Holtek 32-Bit Microcontroller with Arm® Cortex®-M0+ Music Synthesizer

HT32F0006

User Manual

Revision: V1.10 Date: March 18, 2019

www.holtek.com

Table of Contents

List of Tables

List of Figures

1 Introduction

Overview

This user manual provides detailed information including how to use the HT32F0006 device, system and bus architecture, memory organization and peripheral instructions. The target audiences for this document are software developers, application developers and hardware developers. For more information regarding pin assignment, package and electrical characteristics, please refer to the HT32F0006 datasheet.

The HT32F0006 device is a high performance and low power consumption 32-bit microcontroller based around an Arm® Cortex®-M0+ processor core. The Cortex®-M0+ is a next-generation processor core which is tightly coupled with Nested Vectored Interrupt Controller (NVIC), SysTick timer and advanced debug support.

The HT32F0006 device operates at a frequency of up to 48 MHz with a Flash accelerator to obtain maximum efficiency. It provides 128 KB of embedded Flash memory for code / data storage and up to 16 KB of embedded SRAM memory for system operation and application program usage. A variety of peripherals, such as ADC, DAC, I^2C, I^2S, USART, UART, SPI, QSPI, PDMA, GPTM, SCTM, CRC-16 / 32, RTC, WDT, USB2.0 FS, 32-channel Music Synthesizer and SW-DP (Serial Wire Debug Port), etc., are also implemented in the device. Several power saving modes provide the flexibility for maximum optimization between wakeup latency and power consumption, which is an especially important consideration in low power applications.

The above features ensure that the HT32F0006 device is suitable for use in a wide range of applications, especially in areas such as white goods application control, power monitors, alarm systems, consumer products, handheld equipment, data logging applications, motor control, fingerprint recognition and so on.

Features

- Core
  - 32-bit Arm® Cortex®-M0+ processor core
  - Up to 48 MHz operation frequency
  - 0.93 DMIPS / MHz (Dhrystone v2.1)
  - Single-cycle multiplication
  - Integrated Nested Vectored Interrupt Controller (NVIC)
  - 24-bit SysTick timer
- On-chip Memory
  - 128 KB on-chip Flash memory for instruction / data and option bytes
  - 16 KB on-chip SRAM
  - Supports multiple booting modes
- Flash Memory Controller – FMC
  - Flash accelerator to obtain maximum efficiency
  - 32-bit word programming with In System Programming Interface (ISP) and In Application Programming (IAP)
  - Flash protection capability to prevent illegal access
- Reset Control Unit – RSTCU
  - Supply supervisor
  - Power on Reset / Power Down Reset – POR / PDR
  - Brown-out Detector – BOD
  - Programmable Low Voltage Detector – LVD
- Clock Control Unit – CKCU
  - External 4 to 16 MHz crystal oscillator
  - External 32,768 Hz crystal oscillator
  - Internal 8 MHz RC oscillator trimmed to ±2 % accuracy at 3.3 V operating voltage and 25 °C operating temperature
  - Internal 32 kHz RC oscillator
  - Integrated system clock PLL
  - Independent clock divider and gating bits for peripheral clock sources
- Power Management – PWRCU
  - Single V_{DD} power supply: 2.0 V to 3.6 V
  - Integrated 1.5 V LDO regulator for CPU core, peripherals and memories power supply
  - V_{DD} power supply for RTC
  - Two power domains: V_{DD}, 1.5 V
  - Four power saving modes: Sleep, Deep-Sleep1, Deep-Sleep2, Power-Down
- External Interrupt / Event Controller – EXTI
  - Up to 16 EXTI lines with configurable trigger source and type
  - All GPIO pins can be selected as EXTI trigger source
  - Source trigger type includes high level, low level, negative edge, positive edge or both edge
  - Individual interrupt enable, wakeup enable and status bits for each EXTI line
  - Software interrupt trigger mode for each EXTI line
  - Integrated deglitch filter for short pulse blocking

- Analog to Digital Converter – ADC
  - 12-bit SAR ADC engine
  - Up to 1 Msps conversion rate
  - Up to 16 external analog input channels
- I/O Ports – GPIO
  - Up to 52 GPIOs
  - Port A, B, C, D are mapped on 16 external interrupts – EXTI
  - Almost I/O pins are configurable output driving current
- PWM Generation and Capture Timer – GPTM
  - 16-bit up, down, up / down auto-reload counter
  - Up to 4 independent channels
  - 16-bit programmable prescaler allowing counter clock frequency division by any factor between 1 and 65536
  - Input Capture function
  - Compare Match Output
  - PWM waveform generation with Edge-aligned and Center-aligned Counting Modes
  - Single Pulse Mode Output
  - Encoder interface controller with two inputs using quadrature decoder
- Single Channel PWM Generation and Capture Timers – SCTM
  - 16-bit up and auto-reload counter
  - One channel for each timer
  - 16-bit programmable prescaler allowing counter clock frequency division by any factor between 1 and 65536
  - Input Capture function
  - Compare Match Output
  - PWM waveform generation with Edge-aligned
  - Single Pulse Mode Output
- Basic Function Timer – BFTM
  - 32-bit compare / match count-up counters – No I/O control features
  - One shot mode – Counting stops after a match condition
  - Repetitive mode – Restart counter after a match condition
- Digital to Analog Converter – DAC
  - Two 16-bit high-resolution D/A converters with excellent frequency response characteristics and good power consumption for stereo audio output.
- Music Synthesis Engine (MIDI Engine) – MSE
  - Up to 32 simultaneous sounds
  - 10-bit Volume Control
  - Output sampling frequency up to 50 kHz
  - Waveform data lengths of 8 bits, 12 bits or 16 bits
  - Stereo output
  - Supports Repeat loop Play
  - Supports PDMA interface
- Watchdog Timer – WDT
  - 12-bit down-counter with a 3-bit prescaler
  - Reset event for the system

- Programmable watchdog timer window function
- Registers write protection function

■ Real Time Clock – RTC
- 32-bit up-counter with a programmable prescaler
- Alarm function
- Interrupt and Wake-up event

■ Inter-integrated Circuit – I^2C
- Supports both master and slave modes with a frequency of up to 1 MHz
- Provides an arbitration function and clock synchronization
- Supports 7-bit and 10-bit addressing modes and general call addressing
- Supports slave multi-addressing mode with maskable address

■ Serial Peripheral Interface – SPI
- Supports both master and slave mode
- Frequency of up to (f_{PCLK}/2) MHz for master mode and (f_{PCLK}/3) MHz for slave mode
- FIFO Depth: 8 levels
- Multi-master and multi-slave operation

■ Quad Serial Peripheral Interface – QSPI
- Master or slave mode
- Frequency of up to (f_{HCLK}/2) MHz for master mode and (f_{HCLK}/3) MHz for slave mode
- Programmable data frame length up to 16 bits
- FIFO Depth: 8 levels
- MSB or LSB first shift selection
- Programmable slave select high or low active polarity
- Multi-master and multi-slave operation
- Master mode supports the dual / quad output read mode of QSPI series NOR Flash
- Four error flags with individual interrupt
  – Read overrun
  – Write collision
  – Mode fault
  – Slave abort
- Supports PDMA interface

■ Universal Synchronous Asynchronous Receiver Transmitter – USART
- Supports both asynchronous and clocked synchronous serial communication modes
- Asynchronous operating baud rate clock frequency up to (f_{PCLK}/16) MHz and synchronous operating baud rate clock frequency up to (f_{PCLK}/8) MHz
- Capability of full duplex communication
- Fully programmable serial communication characteristics including: Word length, parity bit, stop bit and bit order
- Error detection: Parity, overrun and frame error
- Supports Auto hardware flow control mode – RTS, CTS
- IrDA SIR encoder and decoder
- RS485 mode with output enable control
- FIFO Depth: 8 × 9 bits for both receiver and transmitter

- Universal Asynchronous Receiver Transmitter – UART
  - Asynchronous serial communication operating baud rate clock frequency up to (f_{PCLK}/16) MHz
  - Capability of full duplex communication
  - Fully programmable serial communication characteristics including: Word length, parity bit, stop bit and bit order
  - Error detection: Parity, overrun and frame error
- Inter-IC Sound – I^2S
  - Master or slave mode
  - Mono and stereo
  - I^2S-justified, Left-justified and Right-justified mode
  - 8 / 16 / 24 / 32-bit sample size with 32-bit channel extended
  - 8 × 32-bit TX & RX FIFO with PDMA supported
  - 8-bit Fractional Clock Divider with rate control
- Divider – DIV
  - Signed / unsigned 32-bit divider
  - Operation in 8 clock cycles, load in 1 clock cycle
  - Division by zero error flag
- Cyclic Redundancy Check – CRC
  - Supports CRC16 polynomial: 0x8005, $X^{16} + X^{15} + X^2 + 1$
  - Supports CCITT CRC16 polynomial: 0x1021, $X^{16} + X^{12} + X^5 + 1$
  - Supports IEEE-802.3 CRC32 polynomial: 0x04C11DB7, $X^{32} + X^{26} + X^{23} + X^{22} + X^{16} + X^{12} + X^{11} + X^{10} + X^8 + X^7 + X^5 + X^4 + X^2 + X + 1$
  - Supports 1's complement, byte reverse & bit reverse operations on data and checksum
  - Supports byte, half-word & word data sizes
  - Programmable CRC initial seed value
  - CRC computation done in 1 AHB clock cycle for 8-bit data and 4 AHB clock cycles for 32-bit data
  - Supports PDMA to complete a CRC computation of a block of memory
- Universal Serial Bus Device Controller – USB
  - Complies with USB 2.0 full-speed (12 Mbps) specification
  - On-chip USB full-speed transceiver
  - 1 control endpoint (EP0) for control transfer
  - 3 single-buffered endpoints (EP1 ~ EP3) for bulk and interrupt transfer
  - 4 double-buffered endpoints (EP4 ~ EP7) for bulk, interrupt and isochronous transfer
  - 1,024 bytes EP_SRAM used as the endpoint data buffers
- Peripheral Direct Memory Access – PDMA
  - 6 channels with trigger source grouping
  - 8 / 16 / 32-bit width data transfer
  - Supports Linear address, circular address and fixed address modes
  - 4-level programmable channel priority
  - Auto reload mode
  - Supported trigger sources: ADC, SPI, QSPI, USART, UART, I^2C, I^2S, GPTM, MIDI Engine and software request

- Debug support
  - Serial Wire Debug Port – SW-DP
  - 4 comparators for hardware breakpoint or code patch / literal patch
  - 2 comparators for hardware watch points
- Package and Operation Temperature
  - 48 / 64-pin LQFP package types
  - Operation temperature range: -40 °C to +85 °C

Device Information

Table 1. Features and Peripheral List

Peripherals		HT32F0006
Main Flash (KB)		127.5
Option Bytes Flash (KB)		0.5
SRAM (KB)		16
Timers	GPTM	1
	SCTM	4
	BFTM	2
	RTC	1
	WDT	1
Communication	USB	1
	SPI	1
	QSPI	1
	USART	1
	UART	1
	I^2C	1
	I^2S	1
CRC-16 / 32		1
EXTI		16
12-bit ADC Number of channels		1 16 Channels
Music Synthesis Engine		1
16-bit DAC		2 Channels
GPIO		Up to 52
CPU frequency		Up to 48 MHz
Operating voltage		2.0 V ~ 3.6 V
Operating temperature		-40 °C ~ +85 °C
Package		48 / 64-pin LQFP

Block Diagram

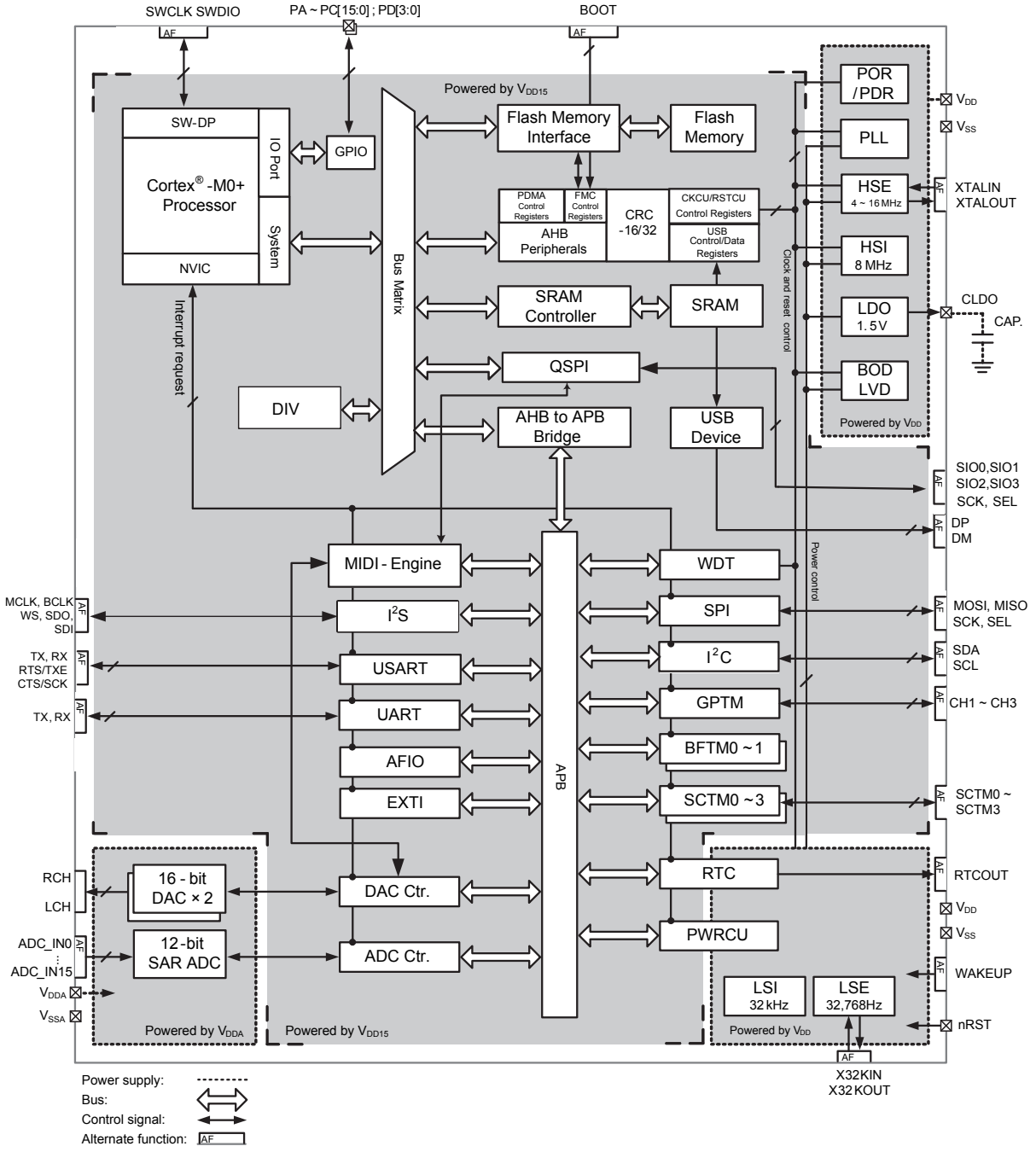


Figure 1. Block Diagram

2 Document Conventions

Unless otherwise specified, this document uses the conventions which showed as follows.

Table 2. Document Conventions

Notation	Example	Description
0x	0x5a05	The number string with a 0x prefix indicates a hexadecimal number.
0xnnnn_nnnn	0x2000_0100	32-bit Hexadecimal address or data.
b	b0101	The number string with a lowercase b prefix indicates a binary number.
NAME [n]	ADDR [5]	Specific bit of NAME. NAME can be a register or field of register. For example, ADDR [5] means bit 5 of ADDR register (field).
NAME [m:n]	ADDR [11:5]	Specific bits of NAME. NAME can be a register or field of register. For example, ADDR [11:5] means bit 11 to 5 of ADDR register (field).
X	b10X1	Don't care notation which means any value is allowed.
RW	19 18 SERDYIE | PLLRDYIE RW 0 RW 0	Software can read and write to this bit.
RO	3 2 HSIRDY | HSERDY RO 1 RO 0	Software can only read this bit. Write operation has no effort.
RC	1 0 PDF | BAK_PORF RC 0 RC 1	Software can read this bit. Read operation clears it to 0 automatically.
WC	3 2 SERDYF | PLLRDYF WC 0 WC 0	Software can read this bit or clear it by writing 1. Write 0 to it has no effort.
W0C	1 0 | MIF W0C 0 W0C 0	Software can read this bit or clear it by writing 0. Write 1 to it has no effort.
WO	31 30 | DB_CKSRC WO 0 WO 0	Software can only write to this bit. Read operation always returns 0.
Reserved	1 0 LLRDY | Reserved RO 0	Reserved bit(s) for future use. Software should not rely on the value of the reserved bit. In general case, reserved bits are set to 0. Note that reserved bit must be kept at reset value.
Word		Data length of word is 32-bit.
Half-word		Data length of half-word is 16-bit.
Byte		Data length of byte is 8-bit.

3 System Architecture

The system architecture of device that includes the Arm® Cortex®-M0+ processor, bus architecture and memory organization will be described in the following sections. The Cortex®-M0+ is a next generation processor core which offers many new features. Integrated and advanced features make the Cortex®-M0+ processor suitable for market products that require microcontrollers with high performance and low power consumption. In brief, The Cortex®-M0+ processor includes the AHB-Lite bus interface. All memory accesses of the Cortex®-M0+ processor are executed on the AHB-Lite bus according to the different purposes and the target memory spaces. The memory organization uses a Harvard architecture, pre-defined memory map and up to 4 GB of memory space, making the system flexible and extendable.

Arm® Cortex®-M0+ Processor

The Cortex®-M0+ processor is a very low gate count, highly energy efficient processor that is intended for microcontroller and deeply embedded applications that require an area optimized, low-power processor. The processor is based on the ARMv6-M architecture and supports Thumb® instruction sets; single-cycle I/O port; hardware multiplier and low latency interrupt respond time. Some system peripherals listed below are also provided by Cortex®-M0+:

- Internal Bus Matrix connected with AHB-Lite Interface, Single-cycle I/O port and Debug Accesses Port (DAP)
- Nested Vectored Interrupt Controller (NVIC)
- Optional Wakeup Interrupt Controller (WIC)
- Breakpoint and Watchpoint Unit
- Optional Memory Protection Unit (MPU)
- Serial Wire debug Port (SW-DP)
- Optional Micro Trace Buffer Interface (MTB)

The following figure shows the Cortex®-M0+ block diagram. For more information, please refer to Arm® Cortex®-M0+ Technical Reference Manual.

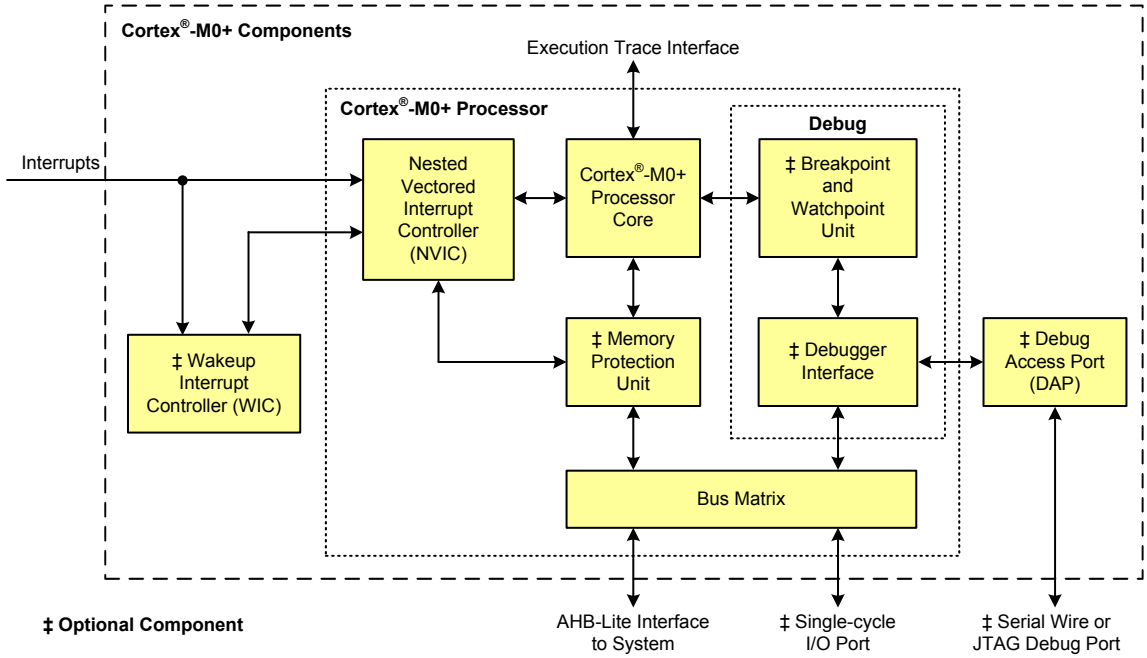


Figure 2. Cortex®-M0+ Block Diagram

Bus Architecture

The HT32F0006 device consists of one master and four slaves in the bus architecture. The Cortex®-M0+ AHB-Lite bus is the master while the internal SRAM access bus, the internal Flash memory access bus, the AHB peripherals access bus and the AHB to APB bridges are the slaves. The single 32-bit AHB-Lite system interface provides simple integration to all system regions including the internal SRAM region and the peripheral region. All of the master buses are based on 32-bit Advanced High-performance Bus-Lite (AHB-Lite) protocol. The following figure shows the bus architecture of the HT32F0006 device.

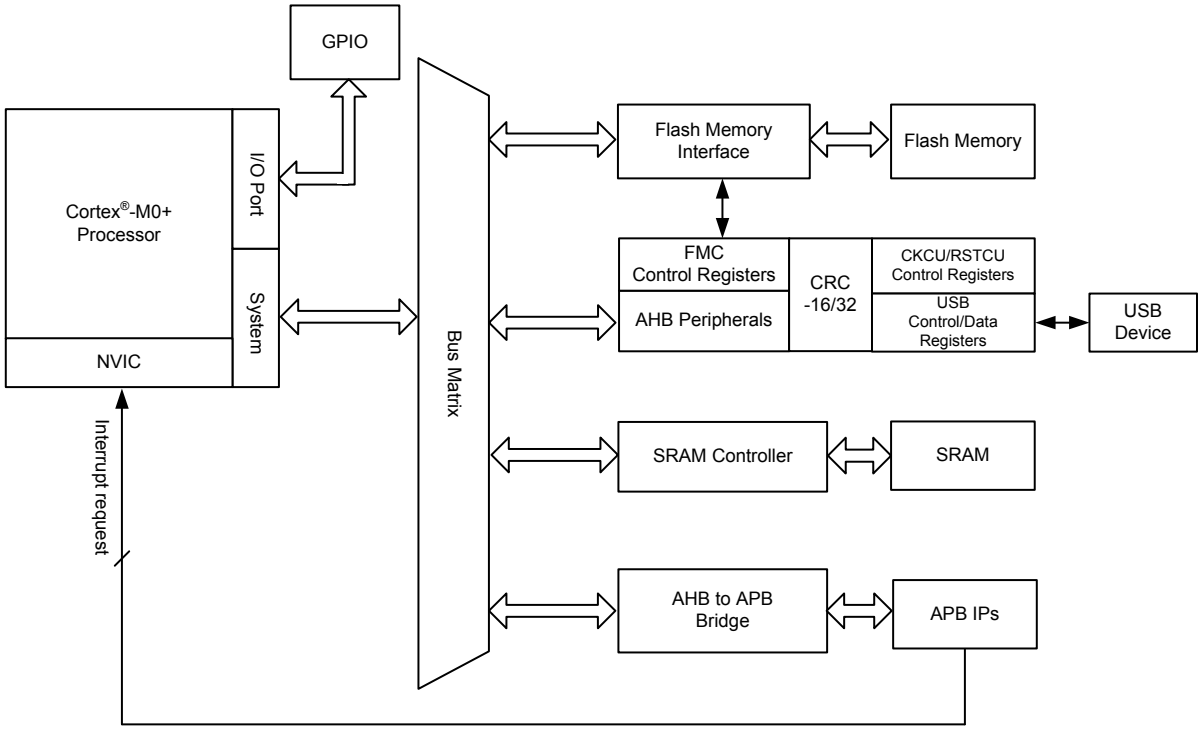


Figure 3. Bus Architecture

Memory Organization

The Arm® Cortex®-M0+ processor accesses and debug accesses share the single external interface to external AHB peripheral. The processor accesses take priority over debug accesses. The maximum address range of the Cortex®-M0+ is 4 GB since it has 32-bit bus address width. Additionally, a pre-defined memory map is provided by the Cortex®-M0+ processor to reduce the software complexity of repeated implementation of different device vendors. However, some regions are used by the Arm® Cortex®-M0+ system peripherals. Refer to the Arm® Cortex®-M0+ Technical Reference Manual for more information. The following figure shows the memory map of the HT32F0006 device, including Code, SRAM, peripheral and other pre-defined regions.

Memory Map

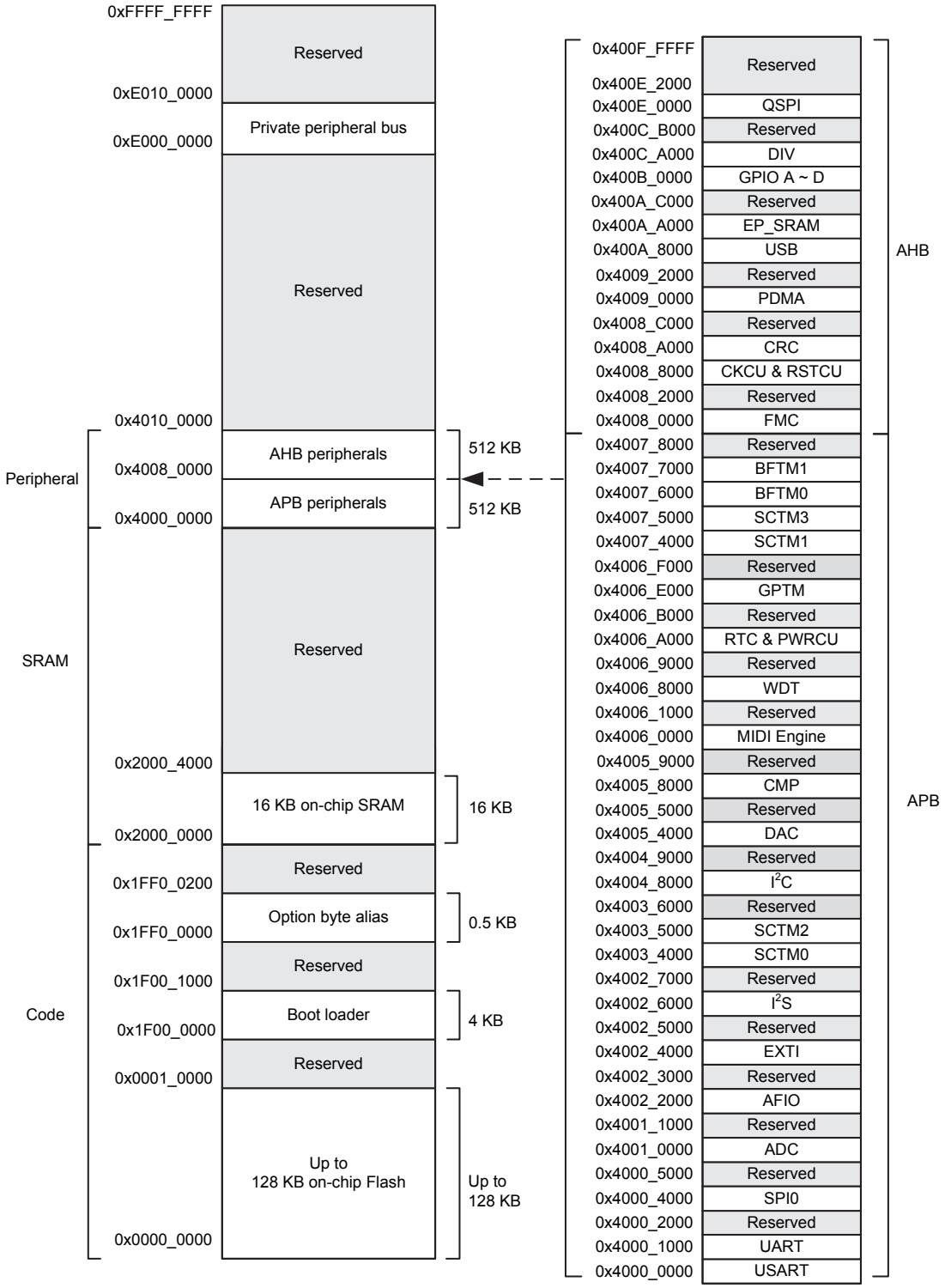


Figure 4. Memory Map

Table 3. Register Map

Start Address	End Address	Peripheral	Bus
0x4000_0000	0x4000_0FFF	USART	
0x4000_1000	0x4000_1FFF	UART	
0x4000_2000	0x4000_3FFF	Reserved	
0x4000_4000	0x4000_4FFF	SPI	
0x4000_5000	0x4000_FFFF	Reserved	
0x4001_0000	0x4001_0FFF	ADC	
0x4001_1000	0x4002_1FFF	Reserved	
0x4002_2000	0x4002_2FFF	AFIO	
0x4002_3000	0x4002_3FFF	Reserved	
0x4002_4000	0x4002_4FFF	EXTI	
0x4002_5000	0x4002_5FFF	Reserved	
0x4002_6000	0x4002_6FFF	I^2S	
0x4002_7000	0x4003_3FFF	Reserved	
0x4003_4000	0x4003_4FFF	SCTM0	
0x4003_5000	0x4003_5FFF	SCTM2	
0x4003_6000	0x4004_7FFF	Reserved	
0x4004_8000	0x4004_8FFF	I^2C	
0x4004_9000	0x4005_3FFF	Reserved	APB
0x4005_4000	0x4005_4FFF	DAC	
0x4005_5000	0x4005_7FFF	Reserved	
0x4005_8000	0x4005_8FFF	Comparator	
0x4005_9000	0x4005_FFFF	Reserved	
0x4006_0000	0x4006_0FFF	MIDI Engine	
0x4006_1000	0x4006_7FFF	Reserved	
0x4006_8000	0x4006_8FFF	WDT	
0x4006_9000	0x4006_9FFF	Reserved	
0x4006_A000	0x4006_AFFF	RTC&PWRCU	
0x4006_B000	0x4006_DFFF	Reserved	
0x4006_E000	0x4006_EFFF	GPTM	
0x4006_F000	0x4007_3FFF	Reserved	
0x4007_4000	0x4007_4FFF	SCTM1	
0x4007_5000	0x4007_5FFF	SCTM3	
0x4007_6000	0x4007_6FFF	BFTM0	
0x4007_7000	0x4007_7FFF	BFTM1	
0x4007_8000	0x4007_FFFF	Reserved	

Start Address	End Address	Peripheral	Bus
0x4008_0000	0x4008_1FFF	FMC	AHB
0x4008_2000	0x4008_7FFF	Reserved	
0x4008_8000	0x4008_9FFF	CKCU&RSTCU	
0x4008_A000	0x4008_BFFF	CRC	
0x4008_C000	0x4008_FFFF	Reserved	
0x4009_0000	0x4009_1FFF	PDMA Control Registers	
0x4009_2000	0x400A_7FFF	Reserved	
0x400A_8000	0x400A_BFFF	USB	
0x400A_C000	0x400A_FFFF	Reserved	
0x400B_0000	0x400B_1FFF	GPIOA	
0x400B_2000	0x400B_3FFF	GPIOB	
0x400B_4000	0x400B_5FFF	GPIOC	
0x400B_6000	0x400B_7FFF	GPIOD	
0x400B_8000	0x400C_9FFF	Reserved	
0x400C_A000	0x400C_AFFF	DIV	
0x400C_B000	0x400D_FFFF	Reserved	
0x400E_0000	0x400E_1FFF	QSPI	
0x400E_2000	0x400F_FFFF	Reserved	

Embedded Flash Memory

The HT32F0006 device provides a 128 KB on-chip Flash memory which is located at address 0x0000_0000. It supports bytes, half-words and words access operations. Note that the Flash memory only supports read operations for the bus access. Any write operations to the Flash memory will cause a bus fault exception. The Flash memory has 256 pages. Each page has a memory capacity of 512 bytes and can be erased independently. A 32-bit programming interface provides the capability of changing bits from 1 to 0. A data storage or firmware upgrade can be implemented using several methods such as In System Programming (ISP), In Application Programming (IAP) or In Circuit Programming (ICP). For more information, refer to the Flash Memory Controller section.

Embedded SRAM Memory

The HT32F0006 device contains an 16 KB on-chip SRAM which is located at address 0x2000_0000. It supports bytes, half-words and full words access operations.

AHB Peripherals

The address of the AHB peripherals ranges from 0x4008_0000 to 0x400F_FFFF. Some peripherals such as Clock Control Unit, Reset Control Unit and Flash Memory Controller are connected to the AHB bus directly. The AHB peripheral clocks are always enabled after a system reset. Access to registers for these peripherals can be achieved directly via the AHB bus. Note that all peripheral registers in AHB bus support only word access.

APB Peripherals

The address of APB peripherals ranges from 0x4000_0000 to 0x4007_FFFF. An APB to AHB Bridge provides access capability between the CPU and the APB peripherals. Additionally, the APB peripheral clocks are disabled after a system reset. Software must enable the peripheral clock by setting up the APBCCRn register in the Clock Control Unit before accessing the corresponding peripheral register. Note that the APB to AHB Bridge will duplicate the half-word or byte data to word width when a half-word or byte access is performed on the APB peripheral registers. In other words, the access result of a half-word or byte access on the APB peripheral register will vary depending on the data bit width of the access operation on the peripheral registers.

4 Flash Memory Controller (FMC)

Introduction

The Flash Memory Controller, FMC, provides all the necessary functions and pre-fetch buffer for the embedded on-chip Flash memory. Figure below shows the block diagram of FMC which includes programming interface, control register, pre-fetch buffer and access interface. Since the access speed of the Flash memory is slower than the CPU, a wide access interface with a pre-fetch buffer is provided to the Flash memory in order to reduce the CPU waiting timing which will cause CPU instruction execution delays. The Flash memory word programing / page erase functions are also provided for instruction / data storage.

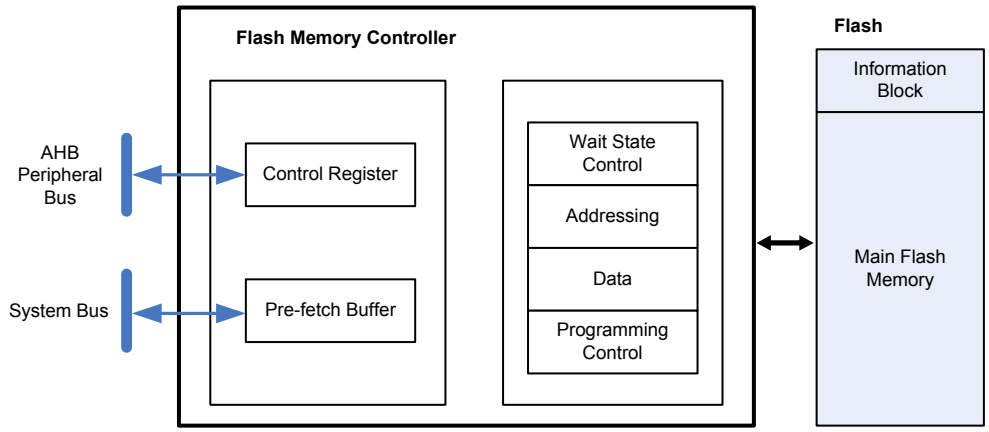


Figure 5. Flash Memory Controller Block Diagram

Features

- Up to 128 KB of on-chip Flash memory for storing instruction / data and option bytes
  - 128 KB (instruction / data + Option Byte)
- Page size of 512 bytes, totally up to 256 pages
- Wide access interface with a pre-fetch buffer to reduce instruction gaps
- Page erase and mass erase capability
- 32-bit word programming
- Interrupt function to indicate end of Flash memory operation or an error occurrence
- Flash read protection to prevent illegal code / data access
- Page erase / program protection to prevent unexpected operation

Functional Descriptions

Flash Memory Map

The following figure is the Flash memory map of the system. The address ranges from 0x0000_0000 to 0x1FFF_FFFF (0.5 GB). The address from 0x1F00_0000 to 0x1F00_0FFF is mapped to Boot Loader Block (4 KB). Additionally, the region addressed from 0x1FF0_0000 to 0x1FF0_01FF is the alias of Option Byte block (0.5 KB) which is located at the last page of the main Flash physically. The memory mapping on system view is shown as below.

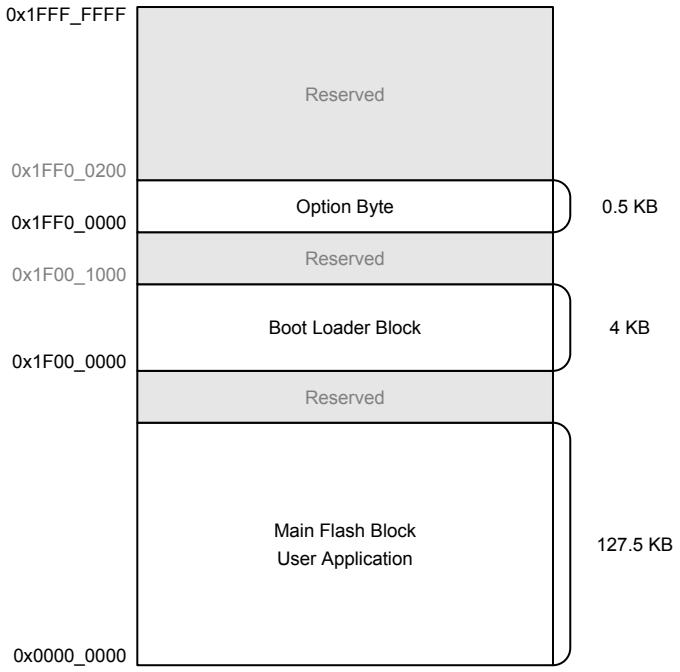


Figure 6. Flash Memory Map

Flash Memory Architecture

The Flash memory consists of up to 128 KB main Flash with 512 bytes per page and 4 KB Information Block for Boot Loader. The main Flash memory contains a total of 256 pages which can be erased individually. The following table shows the base address, size and protection setting bit of each page.

Table 4. Flash Memory and Option Byte

Block	Name	Address	Page Protection Bit	Size
Main Flash Block	Page 0	0x0000_0000 ~ 0x0000_01FF	OB_PP [0]	512 bytes
	Page 1	0x0000_0200 ~ 0x0000_03FF	OB_PP [0]	512 bytes
	Page 2	0x0000_0400 ~ 0x0000_05FF	OB_PP [1]	512 bytes
	Page 3	0x0000_0600 ~ 0x0000_07FF	OB_PP [1]	512 bytes
	⋮	⋮	⋮	⋮
	Page 252	0x0001_F800 ~ 0x0000_F9FF	OB_PP [126]	512 bytes
	Page 253	0x0001_FA00 ~ 0x0001_FBFF	OB_PP [126]	512 bytes
	Page 254	0x0001_FC00 ~ 0x0001_FDFF	OB_PP [1]	512 bytes
	Page 255 (Option Byte)	Physical: 0x0001_FE00 ~ 0x0001_FFFF Alias: 0x1FF0_0000 ~ 0x1FF0_01FF	OB_CP [1]	512 bytes
Information Block	Boot Loader	0x1F00_0000 ~ 0x1F00_0FFF	NA	4 KB

Notes: 1. The Information Block stores the boot loader – This block can not be programmed or erased by users.
2. The Option Byte is always located at the last page of the main Flash block.

Wait State Setting

When the CPU clock, HCLK, is greater than the access speed of the Flash memory, the wait state cycles must be inserted during the CPU fetch instructions or load data from Flash memory. The wait state can be changed by setting the WAIT [2:0] field of the Flash Pre-fetch Control Register, CFCR. In order to match the wait state requirement, the following two rules should be considered.

- HCLK clock is switched from low to high frequency:
  Change the wait state setting first and then switch the HCLK clock.
- HCLK clock is switched from high to low frequency:
  Switch the HCLK clock first and then change the wait state setting.

