DAC input V_{in} by setting ACMP_C1[DACREF]. After the DAC is enabled, it converts the data

set in ACMP_C1[DACVAL] to a stepped analog output, which is fed into ACMP as an internal reference input. This stepped analog output is also mapped out of the module. The output voltage range is from $V_{in}/64$ to V_{in}. The step size is $V_{in}/64$.

The ACMP can achieve the analog comparison between positive input and negative input, and then give out a digital output and relevant interrupt. Both the positive and negative input of ACMP can be selected from the four common inputs: three external reference inputs and one internal reference input from the DAC output. The positive input of ACMP is selected by ACMP_C0[ACPSEL] and the negative input is selected by ACMP_C0[ACNSEL]. Any pair of the eight inputs can be compared by configuring the ACMPC0 with the appropriate value.

After the ACMP is enabled by setting ACMP_CS[ACE], the comparison result appears as a digital output. Whenever a valid edge defined in ACMP_CS[ACMOD] occurs, ACMP_CS[ACF] is asserted. If ACMP_CS[ACIE] is set, a ACMP CPU interrupt occurs. The valid edge is defined by ACMP_CS[ACMOD]. When ACMP_CS[ACMOD] = 00b or 10b, only the falling-edge on ACMP output is valid. When ACMP_CS[ACMOD] = 01b, only rising-edge on ACMP output is valid. When ACMP_CS[ACMOD] = 11b, both the rising-edge and falling-edge on the ACMP output are valid.

The ACMP output is synchronized by the bus clock to generate ACMP_CS[ACO] so that the CPU can read the comparison. ACMP_CS[ACO] changes following the comparison result, so it can serve as a tracking flag that continuously indicates the voltage delta on the inputs.

If a reference input external to the chip is selected as an input of ACMP, the corresponding ACMP_C2[ACIPE] bit must be set to enable the input from pad interface. If the output of the ACMP needs to be put onto the external pin, the ACMP_CS[ACOPE] bit must enable the ACMP pin function of pad logic.

23.6 Setup and operation of ACMP

The two parts of ACMP (DAC and CMP) can be set up and operated independently. But if the DAC works as an input of the CMP, the DAC must be configured before the ACMP is enabled.

Because the input-switching can cause problems on the ACMP inputs, the user should complete the input selection before enabling the ACMP and must not change the input selection setting when the ACMP is enabled to avoid unexpected output. Similarly, because the DAC experiences a setup delay after ACMP_C1[DACVAL] is changed, the user should complete the setting of ACMP_C1[DACVAL] before DAC is enabled.

23.7 Resets

During a reset the ACMP is configured in the default mode. Both CMP and DAC are disabled.

23.8 Interrupts

If the bus clock is available when a valid edge defined in ACMP_CS[ACMOD] occurs, the ACMP_CS[ACF] is asserted. If ACMP_CS[ACIE] is set, a ACMP interrupt event occurs. The ACMP_CS[ACF] bit remains asserted until the ACMP interrupt is cleared by software.The interrupt can be cleared by writing a 0 to the ACMP_CS[ACF] bit.

Chapter 24
Programmable Gain Amplifier (PGA)

24.1 Chip-specific PGA information

24.1.1 Differential input amplifier channel assignment

The differential input amplifier input is the voltage drop on the external R_{sens}. One side of the sense resistor is connected to the pin ISENS and the other side is connected to the pin VOUT.

The differential input amplifier also has a differential output pair V_{op} and V_{on}, which are separately connected to 2 ADC input channels: internal_6 and internal_7.

The ADC should read these 2 channel inputs with minimum interval. The amplified voltage drop is the subtraction of the two channel conversion results.

When no current passes through the external R_{sens}, the subtraction of the V_{op} and V_{on} conversion results can be treated as the system offset. Software could use this offset to calibrate the later amplified result. This offset is tested and stored during the factory test. User can refer to SIM_IFR*x* registers for more information.

24.2 Introduction

The programmable gain amplifier (PGA) module enables the MCU to achieve high-accuracy current measurement. It can amplify the small voltage drop on an external current sensing resistor. Its two differential outputs can be converted by the on-chip ADC. The PGA gain can be configured to 8, 10, 15 or 20.

24.3 Features

PGA includes the following features:

- Programmable gain: ×8, ×10, ×15 or ×20
- Differential inputs from two inputs across the external current sensing resistor
- Differential outputs to two ADC input channels
- Input offset voltage can be calibrated by software

24.4 Overview

This section presents an overview of the PGA module. The following figure illustrates the simplified PGA block diagram.

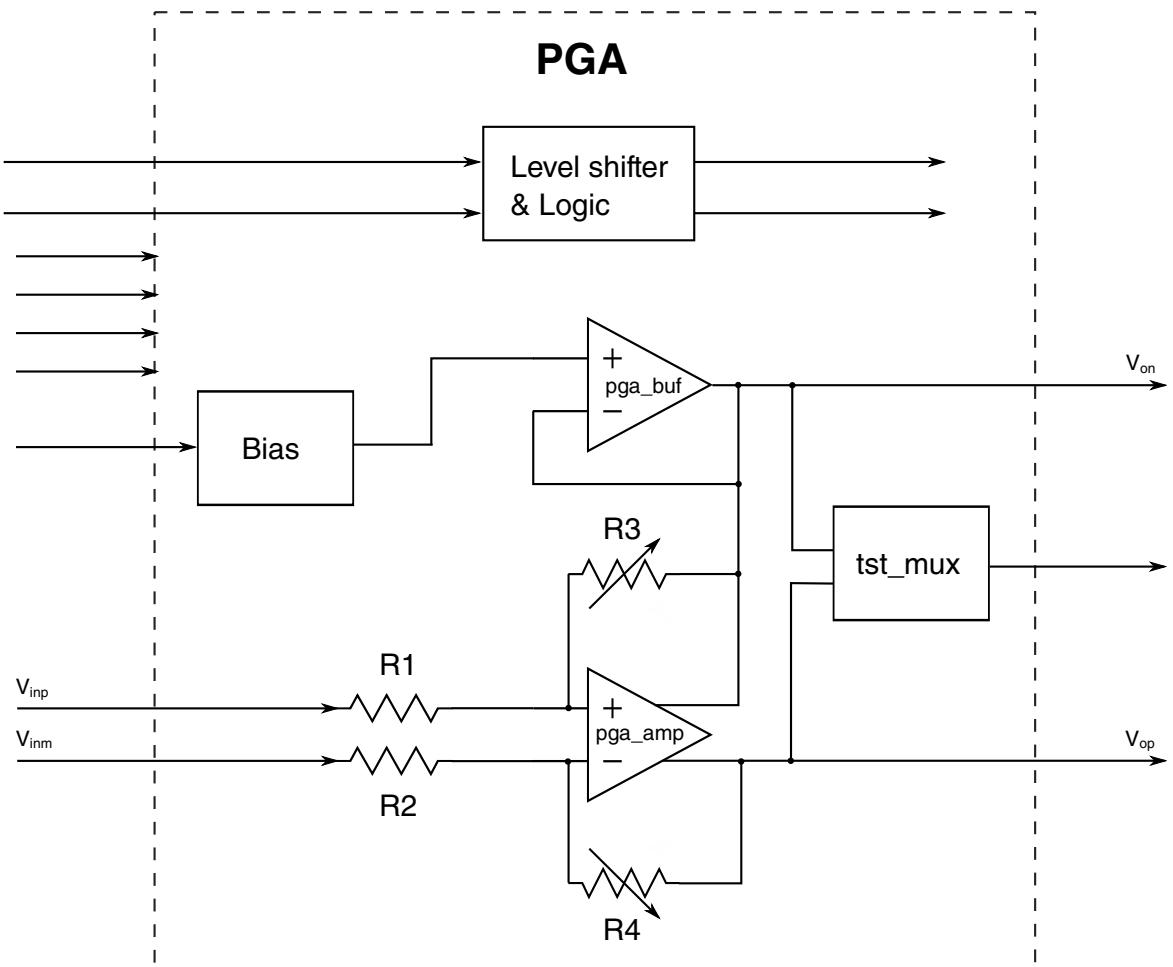


Figure 24-1. Simplified PGA block diagram

24.4.1 PGA application on high-accuracy current sensing

The typical application of PGA is shown in the following figure.

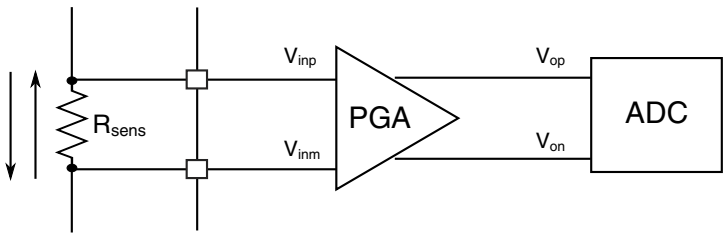


Figure 24-2. PGA application on high-accuracy current sensing

The PGA input is the voltage drop on the external current sensing resistor R_{sens}. The PGA also has a differential output pair vop and von, which can be separately connected to two ADC input channels. To correctly get the PGA output, the ADC should read these two channel inputs with a minimum interval. The amplified voltage drop is the difference of the two channel ADC conversion results.

The differential inputs and outputs of PGA enable user to measure the current passing through the sensing resistor, not only the current magnitude but also the current flow direction.

24.5 Modes of operation

PGA supports two operating modes: the Disabled mode and the Active mode.

24.5.1 Disabled mode

When the enable bit PGA_CTRL[PGAEN] is cleared, the PGA module is disabled and does not perform any function in any MCU operational mode. When entering the Stop mode, the PGA module is disabled automatically.

24.5.2 Active mode

In Active mode, PGA has its full functionality. PGA can be in Active mode with the MCU in any of the following operational modes: Run and Wait.

24.6 External signal description

The following table itemizes all the PGA external pins.

Signal	I/O	Power domain	Detailed description	Connect from/to
V_{inp}	I	5 V	Voltage input plus, across the current sensing resistor	From the sensing resistor
V_{inm}	I	5 V	Voltage input minus, across the current sensing resistor	From the sensing resistor
V_{op}	O	5 V	Voltage output plus	To ADC
V_{on}	O	5 V	Voltage output minus	To ADC

24.6.1 V_{inp} and V_{inm}

V_{inp} and V_{inm} are the two inputs across the current sensing resistor. The current sensing resistor should be connected to PGA through pads.

24.6.2 V_{op} and V_{on}

V_{op} and V_{on} are the two PGA amplified voltage outputs. These two outputs can be connected to two ADC input channels.

24.7 Memory map and register definition

This section includes the module memory map and detailed descriptions of all registers.

PGA memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4007_1000	Control Register (PGA_CTRL)	8	R/W	00h	24.7.1/327

24.7.1 Control Register (PGA_CTRL)

Address: 4007_1000h base + 0h offset = 4007_1000h

Bit	7	6	5	4	3	2	1	0
Read	0					GAIN[1:0]		PGAEN
Write								
Reset	0	0	0	0	0	0	0	0

PGA_CTRL 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 GAIN[1:0]	Programmable Gain These two bits are used to configure the amplifier's gain. 00 The gain is 8. 01 The gain is 10. 10 The gain is 15. 11 The gain is 20.
0 PGAEN	Programmable Gain Amplifier Enable This bit is the enable control signal to the programmable amplifier. **NOTE:** In the Stop mode, PGA is disabled automatically. 0 Disables the programmable amplifier. 1 Enables the programmable amplifier.

24.8 Functional description

The following sections describe functional details of the module.

24.8.1 PGA gain control

The PGA gain can be configured through GAIN[1:0] bits in the PGA_CTRL register.

- GAIN[1:0] = 0b00, Gain = 8
- GAIN[1:0] = 0b01, Gain = 10
- GAIN[1:0] = 0b10, Gain = 15
- GAIN[1:0] = 0b11, Gain = 20

Supposing GAIN[1:0] bits are configured to 0b11, the PGA gain is therefore set to 20. Then the following equation exists:

$V_{op} - V_{on} = 20 \times (V_{inp} - V_{inm})$

24.8.2 PGA input

The PGA inputs V_{inp} and V_{inm} come from two sides of the external current sensing resistor. The current equals to the voltage drop on this resistor divided by the resistor value, as follows:

$I = (V_{inp} - V_{inm}) / R$

The PGA common voltage input ranges from 4.5 V to 5.5 V, and the user should keep V_{inp} or V_{inm} within this range to guarantee PGA proper operations.

24.8.3 PGA output

PGA has a differential output pair V_{op} (voltage output positive) and V_{on} (voltage output negative), with the following equation established.

$V_{op} - V_{on} = Gain \times (V_{inp} - V_{inm})$

V_{on} is always biased to $^1/_2$ V_{DDA} (power supply of PGA). This enables PGA to amplify either positive or negative voltage input.

- With positive voltage input ($V_{inp} - V_{inm} > 0$), the V_{op} ranges from $^1/_2$ V_{DDA} to V_{DDA}.
- With negative voltage input ($V_{inp} - V_{inm} < 0$), the V_{op} ranges from 0 to $^1/_2$ V_{DDA}.

User should select the proper gain, depending on the input voltage range, to fit the effective ADC input voltage range. The recommended gain versus input voltage range is as follows:

- For Gain = 8, it can process the maximum input voltage as 250 mV and the minimum as −250 mV.
- For Gain = 10, the maximum input is 200 mV and the minimum is −200 mV.
- For Gain = 15, the maximum input is 130 mV and the minimum is −130 mV.
- For Gain = 20, the maximum input is 100 mV and the minimum is −100 mV.

24.8.4 Input offset voltage calibration

To achieve high-accuracy current sensing, the PGA input offset voltage under different gains should be calibrated.

With input offset voltage, the actual output of PGA is shown in the following equation:

$V_{op} - V_{on} = (V_{inp} - V_{inm} + V_{offset}) \times Gain$

When $V_{inp} = V_{inm}$, the PGA input offset voltage multiplied by gain equals to the difference of V_{op} and V_{on}, as follows: $V_{op} - V_{on} = V_{offset} \times Gain$.

During the factory test phase, 5 V is applied on two PGA inputs (V_{inp} and V_{inm}). ADC reads PGA output pair V_{op} and V_{on}. Their ADC conversion result difference is stored in the SIM_IFR*x* register, which enables the user to calibrate the PGA input offset voltage.

Chapter 25
FlexTimer Module (FTM)

25.1 Chip-specific FTM information

25.1.1 FTM overview

The FTM timer contains up to six channels which support input capture, output compare and the generation of PWM signals to control electric motor and power management applications. FTM time reference is a 16-bit counter which can be used as an unsigned or signed counter.

This device contains two 2-channel FTM with basic TPM functions. The table below summarizes the configuration of FTM modules.

Table 25-1. FTM modules features

Feature	FTM0/FTM1
Number of channels	2
Initial counting value	no
Periodic TOF	no
Input capture mode	yes
Channel input filter	no
Output compare mode	yes
Edge-Aligned PWM (EPWM)	yes
Center-Aligned PWM (CPWM)	yes
Combine mode	no
Complementary mode	no
PWM synchronization	no
Inverting	no
Software output control (SWOC)	no
Deadtime insertion	no
Output mask	no
Fault control	no

Table continues on the next page...

Table 25-1. FTM modules features (continued)

Feature	FTM0/FTM1
Number of fault inputs	0
Fault input filter	no
Polarity control	no
Initialization	no
Channel match trigger	no
Initialization trigger	yes
Capture test mode	no
DMA	no
Dual edge capture mode	no
Quadrature decoder mode	no
Quadrature decoder input filter	no
Debug modes	no
Intermediary load	no
Global time base enable	no

25.1.2 FTM clock options

The selectable FTM source clock can be the system clock or the fixed frequency clock. The selected control source is controlled by FTMx_SC[CLKS].

- When FTMx_SC[CLKS] = 00, no clock is selected (this in effect, disables the FTM counter).
- When FTMx_SC[CLKS] = 01, the timer clock is selected.
- When FTMx_SC[CLKS] = 10, the fixed frequency clock(ICSFFCLK) is selected.
- When FTMx_SC[CLKS] = 11, reserved.

25.1.3 FTM interconnections

FTM0 and FTM1 counter initialization trigger can be outputted to other on-chip module through trigger crossbar. See TBAR for details.

FTM0 and FTM1 channels could be also connected with other on-chip module through signal crossbar. See SBAR for details.

25.1.4 FTM interrupts

The FlexTimer has multiple sources of interrupt. However, either source can generate a single interrupt request to the interrupt controller. When an FTM interrupt occurs, read the FTM status registers to determine the exact interrupt source.

25.2 Introduction

The FlexTimer module (FTM) is a two-to-eight channel timer that supports input capture and output compare. The FTM time reference is a 16-bit counter that can be used as an unsigned or signed counter.

25.2.1 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 provides low cost and backwards compatibility with the TPM module.

Several key enhancements are made:

- Signed up counter
- Enhanced triggering functionality
- Initialization 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.

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.

25.2.2 Features

The FTM features include:

- FTM source clock is selectable
    - Source clock can be the system clock, the fixed frequency clock

  - Fixed frequency clock is an additional clock input to allow the selection of an on chip clock source other than the system 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
- 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
- The generation of an interrupt per channel
- The generation of an interrupt when the counter overflows
- Backwards compatible with TPM
- Testing of input captures for a stuck at zero and one conditions

25.2.3 Modes of operation

When the MCU is in an active Debug mode, the FTM temporarily suspends all counting until the MCU 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 MCU from Wait mode, the power can then be saved by disabling FTM functions before entering Wait mode.

25.2.4 Block diagram

The FTM uses one input/output (I/O) pin per channel, CHn (FTM channel (n)) where n is the channel number (0–1).

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.

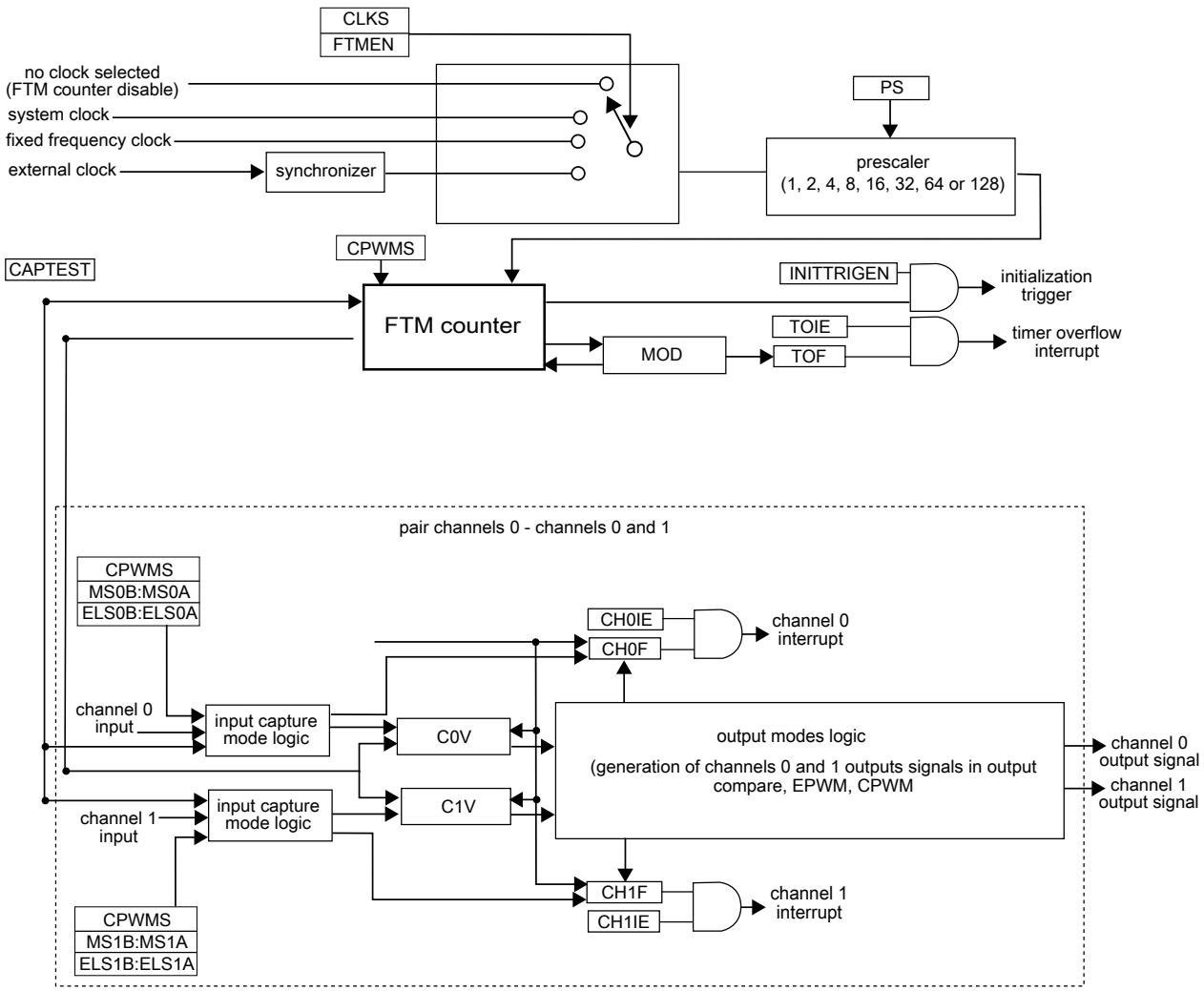


Figure 25-1. FTM block diagram

25.3 FTM signal descriptions

Table 25-2 shows the user-accessible signals for the FTM.

Table 25-2. 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 1-0	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.

25.4 Memory map and register definition

25.4.1 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.

25.4.2 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	25.4.3/337
4003_8004	Counter (FTM0_CNT)	32	R/W	0000_0000h	25.4.4/339
4003_8008	Modulo (FTM0_MOD)	32	R/W	0000_0000h	25.4.5/339
4003_800C	Channel (n) Status And Control (FTM0_C0SC)	32	R/W	0000_0000h	25.4.6/340
4003_8010	Channel (n) Value (FTM0_C0V)	32	R/W	0000_0000h	25.4.7/342
4003_8014	Channel (n) Status And Control (FTM0_C1SC)	32	R/W	0000_0000h	25.4.6/340
4003_8018	Channel (n) Value (FTM0_C1V)	32	R/W	0000_0000h	25.4.7/342
4003_806C	FTM External Trigger (FTM0_EXTTRIG)	32	R/W	0000_0000h	25.4.8/342
4003_9000	Status And Control (FTM1_SC)	32	R/W	0000_0000h	25.4.3/337

Table continues on the next page...

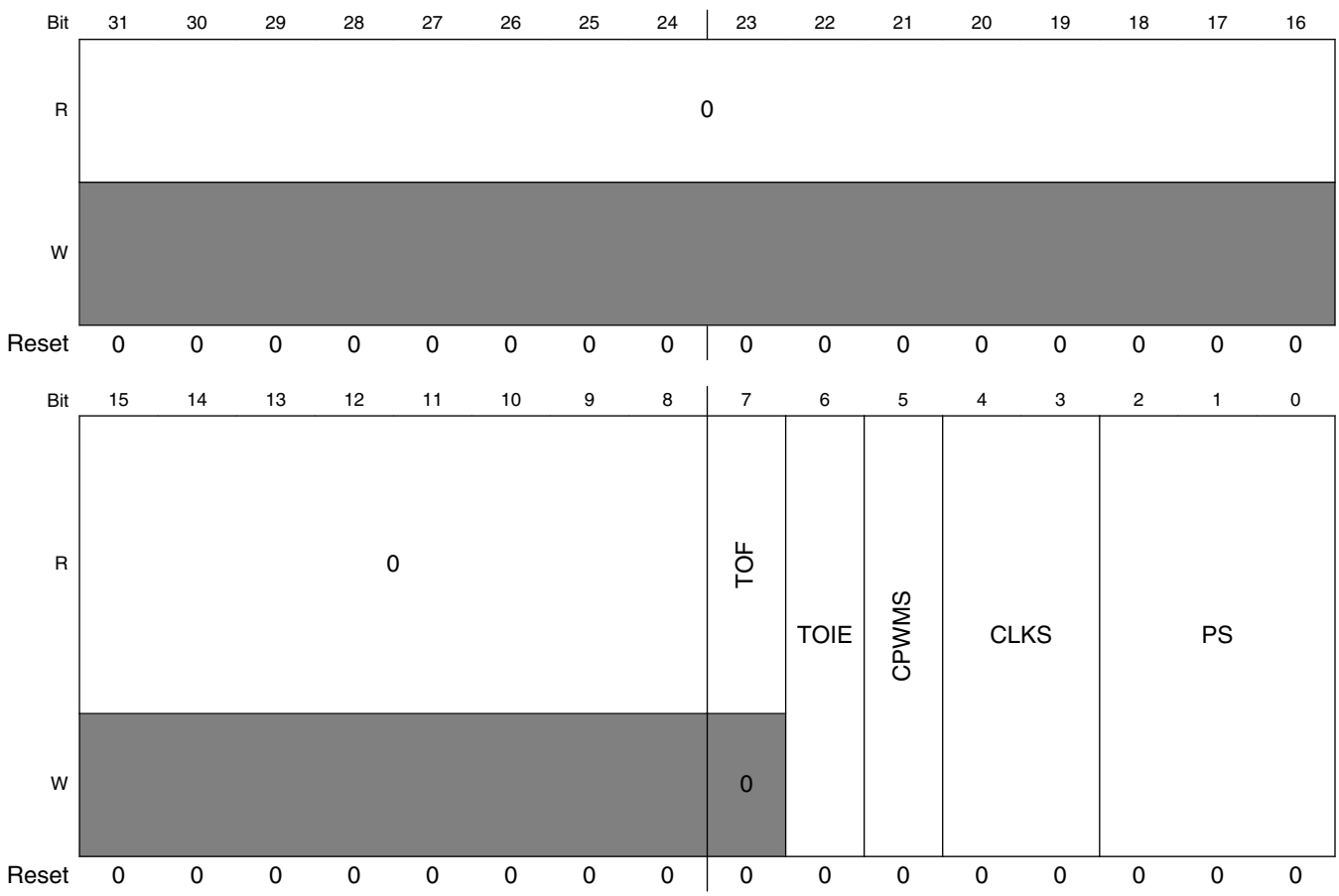
FTM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4003_9004	Counter (FTM1_CNT)	32	R/W	0000_0000h	25.4.4/339
4003_9008	Modulo (FTM1_MOD)	32	R/W	0000_0000h	25.4.5/339
4003_900C	Channel (n) Status And Control (FTM1_C0SC)	32	R/W	0000_0000h	25.4.6/340
4003_9010	Channel (n) Value (FTM1_C0V)	32	R/W	0000_0000h	25.4.7/342
4003_9014	Channel (n) Status And Control (FTM1_C1SC)	32	R/W	0000_0000h	25.4.6/340
4003_9018	Channel (n) Value (FTM1_C1V)	32	R/W	0000_0000h	25.4.7/342
4003_906C	FTM External Trigger (FTM1_EXTTRIG)	32	R/W	0000_0000h	25.4.8/342

25.4.3 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



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.
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 one of the three 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 10 Fixed frequency 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

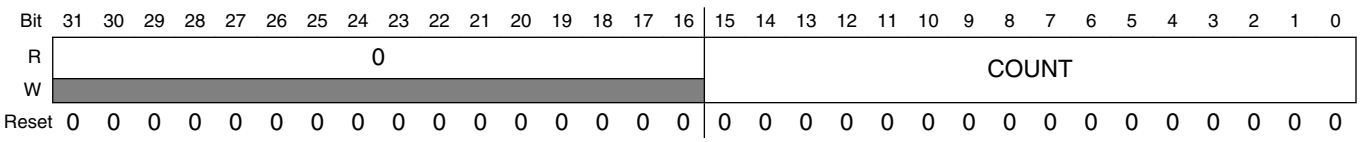
25.4.4 Counter (FTMx_CNT)

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 Debug is active, the FTM counter is frozen. This is the value that you may read.

Address: Base address + 4h offset



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

25.4.5 Modulo (FTMx_MOD)

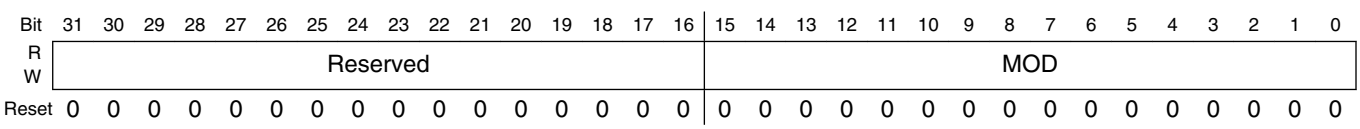
The Modulo register contains the modulo value for the FTM counter. After the FTM counter reaches the modulo value, the overflow flag (TOF) becomes set at the next clock, and the next value of FTM counter depends on the selected counting method; see Counter.

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 Debug 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



FTMx_MOD field descriptions

Field	Description
31–16 Reserved	This field is reserved.
MOD	Modulo Value

25.4.6 Channel (n) Status And Control (FTM*x*_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 25-24. Mode, edge, and level selection

CPWMS	MSnB:MSnA	ELSnB:ELSnA	Mode	Configuration
X	XX	00	Pin not used for FTM—revert the channel pin to general purpose I/O or other peripheral control	
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)

Address: Base address + Ch 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
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	0
W									0							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

FTMx_C*n*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 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.
5 MSB	Channel Mode Select Used for further selections in the channel logic. Its functionality is dependent on the channel mode. See Table 25-8. 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 25-8. 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 25-8. 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 25-8. 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 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

25.4.7 Channel (n) Value (FTMx_CnV)

These registers contain the captured FTM counter value for the input modes or the match value for the output modes.

In Input Captureand Capture Test 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 Debug mode is active or not.

Address: Base address + 10h 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	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_CnV 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

25.4.8 FTM External Trigger (FTMx_EXTTRIG)

This register:

- Enables the generation of a trigger when the FTM counter is equal to its initial value

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									INITTRIGEN	Reserved					
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

FTMx_EXTTRIG field descriptions

Field	Description
31–7 Reserved	This field is reserved.
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.
Reserved	This field is reserved.

25.5 Functional description

The notation used in this document to represent the counters and the generation of the signals is shown in the following figure.

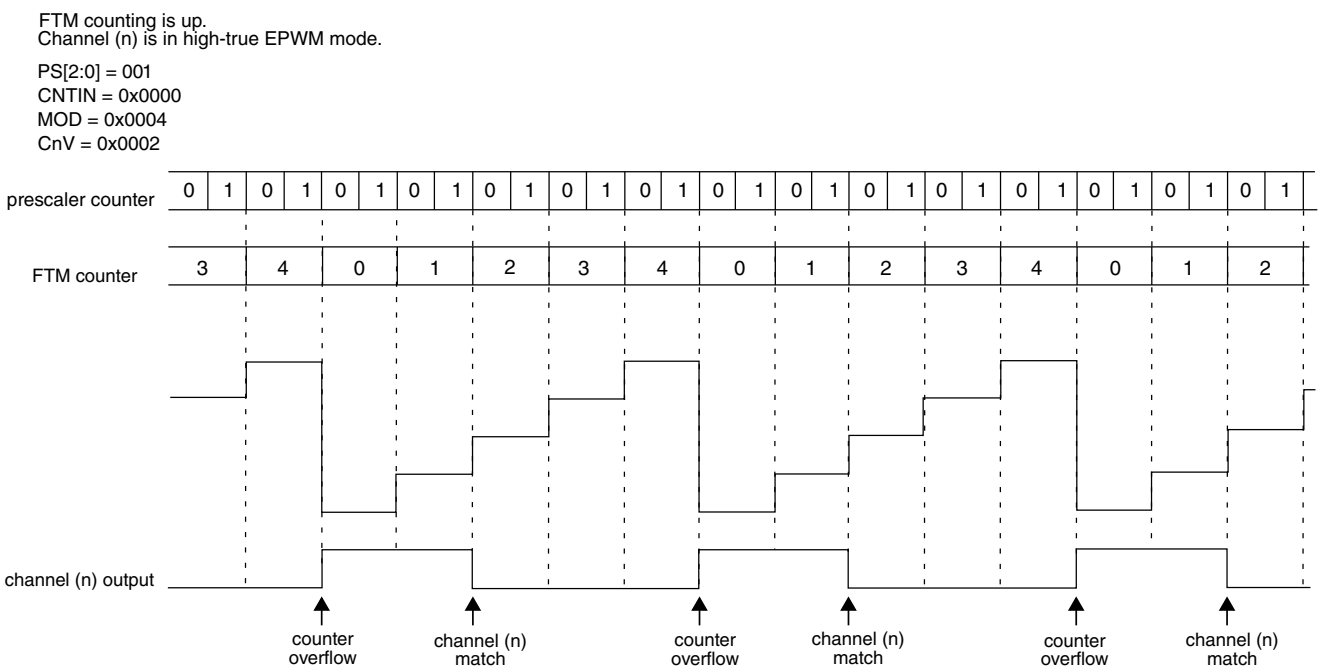


Figure 25-32. Notation used

25.5.1 Clock source

The FTM has only one clock domain: the system clock.

25.5.1.1 Counter clock source

The CLKS[1:0] bits in the SC register select one of three possible clock sources for the FTM counter or disable the FTM counter. After any MCU 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 fixed frequency clock is an alternative clock source for the FTM counter that allows the selection of a clock other than the system clock or an external clock. This clock input is defined by chip integration. Refer to the chip specific documentation for further information. Due to FTM hardware implementation limitations, the frequency of the fixed frequency clock must not exceed 1/2 of the system clock frequency.

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.

25.5.2 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.

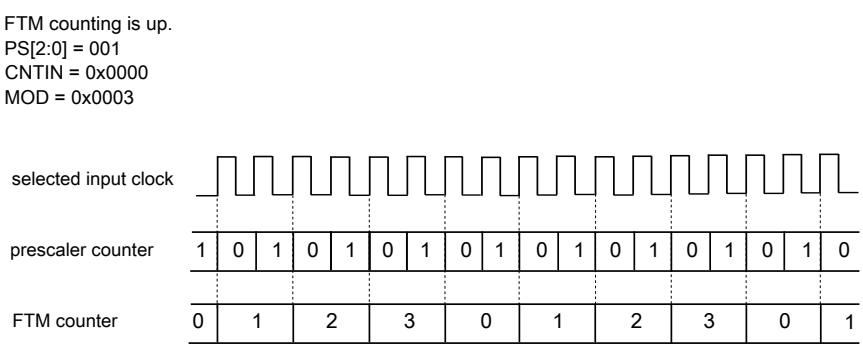


Figure 25-33. Example of the prescaler counter

25.5.3 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

25.5.3.1 Up counting

Up counting is selected when:

- 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

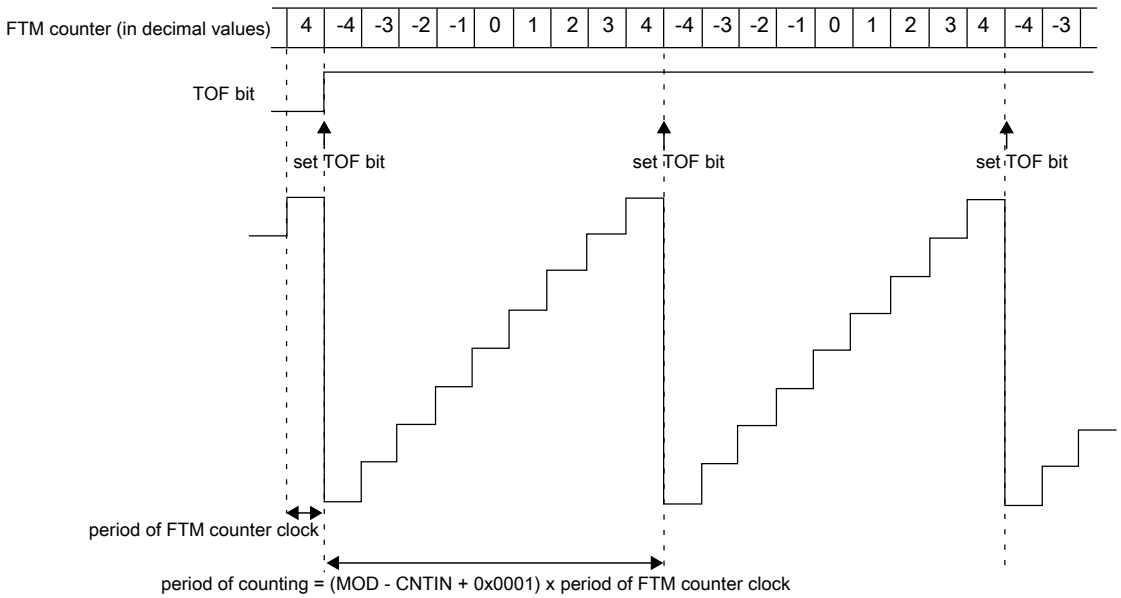


Figure 25-34. Example of FTM up and signed counting

Table 25-45. 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.

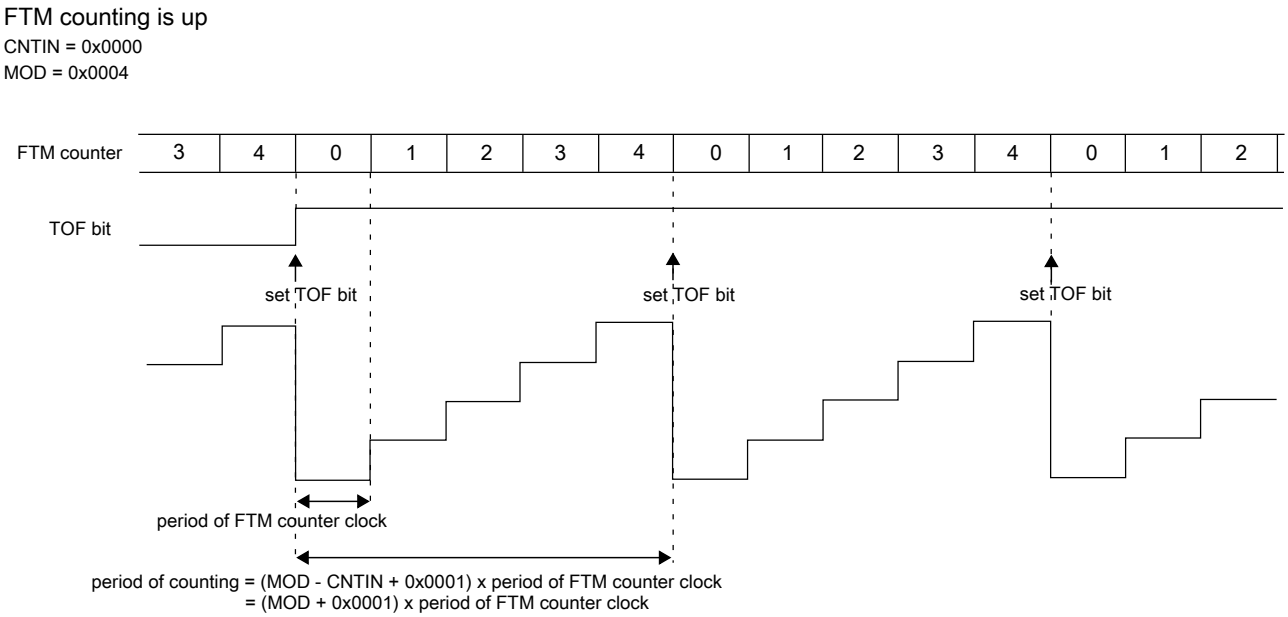


Figure 25-35. Example of FTM up counting with CNTIN = 0x0000

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.

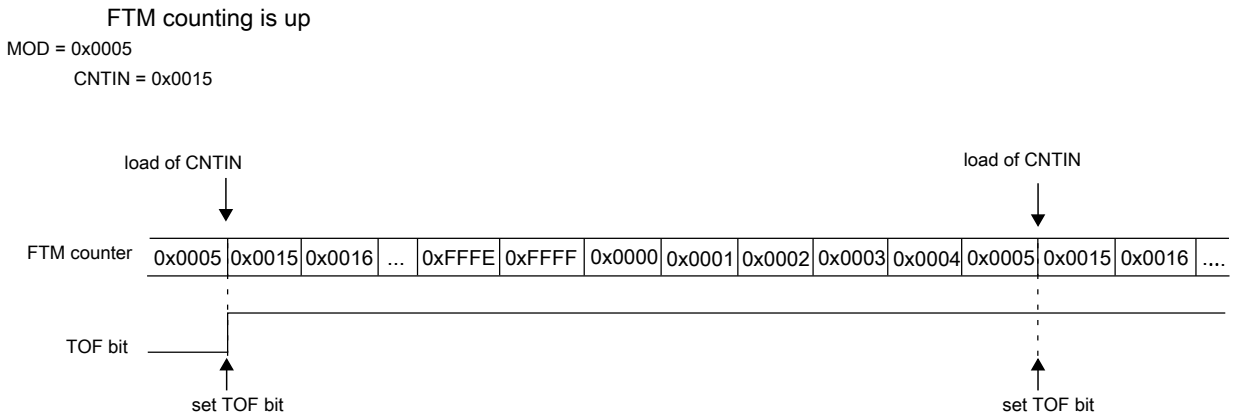


Figure 25-36. Example of up counting when the value of CNTIN is greater than the value of MOD

25.5.3.2 Up-down counting

Up-down counting is selected when:

- 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.

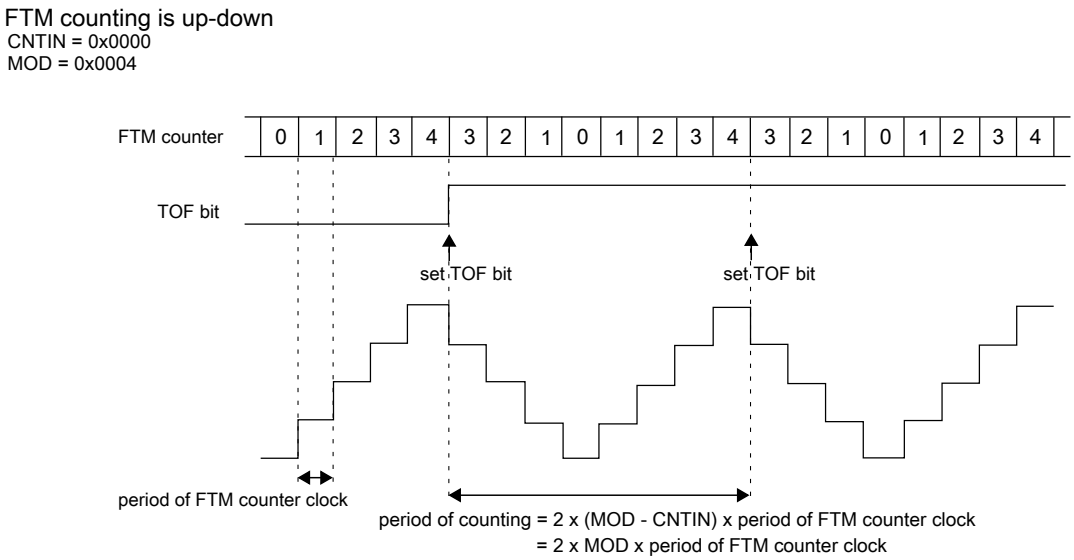


Figure 25-37. Example of up-down counting when CNTIN = 0x0000

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.

25.5.3.3 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.

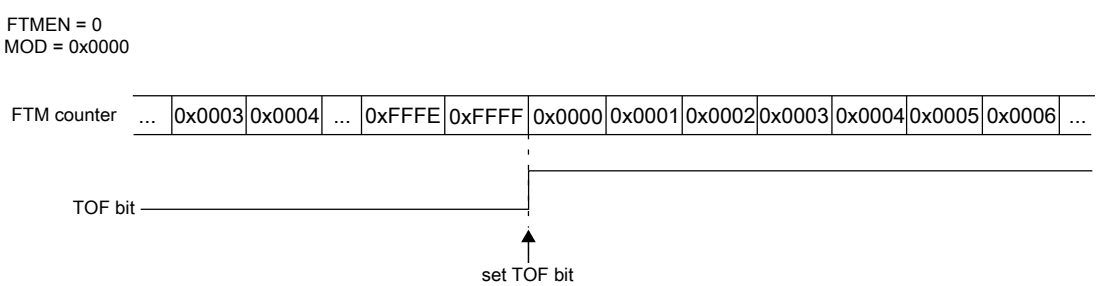


Figure 25-38. Example when the FTM counter is free running

The FTM counter is also a free running counter when:

- FTMEN = 1
- CPWMS = 0
- CNTIN = 0x0000, and
- MOD = 0xFFFF

25.5.3.4 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.

25.5.3.5 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.

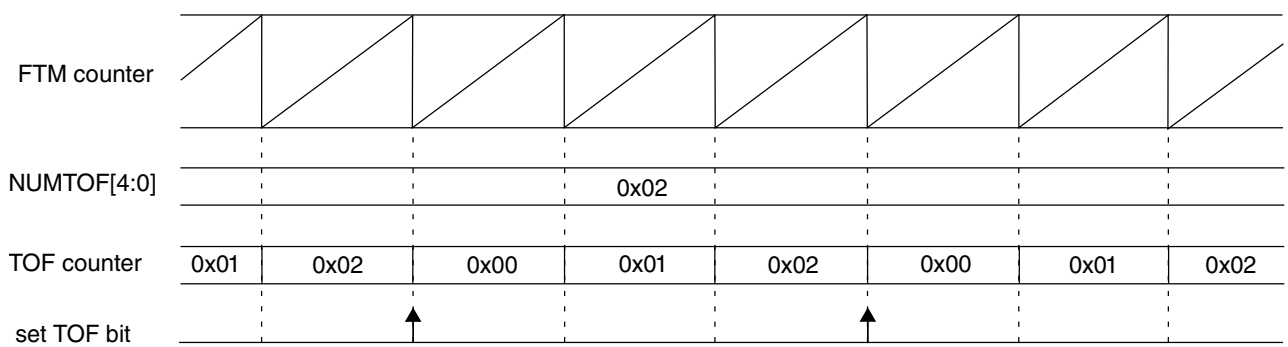


Figure 25-39. Periodic TOF when NUMTOF = 0x02

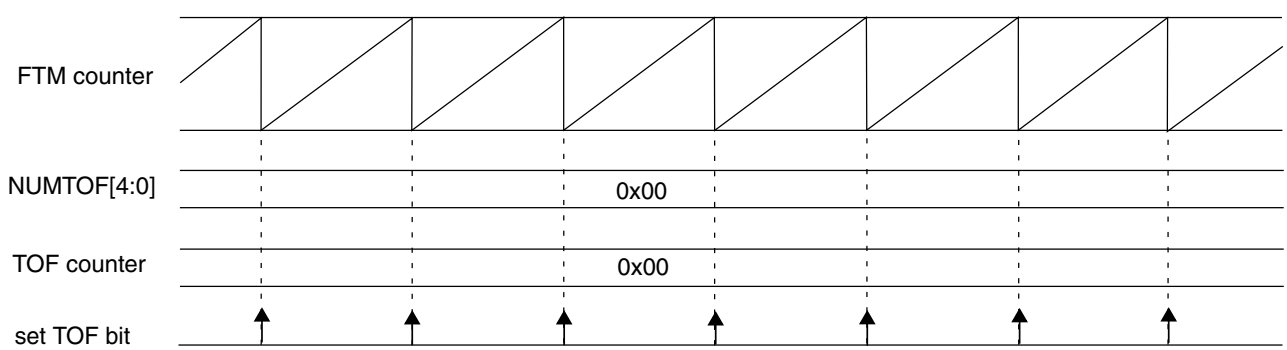


Figure 25-40. Periodic TOF when NUMTOF = 0x00

25.5.4 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 Debug mode, the input capture function works as configured. When a selected edge event occurs, the FTM counter value, which is frozen because of Debug, is captured into the CnV register and the CHnF bit is set.

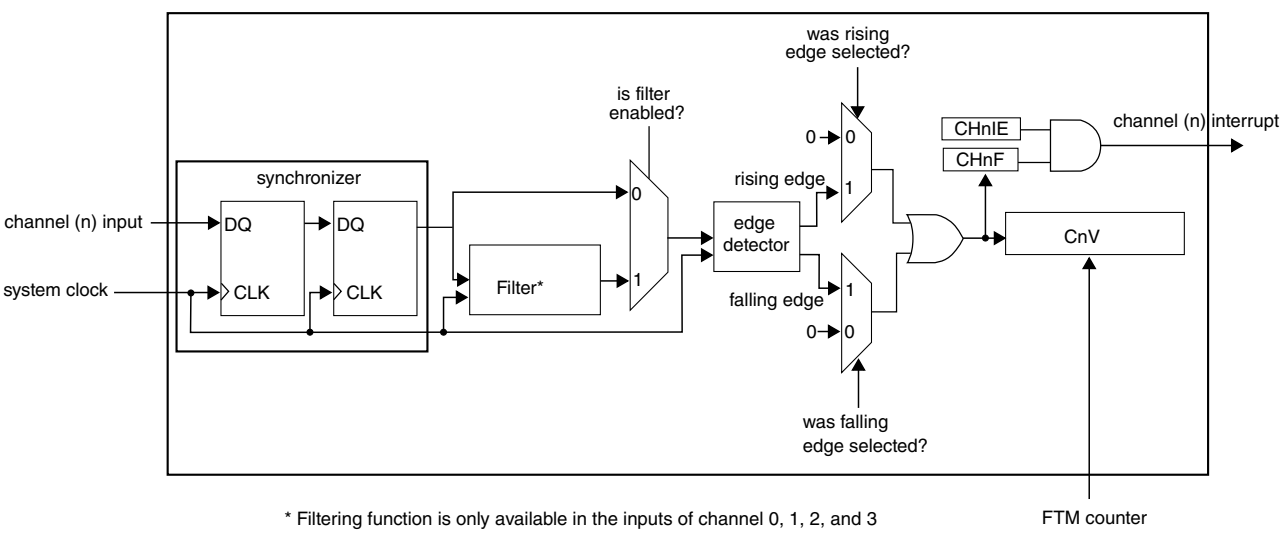


Figure 25-41. Input Capture mode

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.

25.5.4.1 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.

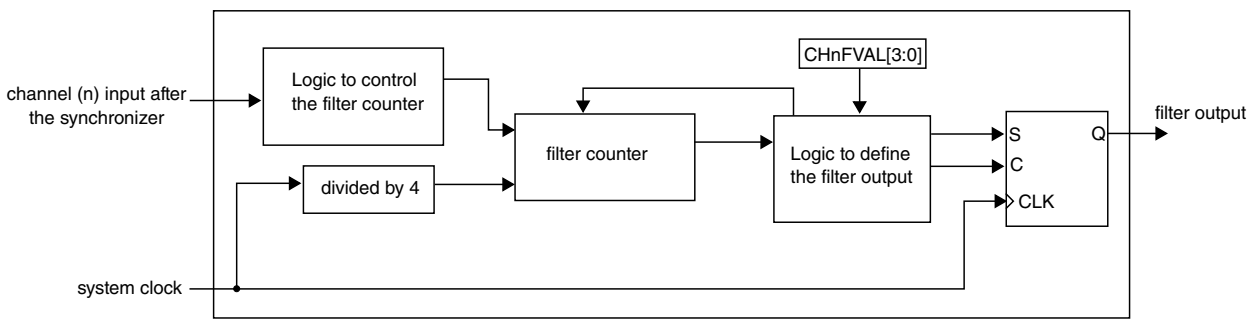


Figure 25-42. Channel input filter

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.

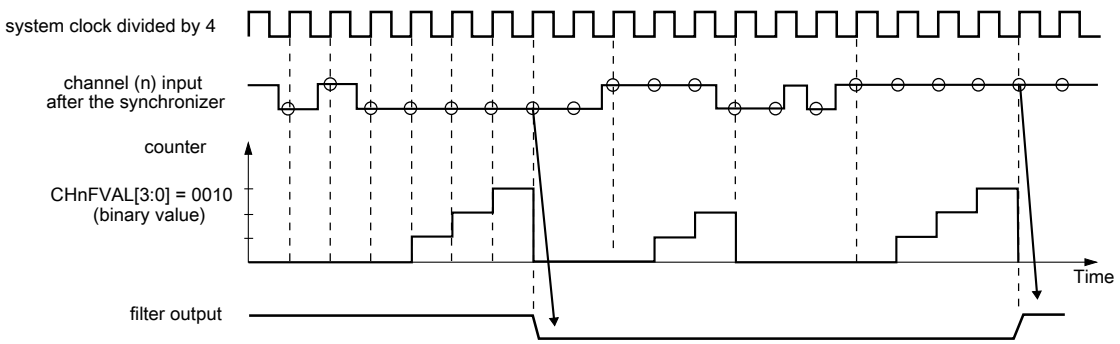


Figure 25-43. Channel input filter example

25.5.5 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).

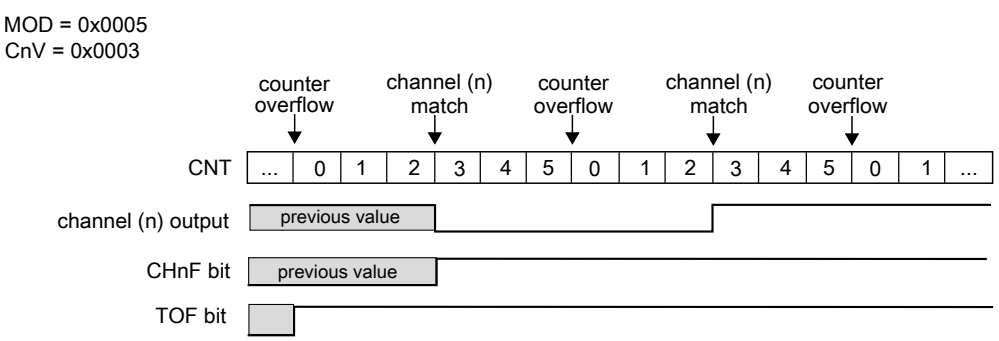


Figure 25-44. Example of the Output Compare mode when the match toggles the channel output

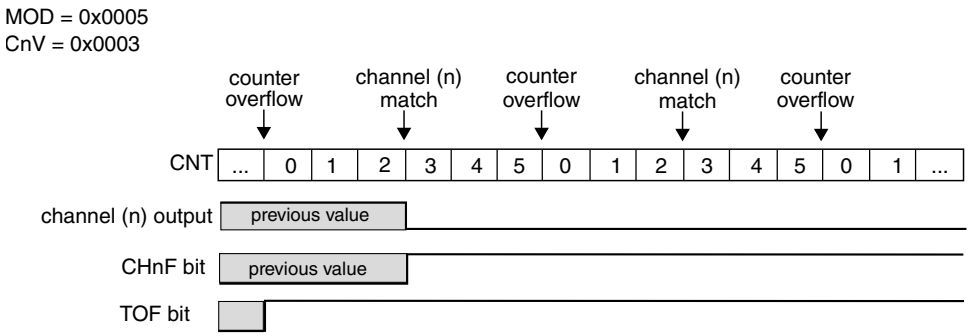


Figure 25-45. Example of the Output Compare mode when the match clears the channel output

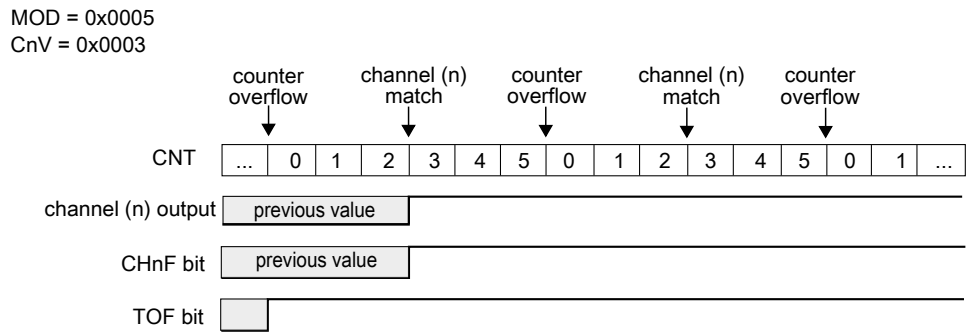


Figure 25-46. Example of the Output Compare mode when the match sets the channel output

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.

25.5.6 Edge-Aligned PWM (EPWM) mode

The Edge-Aligned mode is selected when:

- 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.

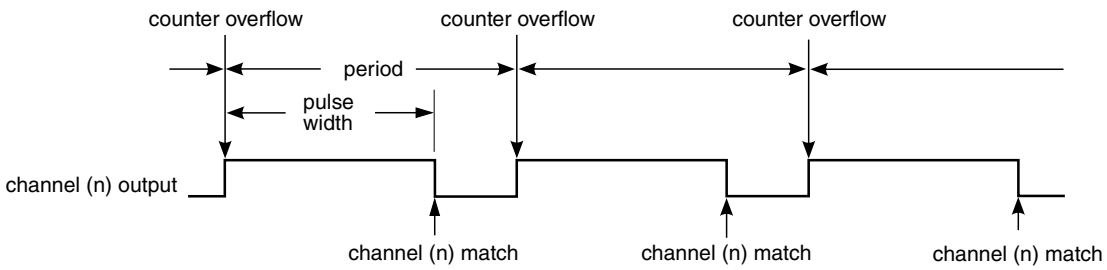


Figure 25-47. EPWM period and pulse width with ELSnB:ELSnA = 1:0

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.

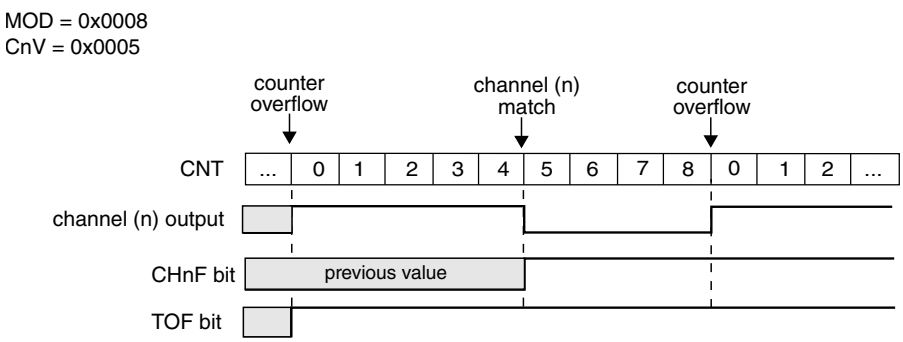


Figure 25-48. EPWM signal with ELSnB:ELSnA = 1:0

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.

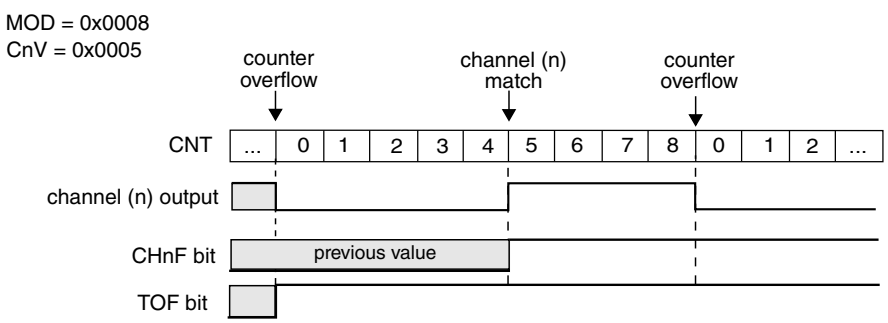


Figure 25-49. EPWM signal with ELSnB:ELSnA = X:1

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 even when there is the channel (n) match. 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.

25.5.7 Center-Aligned PWM (CPWM) mode

The Center-Aligned mode is selected when:

- 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).

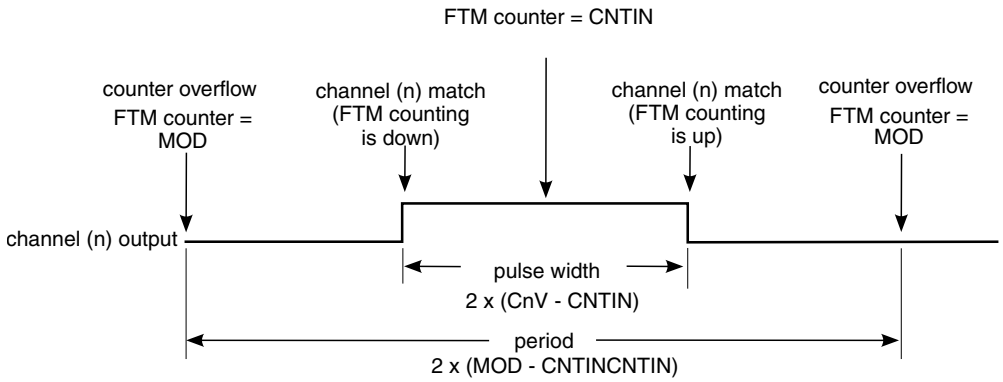


Figure 25-50. CPWM period and pulse width with ELSnB:ELSnA = 1:0

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.

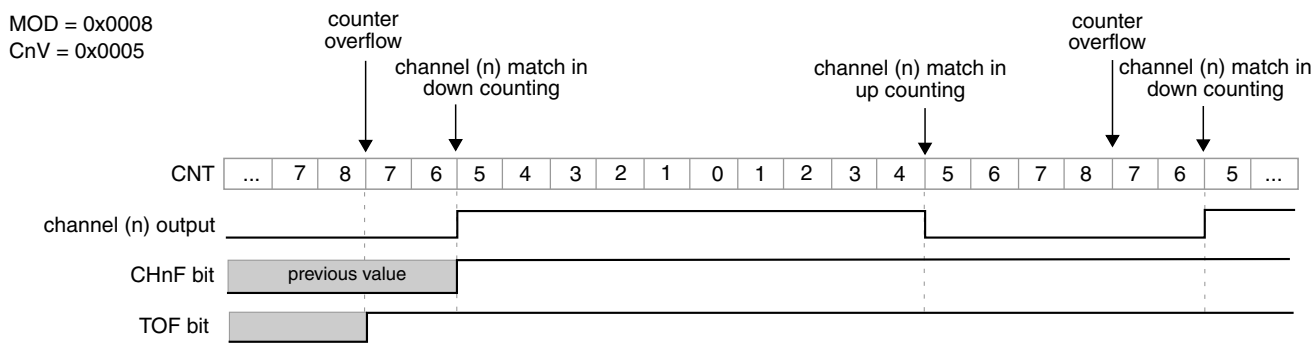


Figure 25-51. CPWM signal with ELSnB:ELSnA = 1:0

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.

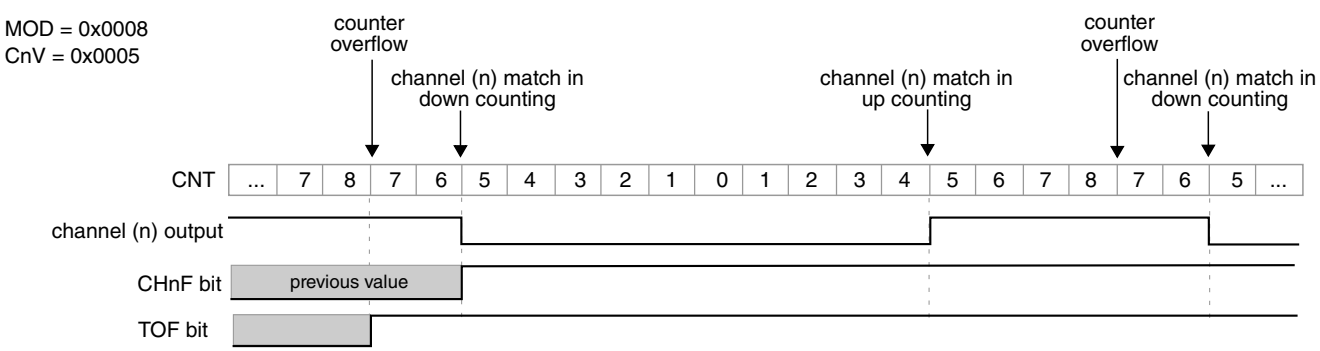


Figure 25-52. CPWM signal with ELSnB:ELSnA = X:1

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.

25.5.8 Registers updated from write buffers

25.5.8.1 CNTIN register update

The following table describes when CNTIN register is updated:

Table 25-46. 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.

25.5.8.2 MOD register update

The following table describes when MOD register is updated:

Table 25-47. 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:

Table continues on the next page...

Table 25-47. MOD register update (continued)

When	Then MOD register is updated
	• 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.

25.5.8.3 CnV register update

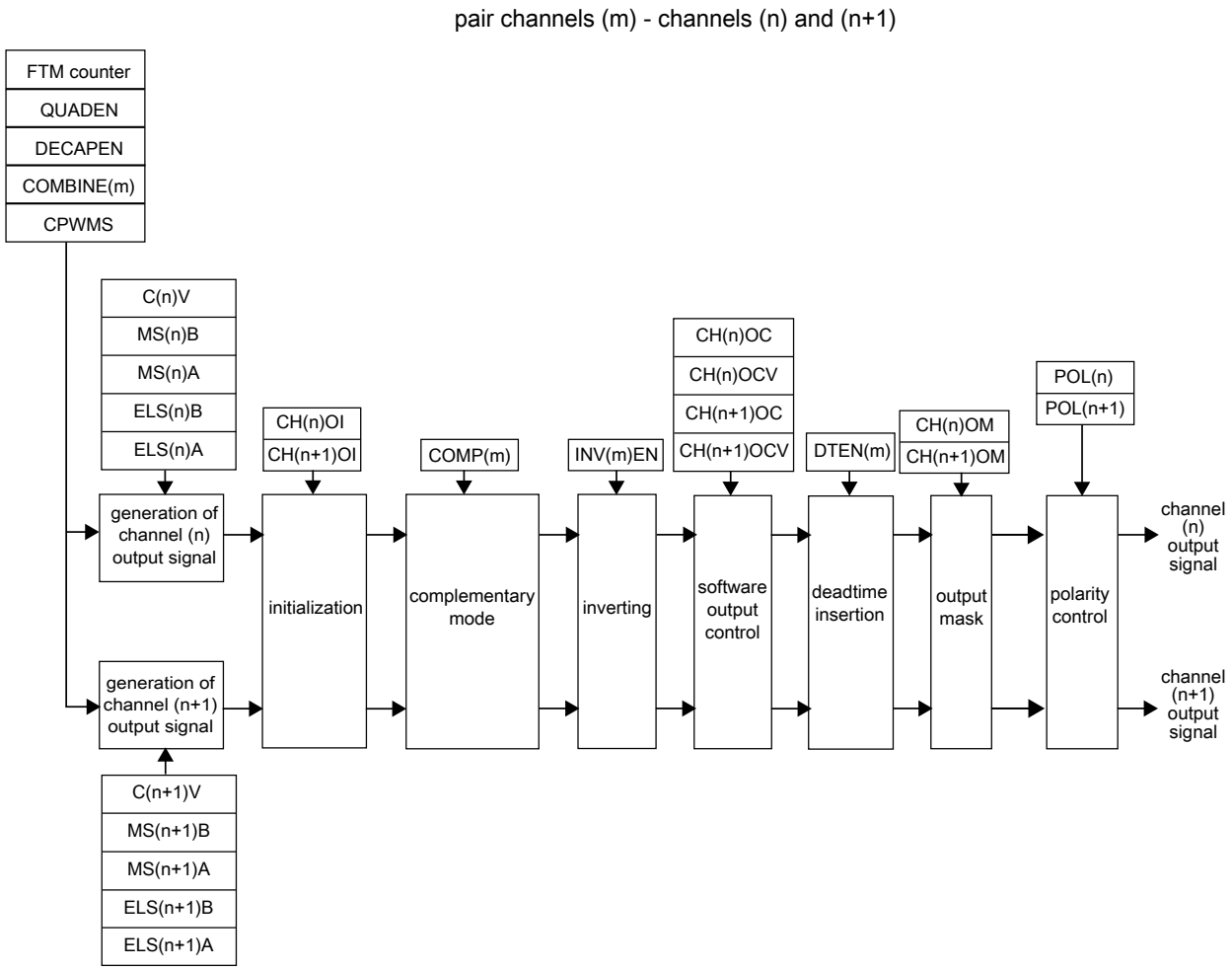
The following table describes when CnV register is updated:

Table 25-48. 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: • 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.

25.5.9 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 25-53. 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.

25.5.10 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.

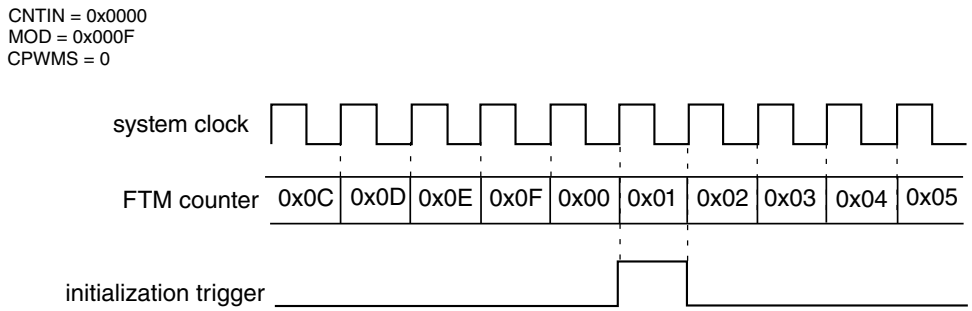


Figure 25-54. Initialization trigger is generated when the FTM counting achieves the CNTIN register value

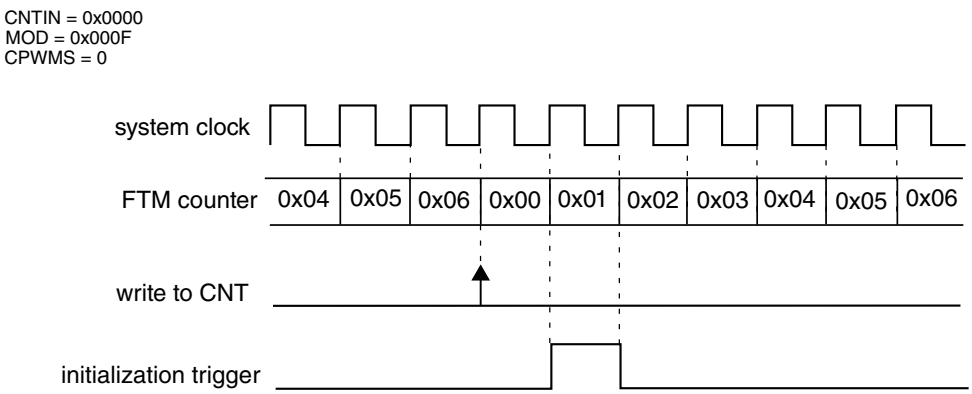


Figure 25-55. Initialization trigger is generated when there is a write to CNT register

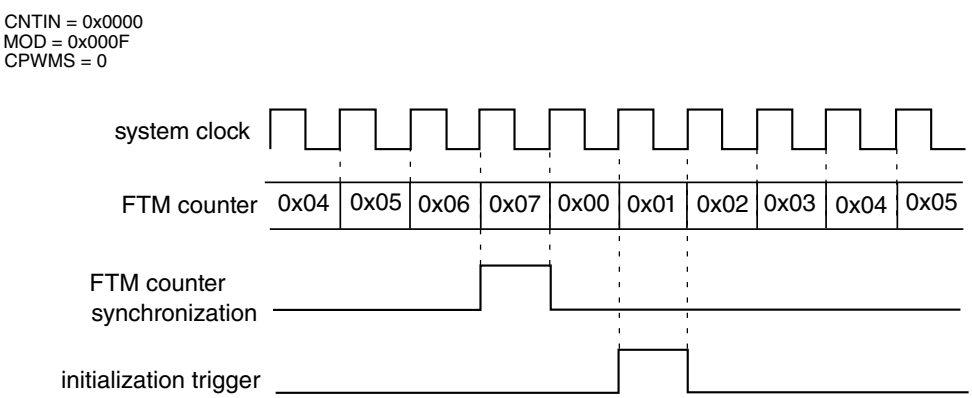


Figure 25-56. Initialization trigger is generated when there is the FTM counter synchronization

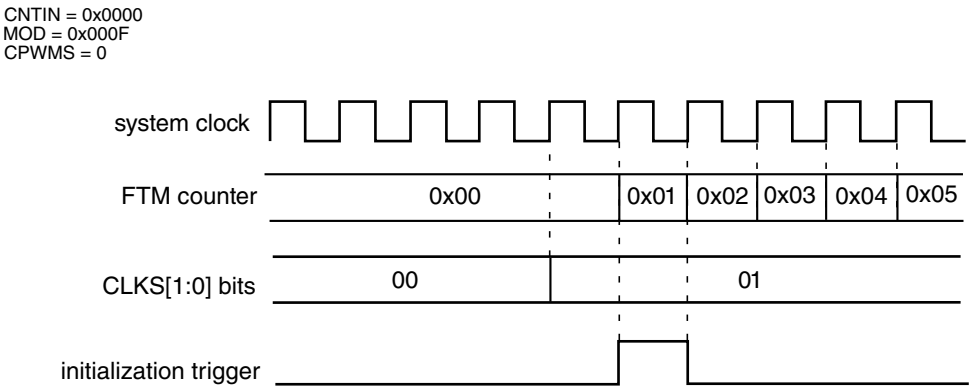


Figure 25-57. Initialization trigger is generated if (CNT = CNTIN), (CLKS[1:0] = 0:0), and a value different from zero is written to CLKS[1:0] bits

The initialization trigger output provides a trigger signal that is used for on-chip modules.

25.5.11 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.

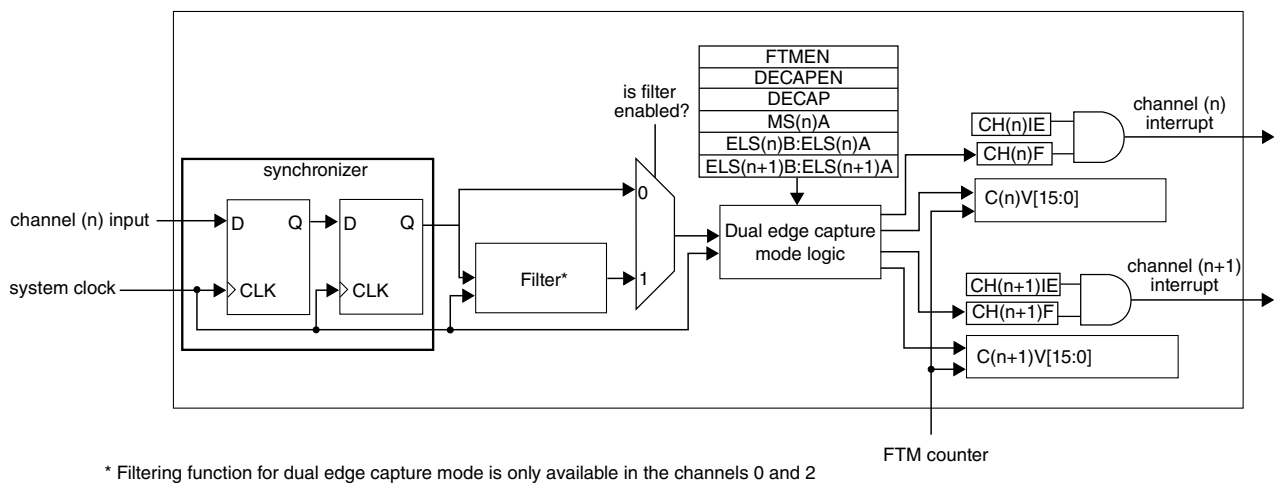


Figure 25-58. Dual Edge Capture mode block diagram

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.