The following table shows the relationship between the wait state cycle and HCLK. The default wait state is 0 since the High Speed Internal oscillator, HSI, which operates at a frequency of 8 MHz is selected as the HCLK clock source after a system reset.

Table 5. Relationship between Wait State Cycle and HCLK

Wait State Cycle	HCLK
0	0 MHz < HCLK ≤ 24 MHz
1	20 MHz < HCLK ≤ 48 MHz

Booting Configuration

The system provides two kinds of booting modes which can be selected using the BOOT pin. The BOOT pin is sampled during the power-on reset or system reset. Once the logic value is decided, the first 4 words of vector will be remapped to the corresponding source according to the booting modes. The booting modes are shown in the following table.

Table 6. Booting Modes

Booting Mode Selection Pin	Mode	Descriptions
BOOT		
0	Boot Loader	The Vector source is Boot Loader
1	Main Flash	The Vector source is Main Flash

The Flash Vector Mapping Control Register, VMCR, is provided to change the vector remapping setting temporarily after the chip reset. The reset initial value of the VMCR register is determined by the BOOT pin status which will be sampled during the reset duration.

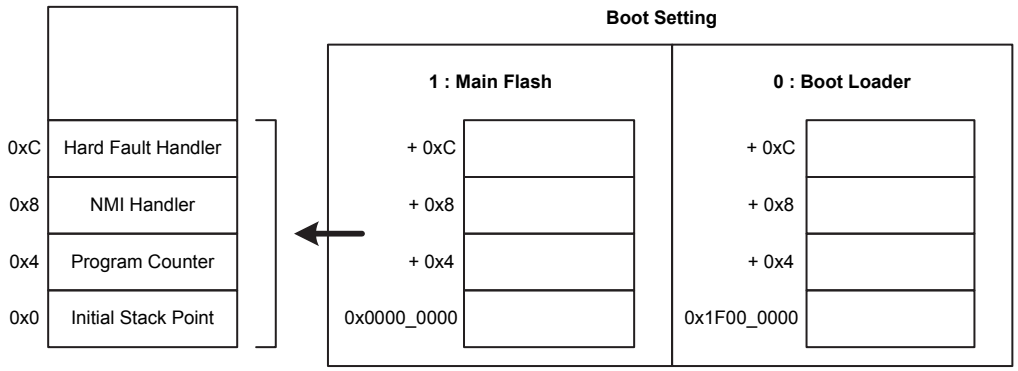


Figure 7. Vector Remapping

Page Erase

The FMC provides a page erase function which is used to initialize the contents of a Flash memory page. Each page can be erased independently without affecting the contents of other pages. The following steps show the access sequence of the register for a page erase operation.

- Check the OPCR register to confirm that no Flash memory operation is in progress (OPM [3:0] equals to 0xE or 0x6). Otherwise, wait until the previous operation has been finished.
- Write the page address to the TADR register.
- Write the page erase command to the OCMR register (Set CMD [3:0] = 0x8).
- Commit the page erase command to the FMC by setting the OPCR register (Set OPM [3:0] = 0xA).
- Wait until all the operations have been completed by checking the value of the OPCR register (OPM [3:0] equals to 0xE).
- Read and verify the page if required.

Note that a correct target page address must be confirmed. The software may run out of control if the target erase page is being used to fetch code or access data. The FMC will not provide any notification when this happens. Additionally, the page erase operation will be ignored on the protected pages. When this occurs, the OREF bit will be set by the FMC and then a Flash Operation Error interrupt will be generated if the OREIEN bit in the OIER register is set. The software can check the PPEF bit in the OISR register to detect this condition in the interrupt handler. The following figure shows the page erase operation flow.

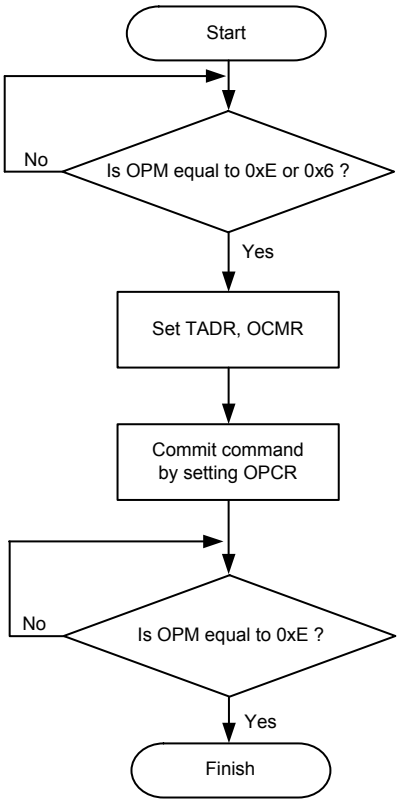


Figure 8. Page Erases Operation Flowchart

Mass Erase

The FMC provides a mass erase function which is used to initialize all the main Flash memory contents to a high state. The following steps show the mass erase operation register access sequence.

- Check the OPCR register to confirm that no Flash memory operation is in progress (OPM [3:0] equals to 0xE or 0x6). Otherwise, wait until the previous operation has been finished.
- Write the mass erase command to the OCMR register (Set CMD [3:0] = 0xA).
- Commit the mass erase command to the FMC by setting the OPCR register (Set OPM [3:0] = 0xA).
- Wait until all operations have been finished by checking the value of the OPCR register (OPM [3:0] equals to 0xE).
- Read and verify the Flash memory if required.

Since all Flash data will be reset as 0xFFFF_FFFF, the mass erase operation can be implemented by the program that runs in the SRAM or by the debugging tool that accesses the FMC registers directly. The following figure shows the mass erase operation flow.

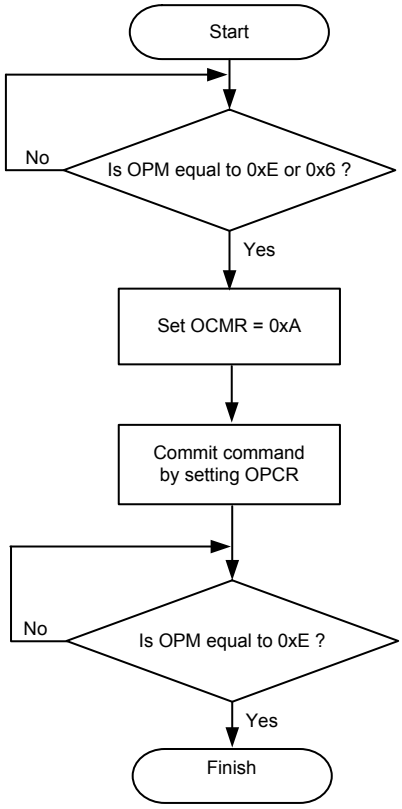


Figure 9. Mass Erases Operation Flowchart

Word Programming

The FMC provides a 32-bit word programming function which is used to modify the Flash memory contents. The following steps show the word programming operation register access sequence.

- Check the OPCR register to confirm that no Flash memory operation is in progress (OPM [3:0] equals to 0xE or 0x6). Otherwise, wait until the previous operation has been finished.
- Write the word address to the TADR register. Write the word data to the WRDR register.
- Write the word programing command to the OCMR register (Set CMD [3:0] = 0x4).
- Commit the word programing command to the FMC by setting the OPCR register (Set OPM [3:0] = 0xA).
- Wait until all operations have been finished by checking the value of the OPCR register (OPM [3:0] equals to 0xE).
- Read and verify the Flash memory if required.

Note that the word programming operation can not be successively applied to the same address twice. Successive word programming operations to the same address must be separated by a page erase operation. Additionally, the word programming operation will be ignored on protected pages. When this occurs, the OREF bit will be set by the FMC and then a Flash Operation Error interrupt will be generated if the OREIEN bit in the OIER register is set. The software can check the PPEF bit in the OISR register to detect this condition in the interrupt handler. The following figure shows the word programming operation flow.

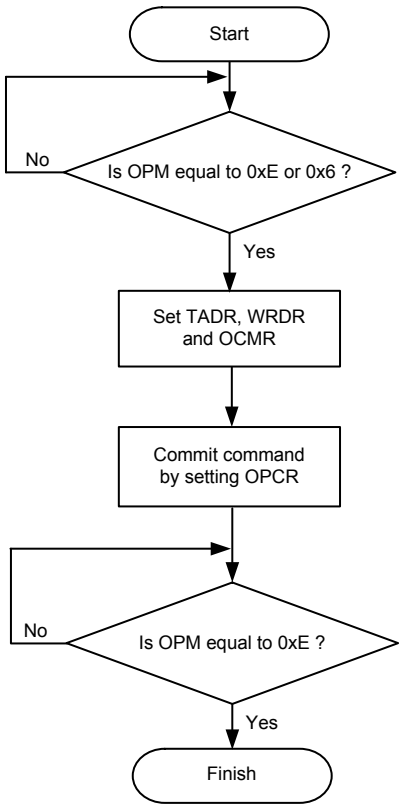


Figure 10. Word Programming Operation Flowchart

Option Byte Description

The Option Byte region can be treated as an independent Flash memory of which the base address is 0x1FF0_0000. The following table shows the functional description and the Option Byte memory map.

Table 7. Option Byte Memory Map

Option Byte	Offset	Description	Reset Value
Option Byte Base Address = 0x1FF0_0000			
OB_PP	0x000 0x004 0x008 0x00C	Flash Page Erase / Program Protection (Page 254 ~ page 0) OB_PP [n] (n = 0 ~ 127) 0: Flash Page 2n and 2n+1 Erase / Program Protection is enabled 1: Flash Page 2n and 2n+1 Erase / Program Protection is disabled	0xFFFF_FFFF 0xFFFF_FFFF 0xFFFF_FFFF 0xFFFF_FFFF
OB_CP	0x010	Flash Security Protection OB_CP [0] 0: Flash Security protection is enabled 1: Flash Security protection is disabled Option Byte Protection OB_CP [1] 0: Option Byte protection is enabled 1: Option Byte protection is disabled OB_CP [31:2] Reserved	0xFFFF_FFFF
OB_CK	0x020	Flash Option Byte Checksum OB_CK [31:0] OB_CK should be set as the sum of the 5 words Option Byte content, of which the offset address ranges from 0x000 to 0x010 (0x000 + 0x004 + 0x008 + 0x00C + 0x010), when the OB_PP or OB_CP register content is not equal to 0xFFFF_FFFF. Otherwise, both page erase / program protection and security protection will be enabled.	0xFFFF_FFFF

Page Erase / Program Protection

The FMC provides a page erase / program protection function to prevent unexpected operation of Flash memory. The page erase (CMD [3:0] = 0x8 in the OCMR register) or word programing (CMD [3:0] = 0x4) command will not be accepted by FMC on the protected pages. When the page erase or word programming command aimed at the protected pages is sent to the FMC, the PPEF bit in the OISR register will then be set by the FMC and the Flash operation error interrupt will be triggered to inform the CPU if the OREIEN bit in the OIER register is set. The page protection function can be individually enabled for each page by configuring the OB_PP registers in the Option Byte. The following table shows the access permission of the main Flash page when the page protection is enabled.

Table 8. Access Permission of Protected Main Flash Page

Mode / Operation	ISP / IAP	ICP / Debug Mode
Read	O	O
Program	X	X
Page Erase	X	X
Mass Erase	O	O

Notes: 1. The write protection is based on specific pages. The above access permission only affects the pages of which the protection function has been enabled. Other pages are not affected.
2. Main Flash page protection is configured by the OB_PP [127:0] field. Option Byte is physically located at the last page of the main Flash. Option Byte page protection is configured by the OB_CP [1] bit.
3. The page erase on the Option Byte area can disable the page protection of the main Flash.
4. The page protection of the Option Byte can only be disabled by a mass erase operation.

The following steps show the page erase / program protection procedure register access sequence.

- Check the OPCR register to confirm that no Flash memory operation is in progress (OPM [3:0] equals to 0xE or 0x6). Otherwise, wait until the previous operation has been finished.
- Write the OB_PP address to the TADR register (TADR = 0x1FF0_0000).
- Write the data, which indicates the protection function of the corresponding page is enabled or disabled, into the WRDR register (0: Enabled, 1: Disabled).
- Write the word programing command to the OCMR register. (CMD [3:0] = 0x4).
- Commit the word programing command to the FMC by setting the OPCR register (Set OPM [3:0] = 0xA).
- Wait until all operations have been finished by checking the value of the OPCR register (OPM [3:0] equals to 0xE).
- Read and verify the Option Byte if required.
- The OB_CK field in the Option Byte must be updated according to the Option Byte checksum rule.
- Apply a system reset to activate the new OB_PP setting.

Security Protection

The FMC provides a Security protection function to prevent illegal code / data access of the Flash memory. This function is useful for protecting the software / firmware from the illegal users. The function is activated by configuring the Option Byte OB_CP [0] bit. Once the function has been enabled, all the main Flash data access through ICP / Debug mode, programming and page erase operation will not be allowed except for the user's application. However, the mass erase operation will still be accepted by the FMC in order to disable this security protection function. The following table shows the access permission of the Flash memory when the security protection is enabled.

Table 9. Access Permission When Security Protection is Enabled

Mode / Operation	User application (Note 1)	ICP / Debug mode
Read	O	X (read as 0)
Program	O (Note 1)	X
Page Erase	O (Note 1)	X
Mass Erase	O	O

Notes: 1. User application means the software that is executed or booted from main Flash memory with the JTAG / SW debugger being disconnected. However, the Option Byte area and page 0 are still under protection, where the Program / Page Erase operations are not accepted.

2. The Mass erase operation can erase the Option Byte block and disable the security protection.

The following steps show the security protection procedure register access sequence:

- Check the OPCR register to confirm that no Flash memory operation is in progress (OPM [3:0] equals to 0xE or 0x6). Otherwise, wait until the previous operation has been finished.
- Write the OB_CP address to the TADR register (Set TADR = 0x1FF0_0010).
- Write the data into the WRDR register to set OB_CP [0] to 0.
- Write the word programing command to the OCMR register (Set CMD [3:0] = 0x4).
- Commit the word programing command to the FMC by setting the OPCR register (Set OPM = 0xA).
- Wait until all operations have been finished by checking the value of the OPCR register (OPM [3:0] equals to 0xE).
- Read and verify the Option Byte if required.
- The OB_CK field in the Option Byte must be updated according to the Option Byte checksum rule.
- Apply a system reset to active the new OB_CP setting.

Register Map

The following table shows the FMC registers and reset values.

Table 10. FMC Register Map

Register	Offset	Description	Reset Value
TADR	0x000	Flash Target Address Register	0x0000_0000
WRDR	0x004	Flash Write Data Register	0x0000_0000
OCMR	0x00C	Flash Operation Command Register	0x0000_0000
OPCR	0x010	Flash Operation Control Register	0x0000_000C
OIER	0x014	Flash Operation Interrupt Enable Register	0x0000_0000
OISR	0x018	Flash Operation Interrupt and Status Register	0x0001_0000
PPSR	0x020 0x024 0x028 0x02C	Flash Page Erase / Program Protection Status Register	0xXXXX_XXXX 0xXXXX_XXXX 0xXXXX_XXXX 0xXXXX_XXXX
CPSR	0x030	Flash Security Protection Status Register	0x0000_000X
VMCR	0x100	Flash Vector Mapping Control Register	0x0000_000X
MDID	0x180	Flash Manufacturer and Device ID Register	0x0376_XXXX
PNSR	0x184	Flash Page Number Status Register	0x0000_00XX
PSSR	0x188	Flash Page Size Status Register	0x0000_0200
CFCR	0x200	Flash Cache and Pre-fetch Control Register	0x0000_1011
CIDR0	0x310	Custom ID Register 0	0xXXXX_XXXX
CIDR1	0x314	Custom ID Register 1	0xXXXX_XXXX
CIDR2	0x318	Custom ID Register 2	0xXXXX_XXXX
CIDR3	0x31C	Custom ID Register 3	0xXXXX_XXXX

Note: "X" means various reset values which depend on the Device, Flash value, Option Byte value or power on reset setting.

Register Descriptions

Flash Target Address Register – TADR

This register specifies the target address of the page erase and word programming operations.

Offset: 0x000
Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	TADB							
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
	TADB							
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
	TADB							
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
	TADB							
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0

Bits	Field	Descriptions
[31:0]	TADB	Flash Target Address Bits For programming operations, the TADR register specifies the address where the data is written. Since the programming length is 32-bit, the TADR should be set as word-aligned (4 bytes). The TADB [1:0] bits will be ignored during programming operations. For page erase operations, the TADR register contains the page address which is going to be erased. Since the page size is 512-byte, the TADB [8:0] bits will be ignored in order to limit the target address as 512 Byte-aligned. For 128 KB main Flash addressing, the TADB [31:17] bits should be zero. The region of which the address ranges from 0x1FF0_0000 to 0x1FF0_01FF is the 0.5 KB Option Byte. This field for available Flash address must be under 0x1FFF_FFFF. Otherwise, the Invalid Target Address interrupt will occur if the corresponding interrupt enable bit is set.

Flash Write Data Register – WRDR

This register specifies the data to be written for programming operation.

Offset: 0x004
Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	WRDB							
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
	WRDB							
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
	WRDB							TA3
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
	WRDB							
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0

Bits	Field	Descriptions
[31:0]	WRDB	Flash Write Data Bits The data value for programming operation.

Flash Operation Command Register – OCMR

This register is used to specify the Flash operation commands that include word programing, page erase and mass erase.

Offset: 0x00C

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	22	21	20	19	18	17	16
	Reserved							
Type/Reset								
	15	14	13	12	11	10	9	8
	Reserved							
Type/Reset								
	7	6	5	4	3	2	1	0
	Reserved				CMD			
Type/Reset					RW 0	RW 0	RW 0	RW 0

Bits	Field	Descriptions
[3:0]	CMD	Flash Operation Command The following table shows the definitions of the operation command bits, CMD [3:0], which specify the Flash memory operation. If an invalid command is set and the IOCMIEN bit is set to 1, an Invalid Operation Command interrupt will occur.

CMD [3:0]	Description
0x0	Idle (default)
0x4	Word programming
0x8	Page erase
0xA	Mass erase
Others	Reserved

Flash Operation Control Register – OPCR

This register is used for controlling the command commitment and checking the status of the FMC operations.

Offset: 0x010

Reset value: 0x0000_000C

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	22	21	20	19	18	17	16
	Reserved							
Type/Reset								
	15	14	13	12	11	10	9	8
	Reserved							
Type/Reset								
	7	6	5	4	3	2	1	0
	Reserved			OPM				Reserved
Type/Reset				RW 0	RW 0	RW 0	RW 0	

Bits	Field	Descriptions
[4:1]	OPM	Operation Mode The following table shows the operation modes of the FMC. User can commit the command which is set by the OCMR register to the FMC according to the address alias setting in the TADR register. The contents of the TADR, WRDR and OCMR registers should be prepared before setting this register. After all the operations have been finished, the OPM field will be set as 0xE by the FMC hardware. The Idle mode can be set when all the operations have been finished for power saving purpose. Note that the operation status should be checked before the next operation is executed to the FMC. The contents of the TADR, WRDR, OCMR and OPCR registers should not be changed until the previous operation has been finished.

OPM [3:0]	Description
0x6	Idle (default)
0xA	Commit command to main Flash
0xE	All operation finished on main Flash
Others	Reserved

Flash Operation Interrupt Enable Register – OIER

This register is used to enable or disable the FMC interrupt function. The FMC will generate interrupts when the corresponding interrupt enable bit is set and the interrupt condition occurs.

Offset: 0x014

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	22	21	20	19	18	17	16
	Reserved							
Type/Reset								
	15	14	13	12	11	10	9	8
	Reserved							
Type/Reset								

	7	6	5	4	3	2	1	0
	Reserved			OREIEN	IOCMIEN	OBEIEN	ITADIEN	ORFIEN
Type/Reset				RW 0	RW 0	RW 0	RW 0	RW 0

Bits	Field	Descriptions
[4]	OREIEN	Operation Error Interrupt Enable 0: Operation error interrupt is disabled 1: Operation error interrupt is enabled
[3]	IOCMIEN	Invalid Operation Command Interrupt Enable 0: Invalid Operation Command interrupt is disabled 1: Invalid Operation Command interrupt is enabled
[2]	OBEIEN	Option Byte Check Sum Error Interrupt Enable 0: Option Byte Check Sum Error interrupt is disabled 1: Option Byte Check Sum Error interrupt is enabled
[1]	ITADIEN	Invalid Target Address Interrupt Enable 0: Invalid Target Address interrupt is disabled 1: Invalid Target Address interrupt is enabled
[0]	ORFIEN	Operation Finished Interrupt Enable 0: Operation Finish interrupt is disabled 1: Operation Finish interrupt is enabled

Flash Operation Interrupt and Status Register – OISR

This register indicates the FMC interrupt status which is used to check if a Flash operation has been finished or an error occurs. The status bits, bit [4:0], if set high, are available to trigger the interrupt when the corresponding enable bits in the OIER register are set high.

Offset: 0x018

Reset value: 0x0001_0000

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	22	21	20	19	18	17	16
	Reserved						PPEF	RORFF
Type/Reset							RO 0	RO 1
	15	14	13	12	11	10	9	8
	Reserved							
Type/Reset								
	7	6	5	4	3	2	1	0
	Reserved			OREF	IOCMF	OBEF	ITADF	ORFF
Type/Reset				WC 0	WC 0	WC 0	WC 0	WC 0

Bits	Field	Descriptions
[17]	PPEF	Page Erase / Program Protected Error Flag 0: Page Erase / Program Protected Error does not occur 1: Operation error occurs due to an invalid erase / program operation being applied to a protected page This bit is reset by hardware once a new flash operation command is committed.
[16]	RORFF	Raw Operation Finished Flag 0: The last flash operation command is not finished 1: The last flash operation command is finished The RORFF bit is directly connected to the Flash memory for debugging purpose.
[4]	OREF	Operation Error Flag 0: No Flash operation error occurred 1: The last flash operation is failed This bit will be set when any Flash operation error occurs such as an invalid command, program error and erase error, etc. The ORE interrupt occurs if the OREIEN bit in the OIER register is set. Reset this bit by writing 1.
[3]	IOCMF	Invalid Operation Command Flag 0: No invalid flash operation command was set 1: An invalid flash operation command has been written into the OCMR register This bit will be set high when an invalid flash operation command has been written into the OCMR register. The IOCM interrupt will be occurred if the IOCMIEN bit in the OIER register is set. Reset this bit by writing 1.

Bits	Field	Descriptions
[2]	OBEF	Option Byte Check Sum Error Flag 0: Option Byte checksum is correct 1: Option Byte checksum is incorrect This bit will be set high when the Option Byte checksum is incorrect. The OBE interrupt will occur if the OBEIEN bit in the OIER register is set. This bit is cleared to 0 by software writing 1 into it. However, the Option Byte Checksum Error Flag can not be cleared by software until the interrupt condition is cleared, which means that the Option Byte check sum value has to be correctly modified or the corresponding interrupt control is disabled. Otherwise, the interrupt will be continually generated
[1]	ITADF	Invalid Target Address Flag 0: The target address is valid 1: The target address is invalid The data in the TADR field must be in the range from 0x0000_0000 to 0x1FFF_FFFF. Otherwise, this bit will be set high and an ITAD interrupt will be generated if the ITADIEN bit in the OIER register is set. Reset this bit by writing 1.
[0]	ORFF	Operation Finished Flag 0: Operation is not finished 1: Last Flash operation is finished This bit will be set high when the last flash operation is finished. The ORF interrupt will be generated if the ORFIEN bit in the OIER register is set. Reset this bit by writing 1.

Flash Page Erase / Program Protection Status Register – PPSR

This register indicates the page protection status of the Flash page erase / program protection functions.

Offset: 0x020 (0) ~ 0x02C (3)
Reset value: 0xXXXX_XXXX

	31	30	29	28	27	26	25	24
	PPSBn							
Type/Reset	RO X	RO X	RO X	RO X	RO X	RO X	RO X	RO X
	23	22	21	20	19	18	17	16
	PPSBn							
Type/Reset	RO X	RO X	RO X	RO X	RO X	RO X	RO X	RO X
	15	14	13	12	11	10	9	8
	PPSBn							
Type/Reset	RO X	RO X	RO X	RO X	RO X	RO X	RO X	RO X
	7	6	5	4	3	2	1	0
	PPSBn							
Type/Reset	RO X	RO X	RO X	RO X	RO X	RO X	RO X	RO X

Bits	Field	Descriptions
[127:0]	PPSBn	Page Erase / Program Protection Status Bits (n = 0 ~ 127) PPSB[n] = OB_PP[n] 0: The corresponding page is protected 1: The corresponding page is not protected The content of this register is not dynamically updated and will only be reloaded from the Option Byte when any kind of reset occurs. The erase or program function of the specific page is not allowed when the corresponding bits of the PPSR registers are reset. The reset value of PPSR [127:0] is determined by the Option Byte OB_PP [127:0] bits. Since the maximum page number of the main flash is various and dependent on the chip specification. Therefore, each page erase / program protection status bit may protect one or two pages, depending upon the chip specification. Other bits of the OB_PP and PPSR registers are reserved for future usage.

Flash Security Protection Status Register – CPSR

This register indicates the Flash Memory security protection status. The content of this register is not dynamically updated and will only be reloaded by the Option Byte loader which is active when any kind of reset occurs.

Offset: 0x030

Reset value: 0xXXXX_XXXX

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	22	21	20	19	18	17	16
	Reserved							
Type/Reset								
	15	14	13	12	11	10	9	8
	Reserved							
Type/Reset								
	7	6	5	4	3	2	1	0
	Reserved						OBPSB	CPSB
Type/Reset							RO X	RO X

Bits	Field	Descriptions
[1]	OBPSB	Option Byte Page Erase / Program Protection Status Bit 0: The Option Byte page is protected 1: The Option Byte page is not protected The reset value of the OBPSB bit is determined by the Option Byte OB_CP [1] bit.
[0]	CPSB	Flash Memory Security Protection Status Bit 0: Flash Memory Security protection is enabled 1: Flash Memory Security protection is not enabled The reset value of the CPSB bit is determined by the Option Byte OB_CP [0] bit.

Flash Vector Mapping Control Register – VMCR

This register is used to control the vector mapping. The reset value of the VMCR register is determined by the external booting pin, BOOT, during the power-on reset period.

Offset 0x100

Reset value: 0x0000_000X

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	22	21	20	19	18	17	16
	Reserved							
Type/Reset								
	15	14	13	12	11	10	9	8
	Reserved							
Type/Reset								
	7	6	5	4	3	2	1	0
	Reserved						VMCB	Reserved
Type/Reset							RW	X

Bits	Field	Descriptions
[1]	VMCB	Vector Mapping Control Bit The VMCB bit is used to control the mapping source of first 4-word vectors addressed from 0x0 to 0xC. The following table shows the vector mapping setting.

BOOT	VMCB	Descriptions
Low	0	Boot Loader mode The vector mapping source is the boot loader area.
High	1	Main Flash mode The vector mapping source is the main Flash area.

The reset value of the VMCB bit is determined by the pin status of the external BOOT pin during power-on reset and system reset. The vector mapping setting can be changed temporarily by configuring the VMCB bit when the application program is executed.

Flash Manufacturer and Device ID Register – MDID

This register is used to store the manufacture ID and device part number information which can be used as the product identity.

Offset: 0x180

Reset value: 0x0376_XXXX

	31	30	29	28	27	26	25	24
	MFID							
Type/Reset	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 1	RO 1
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
	MFID							
Type/Reset	RO 0	RO 1	RO 1	RO 1	RO 0	RO 1	RO 1	RO 0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
	ChipID							
Type/Reset	RO X	RO X	RO X	RO X	RO X	RO X	RO X	RO X
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
	ChipID							
Type/Reset	RO X	RO X	RO X	RO X	RO X	RO X	RO X	RO X

Bits	Field	Descriptions
[31:16]	MFID	Manufacturer ID Read as 0x0376
[15:0]	ChipID	Chip ID The MCU device part number.

Flash Page Number Status Register – PNSR

This register is used to indicate the Flash memory page number.

Offset: 0x184
Reset value: 0x0000_00XX

	31	30	29	28	27	26	25	24
	PNSB							
Type/Reset	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
	PNSB							
Type/Reset	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
	PNSB							
Type/Reset	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
	PNSB							
Type/Reset	RO X	RO X	RO X	RO X	RO X	RO X	RO X	RO X

Bits	Field	Descriptions
[31:0]	PNSB	Flash Page Number Status Bits 0x0000_0010: Totally 16 pages for the on-chip Flash memory device 0x0000_0020: Totally 32 pages for the on-chip Flash memory device 0x0000_0040: Totally 64 pages for the on-chip Flash memory device 0x0000_0080: Totally 128 pages for the on-chip Flash memory device 0x0000_00FF: Totally 255 pages for the on-chip Flash memory device

Flash Page Size Status Register – PSSR

This register is used to indicate the page size in bytes.

Offset: 0x188
Reset value: 0x0000_0200

	31	30	29	28	27	26	25	24
	PSSB							
Type/Reset	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
	PSSB							
Type/Reset	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
	PSSB							
Type/Reset	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 1	RO 0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
	PSSB							
Type/Reset	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0

Bits	Field	Descriptions
[31:0]	PSSB	Status Bits of Flash Page Size 0x200: The page size is 512 Bytes per page 0x400: The page size is 1K Bytes per page 0x800: The page size is 2K Bytes per page

Flash Cache and Pre-fetch Control Register – CFCR

This register is used for controlling the FMC Cache and pre-fetch module.

Offset: 0x200
Reset value: 0x0000_1011

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	22	21	20	19	18	17	16
	Reserved							
Type/Reset								
	15	14	13	12	11	10	9	8
				CE	Reserved			
Type/Reset				RW 1				
	7	6	5	4	3	2	1	0
	Reserved			PFBE	Reserved	WAIT		
Type/Reset				RW 1		RW 0	RW 0	RW 1

Bits	Field	Descriptions
[12]	CE	Branch Cache Enable Bit 0: Cache is disabled 1: Cache is enabled
[4]	PFBE	Pre-fetch Buffer Enable Bit 0: Pre-fetch buffer is disabled 1: Pre-fetch buffer is enabled The pre-fetch buffer is enabled in default. When the pre-fetch buffer is disabled, the instruction and data are directly provided by the Flash memory.
[2:0]	WAIT	Flash Wait State Setting The WAIT[2:0] field is used to set the HCLK wait clock during a non-sequential address Flash access. The actual wait clock is given by (WAIT[2:0] – 1). Since a wide access interface with a pre-fetch buffer is provided, the wait state of sequential Flash access is very close to zero.

WAIT [2:0]	Wait Status	Allowed HCLK Range
001	0	0 MHz < HCLK ≤ 24 MHz
010	1	24 MHz < HCLK ≤ 48 MHz
Others	Reserved	Reserved

Custom ID Register n – CIDRn (n = 0 ~ 3)

This register is used to store the custom ID information which can be used as the custom identity.

Offset: 0x310 (0) ~ 0x31C (3)

Reset value: Various depending on Flash Manufacture Privilege Information Block.

	31	30	29	28	27	26	25	24
	CID							
Type/Reset	RO X	RO X	RO X	RO X	RO X	RO X	RO X	RO X
	23	22	21	20	19	18	17	16
	CID							
Type/Reset	RO X	RO X	RO X	RO X	RO X	RO X	RO X	RO X
	15	14	13	12	11	10	9	8
	CID							
Type/Reset	RO X	RO X	RO X	RO X	RO X	RO X	RO X	RO X
	7	6	5	4	3	2	1	0
	CID							
Type/Reset	RO X	RO X	RO X	RO X	RO X	RO X	RO X	RO X

Bits	Field	Descriptions
[31:0]	CIDn	Custom ID Read as the CIDn[31:0] (n = 0 ~ 3) field in the Custom ID registers in Flash Manufacture Privilege Block.

5 Power Control Unit (PWRCU)

Introduction

The power consumption can be regarded as one of the most important issues for many embedded system applications. Accordingly the Power Control Unit, PWRCU, provides many types of power saving modes such as Sleep, Deep-Sleep1, Deep-Sleep2 and Power-Down modes. These modes reduce the power consumption and allow the application to achieve the best trade-off between the conflicting demands of CPU operating time, speed and power consumption. The dash line in the Figure 11 indicates the power supply source of two digital power domains.

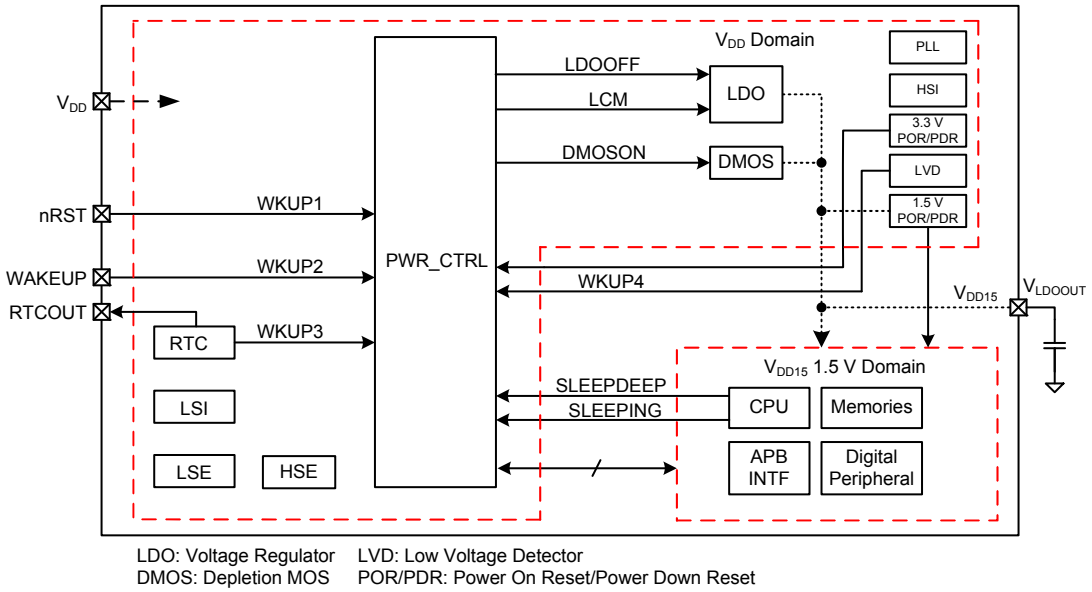


Figure 11. PWRCU Block Diagram

Features

- Two power domains: V_{DD} 3.3 V and V_{DD15} 1.5 V power domains.
- Four power saving modes: Sleep, Deep-Sleep1, Deep-Sleep2 and Power-Down modes.
- Internal Voltage regulator supplies 1.5 V voltage source.
- Additional Depletion MOS supplies 1.5 V voltage source with low leakage and low operating current.
- A power reset is generated when one of the following events occurs:
  - Power-on / Power-down reset (POR / PDR reset).
  - When exiting Power-Down mode.
  - The control bits BODEN = 1, BODRIS = 0 and the supply power $V_{DD} \leq V_{BOD}$.
- BOD Brown-out Detector can issue a system reset or an interrupt when V_{DD} power source is lower than the Brown Out Detector voltage V_{BOD}.
- LVD Low Voltage Detector can issue an interrupt or wakeup event when V_{DD} is lower than a programmable threshold voltage V_{LVD}.

Functional Descriptions

V_{DD} Power Domain

LDO Power Control

The LDO will be automatically switched off when one of the following conditions occurs:

- The Power-Down or Deep-Sleep2 mode is entered.
- The control bits BODEN = 1, BODRIS = 0 and the supply power $V_{DD} \leq V_{BOD}$.
- The supply power $V_{DD} \leq V_{PDR}$

The LDO will be automatically switched on by hardware when the supply power $V_{DD} > V_{POR}$ if any of the following conditions occurs:

- Resume operation from the power saving mode – RTC wakeup, LVD wakeup and WAKEUP pin rising edge.
- Detect a falling edge on the external reset pin (nRST).
- The control bit BODEN = 1 and the supply power $V_{DD} > V_{BOD}$.

To enter the Deep-Sleep1 mode, the PWRCU will request the LDO to operate in a low current mode, LCM. To enter the Deep-Sleep2 mode, the PWRCU will turn off the LDO and turn on the DMOS to supply an alternative 1.5 V power.

Voltage Regulator

The voltage regulator, LDO, Depletion MOS, DMOS, Low voltage Detector, LVD, High Speed Internal oscillator, HSI, Low Speed Internal RC oscillator, LSI, High Speed External Crystal oscillator, HSE, and the Low Speed External Crystal oscillator, LSE, are operated under the V_{DD} power domain. The LDO can be configured to operate in either normal mode (LDOOFF = 0, LDOLCM = 0, I_{OUT} = High current mode) or low current mode (LDOOFF = 0, LDOLCM = 1, I_{OUT} = Low current mode) to supply the 1.5 V power. An alternative 1.5 V power source is the output of the DMOS which has low static and driving current characteristics. It is controlled using the DMOSON bit in the PWRCR register. The DMOS output has weak output current and regulation capability and only operates in the Deep-Sleep2 mode for data retention purposes in the V_{DD15} power domain.

Power On Reset (POR) / Power Down Reset (PDR)

The device has an integrated POR / PDR circuitry that allows proper operation starting from / down to 2.0 V. When the device enters the Power-Down mode, the device will remain in Power-Down mode when V_{DD} is below a specified threshold V_{PDR}, without the need for an external reset circuit. For more details concerning the power on / power down reset threshold voltage, refer to the electrical characteristics of the corresponding datasheet.

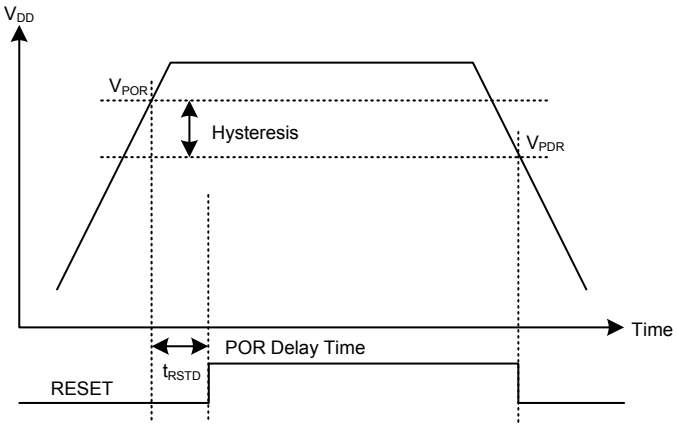


Figure 12. Power On Reset / Power Down Reset Waveform

Low Voltage Detector / Brown Out Detector

The Low Voltage Detector, LVD, can detect whether the supply voltage V_{DD} is lower than a programmable threshold voltage V_{LVD}. It is selected by the LVDS bits in the LVDCSR register. When a low voltage on the V_{DD} power pin is detected, the LVDF flag will be active and an interrupt will be generated and sent to the CPU core if the LVDEN and LVDIWEN bits in the LVDCSR register are set. For more details concerning the LVD programmable threshold voltage V_{LVD}, refer to the electrical characteristics of the corresponding datasheet.

The Brown Out Detector, BOD, is used to detect if the V_{DD} supply voltage is equal to or lower than V_{BOD}. When the BODEN bit in the LVDCSR register is set to 1 and the V_{DD} supply voltage is lower than V_{BOD} then the BODF flag is active. The PWRCU will regard this as a power down reset situation and then immediately disable the internal LDO regulator when the BODRIS bit is cleared to 0 or issue an interrupt to notify the CPU to execute a power down procedure when the BODRIS bit is set to 1. For more details concerning the Brown Out Detector voltage V_{BOD}, refer to the electrical characteristics of the corresponding datasheet.

High Speed Internal Oscillator

The High Speed Internal Oscillator, HSI, is located in the V_{DD} power domain. When exiting from the Deep-Sleep mode, the HSI clock will be configured as the system clock for a certain period by setting the PSRCEN bit to 1 This bit is located in the Global Clock Control Register, GCCR, in the Clock Control Unit, CKCU. The system clock will not be switched back to the original clock source used before entering the Deep-Sleep mode until the original clock source, which may be either sourced from the PLL or HSE stabilizes. Also the system will force the HSI oscillator to be the system clock after a wake up from Power-Down mode since a 1.5 V power on reset will occur.

High Speed External Oscillator

The High Speed External Oscillator, HSE, is located in the V_{DD} power domain. The HSE crystal oscillator can be switched on or off using the HSEEN bit in the Global Clock Control Register (GCCR). The HSE clock can then be used directly as the system clock source or be used as the PLL input clock.

LSE, LSI and RTC

The Real Time Clock Timer clock source can be derived from either the Low Speed Internal RC oscillator, LSI, or the Low Speed External Crystal oscillator, LSE. Before entering the power saving mode by executing WFI / WFE instruction, the MCU needs to setup the compare register with an expected wakeup time and enable the wakeup function to achieve the RTC timer wakeup event. After entering the power saving mode for a certain amount of time, the Compare Match flag, CMFLAG, will be asserted to wakeup the device when the compare match event occurs. The details of the RTC configuration for wakeup timer will be described in the RTC chapter.

Isolation Cells

When the device resumes operation from the 1.5 V power, either by Hardware or Software, access to the RTC registers in the V_{DD} power domain are disabled by the isolation cells which protect these registers against possible parasitic write accesses. To resume access operations, users must disable these isolation cells by setting the VDDISO bit to 1 in the LPCR register of the Clock Control Unit.

1.5 V Power Domain

The main functions that include the APB interface for the V_{DD} domain, CPU core logic, AHB / APB peripherals and memories and so on are located in this power domain. Once the 1.5 V is powered up, the POR will generate a reset sequence on 1.5 V power domain. Subsequently, to enter the expected power saving mode, the associated control bits including the LDOOFF, DMOSON and LDOLCM bits must be configured. Then, once a WFI or WFE instruction is executed, the device will enter an expected power saving mode which will be discussed in the following section.

Operation Modes

Run Mode

In the Run mode, the system operates with full functions and all power domains are active. There are two ways to reduce the power consumption in this mode. The first is to slow down the system clock by setting the AHBPRE field in the CKCU AHBCFGR register, and the second is to turn off the unused peripherals clock by setting the APBCCR0 and APBCCR1 registers or slow down peripherals clock by setting the APBPCSR0 and APBPCSR1 registers to meet the application requirement. Reducing the system clock speed before entering the sleep mode will also help to minimize power consumption.

Additionally, there are several power saving modes to provide maximum optimization between device performance and power consumption.

Table 11. Operation Mode Definitions

Mode Name	Hardware Action
Run	After system reset, CPU fetches instructions to execute.
Sleep	1.CPU clock will be stopped. 2.Peripherals, Flash and SRAM clocks can be stopped by setting.
Deep-Sleep1~2	1.Stop all clocks in the 1.5 V power domain. 2.Disable HSI, HSE and PLL. 3.Turning on the LDO low current mode or DMOS to reduce the 1.5 V power domain current.
Power-Down	Shut down the 1.5 V power domain

Sleep Mode

By default, only the CPU clock will be stopped in the Sleep mode. Clearing the FMCEN or SRAMEN bit in the CKCU AHBCCR register to 0 will have the effect of stopping the Flash clock or SRAM clock after the system enters the Sleep mode. If it is not necessary for the CPU to access the Flash memory and SRAM in the Sleep mode, it is recommended to clear the FMCEN and SRAMEN bits in the AHBCCR register to minimize power consumption. To enter the Sleep mode, it is only CPU executes a WFI or WFE instruction and lets the SLEEPDEEP signal to 0. The system will exit from the Sleep mode via any interrupt or event trigger. The accompanying table provides more information about the power saving modes.

Table 12. Enter / Exit Power Saving Modes

Mode	Mode Entry				Mode Exit
	CPU Instruction	CPU SLEEPDEEP	LDOOFF	DMOSON	
Sleep	WFI or WFE (Takes effect)	0	X	X	WFI: Any interrupt WFE: Any wakeup event(1) or Any interrupt (NVIC on) or Any interrupt with SEVONPEND = 1 (NVIC off)
Deep-Sleep1		1	0	0	Any EXTI in event mode or RTC wakeup or LVD wakeup(2) or WAKEUP pin rising edge or USB resume(3)
Deep-Sleep2		1	X	1	RTC wakeup or LVD wakeup(2) or WAKEUP pin rising edge
Power-Down		1	1	0	RTC wakeup or LVD wakeup(2) or WAKEUP pin rising edge or External reset (nRST)

Notes: 1. Wakeup event means EXTI line in event mode, RTC, LVD and WAKEUP pin rising edge.

2. If the system allows the LVD activity to wake it up after the system has entered the power saving mode, the LVDEWEN and LVDEN bits in the LVDCSR register must be set to 1 to make sure that the system can be woken up by an LVD event and then the LDO regulator can be turned on when system is woken up from the Deep-Sleep2 mode and Power-Down mode.

Deep-Sleep Mode

To enter Deep-Sleep mode, configure the registers as shown in the preceding table and execute the WFI or WFE instruction. In the Deep-Sleep mode, all clocks including PLL and high speed oscillators, known as HSI and HSE, will be stopped. In addition, Deep-Sleep1 turns the LDO into

low current mode while Deep-Sleep2 turns off the LDO and uses a DMOS to keep 1.5 V power. Once the PWRCU receives a wakeup event or an interrupt as shown in the preceding Mode-Exiting table, the LDO will then operate in normal mode and the high speed oscillator will be enabled. Finally, the CPU will return to Run mode to handle the wakeup interrupt if required. A Low Voltage Detection also can be regarded as a wakeup event if the corresponding wakeup control bit LVDEWEN in the LVDCSR register is enabled. The last wakeup event is a transition from low to high on the external WAKEUP pin sent to the PWRCU to resume from Deep-Sleep mode. During the Deep-Sleep mode, retaining the register and memory contents will shorten the wakeup latency.

Power-Down Mode

The Power-Down mode is derived from the Deep-Sleep mode of the CPU together with the additional control bits LDOOFF and DMOSON. To enter the Power-Down mode, users can configure the registers shown in the preceding Mode-Entering table and execute the WFI or WFE instruction. An RTC wakeup trigger event, an LVD wakeup, a low to high transition on the external WAKEUP pin or an external reset (nRST) signal will force the MCU out of the Power-Down mode. In the Power-Down mode, the 1.5 V power supply will be turned off. The remaining active power supplies are the 3.3 V power (V_{DD} / V_{DDA}).

After a system reset, the PORSTF bit in the RSTCU GRSR register, the PDF and PORF bits in the PWRSR register should be checked by software to confirm if the device is being resumed from the Power-Down mode by a power on reset or other reset events (nRST, WDT, …). If the device has entered the Power-Down mode under the correct firmware procedure, then the PDF bit will be set. The system information could be saved in the VDD power domain registers and be retrieved when the 1.5 V power domain is powered on again. More information about the PDF and PORF bits in the PWRSR register and PORSTF bit in the RSTCU GRSR register is shown in the following table.

Table 13. Power Status After System Reset

PORF	PDF	PORSTF	Description
1	0	1	Power-up for the first time after the V_{DD} power domain is reset: Power on reset when V_{DD} is applied for the first time or executing software reset command on the V_{DD} domain.
0	0	1	Restart from unexpected loss of the 1.5 V power or other reset (nRST, WDT, …)
0	1	1	Restart from the Power-Down mode.
1	1	x	Reserved

Register Map

The following table shows the PWRCU registers and reset values. Note all the registers in this unit are located in the V_{DD} power domain.

Table 14. PWRCU Register Map

Register	Offset	Description	Reset Value
PWRSR	0x100	Power Control Status Register	0x0000_0001
PWRCR	0x104	Power Control Register	0x0000_0000
PWRTEST	0x108	V_{DD} Power Domain Test Register	0x0000_0027
LVDCSR	0x110	Low Voltage / Brown Out Detect Control and Status Register	0x0000_0000

Register Descriptions

Power Control Status Register – PWRSR

This register indicates power control status.

Offset: 0x100

Reset value: 0x0000_0001 (Reset only by V_{DD} domain power on reset)

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	22	21	20	19	18	17	16
	Reserved							
Type/Reset								
	15	14	13	12	11	10	9	8
	Reserved							WUPF
Type/Reset								RC 0
	7	6	5	4	3	2	1	0
	Reserved						PDF	PORF
Type/Reset							RC 0	RC 1

Bits	Field	Descriptions
[8]	WUPF	External WAKEUP Pin Flag 0: The Wakeup pin is not asserted 1: The Wakeup pin is asserted This bit is set by hardware when the WAKEUP pin asserts and is cleared by software read. Software should read this bit to clear it after a system wake up from the power saving mode.
[1]	PDF	Power Down Flag 0: Wakeup from abnormal V_{DD15} shutdown (Loss of V_{DD15} is unexpected) 1: Wakeup from Power-Down mode. The loss of V_{DD15} is under expectation This bit is set by hardware when the system has successfully entered the Power-Down mode This bit is cleared by software read.
[0]	PORF	Power On Reset Flag 0: V_{DD} Power Domain reset does not occur 1: V_{DD} Power Domain reset occurs This bit is set by hardware when V_{DD} power on reset occurs, either a hardware power on reset or software reset. The bit is cleared by software read. This bit must be cleared after the system is first powered on, otherwise it will be impossible to detect when a V_{DD} Power Domain reset has been triggered. When this bit is read as 1, a read software loop must be implemented until the bit returns again to 0. This software loop is necessary to confirm that the V_{DD} Power Domain is ready for access.

Power Control Register – PWRCR

This register provides power control bits for the different kinds of power saving modes.

Offset: 0x104

Reset value: 0x0000_0000 (Reset only by V_{DD} domain power on reset)

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	22	21	20	19	18	17	16
	Reserved							
Type/Reset								
	15	14	13	12	11	10	9	8
	DMOSSTS	Reserved		V15RDYSC	Reserved		WUPIEN	WUPEN
Type/Reset	RO 0			RW 0			RW 0	RW 0
	7	6	5	4	3	2	1	0
	DMOSON	Reserved			LDOOFF	LDOLCM	Reserved	PWRST
Type/Reset	RW 0				RW 0	RW 0		WO 0

Bits	Field	Descriptions
[15]	DMOSSTS	Depletion MOS Status This bit is set to 1 if the DMOSON bit in this register has been set to 1. This bit is cleared to 0 if the DMOSON bit has been set to 0 or if a POR / PDR reset occurred.
[12]	V15RDYSC	V_{DD15} Ready Source Selection. 0: VDDISO bit in the LPCR register located in the CKCU 1: V_{DD15} POR Setting this bit to determine what control signal of isolation cells is used to disable the isolation function of the V_{DD15} to V_{DD} power domain level shifter.
[9]	WUPIEN	External WAKEUP Pin Interrupt Enable 0: Disable WAKEUP pin interrupt function 1: Enable WAKEUP pin interrupt function The software can set the WUPIEN bit to 1 to assert the WKUP interrupt in the NVIC unit when both the WUPEN and WUPF bits are set to1.

Bits	Field	Descriptions
[8]	WUPEN	External WAKEUP Pin Enable 0: Disable WAKEUP pin function 1: Enable WAKEUP pin function The Software can set the WUPEN bit as 1 to enable the WAKEUP pin function before entering the power saving mode. When WUPEN = 1, a rising edge on the WAKEUP pin wakes up the system from the power saving mode. As the WAKEUP pin is active high, this bit will set an input pull down mode when the bit is high. The WAKEUP pin alternate function should first be selected by configuring the PBCFG12 bit field in the GPBCFGHR register to 0xF before the WAKEUP pin is used. The corresponding pull-up function on the WAKEUP pin should also be disabled by clearing the PBPU[12] bit in the PBPUR register to 0 while the pull-down function should be enabled by setting the PBPD[12] bit in the PBPDR register to 1. Note: This bit is reset by a V_{DD} Power Domain reset. Because this bit is located in the V_{DD} Power Domain, after reset activity there will be a delay until the bit is active. The bit will not be active until the system reset finished and the V_{DD} Power Domain ISO signal has been disabled. This means that the bit cannot be immediately set by software after a system reset finished and the V_{DD} Power domain ISO signal disabled. The delay time needed is a minimum of three 32 KHz clock periods until the bit reset activity has finished.
[7]	DMOSON	DMOS Control 0: DMOS is OFF 1: DMOS is ON A DMOS is implemented to provide an alternative voltage source for the 1.5 V power domain when the CPU enters the Deep-Sleep mode (SLEEPDEEP = 1). The control bit DMOSON is set by software and cleared by software or V_{DD} power domain reset. If the DMOSON bit is set to 1, the LDO will automatically be turned off when the CPU enters the Deep-Sleep mode.
[3]	LDOOFF	LDO Operating Mode Control 0: The LDO operates in a low current mode when CPU enters the Deep-Sleep mode (SLEEPDEEP = 1). The V_{DD15} power is available. 1: The LDO is turned off when the CPU enters the Deep-Sleep mode (SLEEPDEEP = 1). The V_{DD15} power is not available. Note: This bit is only available when the DMOSON bit is cleared to 0.
[2]	LDOLCM	LDO Low Current Mode 0: The LDO is operated in normal current mode 1: The LDO is operated in low current mode Note: This bit is only available when CPU is in the run mode. The LDO output current capability will be limited at 10mA below and lower static current when the LDOLCM bit is set. It is suitable for CPU which is operated at lower speed system clock to get a lower current consumption. This bit will be cleared to 0 when the LDO is powered down or V_{DD} power domain is reset.
[0]	PWRST	V_{DD} Power Domain Software Reset 0: No action 1: V_{DD} Power Domain Software Reset is activated When this bit is set high, it will reset all the related RTC and PWRCU registers.

V_{DD} Power Domain Test Register – PWRTEST

This register specifies a read-only value for the software to recognize whether V_{DD} Power Domain is ready for access.

Offset: 0x108

Reset value: 0x0000_0027

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	22	21	20	19	18	17	16
	Reserved							
Type/Reset								
	15	14	13	12	11	10	9	8
	Reserved							
Type/Reset								
	7	6	5	4	3	2	1	0
	PWRTEST							
Type/Reset	RO 0	RO 0	RO 1	RO 0	RO 0	RO 1	RO 1	RO 1

Bits	Field	Descriptions
[7:0]	PWRTEST	V_{DD} Power Domain Test Bits A constant 0x27 will be read when the V_{DD} Power Domain is ready for CPU access.

Low Voltage / Brown Out Detect Control and Status Register – LVDCSR

This register specifies flags, enable bits and option bits for low voltage detector.

Offset: 0x110

Reset value: 0x0000_0000 (Reset only by V_{DD} domain power on reset)

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	22	21	20	19	18	17	16
	Reserved	LVDS [2]	LVDEWEN	LVDIWEN	LVDF		LVDS [1:0]	LVDEN
Type/Reset		RW 0	RW 0	RW 0	RO 0	RW 0	RW 0	RW 0
	15	14	13	12	11	10	9	8
	Reserved							
Type/Reset								
	7	6	5	4	3	2	1	0
	Reserved				BODF	Reserved	BODRIS	BODEN
Type/Reset					RO 0		RW 0	RW 0

Bits	Field	Descriptions
[21]	LVDEWEN	LVD Event Wakeup Enable 0: LVD event wakeup is disabled 1: LVD event wakeup is enabled Setting this bit to 1 will enable the LVD event wakeup function to wake up the system when a LVD condition occurs which result in the LVDF bit being asserted. If the system requires to be waked up from the Deep-Sleep or Power-Down mode by an LVD condition, this bit must be set to 1.
[20]	LVDIWEN	LVD Interrupt Wakeup Enable 0: LVD interrupt wakeup is disabled 1: LVD interrupt wakeup is enabled Setting this bit to 1 will enable the LVD interrupt function. When an LVD condition occurs and the LVDIWEN bit is set to 1, an LVD interrupt will be generated and sent to the CPU NVIC unit.
[19]	LVDF	Low Voltage Detect Status Flag 0: V_{DD} is higher than the specific voltage level 1: V_{DD} is equal to or lower than the specific voltage level When the LVD condition occurs, the LVDF flag will be asserted. When the LVDF flag is asserted, an LVD interrupt will be generated for CPU if the LVDIWEN bit is set to 1. However, if the LVDEWEN bit is set to 1 and the LVDIWEN bit is cleared to 0, only a LVD event will be generated rather than an LVD interrupt when the LVDF flag is asserted.
[22], [18:17]	LVDS [2:0]	Low Voltage Detect Level Selection For more details concerning the LVD programmable threshold voltage, refer to the electrical characteristics of the corresponding datasheet.

Bits	Field	Descriptions
[16]	LVDEN	Low Voltage Detect Enable 0: Disable Low Voltage Detect 1: Enable Low Voltage Detect Setting this bit to 1 will generate an LVD event when the VDD power is equal to or lower than the voltage set by LVDS bits. Therefore when the LVD function is enabled before the system is into the Deep-Sleep2 (DMOS is turn on and LDO is power down) or Power-Down mode (DMOS and LDO is power down), the LVDEWEN bit has to be enabled to avoid the LDO does not activate in the meantime when the CPU is woken up by the low voltage detection activity.
[3]	BODF	Brown Out Detect Flag 0: $V_{DD} > V_{BOD}$ 1: $V_{DD} \leq V_{BOD}$
[1]	BODRIS	BOD Reset or Interrupt Selection 0: Reset the whole chip 1: Generate Interrupt
[0]	BODEN	Brown Out Detector Enable 0: Disable Brown Out Detector 1: Enable Brown Out Detector

6 Clock Control Unit (CKCU)

Introduction

The Clock Control unit (CKCU) provides functions of high speed internal RC oscillator (HSI), High speed external crystal oscillator (HSE), Low speed internal RC oscillator (LSI), Low speed external crystal oscillator (LSE), Phase Lock Loop (PLL), HSE clock monitor, clock prescaler, clock multiplexer and clock gating. The clock of AHB, APB and CPU are derived from system clock (CK_SYS) which can come from HSI, HSE, LSI, LSE or PLL. Watchdog Timer and Real Time Clock (RTC) use either LSI or LSE as their clock source. The maximum operating frequency of system clock f_{CK_AHB} can be up to 48 MHz.

A variety of internal clocks can also be wired out though CKOUT for debugging purpose. The clock monitor can be used to get clock failure detection of HSE. Once the clock of HSE does not function (could be broken down or removed or etc), CKCU will force to switch the system clock source to HSI clock to prevent system halt.

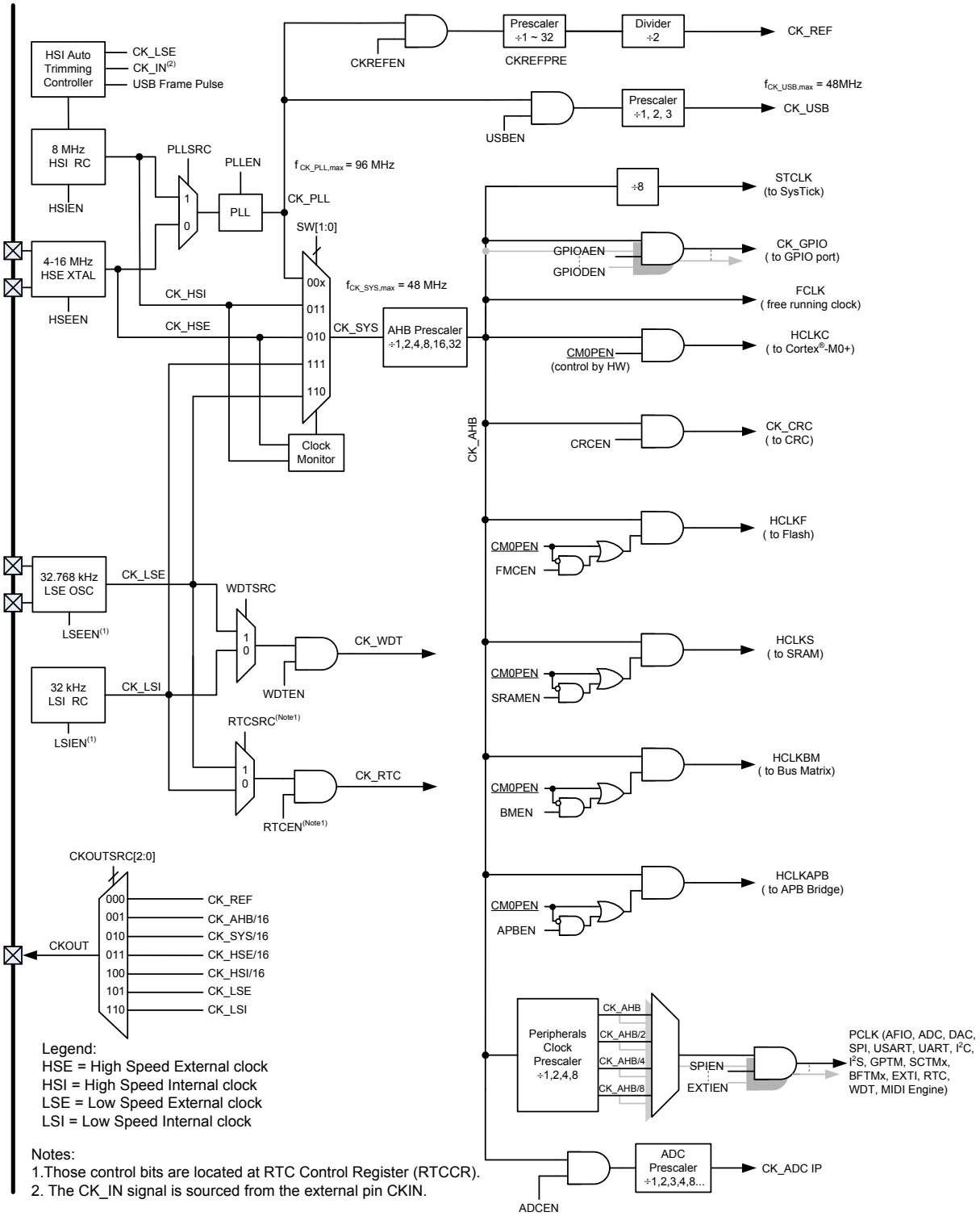


Figure 13. CKCU Block Diagram

Features

- 4 ~ 16 MHz external crystal oscillator (HSE)
- Internal 8 MHz RC oscillator (HSI) with configuration option calibration and custom trimming capability.
- PLL with selectable clock source (from HSE or HSI) for system clock.
- 32,768 Hz external crystal oscillator (LSE) for Watchdog Timer, RTC or system clock.
- Internal 32 kHz RC oscillator (LSI) for Watchdog Timer, RTC or system clock.
- HSE clock monitor

Function Descriptions

High Speed External Crystal Oscillator – HSE

The high speed external 4 to 16 MHz crystal oscillator (HSE) produces a highly accurate clock source to the system clock. The related hardware configuration is shown in the following figure. The crystal with specific frequency must be placed across the two HSE pins (XTALIN / XTALOUT) and the external components such as resistors and capacitors are necessary to make it oscillate properly.

The following guidelines are provided to improve the stability of the crystal circuit PCB layout.

- The crystal oscillator should be located as close as possible to the MCU so that the trace lengths are kept as short as possible to reduce any parasitic capacitances.
- Shield any lines in the vicinity of the crystal by using a ground plane to isolate signals and reduce noise.
- Keep frequently switching signal lines away from the crystal area to prevent crosstalk.

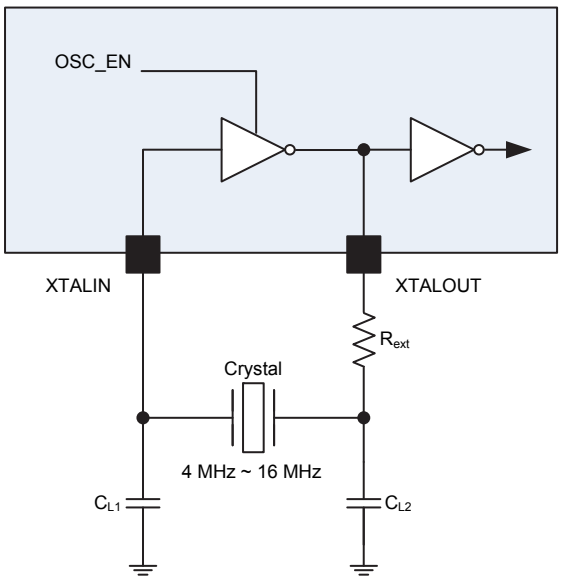


Figure 14. External Crystal, Ceramic and Resonators for HSE

The HSE crystal oscillator can be switched on or off using the HSEEN bit in the Global Clock Control Register (GCCR). The HSERDY flag in the Global Clock Status Register (GCSR) will indicate if the high-speed external crystal oscillator is stable. While switching on the HSE, the HSE clock will still not be released until this HSERDY bit is set by the hardware. The specific delay period is well-known as "Start-up time". The HSE clock can then be used directly as the system clock source or be used as the PLL input clock.

High Speed Internal RC Oscillator – HSI

The high speed internal 8 MHz RC oscillator (HSI) is the default selection of clock source for the CPU when the device is powered up. The HSI RC oscillator provides a clock source in a lower cost because no external components are required. The HSI RC oscillator can be switched on or off using the HSIEN bit in the Global Clock Control Register (GCCR). The HSIRDY flag in the Global Clock Status Register (GCSR) will indicate if the internal RC oscillator is stable. The start-up time of HSI is shorter than the HSE crystal oscillator. The HSI clock can also be used as the PLL input clock.

The accuracy of the frequency of the high speed internal RC oscillator HSI can be calibrated via the configuration options, but it is still less accurate than the HSE crystal oscillator. The applications, the environments and the cost will determine the use of the oscillators.

Software could configure PSRCEN bit (Power Saving Wakeup RC Clock Enable) to 1 to force HSI clock to be system clock when wake-up from Deep-Sleep1/2 or Power-Down mode. Subsequently, the system clock is back to the original clock source (HSE or PLL) if the original clock source ready flag is asserted. This function can reduce the wakeup time when using HSE or PLL as system clock.

Auto Trimming of High Speed Internal RC Oscillator – HSI

The frequency accuracy of the high speed internal RC oscillator HSI can vary from one chip to another due to manufacturing process variations, this is why each device is factory calibrated by HOLTEK for ±2 % accuracy at V_{DD} = 3.3 V and T_A = 25 °C. But the accuracy is not enough for some applications and environments requirement. Therefore, this device provides the trimming mechanism for HSI frequency calibration using a more accurate external reference clock. The detailed block diagram is shown as Figure 15.

After reset, the factory trimming value is loaded in the HSICOARSE[4:0] and HSIFINE[7:0] bits in the HSI Control Register (HSICR). The HSI frequency accuracy may be affected by voltage or temperature variations. If the application has to be driven by a more accurate HSI frequency, users can manually trim the HSI frequency using the HSIFINE[7:0] bits in the HSI Control Register (HSICR) or automatically adjust the HSI frequency using the Auto Trimming Controller (ATC) together with an external reference clock in the application. The reference clock can be provided from the following clock sources:

- 32,768 Hz low speed external crystal or ceramic resonator oscillator LSE output clock
- 1 kHz USB frame pulse
- External pin (CKIN) with 1 kHz pulse

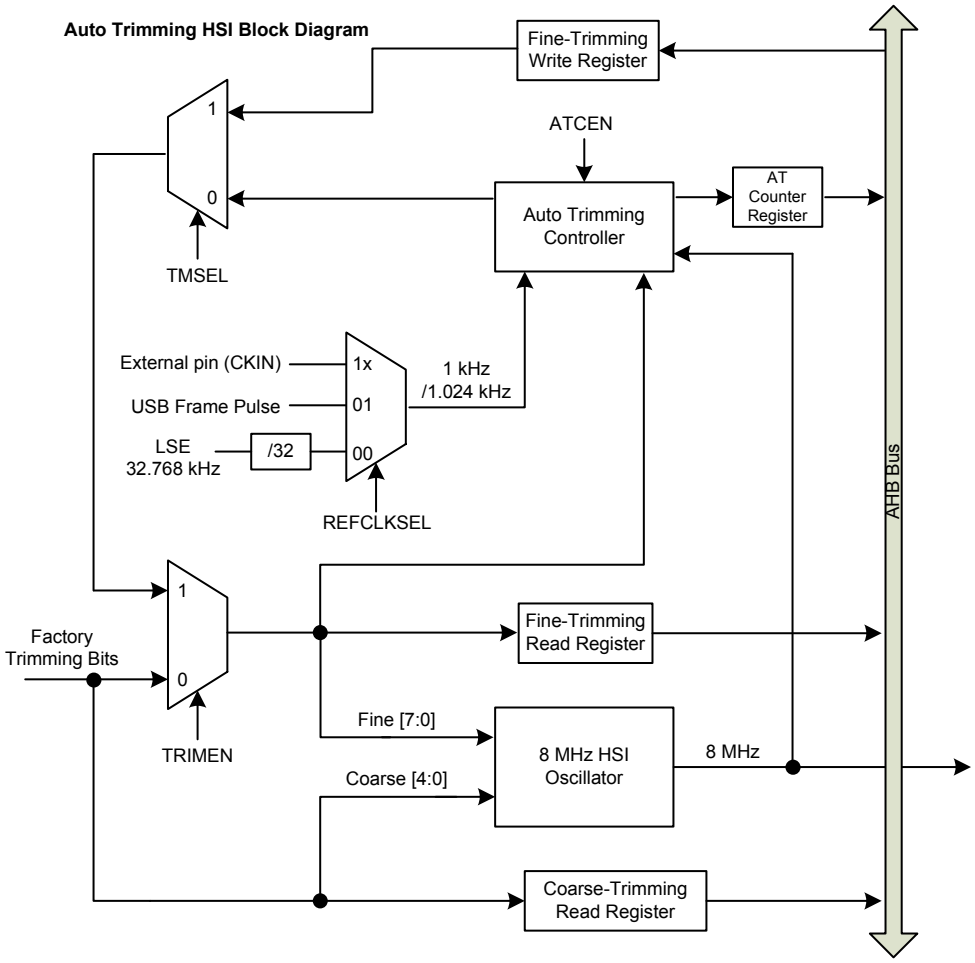


Figure 15. HSI Auto Trimming Block Diagram

Phase Locked Loop – PLL

This PLL can provide 4 ~ 48 MHz clock output which is 1 ~ 12 multiples of a fundamental reference frequency of 4 ~ 16 MHz. The rationale of the clock synthesizer relies on the digital Phase Locked Loop (PLL) which includes a reference divider, a two-stage feedback divider, a two-stage output divider, a digital phase frequency detector (PFD), a current-controlled charge pump, a built-in loop filter and a voltage-controlled oscillator (VCO) to achieve a stable phase-locked state.

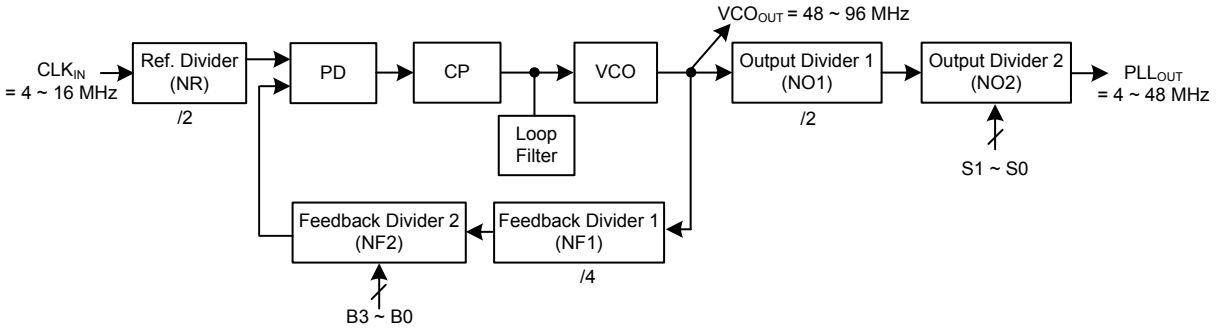


Figure 16. PLL Block Diagram

Frequency of the PLL output clock can be determined by the following formula:

$$PLL_{OUT} = CLK_{IN} \times \frac{NF1 \times NF2}{NR \times NO1 \times NO2} = CLK_{IN} \times \frac{4 \times NF2}{2 \times 2 \times NO2} = CLK_{IN} \times \frac{NF2}{NO2}$$

where NR = Ref divider = 2, NF1 = Feedback Divider 1 = 4, NF2 = Feedback Divider 2 = 1 ~ 16,
NO1 = Output Divider 1 = 2, NO2 = Output Divider 2 = 1, 2, 4 or 8

Considering the duty cycle with 50%, both input frequency and output frequency is divided by 2. Assume that a given CLK_{in} frequency as PLL input generates a specific PLL output frequency; a larger number of NF2 is suggested because it will cause the PLL more stable and less jittered but enlarges the settling time. The output and feedback of divider 2 value are described in Table 15 and Table 16. All the configuration bits (S1 ~ S0, B3 ~ B0) in Table 15 and Table 16 as well as the Bypass mode control are defined in the PLL Configuration Register (PLLCFGR) and PLL Control Register (PLLCR) in the section of Register Definition. Note that VCO_{OUT} is ranged from 48 MHz to 96 MHz. If the selected configurations exceed this range, the output frequency of PLL will not be promised to match the above PLL_{OUT} formula.

The PLL can be switched on or off by using the PLLEN bit in the Global Clock Control Register (GCCR). The PLLRDY flag in the Global Clock Status Register (GCSR) will indicate if the PLL clock is stable.

Table 15. Output Divider 2 Value Mapping

Output Divider 2 Setup Bits S1 ~ S0 (POTD Bits in the PLLCFGR Register)	NO2 (Output Divider 2 Value)
00	1
01	2
10	4
11	8

Table 16. Feedback Divider 2 Value Mapping

Feedback Divider 2 Setup Bits B3 ~ B0 (PFBD Bits in the PLLCFGR Register)	NF2 (Feedback Divider 2 Value)
0000	16
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7
1000	8
1001	9
1010	10
1011	11
1100	12
:	:
:	:
:	:
1111	15

Low Speed External Crystal Oscillator – LSE

The low speed external crystal or ceramic resonator oscillator with 32,768 Hz frequency produces a low power but highly accurate clock source for the circuits of Real-Time-Clock peripheral, Watchdog Timer or system clock. The associated hardware configuration is shown in the following figure. The crystal or ceramic resonator must be placed across the two LSE pins (X32KIN / X32KOUT) and the external components such as resistors and capacitors are necessary to make it oscillate properly. The LSE oscillator can be switched on or off by using the LSEEN bit in the RTC Control Register (RTCCR). The LSERDY flag in the Global Clock Status Register (GCSR) will indicate if the LSE clock is stable.

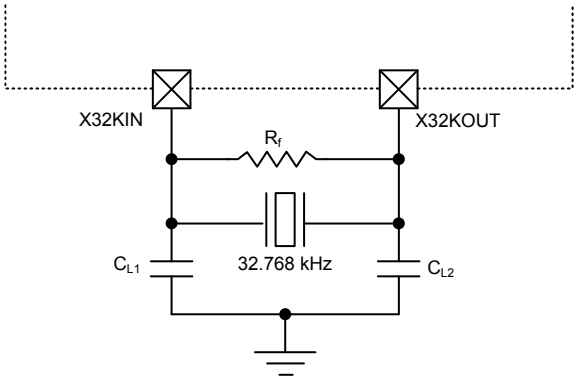


Figure 17. External Crystal, Ceramic and Resonators for LSE

Low Speed Internal RC Oscillator – LSI

The low speed internal RC oscillator with frequency of about 32 kHz produces a low power clock source for the circuits of Real-Time-Clock peripheral, Watchdog Timer or system clock. The LSI is also a low-cost clock source because no external component is needed to make it oscillates. The LSI RC oscillator can be switched on or off by using the LSIEN bit in the RTC Control Register (RTCCR). The accuracy of the frequency of the low speed internal RC oscillator LSI is shown as the corresponding data sheet. The LSIRDY flag in the Global Clock Status Register (GCSR) will indicate if the LSI clock is stable.

Clock Ready Flag

CKCU provides clock ready flags for HSI. HSE, PLL, LSI and LSE to indicate whether these clocks are stable before using them as system clock source or other purpose. Software can check specific clock is ready or not by polling separate clock ready status bits in GCSR register. Besides, CKCU can trigger an interrupt to notify specific clock is ready if the corresponding interrupt enable bit in the GCIR register is set. Software shall clear interrupt status bit in GCIR register by interrupt service routine.

System Clock (CK_SYS) Selection

After a system reset occurs, the default system clock source CK_SYS will be the high speed internal RC oscillator HSI. The CK_SYS may come from the HSI, HSE, LSE, LSI or PLL output clock and it can be switched from one clock source to another via the System Clock Switch bits, SW, in the Global Clock Control Register GCCR. The system will still run under the original clock until the destination clock gets ready. The corresponding clock ready status bits in the Global Clock Status Register GCSR will indicate whether the selected clock is ready to use or not. The CKCU also contains the clock source status bits in the Clock Source Status Register CKST to indicate which clock is currently used as the system clock. If a clock source or the PLL output clock is used as the system clock source, it is not possible to stop it. More detail about the clock enable function is described below.

If any following action takes effect, the HSI is always under enable state.

- Enable PLL and configure its source clock to HSI. (PLLEN, PLLSRC)
- Enable Clock monitor. (CKMEN)
- Configure clock switch register bit to HSI. (SW)
- Configure HSI enable register bit to 1. (HSIEN)

If any following action takes effect, the HSE is always under enable state.

- Enable PLL and configure its source clock to HSE. (PLLEN, PLLSRC)
- Configure clock switch register bit to HSE. (SW)
- Configure HSE enable register bit to 1. (HSEEN)

If any following action takes effect, the PLL is always under enable state.

- Enable USB Enable register bit. (USBEN)
- Configure clock switch register bit to PLL. (SW)
- Configure PLL enable register bit to 1. (PLLEN)

Programming guide of System clock selection is listed below.

1. Enable any source clock which will become system clock or PLL input clock.
2. Configure the PLLSRC register after the ready flags of both HSI and HSE are asserted
3. Configure the SW register to change system clock source after the corresponding ready flag of the clock source is asserted. Note that system clock will force to HSI if clock monitor is enabled and the PLL output or HSE clock configured as system clock is stuck at 0 or 1.

HSE Clock Monitor

The main function of the oscillator check is enabled by the HSE Clock Monitor Enable bit CKMEN in the Global Clock Control Register, GCCR. The HSE clock monitor should be enabled after the HSE oscillator start-up delay and be disabled when the HSE oscillator is stopped. Once the HSE oscillator failure is detected, the HSE oscillator will automatically be disabled. The HSE clock stuck flag CKSF in the Global Clock Interrupt Register GCIR will be set and an event of main oscillator failure will be generated if the clock fail interrupt enable bit CKSIE in the GCIR is set. This failure interrupt is connected to the exception vector of CPU Non-Maskable Interrupt, NMI. If the HSE is directly used as the system clock, When the HSE oscillator failure occurs, the HSE will be turned off and the system clock will be switched to the HSI automatically by the hardware. If the HSE is used as the clock input of the PLL circuit and the system clock comes from the PLL circuit output, the PLL circuit will also be turned off as well as the HSE when the failure happens.

Clock Output Capability

The device has the clock output capability to allow the clocks to be output on the specific external output pin CKOUT. The configuration registers of the corresponding GPIO port must be well configured in the "Alternate Function I/O" section, AFIO, to output the selected clock signal. There are seven output clock signals to be selected via the device clock output source selection bits CKOUTSRC in the Global Clock Configuration Register, GCFGR.

Table 17. CKOUT Clock Source

CKOUTSRC[2:0]	Clock Source
000	CK_REF = CK_PLL / (CKREFPRE + 1) / 2
001	CK_AHB / 16
010	CK_SYS / 16
011	CK_HSE / 16
100	CK_HSI / 16
101	CK_LSE
110	CK_LSI

Register Map

The following table shows the CKCU register and reset value.

Table 18. CKCU Register Map

Register	Offset	Description	Reset Value
GCFGR	0x000	Global Clock Configuration Register	0x0000_0102
GCCR	0x004	Global Clock Control Register	0x0000_0803
GCSR	0x008	Global Clock Status Register	0x0000_0028
GCIR	0x00C	Global Clock Interrupt Register	0x0000_0000
PLLCFGR	0x018	PLL Configuration Register	0x0000_0000
PLLCR	0x01C	PLL Control Register	0x0000_0000
AHBCFGR	0x020	AHB Configuration Register	0x0000_0000
AHBCCR	0x024	AHB Clock Control Register	0x0000_0065
APBCFGR	0x028	APB Configuration Register	0x0000_0000
APBCCR0	0x02C	APB Clock Control Register 0	0x0000_0000
APBCCR1	0x030	APB Clock Control Register 1	0x0000_0000
CKST	0x034	Clock Source Status Register	0x0100_0003
APBPCSR0	0x038	APB Peripheral Clock Selection Register 0	0x0000_0000
APBPCSR1	0x03C	APB Peripheral Clock Selection Register 1	0x0000_0000
HSICR	0x040	HSI Control Register	0xXXXX_0000 where X is undefined
HSIATCR	0x044	HSI Auto Trimming Counter Register	0x0000_0000
APBPCSR2	0x048	APB Peripheral Clock Selection Register 2	0x0000_0000
LPCR	0x300	Low Power Control Register	0x0000_0000
MCUDBGCR	0x304	MCU Debug Control Register	0x0000_0000

Register Descriptions

Global Clock Configuration Register – GCFGR

This register specifies the clock source for PLL / USART / Watchdog Timer / CKOUT.