25.5.11.1 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.

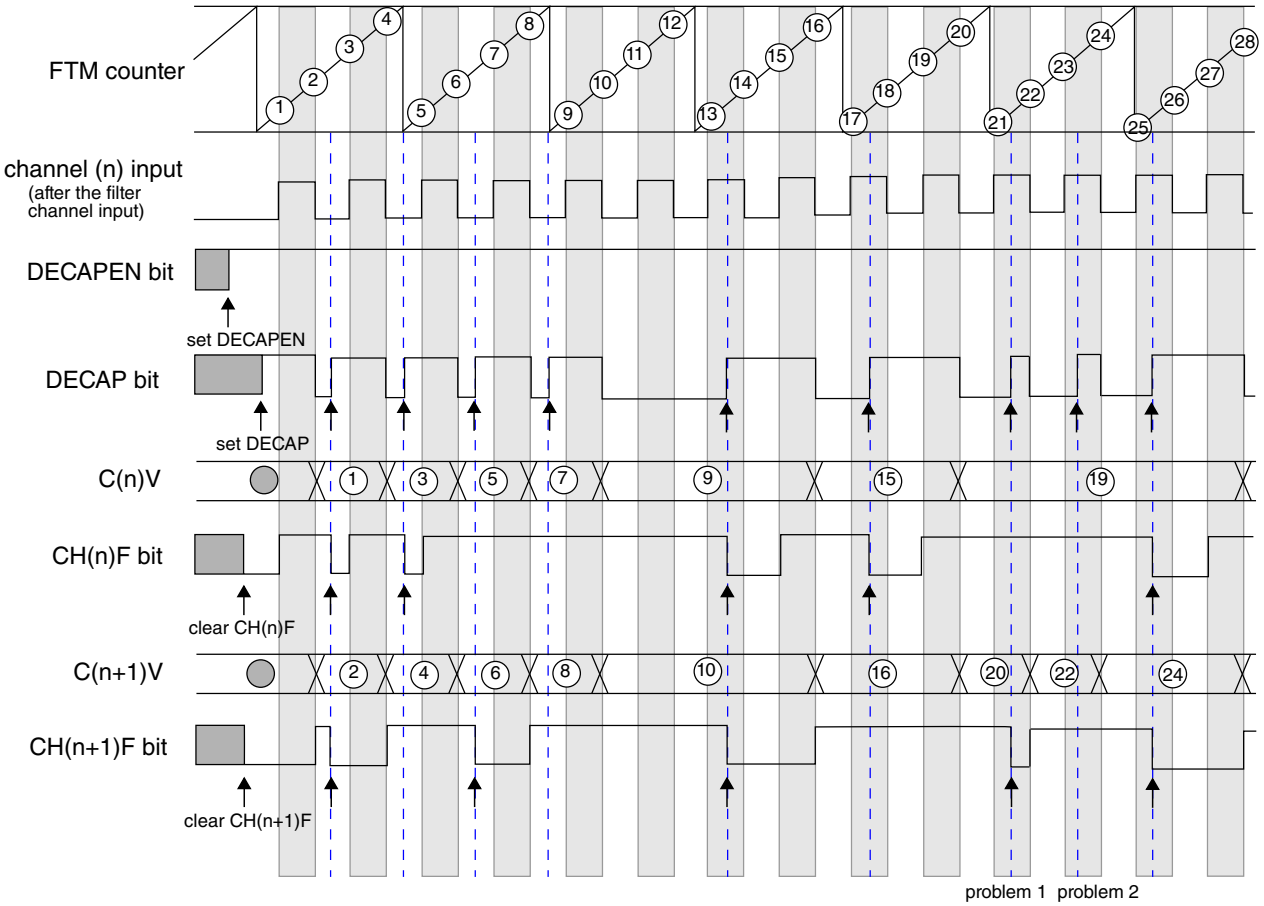
For a new sequence of the measurements in the Dual Edge Capture – Continuous mode, clear the CH(n)F and CH(n+1)F bits to start new measurements.

25.5.11.2 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.

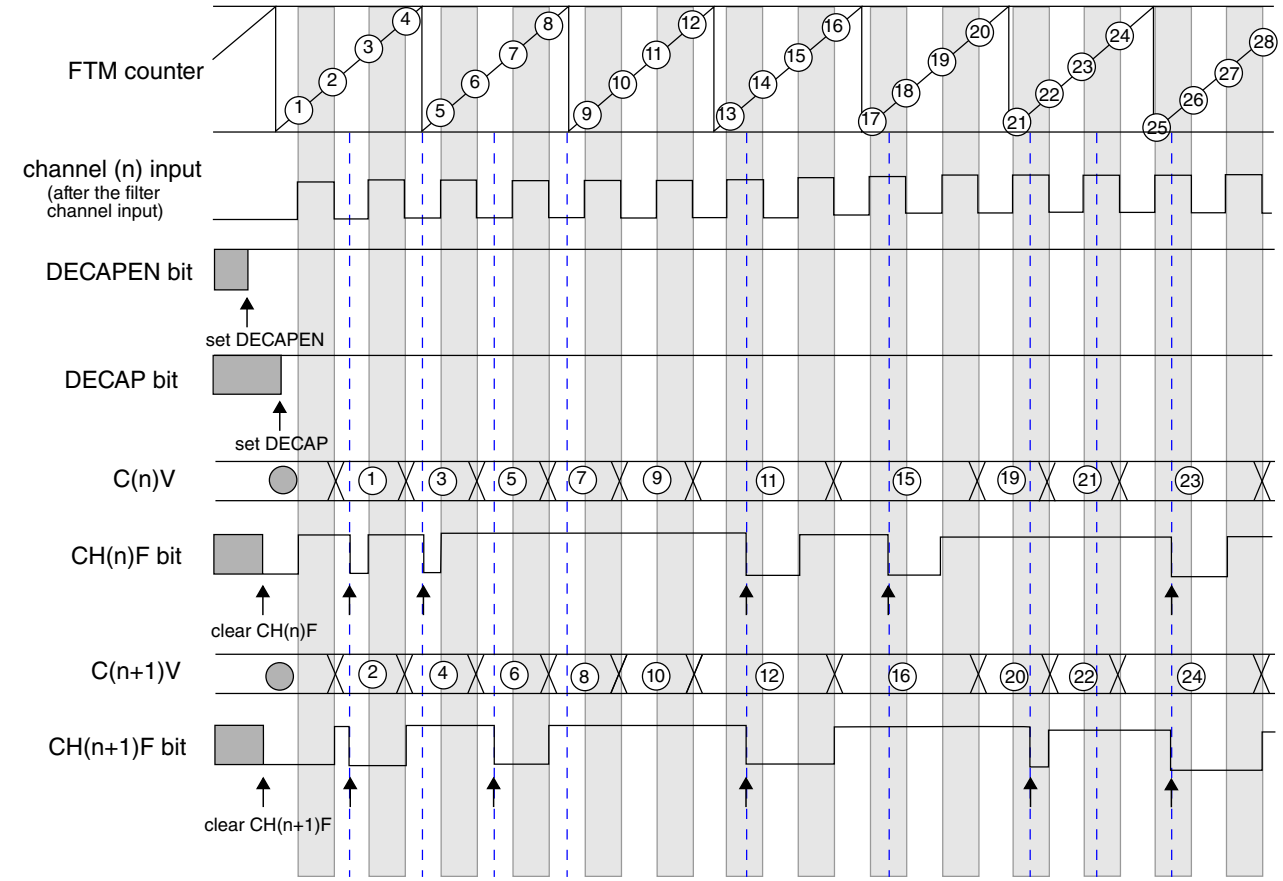
The following figure shows an example of the Dual Edge Capture – One-Shot mode used to measure the positive polarity pulse width. The DECAPEN bit selects the Dual Edge Capture mode, so it remains set. The DECAP bit is set to enable the measurement of next positive polarity pulse width. The CH(n)F bit 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 25-59. 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.

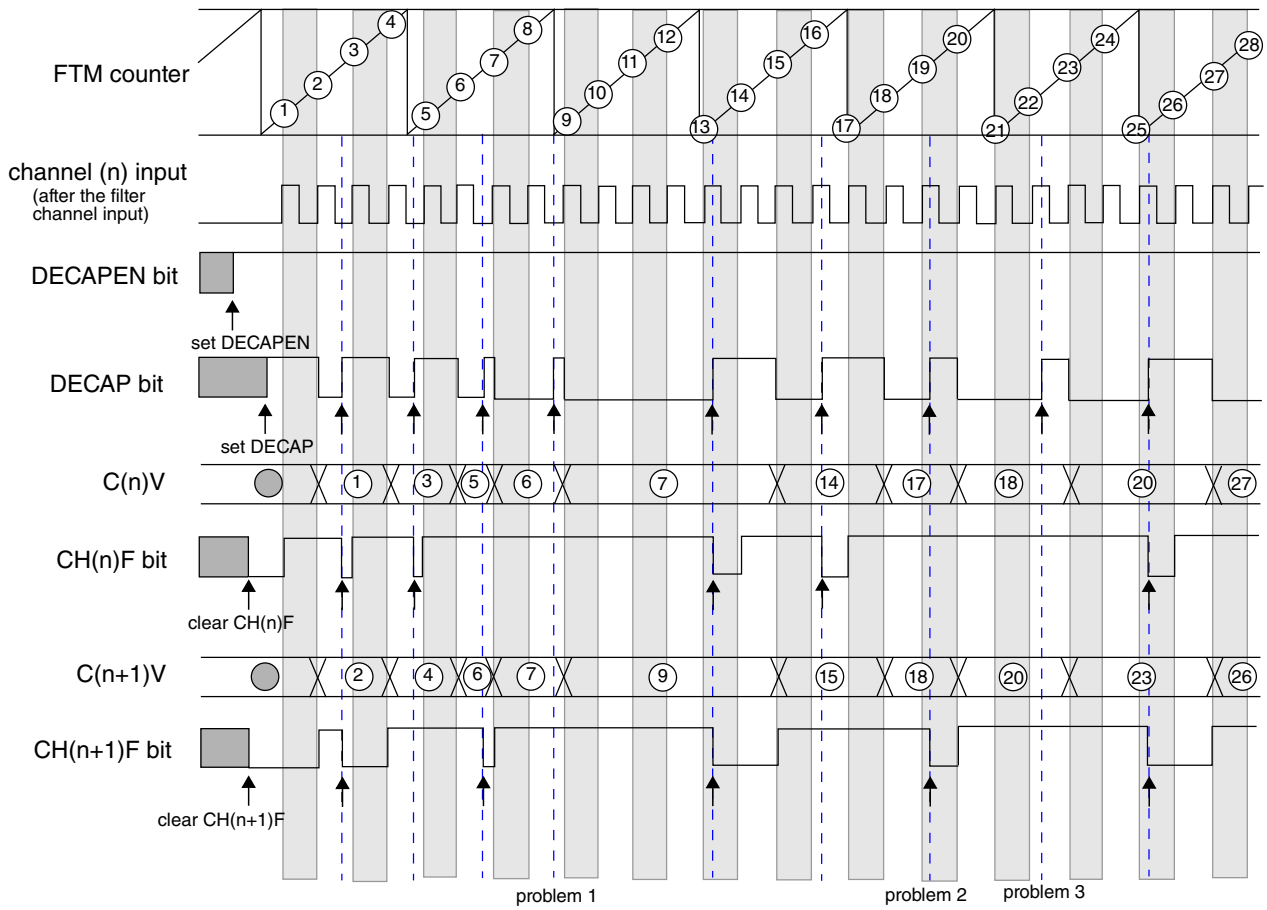
Figure 25-60. Dual Edge Capture – Continuous mode for positive polarity pulse width measurement

25.5.11.3 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.

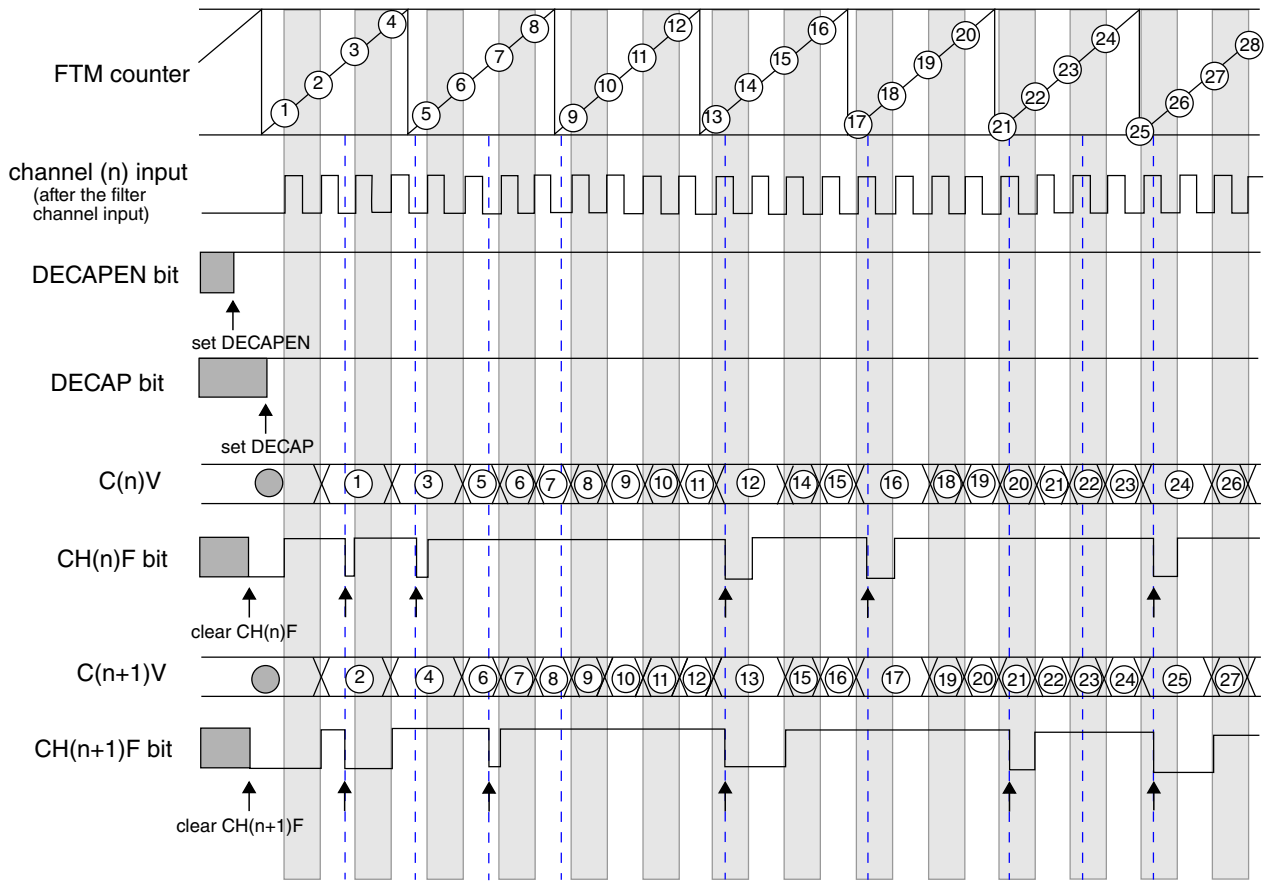


Note
- The commands set DECAPEN, set DECAP, clear CH(n)F, and clear CH(n+1)F are made by the user.
- Problem 1: channel (n) input = 0, set DECAP, not clear CH(n)F, and not clear CH(n+1)F.
- Problem 2: channel (n) input = 1, set DECAP, not clear CH(n)F, and clear CH(n+1)F.
- Problem 3: channel (n) input = 1, set DECAP, not clear CH(n)F, and not clear CH(n+1)F.

Figure 25-61. 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 25-62. Dual Edge Capture – Continuous mode to measure of the period between two consecutive rising edges

25.5.11.4 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.

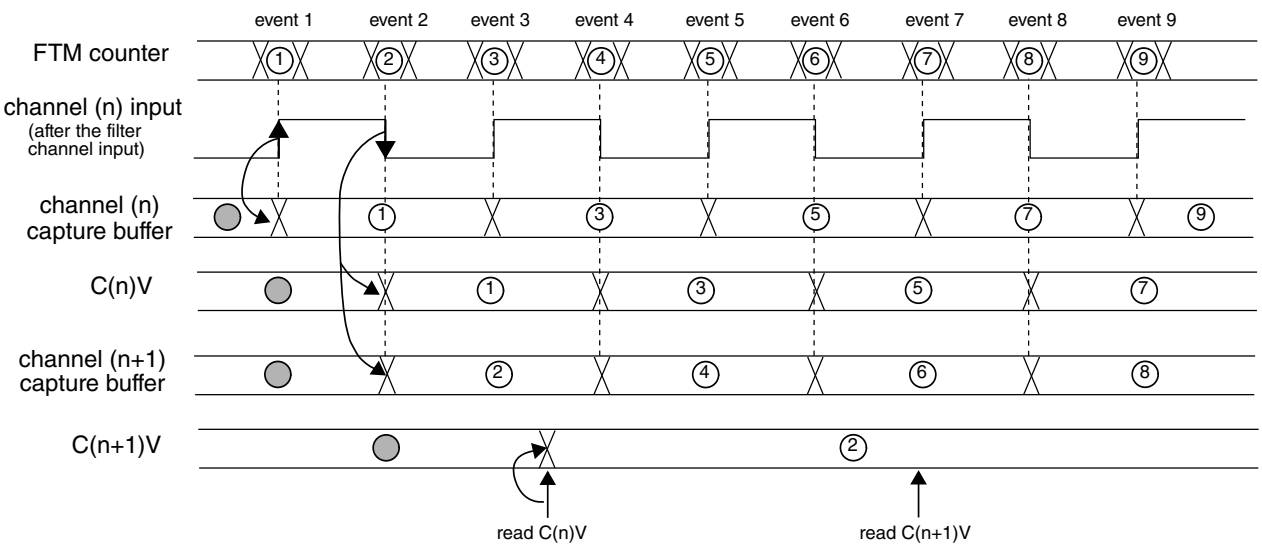


Figure 25-63. Dual Edge Capture mode read coherency mechanism

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.

25.6 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 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).

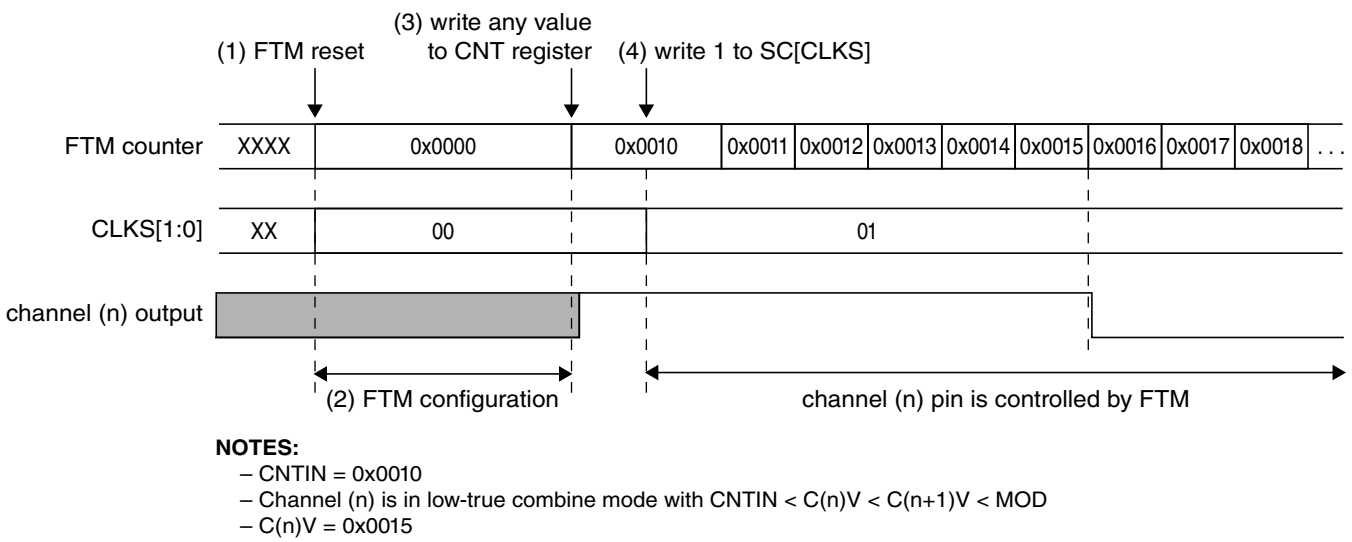


Figure 25-64. FTM behavior after reset when the channel (n) is in Combine mode

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).

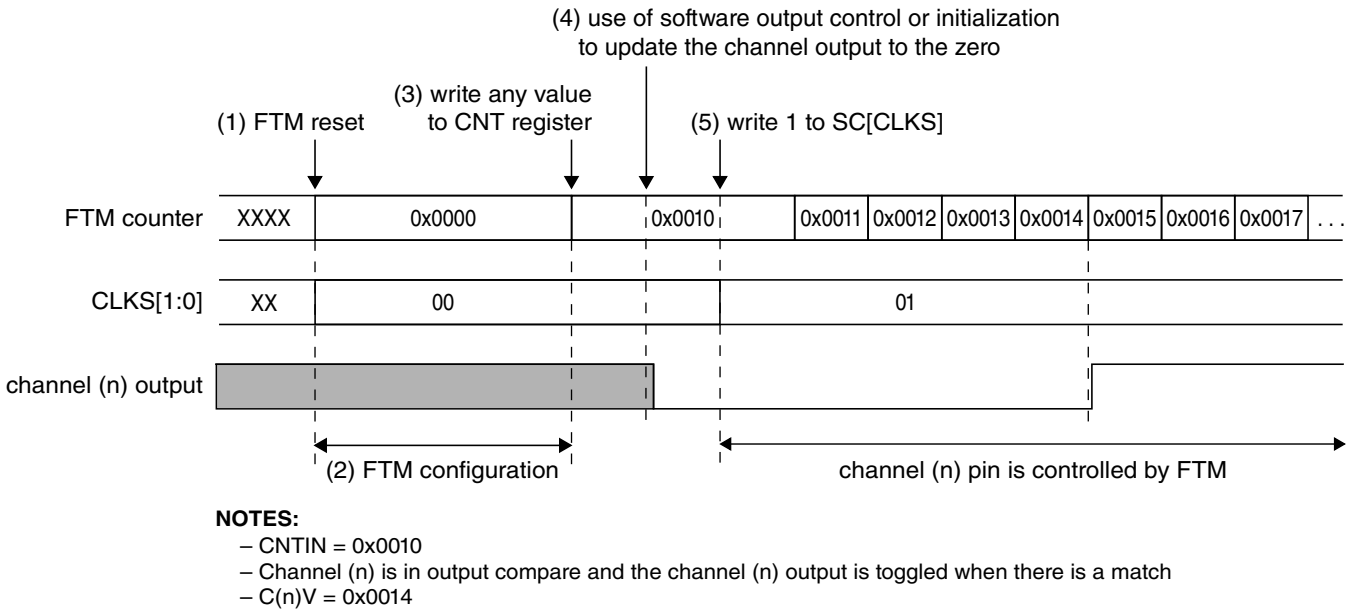


Figure 25-65. FTM behavior after reset when the channel (n) is in Output Compare mode

25.7 FTM Interrupts

25.7.1 Timer Overflow Interrupt

The timer overflow interrupt is generated when (TOIE = 1) and (TOF = 1).

25.7.2 Channel (n) Interrupt

The channel (n) interrupt is generated when (CHnIE = 1) and (CHnF = 1).

25.8 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.
  - 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 26
Frequency-Shift Keying Demodulation Timer (FSKDT)

26.1 Chip-specific FSKDT information

26.1.1 FSKDT interconnections

The FSKDT is designed to demodulate the WPC-Rx coil voltage zero-cross signal from the communication controller. User could configure its input through the signal crossbar (SBAR) in the SIM module, and please refer to the SIM chapter for details.

26.2 Introduction

The frequency-shift keying demodulation timer (FSKDT) module enables the wireless power receiver to demodulate the message sent by WPC-Qi medium power specification compliant transmitter.

26.3 Features

FSKDT includes the following features:

- One frequency-shift keyed signal input channel
- 16-bit free-running counter to count the input signal edge-to-edge period
- Three 16-bit phase counters
  - Each phase counter contains the period count accumulation of programmable consequence (4/8/16/32) cycles of the input signal.

- Interrupt is generated when any individual phase counter is updated or all phase counters are updated.
- Phase counter 0 is always updated by the latest period count accumulation. Previous period accumulations update phase counter 1 and phase counter 2.

- 16-bit current position number, which contains the input signal cycle number since the module is enabled or reset
- Input signal edge-to-edge period error detection
  - Programmable period-too-short error threshold
  - Programmable period-too-long error threshold
  - Interrupt generation when less than 11 bits (FSKDT_DATA[BM] = 0) or 8 bits (FSKDT_DATA[BM] = 1) are received with FSK parking at F_{op}.
- Module software reset
- Message bit-stream detection
  - Bit value (ZERO or ONE) recognition
  - Interrupt generation
- Message byte packing

26.4 Overview

This section presents an overview of the FSKDT module. The following figure illustrates the simplified FSKDT block diagram.

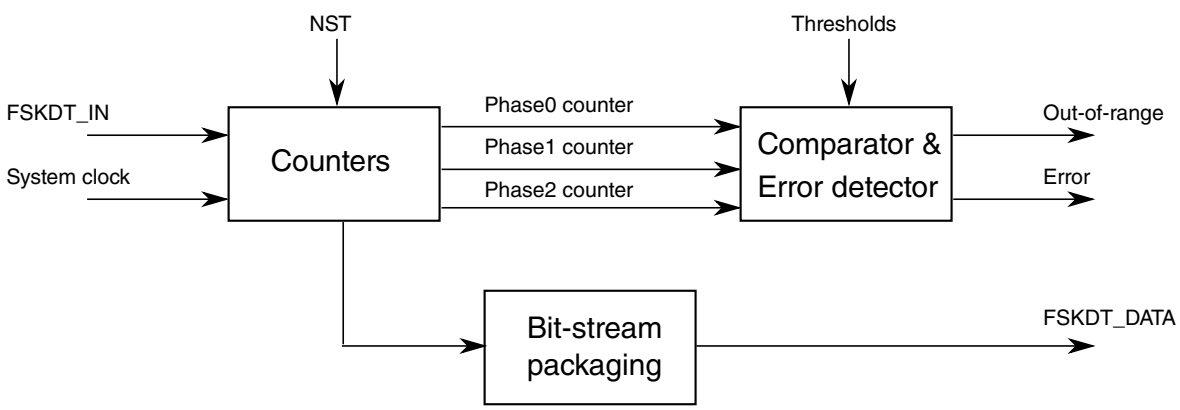


Figure 26-1. Simplified FSKDT block diagram

26.4.1 WPC-Qi transmitter-to-receiver communication

The WPC-Qi specification volume II Medium Power defines that the power transmitter communicates to the power receiver using Frequency Shift Keyed (FSK) modulation. For this purpose, the transmitter modulates the above characteristic of the power signal. The receiver detects this as a change in frequency of the received power signal. In other words, the power transmitter and the power receiver use a modulated power signal to provide a transmitter-to-receiver communication channel.

In the negotiation phase, the power receiver negotiates with the power transmitter to attempt getting a more advanced power transfer contract. The transmitter may modulate the power signal only when responding to Request packets sent by the receiver during the negotiation phase.

26.4.2 WPC-Qi transmitter modulation scheme

The power transmitter modulates the power signal so that the Primary Cell operating frequency varies between two states, namely f_{op} and f_{mod}. A modulation level is characterized in that frequency of the power signal, which is constant at either f_{op} or f_{mod} for 256 ± 3 cycles.

- If FSKPolarity = 0, $1/f_{mod} = 1/f_{op} - (31.25\ ns \times 2^{FSKDepth})$
- If FSKPolarity = 1, $1/f_{mod} = 1/f_{op} + (31.25\ ns \times 2^{FSKDepth})$

The FSKPolarity and the FSKDepth are bit fields in the Tx Modulation Depth Request packet sent from power receiver to power transmitter. FSKDepth value could be 0b00, 0b01, 0b10 or 0b11. The power transmitter uses a differential bi-phase encoding scheme to modulate data bits onto the power signal. For this purpose, the power transmitter aligns each data bit to 512 periods of the power signal frequency.

The transmitter encodes a ONE bit using two transitions in the power signal frequency. The first transition occurs at the beginning of the bit period and the second transition occurs after 256 cycles of the power signal. The transmitter encodes a ZERO bit using a single transition in the power signal frequency and then remaining at the new frequency for 512 cycles of the power signal.

26.5 Modes of operation

FSKDT supports two operating modes: the Disabled mode and the Active mode.

26.5.1 Disabled mode

When the enable bit FSKDT_CR[EN] is cleared, the FSKDT module is disabled, and does not perform any function in any MCU operation mode.

26.5.2 Active mode

In Active mode, FSKDT has its full functionality. FSKDT can be in Active mode with the MCU in any of the following operational modes: Run and Wait.

NOTE

For the clock selection of FSKDT in Active mode, see the chip configuration information.

26.6 Block diagram

The following figure shows the block diagram of FSKDT.

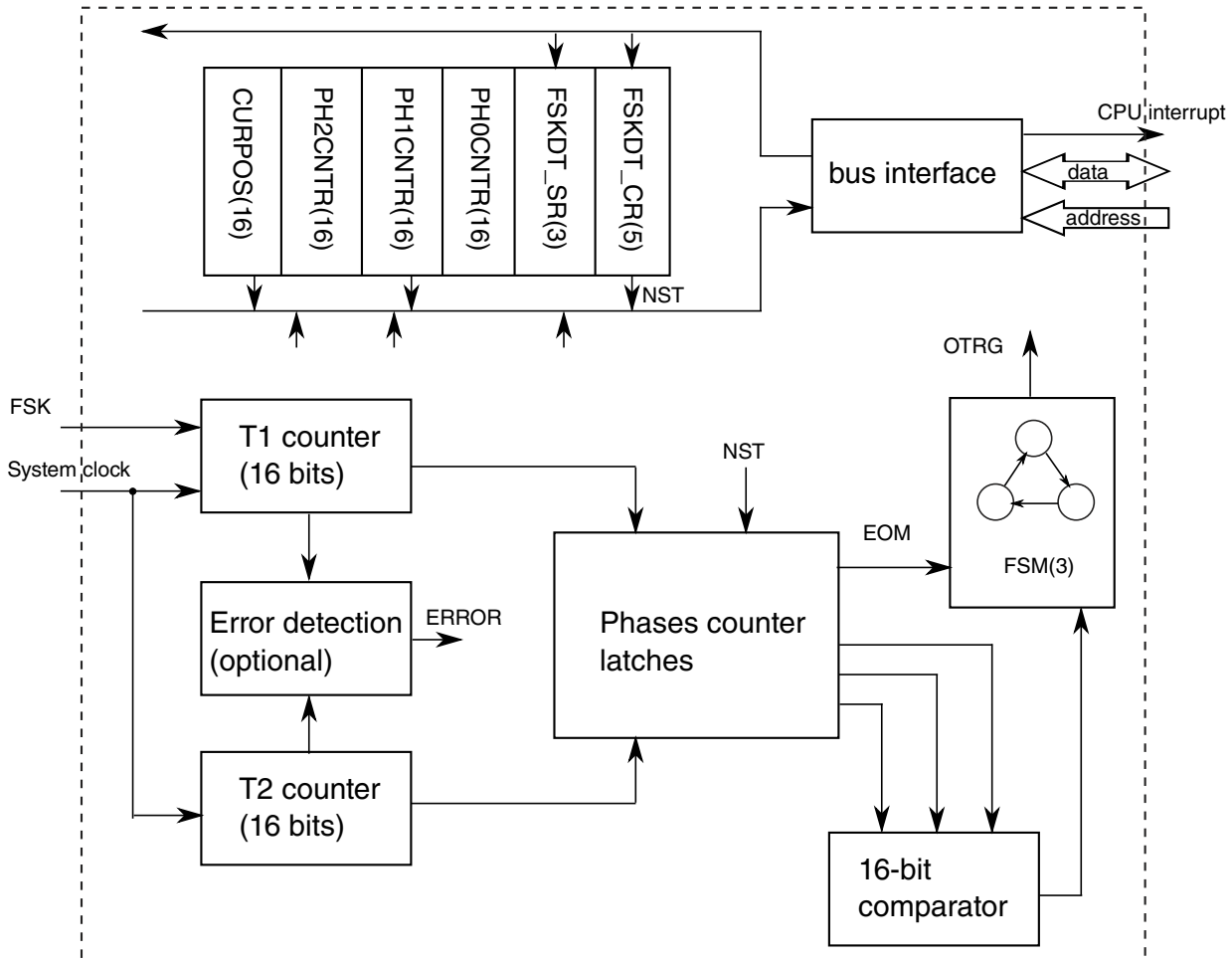


Figure 26-2. FSKDT block diagram

26.7 External signal description

FSKDT has one input signal channel. The following table itemizes all the FSKDT external pins.

Name	Port	Function	Reset State
FSKDT input signal channel	FSKDT_IN	The FSK signal input channel.	0

26.7.1 FSKDT_IN

FSKDT_IN is the FSK signal input channel. This signal represents the wireless power receiver coil voltage frequency. FSKDT_IN signal source can be from GPIO or from other on-chip modules.

26.8 Memory map and register definition

This section includes the module memory map and detailed descriptions of all registers.

FSKDT memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4003_4000	Control Register (FSKDT_CR)	16	R/W	0000h	26.8.1/380
4003_4002	Status Register (FSKDT_SR)	16	R/W	0000h	26.8.2/382
4003_4004	Phase0 counter value (FSKDT_PH0)	16	R/W	0000h	26.8.3/383
4003_4006	Phase1 counter value (FSKDT_PH1)	16	R/W	0000h	26.8.4/383
4003_4008	Phase2 counter value (FSKDT_PH2)	16	R/W	0000h	26.8.5/384
4003_400A	Current Position Register (FSKDT_CURPOS)	16	R/W	0000h	26.8.6/384
4003_400C	Out-of-Period Counter Threshold Register (FSKDT_OPCTH)	16	R/W	0000h	26.8.7/384
4003_400E	Error Threshold Register (FSKDT_ERRTH)	16	R/W	0000h	26.8.8/385
4003_4010	Demodulation Data Register (FSKDT_DATA)	16	R/W	0000h	26.8.9/385
4003_4012	Demodulation Data Status Register (FSKDT_DSR)	16	R/W	0000h	26.8.10/386

26.8.1 Control Register (FSKDT_CR)

Address: 4003_4000h base + 0h offset = 4003_4000h

Bit	15	14	13	12	11	10	9	8
Read	EN	NST			0			
Write				SWRST				
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Read	0				ERRIE	EOAMIE	EOSMIE	OPIE
Write								
Reset	0	0	0	0	0	0	0	0

FSKDT_CR field descriptions

Field	Description
15 EN	FSK demodulation Enable This bit is used to enable or disable the FSKDT module.

Table continues on the next page...

FSKDT_CR field descriptions (continued)

Field	Description
	0 FSK demodulation is disabled. 1 FSK demodulation is enabled.
14–13 NST	Number of Shift Times to measure Defines the number of FSK period to be measured in each phase. 00 4 cycles of FSK are grouped in one phase. 01 8 cycles of FSK are grouped in one phase. 10 16 cycles of FSK are grouped in one phase. 11 32 cycles of FSK are grouped in one phase.
12 SWRST	Software Reset Writing a 1 to this bit resets all the internal states of the FSK demodulator, and the CURPOS register content is reset to 0 as well. Reading this bit always returns 0. **NOTE:** The user needs to perform a software reset after changing any of the FSKDT configurations. All status flags (for example, FSKDT_SR[ERRF], FSKDT_SR[EOAMF] and so on) are cleared by software reset. Moreover, SWRST clears the contents of FSKDT_DATA, FSKDT_DSR, and three phase counters (PH0CNTR, PH1CNTR and PH2CNTR).
11–4 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
3 ERRIE	Interrupt Enable when Error happens See error flag FSKDT_SR[ERRF] to find out which conditions may result in an error response. 0 Interrupt is not enabled when an error happens. 1 Interrupt is enabled when an error happens.
2 EOAMIE	End-Of-All phases Measurement Interrupt Enable This bit is used to enable the interrupt when 3-phase (3 of NST FSK cycles) measurement is completed in each pipeline. 0 End-of-all 3-phase measurement interrupt is disabled. 1 End-of-all 3-phase measurement interrupt is enabled.
1 EOSMIE	End-Of-Single phase Measurement Interrupt Enable This bit is used to enable the interrupt when a single phase measurement (defined by NST) is completed. 0 End-of-single phase measurement interrupt is disabled. 1 End-of-single phase measurement interrupt is enabled.
0 OPIE	Out-of-Period Interrupt Enable This bit is used to enable the interrupt if in the given number of groups(NST) the FSK period counter is out of the specified range. 0 Out-of-Period Interrupt is disabled. 1 Out-of-Period Interrupt is enabled.

26.8.2 Status Register (FSKDT_SR)

Address: 4003_4000h base + 2h offset = 4003_4002h

Bit	15	14	13	12	11	10	9	8
Read	0							
Write								
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Read	0				ERRF	EOAMF	EOSMF	OPF
Write					w1c	w1c	w1c	w1c
Reset	0	0	0	0	0	0	0	0

FSKDT_SR field descriptions

Field	Description
15–4 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
3 ERRF	Error happens during measurement Several possible faults may cause this flag to be set. The first is that no FSK signal is received, or the received FSK signal is too slow. The second is that FSK is too fast to be measured. The threshold to specify how fast/slow of the FSK is configured in the FSKDT_ERRTH register. 0 No error happens. 1 Error happens and the software needs to check the status registers to confirm which is the possible cause to the error.
2 EOAMF	End-Of-All Measurement Flag This bit indicates a completed 3-phase measurement in one pipeline. 0 The 3-phase measurement is not completed. 1 A complete 3-phase measurement finishes.
1 EOSMF	End-Of-Single Measurement Flag This bit indicates one-phase (any of phases 0, 1 and 2) measurement in one pipeline. 0 The one-phase measurement is not completed. 1 A complete one-phase measurement finishes.
0 OPF	Out-of-Period Flag This bit indicates the counter value has its changes out of the range defined by FSKDT_OPCTH in a given number of FSK cycles(NST). 0 The counter value hasn't the changes larger than the threshold. 1 The counter value has significant change that larger than the threshold.

26.8.3 Phase0 counter value (FSKDT_PH0)

Address: 4003_4000h base + 4h offset = 4003_4004h

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	PH0CNTR															
Write																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

FSKDT_PH0 field descriptions

Field	Description
PH0CNTR	Counter values for Phase 0 measurement These read-only bits contain the counter values for phase 0 measurement result. They are refreshed when phase 0 measurement is completed, defined by FSKDT_CR[NST] bits. **NOTE:** Software checks if phase 0 or phase 1 are all zeros. This indicates the beginning of counting and these zeros are ignored.

26.8.4 Phase1 counter value (FSKDT_PH1)

Address: 4003_4000h base + 6h offset = 4003_4006h

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	PH1CNTR															
Write																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

FSKDT_PH1 field descriptions

Field	Description
PH1CNTR	Counter values for Phase 1 measurement These read-only bits contain the counter values for phase 1 measurement result. They are refreshed when phase 1 measurement is completed, defined by FSKDT_CR[NST] bits. **NOTE:** Software checks if phase 0 or phase 1 are all zeros. This indicates the beginning of counting and these zeros are ignored.

26.8.5 Phase2 counter value (FSKDT_PH2)

Address: 4003_4000h base + 8h offset = 4003_4008h

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	PH2CNTR															
Write																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

FSKDT_PH2 field descriptions

Field	Description
PH2CNTR	Counter values for phase 2 measurement These read-only bits contain the counter values for phase 2 measurement result. They are refreshed when phase 2 measurement is completed, defined by FSKDT_CR[NST] bits.

26.8.6 Current Position Register (FSKDT_CURPOS)

Address: 4003_4000h base + Ah offset = 4003_400Ah

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	CURPOS															
Write																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

FSKDT_CURPOS field descriptions

Field	Description
CURPOS	Current Position These read-only bits signalize the current position (indicated with the number of FSK cycles), quantized by every positive edge of FSK clock signal. These bits are cleared by software reset.

26.8.7 Out-of-Period Counter Threshold Register (FSKDT_OPCTH)

Address: 4003_4000h base + Ch offset = 4003_400Ch

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	0						OPCTH									
Write																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

FSKDT_OPCTH field descriptions

Field	Description
15–10 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
OPCTH	Out-of-Period Counter threshold These bits specify the counter value change threshold with a give number of FSK cycles(NST). It is required to configured this register with half-word or word accessing. **NOTE:** Out-of-range condition: each cycle of capture needs to compare with its previous content, except for the first two cycles after software reset.

26.8.8 Error Threshold Register (FSKDT_ERRTH)

Address: 4003_4000h base + Eh offset = 4003_400Eh

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read Write	HILMT										LOLMT					
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

FSKDT_ERRTH field descriptions

Field	Description
15–6 HILMT	High Limitation These bits specify the largest system clock cycles when the FSK must move forward by one cycle, otherwise error happens. It is required to use half-word (16 bits) to access the HILMT filed.
LOLMT	Low Limitation These bits specify the lowest system clock cycles that must be contained by one FSK cycle, otherwise error happens.

26.8.9 Demodulation Data Register (FSKDT_DATA)

Address: 4003_4000h base + 10h offset = 4003_4010h

Bit	15	14	13	12	11	10	9	8
Read	NBIE	WBNBIE	BM	0		DATA		
Write								
Reset	0	0	0	0	0	0	0	0

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

FSKDT_DATA field descriptions

Field	Description
15 NBIE	New Bit decoding Interrupt Enable Interrupt enable when decoding one new bit of FSK signal. 0 Interrupt is not enabled when decoding one new bit of FSK signal. 1 Interrupt is enabled when decoding one new bit of FSK signal.
14 WBNBIE	Whole Byte Bits decoding finish Interrupt Enable Interrupt enable when decoding one whole byte of FSK signal. The bits number per byte is determined by FSKDT_DATA[BM] (11 bits per byte when BM=0, and 8 bits per byte when BM=1). 0 Interrupt is not enabled when decoding one whole byte of FSK signal. 1 Interrupt is enabled when decoding one whole byte of FSK signal.
13 BM	Package Byte Mode 0 11-bit mode. 1 8-bit mode.
12–11 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
DATA	The demodulated FSK data These bits are cleared by software reset.

26.8.10 Demodulation Data Status Register (FSKDT_DSR)

Address: 4003_4000h base + 12h offset = 4003_4012h

Bit	15	14	13	12	11	10	9	8
Read	NBF	WBNF	0					
Write	w1c	w1c						
Reset	0	0	0	0	0	0	0	0

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

FSKDT_DSR field descriptions

Field	Description
15 NBF	New Bit shift to DATA Flag This flag indicates a new decoded FSK bit is shifted to the DATA field. In the first byte(either 8-bit or 11-bit) demodulation, the new bit flag will assert 1 more times than the actual new bits we have received, and the first NBF is just used for synchronization which must be ignored.

Table continues on the next page...

FSKDT_DSR field descriptions (continued)

Field	Description
14 WBNF	New Whole Byte decoded Flag This flag indicates a complete byte (11 bits when FSKDT_DATA[BM]=0, and 8 bits when FSKDT_DATA[BM]=1) is decoded since last time. **NOTE:** It is suggested the software to read out the DATA contents before clearing this flag.
13–4 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
TCNTD	Total Count for bits that have been Decoded These read-only bits indicate how many bits have been decoded to FSK_DATA after a new frame starts.

26.9 Functional description

The following sections describe functional details of the module.

26.9.1 Phase counter

FSKDT is designed to detect the frequency change of input signal. Instead of counting the input signal edge-to-edge period and then alarming the CPU on every cycle, FSKDT groups a configurable number of input signal cycles into one phase. To facilitate the detection process, this module implements three phase counters.

FSKDT contains one free-running shadow counter to monitor the input signal period. Software does not have to access this shadow counter every input signal cycle, but could access the phase which contains the period length information of input signal within certain cycles.

By configuring FSKDT_CR[NST], user could group 4, 8, 16 or 32 input signal cycles into one phase:

- NST = 0b00 — 4 cycles of FSK are grouped in one phase
- NST = 0b01 — 8 cycles of FSK are grouped in one phase
- NST = 0b10 — 16 cycles of FSK are grouped in one phase
- NST = 0b11 — 32 cycles of FSK are grouped in one phase

Supposing NST bits are configured to 0b01, the phase counter then contains the sum of period of eight consequent input signal cycles.

26.9.1.1 Phase counter update

FSKDT contains three phase counters, named FSKDT_PH0CNTR, FSKDT_PH1CNTR and FSKDT_PH2CNTR. These three phase counters form a pipeline structure which enables them to contain the same cycle length information of input signal at different time.

Phase counter 0 (FSKDT_PH0CNTR) always contains the latest phase length of input signal. Phase counter 1 (FSKDT_PH1CNTR) contains the length of the second-to-last (penultimate) phase, while phase counter 2 (FSKDT_PH2CNTR) contains the length of the third-to-last (antepenultimate) phase.

When a phase (grouped by 4, 8, 16 or 32 input signal cycles determined by the NST bits value) length is captured by the shadow free-running counter, its length is then latched by FSKDT_PH0CNTR. At this moment, the previous content of FSKDT_PH0CNTR is shifted to FSKDT_PH1CNTR. Likewise, what previously stored in FSKDT_PH1CNTR is shifted to FSKDT_PH2CNTR, with FSKDT_PH2CNTR's previous value dropped.

User could observe the input signal frequency change by monitoring these three phase counters.

26.9.1.2 Phase counter update interrupt

When FSKDT_CR[EOSMIE] is set, FSKDT can generate interrupt every time a new single phase is captured by phase counter 0, 1 or 2.

When FSKDT_CR[EOAMIE] is set, FSKDT can generate interrupt after three new phases are captured by the three phase counters.

26.9.2 Current position counter

As defined in the WPC-Qi specification, the power transmitter align each data bit to 512 periods of the power signal frequency. The transmitter encodes a ONE bit using two frequency transitions in the power signal frequency. The first frequency transition occurs at the beginning of the bit period, and the second frequency transition occurs after 256 cycles of the power signal. The transmitter encodes a ZERO bit using a single transition in the power signal frequency and then remaining at the new frequency for 512 cycles of the power signal.

To trace the signal period counts, the current position counter (FSKDT_CURPOS) contains the number of cycles, of the FSK input signal detected. This counter has an increment on every input signal rising edge. When this counter reaches its maximum value (0xFFFF), in the next FSK cycle it then counts from 0x20.

26.9.3 Module software reset

Writing 1 to FSKDT_CR[SWRST] can generate software reset to FSKDT. The software reset clears three phase counters (FSKDT_PH0CNTR, FSKDT_PH1CNTR and FSKDT_PH2CNTR), the current position counter (FSKDT_CURPOS) and all status flags, and also resets other FSKDT internal logics. Clearing the FSKDT_CR[EN] bit has all the similar effects during disabling FSKDT, except that FSKDT could not function until the enable bit is set again.

26.9.4 Error detection

FSKDT is capable to detect the following error conditions of input signal:

- Period-too-long error: By configuring FSKDT_ERRTH[HILIM] to a non-zero value, user can set the period-too-long threshold. When the input signal edge-to-edge period is longer than HILIM × 8 FSKDT clock counts, the flag FSKDT_CR[ERRF] is set.
- Period-too-short error: By configuring FSKDT_ERRTH[LOLIM] to a non-zero value, user can set the period-too-short threshold. When the input signal edge-to-edge period is shorter than LOLIM × 8 FSKDT clock counts, the flag FSKDT_CR[ERRF] is set.

If FSKDT_CR[ERRIE] is set, FSKDT generates interrupt when either error condition happens.

26.10 Application information

WPC-Qi is a bi-directional communication protocol. Usually it is the receiver to initialize the transfer request to the transmitter, and then the transmitter responses to this request and sends the packages. Before the request sending to transmitter, it is suggested to first configure FSKDT_CR[NST] with a proper value, and then enable the FSKDT module with the desired interrupts by setting FSKDT_CR[EN] to 1 and setting the corresponding interrupt enable bits in FSKDT_CR. Normally, the smaller depth of FSK is configured for transmitter, the bigger value of FSKDT_CR[NST] should be selected.

To report the case that too wide or too narrow FSK signal is received, FSKDT_ERRTH register can be utilized. Namely, FSKDT_ERRTH[HILMT]×8 is used to specify the maximum system clock cycles that are between two FSK transaction edges, otherwise it means FSK is too slow; while FSKDT_ERRTH[LOLMT]×8 defines the minimal system clock cycles that must contain within one FSK cycle, otherwise it means FSK is too fast. In either case, FSKDT will report an error response.

The three phases of FSKDT counter (FSKDT_PH0, FSKDT_PH1 and FSKDT_PH2), together with FSKDT_CURPOS and FSKDT_FPCTH registers, are opened to end-users to develop specific algorithm for FSK signal parsing. Depending on the CPU load, user can select the interrupt either from each single phase measurement (FSKDT_CR[EOSMIE]) or from all the three phases of measurement (FSKDT_CR[EOAMIE]). Figure 26-13 shows when the single-phase measurement interrupt and the all-phase measurement interrupt will happen.

To relieve the system load, FSKDT can be selected to generate the interrupt only when the counter value's change is larger than the FSK period change threshold, specified by FSKDT_FPCTH. Supposing FSKDT_CR[NST]=2, using the following conditions for example:

- F_{op} operates at 200 kHz and FSKDepth is set to 2'b01 (variations between F_{op} and F_{mod} is 62.5 ns).
- System clock to the FSKDT module runs at 48 MHz.

With these assumptions, we can calculate that when FSK runs at F_{op}, the counter values we get should be around 48M × 16 / 200K = 3840, and the counter value will change to 48M × 16 / 197.53K = 3888. Considering the worst case, which is one of the measured phase counter has half of its content compose of F_{op} and the other half compose of F_{mod}, its counter value should be 48M ×8 / 200K + 48M × 8 / 197.53K = 3864. We can consider to specify the FSKDT_FPCTH < 24, which is 3864 – 3840. Given the FSK clock will not have an ideally good shape, we can specify FSKDT_FPCTH = 18 for this case.

Meanwhile, FSKDT module itself implements a method to parse the received FSK signal to the original signal bits and packages. This process is shown in Figure 26-13 and the data workflow is illustrated in Figure 26-14.

In Figure 26-13, FSKDT_CR[NST] = 3, which means after every 32 FSK cycles, the phase counters will be refreshed. Also, from Figure 26-13 it is shown that for each time, PH0 counter always retains the latest measurement results, while PH2 counter always retains the oldest counter values.

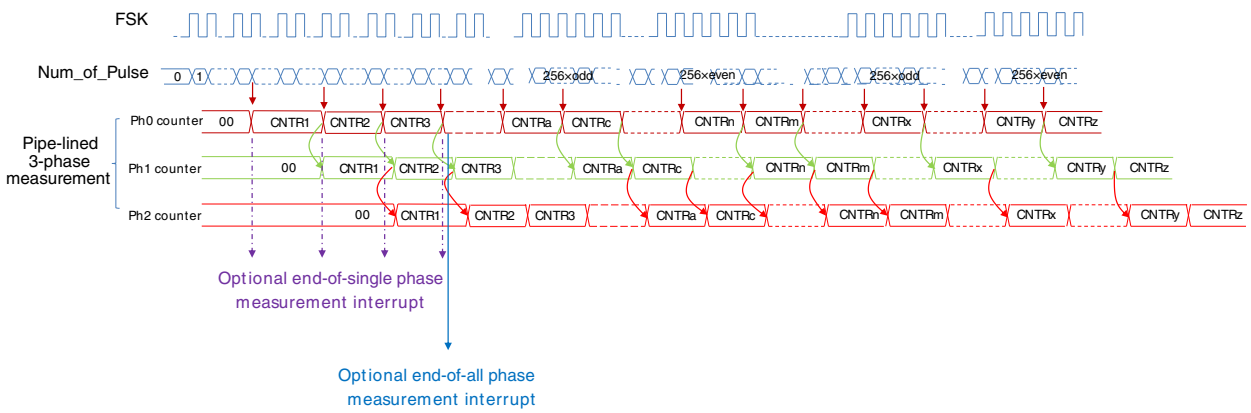


Figure 26-13. FSKDT counters loading sequence

Figure 26-14 illustrates the data workflow by which FSKDT module is used to parse the FSK data. At the end of new bit parse flags by FSKDT_DSR[NBF], the latest FSK data bit is shifted to the most significant bit (MSB) in the FSKDT_DATA register. Depending on the FSK data format, at the end of new frame parse event flags by FSKDT_DSR[WBNF], the whole frame bits are stored in the FSKDT_DATA register.

NOTE

For the 8-bit mode, the 8-bit data are stored in FSKDT_DATA[10:3].

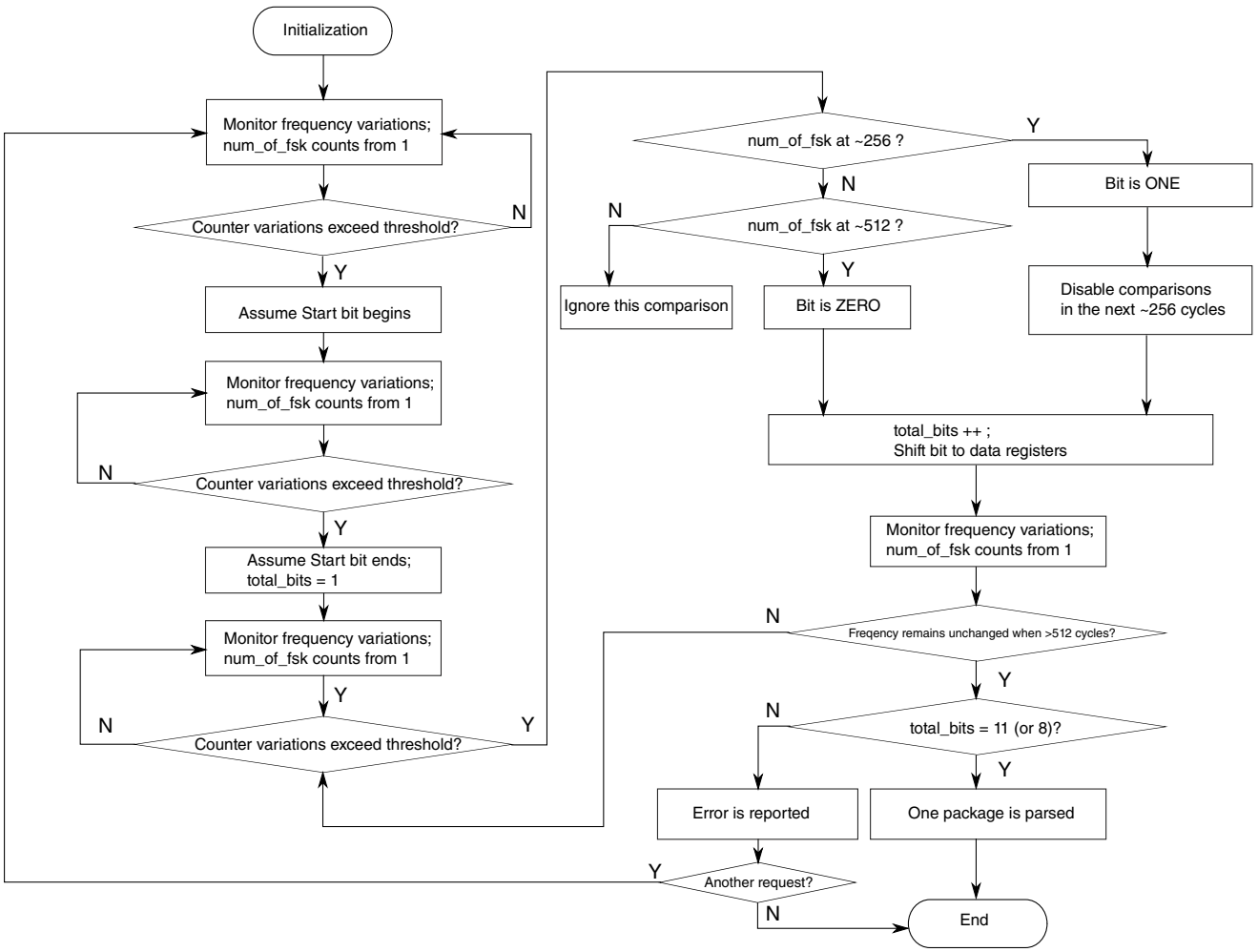


Figure 26-14. FSKDT data parsing flow

Chapter 27
Communication and Clamp Controller (CNC)

27.1 Chip-specific CNC information

27.1.1 Communication and clamp controller interconnections

The communication controller can output the receiver coil voltage (AC1/AC2) zero crossing signal to other on-chip modules.

The rectifier clamp controller can also accept external clamp driver signal from other on-chip modules. User can configure the external clamp source through the trigger crossbar (TBAR) in the SIM module, and please refer to the SIM chapter for details.

27.2 Introduction

The communication and clamp controller (CNC) module is designed for WPC–Qi compliant wireless power receiver. It acts as a communication and control unit of the power receiver.

There are mainly three functions for the CNC module.

- The wireless power transmitter-to-receiver (Tx-to-Rx)
- The external wired power (AD_IN) detection and control
- The wireless power receiver rectifier over-voltage and low voltage detection and protection

27.3 Features

CNC includes the following features:

- A wireless power Tx-to-Rx zero-crossing detection sub-module

- Wireless power receiver coil voltage frequency detection from the pins AC1 and AC2, supporting up to 300 kHz AC input
- Low jitter on AC1 and AC2 voltage zero-crossing comparator
- Programmable glitch rejection

- Supports external wired power (for example, USB adaptor) plug-in
  - Supports 5 V input on the pin AD_IN
  - Switches on/off when valid wired power plugs in/out (could be set as off by default)
  - Over-voltage and low voltage protection on AD_IN
  - Programmable digital filter to make AD_IN status immune to glitches
- Rectified voltage monitor
  - Over-voltage and low voltage detection on VREC
  - Rectifier over-voltage clamp driver
  - Programmable digital filter to make VREC status immune to glitches

27.4 Overview

This section presents an overview of the CNC module. The following figure illustrates the simplified CNC block diagram.

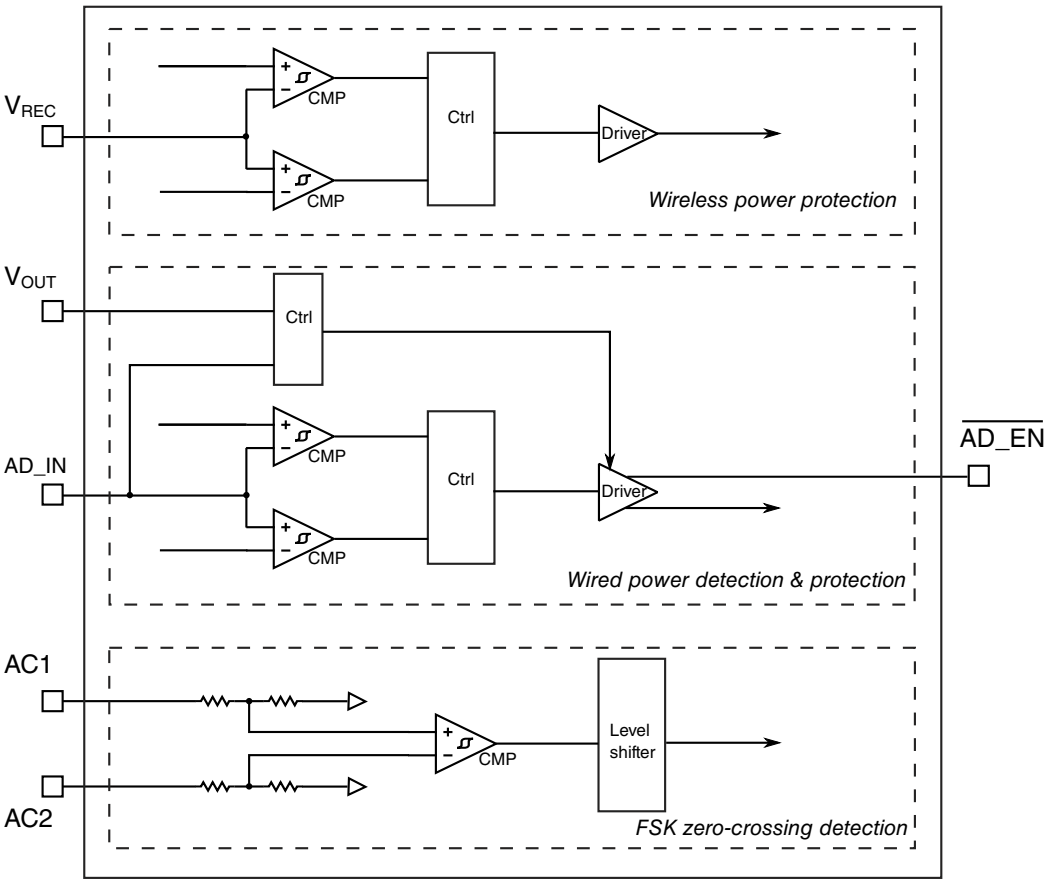


Figure 27-1. Simplified CNC block diagram

27.5 Modes of operation

CNC supports two operating modes: the Disabled mode and the Active mode.

27.5.1 Disabled mode

When the CNC module enable bits are cleared, CNC is disabled and does not perform any function in any MCU operational mode.

27.5.2 Active mode

In the Active mode, the CNC module has its full functionality. CNC can be in Active mode with the MCU in any of the following operational modes: Run and Wait.

27.6 External signal description

CNC has one input signal channel. The following table itemizes all the CNC external pins.

Signal	I/O	Detailed description
V_{REC}	I	Rectifier voltage input
CLAMP	O	Clamp driver for rectifier overvoltage protection
V_{OUT}	I	Wireless power receiver voltage output
AD_IN	I	Wired power voltage input
$\overline{AD\_EN}$	O	Wired power PMOS switch
AC1	I	AC input 1 from wireless power receiver coil
AC2	I	AC input 2 from wireless power receiver coil
AC1_DIV	O	Output after AC1 divided controlled by AC1DIVOE
AD_DIV	O	Output after AD divided controlled by VADDIVOE
VREC_DIV	O	Output after VREC divided

27.6.1 VREC

VREC is an analog ultra-high voltage (UHV) input from wireless power receiver rectifier. It is monitored to prevent over-voltage or low voltage conditions.

27.6.2 CLAMP

CLAMP is the rectifier overvoltage clamp driver output. The driver turns on the external switch to connect AC1/AC2 with GND by two capacitors. Thus it creates a low-impedance pass to prevent device damage.

27.6.3 VOUT

VOUT is an analog input from the wireless power receiver output. It is used to control the two external PMOS switches between AD_IN and VOUT.

27.6.4 AD_IN

AD_IN is an analog UHV input and normally supports external USB power connection. This signal is used to control the two external PMOS switches between AD_IN and VOUT. Besides, it is also monitored when it is too high or too low.

27.6.5 $\overline{\text{AD\_EN}}$

$\overline{\text{AD\_EN}}$ is an analog UHV output, and it is used to control the two external PMOS switches between AD_IN and V_{OUT}.

27.6.6 AC1/AC2

AC1 and AC2 are analog UHV inputs from the wireless power receiver coil. They are also responsible for the communications between power receiver and power transmitter, as the wireless power transmitter modulates message by changing the coil voltage frequency.

27.7 Memory map and register definition

This section includes the module memory map and detailed descriptions of all registers.

CNC memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4006_E000	Control Register (CNC_CR)	16	R/W	0000h	27.7.1/398
4006_E002	Analog Configuration Register 1 (CNC_ANACFG1)	16	R/W	See section	27.7.2/400
4006_E004	Analog Configuration Register 2 (CNC_ANACFG2)	16	R/W	0000h	27.7.3/401
4006_E006	Status Register (CNC_STAS)	16	R/W	0000h	27.7.4/403
4006_E008	VREC Overvoltage clamp drive Enable Register (CNC_VRECOE)	16	R/W	0000h	27.7.5/404
4006_E00A	VREC Filter Control Register (CNC_VRECFLTC)	16	R/W	0000h	27.7.6/404
4006_E00C	AD_IN Filter Control Register (CNC_VADFLTC)	16	R/W	0000h	27.7.7/405

27.7.1 Control Register (CNC_CR)

Address: 4006_E000h base + 0h offset = 4006_E000h

Bit	15	14	13	12	11	10	9	8
Read	ZCDE	ADANE	OVIE	LVIE	0	ADE	SWADS	VADVDIE
Write								
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Read	VRECDE	EXTCLPDE	CLPDPE	0				SWCLPD
Write								
Reset	0	0	0	0	0	0	0	0

CNC_CR field descriptions

Field	Description
15 ZCDE	Zero-Crossing Detection Enable This bit is used for the zero-crossing detection of the FSK signal. In Stop mode the FSK zero-crossing function is disabled automatically. 0 Zero-crossing detection is disabled. 1 Zero-crossing detection is enabled.
14 ADANE	AD_IN Detection Enable This bit is used to enable the analog sub-module to detect the wired power (AD_IN). **NOTE:** When this bit ADANE is set to 1, a short glitch may appear on CNC_STAS[VADOK4P5] and CNC_STAS[VADOV5P5] flags from the analog circuit. 0 Analog is disabled for AD_IN check. 1 Analog is enabled for AD_IN check.
13 OVIE	VREC Overvoltage Interrupt Enable This bit is used to enable the interrupt when the VREC is overvoltaged. 0 VREC overvoltage interrupt is disabled. 1 VREC overvoltage interrupt is enabled.
12 LVIE	VREC Low Voltage Interrupt Enable This bit is used to enable the interrupt when the VREC is low-voltaged. 0 VREC low voltage interrupt is disabled. 1 VREC low voltage interrupt is enabled.
11 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
10 ADE	Adapter Switch Enable This bit is used to enable the adapter switch when a valid AD_IN is plugged in.

Table continues on the next page...

CNC_CR field descriptions (continued)

Field	Description
	0 Adapter switch is disabled. The adapter switch is not enabled even when the plugged AD_IN voltage is valid. 1 Adapter switch is enabled. When analog is enabled for AD_IN check and the plugged AD_IN is valid, the adapter switch is enabled.
9 SWADS	Software Adapter Switch This bit is used by software to assert the adapter switch enable. 0 Software does not assert the adapter switch. 1 Software asserts the adapter switch.
8 VADVDIE	AD_IN Voltage Valid state change Interrupt Enable This bit is used to enable the interrupt, when the wired power (AD_IN) changes from valid range to invalid range, or vice versa. 0 AD_IN voltage valid state change interrupt is disabled. 1 AD_IN voltage valid state change interrupt is enabled.
7 VRECDE	VREC Detection Enable This bit is used to enable the analog sub-module to check VREC range. In Stop mode the VREC range detection function is disabled automatically. 0 Analog is disabled for VREC range check. 1 Analog is enabled for VREC range check.
6 EXTCLPDE	External Clamp Drive Enable This bit is used to enable the external clamp drive, which may come from other on-chip peripherals. It is better to open the clamp drive, when VREC detection is enabled (VRECDE is set). 0 External clamp drive is disabled. 1 External clamp drive is enabled.
5 CLPDPE	Clamp Drive Port Enable This bit is used to enable the port which is used for clamp drive. 0 The port for clamp drive is disabled. 1 The port for clamp drive is enabled.
4–1 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
0 SWCLPD	Software sets Clamp Drive This bit is used by software to set or to clear the clamp drive directly. See CNC_VRECOE[VRECOVE] for more related information. 0 Software does not set clamp drive directly. 1 Clamp drive is set by software.

27.7.2 Analog Configuration Register 1 (CNC_ANACFG1)

Address: 4006_E000h base + 2h offset = 4006_E002h

Bit	15	14	13	12	11	10	9	8
Read	ZCDHYST		AC1DIVOE	VRECOVLVL		0		
Write								
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Read	0			VADOVTRM		VADLVTRM		VADDIVOE
Write								
Reset	0	0	0	*	*	*	*	0

* Notes:
- VADOVTRM field: The reset value is loaded from the IFR automatically after POR.
- VADLVTRM field: The reset value is loaded from the IFR automatically after POR.

CNC_ANACFG1 field descriptions

Field	Description
15–14 ZCDHYST	Hysteresis control for FSK Zero-Crossing Detection Refer to the chip datasheet to get the detailed hysteresis values at each hysteresis level. 00 Hysteresis level 0. 01 Hysteresis level 1. 10 Hysteresis level 2. 11 Hysteresis level 3.
13 AC1DIVOE	AC1 Divided Output Enable **NOTE:** If CNC_CR[ZCDE] is not set, it is suggested this bit should be cleared by software as well. 0 AC1 divided output is disabled. 1 AC1 divided output is enabled.
12–11 VRECOVLVL	VREC Over Voltage Level Configuration This field is used to configure the VREC over-voltage levels, refer to chip datasheet to get the actual level with each configurations. 00 Over-voltage level 0. 01 Over-voltage level 1. 10 Over-voltage level 2. 11 Over-voltage level 3.
10–5 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
4–3 VADOVTRM	Trim value set for the AD_IN overvoltage (VADOV5P5) detection These trim values are loaded from IFR automatically after POR. However, software can update them for different settings. Refer to the chip datasheet to get the detailed threshold values at each threshold level.

Table continues on the next page...

CNC_ANACFG1 field descriptions (continued)

Field	Description
	00 Overvoltage threshold trim level 0. 01 Overvoltage threshold trim level 1. 10 Overvoltage threshold trim level 2. 11 Overvoltage threshold trim level 3.
2–1 VADLVTRM	Trim value set for the AD_IN low voltage (VADOK4P5) detection These trim values are loaded from IFR automatically after POR. However, software can update them for different settings. Refer to the chip datasheet to get the detailed threshold values at each threshold level. 00 Low voltage threshold trim level 0. 01 Low voltage threshold trim level 1. 10 Low voltage threshold trim level 2. 11 Low voltage threshold trim level 3.
0 VADDIVOE	AD_IN Divided Output Enable In Stop mode the AD_IN divided output is disabled automatically. **NOTE:** If CNC_CR[ADANE] is not set, it is suggested this bit should be cleared by software as well. 0 The AD_IN divided output is disabled. 1 The AD_IN divided output is enabled.

27.7.3 Analog Configuration Register 2 (CNC_ANACFG2)

Address: 4006_E000h base + 4h offset = 4006_E004h

Bit	15	14	13	12	11	10	9	8
Read	0						VAOHY40	VAOHY20
Write								
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Read Write	BIAE	ACFLTE	ACFLTC		ACDIV			
Reset	0	0	0	0	0	0	0	0

CNC_ANACFG2 field descriptions

Field	Description
15–10 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
9 VAOHY40	40 mV Hysteresis control for the comparator to compare AD_IN and VOUT When this bit is set, a 40 mV hysteresis is applied to the comparator to compare AD_IN and VOUT. **NOTE:** VAOHY40 and VAOHY20 can be in use as combined. If both bits are set to 1, 60 mV hysteresis is applied to the comparator. 0 There is no 40 mV hysteresis applied. 1 40 mV hysteresis is applied to the comparator.

Table continues on the next page...

CNC_ANACFG2 field descriptions (continued)

Field	Description
8 VAOHY20	20 mV Hysteresis control for the comparator to compare AD_IN and VOUT When this bit is set, a 20 mV hysteresis is applied to the comparator to compare AD_IN and VOUT. **NOTE:** VAOHY40 and VAOHY20 can be in use as combined. If both bits are set to 1, 60 mV hysteresis is applied to the comparator. 0 There is no 20 mV hysteresis applied. 1 20 mV hysteresis is applied to the comparator.
7 BIAE	Bias Enable for the CNC module This bit is used to enable the bias current for the CNC module. Set it about 10 µs before enabling other sub-modules, and clear it at the same time with other sub-modules. 0 The current bias is disabled. 1 The current bias is enabled.
6 ACFLTE	AC Filter Enable 0 The AC filter is disabled. 1 The AC filter is enabled.
5–4 ACFLTC	AC Filter Configuration This field is used to select AC filter configuration. 00 AC filter configuration 0, the cut off frequency is 600 kHz. 01 AC filter configuration 1, the cut off frequency is 400 kHz. 10 AC filter configuration 2, the cut off frequency is 200 kHz. 11 AC filter configuration 3, the cut off frequency is 100 kHz.
ACDIV	AC Divider selection This field is used to select dividing ratio. 0000 The dividing ratio is 2/128. 0001 The dividing ratio is 4/128. 0010 The dividing ratio is 5/128. 0011 The dividing ratio is 6/128. 0100 The dividing ratio is 7/128. 0101 The dividing ratio is 8/128. 0110 The dividing ratio is 9/128. 0111 The dividing ratio is 10/128. 1000 The dividing ratio is 11/128. 1001 The dividing ratio is 12/128. 1010 The dividing ratio is 13/128. 1011 The dividing ratio is 15/128. 1100 The dividing ratio is 16/128. 1101 The dividing ratio is 17/128. 1110 The dividing ratio is 19/128. 1111 The dividing ratio is 21/128.

27.7.4 Status Register (CNC_STAS)

Address: 4006_E000h base + 6h offset = 4006_E006h

Bit	15	14	13	12	11	10	9	8
Read	0							
Write								
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Read	VRECLVS	VRECOVS	VRECLVF	VRECOVF	VADCHGF	VADOK4P5	VADOV5P5	FSKO
Write			w1c	w1c	w1c			
Reset	0	0	0	0	0	0	0	0

CNC_STAS field descriptions

Field	Description
15–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7 VRECLVS	VREC Low Voltage State This read-only bit indicates the states of VREC low voltage after the filter.
6 VRECOVS	VREC Overvoltage State This read-only bit indicates the states of VREC overvoltage after the filter.
5 VRECLVF	VREC Low Voltage Flag This sticky bit indicates the VREC voltage has been across low voltage, at least one time, after the filter. Writing a 1 to this bit clears the flag, as well as the LV filter contents.
4 VRECOVF	VREC Overvoltage Flag This sticky bit indicates the VREC voltage has been across overvoltage, at least one time, after the filter. Writing a 1 to this bit clears the flag, as well as the OV filter contents.
3 VADCHGF	AD_IN Change status Flag This sticky bit indicates the AD_IN voltage has changed its state (either from valid to invalid, or vice versa) at least one time after the filter. Writing a 1 to this bit clears the flag.
2 VADOK4P5	AD_IN above 4.5 V state This read-only bit indicates AD_IN is above 4.5 V, and it is signalized after the filter. **NOTE:** The clock source to this filter is divided by 16 by default.
1 VADOV5P5	AD_IN above 5.5 V state This read-only bit indicates AD_IN is above 5.5 V, and it is signalized after the filter. **NOTE:** The clock source to this filter is NOT divided by 16 by default.
0 FSKO	FSK AC1/AC2 zero-crossing status This read only bit indicates the status of the AC1/AC2 zero-crossing.