Offset: 0x000

Reset value: 0x0000_0102

	31	30	29	28	27	26	25	24
	LPMOD			Reserved				
Type/Reset	RO 0	RO 0	RO 0					
	23	22	21	20	19	18	17	16
	Reserved							
Type/Reset								
	15	14	13	12	11	10	9	8
	CKREFPRE					Reserved		PLLSRC
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0			RW 1
	7	6	5	4	3	2	1	0
	Reserved					CKOUTSRC		
Type/Reset						RW 0	RW 1	RW 0

Bits	Field	Descriptions
[31:29]	LPMOD	Lower Power Mode Status 000: When Chip is in running mode 001: When Chip wants to enter Sleep mode 010 :When Chip wants to enter Deep-Sleep1 mode 011: When Chip wants to enter Deep-Sleep2 mode 100: When Chip wants to enter Power-Down mode Others: Reserved Set and reset by hardware.
[15:11]	CKREFPRE	CK_REF Clock Prescaler Selection CK_REF = CK_PLL / (CKREFPRE + 1) / 2 00000: CK_REF = CK_PLL / 2 00001: CK_REF = CK_PLL / 4 ... 11111: CK_REF = CK_PLL / 64 Set and reset by software to control CK_REF clock prescaler setting.
[8]	PLLSRC	PLL Clock Source Selection 0: External 4 ~ 16 MHz crystal oscillator clock is selected (HSE) 1: Internal 8 MHz RC oscillator clock is selected (HSI) Set and reset by software to control the PLL clock source.
[2:0]	CKOUTSRC	CKOUT Clock Source Selection 000: CK_REF is selected where CK_REF = CK_PLL / (CKREFPRE+1) / 2 001: (CK_AHB / 16) is selected 010: (CK_SYS / 16) is selected 011: (CK_HSE / 16) is selected 100: (CK_HSI / 16) is selected 101: CK_LSE is selected 110: CK_LSI is selected 111: Reserved Set and reset by software.

Global Clock Control Register – GCCR

This register specifies the clock enable bits.

Offset: 0x004

Reset value: 0x0000_0803

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	22	21	20	19	18	17	16
	Reserved						PSRCEN	CKMEN
Type/Reset							RW 0	RW 0
	15	14	13	12	11	10	9	8
	Reserved				HSIEN	HSEEN	PLLEN	HSEGAIN
Type/Reset					RW 1	RW 0	RW 0	RW 0
	7	6	5	4	3	2	1	0
	Reserved					SW		
Type/Reset						RW 0	RW 1	RW 1

Bits	Field	Descriptions
[17]	PSRCEN	Power Saving Wakeup RC Clock Enable 0: No action 1: Use Internal 8 MHz RC clock (HSI) as system clock after a Deep-Sleep1/2 modes wakeup. The software can set PSRCEN high before entering Deep-Sleep1/2 modes in order to reduce the waiting time after wakeup. When PSRCEN = 1, hardware will select HSI as clock source after the system wakeup from Deep-Sleep1/2 modes. Meanwhile, instruction can start execution since the HSI clock is provided to CPU. After the original clock source (which selected as CK_SYS before enter Deep-Sleep1/2 modes mode) is ready, hardware will switch back the clock source as originally.
[16]	CKMEN	HSE Clock Monitor Enable 0: Disable External 4 ~ 16 MHz crystal oscillator clock monitor 1: Enable External 4 ~ 16 MHz crystal oscillator clock monitor When hardware detects the HSE clock stuck at low / high state, internal hardware will switch the system clock to internal high speed RC clock (HSI). The way to recover the system clock is by an external reset, a power on reset or clearing the CKSF by software.
[11]	HSIEN	Internal High Speed Clock Enable 0: Internal 8 MHz RC oscillator clock is set to off 1: Internal 8 MHz RC oscillator clock is set to on Set and reset by software. This bit can not be reset if HSI clock is used as system clock or the PLL input clock.
[10]	HSEEN	External High Speed Clock Enable 0: External 4 ~ 16 MHz crystal oscillator clock is disabled 1: External 4 ~ 16 MHz crystal oscillator clock is enabled Set and reset by software. This bit can not be reset if the HSE clock is used as system clock or the PLL input clock.

Bits	Field	Descriptions
[9]	PLLEN	PLL Enable 0: PLL is disabled 1: PLL is enabled Set and reset by software to enable PLL. This bit cannot be reset if the PLL clock is used as system clock.
[8]	HSEGAIN	External High Speed Clock Gain Selection 0: HSE is in low gain mode 1: HSE is in high gain mode
[2:0]	SW	System Clock Switch 00x: CK_PLL clock out as system clock 010: CK_HSE as system clock 011: CK_HSI as system clock 110: CK_LSE as system clock 111: CK_LSI as system clock Other: CK_HSI as system clock These bits are used to select the CK_SYS source. If the HSE oscillator is used directly or indirectly as the system clock and the HSE clock monitor function is enabled, once the HSE failure is detected, these bits will be set by hardware to force HSI (b011) as the system clock. Note: When switch the system clock using the SW bit, the system clock is not immediately switched and a certain delay is necessary. The S/W can monitor the CKSWST bit in the clock source status register CKSTR to make sure which clock is currently used as system clock.

Global Clock Status Register – GCSR

This register indicates the clock ready status.

Offset: 0x008

Reset value: 0x0000_0028

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	22	21	20	19	18	17	16
	Reserved							
Type/Reset								
	15	14	13	12	11	10	9	8
	Reserved							
Type/Reset								

	7	6	5	4	3	2	1	0
		Reserved	LSIRDY	LSERDY	HSIRDY	HSERDY	PLLRDY	Reserved
Type/Reset			RO 1	RO 0	RO 1	RO 0	RO 0	

Bits	Field	Descriptions
[5]	LSIRDY	Internal Low Speed Clock Ready Flag 0: Internal 32 kHz RC oscillator clock is not ready 1: Internal 32 kHz RC oscillator clock is ready Set by hardware to indicate that the LSI is stable to be used.
[4]	LSERDY	External Low Speed Clock Ready Flag 0: External 32,768 Hz crystal oscillator clock is not ready 1: External 32,768 Hz crystal oscillator clock is ready Set by hardware to indicate that the LSE is stable to be used.
[3]	HSIRDY	Internal High Speed Clock Ready Flag 0: Internal 8 MHz RC oscillator clock is not ready 1: Internal 8 MHz RC oscillator clock is ready Set by hardware to indicate that the HSI is stable to be used.
[2]	HSERDY	External High Speed Clock Ready Flag 0: External 4 ~ 16 MHz crystal oscillator clock is not ready 1: External 4 ~ 16 MHz crystal oscillator clock is ready Set by hardware to indicate that the HSE is stable to be used.
[1]	PLLRDY	PLL Clock Ready Flag 0: PLL is not ready 1: PLL is ready Set by hardware to indicate that the PLL output is stable to be used.

Global Clock Interrupt Register – GCIR

This register specifies interrupt enable and flag bits.

Offset: 0x00C

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	22	21	20	19	18	17	16
	Reserved							CKSIE
Type/Reset								RW 0
	15	14	13	12	11	10	9	8
	Reserved							
Type/Reset								
	7	6	5	4	3	2	1	0
	Reserved							CKSF
Type/Reset								WC 0

Bits	Field	Descriptions
[16]	CKSIE	Clock Stuck Interrupt Enable 0: Disable clock fail interrupt 1: Enable clock fail interrupt Set and reset by software to enable / disable interrupt caused by clock monitor.
[0]	CKSF	Clock Stuck Interrupt Flag 0: Clock works normally 1: HSE clock is stuck Reset by software (Write 1 clear). Set by hardware when HSE clock stuck and CKSIE is set.

PLL Configuration Register – PLLCFGR

This register specifies configuration of PLL.

Offset: 0x018

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved					PFBD		
Type/Reset						RW 0	RW 0	RW 0
	23	22	21	20	19	18	17	16
	PFBD			POTD		Reserved		
Type/Reset	RW 0	RW 0	RW 0					
	15	14	13	12	11	10	9	8
	Reserved							
Type/Reset								
	7	6	5	4	3	2	1	0
	Reserved							
Type/Reset								

Bits	Field	Descriptions
[26:23]	PFBD	PLL VCO Output Clock Feedback Divider (B3 ~ B0 in Figure 16) Feedback Divider divides the output clock from VCO of PLL.
[22:21]	POTD	PLL Output Clock Divider (S1 ~ S0 in Figure 16)

PLL Control Register – PLLCR

This register specifies Bypass mode of PLL.

Offset: 0x01C

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	PLLBPS	Reserved						
Type/Reset	RW 0							
	23	22	21	20	19	18	17	16
	Reserved							
Type/Reset								
	15	14	13	12	11	10	9	8
	Reserved							
Type/Reset								
	7	6	5	4	3	2	1	0
	Reserved							
Type/Reset								

Bits	Field	Descriptions
[31]	PLLBPS	PLL Bypass Mode Enable 0: Disable PLL Bypass mode 1: Enable PLL Bypass mode where $f_{OUT} = f_{IN}$

AHB Configuration Register – AHBCFGR

This register specifies frequency of system clock.

Offset: 0x020

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	22	21	20	19	18	17	16
	Reserved							
Type/Reset								
	15	14	13	12	11	10	9	8
	Reserved							
Type/Reset								
	7	6	5	4	3	2	1	0
	Reserved					AHBPRE		
Type/Reset						RW 0	RW 0	RW 0

Bits	Field	Descriptions
[2:0]	AHBPRE	AHB Pre-scaler 000: CK_AHB = CK_SYS 001: CK_AHB = CK_SYS / 2 010: CK_AHB = CK_SYS / 4 011: CK_AHB = CK_SYS / 8 100: CK_AHB = CK_SYS / 16 101: CK_AHB = CK_SYS / 32 110: CK_AHB = CK_SYS / 32 111: CK_AHB = CK_SYS / 32 Set and reset by software to control the division factor of the AHB clock.

AHB Clock Control Register – AHBCCR

This register specifies AHB clock enable bits.

Offset: 0x024

Reset value: 0x0000_0065

	31	30	29	28	27	26	25	24
	Reserved						QSPIEN	DIVEN
Type/Reset							RW 0	RW 0
	23	22	21	20	19	18	17	16
	Reserved				PDEN	PCEN	PBEN	PAEN
Type/Reset					RW 0	RW 0	RW 0	RW 0
	15	14	13	12	11	10	9	8
		Reserved	CRCEN	Reserved	CKREFEN	USBEN		Reserved
Type/Reset			RW 0		RW 0	RW 0		
	7	6	5	4	3	2	1	0
	Reserved	APBEN	BMEN	PDMAEN	Reserved	SRAMEN	Reserved	FMCEN
Type/Reset		RW 1	RW 1	RW 0		RW 1		RW 1

Bits	Field	Descriptions
[25]	QSPIEN	QSPI Clock Enable 0: QSPI clock is disabled 1: QSPI clock is enabled Set and reset by software
[24]	DIVEN	Divider Clock Enable 0: Divider clock is disabled 1: Divider clock is enabled Set and reset by software
[19]	PDEN	GPIO Port D Clock Enable 0: Port D clock is disabled 1: Port D clock is enabled Set and reset by software
[18]	PCEN	GPIO Port C Clock Enable 0: Port C clock is disabled 1: Port C clock is enabled Set and reset by software
[17]	PBEN	GPIO Port B Clock Enable 0: Port B clock is disabled 1: Port B clock is enabled Set and reset by software
[16]	PAEN	GPIO Port A Clock Enable 0: Port A clock is disabled 1: Port A clock is enabled Set and reset by software
[13]	CRCEN	CRC Module Clock Enable 0: CRC clock disable 1: CRC clock enable Set and reset by software.

Bits	Field	Descriptions
[11]	CKREFEN	CK_REF Clock Enable 0: CK_REF clock is disabled 1: CK_REF clock is enabled Set and reset by software
[10]	USBEN	USB Clock Enable 0: USB clock disabled 1: USB clock enabled Set and reset by software
[6]	APBEN	APB bridge Clock Enable 0: APB bridge clock is automatically disabled by hardware during Sleep mode 1: APB bridge clock is always enabled during Sleep mode Set and reset by software. User can set APBEN as 0 to reduce power consumption if the APB bridge is unused during Sleep mode.
[5]	BMEN	Bus Matrix Clock Enable 0: Bus Matrix clock is automatically disabled by hardware during Sleep mode 1: Bus Matrix clock is always enabled during Sleep mode Set and reset by software. User can set BMEN as 0 to reduce power consumption if the bus matrix is unused during Sleep mode.
[4]	PDMAEN	Peripheral DMA Clock Enable 0: PDMA clock is disabled 1: PDMA clock is enabled Set and reset by software. Note: The PDMA can independently operate when the processor enters the sleep mode. But the relative AHB bus slave or peripherals clock has to be enabled.
[2]	SRAMEN	SRAM Clock Enable 0: SRAM clock auto disable by hardware during Sleep mode 1: SRAM clock always enable during Sleep mode Set and reset by software. Users can set SRAMEN as 0 to reduce power consumption if the SRAM is unused during Sleep mode.
[0]	FMCEN	Flash Memory Controller Clock Enable 0: FMC clock auto disable by hardware during Sleep mode 1: FMC clock always enable during Sleep mode Set and reset by software. Users can set FMCEN as 0 to reduce power consumption if the Flash Memory is unused during Sleep mode.

APB Configuration Register – APBCFGR

This register specifies the frequency of special peripherals operation clock.

Offset: 0x028

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved					MIDIDIV		
Type/Reset						RW 0	RW 0	RW 0
	23	22	21	20	19	18	17	16
	Reserved					ADCDIV		
Type/Reset						RW 0	RW 0	RW 0
	15	14	13	12	11	10	9	8
	Reserved							
Type/Reset								
	7	6	5	4	3	2	1	0
	Reserved							
Type/Reset								

Bits	Field	Descriptions
[26:24]	MIDIDIV	MIDI Engine Clock Frequency Divide Selection 000: CK_MIDI = CK_AHB / 16 001: CK_MIDI = CK_AHB / 13 010: CK_MIDI = CK_AHB / 11 011: CK_MIDI = CK_AHB / 9 100: CK_MIDI = CK_AHB / 8 101: CK_MIDI = CK_AHB / 8 Other: CK_MIDI = CK_AHB / 16 Set and reset by software to control MIDI engine clock division factor.
[18:16]	ADCDIV	ADC Clock Frequency Divide Selection 000: CK_ADC = CK_AHB / 1 001: CK_ADC = CK_AHB / 2 010: CK_ADC = CK_AHB / 4 011: CK_ADC = CK_AHB / 8 100: CK_ADC = CK_AHB / 16 101: CK_ADC = CK_AHB / 32 110: CK_ADC = CK_AHB / 64 111: CK_ADC = CK_AHB / 3 Set and reset by software to control ADC conversion clock division factor.

APB Clock Control Register 0 – APBCCR0

This register specifies clock enable bits of APB peripherals.

Offset: 0x02C

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved			MIDIEN		Reserved	I2SEN	Reserved
Type/Reset				RW 0			RW 0	
	23	22	21	20	19	18	17	16
	Reserved							
Type/Reset								
	15	14	13	12	11	10	9	8
	EXTIEN	AFIOEN	Reserved			UREN	Reserved	USREN
Type/Reset	RW 0	RW 0				RW 0		RW 0
	7	6	5	4	3	2	1	0
	Reserved			SPIEN	Reserved			I2CEN
Type/Reset				RW 0				RW 0

Bits	Field	Descriptions
[28]	MIDIEN	MIDI Clock Enable 0: MIDI clock is disabled 1: MIDI clock is enabled Set and reset by software.
[25]	I2SEN	I^2S Clock Enable 0: I^2S clock is disabled 1: I^2S clock is enabled Set and reset by software.
[15]	EXTIEN	External Interrupt Clock Enable 0: EXTI clock is disabled 1: EXTI clock is enabled Set and reset by software.
[14]	AFIOEN	Alternate Function I/O Clock Enable 0: AFIO clock is disabled 1: AFIO clock is enabled Set and reset by software.
[10]	UREN	UART Clock Enable 0: UART clock is disabled 1: UART clock is enabled Set and reset by software.
[8]	USREN	USART Clock Enable 0: USART clock is disabled 1: USART clock is enabled Set and reset by software.
[4]	SPIEN	SPI Clock Enable 0: SPI clock is disabled 1: SPI clock is enabled Set and reset by software.

Bits	Field	Descriptions
[0]	I2CEN	I^2C Clock Enable 0: I^2C clock is disabled 1: I^2C clock is enabled Set and reset by software.

APB Clock Control Register 1 – APBCCR1

This register specifies clock enable bits APB peripherals.

Offset: 0x030

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	SCTM3EN	SCTM2EN	SCTM1EN	SCTM0EN	Reserved			ADCCEN
Type/Reset	RW 0	RW 0	RW 0	RW 0				RW 0
	23	22	21	20	19	18	17	16
	Reserved		DACCEN	Reserved			BFTM1EN	BFTM0EN
Type/Reset			RW 0				RW 0	RW 0
	15	14	13	12	11	10	9	8
	Reserved							GPTMEN
Type/Reset								RW 0
	7	6	5	4	3	2	1	0
	Reserved	VDDREN	Reserved	WDTREN	Reserved			
Type/Reset		RW 0		RW 0				

Bits	Field	Descriptions
[31]	SCTM3EN	SCTM3 Clock Enable 0: SCTM3 clock is disabled 1: SCTM3 clock is enabled Set and reset by software.
[30]	SCTM2EN	SCTM2 Clock Enable 0: SCTM2 clock is disabled 1: SCTM2 clock is enabled Set and reset by software.
[29]	SCTM1EN	SCTM1 Clock Enable 0: SCTM1 clock is disabled 1: SCTM1 clock is enabled Set and reset by software.
[28]	SCTM0EN	SCTM0 Clock Enable 0: SCTM0 clock is disabled 1: SCTM0 clock is enabled Set and reset by software.
[24]	ADCCEN	ADC Controller Clock Enable 0: ADC clock is disabled 1: ADC clock is enabled Set and reset by software.

Bits	Field	Descriptions
[21]	DACCEN	DAC Controller Clock Enable 0: DAC clock is disabled 1: DAC clock is enabled Set and reset by software.
[17]	BFTM1EN	BFTM1 Clock Enable 0: BFTM1 clock is disabled 1: BFTM1 clock is enabled Set and reset by software.
[16]	BFTM0EN	BFTM0 Clock Enable 0: BFTM0 clock is disabled 1: BFTM0 clock is enabled Set and reset by software.
[8]	GPTMEN	GPTM Clock Enable 0: GPTM clock is disabled 1: GPTM clock is enabled Set and reset by software.
[6]	VDDREN	V_{DD} Domain Clock Enable for Registers Access 0: Register access clock is disabled 1: Register access clock is enabled Set and reset by software.
[4]	WDTREN	Watchdog Timer Clock Enable for Registers Access 0: Register access clock is disabled 1: Register access clock is enabled Set and reset by software.

Clock Source Status Register – CKST

This register specifies status of clock source.

Offset: 0x034

Reset value: 0x0100_0003

	31	30	29	28	27	26	25	24
	Reserved					HSIST		
Type/Reset						RO 0	RO 0	RO 1
	23	22	21	20	19	18	17	16
	Reserved						HSEST	
Type/Reset							RO 0	RO 0
	15	14	13	12	11	10	9	8
	Reserved				PLLST			
Type/Reset					RO 0	RO 0	RO 0	RO 0
	7	6	5	4	3	2	1	0
	Reserved					CKSWST		
Type/Reset						RO 0	RO 1	RO 1

Bits	Field	Descriptions
[26:24]	HSIST	Internal High Speed Clock Occupation Status (CK_HSI) xx1: HSI is used by System Clock (CK_SYS) (SW = 0x03) x1x: HSI is used by PLL 1xx: HSI is used by Clock Monitor
[17:16]	HSEST	External High Speed Clock Occupation Status (CK_HSE) x1: HSE is used by System Clock (CK_SYS) (SW = 0x02) 1x: HSE is used by PLL
[11:8]	PLLST	PLL Clock Occupation Status xxx1: PLL is used by System Clock (CK_SYS) xx1x: PLL is used by USART x1xx: PLL is used by USB 1xxx: PLL is used by CK_REF
[2:0]	CKSWST	Clock Switch Status 00x: CK_PLL clock out as system clock 010: CK_HSE as system clock 011: CK_HSI as system clock 110: CK_LSE as system clock 111: CK_LSI as system clock The fields are status to indicate which clock source is using as system clock currently.

APB Peripheral Clock Selection Register 0 – APBPCSR0

This register specifies APB peripheral clock prescaler selection.

Offset: 0x038

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved		URPCLK		Reserved		USRPCLK	
Type/Reset			RW 0	RW 0			RW 0	RW 0
	23	22	21	20	19	18	17	16
	Reserved		GPTMPCLK		Reserved			
Type/Reset			RW 0	RW 0				
	15	14	13	12	11	10	9	8
	BFTM1PCLK		BFTM0PCLK		Reserved			
Type/Reset	RW 0	RW 0	RW 0	RW 0				
	7	6	5	4	3	2	1	0
	Reserved		SPIPCLK		Reserved		I2CPCLK	
Type/Reset			RW 0	RW 0			RW 0	RW 0

Bits	Field	Descriptions
[29:28]	URPCLK	UART Peripheral Clock Selection 00: PCLK = CK_AHB 01: PCLK = CK_AHB / 2 10: PCLK = CK_AHB / 4 11: PCLK = CK_AHB / 8 PCLK = Peripheral Clock; CK_AHB = AHB and CPU clock
[25:24]	USRPCLK	USART Peripheral Clock Selection 00: PCLK = CK_AHB 01: PCLK = CK_AHB / 2 10: PCLK = CK_AHB / 4 11: PCLK = CK_AHB / 8 PCLK = Peripheral Clock; CK_AHB = AHB and CPU clock
[21:20]	GPTMPCLK	GPTM Peripheral Clock Selection 00: PCLK = CK_AHB 01: PCLK = CK_AHB / 2 10: PCLK = CK_AHB / 4 11: PCLK = CK_AHB / 8 PCLK = Peripheral Clock; CK_AHB = AHB and CPU clock
[15:14]	BFTM1PCLK	BFTM1 Peripheral Clock Selection 00: PCLK = CK_AHB 01: PCLK = CK_AHB/2 10: PCLK = CK_AHB/4 11: PCLK = CK_AHB/8 PCLK = Peripheral Clock; CK_AHB = AHB and CPU clock

Bits	Field	Descriptions
[13:12]	BFTM0PCLK	BFTM0 Peripheral Clock Selection 00: PCLK = CK_AHB 01: PCLK = CK_AHB / 2 10: PCLK = CK_AHB / 4 11: PCLK = CK_AHB / 8 PCLK = Peripheral Clock; CK_AHB = AHB and CPU clock
[5:4]	SPIPCLK	SPI Peripheral Clock Selection 00: PCLK = CK_AHB 01: PCLK = CK_AHB / 2 10: PCLK = CK_AHB / 4 11: PCLK = CK_AHB / 8 PCLK = Peripheral Clock; CK_AHB = AHB and CPU clock
[1:0]	I2CPCLK	I^2C Peripheral Clock Selection 00: PCLK = CK_AHB 01: PCLK = CK_AHB / 2 10: PCLK = CK_AHB / 4 11: PCLK = CK_AHB / 8 PCLK = Peripheral Clock; CK_AHB = AHB and CPU clock

APB Peripheral Clock Selection Register 1 – APBPCSR1

This register specifies APB peripheral clock prescaler selection.

Offset: 0x03C

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
		SCTM3PCLK		SCTM2PCLK		SCTM1PCLK		SCTM0PCLK
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0
	23	22	21	20	19	18	17	16
		Reserved		I2SPCLK		Reserved		
Type/Reset			RW 0	RW 0				
	15	14	13	12	11	10	9	8
		VDDRPCLK		WDTRPCLK		Reserved		
Type/Reset	RW 0	RW 0	RW 0	RW 0				
	7	6	5	4	3	2	1	0
		Reserved		ADCCPCLK		EXTIPCLK		AFIOPCLK
Type/Reset			RW 0	RW 0	RW 0	RW 0	RW 0	RW 0

Bits	Field	Descriptions
[31:30]	SCTM3PCLK	SCTM3 Peripheral Clock Selection 00: PCLK = CK_AHB 01: PCLK = CK_AHB / 2 10: PCLK = CK_AHB / 4 11: PCLK = CK_AHB / 8 PCLK = Peripheral Clock; CK_AHB = AHB and CPU clock
[29:28]	SCTM2PCLK	SCTM2 Peripheral Clock Selection 00: PCLK = CK_AHB 01: PCLK = CK_AHB / 2 10: PCLK = CK_AHB / 4 11: PCLK = CK_AHB / 8 PCLK = Peripheral Clock; CK_AHB = AHB and CPU clock
[27:26]	SCTM1PCLK	SCTM1 Peripheral Clock Selection 00: PCLK = CK_AHB 01: PCLK = CK_AHB / 2 10: PCLK = CK_AHB / 4 11: PCLK = CK_AHB / 8 PCLK = Peripheral Clock; CK_AHB = AHB and CPU clock
[25:24]	SCTM0PCLK	SCTM0 Peripheral Clock Selection 00: PCLK = CK_AHB 01: PCLK = CK_AHB / 2 10: PCLK = CK_AHB / 4 11: PCLK = CK_AHB / 8 PCLK = Peripheral Clock; CK_AHB = AHB and CPU clock
[21:20]	I2SPCLK	I^2S Peripheral Clock Selection 00: PCLK = CK_AHB 01: PCLK = CK_AHB / 2 10: PCLK = CK_AHB / 4 11: PCLK = CK_AHB / 8 PCLK = Peripheral Clock; CK_AHB = AHB and CPU clock

Bits	Field	Descriptions
[15:14]	VDDRCLK	V_{DD} Domain Register Access Clock Selection 00: PCLK = CK_AHB / 4 01: PCLK = CK_AHB / 8 10: PCLK = CK_AHB / 16 11: PCLK = CK_AHB / 32 PCLK = Peripheral Clock; CK_AHB = AHB and CPU clock
[13:12]	WDTRPCLK	WDT Register Access Clock Selection 00: PCLK = CK_AHB 01: PCLK = CK_AHB / 2 10: PCLK = CK_AHB / 4 11: PCLK = CK_AHB / 8 PCLK = Peripheral Clock; CK_AHB = AHB and CPU clock
[5:4]	ADCCPCLK	ADC Controller Peripheral Clock Selection 00: PCLK = CK_AHB 01: PCLK = CK_AHB / 2 10: PCLK = CK_AHB / 4 11: PCLK = CK_AHB / 8 PCLK = Peripheral Clock; CK_AHB = AHB and CPU clock
[3:2]	EXTIPCLK	EXTI Peripheral Clock Selection 00: PCLK = CK_AHB 01: PCLK = CK_AHB / 2 10: PCLK = CK_AHB / 4 11: PCLK = CK_AHB / 8 PCLK = Peripheral Clock; CK_AHB = AHB and CPU clock
[1:0]	AFIOPCLK	AFIO Peripheral Clock Selection 00: PCLK = CK_AHB 01: PCLK = CK_AHB / 2 10: PCLK = CK_AHB / 4 11: PCLK = CK_AHB / 8 PCLK = Peripheral Clock; CK_AHB = AHB and CPU clock

HSI Control Register – HSICR

This register is to control the frequency trimming of HSI RC oscillation.

Offset: 0x040

Reset value: 0xXXXX_0000 where X is undefined

	31	30	29	28	27	26	25	24
	Reserved			HSICOARSE				
Type/Reset				RO X	RO X	RO X	RO X	RO X
	23	22	21	20	19	18	17	16
	HSIFINE							
Type/Reset	RW X	RW X	RW X	RW X	RW X	RW X	RW X	RW X
	15	14	13	12	11	10	9	8
	Reserved							
Type/Reset								
	7	6	5	4	3	2	1	0
	FLOCK	Reserved	REFCLKSEL	TMSEL	Reserved	LTRSEL	ATCEN	TRIMEN
Type/Reset	RO 0		RW 0	RW 0		RW 0	RW 0	RW 0

Bits	Field	Descriptions
[28:24]	HSICOARSE	HSI Clock Coarse Trimming Value These bits are initialized automatically at startup. They are adjusted by factory trimming and can not be trimmed by program.
[23:16]	HSIFINE	HSI Clock Fine Trimming Value These bits are initialized automatically at startup. They are also adjusted by factory trimming. But these bits provide an additional user-programmable trimming value that is added to the HSICOARSE[4:0] bits to get more accurate or compensate the variations in voltage and temperature that influence the frequency of the HSI. It can be programmed by software or automatically adjusted by the Auto Trimming Controller (ATC) together with an external reference clock.
[7]	FLOCK	Frequency Lock 0: HSI frequency is not trimmed into target range 1: HSI frequency is trimmed into target range
[6:5]	REFCLKSEL	Reference Clock Selection 00: Select 32.768 kHz external low speed clock source (LSE) 01: Select 1 kHz USB frame pulse 1x: Select external pin (CKIN) 1 kHz pulse These bits are used to select the reference clock for the HSI Auto Trimming Controller.
[4]	TMSEL	Trimming Mode Selection 0: Automatic by Auto Trimming Controller 1: Manual by user program This bit is used to select the HSI RC oscillator trimming function by ATC hardware or user programming via the HSIFINE[7:0] bits in the HSI Control Register.
[2]	LTRSEL	Lock Target Range Selection 0: 0.1 % variation 1: 0.2 % variation This bit is used to select the lock target range of the internal HSI RC oscillator trimming function for 0.1 % or 0.2 % variation.

Bits	Field	Descriptions
[1]	ATCEN	ATC Enable 0: Disable Auto Trimming Controller 1: Enable Auto Trimming Controller
[0]	TRIMEN	Trimming Enable 0: HSI Trimming is disable 1: HSI Trimming is enable The bit enables the HSI RC oscillator trimming function by ATC hardware or user programming.

HSI Auto Trimming Counter Register – HSIATCR

This register contains the counter value of the HSI auto trimming controller

Offset: 0x044

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	22	21	20	19	18	17	16
	Reserved							
Type/Reset								
	15	14	13	12	11	10	9	8
	Reserved		ATCNT					
Type/Reset			RO 0	RO 0	RO 0	RO 0	RO 0	RO 0
	7	6	5	4	3	2	1	0
	ATCNT							
Type/Reset	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0

Bits	Field	Descriptions
[13:0]	ATCNT	Auto Trimming Counter These bits contain the counter value of the HSI auto trimming controller.

APB Peripheral Clock Selection Register 2 – APBPCSR2

This register specifies APB peripheral clock prescaler selection.

Offset: 0x048

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	22	21	20	19	18	17	16
	Reserved							
Type/Reset								
	15	14	13	12	11	10	9	8
	Reserved							MIDIPCLK
Type/Reset							RW 0	RW 0
	7	6	5	4	3	2	1	0
	Reserved					DACPCLK		AFEPCLK
Type/Reset					RW 0	RW 0	RW 0	RW 0

Bits	Field	Descriptions
[9:8]	MIDIPCLK	MIDI Controller Peripheral Clock Selection 00: PCLK = CK_AHB 01: PCLK = CK_AHB / 2 10: PCLK = CK_AHB / 4 11: PCLK = CK_AHB / 8 PCLK = Peripheral Clock; CK_AHB = AHB and CPU clock
[3:2]	DACPCLK	DAC Controller Peripheral Clock Selection 00: PCLK = CK_AHB 01: PCLK = CK_AHB / 2 10: PCLK = CK_AHB / 4 11: PCLK = CK_AHB / 8 PCLK = Peripheral Clock; CK_AHB = AHB and CPU clock
[1:0]	AFEPCLK	AFE Controller Peripheral Clock Selection 00: PCLK = CK_AHB 01: PCLK = CK_AHB / 2 10: PCLK = CK_AHB / 4 11: PCLK = CK_AHB / 8 PCLK = Peripheral Clock; CK_AHB = AHB and CPU clock

Low Power Control Register – LPCR

This register specifies low power control.

Offset: 0x300

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	22	21	20	19	18	17	16
	Reserved							
Type/Reset								
	15	14	13	12	11	10	9	8
	Reserved							USBSLEEP
Type/Reset								RW 0
	7	6	5	4	3	2	1	0
	Reserved							VDDISO
Type/Reset								RW 0

Bits	Field	Descriptions
[8]	USBSLEEP	USB Sleep Software Control Enable 0: Disable 1: Enable USB Software Sleeping Set and reset by software. Please refer to the Power Control Unit chapter for more information.
[0]	VDDISO	V_{DD} Domain Isolation Control 0: V_{DD} domain is isolated from other power domain 1: V_{DD} domain is accessible by other power domain Set and reset by software. Please refer to the Power Control Unit chapter for more information.

MCU Debug Control Register – MCUDBGCR

This register specifies debug control of MCU.

Offset: 0x304

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved						DBSCTM3	DBSCTM2
Type/Reset							RW 0	RW 0
	23	22	21	20	19	18	17	16
	DBSCTM1	DBSCTM0	Reserved			DBUR	DBBFTM1	DBBFTM0
Type/Reset	RW 0	RW 0				RW 0	RW 0	RW 0
	15	14	13	12	11	10	9	8
	Reserved	DBDSLP2	Reserved	DBI2C	DBQSPI	DBSPI	Reserved	DBUSR
Type/Reset		RW 0		RW 0	RW 0	RW 0		RW 0
	7	6	5	4	3	2	1	0
	Reserved	DBGPTM	Reserved		DBWDT	DBPD	DBDSLP1	DBSLP
Type/Reset		RW 0			RW 0	RW 0	RW 0	RW 0

Bits	Field	Descriptions
[25]	DBSCTM3	SCTM3 Debug Mode Enable 0: SCTM3 counter continues to count even if the core is halted 1: SCTM3 counter is stopped when the core is halted Set and reset by software.
[24]	DBSCTM2	SCTM2 Debug Mode Enable 0: SCTM2 counter continues to count even if the core is halted 1: SCTM2 counter is stopped when the core is halted Set and reset by software.
[23]	DBSCTM1	SCTM1 Debug Mode Enable 0: SCTM1 counter continues to count even if the core is halted 1: SCTM1 counter is stopped when the core is halted Set and reset by software.
[22]	DBSCTM0	SCTM0 Debug Mode Enable 0: SCTM0 counter continues to count even if the core is halted 1: SCTM0 counter is stopped when the core is halted Set and reset by software.
[18]	DBUR	UART Debug Mode Enable 0: Same behavior as in normal mode 1: UART FIFO timeout is frozen Set and reset by software.
[17]	DBBFTM1	BFTM1 Debug Mode Enable 0: BFTM1 counter continues to count even if the core is halted 1: BFTM1 counter is stopped when the core is halted Set and reset by software.
[16]	DBBFTM0	BFTM0 Debug Mode Enable 0: BFTM0 counter continues to count even if the core is halted 1: BFTM0 counter is stopped when the core is halted Set and reset by software.

Bits	Field	Descriptions
[14]	DBDSLP2	Debug Deep-Sleep2 Mode 0: LDO = Off (but turn on DMOS), FCLK = Off and HCLK = Off in Deep-Sleep2 Mode 1: LDO = On, FCLK = On and HCLK = On in Deep-Sleep2 Mode Set and reset by software.
[12]	DBI2C	I^2C Debug Mode Enable 0: Same behavior as in normal mode 1: I^2C timeout is frozen Set and reset by software.
[11]	DBQSPI	QSPI Debug Mode Enable 0: Same behavior as in normal mode 1: QSPI FIFO timeout is frozen Set and reset by software.
[10]	DBSPI	SPI Debug Mode Enable 0: Same behavior as in normal mode 1: SPI FIFO timeout is frozen Set and reset by software.
[8]	DBUSR	USART Debug Mode Enable 0: Same behavior as in normal mode 1: USART FIFO timeout is frozen Set and reset by software.
[6]	DBGPTM	GPTM Debug Mode Enable 0: GPTM counter continues even if the core is halted 1: GPTM counter is stopped when the core is halted Set and reset by software.
[3]	DBWDT	Watchdog Timer Debug Mode Enable 0: Watchdog Timer counter continues even if the core is halted 1: Watchdog Timer counter is stopped when the core is halted Set and reset by software.
[2]	DBPD	Debug Power-Down Mode 0: LDO = Off, FCLK = Off and HCLK = Off in Power-Down mode 1: LDO = On, FCLK = On and HCLK = On in Power-Down mode Set and reset by software.
[1]	DBDSLP1	Debug Deep-Sleep1 Mode 0: LDO = Low power mode, FCLK = Off and HCLK = Off in Deep-Sleep1 Mode 1: LDO = On, FCLK = On and HCLK = On in Deep-Sleep1 Mode Set and reset by software.
[0]	DBSLP	Debug Sleep Mode 0: LDO = On, FCLK = On and HCLK = Off in Sleep Mode 1: LDO = On, FCLK = On and HCLK = On in Sleep Mode Set and reset by software.

7 Reset Control Unit (RSTCU)

Introduction

The Reset Control Unit, RSTCU, has three kinds of reset, the power on reset, system reset and APB unit reset. The power on reset, known as a cold reset, resets the full system during a power up. A system reset resets the processor core and peripheral IP components with the exception of the debug port controller. The resets can be triggered by an external signal, internal events and the reset generators. More information about these resets will be described in the following section.

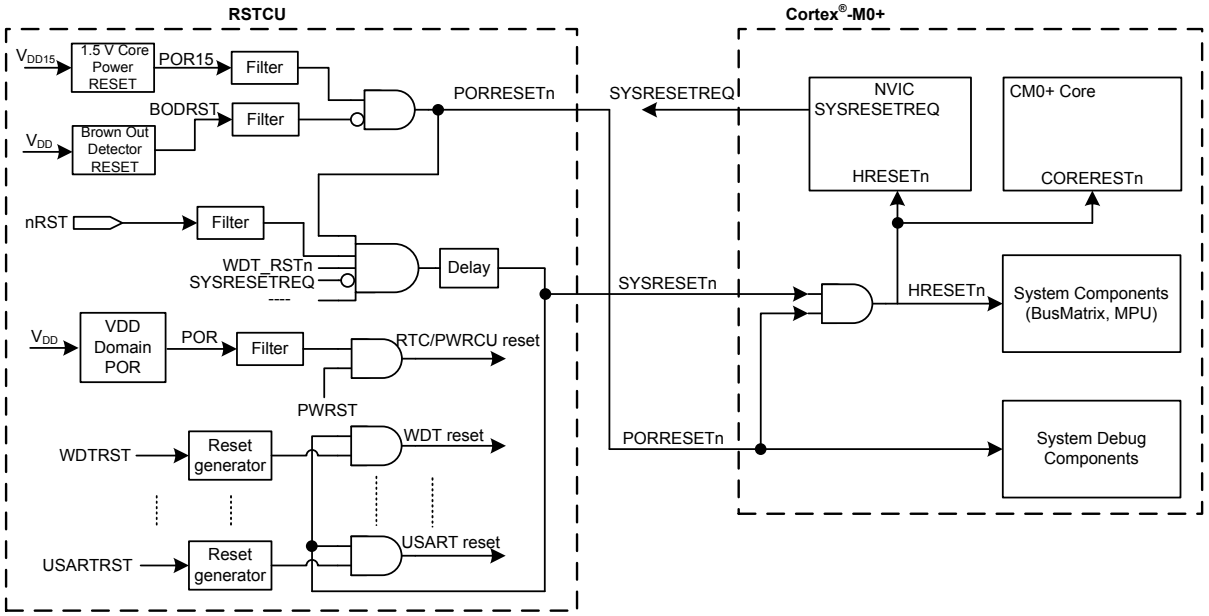


Figure 18. RSTCU Block Diagram

Functional Descriptions

Power On Reset

The Power on reset, POR, is generated by either an external reset or the internal reset generator. Both types have an internal filter to prevent glitches from causing erroneous reset operations. By referring to Figure 19 the POR15 active low signal will be de-asserted when the internal LDO voltage regulator is ready to provide a 1.5 V power. In addition to the POR15 signal, the Power Control Unit, PWRCU, will assert the BODF signal as a Power Down Reset, PDR, when the BODEN bit in the LVDCSR register is set and the brown-out event occurs. For more details about the PWRCU function, refer to the PWRCU chapter.