27.7.5 VREC Overvoltage clamp drive Enable Register (CNC_VRECOE)

Address: 4006_E000h base + 8h offset = 4006_E008h

Bit	15	14	13	12	11	10	9	8
Read	0							
Write								
Reset	0	0	0	0	0	0	0	0

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

CNC_VRECOE field descriptions

Field	Description
15–1 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
0 VRECOVE	VREC Overvoltage clamp drive Enable This bit is used to enable the clamp drive path from VREC overvoltage status. It is write-once after reset. When already set to 1, only the system reset can clear this bit to 0. 0 VREC overvoltage is not enabled for the clamp drive. 1 VREC overvoltage is enabled for the clamp drive.

27.7.6 VREC Filter Control Register (CNC_VRECFLTC)

Address: 4006_E000h base + Ah offset = 4006_E00Ah

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read Write	CNT															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CNC_VRECFLTC field descriptions

Field	Description
CNT	Filter sample Count for VREC overvoltage or low voltage detection The value represents the number of consecutive samples that must agree, prior to that the filter accepts a new overvoltage or low voltage state.The clock source to the VREC overvoltage is bus clock and to the low voltage filters is the bus clock divided by 16. 0 Filter is disabled. None-zero value Any other none-zero value specifies the CNT number of consecutive samples that must agree.

27.7.7 AD_IN Filter Control Register (CNC_VADFLTC)

Address: 4006_E000h base + Ch offset = 4006_E00Ch

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read Write	CNT															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CNC_VADFLTC field descriptions

Field	Description
CNT	Filter sample Count for AD_IN overvoltage or low voltage detection The value represents the number of consecutive samples that must agree, prior to that the filter accepts a new overvoltage or low voltage state. In the Stop mode, the filter is also bypassed and the filter output just follows the raw input. 0 Filter is disabled. Non-zero value Any other none-zero value specifies the CNT number of consecutive samples that must agree.

27.8 Functional description

The following sections describe functional details of the module.

27.8.1 Wireless power Tx-to-Rx message demodulator

CNC integrates a wireless power Tx-to-Rx message demodulator. This module is design to detect the AC input frequency from the pins AC1 and AC2. The module implements a comparator with configurable hysteresis to compare the voltages on AC1 and AC2.

To enable the wireless power Tx-to-Rx message demodulator, user should set CNC_CR[ZCDE].

27.8.1.1 Comparator hysteresis configuration

The AC1/AC2 zero-crossing comparator hysteresis can be configured through the ZCDHYST[1:0] bits in CNC_ANACFG1 register.

27.8.1.2 Low-pass filter on AC1/AC2

Before the AC1/AC2 zero-crossing comparator input, a low-pass filter with configurable cut-off frequency is implemented to filter the noise.

User can configure CNC_ANACFG2[ACFLTC] to set the filter cut-off frequency.

27.8.2 Wired power (AD_IN) detection and control

CNC integrates a sub-module of wired power detection and control. This sub-module can achieve the power supply smooth switching between wireless power and wired power for a mobile device.

27.8.2.1 AD_IN monitor

The wired power, for example the bus voltage from a USB adaptor, can be monitored from the pin AD_IN. To enable this monitor, user should set CNC_CR[ADANE].

This AD_IN monitor can maintain functioning under the MCU Low-power mode.

27.8.2.2 AD_IN monitor interrupt

When CNC_CR[VADVDIE] is set, the AD_IN monitor can generate an interrupt when the wired power status changes.

The status changes include: AD_IN becomes valid/invalid when the AD_IN voltage alters to be within/beyond the valid thresholds (4.5 V ~ 5.5 V).

The AD_IN status digital glitch filter (VADFLT) can be enabled when CNC_VADFLTC[CNT] is configured to any non-zero value. The CNC_STAS[VADCHGF] flag is set and an interrupt is generated only when the AD_IN status change is valid by this filter.

NOTE

When AD_IN status transition occurs across the valid low threshold, the filter clock is the bus clock divided by 16. When AD_IN status transition occurs across the valid high threshold, the filter clock is the bus clock.

27.8.2.3 AD_IN monitor flags

The AD_IN voltage can also be monitored through the VADOK4P5 flag and the VADOV5P5 flag in CNC_STAS register. Whenever the AD_IN voltage is above the valid low threshold, the CNC_STAS[VADOK4P5] flag is set. Whenever the AD_IN voltage is above the valid high threshold, the CNC_STAS[VADOV5P5] flag is set.

The status of these two flags is affected by the AD_IN status digital glitch filter (VADFLT).

27.8.2.4 AD_IN switch ($\overline{\text{AD\_EN}}$) control

AD_IN switch is achieved by the output $\overline{\text{AD\_EN}}$, which is the PMOSFET gate driver. The turning on and off of this switch can be controlled by:

- AD_IN voltage monitor
- Software

NOTE

CNC_CR[ADANE] must be set, as a prerequisite.

When CNC_CR[ADE] is set, $\overline{\text{AD\_EN}}$ is controlled by the AD_IN voltage monitor. The switch is turned on ($\overline{\text{AD\_EN}}$ driven low) when AD_IN is valid, and turned off ($\overline{\text{AD\_EN}}$ driven high) when AD_IN is invalid.

CNC_CR[SWADS] can control the $\overline{\text{AD\_EN}}$ status as well. The switch is turned on ($\overline{\text{AD\_EN}}$ driven low) when SWADS is set, and turned off ($\overline{\text{AD\_EN}}$ driven high) when SWADS is cleared. If software enables this switch, after entering Stop mode, the switch keeps being enabled.

27.8.3 Rectifier voltage monitor

CNC integrates a rectifier voltage monitor sub-module. The rectifier of wireless power receiver provides full-wave rectification of the AC waveform from receiver coil. The sub-module is designed to monitor the rectifier voltage and to implement the overvoltage protection and the low voltage warning.

27.8.3.1 Overvoltage and low voltage detection

After CNC_CR[VRECDE] is set, the rectifier voltage monitor is enabled.

When the rectifier voltage (input from the pin VREC) rises above the overvoltage threshold, the CNC_STAS[VRECOVS] status bit is set. The VREC digital glitch filter (VRECFLT) can be enabled when CNC_VRECFLTC[CNT] is configured to any non-zero value. The VRECOVS bit is set only when the VREC overvoltage condition is valid by this filter.

When the rectifier voltage (input from the pin VREC) falls below the low voltage threshold, the CNC_STAS[VRECLVS] status bit is set. If the VREC digital glitch filter (VRECFLT) is enabled, the VRECLVS bit is set only when the VREC low voltage condition is valid by this filter.

The VREC digital glitch filter clock is the bus clock divided by 16.

27.8.3.2 Overvoltage and low voltage interrupt

If CNC_CR[OVIE] is set, it can generate an interrupt when the CNC_STAS[VRECOVF] flag is set. Writing 1 to VRECOVF can clear this flag.

If CNC_CR[LVIE] is set, it can generate an interrupt when the CNC_STAS[VRECLVF] flag is set. Writing 1 to VRECLVF can clear this flag.

27.8.3.3 Clamp driver control

The VREC monitor sub-module also integrates a clamp driver. The assert and de-assert of this output can be controlled by:

- VREC over-voltage flag
- Software
- External clamp driver from other on-chip modules

When CNC_VRECOE[VRECOVE] is set, the CLAMP is controlled by the VREC voltage monitor. The CLAMP output is asserted when the status bit CNC_STAS[VRECOVS] is set, and is de-asserted when the same bit VRECOVS is cleared.

NOTE

CNC_VRECOE[VRECOVE] is a write-once bit. If this bit is configured, it cannot be configured again until the next system reset.

CNC_CR[SWCLPD] also controls the CLAMP status. The CLAMP output is asserted when SWCLPD is set, and is de-asserted when SWCLPD is cleared.

When CNC_CR[EXTCLPDE] is set, the CLAMP output is controlled by the external clamp driver trigger from other on-chip modules.

For the external clamp driver source connection details, refer to the chip configuration information.

27.8.4 Low-power status

When CNC sub-module enable bits are cleared (except CNC_CR[ADANE]), the CNC module is in a low-power status. All sub-modules, except the wired power (AD_IN) monitor, are disabled. The AD_IN monitor sub-module can maintain its function in any of the following MCU operational modes: Run, Wait and Stop. And if enabled, the AD_IN monitor sub-module can generate interrupt when the AD_IN status has changed, either from valid to invalid or from invalid to valid, to bring MCU out of the low power modes.

27.9 Application information

Communication and Clamp Controller can be used to detect the signal on AC1 and AC2 and also be used to detect VREC, ADP_IN voltage status such as low voltage(lv) or over voltage(ov). When over voltage on VREC happens, this module will report over voltage flag to software and software will make proper action such as enable clamp function. This module also use to control the AD_IN to vout switch, when proper USB plugged in the AD_IN port, this module reports AD_IN voltage status to software and software decides whether to use AD_IN power or not.

When using AC1 and AC2 zero crossing, they are configured for the AC1 and AC2 divider and low pass filters. If the AC1 and AC2 signal is not good (for example, have big distortion), enable the filter function in case of the output of fsk glitch which may kill the fsk demodulation of other modules on chip.

When using wireless power, enable the ov/lv protection on both VREC and AD_IN. When OV happens on VREC, software may probably enable clamp and discharge most of the power from VREC till VREC OV is released. When OV happens on AD_IN, software keeps shutting off of the external pmos unless the software forces enable the external pmos switch.

When wireless power is using and a proper USB plugged, this module will report to software that the AD_IN has good power and can be used. Software decides whether to use it or not. If software decides to use AD_IN power, software needs to send end of power to transmitter and enable the two external pmos switch. Then the chip power would come from diode which connected on AD_IN.

When no wireless power is present, chip is power up by AD_IN, the two external pmos switches need to be shut off. After system startup and this module is enabled, software decides whether to enable the two external pmos switches according to the AD_IN voltage detect result.

Chapter 28
Linear Low Dropout Voltage Regulator Controller (LDO)

28.1 Chip-specific LDO information

28.1.1 LDO regulator controller interconnections

The LDO regulator output could be shut down with external trigger. The external shutdown source can be selected through the trigger crossbar (TBAR) in the SIM module, and please refer to the SIM chapter for details.

28.2 Introduction

The linear Low Dropout voltage regulator controller (LDO) module is designed to deliver power to wireless power receiver loads with the external N-FET power devices. LDO controller has voltage and current loop control, which can provide a maximum 5V/3A power supply to loading system with precise voltage and current control. This module also includes overvoltage protection (OVP) and overcurrent protection (OCP) mechanism to prevent the external power devices from damage due to the high output voltage or large output current. Two high-accuracy 9-bit DACs are included to provide the precise reference voltage for the voltage/current regulator and the OVP/OCP comparator.

28.3 Features

LDO includes the following features:

- Input voltage: 4.6 V ~ 24 V(as maximum) from AC-DC rectifier
- Configurable output voltage: 4.2 V ~ 5.2 V
- Configurable loading current: 1 A ~ 3 A(as maximum)

- Configurable overvoltage protection and overcurrent protection
- High-accuracy 9-bit DACs for output voltage and current trimming
- Configurable charge pump voltage and pump voltage monitor

28.4 Overview

This section presents an overview of the LDO module. The following figure illustrates the simplified LDO block diagram.

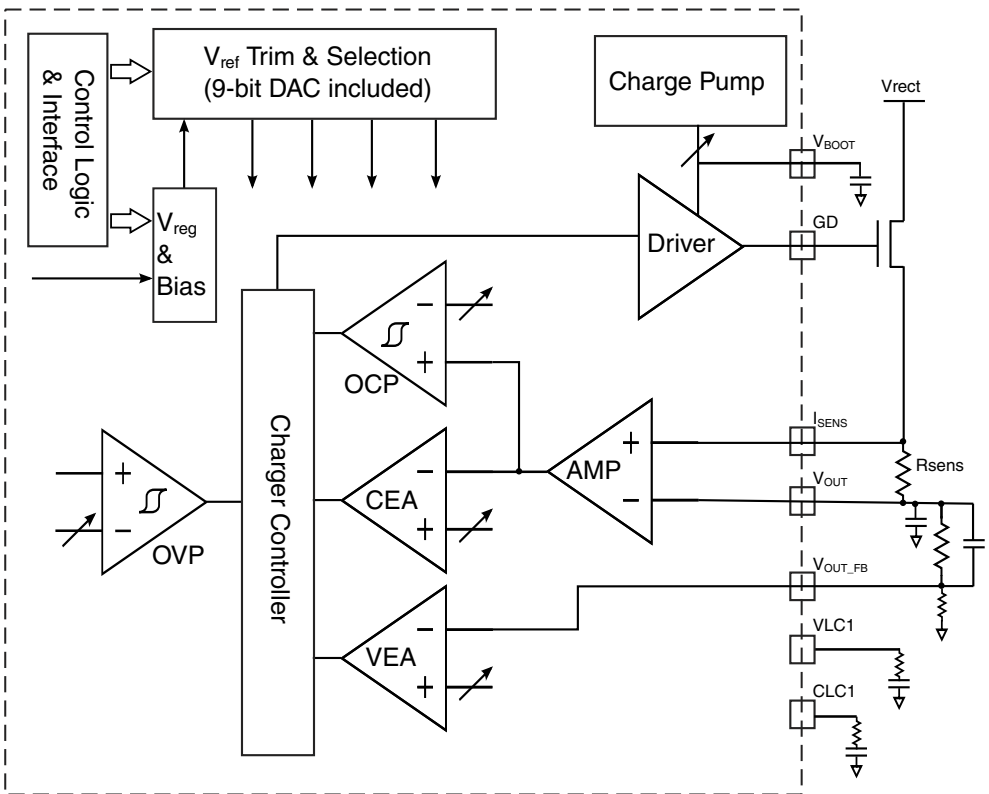


Figure 28-1. Simplified LDO block diagram

28.5 Modes of operation

LDO supports two operating modes: the Disabled mode and the Active mode.

28.5.1 Disabled mode

When the LDO module enable bits are cleared, LDO is disabled and does not perform any function in any MCU operational mode.

28.5.2 Active mode

In the Active mode, the LDO module has its full functionality. LDO can be in Active mode with the MCU in any of the following operational modes: Run and Wait.

28.6 External signal description

LDO has one input signal channel. The following table itemizes all the LDO external pins.

Signal	I/O	Detailed description	Connect from/to
VOUT	I	LDO5V3A output power	From pin VOUT
VOUT_FB	I	VOUT feedback pin	From pin VOUT_FB
ISENS	I	Current sensing signal	From pin ISENS
VBOOT	O	Charge pump power	To pin VBOOT
GD	O	Gate drive signal for the external N-FET	To pin GD
VLC	I	Voltage loop compensation	From pin VLC1
CLC1	I	Current loop compensation	From pin CLC1

28.6.1 VOUT

This pin is the LDO output power. On-board capacitor of 4.7~22 µF needs to be added on this pin. It also provide 7/10 VOUT divider output to ADC. In order to enable the output, it requires LDO_CR[LDOEN], LDO_CR[CPEN], LDO_CR[LDOREGEN] and LDO_CR[CPCLKPS] to be set correctly. After set these registers, it need wait enough settle time before use ADC to measure the VOUT divider output.

28.6.2 VOUT_FB

This pin is the VOUT feedback pin. The signal is divided from VOUT through on-board precise resistor ladder, and is directly used for voltage loop control.

28.6.3 ISENS

This pin is the current sensing signal. The output current is measured through a small sensing resistor between VOUT and ISENS pins. The resistor is 33~100 mΩ according to the maximum IOUT in application, and the voltage drop on the sensing resistor should be 100 mV at the maximum IOUT.

28.6.4 VBOOT

This pin is the charge pump power. The maximum pumped voltage is 10 V and the drive capability is 700 µA. This power cannot be shared with any external component, and it needs a 10 nF capacitor on this pin.

28.6.5 GD

This pin is the gate drive signal for external N-FET power devices. The maximum voltage is 10 V. This pin is pulled down when LDO5V3A is disabled or during overvoltage and overcurrent protections.

28.6.6 VLC

This pin is for voltage loop compensation. An external compensation network is needed to keep the voltage loop stability for varying output voltage and current.

28.6.7 CLC1

This pin is for current loop compensation. An external compensation network is needed to keep the current loop stability for varying output voltage and current.

28.7 Memory map and register definition

This section includes the module memory map and detailed descriptions of all registers.

LDO memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4007_0000	Control Register (LDO_CR)	16	R/W	0010h	28.7.1/415
4007_0002	Status Register (LDO_SR)	16	R/W	0000h	28.7.2/417
4007_0004	Current Trim Register (LDO_CTRM)	16	R/W	0000h	28.7.3/418
4007_0006	Voltage Trim Register (LDO_VTRM)	16	R/W	0000h	28.7.4/418
4007_0008	Overcurrent Filter Register (LDO_OCFILT)	16	R/W	0000h	28.7.5/419
4007_000A	Overvoltage Filter Register (LDO_OVFILT)	16	R/W	0000h	28.7.6/419
4007_000C	Shutdown Control Register (LDO_SCR)	16	R/W	0007h	28.7.7/420

28.7.1 Control Register (LDO_CR)

Address: 4007_0000h base + 0h offset = 4007_0000h

Bit	15	14	13	12	11	10	9	8
Read Write	LDOEN	CPEN	LDOREGEN	OCTHLD			BASHDN	OCDTE
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Read Write	OVTHLD		0	CPCLKPS		OCIE	OVIE	OVDTE
Reset	0	0	0	1	0	0	0	0

LDO_CR field descriptions

Field	Description
15 LDOEN	LDO controller Enable This bit is used to enable the LDO controller. In Stop mode the LDO controller is disabled automatically. 0 LDO controller is disabled. 1 LDO controller is enabled.
14 CPEN	Charge Pump Enable This bit is used to enable the charge pump function. In Stop mode the charge pump is disabled automatically. 0 Charge pump is disabled. 1 Charge pump is enabled.
13 LDOREGEN	LDO internal Regulator Enable This bit is used to enable the LDO controller internal voltage reference. **NOTE:** Set this bit and CPEN to 1, and then wait for LDO_SR[CPOKF] to get asserted, before setting the bit LDOEN. Clear it to 0 at the same time with LDOEN and CPEN. 0 LDO regulator is disabled. 1 LDO regulator is enabled.

Table continues on the next page...

LDO_CR field descriptions (continued)

Field	Description
12–10 OCTHLD	Overcurrent Threshold Selection Selects the threshold level for overcurrent detection. See LDO overcurrent detection for detailed threshold level configuration. 000 Level 0 001 Level 1 010 Level 2 011 Level 3 100 Level 4 101 Level 5 110 Level 6 111 Level 7
9 BASHDN	Block the Automatic Shutdown This bit is used to block the function of automatic shutdown in internal hard-block. It blocks the automatically shutting down the power FET when OVP event occurs. 0 Allows the automatically shutting down the power FET when OVP event occurs. 1 Blocks the automatic shutdown function from hard-block part when OVP event occurs. Software takes it over to shut down the external power FET.
8 OCDTE	Overcurrent Detection Enable **NOTE:** Clear this bit to 0 when the LDO controller is disabled, to prevent some potential leakage. 0 Overcurrent detection is disabled. 1 Overcurrent detection is enabled.
7–6 OVTHLD	Overvoltage Threshold Selection Selects the threshold level for overvoltage detection. See LDO overvoltage detection for detailed threshold level configuration. 00 Level 0 01 Level 1 10 Level 2 11 Level 3
5 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
4–3 CPCLKPS	Prescaler bits of the Clock to be divided for Charge Pump The clock source to this divider is the system clock that runs at a stable frequency. The divided clock frequency to the charge is determined by $F/2^{(CPCLKPS+2)}$, where F is the clock source frequency. 00 The division factor is 4. 01 The division factor is 8. 10 The division factor is 16. 11 The division factor is 32.
2 OCIE	Overcurrent Interrupt Enable

Table continues on the next page...

LDO_CR field descriptions (continued)

Field	Description
	This bit is used to enable the interrupt when the current loop output is over the current limit. 0 No interrupt gets asserted when the current loop output is over the current limit. 1 An interrupt gets asserted when the current loop output is over the current limit.
1 OVIE	Overvoltage Interrupt Enable This bit is used to enable the interrupt when the voltage loop output is over the voltage limit. 0 No interrupt gets asserted when the voltage loop output is over the voltage limit. 1 An interrupt gets asserted when the voltage loop output is over the voltage limit.
0 OVDTE	Overvoltage Detection Enable **NOTE:** Clear this bit to 0 when the LDO controller is disabled, to prevent some potential leakage. 0 Overvoltage detection is disabled. 1 Overvoltage detection is enabled.

28.7.2 Status Register (LDO_SR)

Address: 4007_0000h base + 2h offset = 4007_0002h

Bit	15	14	13	12	11	10	9	8
Read	CPOKF	0						
Write								
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Read	0				OCST	OVST	OCF	OVF
Write							w1c	w1c
Reset	0	0	0	0	0	0	0	0

LDO_SR field descriptions

Field	Description
15 CPOKF	Charge Pump OK status Flag This bit follows the internal charge pump OK status. 0 Indicates the internal charge pump is not in OK status. 1 Indicates the internal charge pump is OK.
14–4 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
3 OCST	Overcurrent Status This bit indicates the OC status.

Table continues on the next page...

LDO_SR field descriptions (continued)

Field	Description
2 OVST	Overvoltage Status This bit indicates the OV status.
1 OCF	Current loop Overcurrent Flag Writing 1 to this bit clears the flag to 0, and also clears all the internal states of the OCF filter. The filter clock source to this flag generation is divided by 16 by default. 0 The current loop output is not over the current limit. 1 The current loop output is over the current limit.
0 OVF	Voltage loop Overvoltage Flag Writing 1 to this bit clears the flag to 0, and also clears all the internal states of the OVF filter. 0 The voltage loop output is not over the voltage limit. 1 The voltage loop output is over the voltage limit.

28.7.3 Current Trim Register (LDO_CTRM)

Address: 4007_0000h base + 4h offset = 4007_0004h

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	0								CTRM							
Write																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

LDO_CTRM field descriptions

Field	Description
15–9 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
CTRM	Current loop configuration **NOTE:** It is requested to use the half-word or word write to this register, so the 9-bit trim value gets updated at the same time.

28.7.4 Voltage Trim Register (LDO_VTRM)

Address: 4007_0000h base + 6h offset = 4007_0006h

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	0								VTRM							
Write																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

LDO_VTRM field descriptions

Field	Description
15–9 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
VTRM	Voltage loop configuration **NOTE:** It is requested to use the half-word or word write to this register, so the 9-bit trim value gets updated at the same time.

28.7.5 Overcurrent Filter Register (LDO_OCFILT)

Address: 4007_0000h base + 8h offset = 4007_0008h

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read Write	CNT															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

LDO_OCFILT field descriptions

Field	Description
CNT	Filter sample count for the overcurrent detection The value represents the number of consecutive samples that must agree, prior to that the filter accepts a new overcurrent state. 0 Filter is disabled. Overcurrent status is passing directly. None-zero Any other none-zero value specifies the CNT number of consecutive samples that must agree.

28.7.6 Overvoltage Filter Register (LDO_OVFILT)

Address: 4007_0000h base + Ah offset = 4007_000Ah

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read Write	CNT															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

LDO_OVFILT field descriptions

Field	Description
CNT	Filter sample count for the overvoltage detection The value represents the number of consecutive samples that must agree, prior to that the filter accepts a new overvoltage state.

LDO_OVFILT field descriptions (continued)

Field	Description
	0 Filter is disabled. Overcurrent status is passing directly. None-zero Any other none-zero value specifies the CNT number of consecutive samples that must agree.

28.7.7 Shutdown Control Register (LDO_SCR)

Address: 4007_0000h base + Ch offset = 4007_000Ch

Bit	15	14	13	12	11	10	9	8
Read	0							
Write								
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Read	0					EXTSDE	OVASDE	OCASDE
Write								
Reset	0	0	0	0	0	1	1	1

LDO_SCR field descriptions

Field	Description
15–3 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
2 EXTSDE	External Shutdown Enable This bit is used to enable the external shutdown request to turn off the whole LDO system. It is write-once after reset, and when already cleared to 0 only the system reset can set this bit value to 1. 0 External shutdown request is disabled to turn off the LDO system. 1 External shutdown request is enabled to turn off the LDO system.
1 OVASDE	LDO voltage loop automatic shut-down enables when overvoltage happens. It is write-once after reset, and when already cleared to 0 only the system reset can set this bit value to 1. 0 Voltage loop and current loop are not automatically shut down when the overvoltage happens. 1 Voltage loop and current loop are automatically shut down when the overvoltage happens.
0 OCASDE	LDO voltage loop automatic shut-down enables when overcurrent happens. It is write-once after reset, and when already cleared to 0 only the system reset can set this bit value to 1. 0 Voltage loop and current loop are not automatically shut down when the overcurrent happens. 1 Voltage loop and current loop are automatically shut down when the overcurrent happens.

28.8 Functional description

The following sections describe functional details of the module.

28.8.1 N-FET gate driver and internal charge pump

The LDO regulator is designed to support 5 V output voltage with an external N-FET. The N-FET gate voltage is the output from the pin GD.

To generate GD higher than the MCU power supply voltage, the LDO module integrates a charge pump which provides pumped voltage doubling the MCU supply voltage.

The charge pump needs an external capacitor of 10 nF, connected to pin V_{BOOT} to reduce the ripple voltage.

To enable the charge pump, set LDO_CR[CPEN] to 1.

The charge pump clock frequency can be configured through CPCLKPS [1:0] bits in the LDO_CR register. For example, assuming the input clock source frequency is 48 MHz, the charge pump clock frequency is calculated as follows:

- CPCLKPS = 00, charge pump clock = 12 MHz
- CPCLKPS = 01, charge pump clock = 6 MHz
- CPCLKPS = 10, charge pump clock = 3 MHz
- CPCLKPS = 11, charge pump clock = 1.5 MHz

28.8.2 LDO output voltage control

The LDO regulator output voltage is controlled by the voltage control loop.

After setting LDO_CR[LDOEN] to 1 and the V_{BOOT_OK} is effective, the LDO voltage control loop is enabled.

The LDO output voltage V_{OUT} is controlled by the pin V_{OUT_FB} which senses the V_{OUT} voltage by a resistor divider and the internal voltage reference in.

28.8.2.1 Voltage error amplifier

This error amplifier monitors and amplifies the voltage error between V_{OUT} feedback (V_{OUT_FB}) and a reference voltage (V_{REF_REG}). The error of these 2 signals is amplified and is used to control the gate voltage of external NMOS power devices to dynamically adjust the output voltage, according to different loads.

The reference voltage V_{REF_REG}, for this amplifier, is generated by a 9-bit configurable DAC.

By configuring the VTRM[8:0] bits in LDO_VTRM register, user can set the V_{REF_REG} level and thus set the V_{OUT} level.

Supposing V_{OUT_FB} = 1/5 V_{OUT}, refer to the following table for the VTRM bit value and its corresponding V_{REF_REG} and V_{OUT} target level.

NOTE

User should set a proper VTRM value to let V_{OUT} fall in the range 4.2 ~ 5 V.

Table 28-9. LDO VTRM and V_{OUT}

VTRM	V_{REF_REG}	Target V_{OUT_FB}	Target V_{OUT}
0 0000 0000	0 V	0 V	0 V
…	…	…	…
N	N × 1.4 V / 511	N × 1.4 V / 511	N × 7 V / 511
…	…	…	…
1 0110 1101 (max)	1 V (max)	1 V (max)	5 V (max)

28.8.2.2 Voltage loop compensation

VLC pin is used for voltage loop compensation. An external compensation network is needed to keep the voltage loop stability for varying output voltage and current.

28.8.3 LDO output current control

The LDO regulator output current can be controlled by the current control loop. With the LDO output voltage given by the voltage control loop, the current control loop can serve as the output current limiter.

After setting LDO_CR[LDOEN] to 1, the LDO current control loop is enabled.

NOTE

The charge pump should be enabled before the LDO module. User should first set the CPEN bit and the LDOREGEN bit, wait the LDO_SR[CPOKF] flag to be set, and then set the LDOEN bit to enable the LDO module.

The output current is measured through a small external sensing resistor between V_{OUT} and I_{SENS} pins.

28.8.3.1 Current sensing amplifier and current error amplifier

The output current is sensed by the small voltage drop (10~100 mV) over the external sensing resistor. This small voltage is amplified by a current sensing amplifier whose gain is fixed to 10.

The error amplifier monitors and amplifies the voltage error between this current sensing amplifier output (V_{CSA_OUT}) and a reference voltage (V_{REF_ILOOP}) . The error of these 2 signals is amplified and is used to control the gate voltage of external NMOS power devices to dynamically adjust the output current.

The current error amplifier reference V_{REF_ILOOP}, is generated by a 9-bit configurable DAC.

By configuring the CTRM[8:0] bit in LDO_CTRM register, user can set the V_{REF_ILOOP} level and thus set the I_{OUT} level.

Supposing that a 0.1 Ω external sensing resistor is used, refer to the following table for the CTRM bit value and its corresponding V_{REF_ILOOP} and I_{OUT} target level.

Table 28-10. LDO CTRM and I_{OUT}

CTRM	V_{REF_ILOOP}	Target V_{Rsens}	Target I_{OUT}
0 0000 0000	0 V	0 V	0 A
…	…	…	…
N	N × 1.4 V / 511	N × 0.14 V / 511	N × 1.4 A / 511
…	…	…	…
1 1111 11111	1.4 V	0.14 V	1.4 A

28.8.3.2 Current loop compensation

CLC1 pin is used for current loop compensation. An external compensation network is needed to keep the voltage loop stability for varying output voltage and current.

28.8.4 LDO overvoltage and overcurrent protection

The LDO module integrates an overvoltage and overcurrent protection block.

When the output voltage is controlled by the voltage control loop, it is also monitored by the overvoltage protection block (OVP). Likewise, the output current is also monitored by the overcurrent protection block (OCP).

28.8.4.1 LDO overvoltage detection

The OVP monitors the V_{OUT} by sensing the internal resistor divider of V_{OUT}.

When LDO_CR[OVDTE] is set to 1, the LDO output overvoltage detection is enabled.

There is an internal divider sensing the V_{OUT} voltage, and then compares the internal voltage reference. The calculation for the threshold of OVP is as the following formula:

$V_{TH1} = 40/9 \times V_{REF\_OVP}$,
$V_{TH2} = 5 \times V_{REF\_OVP}$.

To configure V_{REF_OVP}, user should write the OVTHLD [1:0] bits in LDO_CR register.

- OVTHLD = 00, V_{REF_OVP} = 0.96 V
- OVTHLD = 01, V_{REF_OVP} = 1.04 V
- OVTHLD = 10, V_{REF_OVP} = 1.20 V
- OVTHLD = 11, V_{REF_OVP} = 1.30 V

When V_{OUT} is higher than the OVP threshold, the flag LDO_SR[OVF] is set. Writing 1 to OVF can clear this flag.

28.8.4.2 LDO overcurrent detection

Like the current control loop, the OCP monitors the output current by the same external sensing resistor and the same current sensing amplifier.

When LDO_CR[OCDTE] is set to 1, the LDO output overcurrent detection is enabled.

The current sensing amplifier output (V_{CSA_OUT}) is compared with a reference voltage (V_{REF_OCP}). With a given external resistor between I_{SENS} and V_{OUT}, user can set the overcurrent threshold by configuring this reference voltage.

The calculation for the OCP threshold is as the following formula:

$I_{TH1} = V_{REF\_OCP} / (10 \times R_{sens})$.

To configure V_{REF_OCP}, user should write the OCTHLD[2:0] bits in LDO_CR register.

- OCTHLD = 000, V_{REF_OCP} = 1.05 V
- OCTHLD = 001, V_{REF_OCP} = 1.10 V
- OCTHLD = 010, V_{REF_OCP} = 1.15 V

- OCTHLD = 011, V_{REF_OCP} = 1.20 V
- OCTHLD = 100, V_{REF_OCP} = 1.25 V
- OCTHLD = 101, V_{REF_OCP} = 1.30 V
- OCTHLD = 110, V_{REF_OCP} = 1.35 V
- OCTHLD = 111, V_{REF_OCP} = 1.40 V

Suppose that a 0.1 Ω external sensing resistor is used for 1 A maximum output current. The maximum output then causes 100 mV voltage drop on the R_{sens}, and the current sensing amplifier output (V_{CSA_OUT}) is 1 V. When OCTHLD is configured to 001, the output overcurrent threshold is 1.1 A.

When I_{OUT} is higher than the OCP threshold, the flag LDO_SR[OCF] is set. Writing 1 to OCF can clear this flag.

28.8.4.3 Digital filter for OVP and OCP flag

To prevent the flag OVF or OCF being set frequently due to the noise, LDO integrates an OVP glitch filter and an OCP glitch filter.

The OVP digital glitch filter can be enabled when the CNT[15:0] bits in LDO_OVFILT register is configured to any non-zero value. The OVF flag is set only when the V_{OUT} overvoltage condition is valid by this filter.

The bus clock is used as the OVP filter clock.

When LDO_OVFILT[CNT] bits are configured to a non-zero value, the OVF flag is valid and is set only if V_{OUT} remains higher than the OVP threshold for consequent CNT clock cycles.

Similarly, the OCP digital glitch filter can be enabled when the CNT[15:0] bits in LDO_OCFILT register is configured to any non-zero value. The OCF flag is set only when the I_{OUT} overcurrent condition is valid by this filter.

The bus clock divided by 16 is used as the OCP filter clock.

When LDO_OCFILT[CNT] bits are configured to a non-zero value, the OCF flag is valid and is set only if I_{OUT} remains higher than the OCP threshold for consequent CNT clock cycles.

28.8.4.4 LDO overvoltage and overcurrent interrupt

If LDO_CR[OVIE] is set, when the flag LDO_STAS[OVF] is set, it can generate an interrupt. Writing 1 to OVF can clear this flag.

If LDO_CR[OCIE] is set, when the flag LDO_STAS[OCF] is set, it can generate an interrupt. Writing 1 to OCF can clear this flag.

28.8.4.5 LDO shutdown control

The LDO regulator output can be shut down when fault condition like overvoltage or overcurrent happens.

If LDO_SCR[OVASDE] is set to 1, and when the flag LDO_STAS[OVF] is set due to output overvoltage, the LDO regulator is shut down immediately.

If LDO_SCR[OCASDE] is set to 1, and when the flag LDO_STAS[OCF] is set due to output overcurrent, the LDO regulator is shut down immediately.

If LDO_SCR[EXTSDE] is set to 1, LDO can be shut down by external trigger from other on-chip modules. Refer to TBAR to check the external shutdown trigger source connection details.

NOTE

LDO_SCR[OVASDE], LDO_SCR[OCASDE] and LDO_SCR[EXTSDE] are write-once register bits. Once they are configured, they cannot be configured again until the next system reset.

28.9 Application information

28.9.1 LDO soft start

The module provides two methods for soft start.

- Use the voltage loop to get the soft start:

  configure the VTRM step by step, from 0 to V_{set}, with a total ramp up time as 1 ms.

- Use the current limit loop to get the soft start:

  configure according to the formula: $I_{set\_softstart} = V_{config} / (10 \times R_{sens})$, meanwhile set the target voltage through VTRM.

Chapter 29
Periodic Interrupt Timer (PIT)

29.1 Chip-specific PIT information

29.1.1 PIT interconnections

The PIT channel 0 and channel 1 trigger output can be used as trigger source for other on chip modules throught the TBAR.

29.2 Introduction

The PIT module is an array of timers that can be used to raise interrupts and triggers.

29.2.1 Block diagram

The following figure shows the block diagram of the PIT module.

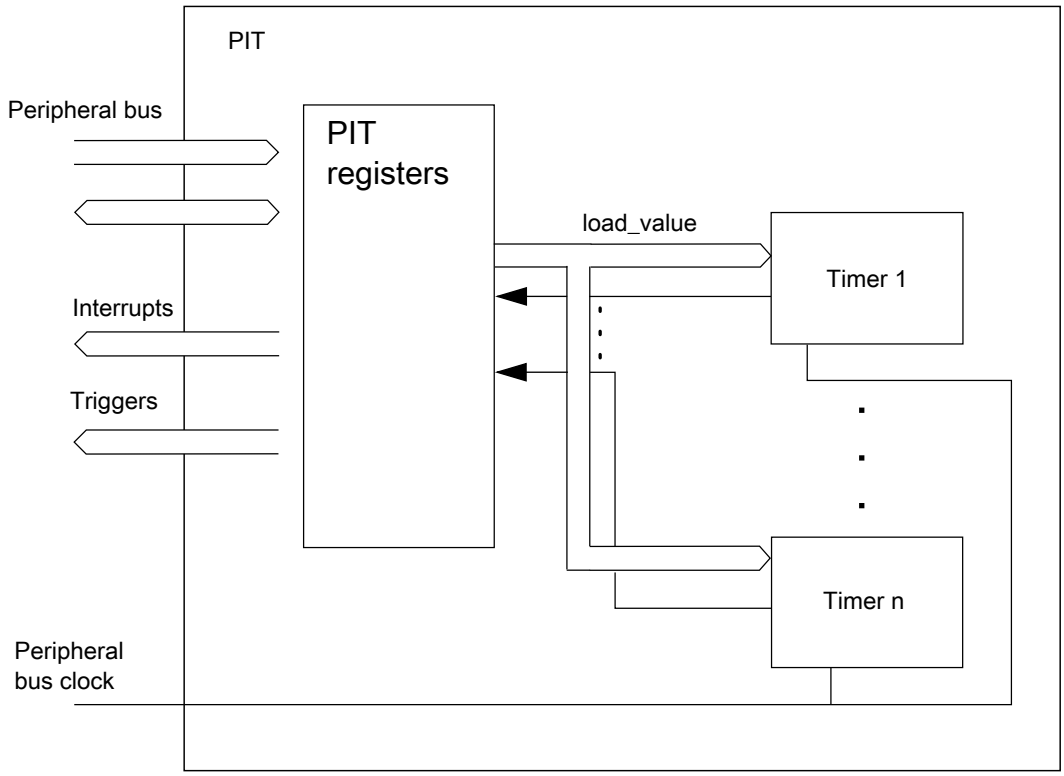


Figure 29-1. Block diagram of the PIT

NOTE

See the chip-specific PIT information for the number of PIT channels used in this MCU.