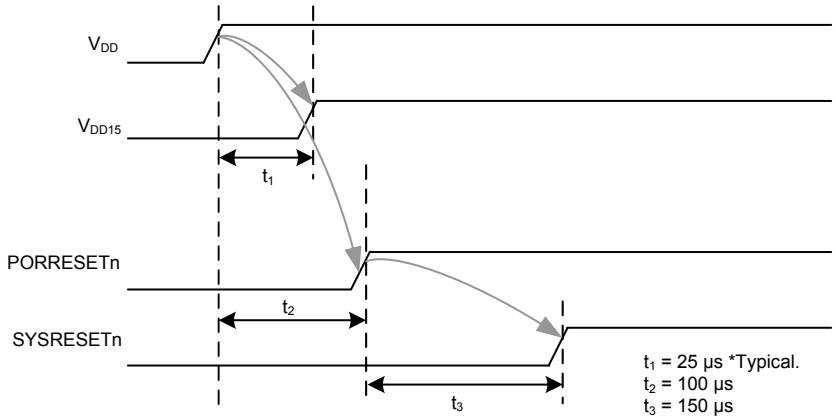


Figure 19. Power On Reset Sequence

System Reset

A system reset is generated by a power on reset (PORRESETn), a Watchdog Timer reset (WDT_RSTn), nRST pin or a software reset (SYSRESETREQ) event. For more information about SYSRESETREQ event, refer to the related chapter in the Cortex®-M0+ reference manual.

AHB and APB Unit Reset

The AHB and APB unit reset can be divided into hardware and software resets. A hardware reset can be generated by either power on reset or system reset for all AHB and APB units. Each functional IP connected to the AHB and APB buses can be reset individually through the associated software reset bits in the RSTCU. For example, the application software can generate a USART reset via the USRRST bit in the APBPRSTR0 register.

Register Map

The following table shows the RSTCU registers and reset values.

Table 19. RSTCU Register Map

Register	Offset	Description	Reset Value
RSTCU Base Address = 0x4008_8000			
GRSR	0x100	Global Reset Status Register	0x0000_0008
AHBPRSTR	0x104	AHB Peripheral Reset Register	0x0000_0000
APBPRSTR0	0x108	APB Peripheral Reset Register 0	0x0000_0000
APBPRSTR1	0x10C	APB Peripheral Reset Register 1	0x0000_0000

Register Descriptions

Global Reset Status Register – GRSR

This register specifies a variety of reset status conditions.

Offset: 0x100
Reset value: 0x0000_0008

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	22	21	20	19	18	17	16
	Reserved							
Type/Reset								
	15	14	13	12	11	10	9	8
	Reserved							
Type/Reset								
	7	6	5	4	3	2	1	0
	Reserved				PORSTF	WDTRSTF	EXTRSTF	NVICRSTF
Type/Reset					WC 1	WC 0	WC 0	WC 0

Bits	Field	Descriptions
[3]	PORSTF	Core 1.5 V Power On Reset Flag 0: No POR occurred 1: POR occurred This bit is set by hardware when a power on reset occurs and reset by writing 1 into it.
[2]	WDTRSTF	Watchdog Timer Reset Flag 0: No Watchdog Timer reset occurred 1: Watchdog Timer occurred This bit is set by hardware when a watchdog timer reset occurs and reset by writing 1 into it or by hardware when a power on reset occurs.
[1]	EXTRSTF	External Pin Reset Flag 0: No pin reset occurred 1: Pin reset occurred This bit is set by hardware when an external pin reset occurs and reset by writing 1 into it or by hardware when a power on reset occurs.

Bits	Field	Descriptions
[0]	NVICRSTF	NVIC Reset Flag 0: No NVIC asserting system reset occurred 1: NVIC asserting system reset occurred This bit is set by hardware when a system reset occurs and reset by writing 1 into it or by hardware when a power on reset occurs.

AHB Peripheral Reset Register – AHBPRSTR

This register specifies several AHB peripherals software reset control bits.

Offset: 0x104

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved						QSPIRST	DIVRST
Type/Reset							RW 0	RW 0
	23	22	21	20	19	18	17	16
	Reserved							
Type/Reset								
	15	14	13	12	11	10	9	8
	Reserved				PDRST	PCRST	PBRST	PARST
Type/Reset					RW 0	RW 0	RW 0	RW 0
	7	6	5	4	3	2	1	0
	CRCRST	Reserved	USBRST	Reserved				DMARST
Type/Reset	RW 0		RW 0					RW 0

Bits	Field	Descriptions
[25]	QSPIRST	QSPI Reset Control 0: No reset 1: Reset QSPI This bit is set by software and cleared to 0 by hardware automatically.
[24]	DIVRST	Divider Reset Control 0: No reset 1: Reset Divider This bit is set by software and cleared to 0 by hardware automatically.
[11]	PDRST	GPIO Port D Reset Control 0: No reset 1: Reset Port D This bit is set by software and cleared to 0 by hardware automatically.
[10]	PCRST	GPIO Port C Reset Control 0: No reset 1: Reset Port C This bit is set by software and cleared to 0 by hardware automatically.
[9]	PBRST	GPIO Port B Reset Control 0: No reset 1: Reset Port B This bit is set by software and cleared to 0 by hardware automatically.

Bits	Field	Descriptions
[8]	PARST	GPIO Port A Reset Control 0: No reset 1: Reset Port A This bit is set by software and cleared to 0 by hardware automatically.
[7]	CRCRST	CRC Reset Control 0: No reset 1: Reset CRC This bit is set by software and cleared to 0 by hardware automatically.
[5]	USBRST	USB Reset Control 0: No reset 1: Reset USB This bit is set by software and cleared to 0 by hardware automatically.
[0]	DMARST	Peripheral DMA (PDMA) Reset Control 0: No reset 1: Reset Peripheral DMA (PDMA) This bit is set by software and cleared to 0 by hardware automatically.

APB Peripheral Reset Register 0 – APBPRSTR0

This register specifies several APB peripherals software reset control bits.

Offset: 0x108

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved			MIDIRST		Reserved	I2SRST	Reserved
Type/Reset				RW 0			RW 0	
	23	22	21	20	19	18	17	16
	Reserved							
Type/Reset								
	15	14	13	12	11	10	9	8
	EXTIRST	AFIORST			Reserved	URRST	Reserved	USRRST
Type/Reset	RW 0	RW 0				RW 0		RW 0
	7	6	5	4	3	2	1	0
	Reserved			SPIRST	Reserved			I2CRST
Type/Reset				RW 0				RW 0

Bits	Field	Descriptions
[28]	MIDIRST	MIDI Reset Control 0: No reset 1: Reset MIDI This bit is set by software and cleared to 0 by hardware automatically.
[25]	I2SRST	I^2S Reset Control 0: No reset 1: Reset I^2S This bit is set by software and cleared to 0 by hardware automatically.

Bits	Field	Descriptions
[15]	EXTIRST	External Interrupt Controller Reset Control 0: No reset 1: Reset EXTI This bit is set by software and cleared to 0 by hardware automatically.
[14]	AFIORST	Alternate Function I/O Reset Control 0: No reset 1: Reset Alternate Function I/O This bit is set by software and cleared to 0 by hardware automatically.
[10]	URRST	UART Reset Control 0: No reset 1: Reset UART This bit is set by software and cleared to 0 by hardware automatically.
[8]	USRRST	USART Reset Control 0: No reset 1: Reset USART This bit is set by software and cleared to 0 by hardware automatically.
[4]	SPIRST	SPI Reset Control 0: No reset 1: Reset SPI This bit is set by software and cleared to 0 by hardware automatically.
[0]	I2CRST	I^2C Reset Control 0: No reset 1: Reset I^2C This bit is set by software and cleared to 0 by hardware automatically.

APB Peripheral Reset Register 1 – APBPRSTR1

This register specifies several APB peripherals software reset control bits.

Offset: 0x10C

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	SCTM3RST	SCTM2RST	SCTM1RST	SCTM0RST	Reserved			ADCRST
Type/Reset	RW 0	RW 0	RW 0	RW 0				RW 0
	23	22	21	20	19	18	17	16
	Reserved						BFTM1RST	BFTM0RST
Type/Reset							RW 0	RW 0
	15	14	13	12	11	10	9	8
	Reserved							GPTMRST
Type/Reset								RW 0
	7	6	5	4	3	2	1	0
	Reserved			WDTRST	Reserved			
Type/Reset				RW 0				

Bits	Field	Descriptions
[31]	SCTM3RST	SCTM3 Reset Control 0: No reset 1: Reset SCTM3 This bit is set by software and cleared to 0 by hardware automatically.
[30]	SCTM2RST	SCTM2 Reset Control 0: No reset 1: Reset SCTM2 This bit is set by software and cleared to 0 by hardware automatically.
[29]	SCTM1RST	SCTM1 Reset Control 0: No reset 1: Reset SCTM1 This bit is set by software and cleared to 0 by hardware automatically.
[28]	SCTM0RST	SCTM0 Reset Control 0: No reset 1: Reset SCTM0 This bit is set by software and cleared to 0 by hardware automatically.
[24]	ADCRST	A/D Converter Reset Control 0: No reset 1: Reset A/D Converter This bit is set by software and cleared to 0 by hardware automatically.
[17]	BFTM1RST	BFTM1 Reset Control 0: No reset 1: Reset BFTM1 This bit is set by software and cleared to 0 by hardware automatically.
[16]	BFTM0RST	BFTM0 Reset Control 0: No reset 1: Reset BFTM0 This bit is set by software and cleared to 0 by hardware automatically.
[8]	GPTMRST	GPTM Reset Control 0: No reset 1: Reset GPTM This bit is set by software and cleared to 0 by hardware automatically.
[4]	WDTRST	Watchdog Timer Reset Control 0: No reset 1: Reset Watchdog Timer This bit is set by software and cleared to 0 by hardware automatically.

8 General Purpose I/O (GPIO)

Introduction

There are up to 52 General Purpose I/O port, GPIO, named PA0 ~ PA15, PB0 ~ PB15, PC0 ~ PC15 and PD0 ~ PD3 for the device to implement the logic input / output functions. Each of the GPIO ports has related control and configuration registers to satisfy the requirement of specific applications. The actual available General Purpose I/O port numbers are dependent on the device specification and package type. Refer to the device datasheet for detailed information.

The GPIO ports are pin-shared with other alternative functions (AFs) to obtain maximum flexibility on the package pins. The GPIO pins can be used as alternative functional pins by configuring the corresponding registers regardless of the AF input or output pins.

The external interrupts on the GPIO pins of the device have related control and configuration registers in the External Interrupt Control Unit (EXTI).

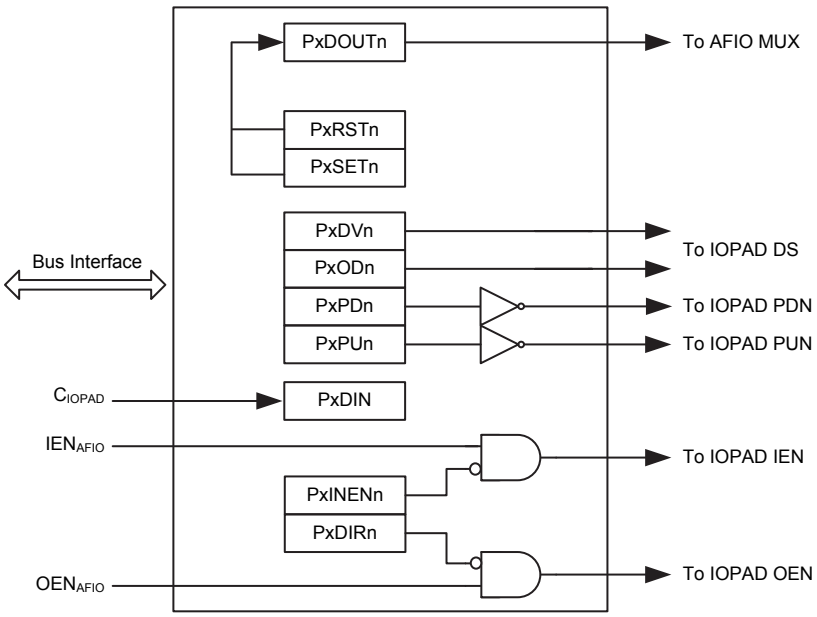


Figure 20. GPIO Block Diagram

Features

- Input / output direction control
- Schmitt Trigger Input function enable control
- Input weak pull-up / pull-down control
- Output push-pull / open-drain enable control
- Output set / reset control
- Output drive current selection
- External interrupt with programmable trigger edge – Using EXTI configuration registers
- Analog input / output configurations – Using AFIO configuration registers
- Alternate function input / output configurations – Using AFIO configuration registers
- Port configuration lock

Functional Descriptions

Default GPIO Pin Configuration

During or just after the reset period, the alternative functions are all inactive and the GPIO ports are configured into the input disable floating mode, i.e. input disabled without pull-up / pull-down resistors. Only the boot and Serial-Wired Debug pins which are pin-shared with the I/O pins are active after a device reset.

- PA9_BOOT: Input enable with internal pull-up
- SWCLK: Input enable with internal pull-up
- SWDIO: Input enable with internal pull-up

General Purpose I/O – GPIO

The GPIO pins can be configured as inputs or outputs via the data direction control registers PxDIRCR (where x = A ~ D). When the GPIO pins are configured as input pins, the data on the external pads can be read if the enable bits in the input enable function register PxINER are set. The GPIO pull-up / pull-down registers PxPUR / PxPDR can be configured to fit specific applications. When the pull-up and pull-down functions are both enabled, the pull-up function has the higher priority while the pull-down function will be blocked until the pull-up function is released.

The GPIO pins can be configured as output pins where the output data is latched into the data register PxDOUTR. The output type can be setup to be either push-pull or open-drain by the open-drain selection register PxODR. Only one or several specific bits of the output data will be set or reset by configuring the port output set / reset control register PxSRR or the port output reset control register PxRR without affecting the unselected bits. As the port output set and reset functions are both enabled, the port output set function has the higher priority and the port output reset function will be blocked. The output driving current of the GPIO pins can be selected by configuring the drive current selection register PxDRVR.

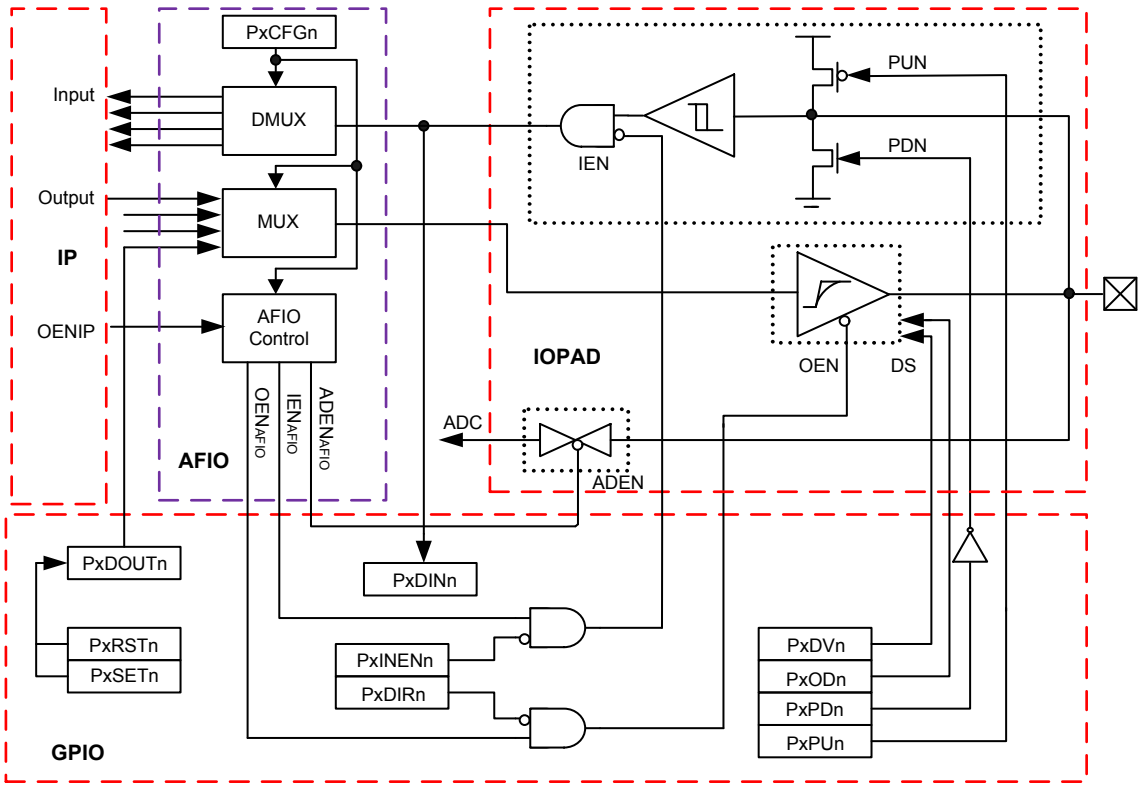


Figure 21. AFIO / GPIO Control Signal

PxDINn / PxDOUTn (x = A ~ D): Data Input / Data Output

PxRSTn / PxSETn (x = A ~ D): Reset / Set

PxDIRn (x = A ~ D): Direction

PxINENn (x = A ~ D): Input Enable

PxDVn (x = A ~ D): Output Drive

PxODn (x = A ~ D): Open Drain

PxPDn / PxPUn (x = A ~ D): Pull Down / Up

PxCFGn (x = A ~ D): AFIO Configuration

Table 20. AFIO, GPIO and IO Pad Control Signal True Table

Type	AFIO			GPIO		PAD		
	$ADEN_{AFIO}$	OEN_{AFIO}	IEN_{AFIO}	PxDIRn	PxINENn	ADEN	OEN	IEN
GPIO Input (Note)	1	1	1	0	1	1	1	0
GPIO Output (Note)	1	1	1	1	0 (1 if need)	1	0	1 (0)
AFIO Input	1	1	0	0	X	1	1	0
AFIO Output	1	0	1	X	0 (1 if need)	1	0	1 (0)
ADC Input	0	1	1	0	0 (1 if need)	0	1	1 (0)
OSC Output	0	1	1	0	0 (1 if need)	0	1	1 (0)

Note: The signals, IEN and OEN, for I/O pads are derived from the GPIO register bits PxINENn and PxDIRn respectively when the associated pin is configured in the GPIO input / output mode.

GPIO Locking Mechanism

The GPIO also offers a lock function to lock the port until a reset event occurs. The PxLOCKR (x = A ~ D) registers are used to lock the port x and lock control options. The value 0x5FA0 is written into the PxLKEY field in the PxLOCKR registers to freeze the PxDIRCR, PxINER, PxPUR, PxPDR, PxODR, PxDRVR control and AFIO mode configuration (GPxCFGHR or GPxCFGLR, where x = A ~ D). If the value in the PxLOCKR register is 0x5FA0_0001, it means that the Port x Lock function is enabled and the Port x pin 0 is frozen.

Register Map

The following table shows the GPIO registers and reset values of the Port A ~ D.

Table 21. GPIO Register Map

Register	Offset	Description	Reset Value
GPIO A Base Address = 0x400B_0000			
PADIRCR	0x000	Port A Data Direction Control Register	0x0000_0000
PAINER	0x004	Port A Input Function Enable Control Register	0x0000_3300
PAPUR	0x008	Port A Pull-Up Selection Register	0x0000_3300
PAPDR	0x00C	Port A Pull-Down Selection Register	0x0000_0000
PAODR	0x010	Port A Open-Drain Selection Register	0x0000_0000
PADRVR	0x014	Port A Drive Current Selection Register	0x0000_0000
PALOCKR	0x018	Port A Lock Register	0x0000_0000
PADINR	0x01C	Port A Data Input Register	0x0000_3300
PADOUTR	0x020	Port A Data Output Register	0x0000_0000
PASRR	0x024	Port A Output Set / Reset Control Register	0x0000_0000
PARR	0x028	Port A Output Reset Control Register	0x0000_0000
GPIO B Base Address = 0x400B_2000			
PBDIRCR	0x000	Port B Data Direction Control Register	0x0000_0000
PBINER	0x004	Port B Input Function Enable Control Register	0x0000_0000
PBPUR	0x008	Port B Pull-Up Selection Register	0x0000_0000
PBPDR	0x00C	Port B Pull-Down Selection Register	0x0000_0000
PBODR	0x010	Port B Open-Drain Selection Register	0x0000_0000
PBDRVR	0x014	Port B Drive Current Selection Register	0x0000_0000
PBLOCKR	0x018	Port B Lock Register	0x0000_0000
PBDINR	0x01C	Port B Data Input Register	0x0000_0000
PBDOUTR	0x020	Port B Data Output Register	0x0000_0000
PBSRR	0x024	Port B Output Set / Reset Control Register	0x0000_0000
PBRR	0x028	Port B Output Reset Control Register	0x0000_0000
GPIO C Base Address = 0x400B_4000			
PCDIRCR	0x000	Port C Data Direction Control Register	0x0000_0000
PCINER	0x004	Port C Input Function Enable Control Register	0x0000_0000
PCPUR	0x008	Port C Pull-Up Selection Register	0x0000_0000
PCPDR	0x00C	Port C Pull-Down Selection Register	0x0000_0000
PCODR	0x010	Port C Open-Drain Selection Register	0x0000_0000
PCDRVR	0x014	Port C Drive Current Selection Register	0x0000_0000
PCLOCKR	0x018	Port C Lock Register	0x0000_0000
PCDINR	0x01C	Port C Data Input Register	0x0000_0000
PCDOUTR	0x020	Port C Data Output Register	0x0000_0000
PCSRR	0x024	Port C Output Set / Reset Control Register	0x0000_0000
PCRR	0x028	Port C Output Reset Control Register	0x0000_0000

Register	Offset	Description	Reset Value
GPIO D Base Address = 0x400B_6000			
PDDIRCR	0x000	Port D Data Direction Control Register	0x0000_0000
PDINER	0x004	Port D Input Function Enable Control Register	0x0000_0000
PDPUR	0x008	Port D Pull-Up Selection Register	0x0000_0000
PDPDR	0x00C	Port D Pull-Down Selection Register	0x0000_0000
PDODR	0x010	Port D Open-Drain Selection Register	0x0000_0000
PDDRVR	0x014	Port D Drive Current Selection Register	0x0000_0000
PDLOCKR	0x018	Port D Lock Register	0x0000_0000
PDDINR	0x01C	Port D Data Input Register	0x0000_0000
PDDOUTR	0x020	Port D Data Output Register	0x0000_0000
PDSRR	0x024	Port D Output Set / Reset Control Register	0x0000_0000
PDRR	0x028	Port D Output Reset Control Register	0x0000_0000

Register Descriptions

Port A Data Direction Control Register – PADIRCR

This register is used to control the direction of the GPIO Port A pin as input or output.

Offset: 0x000
Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
	Reserved							
Type/Reset								
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
	PADIR							
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
	PADIR							
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0

Bits	Field	Descriptions
[15:0]	PADIRn	GPIO Port A pin n Direction Control Bits (n = 0 ~ 15) 0: Pin n is in input mode 1: Pin n is in output mode

Port A Input Function Enable Control Register – PAINER

This register is used to enable or disable the GPIO Port A input function.

Offset: 0x004

Reset value: 0x0000_3300

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	22	21	20	19	18	17	16
	Reserved							
Type/Reset								
	15	14	13	12	11	10	9	8
	PAINEN							
Type/Reset	RW 0	RW 0	RW 1	RW 1	RW 0	RW 0	RW 1	RW 1
	7	6	5	4	3	2	1	0
	PAINEN							
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0

Bits	Field	Descriptions
[15:0]	PAINENn	GPIO Port A pin n Input Enable Control Bits (n = 0 ~ 15) 0: Pin n input function is disabled 1: Pin n input function is enabled When the pin n input function is disabled, the input Schmitt trigger will be turned off and the Schmitt trigger output will remain at a zero state.

Port A Pull-Up Selection Register – PAPUR

This register is used to enable or disable the GPIO Port A pull-up function.

Offset: 0x008

Reset value: 0x0000_3300

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	22	21	20	19	18	17	16
	Reserved							
Type/Reset								
	15	14	13	12	11	10	9	8
	PAPU							
Type/Reset	RW 0	RW 0	RW 1	RW 1	RW 0	RW 0	RW 1	RW 1
	7	6	5	4	3	2	1	0
	PAPU							
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0

Bits	Field	Descriptions
[15:0]	PAPUn	GPIO Port A pin n Pull-Up Selection Control Bits (n = 0 ~ 15) 0: Pin n pull-up function is disabled 1: Pin n pull-up function is enabled Note: When the pull-up and pull-down functions are both enabled, the pull-up function will have the higher priority and therefore the pull-down function will be blocked and disabled.

Port A Pull-Down Selection Register – PAPDR

This register is used to enable or disable the GPIO Port A pull-down function.

Offset: 0x00C

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	22	21	20	19	18	17	16
	Reserved							
Type/Reset								
	15	14	13	12	11	10	9	8
	PAPD							
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0
	7	6	5	4	3	2	1	0
	PAPD							
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0

Bits	Field	Descriptions
[15:0]	PAPDn	GPIO Port A pin n Pull-Down Selection Control Bits (n = 0 ~ 15) 0: Pin n pull-down function is disabled 1: Pin n pull-down function is enabled Note: When the pull-up and pull-down functions are both enabled, the pull-up function will have the higher priority and therefore the pull-down function will be blocked and disabled.

Port A Open-Drain Selection Register – PAODR

This register is used to enable or disable the GPIO Port A open-drain function.

Offset: 0x010

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
	Reserved							
Type/Reset								
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
	PAOD							
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
	PAOD							
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0

Bits	Field	Descriptions
[15:0]	PAODn	GPIO Port A pin n Open-Drain Selection Control Bits (n = 0 ~ 15) 0: Pin n Open-Drain output is disabled (The output type is CMOS output) 1: Pin n Open-Drain output is enabled (The output type is open-drain output)

Port A Output Drive Current Selection Register – PADRVR

This register specifies the GPIO Port A output driving current.

Offset: 0x014

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	PADV15		PADV14		PADV13		PADV12	
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0
	23	22	21	20	19	18	17	16
	PADV11		PADV10		PADV9		PADV8	
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0
	15	14	13	12	11	10	9	8
	PADV7		PADV6		PADV5		PADV4	
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0
	7	6	5	4	3	2	1	0
	PADV3		PADV2		PADV1		PADV0	
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0

Bits	Field	Descriptions
[31:0]	PADVn[1:0]	GPIO Port A pin n Output Current Drive Selection Control Bits (n = 0 ~ 15) 00: 4 mA source / sink current 01: 8 mA source / sink current 10: 12 mA source / sink current 11: 16 mA source / sink current

Port A Lock Register – PALOCKR

This register specifies the GPIO Port A lock configuration.

Offset: 0x018

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	PALKEY							
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
	PALKEY							
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
	PALOCK							
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
	PALOCK							
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0

Bits	Field	Descriptions
[31:16]	PALKEY	GPIO Port A Lock Key 0x5FA0: Port A Lock function is enabled Others: Port A Lock function is disabled To lock the Port A function, a value of 0x5FA0 should be written into the PALKEY field in this register. To execute a successful write operation on this lock register, the value written into the PALKEY field must be 0x5FA0. If the value written into this field is not equal to 0x5FA0, any write operations on the PALOCKR register will be aborted. The result of a read operation on the PALKEY field returns the GPIO Port A Lock Status which indicates whether the GPIO Port A is locked or not. If the read value of the PALKEY field is 0, this indicates that the GPIO Port A Lock function is disabled. Otherwise, it indicates that the GPIO Port A Lock function is enabled as the read value is equal to 1.
[15:0]	PALOCKn	GPIO Port A Pin n Lock Control Bits (n = 0 ~ 15) 0: Port A Pin n is not locked 1: Port A Pin n is locked The PALOCKn bits are used to lock the configurations of corresponding GPIO Pins when the correct Lock Key is applied to the PALKEY field. The locked configurations including PADIRn, PAINENn, PAPUn, PAPDn, PAODn and PADVn setting in the related GPIO registers. Additionally, the GPACFGHR or GPACFGLR register which is used to configure the alternative function of the associated GPIO pin will also be locked. Note that the PALOCKR register can only be written once which means that PALKEY and PALOCKn (lock control bit) should be written together and can not be changed until a system reset or GPIO Port A reset occurs.

Port A Data Input Register – PADINR

This register specifies the GPIO Port A input data.

Offset: 0x01C

Reset value: 0x0000_3200

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	22	21	20	19	18	17	16
	Reserved							
Type/Reset								
	15	14	13	12	11	10	9	8
	PADIN							
Type/Reset	RO 0	RO 0	RO 1	RO 1	RO 0	RO 0	RO 1	RO 1
	7	6	5	4	3	2	1	0
	PADIN							
Type/Reset	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0

Bits	Field	Descriptions
[15:0]	PADINn	GPIO Port A pin n Data Input Bits (n = 0 ~ 15) 0: The input data of the corresponding pin is 0 1: The input data of the corresponding pin is 1

Port A Output Data Register – PADOUTR

This register specifies the GPIO Port A output data.

Offset: 0x020

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	22	21	20	19	18	17	16
	Reserved							
Type/Reset								
	15	14	13	12	11	10	9	8
	PADOUT							
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0
	7	6	5	4	3	2	1	0
	PADOUT							
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0

Bits	Field	Descriptions
[15:0]	PADOUTn	GPIO Port A pin n Data Output Bits (n = 0 ~ 15) 0: Data to be output on pin n is 0 1: Data to be output on pin n is 1

Port A Output Set / Reset Control Register – PASRR

This register is used to set or reset the corresponding bit of the GPIO Port A output data.

Offset: 0x024

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	PARST							
Type/Reset	WO 0	WO 0	WO 0	WO 0	WO 0	WO 0	WO 0	WO 0
	23	22	21	20	19	18	17	16
	PARST							
Type/Reset	WO 0	WO 0	WO 0	WO 0	WO 0	WO 0	WO 0	WO 0
	15	14	13	12	11	10	9	8
	PASET							
Type/Reset	WO 0	WO 0	WO 0	WO 0	WO 0	WO 0	WO 0	WO 0
	7	6	5	4	3	2	1	0
	PASET							
Type/Reset	WO 0	WO 0	WO 0	WO 0	WO 0	WO 0	WO 0	WO 0

Bits	Field	Descriptions
[31:16]	PARSTn	GPIO Port A pin n Output Reset Control Bits (n = 0 ~ 15) 0: No effect on the PADOUTn bit 1: Reset the PADOUTn bit Note that when the PARSTn bit in this register or (and) the PARSTn bit in the PARR register is enabled, the reset function on the PADOUTn bit will take effect.
[15:0]	PASETn	GPIO Port A pin n Output Set Control Bits (n = 0 ~ 15) 0: No effect on the PADOUTn bit 1: Set the PADOUTn bit Note that the function enabled by the PASETn bit has the higher priority if both the PASETn and PARSTn bits are set at the same time.

Port A Output Reset Register – PARR

This register is used to reset the corresponding bit of the GPIO Port A output data.

Offset: 0x028

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	22	21	20	19	18	17	16
	Reserved							
Type/Reset								
	15	14	13	12	11	10	9	8
	PARST							
Type/Reset	WO 0	WO 0	WO 0	WO 0	WO 0	WO 0	WO 0	WO 0
	7	6	5	4	3	2	1	0
	PARST							
Type/Reset	WO 0	WO 0	WO 0	WO 0	WO 0	WO 0	WO 0	WO 0

Bits	Field	Descriptions
[15:0]	PARSTn	GPIO Port A pin n Output Reset Control Bits (n = 0 ~ 15) 0: No effect on the PADOUTn bit 1: Reset the PADOUTn bit

Port B Data Direction Control Register – PBDIRCR

This register is used to control the direction of GPIO Port B pin as input or output.

Offset: 0x000

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	22	21	20	19	18	17	16
	Reserved							
Type/Reset								
	15	14	13	12	11	10	9	8
	PBDIR							
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0
	7	6	5	4	3	2	1	0
	PBDIR							
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0

Bits	Field	Descriptions
[15:0]	PBDIRn	GPIO Port B pin n Direction Control Bits (n = 0 ~ 15) 0: Pin n is in input mode 1: Pin n is in output mode

Port B Input Function Enable Control Register – PBINER

This register is used to enable or disable the GPIO Port B input function.

Offset: 0x004

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	22	21	20	19	18	17	16
	Reserved							
Type/Reset								
	15	14	13	12	11	10	9	8
	PBINEN							
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0
	7	6	5	4	3	2	1	0
	PBINEN							
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0

Bits	Field	Descriptions
[15:0]	PBINENn	GPIO Port B pin n Input Enable Control Bits (n = 0 ~ 15) 0: Pin n input function is disabled 1: Pin n input function is enabled When the pin n input function is disabled, the input Schmitt trigger will be turned off and the Schmitt trigger output will remain at a zero state.

Port B Pull-Up Selection Register – PBPUR

This register is used to enable or disable the GPIO Port B pull-up function.

Offset: 0x008

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	22	21	20	19	18	17	16
	Reserved							
Type/Reset								
	15	14	13	12	11	10	9	8
	PBPU							
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0
	7	6	5	4	3	2	1	0
	PBPU							
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0

Bits	Field	Descriptions
[15:0]	PBPUn	GPIO Port B pin n Pull-Up Selection Control Bits (n = 0 ~ 15) 0: Pin n pull-up function is disabled 1: Pin n pull-up function is enabled Note: When the pull-up and pull-down functions are both enabled, the pull-up function will have the higher priority and therefore the pull-down function will be blocked and disabled.

Port B Pull-Down Selection Register – PBPDR

This register is used to enable or disable the GPIO Port B pull-down function.

Offset: 0x00C

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	22	21	20	19	18	17	16
	Reserved							
Type/Reset								
	15	14	13	12	11	10	9	8
	PBPD							
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0
	7	6	5	4	3	2	1	0
	PBPD							
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0

Bits	Field	Descriptions
[15:0]	PBPDn	GPIO Port B pin n Pull-Down Selection Control Bits (n = 0 ~ 15) 0: Pin n pull-down function is disabled 1: Pin n pull-down function is enabled Note: When the pull-up and pull-down functions are both enabled, the pull-up function will have the higher priority and therefore the pull-down function will be blocked and disabled.

Port B Open Drain Selection Register – PBODR

This register is used to enable or disable the GPIO Port B open-drain function.

Offset: 0x010

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	22	21	20	19	18	17	16
	Reserved							
Type/Reset								
	15	14	13	12	11	10	9	8
	PBOD							
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0
	7	6	5	4	3	2	1	0
	PBOD							
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0

Bits	Field	Descriptions
[15:0]	PBODn	GPIO Port B pin n Open Drain Selection Control Bits (n = 0 ~ 15) 0: Pin n Open Drain output is disabled. (The output type is CMOS output) 1: Pin n Open Drain output is enabled. (The output type is open-drain output)

Port B Output Drive Current Selection Register – PBDRVR

This register specifies the GPIO Port B output driving current.

Offset: 0x014

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	PBDV15		PBDV14		PBDV13		PBDV12	
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0
	23	22	21	20	19	18	17	16
	PBDV11		PBDV10		PBDV9		PBDV8	
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0
	15	14	13	12	11	10	9	8
	PBDV7		PBDV6		PBDV5		PBDV4	
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0
	7	6	5	4	3	2	1	0
	PBDV3		PBDV2		PBDV1		PBDV0	
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0

Bits	Field	Descriptions
[31:0]	PBDVn[1:0]	GPIO Port B pin n Output Current Drive Selection Control Bits (n = 0 ~ 15) 00: 4 mA source / sink current 01: 8 mA source / sink current 10: 12 mA source / sink current 11: 16 mA source / sink current

Port B Lock Register – PBLOCKR

This register specifies the GPIO Port B lock configuration.

Offset: 0x018

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	PBLKEY							
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0
	23	22	21	20	19	18	17	16
	PBLKEY							
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0
	15	14	13	12	11	10	9	8
	PBLOCK							
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0
	7	6	5	4	3	2	1	0
	PBLOCK							
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0

Bits	Field	Descriptions
[31:16]	PBLKEY	GPIO Port B Lock Key 0x5FA0: Port B Lock function is enabled Others: Port B Lock function is disabled To lock the Port B function, a value of 0x5FA0 should be written into the PBLKEY field in this register. To execute a successful write operation on this lock register, the value written into the PBLKEY field must be 0x5FA0. If the value written into this field is not equal to 0x5FA0, any write operations on the PBLOCKR register will be aborted. The result of a read operation on the PBLKEY field returns the GPIO Port B Lock Status which indicates whether the GPIO Port B is locked or not. If the read value of the PBLKEY field is 0, this indicates that the GPIO Port B Lock function is disabled. Otherwise, it indicates that the GPIO Port B Lock function is enabled as the read value is equal to 1.
[15:0]	PBLOCKn	GPIO Port B pin n Lock Control Bits (n = 0 ~ 15) 0: Port B pin n is not locked 1: Port B pin n is locked The PBLOCKn bits are used to lock the configurations of corresponding GPIO Pins when the correct Lock Key is applied to the PBLKEY field. The locked configurations including PBDIRn, PBINENn, PBPUn, PBPDn, PBODn and PBDVn setting in the related GPIO registers. Additionally, the GPBCFGHR or GPBCFGLR register which is used to configure the alternative function of the associated GPIO pin will also be locked. Note that the PBLOCKR register can only be written once which means that PBLKEY and PBLOCKn (lock control bit) should be written together and can not be changed until a system reset or GPIO Port B reset occurs.

Port B Data Input Register – PBDINR

This register specifies the GPIO Port B input data.

Offset: 0x01C

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	22	21	20	19	18	17	16
	Reserved							
Type/Reset								
	15	14	13	12	11	10	9	8
	PBDIN							
Type/Reset	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0
	7	6	5	4	3	2	1	0
	PBDIN							
Type/Reset	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0

Bits	Field	Descriptions
[15:0]	PBDINn	GPIO Port B pin n Data Input Bits (n = 0 ~ 15) 0: The input data of the corresponding pin is 0 1: The input data of the corresponding pin is 1

Port B Output Data Register – PBDOUTR

This register specifies the GPIO Port B output data.

Offset: 0x020

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	22	21	20	19	18	17	16
	Reserved							
Type/Reset								
	15	14	13	12	11	10	9	8
	PBDOUT							
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0
	7	6	5	4	3	2	1	0
	PBDOUT							
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0

Bits	Field	Descriptions
[15:0]	PBDOUTn	GPIO Port B pin n Data Output Bits (n = 0 ~ 15) 0: Data to be output on pin n is 0 1: Data to be output on pin n is 1

Port B Output Set / Reset Control Register – PBSRR

This register is used to set or reset the corresponding bit of the GPIO Port B output data.

Offset: 0x024

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	PBRST							
Type/Reset	WO 0	WO 0	WO 0	WO 0	WO 0	WO 0	WO 0	WO 0
	23	22	21	20	19	18	17	16
	PBRST							
Type/Reset	WO 0	WO 0	WO 0	WO 0	WO 0	WO 0	WO 0	WO 0
	15	14	13	12	11	10	9	8
	PBSET							
Type/Reset	WO 0	WO 0	WO 0	WO 0	WO 0	WO 0	WO 0	WO 0
	7	6	5	4	3	2	1	0
	PBSET							
Type/Reset	WO 0	WO 0	WO 0	WO 0	WO 0	WO 0	WO 0	WO 0

Bits	Field	Descriptions
[31:16]	PBRSTn	GPIO Port B pin n Output Reset Control Bits (n = 0 ~ 15) 0: No effect on the PBDOUTn bit 1: Reset the PBDOUTn bit Note that when the PBRSTn bit in this register or (and) the PBRSTn bit in the PBRR register is enabled, the reset function on the PBDOUTn bit will take effect.
[15:0]	PBSETn	GPIO Port B pin n Output Set Control Bits (n = 0 ~ 15) 0: No effect on the PBDOUTn bit 1: Set the PBDOUTn bit Note that the function enabled by the PBSETn bit has the higher priority if both the PBSETn and PBRSTn bits are set at the same time.