29.2.2 Features

The main features of this block are:

- Ability of timers to generate trigger pulses
- Ability of timers to generate interrupts
- Maskable interrupts
- Independent timeout periods for each timer

29.3 Signal description

The PIT module has no external pins.

29.4 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

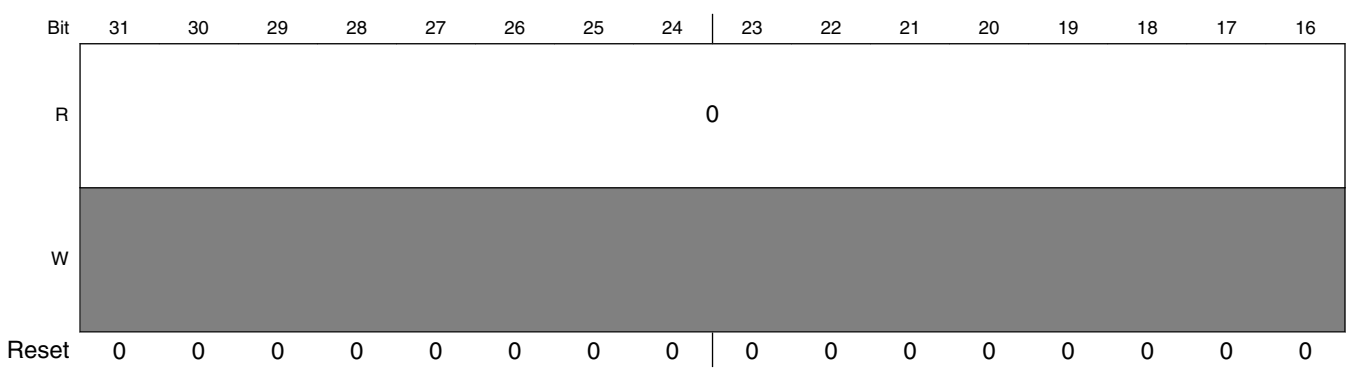
Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4003_7000	PIT Module Control Register (PIT_MCR)	32	R/W	0000_0002h	29.4.1/429
4003_7100	Timer Load Value Register (PIT_LDVAL0)	32	R/W	0000_0000h	29.4.2/430
4003_7104	Current Timer Value Register (PIT_CVAL0)	32	R	0000_0000h	29.4.3/431
4003_7108	Timer Control Register (PIT_TCTRL0)	32	R/W	0000_0000h	29.4.4/431
4003_710C	Timer Flag Register (PIT_TFLG0)	32	R/W	0000_0000h	29.4.5/432
4003_7110	Timer Load Value Register (PIT_LDVAL1)	32	R/W	0000_0000h	29.4.2/430
4003_7114	Current Timer Value Register (PIT_CVAL1)	32	R	0000_0000h	29.4.3/431
4003_7118	Timer Control Register (PIT_TCTRL1)	32	R/W	0000_0000h	29.4.4/431
4003_711C	Timer Flag Register (PIT_TFLG1)	32	R/W	0000_0000h	29.4.5/432

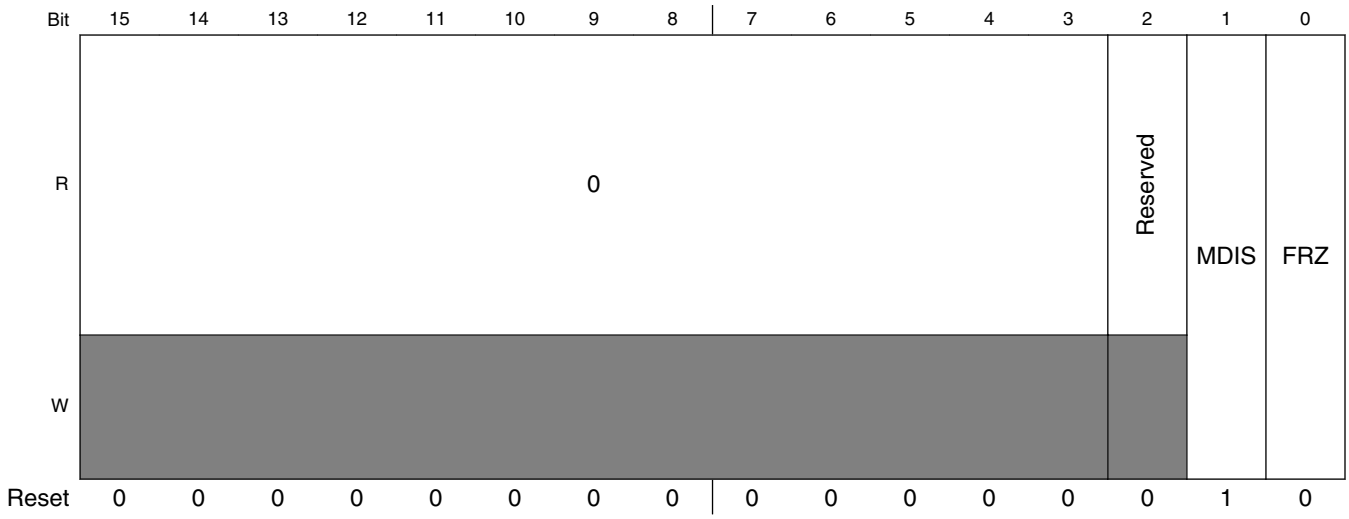
29.4.1 PIT Module Control Register (PIT_MCR)

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





PIT_MCR field descriptions

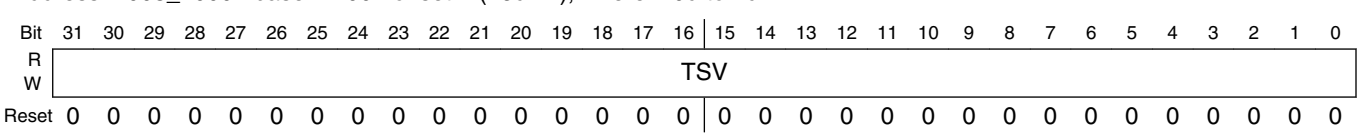
Field	Description
31–3 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
2 Reserved	This field is reserved.
1 MDIS	Module Disable - (PIT section) Disables the standard timers. This field must be enabled before any other setup is done. 0 Clock for standard PIT 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.

29.4.2 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 1d



PIT_LDVAL*n* field descriptions

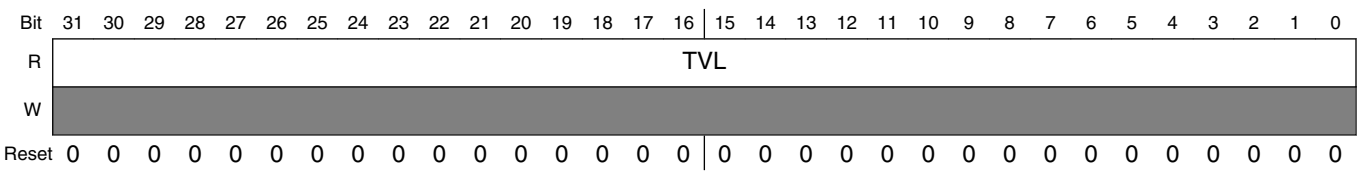
Field	Description
TSV	Timer Start Value Sets the timer start value. The timer will count down until it reaches 0, then it will generate an interrupt and load this register value again. Writing a new value to this register will not restart the timer; instead the value will be loaded after the timer expires. To abort the current cycle and start a timer period with the new value, the timer must be disabled and enabled again.

29.4.3 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 1d



PIT_CVAL*n* field descriptions

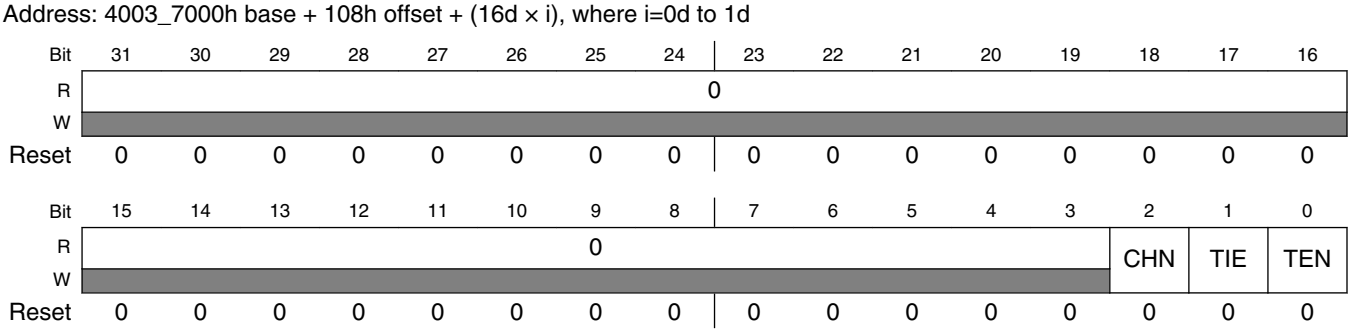
Field	Description
TVL	Current Timer Value Represents the current timer value, if the timer is enabled. **NOTE:** • If the timer is disabled, do not use this field as its value is unreliable. • The timer uses a downcounter. The timer values are frozen in Debug mode if MCR[FRZ] is set.

29.4.4 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 1d



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.

29.4.5 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 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															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.

Table continues on the next page...

PIT_TFLG*n* field descriptions (continued)

Field	Description
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.

29.5 Functional description

This section provides the functional description of the module.

29.5.1 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.

29.5.1.1 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.

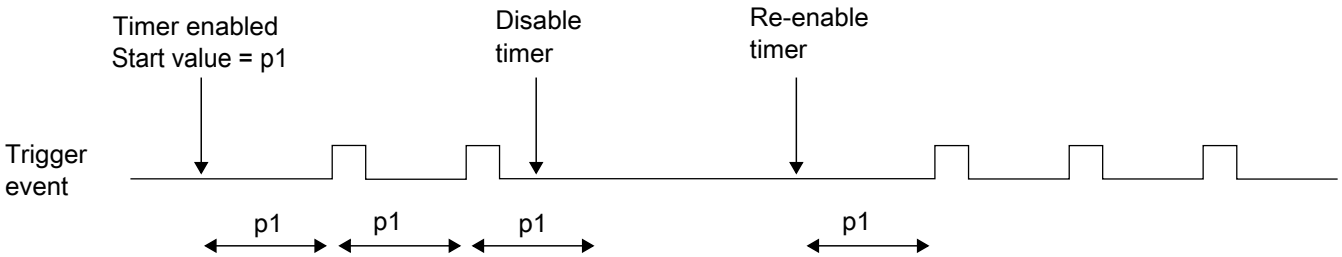


Figure 29-15. Stopping and starting a timer

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.

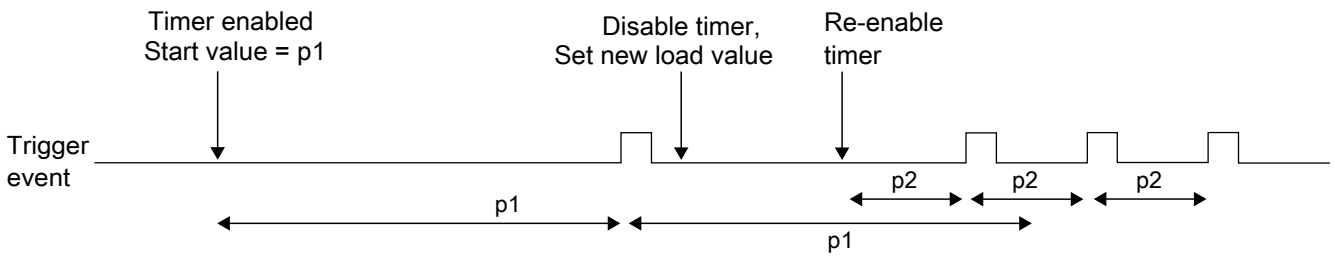


Figure 29-16. Modifying running timer period

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.

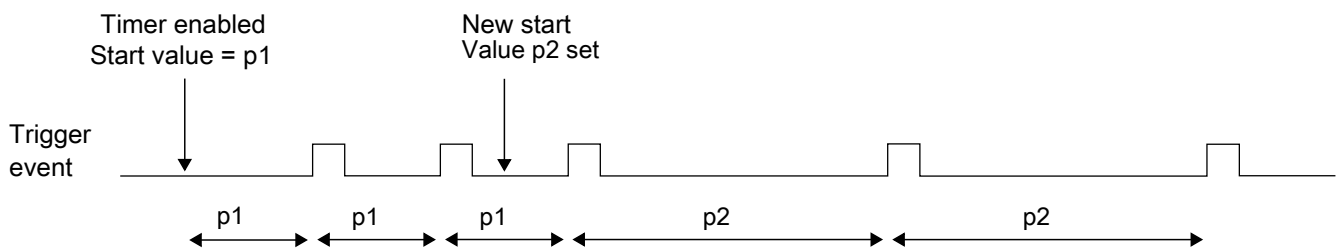


Figure 29-17. Dynamically setting a new load value

29.5.1.2 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.

29.5.2 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].

29.5.3 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.

29.6 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
```

29.7 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 30
Real-Time Counter (RTC)

30.1 Chip-specific RTC information

30.1.1 RTC interconnections

Four software selectable clock sources are available for input to prescaler with selectable binary-based and decimal-based divider values

- 20 kHz internal low-power oscillator (LPOCLK)
- External clock (OSCERCLK)
- 32 kHz internal reference clock (ICSIRCLK)
- Bus clock

The RTC overflow flag can also serve as trigger source for other on chip modules through TBAR.

30.2 Introduction

The real-time counter (RTC) consists of one 16-bit counter, one 16-bit comparator, several binary-based and decimal-based prescaler dividers, three clock sources, one programmable periodic interrupt, and one programmable external toggle pulse output. This module can be used for time-of-day, calendar or any task scheduling functions. It can also serve as a cyclic wake-up from low-power modes, Stop and Wait without the need of external components.

30.3 Features

Features of the RTC module include:

- 16-bit up-counter

- 16-bit modulo match limit
- Software controllable periodic interrupt on match

- Software selectable clock sources for input to prescaler with programmable 16 bit prescaler
  - OSC 32.768KHz nominal.
  - LPO (~20 kHz)
  - Bus clock
  - Internal reference clock (32 kHz)

30.3.1 Modes of operation

This section defines the RTC operation in Stop, Wait, and Background Debug modes.

30.3.1.1 Wait mode

The RTC continues to run in Wait mode if enabled before executing the WAIT instruction. Therefore, the RTC can be used to bring the MCU out of Wait mode if the real-time interrupt is enabled. For lowest possible current consumption, the RTC must be stopped by software if not needed as an interrupt source during Wait mode.

30.3.1.2 Stop modes

The RTC continues to run in Stop mode if the RTC is enabled before executing the STOP instruction. Therefore, the RTC can be used to bring the MCU out of stop modes with no external components, if the real-time interrupt is enabled.

30.3.2 Block diagram

The block diagram for the RTC module is shown in the following figure.

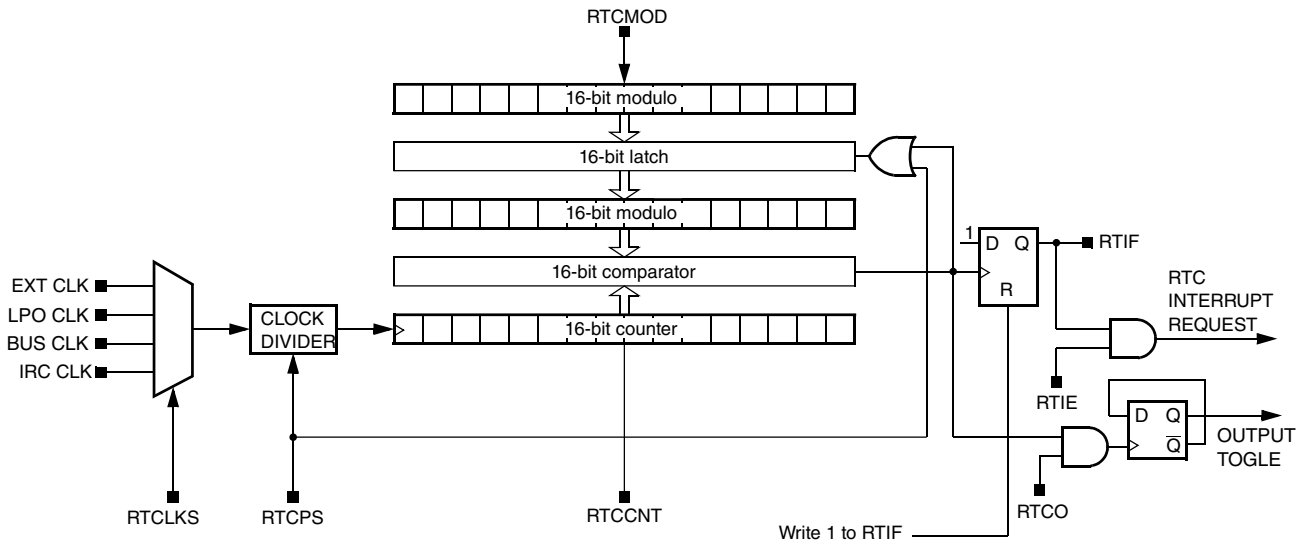


Figure 30-1. Real-time counter (RTC) block diagram

30.4 External signal description

RTCO is the output of RTC. After MCU reset, the RTC_SC[RTCO] is set to high. When the counter overflows, the output is toggled.

30.5 Register definition

The RTC includes a status and control register, a 16-bit counter register, and a 16-bit modulo register.

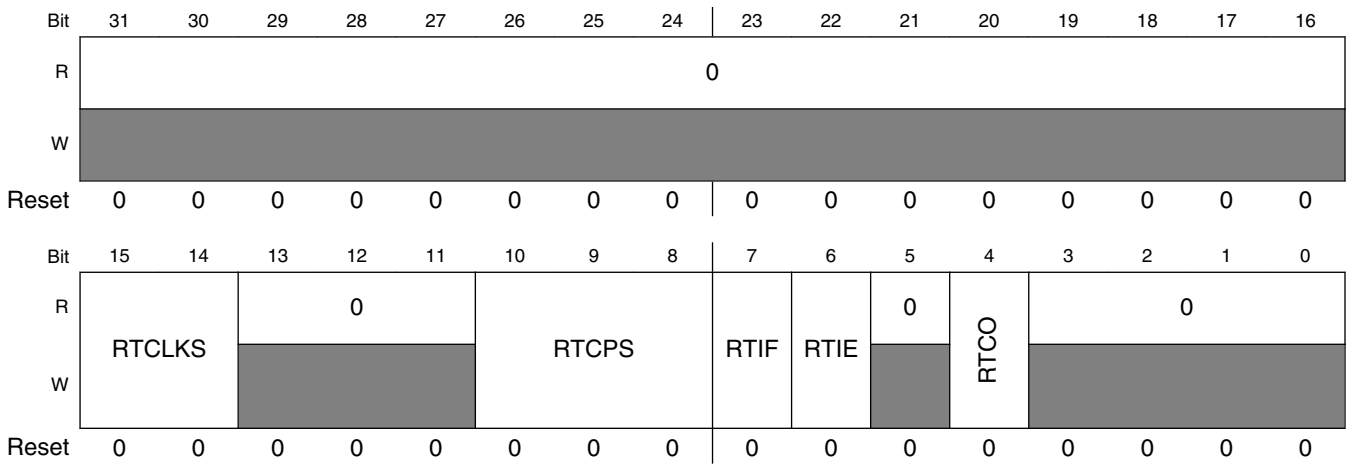
RTC memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4003_D000	RTC Status and Control Register (RTC_SC)	32	R/W	0000_0000h	30.5.1/442
4003_D004	RTC Modulo Register (RTC_MOD)	32	R/W	0000_0000h	30.5.2/443
4003_D008	RTC Counter Register (RTC_CNT)	32	R	0000_0000h	30.5.3/444

30.5.1 RTC Status and Control Register (RTC_SC)

RTC_SC contains the real-time interrupt status flag (RTIF), and the toggle output enable bit (RTCO).

Address: 4003_D000h base + 0h offset = 4003_D000h



RTC_SC field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
15–14 RTCLKS	Real-Time Clock Source Select This read/write field selects the clock source input to the RTC prescaler. Changing the clock source clears the prescaler and RTCCNT counters. Reset clears RTCLKS to 00. 00 External clock source. 01 Real-time clock source is 20 kHz (LPOCLK). 10 Internal reference clock (ICSIRCLK). 11 Bus clock.
13–11 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
10–8 RTCPS	Real-Time Clock Prescaler Select This read/write field selects binary-based or decimal-based divide-by values for the clock source. Changing the prescaler value clears the prescaler and RTCCNT counters. Reset clears RTCPS to 000. 000 Off 001 If RTCLKS = x0, it is 1; if RTCLKS = x1, it is 128. 010 If RTCLKS = x0, it is 2; if RTCLKS = x1, it is 256. 011 If RTCLKS = x0, it is 4; if RTCLKS = x1, it is 512. 100 If RTCLKS = x0, it is 8; if RTCLKS = x1, it is 1024. 101 If RTCLKS = x0, it is 16; if RTCLKS = x1, it is 2048.

Table continues on the next page...

RTC_SC field descriptions (continued)

Field	Description
	110 If RTCLKS = x0, it is 32; if RTCLKS = x1, it is 100. 111 If RTCLKS = x0, it is 64; if RTCLKS = x1, it is 1000.
7 RTIF	Real-Time Interrupt Flag This status bit indicates the RTC counter register reached the value in the RTC modulo register. Writing a logic 0 has no effect. Writing a logic 1 clears the bit and the real-time interrupt request. Reset clears RTIF to 0. 0 RTC counter has not reached the value in the RTC modulo register. 1 RTC counter has reached the value in the RTC modulo register.
6 RTIE	Real-Time Interrupt Enable This read/write bit enables real-time interrupts. If RTIE is set, then an interrupt is generated when RTIF is set. Reset clears RTIE to 0. 0 Real-time interrupt requests are disabled. Use software polling. 1 Real-time interrupt requests are enabled.
5 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
4 RTCO	Real-Time Counter Output The read/write bit enables real-time to toggle output on pinout. If this bit is set, the RTCO pinout will be toggled when RTC counter overflows. 0 Real-time counter output disabled. 1 Real-time counter output enabled.
Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

30.5.2 RTC Modulo Register (RTC_MOD)

RTC_MOD indicates the value of the 16-bit modulo value.

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																MOD															
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_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	RTC Modulo

Table continues on the next page...

RTC_MOD field descriptions (continued)

Field	Description
	This read/write field contains the modulo value used to reset the count to 0x0000 upon a compare match and set SC[RTIF] status field. A value of 0x0000 sets SC[RTIF] on each rising-edge of the prescaler output. Reset sets the modulo to 0x0000.

30.5.3 RTC Counter Register (RTC_CNT)

RTC_CNT indicates the read-only value of the current RTC count of the 16-bit counter.

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	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

RTC_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	RTC Count This read-only field contains the current value of the 16-bit counter, read CNT[7:0] first and, then CNT[15:8]. Writes have no effect to this register. Reset or writing different values to SC[RTCLKS] and SC[RTCPS] clear the count to 0x0000.

30.6 Functional description

The RTC is composed of a main 16-bit up-counter with a 16-bit modulo register, a clock source selector, and a prescaler block with binary-based and decimal-based selectable values. The module also contains software selectable interrupt logic and toggle logic for pinout.

After any MCU reset, the counter is stopped and reset to 0x0000, the modulus register is set to 0x0000, and the prescaler is off. The external oscillator clock is selected as the default clock source. To start the prescaler, write any value other than 0 to the Prescaler Select field (RTC_SC[RTCPS]).

The clock sources are software selectable: the external oscillator (OSC), on-chip low power oscillator (LPO), 32-kHz internal reference clock, and bus clock. The RTC Clock Select field (RTC_SC[RTCLKS]) is used to select the desired clock source to the prescaler dividers. If a different value is written to RTC_SC[RTCLKS], the prescaler and CNT counters are reset to 0x00.

RTC_SC[RTCPS] and RTC_SC[RTCLKS] select the desired divide-by value. If a different value is written to RTC_SC[RTCPS], the prescaler and RTCCNT counters are reset to 0x00. The following table shows different prescaler period values.

Table 30-5. Prescaler period

RTCPS	32768Hz OSC clock source prescaler period (RTCLKS = 00)	LPO clock (20 kHz) source prescaler period (RTCLKS = 01)	Internal reference clock (32.768 kHz) source prescaler period (RTCLKS = 10)	Bus clock (8 MHz) source prescaler period (RTCLKS = 11)
000	Off	Off	Off	Off
001	30.5176 µs	6.4 ms	30.5176 µs	16 µs
010	61.0351 µs	12.8 ms	61.0351 µs	32 µs
011	122.0703 µs	25.6 ms	122.0703 µs	64 µs
100	244.1406 µs	51.2 ms	244.1406 µs	128 µs
101	488.28125 µs	102.4 ms	488.28125 µs	256 µs
110	976.5625 µs	5 ms	976.5625 µs	12.5 µs
111	1.9531 ms	50 ms	1.9531 ms	125 µs

The RTC Modulo register (RTC_MOD) allows the compare value to be set to any value from 0x0000 to 0xFFFF. When the counter is active, the counter increments at the selected rate until the count matches the modulo value. When these values match, the counter resets to 0x0000 and continues counting. The Real-Time Interrupt Flag (RTC_SC[RTIF]) is set whenever a match occurs. The flag sets on the transition from the modulo value to 0x0000. The modulo value written to RTC_MOD is latched until the RTC counter overflows or RTC_SC[RTCPS] is selected nonzero.

The RTC allows for an interrupt to be generated whenever RTC_SC[RTIF] is set. To enable the real-time interrupt, set the Real-Time Interrupt Enable field (RTC_SC[RTIE]). RTC_SC[RTIF] is cleared by writing a 1 to RTC_SC[RTIF].

The RTC also allows an output to external pinout by toggling the level. RTC_SC[RTCO] must be set to enable toggling external pinout. The level depends on the previous state of the pinout when the counter overflows if this function is active.

30.6.1 RTC operation example

This section shows an example of the RTC operation as the counter reaches a matching value from the modulo register.

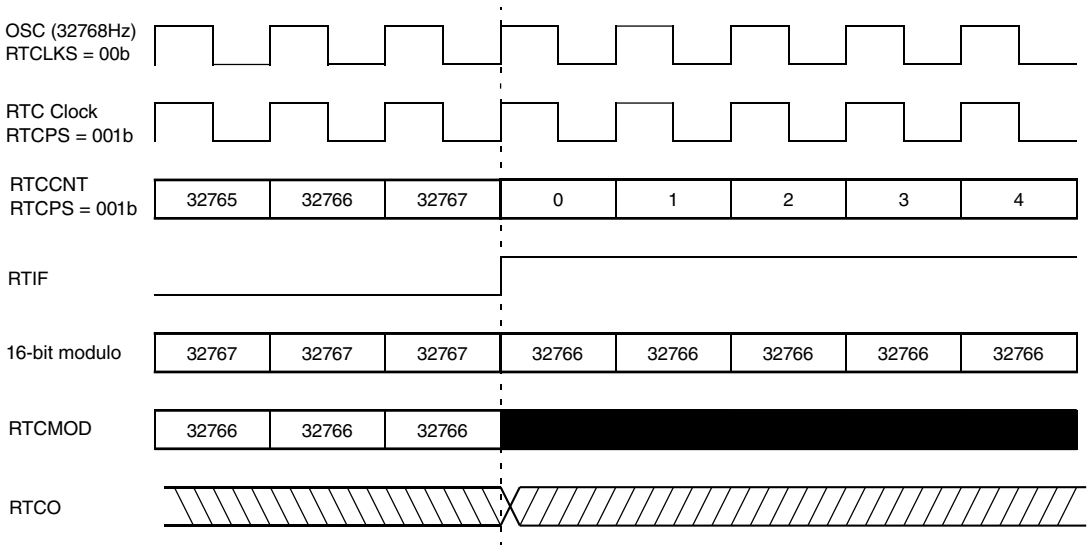


Figure 30-5. RTC counter overflow example

In the above example, the external clock source is selected. The prescaler is set to RTC_SC[RTCPS] = 001b or passthrough. The actual modulo value used by 16-bit comparator is 32767, when the modulo value in the RTC_MOD register is set to 32766. When the counter, RTC_CNT, reaches the modulo value of 32767, the counter overflows to 0x00 and continues counting. The modulo value is updated by fetching from RTC_MOD register. The real-time interrupt flag, RTC_SC[RTIF], sets when the counter value changes from 0x7FFF to 0x0000. The RTC_SC[RTCO] toggles as well when the RTC_SC[RTIF] is set.

30.7 Initialization/application information

This section provides example code to give some basic direction to a user on how to initialize and configure the RTC module. The example software is implemented in C language.

The example below shows how to implement time of day with the RTC using the OSC clock source to achieve the lowest possible power consumption.

Example: 30.7.1 Software calendar implementation in RTC ISR

```
/* Initialize the elapsed time counters */
Seconds = 0;
Minutes = 0;
Hours = 0;
Days=0;

/* Configure RTC to interrupt every 1 second from OSC (32.768KHz) clock source */
RTC_MOD = 511; // overflow every 32 times
RTC_SC = RTC_SC_RTCPS_MASK; // external 32768 clock selected with 1/64 predivider.
RTC_SC = RTC_SC_RTIF_MASK | RTC_SC_RTIE_MASK; // interrupt cleared and enabled

/*********************************************************************
Function Name : RTC_ISR
Notes : Interrupt service routine for RTC module.
*********************************************************************/
void RTC_ISR(void)
{
/* Clears the interrupt flag, RTIF, and interrupt request */
RTC_SC |= RTC_SC_RTIF_MASK;

/* RTC interrupts every 1 Second */
Seconds++;