Port B Output Reset Register – PBRR

This register is used to reset the corresponding bit of the GPIO Port B output data.

Offset: 0x028

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	22	21	20	19	18	17	16
	Reserved							
Type/Reset								
	15	14	13	12	11	10	9	8
	PBRST							
Type/Reset	WO 0	WO 0	WO 0	WO 0	WO 0	WO 0	WO 0	WO 0
	7	6	5	4	3	2	1	0
	PBRST							
Type/Reset	WO 0	WO 0	WO 0	WO 0	WO 0	WO 0	WO 0	WO 0

Bits	Field	Descriptions
[15:0]	PBRSTn	GPIO Port B pin n Output Reset Control Bits (n = 0 ~ 15) 0: No effect on the PBDOUTn bit 1: Reset the PBDOUTn bit

Port C Data Direction Control Register – PCDIRCR

This register is used to control the direction of GPIO Port C pin as input or output.

Offset: 0x000

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	22	21	20	19	18	17	16
	Reserved							
Type/Reset								
	15	14	13	12	11	10	9	8
	PCDIR							
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0
	7	6	5	4	3	2	1	0
	PCDIR							
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0

Bits	Field	Descriptions
[15:0]	PCDIRn	GPIO Port C pin n Direction Control Bits (n = 0 ~ 15) 0: Pin n is in input mode 1: Pin n is in output mode

Port C Input Function Enable Control Register – PCINER

This register is used to enable or disable the GPIO Port C input function.

Offset: 0x004

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	22	21	20	19	18	17	16
	Reserved							
Type/Reset								
	15	14	13	12	11	10	9	8
	PCINEN							
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0
	7	6	5	4	3	2	1	0
	PCINEN							
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0

Bits	Field	Descriptions
[15:0]	PCINENn	GPIO Port C pin n Input Enable Control Bits (n = 0 ~ 15) 0: Pin n input function is disabled. 1: Pin n input function is enabled. When the pin n input function is disabled, the input Schmitt trigger will be turned off and the Schmitt trigger output will remain at a zero state.

Port C Pull-Up Selection Register – PCPUR

This register is used to enable or disable the GPIO Port C pull-up function.

Offset: 0x008

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	22	21	20	19	18	17	16
	Reserved							
Type/Reset								
	15	14	13	12	11	10	9	8
	PCPU							
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0
	7	6	5	4	3	2	1	0
	PCPU							
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0

Bits	Field	Descriptions
[15:0]	PCPUn	GPIO Port C pin n Pull-Up Selection Control Bits (n = 0 ~ 15) 0: Pin n pull-up function is disabled 1: Pin n pull-up function is enabled Note: When the pull-up and pull-down functions are both enabled, the pull-up function will have the higher priority and therefore the pull-down function will be blocked and disabled.

Port C Pull-Down Selection Register – PCPDR

This register is used to enable or disable the GPIO Port C pull-down function.

Offset: 0x00C

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	22	21	20	19	18	17	16
	Reserved							
Type/Reset								
	15	14	13	12	11	10	9	8
	PCPD							
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0
	7	6	5	4	3	2	1	0
	PCPD							
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0

Bits	Field	Descriptions
[15:0]	PCPDn	GPIO Port C pin n Pull-Down Selection Control Bits (n = 0 ~ 15) 0: Pin n pull-down function is disabled 1: Pin n pull-down function is enabled Note: When the pull-up and pull-down functions are both enabled, the pull-up function will have the higher priority and therefore the pull-down function will be blocked and disabled.

Port C Open Drain Selection Register – PCODR

This register is used to enable or disable the GPIO Port C open drain function.

Offset: 0x010

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	22	21	20	19	18	17	16
	Reserved							
Type/Reset								
	15	14	13	12	11	10	9	8
	PCOD							
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0
	7	6	5	4	3	2	1	0
	PCOD							
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0

Bits	Field	Descriptions
[15:0]	PCODn	GPIO Port C pin n Open-Drain Selection Control Bits (n = 0 ~ 15) 0: Pin n Open-Drain output is disabled (The output type is CMOS output) 1: Pin n Open-Drain output is enabled (The output type is open-drain output)

Port C Output Current Drive Selection Register – PCDRVR

This register specifies the GPIO Port C output driving current.

Offset: 0x014

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	PCDV15		PCDV14		PCDV13		PCDV12	
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0
	23	22	21	20	19	18	17	16
	PCDV11		PCDV10		PCDV9		PCDV8	
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0
	15	14	13	12	11	10	9	8
	PCDV7		PCDV6		PCDV5		PCDV4	
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0
	7	6	5	4	3	2	1	0
	PCDV3		PCDV2		PCDV1		PCDV0	
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0

Bits	Field	Descriptions
[31:0]	PCDVn[1:0]	GPIO Port C pin n Output Current Drive Selection Control Bits (n = 0 ~ 15) 00: 4 mA source / sink current 01: 8 mA source / sink current 10: 12 mA source / sink current 11: 16 mA source / sink current

Port C Lock Register – PCLOCKR

This register specifies the GPIO Port C lock configuration.

Offset: 0x018

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	PCLKEY							
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0
	23	22	21	20	19	18	17	16
	PCLKEY							
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0
	15	14	13	12	11	10	9	8
	PCLOCK							
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0
	7	6	5	4	3	2	1	0
	PCLOCK							
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0

Bits	Field	Descriptions
[31:16]	PCLKEY	GPIO Port C lock Key 0x5FA0: Port C Lock function is enabled Others: Port C Lock function is disabled To lock the Port C function, a value of 0x5FA0 should be written into the PCLKEY field in this register. To execute a successful write operation on this lock register, the value written into the PCLKEY field must be 0x5FA0. If the value written into this field is not equal to 0x5FA0, any write operations on the PCLOCKR register will be aborted. The result of a read operation on the PCLKEY field returns the GPIO Port C Lock Status which indicates whether the GPIO Port C is locked or not. If the read value of the PCLKEY field is 0, this indicates that the GPIO Port C Lock function is disabled. Otherwise, it indicates that the GPIO Port C Lock function is enabled as the read value is equal to 1.
[15:0]	PCLOCKn	GPIO Port C pin n Lock Control Bits (n = 0 ~ 15) 0: Port C pin n is not locked 1: Port C pin n is locked The PCLOCKn bits are used to lock the configurations of corresponding GPIO Pins when the correct Lock Key is applied to the PCLKEY field. The locked configurations including PCDIRn, PCINENn, PCPUn, PCPDn, PCODn and PCDVn setting in the related GPIO registers. Additionally, the GPCCFGHR or GPCCFGLR register which is used to configure the alternative function of the associated GPIO pin will also be locked. Note that the PCLOCKR register can only be written once which means that PCLKEY and PCLOCKn (lock control bit) should be written together and can not be changed until a system reset or GPIO Port C reset occurs.

Port C Data Input Register – PCDINR

This register specifies the GPIO Port C input data.

Offset: 0x01C

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	22	21	20	19	18	17	16
	Reserved							
Type/Reset								
	15	14	13	12	11	10	9	8
	PCDIN							
Type/Reset	RO	RO	RO	RO	RO	RO	RO	RO
	7	6	5	4	3	2	1	0
	PCDIN							
Type/Reset	RO	RO	RO	RO	RO	RO	RO	RO

Bits	Field	Descriptions
[15:0]	PCDINn	GPIO Port C pin n Data Input Bits (n = 0 ~ 15) 0: The input data of the corresponding pin is 0 1: The input data of the corresponding pin is 1

Port C Output Data Register – PCDOUTR

This register specifies the GPIO Port C output data.

Offset: 0x020

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	22	21	20	19	18	17	16
	Reserved							
Type/Reset								
	15	14	13	12	11	10	9	8
	PCDOUT							
Type/Reset	RW	RW	RW	RW	RW	RW	RW	RW
	7	6	5	4	3	2	1	0
	PCDOUT							
Type/Reset	RW	RW	RW	RW	RW	RW	RW	RW

Bits	Field	Descriptions
[15:0]	PCDOUTn	GPIO Port C pin n Data Output Bits (n = 0 ~ 15) 0: Data to be output on pin n is 0 1: Data to be output on pin n is 1

Port C Output Set / Reset Control Register – PCSRR

This register is used to set or reset the corresponding bit of the GPIO Port C output data.

Offset: 0x024

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	PCRST							
Type/Reset	WO 0	WO 0	WO 0	WO 0	WO 0	WO 0	WO 0	WO 0
	23	22	21	20	19	18	17	16
	PCRST							
Type/Reset	WO 0	WO 0	WO 0	WO 0	WO 0	WO 0	WO 0	WO 0
	15	14	13	12	11	10	9	8
	PCSET							
Type/Reset	WO 0	WO 0	WO 0	WO 0	WO 0	WO 0	WO 0	WO 0
	7	6	5	4	3	2	1	0
	PCSET							
Type/Reset	WO 0	WO 0	WO 0	WO 0	WO 0	WO 0	WO 0	WO 0

Bits	Field	Descriptions
[31:16]	PCRSTn	GPIO Port C pin n Output Reset Control Bits (n = 0 ~ 15) 0: No effect on the PCDOUTn bit 1: Reset the PCDOUTn bit Note that when the PCRSTn bit in this register or (and) the PCRSTn bit in the PCRR register is enabled, the reset function on the PCDOUTn bit will take effect.
[15:0]	PCSETn	GPIO Port C pin n Output Set Control Bits (n = 0 ~ 15) 0: No effect on the PCDOUTn bit 1: Set the PCDOUTn bit Note that the function enabled by the PCSETn bit has the higher priority if both the PCSETn and PCRSTn bits are set at the same time.

Port C Output Reset Register – PCRR

This register is used to reset the corresponding bit of the GPIO Port C output data.

Offset: 0x028

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	22	21	20	19	18	17	16
	Reserved							
Type/Reset								
	15	14	13	12	11	10	9	8
	PCRST							
Type/Reset	WO 0	WO 0	WO 0	WO 0	WO 0	WO 0	WO 0	WO 0
	7	6	5	4	3	2	1	0
	PCRST							
Type/Reset	WO 0	WO 0	WO 0	WO 0	WO 0	WO 0	WO 0	WO 0

Bits	Field	Descriptions
[15:0]	PCRSTn	GPIO Port C pin n Output Reset Control Bits (n = 0 ~ 15) 0: No effect on the PCDOUTn bit 1: Reset the PCDOUTn bit

Port D Data Direction Control Register – PDDIRCR

This register is used to control the direction of GPIO Port D pin as input or output.

Offset: 0x000

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	22	21	20	19	18	17	16
	Reserved							
Type/Reset								
	15	14	13	12	11	10	9	8
	Reserved							
Type/Reset								
	7	6	5	4	3	2	1	0
	Reserved				PDDIR			
Type/Reset					RW 0	RW 0	RW 0	RW 0

Bits	Field	Descriptions
[3:0]	PDDIRn	GPIO Port D pin n Direction Control Bits (n = 0 ~ 3) 0: Pin n is in input mode 1: Pin n is in output mode

Bits	Field	Descriptions

Port D Input Function Enable Control Register – PDINER

This register is used to enable or disable the GPIO Port D input function.

Offset: 0x004

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	22	21	20	19	18	17	16
	Reserved							
Type/Reset								
	15	14	13	12	11	10	9	8
	Reserved							
Type/Reset								
	7	6	5	4	3	2	1	0
	Reserved				PDINEN			
Type/Reset					RW 0	RW 0	RW 0	RW 0

Bits	Field	Descriptions
[3:0]	PDINENn	GPIO Port D pin n Input Enable Control Bits (n = 0 ~ 3) 0: Pin n input function is disabled 1: Pin n input function is enabled When the pin n input function is disabled, the input Schmitt trigger will be turned off and the Schmitt trigger output will remain at a zero state.

Port D Pull-Up Selection Register – PDPUR

This register is used to enable or disable the GPIO Port D pull-up function.

Offset: 0x008

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	22	21	20	19	18	17	16
	Reserved							
Type/Reset								
	15	14	13	12	11	10	9	8
	Reserved							
Type/Reset								
	7	6	5	4	3	2	1	0
	Reserved				PDPU			
Type/Reset					RW 0	RW 0	RW 0	RW 0

Bits	Field	Descriptions
[3:0]	PDPUn	GPIO Port D pin n Pull-Up Selection Control Bits (n = 0 ~ 3) 0: Pin n pull-up function is disabled 1: Pin n pull-up function is enabled Note: When the pull-up and pull-down functions are both enabled, the pull-up function will have the higher priority and therefore the pull-down function will be blocked and disabled.

Port D Pull-Down Selection Register – PDPDR

This register is used to enable or disable the GPIO Port D pull-down function.

Offset: 0x00C

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	22	21	20	19	18	17	16
	Reserved							
Type/Reset								
	15	14	13	12	11	10	9	8
	Reserved							
Type/Reset								
	7	6	5	4	3	2	1	0
	Reserved				PDPD			
Type/Reset					RW 0	RW 0	RW 0	RW 0

Bits	Field	Descriptions
[3:0]	PDPDn	GPIO Port D pin n Pull-Down Selection Control Bits (n = 0 ~ 3) 0: Pin n pull-down function is disabled 1: Pin n pull-down function is enabled Note: When the pull-up and pull-down functions are both enabled, the pull-up function will have the higher priority and therefore the pull-down function will be blocked and disabled.

Port D Open-Drain Selection Register – PDODR

This register is used to enable or disable the GPIO Port D open-drain function.

Offset: 0x010

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	22	21	20	19	18	17	16
	Reserved							
Type/Reset								
	15	14	13	12	11	10	9	8
	Reserved							
Type/Reset								
	7	6	5	4	3	2	1	0
	Reserved				PDOD			
Type/Reset					RW 0	RW 0	RW 0	RW 0

Bits	Field	Descriptions
[3:0]	PDODn	GPIO Port D pin n Open-Drain Selection Control Bits (n = 0 ~ 3) 0: Pin n Open-Drain output is disabled (The output type is CMOS output) 1: Pin n Open-Drain output is enabled (The output type is open-drain output)

Port D Output Drive Current Selection Register – PDDRVR

This register specifies the GPIO Port D output driving current.

Offset: 0x014

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	22	21	20	19	18	17	16
	Reserved							
Type/Reset								
	15	14	13	12	11	10	9	8
	Reserved							
Type/Reset								
	7	6	5	4	3	2	1	0
	PDDV3		PDDV2		PDDV1		PDDV0	
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0

Bits	Field	Descriptions
[7:0]	PDDVn[1:0]	GPIO Port D pin n Output Current Drive Selection Control Bits (n = 0 ~ 3) 00: 4 mA source / sink current 01: 8 mA source / sink current 10: 12 mA source / sink current 11: 16 mA source / sink current

Port D Lock Register – PDLOCKR

This register specifies the GPIO Port D lock configuration.

Offset: 0x018

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	PDLKEY							
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
	PDLKEY							
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
	Reserved							
Type/Reset								
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
	Reserved				PDLOCK			
Type/Reset					RW 0	RW 0	RW 0	RW 0

Bits	Field	Descriptions
[31:16]	PDLKEY	GPIO Port D lock Key 0x5FA0: Port D Lock function is enabled Others: Port D Lock function is disabled To lock the Port D function, a value of 0x5FA0 should be written into the PDLKEY field in this register. To execute a successful write operation on this lock register, the value written into the PDLKEY field must be 0x5FA0. If the value written into this field is not equal to 0x5FA0, any write operations on the PDLOCKR register will be aborted. The result of a read operation on the PDLKEY field returns the GPIO Port D Lock Status which indicates whether the GPIO Port D is locked or not. If the read value of the PDLKEY field is 0, this indicates that the GPIO Port D Lock function is disabled. Otherwise, it indicates that the GPIO Port D Lock function is enabled as the read value is equal to 1.
[3:0]	PDLOCKn	GPIO Port D pin n Lock Control Bits (n = 0 ~ 3) 0: Port D pin n is not locked 1: Port D pin n is locked The PDLOCKn bits are used to lock the configurations of corresponding GPIO Pins when the correct Lock Key is applied to the PDLKEY field. The locked configurations including PDDIRn, PDINENn, PDPUn, PDPDn, PDODn and PDDVn setting in the related GPIO registers. Additionally, the GPDCFGHR or GPDCFGLR register which is used to configure the alternative function of the associated GPIO pin will also be locked. Note that the PDLOCKR register can only be written once which means that PDLKEY and PDLOCKn (lock control bit) should be written together and can not be changed until a system reset or GPIO Port D reset occurs.

Port D Data Input Register – PDDINR

This register specifies the GPIO Port D input data.

Offset: 0x01C

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	22	21	20	19	18	17	16
	Reserved							
Type/Reset								
	15	14	13	12	11	10	9	8
	Reserved							
Type/Reset								
	7	6	5	4	3	2	1	0
	Reserved				PDDIN			
Type/Reset					RO 0	RO 0	RO 0	RO 0

Bits	Field	Descriptions
[3:0]	PDDINn	GPIO Port D pin n Data Input Bits (n = 0 ~ 3) 0: The input data of the corresponding pin is 0 1: The input data of the corresponding pin is 1

Port D Output Data Register – PDDOUTR

This register specifies the GPIO Port D output data.

Offset: 0x020

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	22	21	20	19	18	17	16
	Reserved							
Type/Reset								
	15	14	13	12	11	10	9	8
	Reserved							
Type/Reset								
	7	6	5	4	3	2	1	0
	Reserved				PDDOUT			
Type/Reset					RW 0	RW 0	RW 0	RW 0

Bits	Field	Descriptions
[3:0]	PDDOUTn	GPIO Port D pin n Data Output Bits (n = 0 ~ 3) 0: Data to be output on pin n is 0 1: Data to be output on pin n is 1

Port D Output Set / Reset Control Register – PDSRR

This register is used to set or reset the corresponding bit of the GPIO Port D output data.

Offset: 0x024
Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	22	21	20	19	18	17	16
	Reserved				PDRST			
Type/Reset					WO 0	WO 0	WO 0	WO 0
	15	14	13	12	11	10	9	8
	Reserved							
Type/Reset								
	7	6	5	4	3	2	1	0
	Reserved				PDSET			
Type/Reset					WO 0	WO 0	WO 0	WO 0

Bits	Field	Descriptions
[19:16]	PDRSTn	GPIO Port D pin n Output Reset Control Bits (n = 0 ~ 3) 0: No effect on the PDDOUTn bit 1: Reset the PDDOUTn bit Note that when the PDRSTn bit in this register or (and) the PDRSTn bit in the PDRR register is enabled, the reset function on the PDDOUTn bit will take effect.
[3:0]	PDSETn	GPIO Port D pin n Output Set Control Bits (n = 0 ~ 3) 0: No effect on the PDDOUTn bit 1: Set the PDDOUTn bit Note that the function enabled by the PDSETn bit has the higher priority if both the PDSETn and PDRSTn bits are set at the same time.

Port D Output Reset Register – PDRR

This register is used to reset the corresponding bit of the GPIO Port D output data.

Offset: 0x028
Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	22	21	20	19	18	17	16
	Reserved							
Type/Reset								
	15	14	13	12	11	10	9	8
	Reserved							
Type/Reset								
	7	6	5	4	3	2	1	0
	Reserved				PDRST			
Type/Reset					WO 0	WO 0	WO 0	WO 0

Bits	Field	Descriptions
[3:0]	PDRSTn	GPIO Port D pin n Output Reset Control Bits (n = 0 ~ 3) 0: No effect on the PDDOUTn bit 1: Reset the PDDOUTn bit

9 Alternate Function Input / Output Control Unit (AFIO)

Introduction

In order to expand the flexibility of the GPIO or the usage of peripheral functions, each I/O pin can be configured to have up to sixteen different functions such as GPIO or IP functions by setting the GPxCFGLR or GPxCFGHR register where x is the different port name. According to the usage of the IP resource and application requirements, suitable pin-out locations can be selected by using the peripheral I/O remapping mechanism. Additionally, various GPIO pins can be selected to be the EXTI interrupt line by setting the EXTInPIN [3:0] field in the ESSRn register to trigger an interrupt or event. Refer to the EXTI section for more details.

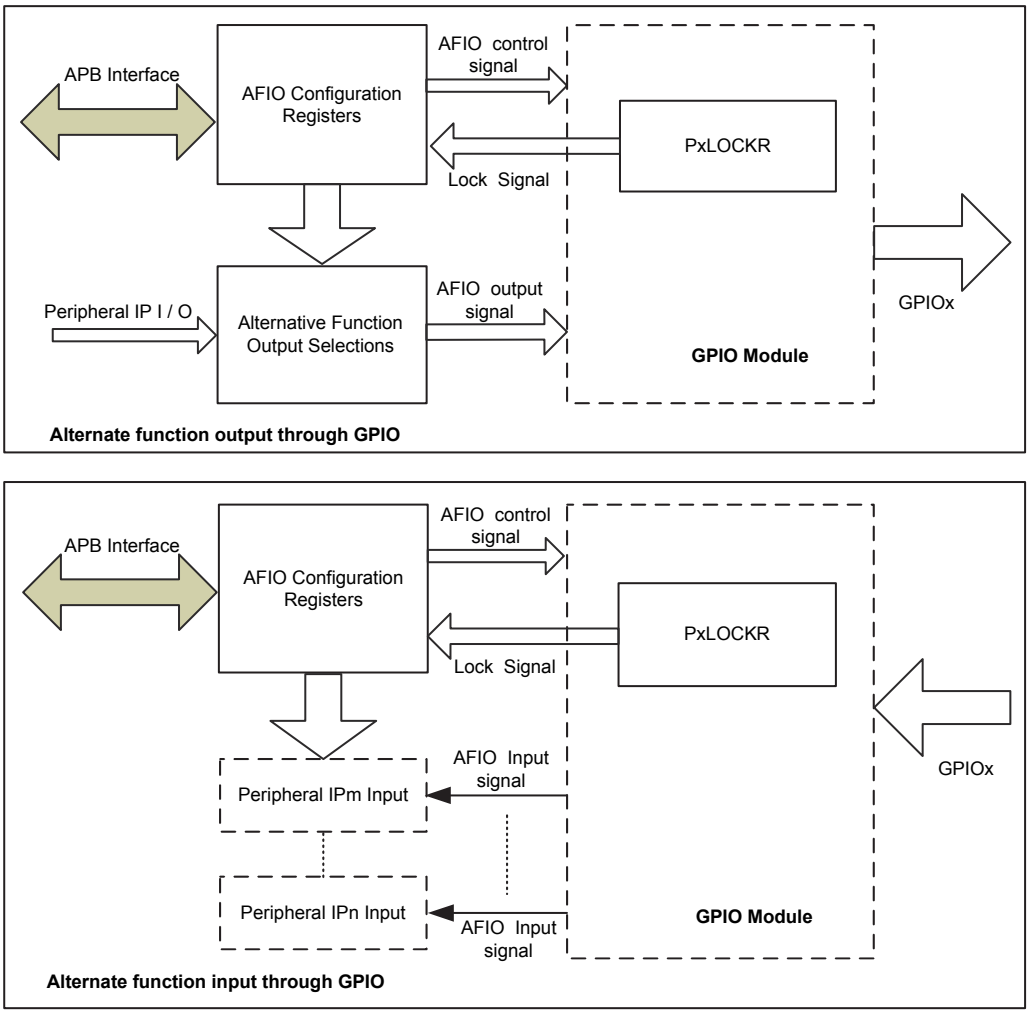


Figure 22. AFIO Block Diagram

Features

- APB slave interface for register access
- EXTI source selection
- Configurable pin function for each GPIO, up to sixteen alternative functions on each pin
- AFIO lock mechanism

Functional Descriptions

External Interrupt Pin Selection

The GPIO pins are connected to the 16 EXTI lines as shown in the accompanying figure. For example, users can set the EXTI0PIN [3:0] field in the ESSR0 register to b0000 to select the GPIO PA0 pin as EXTI line 0 input. Since not all the pins of the Port A ~ D are available in all package types, refer to the pin assignment section for detailed pin information. The setting of the EXTInPIN [3:0] field is invalid when the corresponding pin is not available.

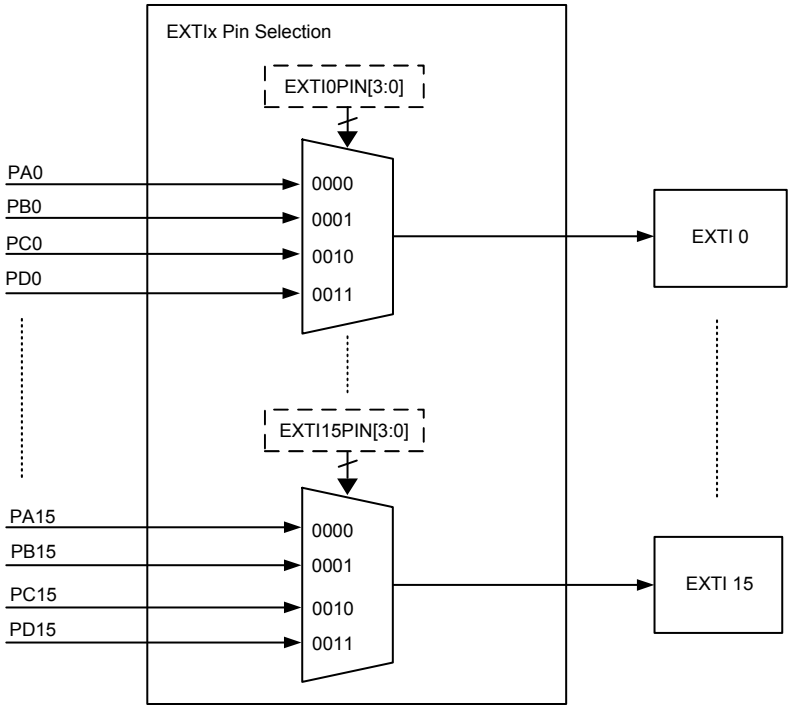


Figure 23. EXTI Channel Input Selection

Alternate Function

Up to sixteen alternative functions can be chosen for each I/O pad by setting the PxCFGn [3:0] field in the GPxCFGLR or GPxCFGHR (n = 0 ~ 15, x = A ~ D) registers. If the pin is selected as unavailable item which is noted as "N/A" item in the "Alternate Function Mapping" table of the device datasheet, this pin will be defined as default alternate function. Refer to the "Alternate Function Mapping" table in the device datasheet for the detailed mapping of the alternate function I/O pins. In addition to this flexible I/O multiplexing architecture, each peripheral has alternate functions mapped onto different I/O pins to optimize the number of peripherals available in smaller packages. The following description shows the setting of the PxCFGn [3:0] field.

- PxCFGn [3:0] = 0000: The default alternated function (after reset, AF0)
- PxCFGn [3:0] = 0001: Alternate Function 1 (AF1)
- PxCFGn [3:0] = 0010: Alternate Function 2 (AF2)
- …….
- PxCFGn [3:0] = 1110: Alternate Function 14 (AF14)
- PxCFGn [3:0] = 1111: Alternate Function 15 (AF15)

Table 22. AFIO Selection for Peripheral Map Example

AF0	AF1	AF2	AF3	AF4	AF5	AF6	AF7	AF8	AF9	AF10	AF11	AF12	AF13	AF14	AF15
System Default	GPIO	ADC /DAC	N/A	GPTM	SPI /QSPI	USART /UART	I^2C	N/A	N/A	I^2S	N/A	N/A	SCTM	N/A	System Other

Lock Mechanism

The device also offers a lock function to lock the AFIO configuration using the GPIO lock register, PxLOCKR, until a reset event occurs. Refer to the GPIO Locking Mechanism section in the GPIO chapter for more details.

Register Map

The following table shows the AFIO registers and reset value.

Table 23. AFIO Register Map

Register	Offset	Description	Reset Value
ESSR0	0x000	EXTI Source Selection Register 0	0x0000_0000
ESSR1	0x004	EXTI Source Selection Register 1	0x0000_0000
GPACFGLR	0x020	GPIO Port A Configuration Low Register	0x0000_0000
GPACFGHR	0x024	GPIO Port A Configuration High Register	0x0000_0000
GPBCFGLR	0x028	GPIO Port B Configuration Low Register	0x0000_0000
GPBCFGHR	0x02C	GPIO Port B Configuration High Register	0x0000_0000
GPCCFGLR	0x030	GPIO Port C Configuration Low Register	0x0000_0000
GPCCFGHR	0x034	GPIO Port C Configuration High Register	0x0000_0000
GPDCFGLR	0x038	GPIO Port D Configuration Low Register	0x0000_0000
GPDCFGHR	0x03C	GPIO Port D Configuration High Register	0x0000_0000

Register Descriptions

EXTI Source Selection Register 0 – ESSR0

This register specifies the I/O selection of EXTI0 ~ EXTI7.

Offset: 0x000

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	EXTI7PIN				EXTI6PIN			
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0
	23	22	21	20	19	18	17	16
	EXTI5PIN				EXTI4PIN			
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0
	15	14	13	12	11	10	9	8
	EXTI3PIN				EXTI2PIN			
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0
	7	6	5	4	3	2	1	0
	EXTI1PIN				EXTI0PIN			
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0

Bits	Field	Descriptions
[31:0]	EXTInPIN[3:0]	EXTIn Pin Selection (n = 0 ~ 7) 0000: PA Bit n is selected as EXTIn source signal 0001: PB Bit n is selected as EXTIn source signal 0010: PC Bit n is selected as EXTIn source signal 0011: PD Bit n is selected as EXTIn source signal Others: Reserved Note: Since not all GPIO pins are available in all products and package types, refer to the pin assignment section for detailed pin information. The EXTInPIN [3:0] field setting is invalid when the corresponding pin is not available.

EXTI Source Selection Register 1 – ESSR1

This register specifies the I/O selection of EXTI8 ~ EXTI15.

Offset: 0x004

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	EXTI15PIN				EXTI14PIN			
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0
	23	22	21	20	19	18	17	16
	EXTI13PIN				EXTI12PIN			
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0
	15	14	13	12	11	10	9	8
	EXTI11PIN				EXTI10PIN			
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0
	7	6	5	4	3	2	1	0
	EXTI9PIN				EXTI8PIN			
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0

Bits	Field	Descriptions
[31:0]	EXTInPIN[3:0]	EXTIn Pin Selection (n = 8 ~ 15) 0000: PA Bit n is selected as EXTIn source signal 0001: PB Bit n is selected as EXTIn source signal 0010: PC Bit n is selected as EXTIn source signal 0011: PD Bit n is selected as EXTIn source signal Others: Reserved Note: Since not all GPIO pins are available in all products and package types, refer to the pin assignment section for detailed pin information. The EXTInPIN [3:0] field setting is invalid when the corresponding pin is not available.

GPIO x Configuration Low Register – GPxCFGLR, x = A, B, C, D

This low register specifies the alternate function of GPIO Port x. x = A, B, C, D

Offset: 0x020, 0x028, 0x030, 0x038

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	PxCFG7				PxCFG6			
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0
	23	22	21	20	19	18	17	16
	PxCFG5				PxCFG4			
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0
	15	14	13	12	11	10	9	8
	PxCFG3				PxCFG2			
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0
	7	6	5	4	3	2	1	0
	PxCFG1				PxCFG0			
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0

Bits	Field	Descriptions
[31:0]	PxCFGn[3:0]	Alternate function selection for port x pin n (n = 0 ~ 7) 0000: Port x pin n is selected as AF0 0001: Port x pin n is selected as AF1 . . 1110: Port x pin n is selected as AF14 1111: Port x pin n is selected as AF15 If the pin is selected as unavailable item which is noted as “N/A” item in the “Alternate Function Mapping” table of the device datasheet, this pin will be defined as default alternate function. Refer to the “Alternate Function Mapping” table in the device datasheet for the detailed mapping of the alternate function I/O pins.

GPIO x Configuration High Register – GPxCFGHR, x = A, B, C, D

This high register specifies the alternate function of GPIO Port x. x = A, B, C, D

Offset: 0x024, 0x02C, 0x034, 0x03C

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	PxCFG15				PxCFG14			
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0
	23	22	21	20	19	18	17	16
	PxCFG13				PxCFG12			
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0
	15	14	13	12	11	10	9	8
	PxCFG11				PxCFG10			
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0
	7	6	5	4	3	2	1	0
	PxCFG9				PxCFG8			
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0

Bits	Field	Descriptions
[31:0]	PxCFGn[3:0]	Alternate function selection for port x pin n (n = 8 ~ 15) 0000: Port x pin n is selected as AF0 0001: Port x pin n is selected as AF1 . . 1110: Port x pin n is selected as AF14 1111: Port x pin n is selected as AF15 If the pin is selected as unavailable item which is noted as “N/A” item in the “Alternate Function Mapping” table of the device datasheet, this pin will be defined as default alternate function. Refer to the “Alternate Function Mapping” table in the device datasheet for the detailed mapping of the alternate function I/O pins.

10 Nested Vectored Interrupt Controller (NVIC)

Introduction

In order to reduce the latency and increase the interrupt processing efficiency, a tightly coupled integrated section, which is named as Nested Vectored Interrupt Controller (NVIC) is provided by the Cortex®-M0+. The NVIC controls the system exceptions and the peripheral interrupts which include functions such as the enable / disable control, priority, clear-pending, active status report, software trigger and vector table remapping. Refer to the Technical Reference Manual of Cortex®-M0+ for more details.

Additionally, an integrated simple, 24-bit down count timer (SysTick) is provided by the Cortex®-M0+ to be used as a tick timer for the Real Timer Operation System (RTOS) or as a simple counter. The SysTick counts down from the reloaded value and generates a system interrupt when it reaches zero. The accompanying table lists the system exceptions types and a variety of peripheral interrupts.

Table 24. Exception Types

Interrupt Number	Exception Number	Exception Type	Priority	Vector Address	Description
—	0	—	—	0x000	Initial Stack Point value
—	1	Reset	-3 (Highest)	0x004	Reset
-14	2	NMI	-2	0x008	Non-Maskable Interrupt. The clock stuck interrupt signal (clock monitor function provided by Clock Control Unit) is connected to the NMI input
-13	3	Hard Fault	-1	0x00C	All fault classes
—	4-10	Reserved	—	—	—
-5	11	SVCall	Configurable(1)	0x02C	SVC instruction System service call
—	12-13	Reserved	—	—	—
-2	14	PendSV	Configurable(1)	0x038	System Service Pendable request
-1	15	SysTick	Configurable(1)	0x03C	SysTick timer decremented to zero
0	16	LVD	Configurable(2)	0x040	Low voltage detection interrupt
1	17	RTC	Configurable(2)	0x044	RTC global interrupt
2	18	FMC	Configurable(2)	0x048	FMC global interrupt
3	19	WKUP	Configurable(2)	0x04C	EXTI event wakeup or external WAKEUP pin interrupt
4	20	EXTI0 ~ 1	Configurable(2)	0x050	EXTI Line 0 & 1 interrupt
5	21	EXTI2 ~ 3	Configurable(2)	0x054	EXTI Line 2 & 3 interrupt
6	22	EXTI4 ~ 15	Configurable(2)	0x058	EXTI Line 4 ~ 15 interrupt
7	23	Reserved	—	0x05C	—
8	24	ADC	Configurable(2)	0x060	ADC global interrupt
9	25	Reserved	—	0x064	—
10	26	Reserved	—	0x068	—
11	27	Reserved	—	0x06C	—
12	28	GPTM	Configurable(2)	0x070	GPTM global interrupt
13	29	SCTM0	Configurable(2)	0x074	SCTM0 global interrupt

Interrupt Number	Exception Number	Exception Type	Priority	Vector Address	Description
14	30	SCTM1	Configurable(2)	0x078	SCTM1 global interrupt
15	31	SCTM2	Configurable(2)	0x07C	SCTM2 global interrupt
16	32	SCTM3	Configurable(2)	0x080	SCTM3 global interrupt
17	33	BFTM0	Configurable(2)	0x084	BFTM0 global interrupt
18	34	BFTM1	Configurable(2)	0x088	BFTM1 global interrupt
19	35	I^2C	Configurable(2)	0x08C	I^2C global interrupt
20	36	Reserved	—	0x090	—
21	37	SPI	Configurable(2)	0x094	SPI global interrupt
22	38	QSPI	Configurable(2)	0x098	QSPI global interrupt
23	39	USART	Configurable(2)	0x09C	USART global interrupt
24	40	Reserved	—	0x0A0	—
25	41	UART	Configurable(2)	0x0A4	UART global interrupt
26	42	Reserved	—	0x0A8	—
27	43	MIDI	Configurable(2)	0x0AC	MIDI global interrupt
28	44	I^2S	Configurable(2)	0x0B0	I^2S global interrupt
29	45	USB	Configurable(2)	0x0B4	USB global interrupt
30	46	PDMA_CH0 ~ 1	Configurable(2)	0x0B8	PDMA channel 0 & 1 global interrupt
31	47	PDMA_CH2 ~ 5	Configurable(2)	0x0BC	PDMA channel 2 ~ 5 global interrupt

Notes: 1. The exception priority can be changed using the NVIC System Handler Priority Registers. For more information, refer to the Arm® "Cortex®-M0+ Devices Generic User Guide" document.
2. The interrupt priority can be changed using the NVIC Interrupt Priority Registers. For more information, refer to the Arm® "Cortex®-M0+ Devices Generic User Guide" document.

Features

- 7 system Cortex®-M0+ exceptions
- Up to 32 Maskable peripheral interrupts
- 4 programmable priority levels (2 bits for interrupt priority setting)
- Non-Maskable interrupt
- Low-latency exception and interrupt handling
- Vector table remapping capability
  - Integrated simple, 24-bit system timer, SysTick
  - 24-bit down-counter
  - Auto-reloading capability
  - Maskable system interrupt generation when counter decreases to 0
  - SysTick clock source derived from the HCLK clock divided by 8

Function Descriptions

SysTick Calibration

The SysTick Calibration Value Register (SYST_CALIB) is provided by the NVIC to give a reference time base of 1ms for the RTOS tick timer or other purposes. The TENMS field in the SYST_CALIB register has a fixed value of 6000 which is the counter reload value to indicate 1 ms when the clock source comes from the SysTick reference input clock STCLK with a frequency of 6 MHz (48 MHz divide by 8).

Register Map

The following table shows the NVIC registers and reset values.

Table 25. NVIC Register Map

Register	Offset	Description	Reset Value
NVIC Base Address = 0xE000_E000			
SYST_CSR	0x010	SysTick Control and Status Register	0x0000_0000
SYST_RVR	0x014	SysTick Reload Value Register	Unpredictable
SYST_CVR	0x018	SysTick Current Value Register	Unpredictable
SYST_CALIB	0x01C	SysTick Calibration Value Register	0x4000_2328
NVIC_ISER	0x100	Interrupt Set Enable Register	0x0000_0000
NVIC_ICER	0x180	Interrupt Clear Enable Register	0x0000_0000
NVIC_ISPR	0x200	Interrupt Set Pending Register	0x0000_0000
NVIC_ICPR	0x280	Interrupt Clear Pending Register	0x0000_0000
NVIC_IPR0	0x400	Interrupt 0 ~ 3 Priority Register	0x0000_0000
NVIC_IPR1	0x404	Interrupt 4 ~ 7 Priority Register	0x0000_0000
NVIC_IPR2	0x408	Interrupt 8 ~ 11 Priority Register	0x0000_0000
NVIC_IPR3	0x40C	Interrupt 12 ~ 15 Priority Register	0x0000_0000
NVIC_IPR4	0x410	Interrupt 16 ~ 19 Priority Register	0x0000_0000
NVIC_IPR5	0x414	Interrupt 20 ~ 23 Priority Register	0x0000_0000
NVIC_IPR6	0x418	Interrupt 24 ~ 27 Priority Register	0x0000_0000
NVIC_IPR7	0x41C	Interrupt 28 ~ 31 Priority Register	0x0000_0000
CPUID	0xD00	CPUID register	0x410C_C601
ICSR	0xD04	Interrupt Control and State Register	0x0000_0000
VTOR	0xD08	Vector Table Offset Register	0x0000_0000
AIRCR	0xD0C	Application Interrupt and Reset Control Register	0xFA05_0000
SCR	0xD10	System Control Register	0x0000_0000
CCR	0xD14	Configuration and Control Register	0x0000_0204
SHPR2	0xD1C	System Handlers Priority Register 2	0x0000_0000
SHPR3	0xD20	System Handlers Priority Register 3	0x0000_0000

Note: For more detailed descriptions of the above registers, please refer to the "Cortex®-M0+ Devices Generic User Guide" document from Arm.

11 External Interrupt / Event Controller (EXTI)

Introduction

The External Interrupt / Event Controller, EXTI, comprises 16 edge detectors which can generate a wakeup event or interrupt requests independently. In the interrupt mode there are five trigger types which can be selected as the external interrupt trigger type, low level, high level, negative edge, positive edge and both edges, selectable using the SRCnTYPE field in the EXTICFGRn (n = 0 ~ 15) register. In the wakeup event mode, the wakeup event polarity can be configured by setting the EXTInWPOL (n = 0 ~ 15) field in the EXTIWAKUPPOLR register. If the EVWUPIEN bit in the EXTIWAKUPCR Register is set, the EVWUP interrupt can be generated when the associated wakeup event occurs and the corresponding EXTI wakeup enable bit is set. Each EXTI line can also be masked independently.

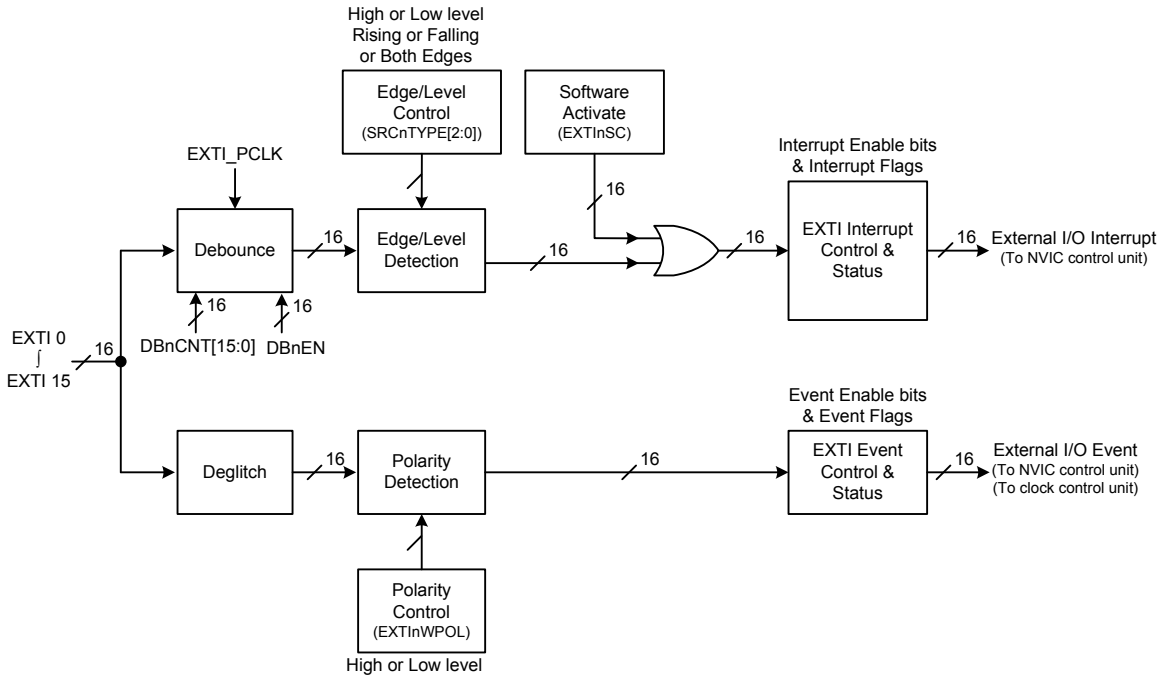


Figure 24. EXTI Block Diagram

Features

- Up to 16 EXTI lines with configurable trigger source and type
  - All GPIO pins can be selected as EXTI trigger source
  - Source trigger type includes high level, low level, negative edge, positive edge or both edge
- Individual interrupt enable, wakeup enable and status bits for each EXTI line
- Software interrupt trigger mode for each EXTI line
- Integrated deglitch filter for short pulse blocking

Function Descriptions

Wakeup Event Management

In order to wakeup the system from the power saving mode, the EXTI controller provides a function which can monitor external events and send them to the CPU core and the Clock Control Unit, CKCU. These external events include EXTI events, Low Voltage Detection, WAKEUP input pin, USB and RTC wakeup functions. By configuring the wakeup event enable bit in the corresponding peripheral, the wakeup signal will be sent to the CPU and the CKCU via the EXTI controller when the corresponding wakeup event occurs. Additionally, the software can enable the event wakeup interrupt function by setting the EVWUPIEN bit in the EXTIWAKUPCR register and the EXTI controller will then assert an interrupt when the wakeup event occurs.

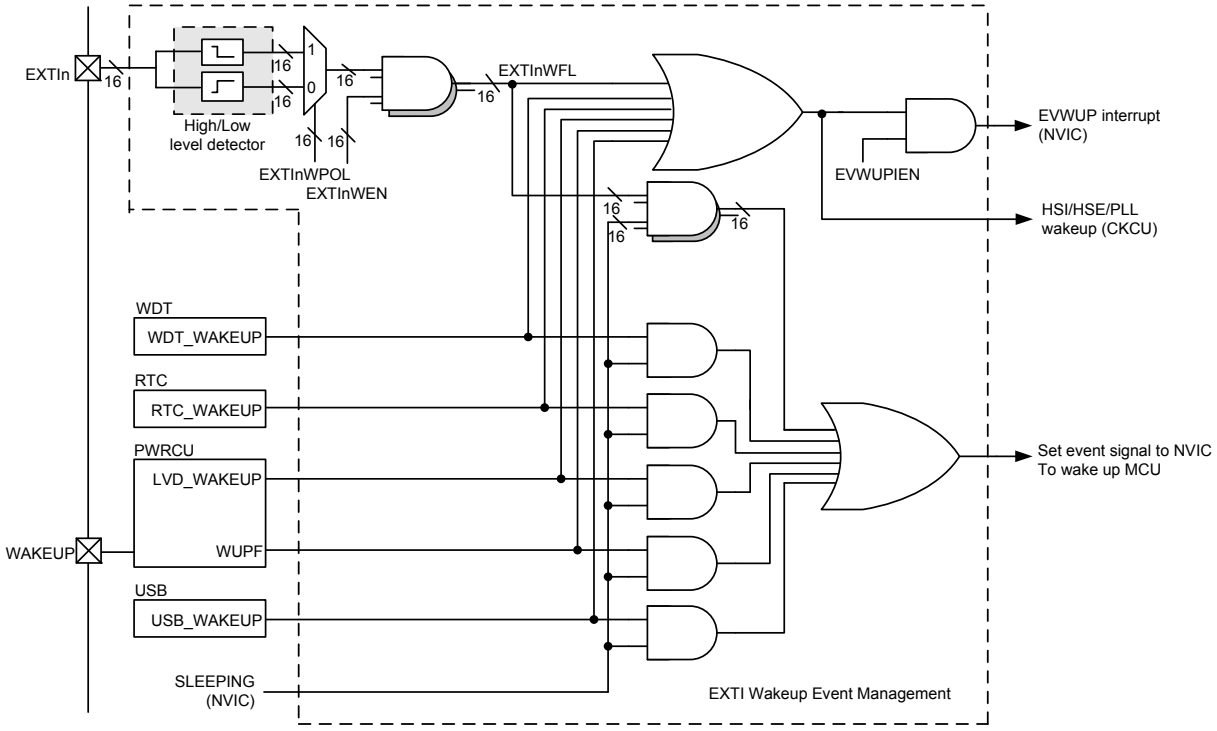


Figure 25. EXTI Wakeup Event Management

External Interrupt / Event Line Mapping

All GPIO pins can be selected as EXTI trigger sources by configuring the EXTInPIN [3:0] field in the AFIO ESSRn (n = 0 ~ 1) register to trigger an interrupt or event. Refer to the AFIO section for more details.

Interrupt and Debounce

The application software can set the DBnEN bit in the EXTIn Interrupt Configuration Register EXTICFGRn (n = 0 ~ 15) to enable the corresponding pin debounce function and configure the DBnCNT field in the EXTICFGRn register so as to select an appropriate debounce time for specific applications. The interrupt signal will however be delayed due to the debounce function. When the device is woken up from the power saving mode by an external interrupt, an interrupt request will be generated by the EXTI wakeup flag. After the device has been woken up and the clock has recovered, the EXTI wakeup flag that was triggered by the EXTI line must be read and then cleared by application software. The accompanying diagram shows the relationship between the EXTI input signal and the EXTI interrupt / event request signal.

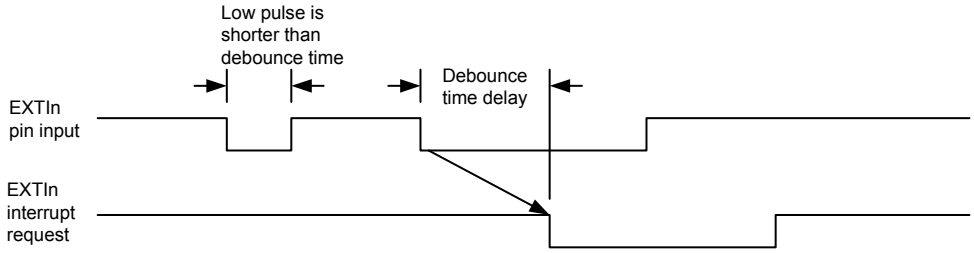


Figure 26. EXTI Interrupt Debounce Function

Register Map

The following table shows the EXTI registers and reset values.

Table 26. EXTI Register Map

Register	Offset	Description	Reset Value
EXTICFGR0	0x000	EXTI Interrupt 0 Configuration Register	0x0000_0000
EXTICFGR1	0x004	EXTI Interrupt 1 Configuration Register	0x0000_0000
EXTICFGR2	0x008	EXTI Interrupt 2 Configuration Register	0x0000_0000
EXTICFGR3	0x00C	EXTI Interrupt 3 Configuration Register	0x0000_0000
EXTICFGR4	0x010	EXTI Interrupt 4 Configuration Register	0x0000_0000
EXTICFGR5	0x014	EXTI Interrupt 5 Configuration Register	0x0000_0000
EXTICFGR6	0x018	EXTI Interrupt 6 Configuration Register	0x0000_0000
EXTICFGR7	0x01C	EXTI Interrupt 7 Configuration Register	0x0000_0000
EXTICFGR8	0x020	EXTI Interrupt 8 Configuration Register	0x0000_0000
EXTICFGR9	0x024	EXTI Interrupt 9 Configuration Register	0x0000_0000
EXTICFGR10	0x028	EXTI Interrupt 10 Configuration Register	0x0000_0000
EXTICFGR11	0x02C	EXTI Interrupt 11 Configuration Register	0x0000_0000
EXTICFGR12	0x030	EXTI Interrupt 12 Configuration Register	0x0000_0000
EXTICFGR13	0x034	EXTI Interrupt 13 Configuration Register	0x0000_0000
EXTICFGR14	0x038	EXTI Interrupt 14 Configuration Register	0x0000_0000
EXTICFGR15	0x03C	EXTI Interrupt 15 Configuration Register	0x0000_0000
EXTICR	0x040	EXTI Interrupt Control Register	0x0000_0000
EXTIEDGEFLGR	0x044	EXTI Interrupt Edge Flag Register	0x0000_0000
EXTIEDGESR	0x048	EXTI Interrupt Edge Status Register	0x0000_0000
EXTISSCR	0x04C	EXTI Interrupt Software Set Command Register	0x0000_0000
EXTIWAKUPCR	0x050	EXTI Interrupt Wakeup Control Register	0x0000_0000
EXTIWAKUPPOLR	0x054	EXTI Interrupt Wakeup Polarity Register	0x0000_0000
EXTIWAKUPFLG	0x058	EXTI Interrupt Wakeup Flag Register	0x0000_0000

Register Descriptions

EXTI Interrupt Configuration Register n – EXTICFGRn, n = 0 ~ 15

This register is used to specify the debounce function and select the trigger type.

Offset: 0x000 (0) ~ 0x03C (15)

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	DBnEN	SRCnTYPE			Reserved			
Type/Reset	RW 0	RW 0	RW 0	RW 0				
	23	22	21	20	19	18	17	16
	Reserved							
Type/Reset								
	15	14	13	12	11	10	9	8
	DBnCNT							
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0
	7	6	5	4	3	2	1	0
	DBnCNT							
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0

Bits	Field	Descriptions
[31]	DBnEN	EXTIn De-bounce Circuit Enable Bit (n = 0 ~ 15) 0: De-bounce circuit is disabled 1: De-bounce circuit is enabled
[30:28]	SRCnTYPE	EXTIn Interrupt Source Trigger Type (n = 0 ~ 15)
[15:0]	DBnCNT	EXTIn De-bounce Counter (n = 0 ~ 15) The de-bounce time is calculated with DBnCNT x APB clock (EXTI_PCLK) period and should be long enough to take effect on the input signal.

SRCnTYPE [2:0]			Interrupt Source Type
0	0	0	Low-level Sensitive
0	0	1	High-level Sensitive
0	1	0	Negative-edge Triggered
0	1	1	Positive-edge Triggered
1	X	X	Both-edge Triggered

EXTI Interrupt Control Register – EXTICR

This register is used to control the EXTI interrupt.

Offset: 0x040

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	22	21	20	19	18	17	16
	Reserved							
Type/Reset								
	15	14	13	12	11	10	9	8
	EXTI15EN	EXTI14EN	EXTI13EN	EXTI12EN	EXTI11EN	EXTI10EN	EXTI9EN	EXTI8EN
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0
	7	6	5	4	3	2	1	0
	EXTI7EN	EXTI6EN	EXTI5EN	EXTI4EN	EXTI3EN	EXTI2EN	EXTI1EN	EXTI0EN
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0

Bits	Field	Descriptions
[15:0]	EXTInEN	EXTIn Interrupt Enable Bit (n = 0 ~ 15) 0: EXTI line n interrupt is disabled 1: EXTI line n interrupt is enabled

EXTI Interrupt Edge Flag Register – EXTIEDGEFLGR

This register is used to indicate if an EXTI edge has been detected.

Offset: 0x044

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
	Reserved							
Type/Reset								
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
	EXTI15EDF	EXTI14EDF	EXTI13EDF	EXTI12EDF	EXTI11EDF	EXTI10EDF	EXTI9EDF	EXTI8EDF
Type/Reset	WC 0	WC 0	WC 0	WC 0	WC 0	WC 0	WC 0	WC 0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
	EXTI7EDF	EXTI6EDF	EXTI5EDF	EXTI4EDF	EXTI3EDF	EXTI2EDF	EXTI1EDF	EXTI0EDF
Type/Reset	WC 0	WC 0	WC 0	WC 0	WC 0	WC 0	WC 0	WC 0

Bits	Field	Descriptions
[15:0]	EXTInEDF	EXTIn Edge Detection Flag (n = 0 ~ 15) 0: No edge is detected 1: Positive or negative edge is detected This bit is set by the hardware circuitry when a positive or negative edge is detected on the corresponding EXTI line. Software should write 1 to clear it.

EXTI Interrupt Edge Status Register – EXTIEDGESR

This register indicates the polarity of a detected EXTI edge.

Offset: 0x048

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
	Reserved							
Type/Reset								
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
	EXTI15EDS	EXTI14EDS	EXTI13EDS	EXTI12EDS	EXTI11EDS	EXTI10EDS	EXTI9EDS	EXTI8EDS
Type/Reset	WC 0	WC 0	WC 0	WC 0	WC 0	WC 0	WC 0	WC 0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
	EXTI7EDS	EXTI6EDS	EXTI5EDS	EXTI4EDS	EXTI3EDS	EXTI2EDS	EXTI1EDS	EXTI0EDS
Type/Reset	WC 0	WC 0	WC 0	WC 0	WC 0	WC 0	WC 0	WC 0

Bits	Field	Descriptions
[15:0]	EXTInEDS	EXTIn Edge Detection Status (n = 0 ~ 15) 0: Negative edge is detected 1: Positive edge is detected Software should write 1 to clear it.

EXTI Interrupt Software Set Command Register – EXTISSCR

This register is used to activate the EXTI interrupt.

Offset: 0x04C

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	22	21	20	19	18	17	16
	Reserved							
Type/Reset								
	15	14	13	12	11	10	9	8
	EXTI15SC	EXTI14SC	EXTI13SC	EXTI12SC	EXTI11SC	EXTI10SC	EXTI9SC	EXTI8SC
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0
	7	6	5	4	3	2	1	0
	EXTI7SC	EXTI6SC	EXTI5SC	EXTI4SC	EXTI3SC	EXTI2SC	EXTI1SC	EXTI0SC
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0

Bits	Field	Descriptions
[15:0]	EXTInSC	EXTIn Software Set Command (n = 0 ~ 15) 0: Deactivates the corresponding EXTI interrupt 1: Activates the corresponding EXTI interrupt

EXTI Interrupt Wakeup Control Register – EXTIWAKUPCR

This register is used to control the EXTI interrupt and wakeup function.

Offset: 0x050

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	EVWUPIEN	Reserved						
Type/Reset	RW 0							
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
	Reserved							
Type/Reset								
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
	EXTI15WEN	EXTI14WEN	EXTI13WEN	EXTI12WEN	EXTI11WEN	EXTI10WEN	EXTI9WEN	EXTI8WEN
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
	EXTI7WEN	EXTI6WEN	EXTI5WEN	EXTI4WEN	EXTI3WEN	EXTI2WEN	EXTI1WEN	EXTI0WEN
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0

Bits	Field	Descriptions
[31]	EVWUPIEN	EXTI Event Wakeup Interrupt Enable Bit 0: Disable EVWUP interrupt 1: Enable EVWUP interrupt
[15:0]	EXTInWEN	EXTIn Wakeup Enable Bit (n = 0 ~ 15) 0: Power saving mode wakeup is disabled 1: Power saving mode wakeup is enabled

EXTI Interrupt Wakeup Polarity Register – EXTIWAKUPPOLR

This register is used to select the EXTI line interrupt wakeup polarity.

Offset: 0x054

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
	Reserved							
Type/Reset								
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
	EXTI15WPOL	EXTI14WPOL	EXTI13WPOL	EXTI12WPOL	EXTI11WPOL	EXTI10WPOL	EXTI9WPOL	EXTI8WPOL
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
	EXTI7WPOL	EXTI6WPOL	EXTI5WPOL	EXTI4WPOL	EXTI3WPOL	EXTI2WPOL	EXTI1WPOL	EXTI0WPOL
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0

Bits	Field	Descriptions
[15:0]	EXTInWPOL	EXTIn Wakeup Polarity (n = 0 ~ 15) 0: EXTIn wakeup is high level active 1: EXTIn wakeup is low level active

EXTI Interrupt Wakeup Flag Register – EXTIWAKUPFLG

This register is the EXTI interrupt wakeup flag register.

Offset: 0x058

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
	Reserved							
Type/Reset								
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
	EXTI15WFL	EXTI14WFL	EXTI13WFL	EXTI12WFL	EXTI11WFL	EXTI10WFL	EXTI9WFL	EXIT8WFL
Type/Reset	WC 0	WC 0	WC 0	WC 0	WC 0	WC 0	WC 0	WC 0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
	EXTI7WFL	EXTI6WFL	EXTI5WFL	EXTI4WFL	EXTI3WFL	EXTI2WFL	EXTI1WFL	EXTI0WFL
Type/Reset	WC 0	WC 0	WC 0	WC 0	WC 0	WC 0	WC 0	WC 0

Bits	Field	Descriptions
[15:0]	EXTInWFL	EXTIn Wakeup Flag (n = 0 ~ 15) 0: No wakeup occurs 1: System is woken up by EXTIn Software should write 1 to clear it.

12 Analog to Digital Converter (ADC)

Introduction

A 12-bit multi-channel Analog to Digital Converter is integrated in the device. There are a total of 18 multiplexed channels including 16 external channels on which the external analog signal can be supplied and 2 internal channels. If the input voltage is required to remain within a specific threshold window, the Analog Watchdog function will monitor and detect the signal. An interrupt will then be generated to inform that the input voltage is higher or lower than the set thresholds. There are three conversion modes to convert an analog signal to digital data. The A/D conversion can be operated in one shot, continuous and discontinuous conversion modes. A 16-bit data register is provided to store the data after conversion.

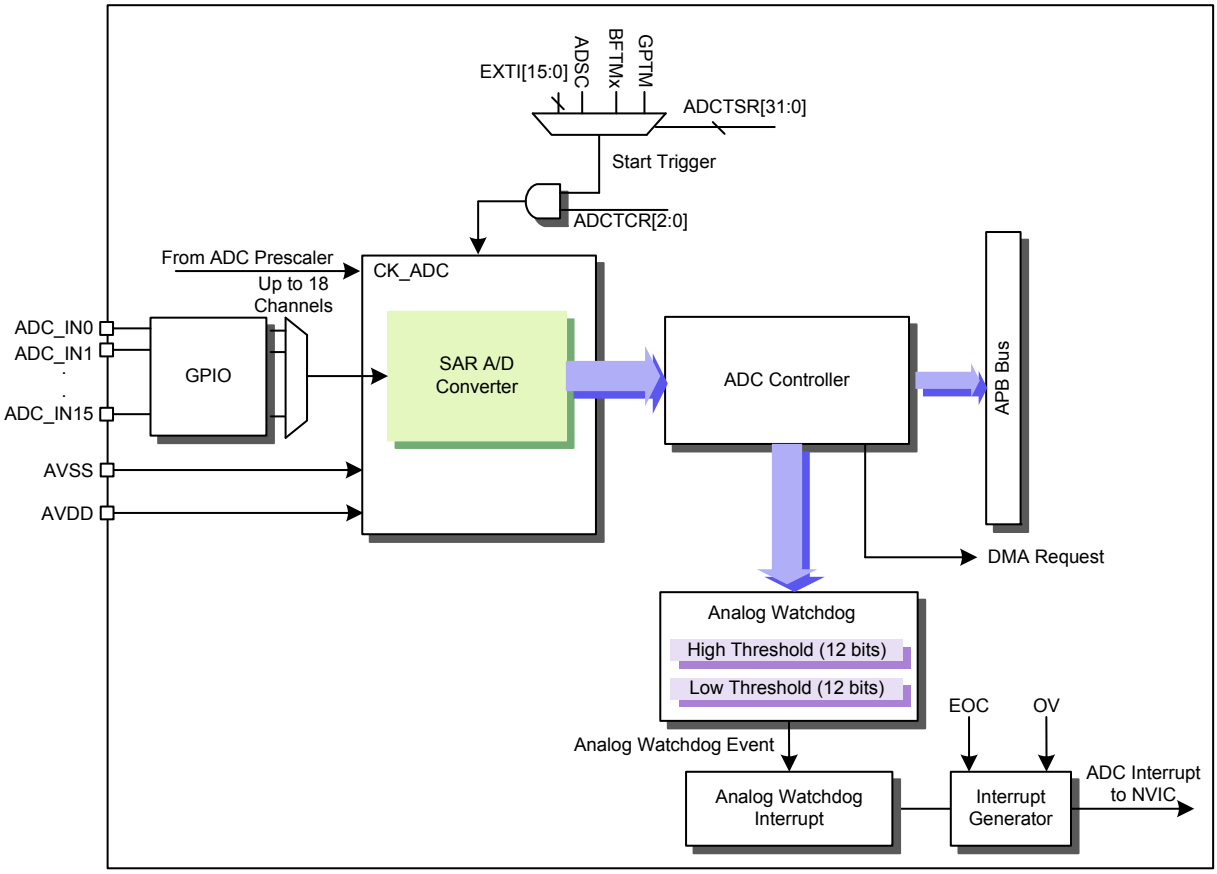


Figure 27. ADC Block Diagram

Features

- 12-bit SAR ADC engine
- Up to 1 MSPS conversion rate
- 16 external analog input channels
- 2 internal analog input channels for reference voltage detection
- Programmable sampling time for conversion channel
- Up to 8 programmable conversion channel sequence and dedicated data registers for conversion result
- Three conversion modes
  - One shot conversion mode
  - Continuous conversion mode
  - Discontinuous conversion mode
- Analog watchdog for predefined voltage range monitor
  - Lower / upper threshold register
  - Interrupt generation
- Various trigger start sources for conversion modes
  - Software trigger
  - EXTI – External interrupt input pin
  - GPTM trigger
  - BFTM0 / BFTM1 trigger
- Multiple generated interrupts
  - End of single conversion
  - End of subgroup conversion
  - End of cycle conversion
  - Analog Watchdog
  - Data register overwriting
- PDMA request on end of conversion occurred

Function Descriptions

ADC Clock Setup

The ADC clock, CK_ADC is provided by the Clock Controller which is synchronous and divided by with the AHB clock known as HCLK. Refer to the Clock Control Unit chapter for more details. Notes that the ADC requires at least two ADC clock cycles to switch between power-on and power-off conditions (ADEN bit = '0').

Channel Selection

The A/D converter supports 16 multiplexed channels and organizes the conversion results into a specific group. A conversion group can organize a sequence which can be implemented on the channels arranged in a specific conversion sequence length from 1 to 8. For example, conversion can be carried out with the following channel sequence: CH2, CH4, CH7, CH5, CH6, CH3, CH0 and CH1 one after another.

A group is composed of up to 8 conversions. The selected channels of the group conversion can be specified in the ADCLST0 ~ ADCLST1 registers. The total conversion sequence length is setup using the ADSEQL[2:0] bits in the ADCCR register.

Modifying the ADCCR or ADCLSTn register during a conversion process will reset the current conversion, after which a new start pulse is required to restart a new conversion.

Conversion Mode

The A/D has three operating conversion modes. The conversion modes are One Shot Conversion Mode, Continuous Conversion Mode and Discontinuous Conversion mode. Details are provided later.

One Shot Conversion Mode

In the One Shot Conversion mode, the ADC will perform conversion cycles on the channels specified in the A/D conversion list registers ADCLSTn with a specific sequence when an A/D converter trigger event occurs. When the A/D conversion mode field ADMODE [1:0] in the ADCCR register is set to 0x0, the A/D converter will operate in the One Shot Conversion Mode. This mode can be started by a software trigger, an external EXTI event or a TM event determined by the Trigger Control Register ADCTCR and the Trigger Source Register ADCTSR.

After Conversion

- The converted data will be stored in the 16-bit ADCDRy (y = 0 ~ 7) registers.
- The ADC single sample end of conversion event raw status flag, ADIRAWS, in the ADCIRAW register will be set when the single sample conversion is finished.
- An interrupt will be generated after a single sample end of conversion if the ADIES bit in the ADCIER register is enabled.
- An interrupt will be generated after a group cycle end of conversion if the ADIEC bit in the ADCIER register is enabled.

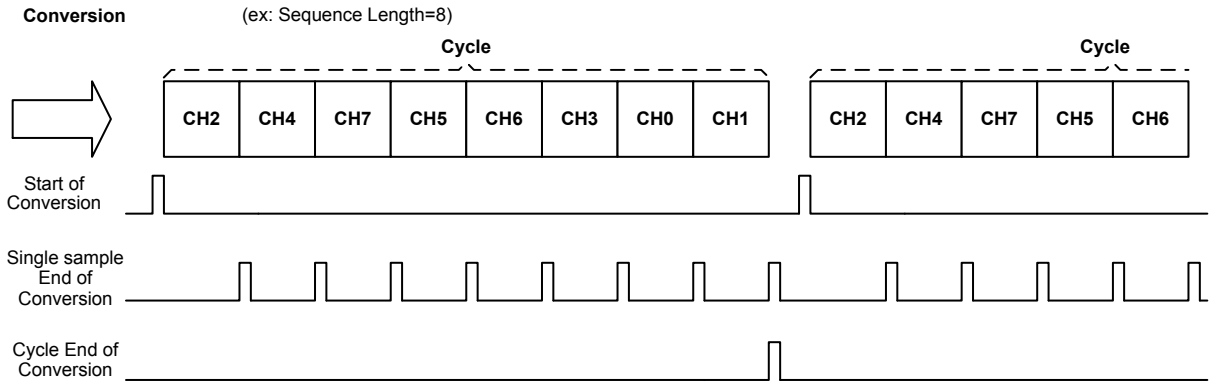


Figure 28. One Shot Conversion Mode

Continuous Conversion Mode

In the Continuous Conversion Mode, repeated conversion cycle will restart automatically without requiring additional A/D start trigger signals after a channel group conversion has completed. When the A/D conversion mode field ADMODE[1:0] is set to 0x2, the A/D converter will operate in the Continuous Conversion Mode which can be started by a software trigger, an external EXTI event or a TM event determined by the Trigger Control Register ADCTCR and the Trigger Source Register ADCTSR.

After conversion:

- The converted data will be stored in the 16-bit ADCDRy (y = 0 ~ 7) registers.
- The ADC group cycle end of conversion event raw status flag, ADIRAWC, in the ADCIRAW register will be set when the conversion cycle is finished.
- An interrupt will be generated after a group cycle end of conversion if the ADIEC bit in the ADCIER register is enabled.

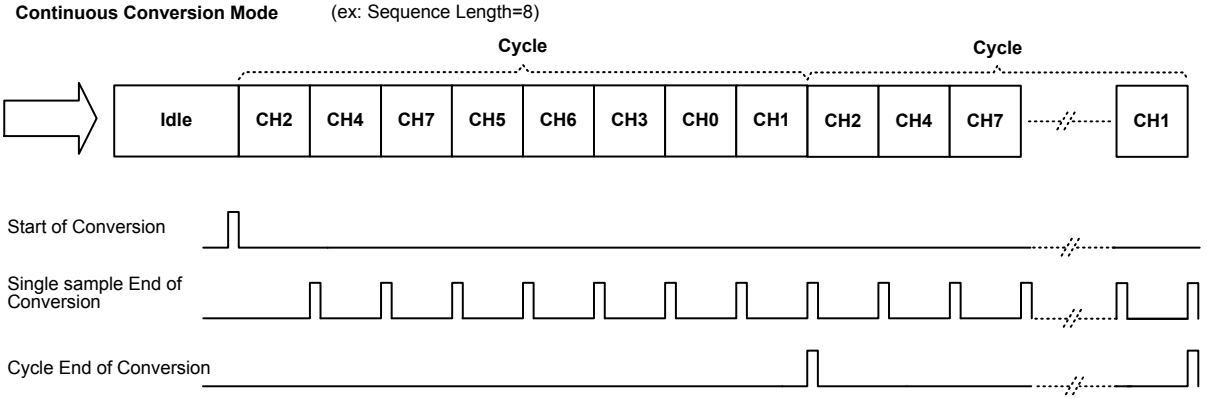


Figure 29. Continuous Conversion Mode

Discontinuous Conversion Mode

The A/D converter will operate in the Discontinuous Conversion Mode for channels group when the A/D conversion mode bit field ADMODE [1:0] in the ADCCR register is set to 0x3. The group to be converted can have up to 8 channels and can be arranged in a specific sequence by configuring the ADCLSTn registers where n ranges from 0 to 1. This mode is provided to convert data for the group with a short sequence, named as the A/D conversion subgroup, each time a trigger event occurs. The subgroup length is defined by the ADSUBL [2:0] field in the ADCCR register to specify the subgroup length. In the Discontinuous Conversion Mode the A/D converter can be started by a software trigger, an external EXTI event or a TM event for the groups determined by the Trigger Control Register ADCTCR and the Trigger Source Register ADCTSR.

In the Discontinuous Conversion Mode, the A/D Converter will start to convert the next n conversions where the number n is the subgroup length defined by the ADSUBL field. When a trigger event occurs, the channels to be converted with a specific sequence are specified in the ADCLSTn registers. After n conversions have completed, the subgroup EOC interrupt raw flag ADIRAWG in the ADCIRAW register will be asserted. The A/D converter will now not continue to perform the next n conversions until the next trigger event occurs. The conversion cycle will end after all the group channels, of which the total number is defined by the ADSEQL[2:0] bits in the ADCCR register, have finished their conversion, at which point the cycle EOC interrupt raw flag ADIRAWC in the ADCIRAW register will be asserted. If a new trigger event occurs after all the subgroup channels have all been converted, i.e., a complete conversion cycle has been finished, the conversion will restart from the first subgroup.

Example:

A/D subgroup length = 3 (ADSUBL = 2) and sequence length = 8 (ADSEQL = 7), channels to be converted = 2, 4, 7, 5, 6, 3, 0 and 1 – specific converting sequence as defined in the ADCLSTn registers,

- Trigger 1: subgroup channels to be converted are CH2, CH4 and CH7 with the ADIRAWG flag being asserted after subgroup EOC.
- Trigger 2: subgroup channels to be converted are CH5, CH6 and CH3 with the ADIRAWG flag being asserted after subgroup EOC.
- Trigger 3: subgroup channels to be converted are CH0 and CH1 with the ADIRAWG flag being asserted after subgroup EOC. Also a Cycle end of conversion (EOC) interrupt raw flag ADIRAWC will be asserted.
- Trigger 4: subgroup channels to be converted are CH2, CH4 and CH7 with the ADIRAWG flag being asserted – conversion sequence restarts from the beginning.

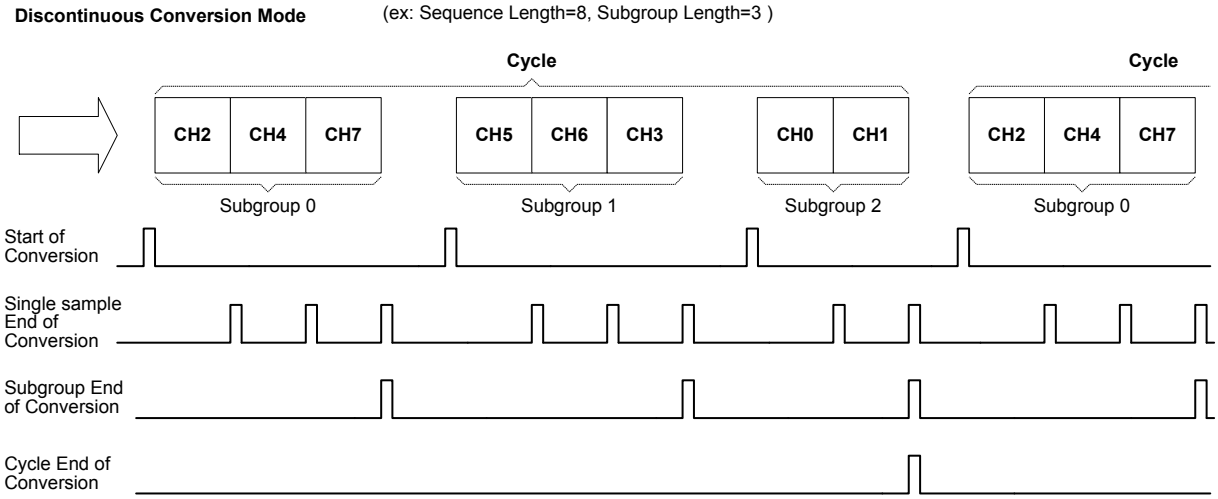


Figure 30. Discontinuous Conversion Mode

Start Conversion on External Event

An A/D conversion can be initiated by a software trigger, a General-Purpose Timer Module (GPTM) event, a Basic Function Timer Module (BFTM) event or an external trigger. Each trigger source can be enabled by setting the corresponding enable control bit in the ADCTCR register and then selected by configuring the associated selection bits in the ADCTSR register to start a group channel conversion.

An A/D converter conversion can be started by setting the software trigger bit, ADSC, in the ADCTSR register for the group channel when the software trigger enable bit, ADSW, in the ADCTCR register is set to 1. After the A/D converter starts converting the analog data, the corresponding enable bit ADSC will be cleared to 0 automatically.

The A/D converter can also be triggered to start a group conversion by a TM event. The TM events include a GPTM master trigger output MTO, four GPTM channel outputs CH0 ~ CH3 and a BFTM trigger output. If the corresponding Timer trigger enable bit is set to 1 and the trigger output or the TM channel event is selected via the relevant TM event selection bits, the A/D converter will start a conversion when a rising edge of the selected trigger event occurs.

In addition to the internal trigger sources, the A/D converter can be triggered to start a conversion by an external trigger event. The external trigger event is derived from the external lines, EXTIn. If the external trigger enable bit ADEXTI is set to 1 and the corresponding EXTI line is selected by configuring the ADEXTIS field in the ADCTSR register, the A/D converter will start a conversion when an EXTI line active edge determined in the EXTI Unit occurs.

Sampling Time Setting

The conversion channel sampling time can be programmed according to the input resistance of the input voltage source. This sampling time must be enough for the input voltage source to charge the internal sample and hold capacitor in A/D the converter to the input voltage level. Each conversion channel is sampled with the same sampling time. By modifying the ADST [7:0] bits in the ADCSTR register, the sampling time of the analog input signal can be determined.

The total conversion time (T_{conv}) is calculated using the following formula:

$$T_{conv} = T_{Sampling} + T_{Latency}$$

Where the minimum sampling time $T_{Sampling}$ = 1.5 cycles (when ADST[7:0] = 0) and the minimum channel conversion latency $T_{Latency}$ = 12.5 cycles.

Example:

With the A/D Converter clock CK_ADC = 14 MHz and a sampling time = 1.5 cycles:

$$T_{conv} = 1.5 + 12.5 = 14 \text{ cycles} = 1\ \mu s$$

Data Format

The ADC conversion result can be read in the ADCDRy register and the data format is shown in the following Table 27.

Table 27. Data format in ADCDR [15:0]

Description	ADCDR Register Data Format
Right aligned	"0_0_0_0_d11_d10_d9_d8_d7_d6_d5_d4_d3_d2_d1_d0"

Analog Watchdog

The A/D converter includes a watchdog function to monitor the converted data. There are two kinds of thresholds for the watchdog monitor function, known as the watchdog upper threshold and watchdog lower threshold, which are specified by the ADLT bit field and ADUT bit field in the ADCTR register respectively. The watchdog monitor function is enabled by setting the watchdog upper and lower threshold monitor function enable bits, ADWUE and ADWLE, in the watchdog control register ADCWCR. The channel to be monitored can be specified by configuring the ADWCH and ADWALL bits. When the converted data is less or higher than the lower or upper threshold, as defined by the ADLT bit field and ADUT bit field in the ADCTR register respectively, the watchdog lower or upper threshold interrupt raw flags, ADIRAWL or ADIRAWU in the ADCIRAW register, will be asserted if the watchdog lower or upper threshold monitor function is enabled. If the lower or upper threshold interrupt raw flag is asserted and the corresponding interrupt is enabled by setting the ADIEL or ADIEU bit in the ADCIER register, the A/D watchdog lower or upper threshold interrupt will be generated.

Interrupts

When an A/D conversion is completed, an End of Conversion EOC event will occur. There are three kinds of EOC events which are known as single sample EOC, subgroup EOC and cycle EOC for A/D conversion. A single sample EOC event will occur and the single sample EOC interrupt raw flag, ADIRAWS bits in the ADCIRAW register, will be asserted when a single channel conversion has completed. A subgroup EOC event will occur and the subgroup EOC interrupt raw flag, ADIRAWG in the ADCIRAW register, will be asserted when a subgroup conversion has completed. A cycle EOC event will occur and the cycle EOC interrupt raw flag, ADIRAWC bits in the ADCIRAW register, will be asserted when a cycle conversion is finished. When a single sample EOC, a subgroup EOC or a cycle EOC raw flag is asserted and the corresponding interrupt enable bit, ADIES, ADIEG or ADIEC bit in the ADCIER register, is set to 1, the associated interrupt will be generated.

After a conversion has completed, the 12-bit digital data will be stored in the associated ADCDRy registers and the value of the data valid flag named as ADVLDy will be changed from low to high. The converted data should be read by the application program, after which the data valid flag ADVLDy will be automatically changed from high to low. Otherwise, a data overwrite event will occur and the data overwrite interrupt raw flag ADIRAWO bit in the ADCIRAW register will be asserted. When the related data overwrite raw flag is asserted, the data overwrite interrupt will be generated if the interrupt enable bit ADIEO in the ADCIER register is set to 1.

If the A/D watchdog monitor function is enabled and the data after a channel conversion is less than the lower threshold or higher than the upper threshold, the watchdog lower or upper threshold interrupt raw flag ADIRAWL or ADIRAWU in the ADCIRAW register will be asserted. When the ADIRAWL or ADIRAWU flag is asserted and the corresponding interrupt enable bit, ADIEL or ADIEU in the ADCIER register, is set a watchdog lower or upper threshold interrupt will be generated.