/* 60 seconds in a minute */
if (Seconds > 59)
{
Minutes++;
Seconds = 0;
}

/* 60 minutes in an hour */
if (Minutes > 59)
{
Hours++;
Minutes = 0;
}

/* 24 hours in a day */
if (Hours > 23)
{
Days ++;
Hours = 0;
}
}
```

Chapter 31
Inter-Integrated Circuit (I2C)

31.1 Chip-specific I2C information

31.1.1 I2C overview

This device contains a inter-integrated circuit (I2C) module providing 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 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.

31.2 Introduction

The inter-integrated circuit (I^2C, 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*.

31.2.1 Features

The I2C module has the following features:

- Compatible with *The I^2C-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

31.2.2 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.

31.2.3 Block diagram

The following figure is a functional block diagram of the I2C module.

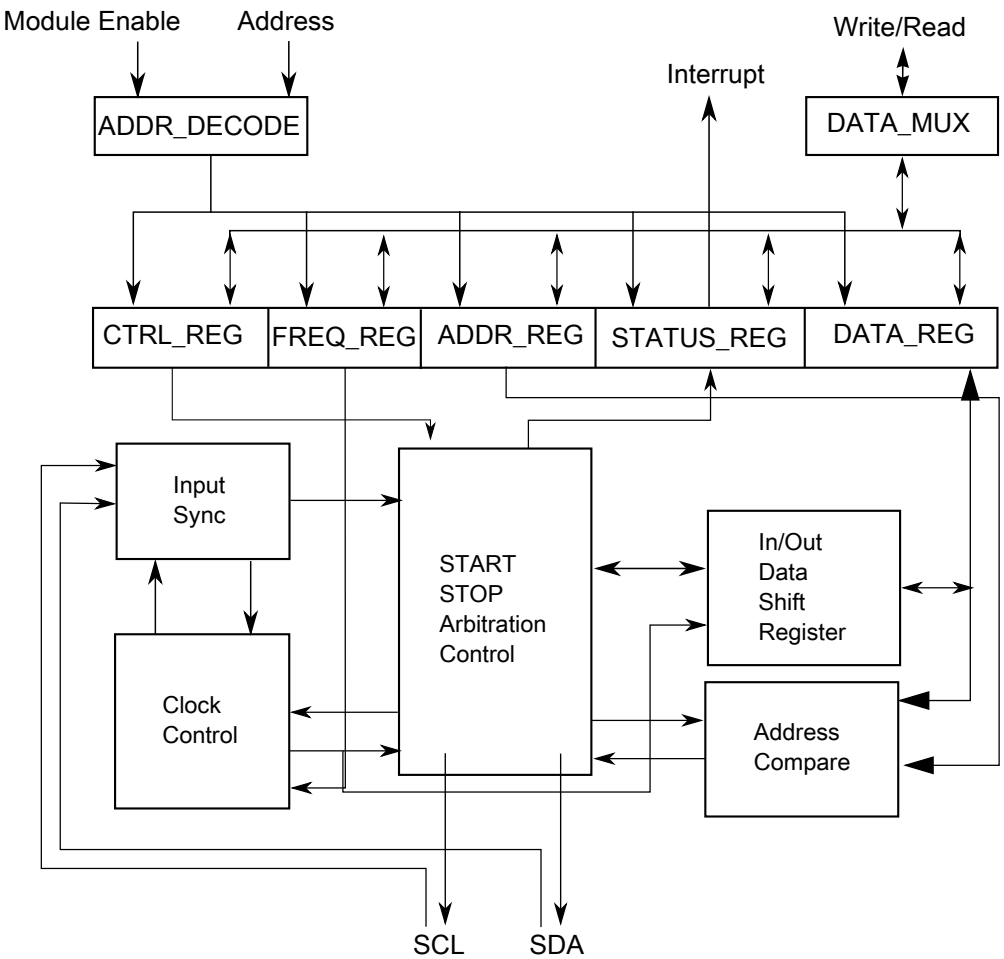


Figure 31-1. I2C Functional block diagram

31.3 I²C signal descriptions

The signal properties of I²C are shown in the table found here.

Table 31-1. I²C signal descriptions

Signal	Description	I/O
SCL	Bidirectional serial clock line of the I²C system.	I/O
SDA	Bidirectional serial data line of the I²C system.	I/O

31.4 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	31.4.1/452
4006_6001	I2C Frequency Divider register (I2C0_F)	8	R/W	00h	31.4.2/453
4006_6002	I2C Control Register 1 (I2C0_C1)	8	R/W	00h	31.4.3/454
4006_6003	I2C Status register (I2C0_S)	8	R/W	80h	31.4.4/455
4006_6004	I2C Data I/O register (I2C0_D)	8	R/W	00h	31.4.5/457
4006_6005	I2C Control Register 2 (I2C0_C2)	8	R/W	00h	31.4.6/458
4006_6006	I2C Programmable Input Glitch Filter Register (I2C0_FLT)	8	R/W	00h	31.4.7/459
4006_6007	I2C Range Address register (I2C0_RA)	8	R/W	00h	31.4.8/460
4006_6008	I2C SMBus Control and Status register (I2C0_SMB)	8	R/W	00h	31.4.9/460
4006_6009	I2C Address Register 2 (I2C0_A2)	8	R/W	C2h	31.4.10/462
4006_600A	I2C SCL Low Timeout Register High (I2C0_SLTH)	8	R/W	00h	31.4.11/462
4006_600B	I2C SCL Low Timeout Register Low (I2C0_SLTL)	8	R/W	00h	31.4.12/463

31.4.1 I2C Address Register 1 (I2Cx_A1)

This register contains the slave address to be used by the I2C module.

Address: 4006_6000h base + 0h offset = 4006_6000h

Bit	7	6	5	4	3	2	1	0
Read Write	AD[7:1]							0
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.

31.4.2 I2C Frequency Divider register (I2Cx_F)

Address: 4006_6000h base + 1h offset = 4006_6001h

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)`
	
	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

Table continues on the next page...

I2Cx_F field descriptions (continued)

Field	Description				
	MULT	ICR	Hold times (µs)		
			SDA	SCL Start	SCL Stop
	0h	18h	1.125	4.750	5.125

31.4.3 I2C Control Register 1 (I2Cx_C1)

Address: 4006_6000h base + 2h offset = 4006_6002h

Bit	7	6	5	4	3	2	1	0
Read	IICEN	IICIE	MST	TX	TXAK	0	WUEN	0
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. 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

Table continues on the next page...

I2Cx_C1 field descriptions (continued)

Field	Description
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 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

31.4.4 I2C Status register (I2Cx_S)

Address: 4006_6000h base + 3h offset = 4006_6003h

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:

Table continues on the next page...

I2Cx_S field descriptions (continued)

Field	Description
	• 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. 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. 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.

Table continues on the next page...

I2Cx_S field descriptions (continued)

Field	Description
	• 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 • I2C bus stop or start detection if the SSIE bit in the Input Glitch Filter register is 1 **NOTE:** To clear the I2C bus stop or start detection interrupt: In the interrupt service routine, first clear the STOPF or STARTF bit in the Input Glitch Filter register by writing 1 to it, and then clear the IICIF bit. If this sequence is reversed, the IICIF bit is asserted again. 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

31.4.5 I2C Data I/O register (I2Cx_D)

Address: 4006_6000h base + 4h offset = 4006_6004h

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).

31.4.6 I2C Control Register 2 (I2Cx_C2)

Address: 4006_6000h base + 5h offset = 4006_6005h

Bit	7	6	5	4	3	2	1	0
Read	GCAEN	ADEXT	0	SBRC	RMEN	AD[10:8]		
Write								
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
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 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
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.

31.4.7 I2C Programmable Input Glitch Filter Register (I2Cx_FLT)

Address: 4006_6000h base + 6h offset = 4006_6006h

Bit	7	6	5	4	3	2	1	0
Read	SHEN	STOPF	SSIE	STARTF	FLT			
Write		w1c		w1c				
Reset	0	0	0	0	0	0	0	0

I2Cx_FLT field descriptions

Field	Description
7 SHEN	Stop Hold Enable Set this bit to hold off entry to stop mode when any data transmission or reception is occurring. The following scenario explains the holdoff functionality: 1. The I2C module is configured for a basic transfer, and the SHEN bit is set to 1. 2. A transfer begins. 3. The MCU signals the I2C module to enter stop mode. 4. The byte currently being transferred, including both address and data, completes its transfer. 5. The I2C slave or master acknowledges that the in-transfer byte completed its transfer and acknowledges the request to enter stop mode. 6. After receiving the I2C module's acknowledgment of the request to enter stop mode, the MCU determines whether to shut off the I2C module's clock. If the SHEN bit is set to 1 and the I2C module is in an idle or disabled state when the MCU signals to enter stop mode, the module immediately acknowledges the request to enter stop mode. If SHEN is cleared to 0 and the overall data transmission or reception that was suspended by stop mode entry was incomplete: To resume the overall transmission or reception after the MCU exits stop mode, software must reinitialize the transfer by resending the address of the slave. If the I2C Control Register 1's IICIE bit was set to 1 before the MCU entered stop mode, system software will receive the interrupt triggered by the I2C Status Register's TCF bit after the MCU wakes from the stop mode. 0 Stop holdoff is disabled. The MCU's entry to stop mode is not gated. 1 Stop holdoff is enabled.
6 STOPF	I2C Bus Stop Detect Flag Hardware sets this bit when the I2C bus's stop status is detected. The STOPF bit must be cleared by writing 1 to it. 0 No stop happens on I2C bus 1 Stop detected on I2C bus
5 SSIE	I2C Bus Stop or Start Interrupt Enable This bit enables the interrupt for I2C bus stop or start detection. **NOTE:** To clear the I2C bus stop or start detection interrupt: In the interrupt service routine, first clear the STOPF or STARTF bit by writing 1 to it, and then clear the IICIF bit in the status register. If this sequence is reversed, the IICIF bit is asserted again. 0 Stop or start detection interrupt is disabled 1 Stop or start detection interrupt is enabled

Table continues on the next page...

I2Cx_FLT field descriptions (continued)

Field	Description
4 STARTF	I2C Bus Start Detect Flag Hardware sets this bit when the I2C bus's start status is detected. The STARTF bit must be cleared by writing 1 to it. 0 No start happens on I2C bus 1 Start detected on I2C bus
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. 0h No filter/bypass 1-Fh Filter glitches up to width of *n* I2C module clock cycles, where *n*=1-15d

31.4.8 I2C Range Address register (I2Cx_RA)

Address: 4006_6000h base + 7h offset = 4006_6007h

Bit	7	6	5	4	3	2	1	0
Read Write	RAD							0
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.

31.4.9 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: 4006_6000h base + 8h offset = 4006_6008h

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

Table continues on the next page...

I2Cx_SMB field descriptions (continued)

Field	Description
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
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

31.4.10 I2C Address Register 2 (I2Cx_A2)

Address: 4006_6000h base + 9h offset = 4006_6009h

Bit	7	6	5	4	3	2	1	0
Read Write	SAD							0
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.

31.4.11 I2C SCL Low Timeout Register High (I2Cx_SLTH)

Address: 4006_6000h base + Ah offset = 4006_600Ah

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.

31.4.12 I2C SCL Low Timeout Register Low (I2Cx_SLTL)

Address: 4006_6000h base + Bh offset = 4006_600Bh

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.

31.5 Functional description

This section provides a comprehensive functional description of the I2C module.

31.5.1 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.

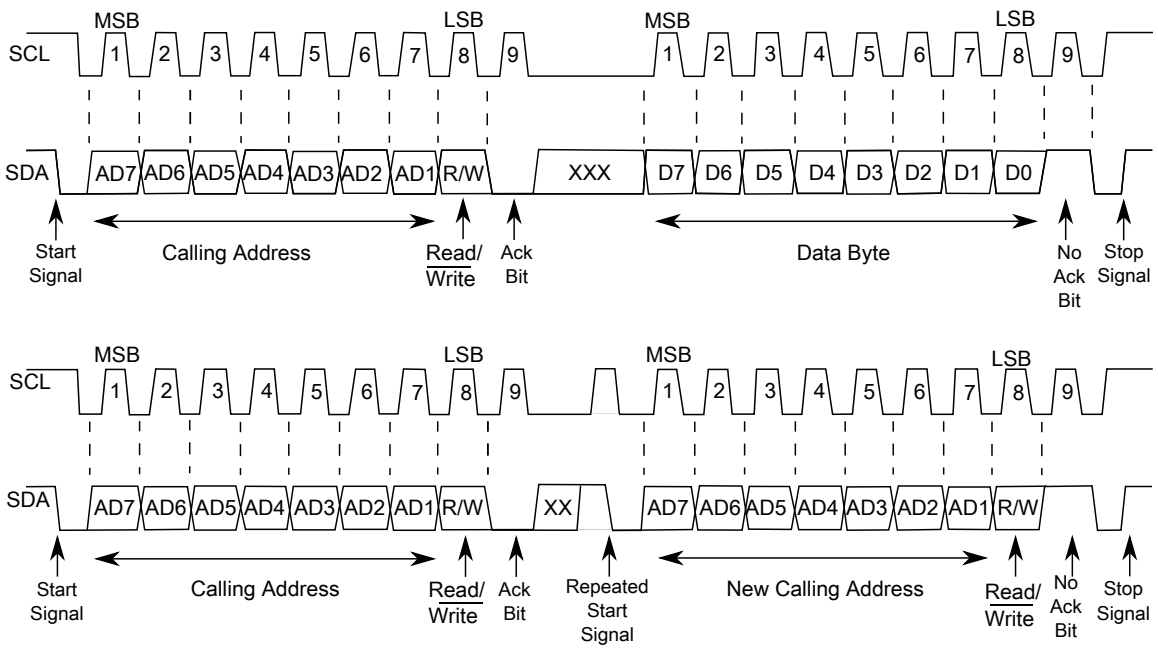


Figure 31-26. I2C bus transmission signals

31.5.1.1 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.

31.5.1.2 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/$\overline{W}$ bit. The R/$\overline{W}$ 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.

31.5.1.3 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/$\overline{W}$ 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.

31.5.1.4 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.

31.5.1.5 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.

31.5.1.6 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.

31.5.1.7 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.

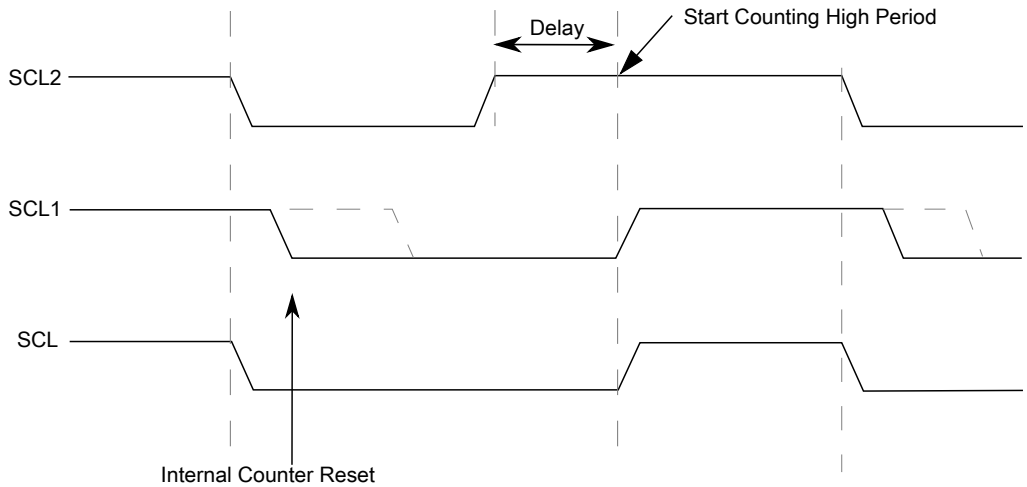


Figure 31-27. I2C clock synchronization

31.5.1.8 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.

31.5.1.9 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.

31.5.1.10 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 31-28. 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

31.5.2 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.

31.5.2.1 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 31-29. 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.

31.5.2.2 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 31-30. 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.

31.5.3 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.

31.5.4 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.

31.5.4.1 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.

31.5.4.1.1 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.

31.5.4.1.2 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.

31.5.4.1.3 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.

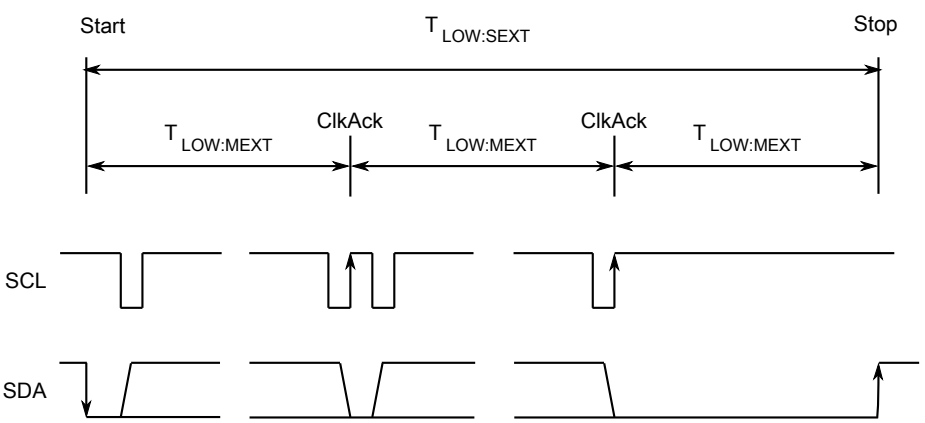


Figure 31-28. Timeout measurement intervals

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.

31.5.4.2 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.

31.5.5 Resets

The I2C module is disabled after a reset. The I2C module cannot cause a core reset.

31.5.6 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 31-31. 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
I^2C bus stop detection	STOPF	IICIF	IICIE & SSIE
I^2C bus start detection	STARTF	IICIF	IICIE & SSIE
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

31.5.6.1 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.

31.5.6.2 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.

31.5.6.3 Stop Detect Interrupt

When the stop status is detected on the I^2C bus, the SSIE bit is set to 1. The CPU is interrupted, provided the IICIE and SSIE bits are both set to 1.

31.5.6.4 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.

31.5.6.5 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.

31.5.6.6 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.

31.5.7 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.

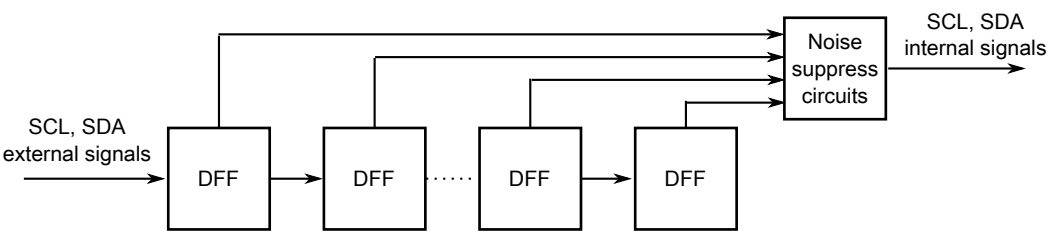


Figure 31-29. Programmable input glitch filter diagram

31.5.8 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.

After the address matching IAAS bit is set, an interrupt is sent at the end of address matching to wake the core.

NOTE

During the wake-up process, if an external master continues to send data to the slave, the baud rate under Stop mode must be less than 50 kbit/s. To avoid the slower baud rate under Stop mode, the master can add a short delay in firmware to wait until the wake-up process is complete and then send data.

NOTE

Wake-up caused by an address match is not supported for SMBus mode.

31.6 Initialization/application information

Module Initialization (Slave)

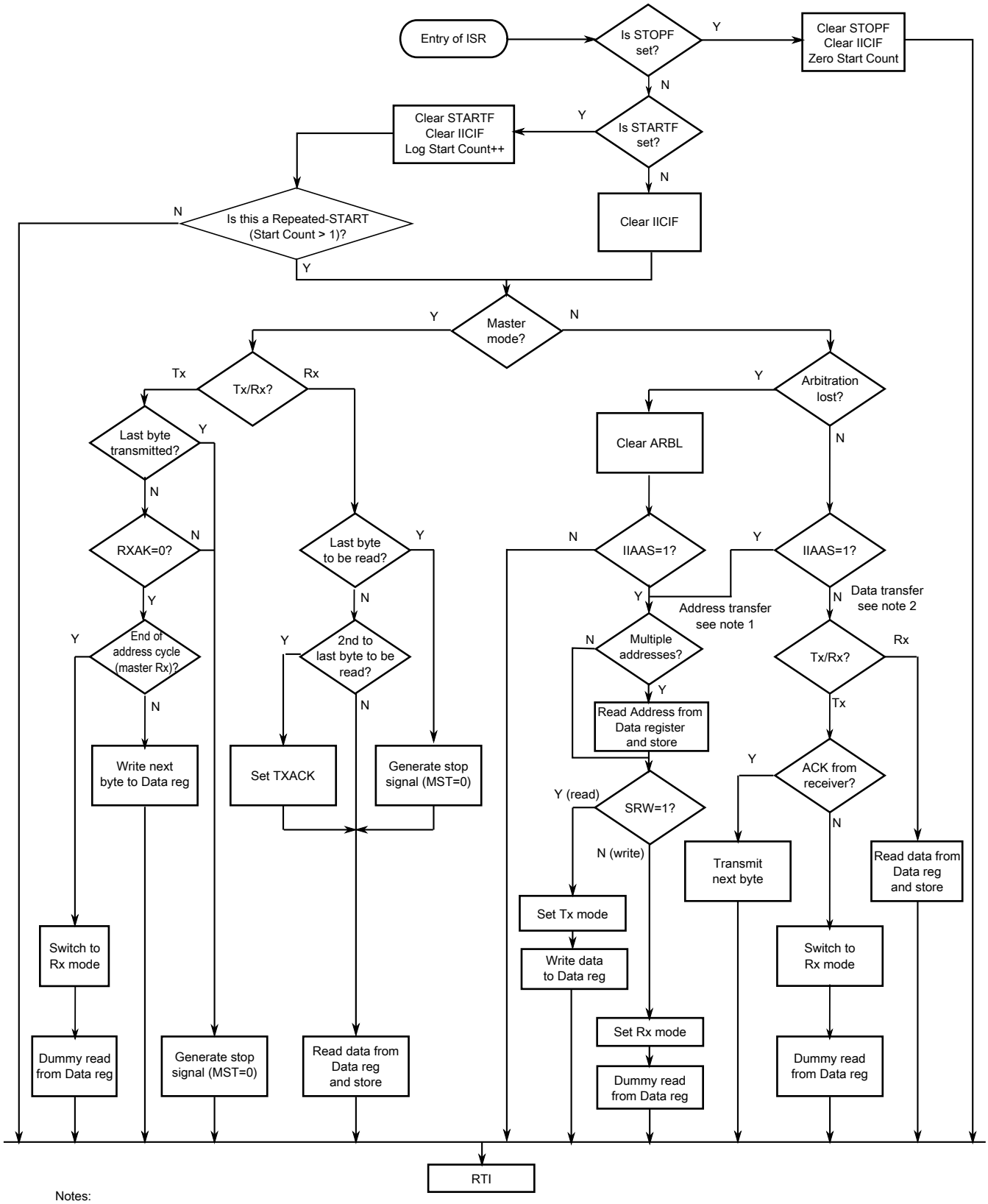
1. Write: Control Register 2
   - to enable or disable general call
   - to select 10-bit or 7-bit addressing mode
2. Write: Address Register 1 to set the slave address
3. Write: Control Register 1 to enable the I2C module and interrupts
4. Initialize RAM variables (IICEN = 1 and IICIE = 1) for transmit 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)

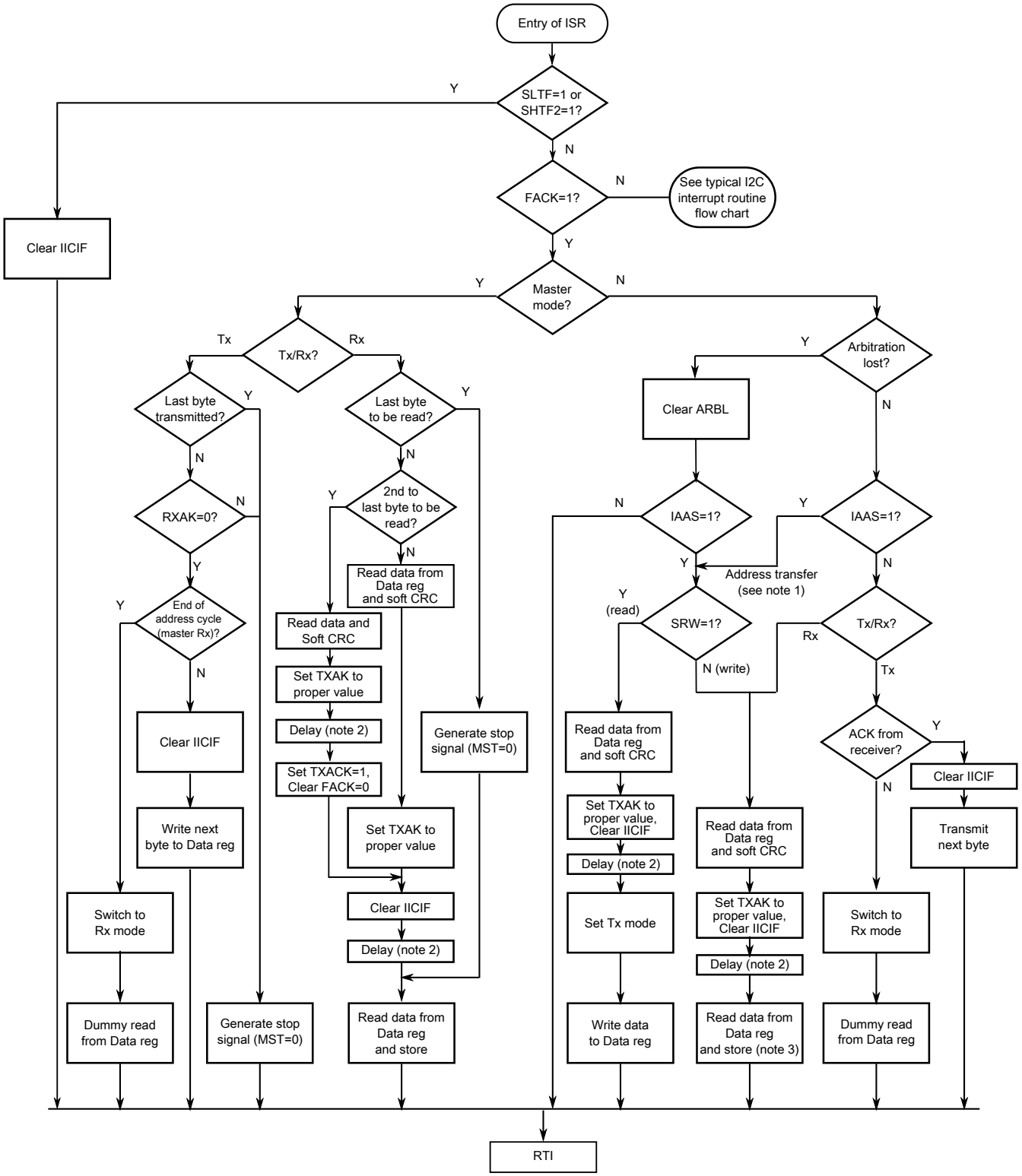
The routine shown in the following figure encompasses both master and slave I2C operations. For slave operation, an incoming I2C message that contains the proper address begins I2C communication. For master operation, communication must be

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 31-30. 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 31-31. Typical I2C SMBus interrupt routine

Chapter 32
Universal Asynchronous Receiver/Transmitter (UART)

32.1 Chip-specific UART information

32.1.1 UART interconnection

UART0 signal interconnection with other modules can be implemented by configuring the SBAR (signal crossbar) and AOI(And-Or-Invert module), See System Integration Module (SIM) for details.

UART0_TX Modulation:
- UART0_TX output can be modulated by PWM output from FTM

UART0_Rx filter:
- UART0_Rx be from ACMP0_OUT

UART0_RX capture:
- UART0_RX input captured by FTM input channel

32.2 Introduction

32.2.1 Features

Features of UART module include:
- Full-duplex, standard non-return-to-zero (NRZ) format
- Double-buffered transmitter and receiver with separate enables
- Programmable baud rates (13-bit modulo divider)
- Interrupt-driven or polled operation:

  - Transmit data register empty and transmission complete
  - Receive data register full
  - Receive overrun, parity error, framing error, and noise error
  - Idle receiver detect
  - Active edge on receive pin
  - Break detect supporting LIN
- Hardware parity generation and checking
- Programmable 8-bit or 9-bit character length
- Programmable 1-bit or 2-bit stop bits
- Receiver wakeup by idle-line or address-mark
- Optional 13-bit break character generation / 11-bit break character detection
- Selectable transmitter output polarity

32.2.2 Modes of operation

See Section Functional description for details concerning UART operation in these modes:

- 8- and 9-bit data modes
- Stop mode operation
- Loop mode
- Single-wire mode

32.2.3 Block diagram

The following figure shows the transmitter portion of the UART.

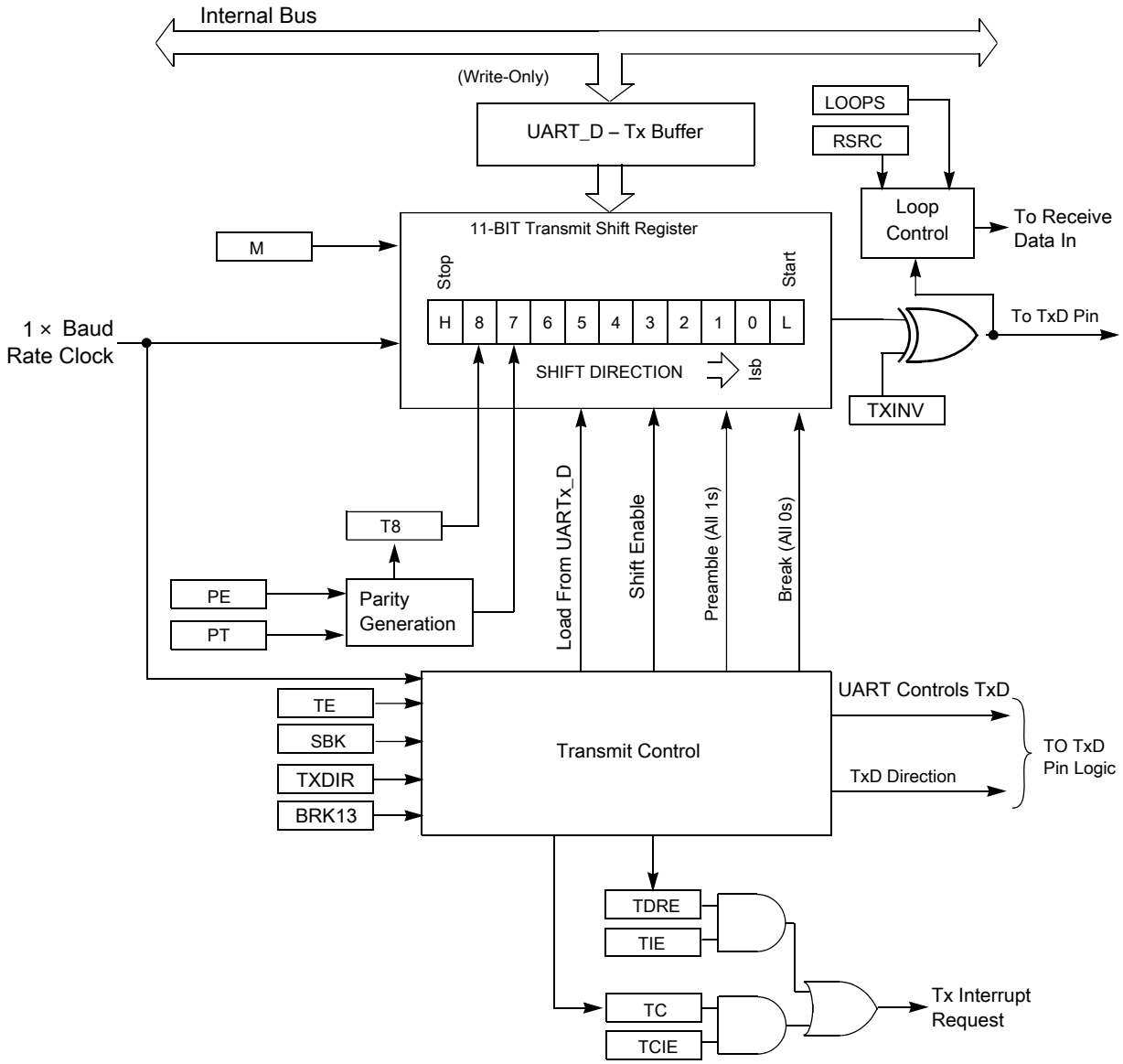


Figure 32-1. UART transmitter block diagram

The following figure shows the receiver portion of the UART.

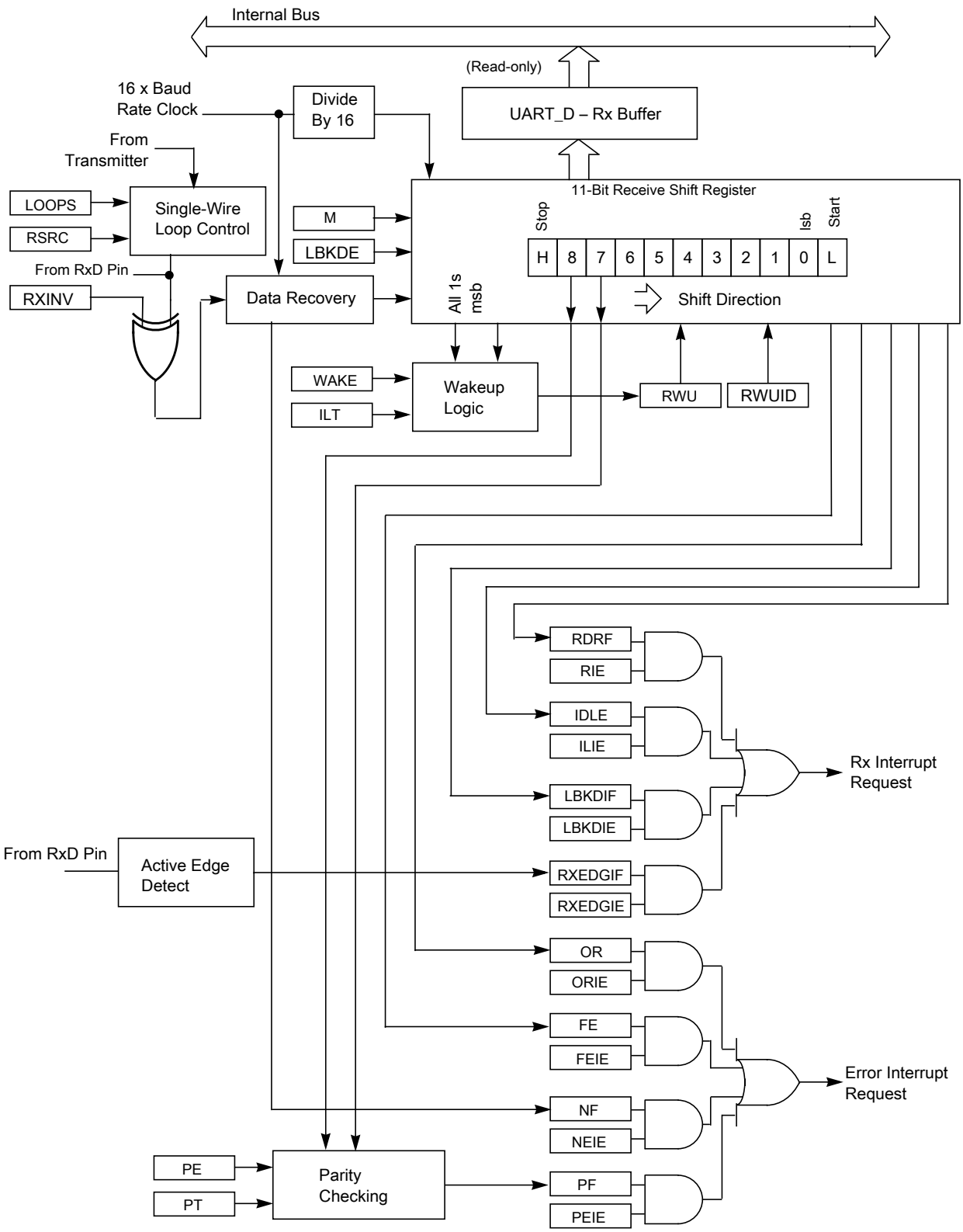


Figure 32-2. UART receiver block diagram

32.3 UART signal descriptions

The UART signals are shown in the table found here.

Table 32-1. UART signal descriptions

Signal	Description	I/O
RxD	Receive data	I
TxD	Transmit data	I/O

32.3.1 Detailed signal descriptions

The detailed signal descriptions of the UART are shown in the following table.

Table 32-2. UART—Detailed signal descriptions

Signal	I/O	Description	
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	I/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.

32.4 Register definition

The UART has 8-bit registers to control baud rate, select UART options, report UART status, and for transmit/receive data.

Refer to the direct-page register summary in the memory chapter of this document or the absolute address assignments for all UART registers. This section refers to registers and control bits only by their names. A Freescale-provided equate or header file is used to translate these names into the appropriate absolute addresses.

UART memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4006_A000	UART Baud Rate Register: High (UART0_BDH)	8	R/W	00h	32.4.1/486
4006_A001	UART Baud Rate Register: Low (UART0_BDL)	8	R/W	04h	32.4.2/487
4006_A002	UART Control Register 1 (UART0_C1)	8	R/W	00h	32.4.3/487
4006_A003	UART Control Register 2 (UART0_C2)	8	R/W	00h	32.4.4/489
4006_A004	UART Status Register 1 (UART0_S1)	8	R	C0h	32.4.5/490
4006_A005	UART Status Register 2 (UART0_S2)	8	R/W	00h	32.4.6/492
4006_A006	UART Control Register 3 (UART0_C3)	8	R/W	00h	32.4.7/494
4006_A007	UART Data Register (UART0_D)	8	R/W	00h	32.4.8/495

32.4.1 UART Baud Rate Register: High (UARTx_BDH)

This register, along with UART_BDL, controls the prescale divisor for UART baud rate generation. To update the 13-bit baud rate setting SBR[12:0], first write to UART_BDH to buffer the high half of the new value and then write to UART_BDL. The working value in UART_BDH does not change until UART_BDL is written.

Address: 4006_A000h base + 0h offset = 4006_A000h

Bit	7	6	5	4	3	2	1	0
Read Write	LBKDIE	RXEDGIE	SBNS	SBR				
Reset	0	0	0	0	0	0	0	0

UARTx_BDH field descriptions

Field	Description
7 LBKDIE	LIN Break Detect Interrupt Enable (for LBKDIF) 0 Hardware interrupts from UART_S2[LBKDIF] disabled (use polling). 1 Hardware interrupt requested when UART_S2[LBKDIF] flag is 1.
6 RXEDGIE	RxD Input Active Edge Interrupt Enable (for RXEDGIF) 0 Hardware interrupts from UART_S2[RXEDGIF] disabled (use polling). 1 Hardware interrupt requested when UART_S2[RXEDGIF] flag is 1.
5 SBNS	Stop Bit Number Select SBNS determines whether data characters are one or two stop bits. 0 One stop bit. 1 Two stop bit.
SBR	Baud Rate Modulo Divisor.

Table continues on the next page...

UARTx_BDH field descriptions (continued)

Field	Description
	The 13 bits in SBR[12:0] are referred to collectively as BR, and they set the modulo divide rate for the UART baud rate generator. When BR is cleared, the UART baud rate generator is disabled to reduce supply current. When BR is 1 - 8191, the UART baud rate equals BUSCLK/(16×BR).

32.4.2 UART Baud Rate Register: Low (UARTx_BDL)

This register, along with UART_BDH, control the prescale divisor for UART baud rate generation. To update the 13-bit baud rate setting [SBR12:SBR0], first write to UART_BDH to buffer the high half of the new value and then write to UART_BDL. The working value in UART_BDH does not change until UART_BDL is written.

UART_BDL is reset to a non-zero value, so after reset, the baud rate generator remains disabled until the first time the receiver or transmitter is enabled; that is, 1 is written to UART_C2[RE] or UART_C2[TE].

Address: 4006_A000h base + 1h offset = 4006_A001h

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

UARTx_BDL field descriptions

Field	Description
SBR	Baud Rate Modulo Divisor These 13 bits in SBR[12:0] are referred to collectively as BR, which set the modulo divide rate for the UART baud rate generator. When BR is cleared, the UART baud rate generator is disabled to reduce supply current. When BR is 1 - 8191, the UART baud rate equals BUSCLK/(16×BR).

32.4.3 UART Control Register 1 (UARTx_C1)

This read/write register controls various optional features of the UART system.

Address: 4006_A000h base + 2h offset = 4006_A002h

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 Selects between loop mode and normal 2-pin full-duplex modes. When LOOPS is set, the transmitter output is internally connected to the receiver input. 0 Normal operation - RxD and TxD use separate pins. 1 Loop mode or single-wire mode where transmitter outputs are internally connected to receiver input. (See RSRC bit.) RxD pin is not used by UART.
6 UARTSWAI	UART Stops in Wait Mode 0 UART clocks continue to run in Wait mode so the UART can be the source of an interrupt that wakes up the CPU. 1 UART clocks freeze while CPU is in Wait mode.
5 RSRC	Receiver Source Select This field has no meaning or effect unless LOOPS is set to 1. When LOOPS is set, the receiver input is internally connected to the TxD pin and RSRC determines whether this connection is also connected to the transmitter output. 0 Provided LOOPS is set, RSRC is cleared, selects internal loop back mode and the UART does not use the RxD pins. 1 Single-wire UART mode where the TxD pin is connected to the transmitter output and receiver input.
4 M	9-Bit or 8-Bit Mode Select This field configures the UART to be operated in 9-bit or 8-bit data mode. 0 Normal - start + 8 data bits (lsb first) + stop. 1 Receiver and transmitter use 9-bit data characters start + 8 data bits (lsb first) + 9th data bit + stop.
3 WAKE	Receiver Wakeup Method Select This field selects the receiver wakeup method. 0 Idle-line wake-up. 1 Address-mark wake-up.
2 ILT	Idle Line Type Select Setting this field to 1 ensures that the stop bits and logic 1 bits at the end of a character do not count toward the 10 or 11 bit times of logic high level needed by the idle line detection logic. 0 Idle character bit count starts after start bit. 1 Idle character bit count starts after stop bit.
1 PE	Parity Enable Enables hardware parity generation and checking. When parity is enabled, the most significant bit (msb) of the data character, eighth or ninth data bit, is treated as the parity bit. 0 No hardware parity generation or checking. 1 Parity enabled.
0 PT	Parity Type

Table continues on the next page...

UARTx_C1 field descriptions (continued)

Field	Description
	Provided parity is enabled (PE = 1), this field selects even or odd parity. Odd parity means the total number of 1s in the data character, including the parity bit, is odd. Even parity means the total number of 1s in the data character, including the parity bit, is even. 0 Even parity. 1 Odd parity.

32.4.4 UART Control Register 2 (UARTx_C2)

This register can be read or written at any time.

Address: 4006_A000h base + 3h offset = 4006_A003h

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	Transmit Interrupt Enable for TDRE 0 Hardware interrupts from TDRE disabled; use polling. 1 Hardware interrupt requested when TDRE flag is 1.
6 TCIE	Transmission Complete Interrupt Enable for TC 0 Hardware interrupts from TC disabled; use polling. 1 Hardware interrupt requested when TC flag is 1.
5 RIE	Receiver Interrupt Enable for RDRF 0 Hardware interrupts from S1[RDRF] disabled; use polling. 1 Hardware interrupt requested when S1[RDRF] flag is 1.
4 ILIE	Idle Line Interrupt Enable for IDLE 0 Hardware interrupts from S1[IDLE] disabled; use polling. 1 Hardware interrupt requested when S1[IDLE] flag is 1.
3 TE	Transmitter Enable TE must be 1 to use the UART transmitter. When TE is set, the UART forces the TxD pin to act as an output for the UART system. When the UART is configured for single-wire operation (LOOPS = RSRC = 1), TXDIR controls the direction of traffic on the single UART communication line (TxD pin). TE can also queue an idle character by clearing TE and then setting TE while a transmission is in progress. When 0 is written to TE, the transmitter keeps control of the port TxD pin until any data, queued idle, or queued break character finishes transmitting before allowing the pin to revert to a general-purpose I/O pin.

Table continues on the next page...

UARTx_C2 field descriptions (continued)

Field	Description
	0 Transmitter off. 1 Transmitter on.
2 RE	Receiver Enable When the UART receiver is off, the RxD pin reverts to being a general-purpose port I/O pin. If C1[LOOPS] is set, the RxD pin reverts to being a general-purpose I/O pin even if RE is set. 0 Receiver off. 1 Receiver on.
1 RWU	Receiver Wakeup Control A 1 can be written to this field to place the UART receiver in a standby state where it waits for automatic hardware detection of a selected wake-up condition. The wake-up condition is an idle line between messages, WAKE = 0, idle-line wake-up, or a logic 1 in the most significant data bit in a character, WAKE = 1, address-mark wake-up. Application software sets RWU and, normally, a selected hardware condition automatically clears RWU. 0 Normal UART receiver operation. 1 UART receiver in standby waiting for wake-up condition.
0 SBK	Send Break Writing a 1 and then a 0 to SBK queues a break character in the transmit data stream. Additional break characters of 10 or 11 or 12, 13 or 14 or 15 if BRK13 = 1, bit times of logic 0 are queued as long as SBK is set. Depending on the timing of the set and clear of SBK relative to the information currently being transmitted, a second break character may be queued before software clears SBK. 0 Normal transmitter operation. 1 Queue break character(s) to be sent.

32.4.5 UART Status Register 1 (UARTx_S1)

This register has eight read-only status flags. Writes have no effect. Special software sequences, which do not involve writing to this register, clear these status flags.

Address: 4006_A000h base + 4h offset = 4006_A004h

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 is set out of reset and when a transmit data value transfers from the transmit data buffer to the transmit shifter, leaving room for a new character in the buffer. To clear TDRE, read UART_S1 with TDRE set and then write to the UART data register (UART_D).

Table continues on the next page...

UARTx_S1 field descriptions (continued)

Field	Description
	0 Transmit data register (buffer) full. 1 Transmit data register (buffer) empty.
6 TC	Transmission Complete Flag TC is set out of reset and when TDRE is set and no data, preamble, or break character is being transmitted. TC is cleared automatically by reading UART_S1 with TC set and then executing one of the following operations: • Write to the UART data register (UART_D) to transmit new data • Queue a preamble by changing TE from 0 to 1 • Queue a break character by writing 1 to UART_C2[SBK] 0 Transmitter active (sending data, a preamble, or a break). 1 Transmitter idle (transmission activity complete).
5 RDRF	Receive Data Register Full Flag RDRF becomes set when a character transfers from the receive shifter into the receive data register (UART_D). To clear RDRF, read UART_S1 with RDRF set and then read the UART data register (UART_D). 0 Receive data register empty. 1 Receive data register full.
4 IDLE	Idle Line Flag IDLE is set when the UART receive line becomes idle for a full character time after a period of activity. When C1[ILT] is cleared, the receiver starts counting idle bit times after the start bit. If the receive character is all 1s, these bit times and the stop bits time count toward the full character time of logic high, 10 or 11 bit times depending on the M control bit, needed for the receiver to detect an idle line. When ILT is set, the receiver doesn't start counting idle bit times until the stop bits. The stop bits and any logic high bit times at the end of the previous character do not count toward the full character time of logic high needed for the receiver to detect an idle line. To clear IDLE, read UART_S1 with IDLE set and then read the UART data register (UART_D). After IDLE has been cleared, it cannot become set again until after a new character has been received and RDRF has been set. IDLE is set only once even if the receive line remains idle for an extended period. 0 No idle line detected. 1 Idle line was detected.
3 OR	Receiver Overrun Flag OR is set when a new serial character is ready to be transferred to the receive data register (buffer), but the previously received character has not been read from UART_D yet. In this case, the new character, and all associated error information, is lost because there is no room to move it into UART_D. To clear OR, read UART_S1 with OR set and then read the UART data register (UART_D). 0 No overrun. 1 Receive overrun (new UART data lost).
2 NF	Noise Flag The advanced sampling technique used in the receiver takes seven samples during the start bit and three samples in each data bit and the stop bits. If any of these samples disagrees with the rest of the samples within any bit time in the frame, the flag NF is set at the same time as RDRF is set for the character. To clear NF, read UART_S1 and then read the UART data register (UART_D).

Table continues on the next page...

UARTx_S1 field descriptions (continued)

Field	Description
	0 No noise detected. 1 Noise detected in the received character in UART_D.
1 FE	Framing Error Flag FE is set at the same time as RDRF when the receiver detects a logic 0 where the stop bits were expected. This suggests the receiver was not properly aligned to a character frame. To clear FE, read UART_S1 with FE set and then read the UART data register (UART_D). 0 No framing error detected. This does not guarantee the framing is correct. 1 Framing error.
0 PF	Parity Error Flag PF is set at the same time as RDRF when parity is enabled (PE = 1) and the parity bit in the received character does not agree with the expected parity value. To clear PF, read UART_S1 and then read the UART data register (UART_D). 0 No parity error. 1 Parity error.

32.4.6 UART Status Register 2 (UARTx_S2)

This register contains one read-only status flag.

When using an internal oscillator in a LIN system, it is necessary to raise the break detection threshold one bit time. Under the worst case timing conditions allowed in LIN, it is possible that a 0x00 data character can appear to be 10.26 bit times long at a slave running 14% faster than the master. This would trigger normal break detection circuitry designed to detect a 10-bit break symbol. When the LBKDE bit is set, framing errors are inhibited and the break detection threshold changes from 10 bits to 11 bits, preventing false detection of a 0x00 data character as a LIN break symbol.

Address: 4006_A000h base + 5h offset = 4006_A005h

Bit	7	6	5	4	3	2	1	0
Read	LBKDIF	RXEDGIF	0	RXINV	RWUID	BRK13	LBKDE	RAF
Write								
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 the LIN break detect circuitry is enabled and a LIN break character is detected. LBKDIF is cleared by writing a 1 to it.

Table continues on the next page...

UARTx_S2 field descriptions (continued)

Field	Description
	0 No LIN break character has been detected. 1 LIN break character has been detected.
6 RXEDGIF	RxD Pin Active Edge Interrupt Flag RXEDGIF is set when an active edge, falling if RXINV = 0, rising if RXINV=1, on the RxD pin occurs. RXEDGIF is cleared by writing a 1 to it. 0 No active edge on the receive pin has occurred. 1 An active edge on the receive pin has occurred.
5 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
4 RXINV	Receive Data Inversion Setting this field reverses the polarity of the received data input. **NOTE:** Setting RXINV inverts the RxD input for all cases: data bits, start and stop bits, break, and idle. 0 Receive data not inverted. 1 Receive data inverted.
3 RWUID	Receive Wake Up Idle Detect RWUID controls whether the idle character that wakes up the receiver sets S1[IDLE]. 0 During receive standby state (RWU = 1), S1[IDLE] does not get set upon detection of an idle character. 1 During receive standby state (RWU = 1), S1[IDLE] gets set upon detection of an idle character.
2 BRK13	Break Character Generation Length BRK13 selects a longer transmitted break character length. Detection of a framing error is not affected by the state of this field. 0 Break character is transmitted with length of 10 bit times (if M = 0, SBNS = 0) or 11 (if M = 1, SBNS = 0 or M = 0, SBNS = 1) or 12 (if M = 1, SBNS = 1). 1 Break character is transmitted with length of 13 bit times (if M = 0, SBNS = 0) or 14 (if M = 1, SBNS = 0 or M = 0, SBNS = 1) or 15 (if M = 1, SBNS = 1).
1 LBKDE	LIN Break Detection Enable LBKDE enables the break detection. While LBKDE is set, S1[FE] and S1[RDRF] flags are prevented from setting. 0 Break detection is disabled. 1 Break detection is enabled (Break character is detected at length 11 bit times (if C1[M] = 0, BDH[SBNS] = 0) or 12 (if C1[M] = 1, BDH[SBNS] = 0 or C1[M] = 0, BDH[SBNS] = 1) or 13 (if C1[M] = 1, BDH[SBNS] = 1)).
0 RAF	Receiver Active Flag RAF is set when the UART receiver detects the beginning of a valid start bit, and RAF is cleared automatically when the receiver detects an idle line. This status flag can be used to check whether an UART character is being received before instructing the MCU to go to stop mode. 0 UART receiver idle waiting for a start bit. 1 UART receiver active (RxD input not idle).

32.4.7 UART Control Register 3 (UARTx_C3)

Address: 4006_A000h base + 6h offset = 4006_A006h

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	Ninth Data Bit for Receiver When the UART is configured for 9-bit data (C1[M] = 1), R8 can be thought of as a ninth receive data bit to the left of the msb of the buffered data in the UART_D register. When reading 9-bit data, read R8 before reading UART_D because reading UART_D completes automatic flag clearing sequences that could allow R8 and UART_D to be overwritten with new data.
6 T8	Ninth Data Bit for Transmitter When the UART is configured for 9-bit data (C1[M] = 1), T8 may be thought of as a ninth transmit data bit to the left of the msb of the data in the UART_D register. When writing 9-bit data, the entire 9-bit value is transferred to the UART shift register after UART_D is written so T8 should be written, if it needs to change from its previous value, before UART_D is written. If T8 does not need to change in the new value, such as when it is used to generate mark or space parity, it need not be written each time UART_D is written.
5 TXDIR	TxD Pin Direction in Single-Wire Mode When the UART is configured for single-wire half-duplex operation (LOOPS = RSRC = 1), this field determines the direction of data at the TxD pin. 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. **NOTE:** Setting TXINV inverts the TxD output for all cases: data bits, start and stop bits, break, and idle. 0 Transmit data not inverted. 1 Transmit data inverted.
3 ORIE	Overrun Interrupt Enable Enables the overrun flag (OR) to generate hardware interrupt requests. 0 OR interrupts disabled; use polling. 1 Hardware interrupt requested when OR is set.
2 NEIE	Noise Error Interrupt Enable Enables the noise flag (NF) to generate hardware interrupt requests.

Table continues on the next page...

UARTx_C3 field descriptions (continued)

Field	Description
	0 NF interrupts disabled; use polling). 1 Hardware interrupt requested when NF is set.
1 FEIE	Framing Error Interrupt Enable Enables the framing error flag (FE) to generate hardware interrupt requests. 0 FE interrupts disabled; use polling). 1 Hardware interrupt requested when FE is set.
0 PEIE	Parity Error Interrupt Enable Enables the parity error flag (PF) to generate hardware interrupt requests. 0 PF interrupts disabled; use polling). 1 Hardware interrupt requested when PF is set.

32.4.8 UART Data Register (UARTx_D)

This register is actually two separate registers. Reads return the contents of the read-only receive data buffer and writes go to the write-only transmit data buffer. Reads and writes of this register are also involved in the automatic flag clearing mechanisms for the UART status flags.

Address: 4006_A000h base + 7h offset = 4006_A007h

Bit	7	6	5	4	3	2	1	0
Read Write	R7T7	R6T6	R5T5	R4T4	R3T3	R2T2	R1T1	R0T0
Reset	0	0	0	0	0	0	0	0

UARTx_D field descriptions

Field	Description
7 R7T7	Read receive data buffer 7 or write transmit data buffer 7.
6 R6T6	Read receive data buffer 6 or write transmit data buffer 6.
5 R5T5	Read receive data buffer 5 or write transmit data buffer 5.
4 R4T4	Read receive data buffer 4 or write transmit data buffer 4.
3 R3T3	Read receive data buffer 3 or write transmit data buffer 3.
2 R2T2	Read receive data buffer 2 or write transmit data buffer 2.

Table continues on the next page...

UARTx_D field descriptions (continued)

Field	Description
1 R1T1	Read receive data buffer 1 or write transmit data buffer 1.
0 R0T0	Read receive data buffer 0 or write transmit data buffer 0.

32.5 Functional description

The UART allows full-duplex, asynchronous, NRZ serial communication among the MCU and remote devices, including other MCUs.

The UART comprises a baud rate generator, transmitter, and receiver block. The transmitter and receiver operate independently, although they use the same baud rate generator. During normal operation, the MCU monitors the status of the UART, writes the data to be transmitted, and processes received data. The following describes each of the blocks of the UART.

32.5.1 Baud rate generation

As shown in the figure found here, the clock source for the UART baud rate generator is the bus-rate clock.

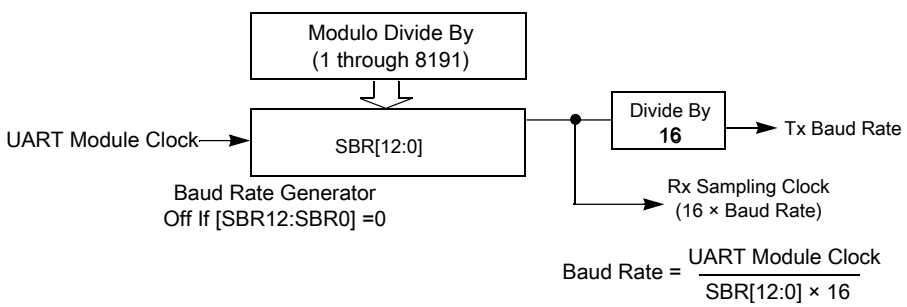


Figure 32-19. UART baud rate generation

UART communications require the transmitter and receiver, which typically derive baud rates from independent clock sources, to use the same baud rate. Allowed tolerance on this baud frequency depends on the details of how the receiver synchronizes to the leading edge of the start bit and how bit sampling is performed.

The MCU resynchronizes to bit boundaries on every high-to-low transition. In the worst case, there are no such transitions in the full 10- or 11-bit or 12-bittime character frame so any mismatch in baud rate is accumulated for the whole character time. For a

Freescale UART system whose bus frequency is driven by a crystal, the allowed baud rate mismatch is about ±4.5 percent for 8-bit data format and about ±4 percent for 9-bit data format. Although baud rate modulo divider settings do not always produce baud rates that exactly match standard rates, it is normally possible to get within a few percent, which is acceptable for reliable communications.

32.5.2 Transmitter functional description

This section describes the overall block diagram for the UART transmitter, as well as specialized functions for sending break and idle characters.

The transmitter output (TxD) idle state defaults to logic high, UART_C3[TXINV] is cleared following reset. The transmitter output is inverted by setting UART_C3[TXINV]. The transmitter is enabled by setting the TE bit in UART_C2. This queues a preamble character that is one full character frame of the idle state. The transmitter then remains idle until data is available in the transmit data buffer. Programs store data into the transmit data buffer by writing to the UART data register (UART_D).

The central element of the UART transmitter is the transmit shift register that is 10 or 11 or 12 bits long depending on the setting in the UART_C1[M] control bit and UART_BDH[SBNS] bit. For the remainder of this section, assume UART_C1[M] is cleared, UART_BDH[SBNS] is also cleared, selecting the normal 8-bit data mode. In 8-bit data mode, the shift register holds a start bit, eight data bits, and a stop bit. When the transmit shift register is available for a new UART character, the value waiting in the transmit data register is transferred to the shift register, synchronized with the baud rate clock, and the transmit data register empty (UART_S1[TDRE]) status flag is set to indicate another character may be written to the transmit data buffer at UART_D.

NOTE

Always read UART_S1 before writing to UART_D to allow data to be transmitted.

If no new character is waiting in the transmit data buffer after a stop bit is shifted out the TxD pin, the transmitter sets the transmit complete flag and enters an idle mode, with TxD high, waiting for more characters to transmit.

Writing 0 to UART_C2[TE] does not immediately release the pin to be a general-purpose I/O pin. Any transmit activity in progress must first be completed. This includes data characters in progress, queued idle characters, and queued break characters.

32.5.2.1 Send break and queued idle

UART_C2[SBK] sends break characters originally used to gain the attention of old teletype receivers. Break characters are a full character time of logic 0, 10 bit times including the start and stop bits. A longer break of 13 bit times can be enabled by setting UART_S2[BRK13]. Normally, a program would wait for UART_S1[TDRE] to become set to indicate the last character of a message has moved to the transmit shifter, write 1, and then write 0 to UART_C2[SBK]. This action queues a break character to be sent as soon as the shifter is available. If UART_C2[SBK] remains 1 when the queued break moves into the shifter, synchronized to the baud rate clock, an additional break character is queued. If the receiving device is another Freescale Semiconductor UART, the break characters are received as 0s in all eight data bits and a framing error (UART_S1[FE] = 1) occurs.

When idle-line wake-up is used, a full character time of idle (logic 1) is needed between messages to wake up any sleeping receivers. Normally, a program would wait for UART_S1[TDRE] to become set to indicate the last character of a message has moved to the transmit shifter, then write 0 and then write 1 to the UART_C2[TE] bit. This action queues an idle character to be sent as soon as the shifter is available. As long as the character in the shifter does not finish while UART_C2[TE] is cleared, the UART transmitter never actually releases control of the TxD pin. If there is a possibility of the shifter finishing while UART_C2[TE] is cleared, set the general-purpose I/O controls so the pin shared with TxD is an output driving a logic 1. This ensures that the TxD line looks like a normal idle line even if the UART loses control of the port pin between writing 0 and then 1 to UART_C2[TE].

The length of the break character is affected by the UART_S2[BRK13] and UART_C1[M] as shown below.

Table 32-21. Break character length

BRK13	M	SBNS	Break character length
0	0	0	10 bit times
0	0	1	11 bit times
0	1	0	11 bit times
0	1	1	12 bit times
1	0	0	13 bit times
1	0	1	14 bit times
1	1	0	14 bit times
1	1	1	15 bit times

32.5.3 Receiver functional description

In this section, the receiver block diagram is a guide for the overall receiver functional description.

Next, the data sampling technique used to reconstruct receiver data is described in more detail. Finally, two variations of the receiver wakeup function are explained.

The receiver input is inverted by setting UART_S2[RXINV]. The receiver is enabled by setting the UART_C2[RE] bit. Character frames consist of a start bit of logic 0, eight (or nine) data bits (lsb first), and one (or two) stop bits of logic 1. For information about 9-bit data mode, refer to 8- and 9-bit data modes. For the remainder of this discussion, assume the UART is configured for normal 8-bit data mode.

After receiving the stop bit into the receive shifter, and provided the receive data register is not already full, the data character is transferred to the receive data register and the receive data register full (UART_S1[RDRF]) status flag is set. If UART_S1[RDRF] was already set indicating the receive data register (buffer) was already full, the overrun (OR) status flag is set and the new data is lost. Because the UART receiver is double-buffered, the program has one full character time after UART_S1[RDRF] is set before the data in the receive data buffer must be read to avoid a receiver overrun.

When a program detects that the receive data register is full (UART_S1[RDRF] = 1), it gets the data from the receive data register by reading UART_D. The UART_S1[RDRF] flag is cleared automatically by a two-step sequence normally satisfied in the course of the user's program that manages receive data. Refer to Interrupts and status flags for more details about flag clearing.

32.5.3.1 Data sampling technique

The UART receiver uses a 16× baud rate clock for sampling. The oversampling ratio is fixed at 16. The receiver starts by taking logic level samples at 16 times the baud rate to search for a falling edge on the RxD serial data input pin. A falling edge is defined as a logic 0 sample after three consecutive logic 1 samples. The 16× baud rate clock divides the bit time into 16 segments labeled UART_D[RT1] through UART_D[RT16]. When a falling edge is located, three more samples are taken at UART_D[RT3], UART_D[RT5], and UART_D[RT7] to make sure this was a real start bit and not merely noise. If at least two of these three samples are 0, the receiver assumes it is synchronized to a receive character.

The receiver then samples each bit time, including the start and stop bits, at UART_D[RT8], UART_D[RT9], and UART_D[RT10] to determine the logic level for that bit. The logic level is interpreted to be that of the majority of the samples taken

during the bit time. In the case of the start bit, the bit is assumed to be 0 if at least two of the samples at UART_D[RT3], UART_D[RT5], and UART_D[RT7] are 0 even if one or all of the samples taken at UART_D[RT8], UART_D[RT9], and UART_D[RT10] are 1s. If any sample in any bit time, including the start and stop bits, in a character frame fails to agree with the logic level for that bit, the noise flag (UART_S1[NF]) is set when the received character is transferred to the receive data buffer.

The falling edge detection logic continuously looks for falling edges. If an edge is detected, the sample clock is resynchronized to bit times. This improves the reliability of the receiver in the presence of noise or mismatched baud rates. It does not improve worst case analysis because some characters do not have any extra falling edges anywhere in the character frame.

In the case of a framing error, provided the received character was not a break character, the sampling logic that searches for a falling edge is filled with three logic 1 samples so that a new start bit can be detected almost immediately.

In the case of a framing error, the receiver is inhibited from receiving any new characters until the framing error flag is cleared. The receive shift register continues to function, but a complete character cannot transfer to the receive data buffer if UART_S1[FE] remains set.

32.5.3.2 Receiver wake-up operation

Receiver wake-up is a hardware mechanism that allows an UART receiver to ignore the characters in a message intended for a different UART receiver. In such a system, all receivers evaluate the first character(s) of each message, and as soon as they determine the message is intended for a different receiver, they write logic 1 to the receiver wake up control field (UART_C2[RWU]). When UART_C2[RWU] is set, the status flags associated with the receiver, (with the exception of the idle bit, IDLE, when UART_S2[RWUID] is set), are inhibited from setting, thus eliminating the software overhead for handling the unimportant message characters. At the end of a message, or at the beginning of the next message, all receivers automatically force UART_C2[RWU] to 0, so all receivers wake up in time to look at the first character(s) of the next message.

32.5.3.2.1 Idle-line wakeup

When wake is cleared, the receiver is configured for idle-line wakeup. In this mode, UART_C2[RWU] is cleared automatically when the receiver detects a full character time of the idle-line level. The UART_C1[M] control field selects 8-bit or 9-bit data mode and

UART_BDH[SBNS] selects 1-bit or 2-bit stop bit number that determines how many bit times of idle are needed to constitute a full character time, 10 or 11 or 12 bit times because of the start and stop bits.

When UARTI_C2[RWU] is 1 and UART_S2[RWUID] is 0, the idle condition that wakes up the receiver does not set UART_S1[IDLE]. The receiver wakes up and waits for the first data character of the next message that sets UART_S1[RDRF] and generates an interrupt, if enabled. When UART_S2[RWUID] is 1, any idle condition sets UART_S1[IDLE] flag and generates an interrupt if enabled, regardless of whether UART_C2[RWU] is 0 or 1.

The idle-line type (UART_C1[ILT]) control bit selects one of two ways to detect an idle line. When UART_C1[ILT] is cleared, the idle bit counter starts after the start bit so the stop bit and any logic 1s at the end of a character count toward the full character time of idle. When UART_C1[ILT] is set, the idle bit counter does not start until after a stop bit time, so the idle detection is not affected by the data in the last character of the previous message.

32.5.3.2.2 Address-mark wakeup

When wake is set, the receiver is configured for address-mark wakeup. In this mode, UART_C2[RWU] is cleared automatically when the receiver detects a, or two, if UART_BDH[SBNS] = 1, logic 1 in the most significant bits of a received character, eighth bit when UART_C1[M] is cleared and ninth bit when UART_C1[M] is set.

Address-mark wakeup allows messages to contain idle characters, but requires the msb be reserved for use in address frames. The one, or two, if UART_BDH[SBNS] = 1, logic 1s msb of an address frame clears the UART_C2[RWU] bit before the stop bits are received and sets the UART_S1[RDRF] flag. In this case, the character with the msb set is received even though the receiver was sleeping during most of this character time.

32.5.4 Interrupts and status flags

The UART system has one interrupt vector which has three separate interrupt types. One interrupt type is associated with the transmitter for UART_S1[TDRE] and UART_S1[TC] events. Another interrupt type is associated with the receiver for RDRF, IDLE, RXEDGIF, and LBKDIF events. A third type is used for OR, NF, FE, and PF error conditions. Each of these ten interrupt sources can be separately masked by local interrupt enable masks. The flags can be polled by software when the local masks are cleared to disable generation of hardware interrupt requests.

The UART transmitter has two status flags that can optionally generate hardware interrupt requests. Transmit data register empty (UART_S1[TDRE]) indicates when there is room in the transmit data buffer to write another transmit character to UART_D. If the transmit interrupt enable (UART_C2[TIE]) bit is set, a hardware interrupt is requested when UART_S1[TDRE] is set. Transmit complete (UART_S1[TC]) indicates that the transmitter is finished transmitting all data, preamble, and break characters and is idle with TxD at the inactive level. This flag is often used in systems with modems to determine when it is safe to turn off the modem. If the transmit complete interrupt enable (UART_C2[TCIE]) bit is set, a hardware interrupt is requested when UART_S1[TC] is set. Instead of hardware interrupts, software polling may be used to monitor the UART_S1[TDRE] and UART_S1[TC] status flags if the corresponding UART_C2[TIE] or UART_C2[TCIE] local interrupt masks are cleared.

When a program detects that the receive data register is full (UART_S1[RDRF] = 1), it gets the data from the receive data register by reading UART_D. The UART_S1[RDRF] flag is cleared by reading UART_S1 while UART_S1[RDRF] is set and then reading UART_D.

When polling is used, this sequence is naturally satisfied in the normal course of the user program. If hardware interrupts are used, UART_S1 must be read in the interrupt service routine (ISR). Normally, this is done in the ISR anyway to check for receive errors, so the sequence is automatically satisfied.

The IDLE status flag includes logic that prevents it from getting set repeatedly when the RxD line remains idle for an extended period of time. IDLE is cleared by reading UART_S1 while UART_S1[IDLE] is set and then reading UART_D. After UART_S1[IDLE] has been cleared, it cannot become set again until the receiver has received at least one new character and has set UART_S1[RDRF].

If the associated error was detected in the received character that caused UART_S1[RDRF] to be set, the error flags - noise flag (UART_S1[NF]), framing error (UART_S1[FE]), and parity error flag (UART_S1[PF]) - are set at the same time as UART_S1[RDRF]. These flags are not set in overrun cases.

If UART_S1[RDRF] was already set when a new character is ready to be transferred from the receive shifter to the receive data buffer, the overrun (UART_S1[OR]) flag is set instead of the data along with any associated NF, FE, or PF condition is lost.

At any time, an active edge on the RxD serial data input pin causes the UART_S2[RXEDGIF] flag to set. The UART_S2[RXEDGIF] flag is cleared by writing a 1 to it. This function depends on the receiver being enabled (UART_C2[RE] = 1).

32.5.5 Baud rate tolerance

A transmitting device may operate at a baud rate below or above that of the receiver.

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 zero.

As the receiver samples an incoming frame, it re-synchronizes the RT clock on any valid falling edge within the frame. Resynchronization within frames will correct a misalignment between transmitter bit times and receiver bit times.

32.5.5.1 Slow data tolerance

Figure 32-20 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.

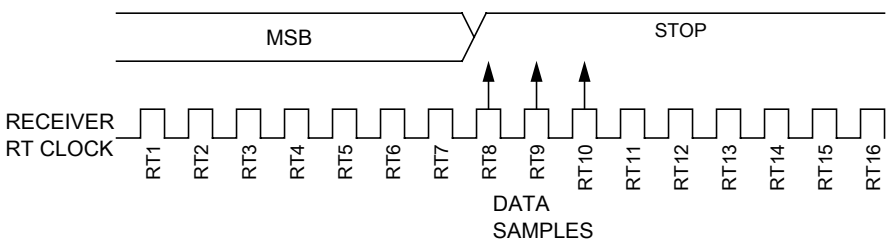


Figure 32-20. Slow data

For an 8-bit data and 1 stop bit character, data sampling of the stop bit takes the receiver 9 bit times x 16 RT cycles +10 RT cycles =154 RT cycles.

With the misaligned character shown in Figure 32-20, the receiver counts 154 RT cycles at the point when the count of the transmitting device is 9 bit times x 16 RT cycles + 3 RT cycles = 147 RT cycles.

The maximum percent difference between the receiver count and the transmitter count of a slow 8-bit data and 1 stop bit character with no errors is:

((154 - 147) / 154) x 100 = 4.54%

For a 9-bit data or 2 stop bits character, data sampling of the stop bit takes the receiver 10 bit times x 16 RT cycles + 10 RT cycles = 170 RT cycles.

With the misaligned character shown in Figure 32-20, the receiver counts 170 RT cycles at the point when the count of the transmitting device is 10 bit times x 16 RT cycles + 3 RT cycles = 163 RT cycles.

The maximum percent difference between the receiver count and the transmitter count of a slow 9-bit or 2 stop bits character with no errors is:

((170 - 163) / 170) X 100 = 4.12%

For a 9-bit data and 2 stop bit character, data sampling of the stop bit takes the receiver 11 bit times x 16 RT cycles + 10 RT cycles = 186 RT cycles.

With the misaligned character shown in Figure 32-20, the receiver counts 186 RT cycles at the point when the count of the transmitting device is 11 bit times x 16 RT cycles + 3 RT cycles = 179 RT cycles.

The maximum percent difference between the receiver count and the transmitter count of a slow 9-bit and 2 stop bits character with no errors is: ((186 - 179) / 186) X 100 = 3.76%

32.5.5.2 Fast data tolerance

Figure 32-21 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.

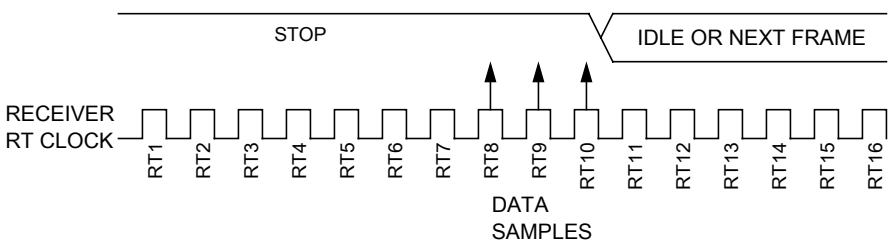


Figure 32-21. Fast data

For an 8-bit data and 1 stop bit character, data sampling of the stop bit takes the receiver 9 bit times x 16 RT cycles + 10 RT cycles = 154 RT cycles.

With the misaligned character shown in Figure 32-21, the receiver counts 154 RT cycles at the point when the count of the transmitting device is 10 bit times x 16 RT cycles = 160 RT cycles.

The maximum percent difference between the receiver count and the transmitter count of a fast 8-bit and 1 stop bit character with no errors is:

((154 - 160) / 154) x 100 = 3.90%

For a 9-bit data or 2 stop bits character, data sampling of the stop bit takes the receiver 10 bit times x 16 RT cycles + 10 RT cycles = 170 RT cycles.

With the misaligned character shown in, the receiver counts 170 RT cycles at the point when the count of the transmitting device is 11 bit times x 16 RT cycles = 176 RT cycles.

The maximum percent difference between the receiver count and the transmitter count of a fast 9-bit or 2 stop bits character with no errors is:

((170 - 176) / 170) x 100 = 3.53%

For a 9-bit data and 2 stop bits character, data sampling of the stop bit takes the receiver 11 bit times x 16 RT cycles + 10 RT cycles = 186 RT cycles.

With the misaligned character shown in, the receiver counts 186 RT cycles at the point when the count of the transmitting device is 12 bit times x 16 RT cycles = 192 RT cycles.

The maximum percent difference between the receiver count and the transmitter count of a fast 9-bit and 2 stop bits character with no errors is:

((186 - 192) / 186) x 100 = 3.23%

32.5.6 Additional UART functions

The following sections describe additional UART functions.

32.5.6.1 8- and 9-bit data modes

The UART system, transmitter and receiver, can be configured to operate in 9-bit data mode by setting UART_C1[M]. In 9-bit mode, there is a ninth data bit to the left of the most significant bit of the UART data register. For the transmit data buffer, this bit is stored in T8 in UART_C3. For the receiver, the ninth bit is held in UART_C3[R8].

For coherent writes to the transmit data buffer, write to UART_C3[T8] before writing to UART_D.

If the bit value to be transmitted as the ninth bit of a new character is the same as for the previous character, it is not necessary to write to UART_C3[T8] again. When data is transferred from the transmit data buffer to the transmit shifter, the value in UART_C3[T8] is copied at the same time data is transferred from UART_D to the shifter.

The 9-bit data mode is typically used with parity to allow eight bits of data plus the parity in the ninth bit, or it is used with address-mark wake-up so the ninth data bit can serve as the wakeup bit. In custom protocols, the ninth bit can also serve as a software-controlled marker.

32.5.6.2 Stop mode operation

During all stop modes, clocks to the UART module are halted.

No UART module registers are affected in Stop mode.

The receive input active edge detect circuit remains active in Stop mode. An active edge on the receive input brings the CPU out of Stop mode if the interrupt is not masked (UART_BDH[RXEDGIE] = 1).

Because the clocks are halted, the UART module resumes operation upon exit from stop, only in Stop mode. Software must ensure stop mode is not entered while there is a character (including preamble, break and normal data) being transmitted out of or received into the UART module, that means UART_S1[TC] =1, UART_S1[TDRE] = 1, and UART_S2[RAF] = 0 must all meet before entering stop mode.

32.5.6.3 Loop mode

When UART_C1[LOOPS] is set, the UART_C1[RSRC] bit in the same register chooses between loop mode (UART_C1[RSRC] = 0) or single-wire mode (UART_C1[RSRC] = 1). Loop mode is sometimes used to check software, independent of connections in the external system, to help isolate system problems. In this mode, the internal loop back connection from the transmitter to the receiver causes the receiver to receive characters that are sent out by the transmitter.

32.5.6.4 Single-wire operation

When UART_C1[LOOPS] is set, UART_C1[RSRC] chooses between loop mode (UART_C1[RSRC] = 0) or single-wire mode (UART_C1[RSRC] = 1). Single-wire mode implements a half-duplex serial connection. The receiver is internally connected to the transmitter output and to the TxD pin. The RxD pin is not used and reverts to a general-purpose port I/O pin.

In single-wire mode, the UART_C3[TXDIR] bit controls the direction of serial data on the TxD pin. When UART_C3[TXDIR] is cleared, the TxD pin is an input to the UART receiver and the transmitter is temporarily disconnected from the TxD pin so an external device can send serial data to the receiver. When UART_C3[TXDIR] is set, the TxD pin is an output driven by the transmitter. In single-wire mode, the transmitter output is internally connected to the receiver input and the RxD pin is not used by the UART, so it reverts to a general-purpose port I/O pin.

Chapter 33
General-Purpose Input/Output (GPIO)

33.1 Chip-specific GPIO information

33.1.1 GPIO overview

The GPIO can be accessed by the core through the crossbar/AIPS interface at 0x400F_F000 and at an aliased slot (15) at address 0x4000_F000.

33.1.2 GPIO register bits assignment

In this device, each 8-bit port pin is mapped to the 32-bit GPIO registers as described in the table.

Table 33-1. GPIOA register bits assignment

Register 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
Port pin	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	PTB4	PTB3	PTB2	PTB1	PTB0	PTA7	PTA6	PTA5	PTA4	PTA3	PTA2	PTA1	PTA0

33.2 Introduction

The general-purpose input and output (GPIO) module is accessible via the peripheral bus. 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.

33.2.1 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.

33.2.2 Modes of operation

The following table depicts different modes of operation and the behavior of the GPIO module in these modes.

Table 33-2. 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.

33.2.3 GPIO signal descriptions

Table 33-3. GPIO signal descriptions

GPIO signal descriptions	Description	I/O
PTA7–PTA0	General-purpose input/output	I/O
PTB7–PTB0	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.

33.2.3.1 Detailed signal description

Table 33-4. GPIO interface-detailed signal descriptions

Signal	I/O	Description	
PTA7–PTA0 PTB7–PTB0	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 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.

33.3 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
4000_F000	Port Data Output Register (GPIOA_PDOR)	32	R/W	0000_0000h	33.3.1/510

Table continues on the next page...

GPIO memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4000_F004	Port Set Output Register (GPIOA_PSOR)	32	W (always reads 0)	0000_0000h	33.3.2/511
4000_F008	Port Clear Output Register (GPIOA_PCOR)	32	W (always reads 0)	0000_0000h	33.3.3/511
4000_F00C	Port Toggle Output Register (GPIOA_PTOR)	32	W (always reads 0)	0000_0000h	33.3.4/512
4000_F010	Port Data Input Register (GPIOA_PDIR)	32	R	0000_0000h	33.3.5/512
4000_F014	Port Data Direction Register (GPIOA_PDDR)	32	R/W	0000_0000h	33.3.6/513
4000_F018	Port Input Disable Register (GPIOA_PIDR)	32	R/W	0000_1FFFh	33.3.7/513

33.3.1 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: 4000_F000h base + 0h offset = 4000_F000h

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.

33.3.2 Port Set Output Register (GPIOx_PSOR)

This register configures whether to set the fields of the PDOR.

Address: 4000_F000h base + 4h offset = 4000_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																															
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.

33.3.3 Port Clear Output Register (GPIOx_PCOR)

This register configures whether to clear the fields of PDOR.

Address: 4000_F000h base + 8h offset = 4000_F008h

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.

33.3.4 Port Toggle Output Register (GPIOx_PTOR)

Address: 4000_F000h base + Ch offset = 4000_F00Ch

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.

33.3.5 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: 4000_F000h base + 10h offset = 4000_F010h

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.

33.3.6 Port Data Direction Register (GPIOx_PDDR)

The PDDR configures the individual port pins for input or output.

Address: 4000_F000h base + 14h offset = 4000_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 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.

33.3.7 Port Input Disable Register (GPIOx_PIDR)

Address: 4000_F000h base + 18h offset = 4000_F018h

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	PID																															
Reset	0	0	0	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

GPIOx_PIDR field descriptions

Field	Description
PID	Port Input Disable 0 Pin is configured for General Purpose Input, provided the pin is configured for any digital function. 1 Pin is not configured as General Purpose Input.Corresponding Port Data Input Register bit will read zero.

33.4 Functional description

33.4.1 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.

33.4.2 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.

How to Reach Us:

Home Page:
freescale.com

Web Support:
freescale.com/support

Document Number WPR1516RM
Revision 2, 12/2014