The A/D Converter interrupt clear bits are used to clear the associated A/D converter interrupt raw and interrupt status bits. Writing a 1 into the specific A/D converter interrupt clear bit in the A/D converter interrupt clear register ADCICLR will clear the corresponding A/D converter interrupt raw and masked status bits. These bits are automatically cleared to 0 by hardware after being set to 1.

PDMA Request

The converted channel value will be stored in the corresponding data register. The A/D Converter can inform the MCU using the A/D Converter EOC interrupt if a new conversion data is already stored in the ADCDRy register. Users also can determine if the PDMA request is asserted by setting the ADDMAC, ADDMAG or ADDMAS bits in the ADCDMAR register. A PDMA request will be automatically generated at the relevant end of A/D conversion. The detail description will be introduced in the ADCDMAR register description.

Register Map

The following table shows the A/D Converter registers and reset values.

Table 28. A/D Converter Register Map

Register	Offset	Description	Reset Value
ADCCR	0x000	ADC Conversion Control Register	0x0000_0000
ADCLST0	0x004	ADC Conversion List Register 0	0x0000_0000
ADCLST1	0x008	ADC Conversion List Register 1	0x0000_0000
ADCSTR	0x020	ADC Input Sampling Time Register	0x0000_0000
ADCDR0	0x030	ADC Conversion Data Register 0	0x0000_0000
ADCDR1	0x034	ADC Conversion Data Register 1	0x0000_0000
ADCDR2	0x038	ADC Conversion Data Register 2	0x0000_0000
ADCDR3	0x03C	ADC Conversion Data Register 3	0x0000_0000
ADCDR4	0x040	ADC Conversion Data Register 4	0x0000_0000
ADCDR5	0x044	ADC Conversion Data Register 5	0x0000_0000
ADCDR6	0x048	ADC Conversion Data Register 6	0x0000_0000
ADCDR7	0x04C	ADC Conversion Data Register 7	0x0000_0000
ADCTCR	0x070	ADC Trigger Control Register	0x0000_0000
ADCTSR	0x074	ADC Trigger Source Register	0x0000_0000
ADCWCR	0x078	ADC Watchdog Control Register	0x0000_0000
ADCTR	0x07C	ADC Watchdog Threshold Register	0x0000_0000
ADCIER	0x080	ADC Interrupt Enable register	0x0000_0000
ADCIRAW	0x084	ADC Interrupt Raw Status Register	0x0000_0000
ADCISR	0x088	ADC Interrupt Status Register	0x0000_0000
ADCICLR	0x08C	ADC Interrupt Clear Register	0x0000_0000
ADCDMAR	0x090	ADC DMA Request Register	0x0000_0000

Register Descriptions

ADC Conversion Control Register – ADCCR

This register specifies the mode setting, sequence length and subgroup length of the ADC conversion mode. Note that once the content of ADCCR is changed, the conversion in progress will be aborted and the A/D converter will return to an idle state. The application program has to wait for at least one CK_ADC clock before issuing the next command.

Offset: 0x000

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	22	21	20	19	18	17	16
	Reserved					ADSUBL		
Type/Reset						RW 0	RW 0	RW 0
	15	14	13	12	11	10	9	8
	Reserved					ADSEQL		
Type/Reset						RW 0	RW 0	RW 0
	7	6	5	4	3	2	1	0
	ADCEN	ADCRST	Reserved					ADMODE
Type/Reset	RW 0	RW 0					RW 0	RW 0

Bits	Field	Descriptions
[18:16]	ADSUBL	ADC Conversion Subgroup Length The ADSUBL field specifies the conversion channel length of each subgroup in the Discontinuous Conversion Mode. Subgroup length = ADSUBL [2:0] + 1. If the sequence length (ADSEQL [2:0] + 1) is not a multiple of the subgroup length (ADSUBL [2:0] + 1), the last subgroup will be the rest of the group channels that have not been converted.
[10:8]	ADSEQL	ADC Conversion Length 0x00: The channel specified by the ADSEQ0 field in the ADCLST0 register will be converted. Others: Sequence length = ADSEQL [2:0] + 1. The ADSEQL field specifies the whole conversion sequence length for the conversion group.
[7]	ADCEN	ADC Enable 0: ADC disable 1: ADC enable When this bit is cleared to 0, the A/D converter will be disabled and the CK_ADC clock will also be switched off.
[6]	ADCRST	ADC Reset 0: No effect 1: Reset A/D converter except for the A/D Converter controller

Bits	Field	Descriptions
[1:0]	ADMODE	ADC Conversion Mode

ADMODE [1:0]	Mode	Descriptions
00	One shot mode	After a start trigger, the conversion will be executed on the specific channels for the whole conversion sequence once.
01	Reserved	
10	Continuous mode	After a start trigger, the conversion will be executed on the specific channels for the whole sequence continuously until conversion mode is changed.
11	Discontinuous mode	After a start trigger, the conversion will be executed on the current subgroup. When the last subgroup is finished, the conversion will restart from the first subgroup if another start trigger occurs.

ADC Conversion List Register 0 – ADCLST0

This register specifies the conversion sequence order No.0 ~ No.3 of the ADC.

Offset: 0x004

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved			ADSEQ3				
Type/Reset				RW 0	RW 0	RW 0	RW 0	RW 0
	23	22	21	20	19	18	17	16
	Reserved			ADSEQ2				
Type/Reset				RW 0	RW 0	RW 0	RW 0	RW 0
	15	14	13	12	11	10	9	8
	Reserved			ADSEQ1				
Type/Reset				RW 0	RW 0	RW 0	RW 0	RW 0
	7	6	5	4	3	2	1	0
	Reserved			ADSEQ0				
Type/Reset				RW 0	RW 0	RW 0	RW 0	RW 0

Bits	Field	Descriptions
[28:24]	ADSEQ3	ADC Conversion Sequence Select 3 Select the ADC input channel for the 3rd ADC conversion sequence. 0x0: ADC_IN0 0x1: ADC_IN1 0x2: ADC_IN2 0x3: ADC_IN3 0x4: ADC_IN4 0x5: ADC_IN5 0x6: ADC_IN6 0x7: ADC_IN7 0x8: ADC_IN8 0x9: ADC_IN9 0xA: ADC_IN10 0xB: ADC_IN11 0xC: ADC_IN12 0xD: ADC_IN13 0xE: ADC_IN14 0xF: ADC_IN15 0x10: Analog ground, AV_{SS} (V_{REF-}) 0x11: Analog power, AV_{DD} (V_{REF+}) 0x12 ~ 0x1F: Invalid setting. These values must not be selected as it may cause the ADC abnormal operations.
[20:16]	ADSEQ2	ADC Conversion Sequence Select 2
[12:8]	ADSEQ1	ADC Conversion Sequence Select 1
[4:0]	ADSEQ0	ADC Conversion Sequence Select 0

ADC Conversion List Register 1 – ADCLST1

This register specifies the conversion sequence order No.4 ~ No.7 of the ADC.

Offset: 0x008

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved			ADSEQ7				
Type/Reset				RW 0	RW 0	RW 0	RW 0	RW 0
	23	22	21	20	19	18	17	16
	Reserved			ADSEQ6				
Type/Reset				RW 0	RW 0	RW 0	RW 0	RW 0
	15	14	13	12	11	10	9	8
	Reserved			ADSEQ5				
Type/Reset				RW 0	RW 0	RW 0	RW 0	RW 0
	7	6	5	4	3	2	1	0
	Reserved			ADSEQ4				
Type/Reset				RW 0	RW 0	RW 0	RW 0	RW 0

Bits	Field	Descriptions
[28:24]	ADSEQ7	ADC Conversion Sequence Select 7 Select the ADC input channel for the 7th ADC conversion sequence. 0x0: ADC_IN0 0x1: ADC_IN1 0x2: ADC_IN2 0x3: ADC_IN3 0x4: ADC_IN4 0x5: ADC_IN5 0x6: ADC_IN6 0x7: ADC_IN7 0x8: ADC_IN8 0x9: ADC_IN9 0xA: ADC_IN10 0xB: ADC_IN11 0xC: ADC_IN12 0xD: ADC_IN13 0xE: ADC_IN14 0xF: ADC_IN15 0x10: Analog ground, AV_{SS} (V_{REF-}) 0x11: Analog power, AV_{DD} (V_{REF+}) 0x12 ~ 0x1F: Invalid setting. These values must not be selected as it may cause the ADC abnormal operations.
[20:16]	ADSEQ6	ADC Conversion Sequence Select 6
[12:8]	ADSEQ5	ADC Conversion Sequence Select 5
[4:0]	ADSEQ4	ADC Conversion Sequence Select 4

ADC Input Sampling Time Register – ADCSTR

This register specifies the A/D converter input channel sampling time.

Offset: 0x020

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	22	21	20	19	18	17	16
	Reserved							
Type/Reset								
	15	14	13	12	11	10	9	8
	Reserved							
Type/Reset								
	7	6	5	4	3	2	1	0
	ADST							
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0

Bits	Field	Descriptions
[7:0]	ADST	ADC Input Channel Sampling Time Sampling time = (ADST[7:0] + 1.5) × CK_ADC clocks.

ADC Conversion Data Register y – ADCDRy, y = 0 ~ 7

This register is used to store the conversion data of the conversion sequence order No.y which is specified by the ADSEQy field in the ADCLSTn (n = 0 ~ 1) registers.

Offset: 0x030 ~ 0x04C

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	ADVLDy	Reserved						
Type/Reset	RC	0						
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
	Reserved							
Type/Reset								
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
	ADDy							
Type/Reset	RO	0	RO	0	RO	0	RO	0	RO	0	RO	0	RO	0	RO	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
	ADDy							
Type/Reset	RO	0	RO	0	RO	0	RO	0	RO	0	RO	0	RO	0	RO	0

Bits	Field	Descriptions
[31]	ADVLDy	ADC Conversion Data of Sequence Order No.y Valid Bit (y = 0 ~ 7) 0: Data are invalid or have been read 1: New data is valid
[15:0]	ADDy	ADC Conversion Data of Sequence Order No.y (y = 0 ~ 7) The conversion result of Sequence Order ADSEQy in the ADCLSTn (n = 0 ~ 1) registers

ADC Trigger Control Register – ADCTCR

This register contains the ADC start conversion trigger enable bits.

Offset: 0x070

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	22	21	20	19	18	17	16
	Reserved							
Type/Reset								
	15	14	13	12	11	10	9	8
	Reserved							
Type/Reset								
	7	6	5	4	3	2	1	0
	Reserved				BFTM	TM	ADEXTI	ADSW
Type/Reset					RW 0	RW 0	RW 0	RW 0

Bits	Field	Descriptions
[3]	BFTM	ADC Conversion BFTM Event Trigger enable control 0: Disable conversion trigger by BFTM events 1: Enable conversion trigger by BFTM events
[2]	TM	ADC Conversion GPTM Event Trigger enable control 0: Disable conversion trigger by GPTM events 1: Enable conversion trigger by GPTM events
[1]	ADEXTI	ADC Conversion EXTI Event Trigger enable control 0: Disable conversion trigger by EXTI lines 1: Enable conversion trigger by EXTI lines
[0]	ADSW	ADC Conversion Software Trigger enable control 0: Disable conversion trigger by software trigger bit 1: Enable conversion trigger by software trigger bit

ADC Trigger Source Register – ADCTSR

This register contains the trigger source selection and the software trigger bit of the conversion.

Offset: 0x074

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved					TME		
Type/Reset						RW 0	RW 0	RW 0
	23	22	21	20	19	18	17	16
	Reserved				BFTMS	TMS		
Type/Reset					RW 0	RW 0	RW 0	RW 0
	15	14	13	12	11	10	9	8
	Reserved				ADEXTIS			
Type/Reset					RW 0	RW 0	RW 0	RW 0
	7	6	5	4	3	2	1	0
	Reserved							ADSC
Type/Reset								RW 0

Bits	Field	Descriptions
[26:24]	TME	GPTM Trigger Event Selection of ADC Conversion 000: GPTM MTO event 001: GPTM CH0O event 010: GPTM CH1O event 011: GPTM CH2O event 100: GPTM CH3O event Others: Reserved – Should not be used to avoid unpredictable results
[19]	BFTMS	BFTM Trigger Timer Selection of ADC Conversion 0: BFTM0 1: BFTM1
[18:16]	TMS	GPTM Trigger Timer Selection of ADC Conversion 010: GPTM Others: Reserved – Should not be used to avoid unpredictable results
[11:8]	ADEXTIS	EXTI Trigger Source Selection of ADC Conversion 0x00: EXTI line 0 0x01: EXTI line 1 … 0x0F: EXTI line 15 Note that the EXTI line active edge to start an A/D conversion is determined in the External Interrupt / Event Control Unit, EXTI.
[0]	ADSC	ADC Conversion Software Trigger Bit 0: No operation 1: Start conversion immediately This bit is set by software to start a conversion manually and then cleared by hardware automatically after conversion is started.

ADC Watchdog Control Register – ADCWCR

This register provides the control bits and status of the ADC watchdog function.

Offset: 0x078

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved				ADUCH			
Type/Reset					RO 0	RO 0	RO 0	RO 0
	23	22	21	20	19	18	17	16
	Reserved				ADLCH			
Type/Reset					RO 0	RO 0	RO 0	RO 0
	15	14	13	12	11	10	9	8
	Reserved				ADWCH			
Type/Reset					RW 0	RW 0	RW 0	RW 0
	7	6	5	4	3	2	1	0
	Reserved					ADWALL	ADWUE	ADWLE
Type/Reset						RW 0	RW 0	RW 0

Bits	Field	Descriptions
[27:24]	ADUCH	Upper Threshold Channel Status 0000: ADC_IN0 converted data is higher than the upper threshold 0001: ADC_IN1 converted data is higher than the upper threshold ... 1111: ADC_IN15 converted data is higher than the upper threshold Others: Reserved If both the ADWUE and ADWALL status bits are set to 1 by the watchdog monitor function, this status field value should first be stored in the user-defined memory location in the corresponding ISR. Otherwise, the ADUCH field will be changed if another input channel converted data is higher than the upper threshold.
[19:16]	ADLCH	Lower Threshold Channel Status 0000: ADC_IN0 converted data is lower than the lower threshold 0001: ADC_IN1 converted data is lower than the lower threshold ... 1111: ADC_IN15 converted data is lower than the lower threshold Others: Reserved If both the ADWLE and ADWALL status bits are set to 1 by the watchdog monitor function, this status field value should first be stored in the user-defined memory location in the corresponding ISR. Otherwise, the ADLCH field will be changed if another input channel converted data is lower than the lower threshold.
[11:8]	ADWCH	ADC Watchdog Specific Channel Selection 0000: ADC_IN0 is monitored 0001: ADC_IN1 is monitored ... 1111: ADC_IN15 is monitored Others: Reserved
[2]	ADWALL	ADC Watchdog Specific or All Channel Setting 0: Only the channel which specified by the ADWCH field is monitored 1: All channels are monitored

Bits	Field	Descriptions
[1]	ADWUE	ADC Watchdog Upper Threshold Enable Bit 0: Disable upper threshold monitor function 1: Enable upper threshold monitor function
[0]	ADWLE	ADC Watchdog Lower Threshold Enable Bit 0: Disable lower threshold monitor function 1: Enable lower threshold monitor function

ADC Watchdog Threshold Register – ADCTR

This register specifies the upper and lower threshold of the ADC watchdog function.

Offset: 0x07C

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved				ADUT			
Type/Reset					RW 0	RW 0	RW 0	RW 0
	23	22	21	20	19	18	17	16
	ADUT							
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0
	15	14	13	12	11	10	9	8
	Reserved				ADLT			
Type/Reset					RW 0	RW 0	RW 0	RW 0
	7	6	5	4	3	2	1	0
	ADLT							
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0

Bits	Field	Descriptions
[27:16]	ADUT	ADC Watchdog Upper Threshold Value Specify the upper threshold for the ADC watchdog monitor function.
[11:0]	ADLT	ADC Watchdog Lower Threshold Value Specify the lower threshold for the ADC watchdog monitor function.

ADC Interrupt Enable Register – ADCIER

This register contains the ADC interrupt enable bits.

Offset: 0x080

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved							ADIEO
Type/Reset								RW 0
	23	22	21	20	19	18	17	16
	Reserved						ADIEU	ADIEL
Type/Reset							RW 0	RW 0
	15	14	13	12	11	10	9	8
	Reserved							
Type/Reset								
	7	6	5	4	3	2	1	0
	Reserved					ADIEC	ADIEG	ADIES
Type/Reset						RW 0	RW 0	RW 0

Bits	Field	Descriptions
[24]	ADIEO	ADC Data Register Overwrite Interrupt enable 0: ADC data register overwrite interrupt is disabled 1: ADC data register overwrite interrupt is enabled
[17]	ADIEU	ADC Watchdog Upper Threshold Interrupt enable 0: ADC watchdog upper threshold interrupt is disabled 1: ADC watchdog upper threshold interrupt is enabled
[16]	ADIEL	ADC Watchdog Lower Threshold Interrupt enable 0: ADC watchdog lower threshold interrupt is disabled 1: ADC watchdog lower threshold interrupt is enabled
[2]	ADIEC	ADC Cycle EOC Interrupt enable 0: ADC cycle end of conversion interrupt is disabled 1: ADC cycle end of conversion interrupt is enabled
[1]	ADIEG	ADC Subgroup EOC Interrupt enable 0: ADC subgroup end of conversion interrupt is disabled 1: ADC subgroup end of conversion interrupt is enabled
[0]	ADIES	ADC Single EOC Interrupt enable 0: ADC single end of conversion interrupt is disabled 1: ADC single end of conversion interrupt is enabled

ADC Interrupt Raw Status Register – ADCIRAW

This register contains the ADC interrupt raw status bits.

Offset: 0x084

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved							ADIRAWO
Type/Reset								RO 0
	23	22	21	20	19	18	17	16
	Reserved						ADIRAWU	ADIRAWL
Type/Reset							RO 0	RO 0
	15	14	13	12	11	10	9	8
	Reserved							
Type/Reset								
	7	6	5	4	3	2	1	0
	Reserved					ADIRAWC	ADIRAWG	ADIRAWS
Type/Reset						RO 0	RO 0	RO 0

Bits	Field	Descriptions
[24]	ADIRAWO	ADC Data Register Overwrite Interrupt Raw Status 0: ADC data register overwrite event does not occur 1: ADC data register overwrite event occurs
[17]	ADIRAWU	ADC Watchdog Upper Threshold Interrupt Raw Status 0: ADC watchdog upper threshold event does not occur 1: ADC watchdog upper threshold event occurs
[16]	ADIRAWL	ADC Watchdog Lower Threshold Interrupt Raw Status 0: ADC watchdog lower threshold event does not occurs 1: ADC watchdog lower threshold event occurs
[2]	ADIRAWC	ADC Watchdog Lower Threshold Interrupt Raw Status 0: ADC watchdog lower threshold event does not occurs 1: ADC watchdog lower threshold event occurs
[1]	ADIRAWG	ADC Subgroup EOC Interrupt Raw Status 0: ADC subgroup end of conversion event does not occur 1: ADC subgroup end of conversion event occurs
[0]	ADIRAWS	ADC Single EOC Interrupt Raw Status 0: ADC single end of conversion event does not occur 1: ADC single end of conversion event occurs

ADC Interrupt Status Register – ADCISR

This register contains the ADC interrupt masked status bits. The corresponding interrupt status will be set to 1 if the associated interrupt event occurs and the related enable bit is set to 1.

Offset: 0x088

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved							ADISRO
Type/Reset								RO 0
	23	22	21	20	19	18	17	16
	Reserved						ADISRU	ADISRL
Type/Reset							RO 0	RO 0
	15	14	13	12	11	10	9	8
	Reserved							
Type/Reset								
	7	6	5	4	3	2	1	0
	Reserved					ADISRC	ADISRG	ADISRS
Type/Reset						RO 0	RO 0	RO 0

Bits	Field	Descriptions
[24]	ADISRO	ADC Data Register Overwrite Interrupt Status 0: ADC data register overwrite interrupt does not occur or the data register overwrite interrupt is disabled. 1: ADC data register overwrite interrupt occurs as the data register overwrite interrupt is enabled.
[17]	ADISRU	ADC Watchdog Upper Threshold Interrupt Status 0: ADC watchdog upper threshold interrupt does not occur or the watchdog upper threshold interrupt is disabled. 1: ADC watchdog upper threshold interrupt occurs as the watchdog upper threshold interrupt is enabled.
[16]	ADISRL	ADC Watchdog Lower Threshold Interrupt Status 0: ADC watchdog lower threshold interrupt does not occur or the watchdog lower threshold interrupt is disabled. 1: ADC watchdog lower threshold interrupt occurs as the watchdog lower threshold interrupt is enabled.
[2]	ADISRC	ADC Cycle EOC Interrupt Status 0: ADC cycle end of conversion interrupt does not occur or the cycle end of conversion interrupt is disabled. 1: ADC cycle end of conversion interrupt occurs as the cycle end of conversion interrupt is enabled.
[1]	ADISRG	ADC Subgroup EOC Interrupt Status 0: ADC subgroup end of conversion interrupt does not occur or the subgroup end of conversion interrupt is disabled. 1: ADC subgroup end of conversion interrupt occurs as the subgroup end of conversion interrupt is enabled.
[0]	ADISRS	ADC Single EOC Interrupt Status 0: ADC single end of conversion interrupt does not occur or the single end of conversion interrupt is disabled. 1: ADC single end of conversion interrupt occurs as the single end of conversion interrupt is enabled.

ADC Interrupt Clear Register – ADCICLR

This register provides the clear bits used to clear the interrupt raw and masked status of the ADC. These bits are set to 1 by software to clear the interrupt status and automatically cleared to 0 by hardware after being set to 1.

Offset: 0x08C

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved							ADICLRO
Type/Reset								WO 0
	23	22	21	20	19	18	17	16
	Reserved						ADICLRU	ADICLRL
Type/Reset							WO 0	WO 0
	15	14	13	12	11	10	9	8
	Reserved							
Type/Reset								
	7	6	5	4	3	2	1	0
	Reserved					ADICLRC	ADICLRG	ADICLRS
Type/Reset						WO 0	WO 0	WO 0

Bits	Field	Descriptions
[24]	ADICLRO	ADC Data Register Overwrite Interrupt Status Clear Bit 0: No effect 1: Clear ADISRO and ADIRAWO bits
[17]	ADICLRU	ADC Watchdog Upper Threshold Interrupt Status Clear Bit 0: No effect 1: Clear ADISRU and ADIRAWU bits
[16]	ADICLRL	ADC Watchdog Lower Threshold Interrupt Status Clear Bit 0: No effect 1: Clear ADISRL and ADIRAWL bits
[2]	ADICLRC	ADC Cycle EOC Interrupt Status Clear Bit 0: No effect 1: Clear ADISRC and ADIRAWC bits
[1]	ADICLRG	ADC Subgroup EOC Interrupt Status Clear Bit 0: No effect 1: Clear ADISRG and ADIRAWG bits
[0]	ADICLRS	ADC Single EOC Interrupt Status Clear Bit 0: No effect 1: Clear ADISRS and ADIRAWS bits

ADC DMA Request Register – ADCDMAR

This register contains the ADC DMA request enable bits.

Offset: 0x090

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	22	21	20	19	18		
	Reserved							
Type/Reset								
	15	14	13	12	11	10		
	Reserved							
Type/Reset								

	7	6	5	4	3	2	1	0
	Reserved					ADDMAC	ADDMAG	ADDMAS
Type/Reset						RW 0	RW 0	RW 0

Bits	Field	Descriptions
[2]	ADDMAC	ADC Cycle EOC DMA Request Enable Bit 0: ADC cycle end of conversion DMA request is disabled 1: ADC cycle end of conversion DMA request is enabled
[1]	ADDMAG	ADC Subgroup EOC DMA Request Enable Bit 0: ADC subgroup end of conversion DMA request is disabled 1: ADC subgroup end of conversion DMA request is enabled
[0]	ADDMAS	ADC Single EOC DMA Request Enable Bit 0: ADC single end of conversion DMA request is disabled 1: ADC single end of conversion DMA request is enabled

13 General-Purpose Timer (GPTM)

Introduction

The General-Purpose Timer consists of one 16-bit up / down-counter, four 16-bit Capture / Compare Registers (CCRs), one 16-bit Counter-Reload Register (CRR) and several control / status registers. It can be used for a variety of purposes including general timer, input signal pulse width measurement or output waveform generation such as single pulse generation or PWM output. The GPTM supports an encoder interface using a quadrature decoder with two inputs.

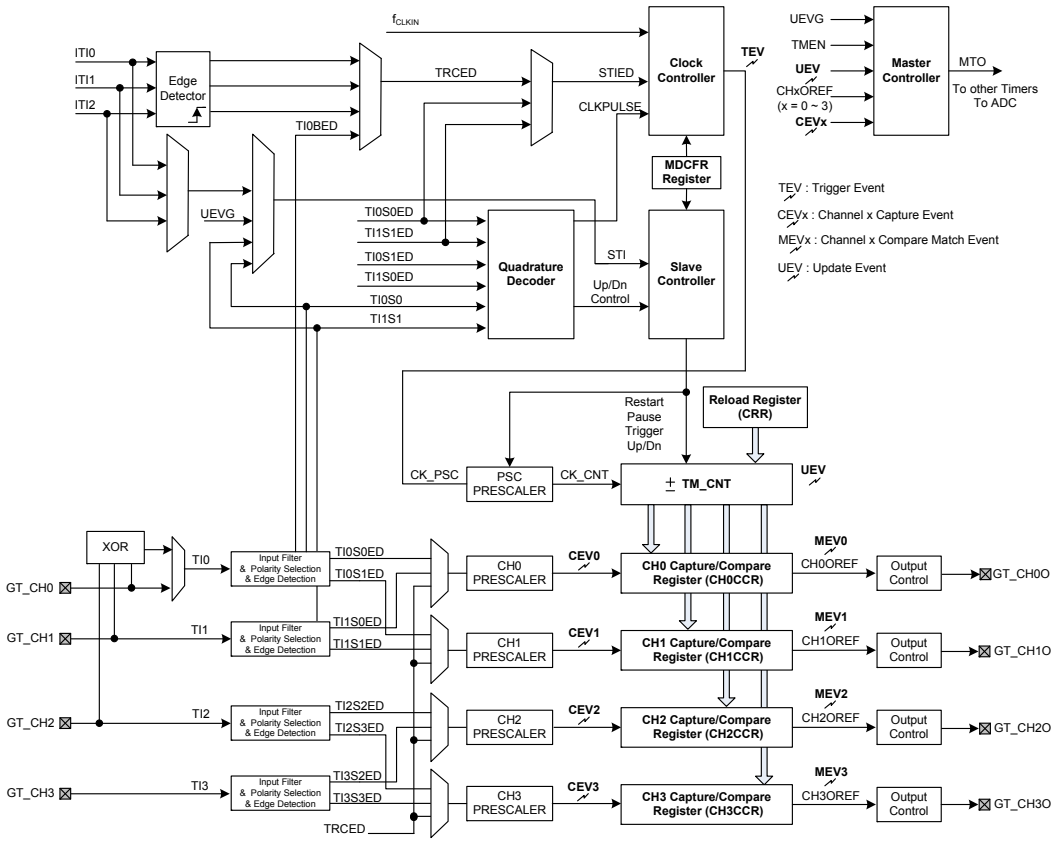


Figure 31. GPTM Block Diagram

Features

- 16-bit up / down auto-reload counter
- 16-bit programmable prescaler that allows division of the counter clock frequency by any factor between 1 and 65536
- Up to 4 independent channels for:
  - Input Capture function
  - Compare Match Output

- Generation of PWM waveform – Edge and Center-aligned Mode
- Single Pulse Mode Output

- Encoder interface controller with two inputs using quadrature decoder
- Synchronization circuit to control the timer with external signals and to interconnect several timers together
- Interrupt / PDMA generation with the following events:
  - Update event
  - Trigger event
  - Input capture event
  - Output compare match event
- GPTM Master / Slave mode controller

Functional Descriptions

Counter Mode

Up-Counting

In this mode the counter counts continuously from 0 to the counter-reload value, which is defined in the CRR register, in a count-up direction. Once the counter reaches the counter-reload value, the Timer Module generates an overflow event and the counter restarts to count once again from 0. This action will continue repeatedly. The counting direction bit DIR in the CNTCFR register should be set to 0 for the up-counting mode.

When the update event is generated by setting the UEVG bit in the EVGR register to 1, the counter value will also be initialized to 0.

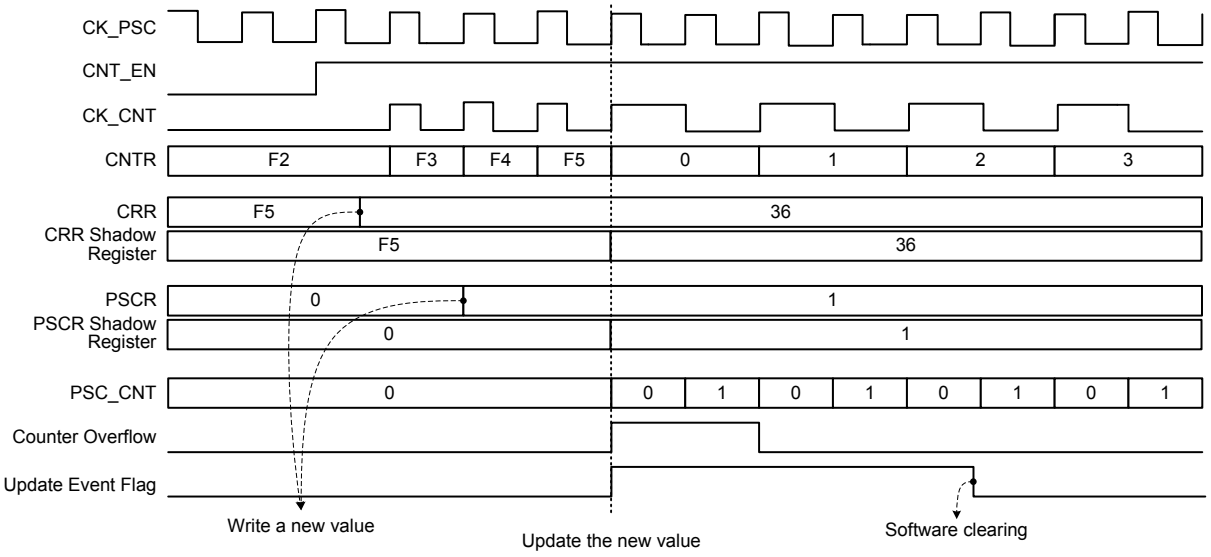


Figure 32. Up-counting Example

Down-Counting

In this mode the counter counts continuously from the counter-reload value, which is defined in the CRR register, to 0 in a count-down direction. Once the counter reaches 0, the Timer module generates an underflow event and the counter restarts to count once again from the counter-reload value. This action will continue repeatedly. The counting direction bit DIR in the CNTCFR register should be set to 1 for the down-counting mode.

When the update event is set by the UEVG bit in the EVGR register, the counter value will also be initialized to the counter-reload value.

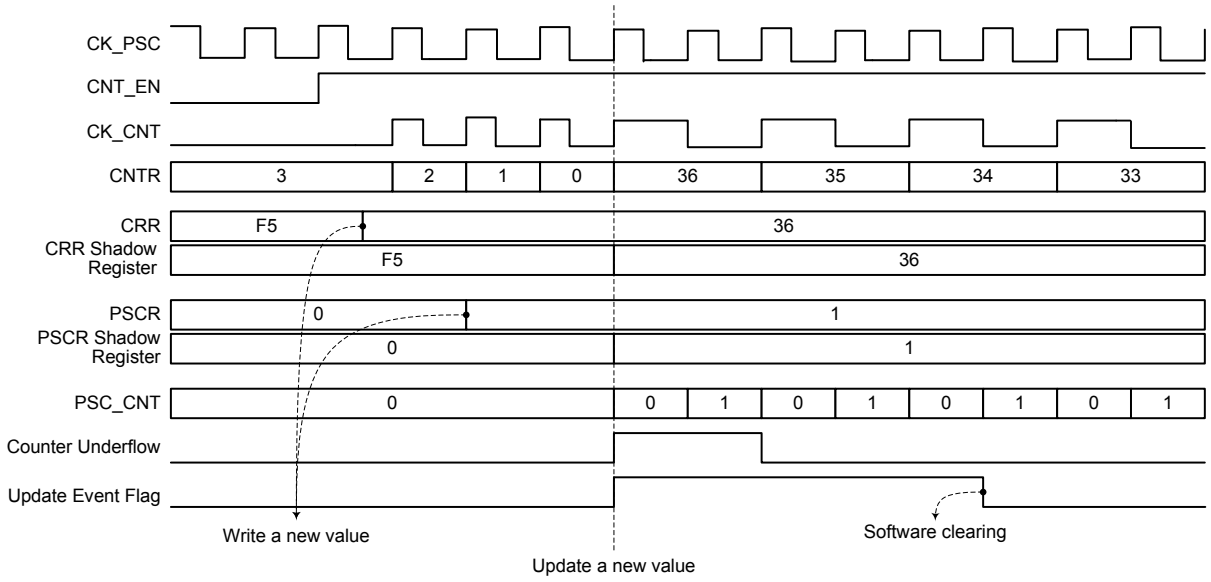


Figure 33. Down-counting Example

Center-Aligned Counting

In the center-aligned counting mode, the counter counts up from 0 to the counter-reload value and then counts down to 0 alternatively. The Timer module generates an overflow event when the counter counts to the counter-reload value in the up-counting mode and generates an underflow event when the counter counts to 0 in the down-counting mode. The counting direction bit DIR in the CNTCFR register is read-only and indicates the counting direction when in the center-aligned mode. The counting direction is updated by hardware automatically.

Setting the UEVG bit in the EVGR register will initialize the counter value to 0 irrespective of whether the counter is counting up or down in the center-aligned counting mode.

The UEVIF bit in the INTSR register will be set to 1 when an overflow or underflow event or both of them occur according to the CMSEL field setting in the CNTCFR register.

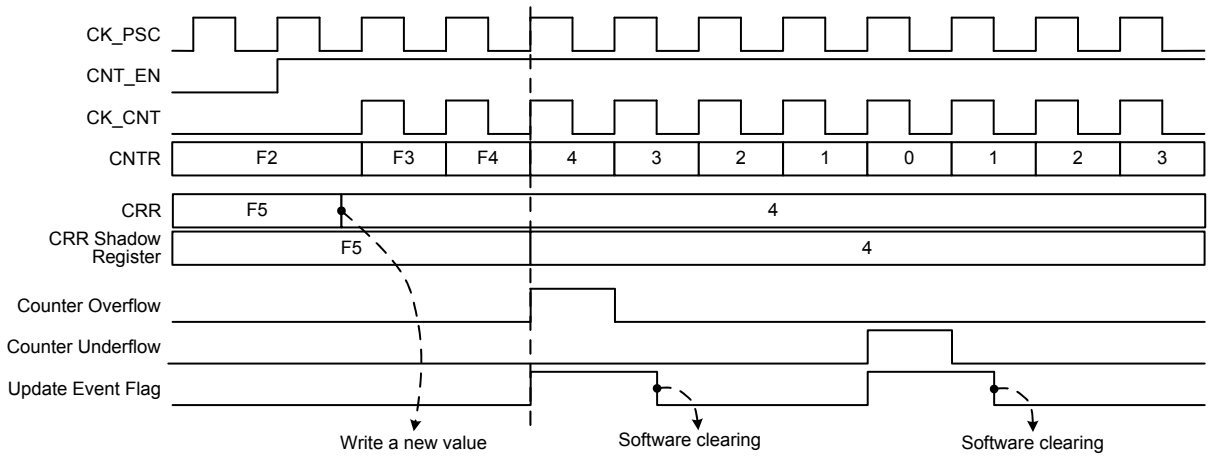


Figure 34. Center-aligned Counting Example

Clock Controller

The following describes the Timer Module clock controller which determines the clock source of the internal prescaler counter.

- Internal APB clock f_{CLKIN}

  The default internal clock source is the APB clock f_{CLKIN} used to drive the counter prescaler when the slave mode is disabled. When the slave mode selection bits SMSEL in the MDCFR register are set to 0x4, 0x5 or 0x6, the internal APB clock f_{CLKIN} is the counter prescaler driving clock source. If the slave mode controller is enabled by setting SMSEL field in the MDCFR register to an available value including 0x1, 0x2, 0x3 and 0x7, the prescaler is clocked by other clock sources selected by the TRSEL field in the TRCFR register and described as follows.

- Quadrature Decoder

  To select Quadrature Decoder mode the SMSEL field should be set to 0x1, 0x2 or 0x3 in the MDCFR register. The Quadrature Decoder function uses two input states of the GT_CH0 and GT_CH1 pins to generate the clock pulse to drive the counter prescaler. The counting direction bit DIR is modified by hardware automatically at each transition on the input source signal. The input source signal can be derived from the GT_CH0 pin only, the GT_CH1 pin only or both GT_CH0 and GT_CH1 pins.

- STIED

  The counter prescaler can count during each rising edge of the STI signal. This mode can be selected by setting the SMSEL field to 0x7 in the MDCFR register. Here the counter will act as an event counter. The input event, known as STI here, can be selected by setting the TRSEL field to an available value except the value of 0x0. When the STI signal is selected as the clock source, the internal edge detection circuitry will generate a clock pulse during each STI signal rising edge to drive the counter prescaler. It is important to note that if the TRSEL field is set to 0x0 to select the software UEVG bit as the trigger source, then when the SMSEL field is set to 0x7, the counter will be updated instead of counting.

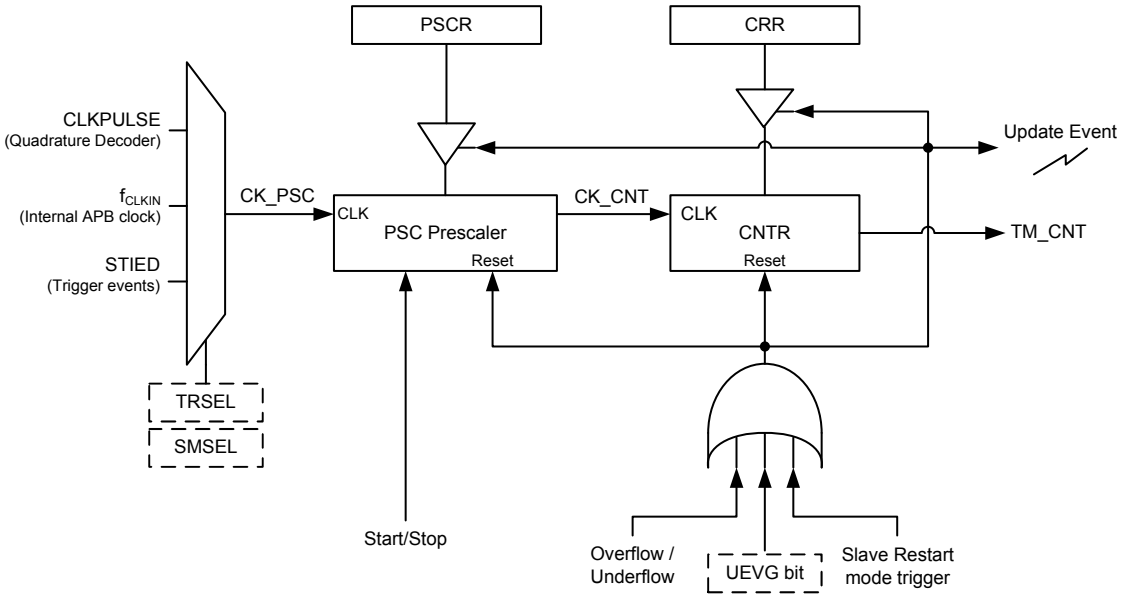


Figure 35. GPTM Clock Source Selection

Trigger Controller

The trigger controller is used to select the trigger source and setup the trigger level or edge trigger condition. For the internal trigger input, it can be selected by the Trigger Selection bits TRSEL in the TRCFR register. For all the trigger sources except the UEVG bit software trigger, the internal edge detection circuitry will generate a clock pulse at each trigger signal rising edge to stimulate some GPTM functions which are triggered by a trigger signal rising edge.

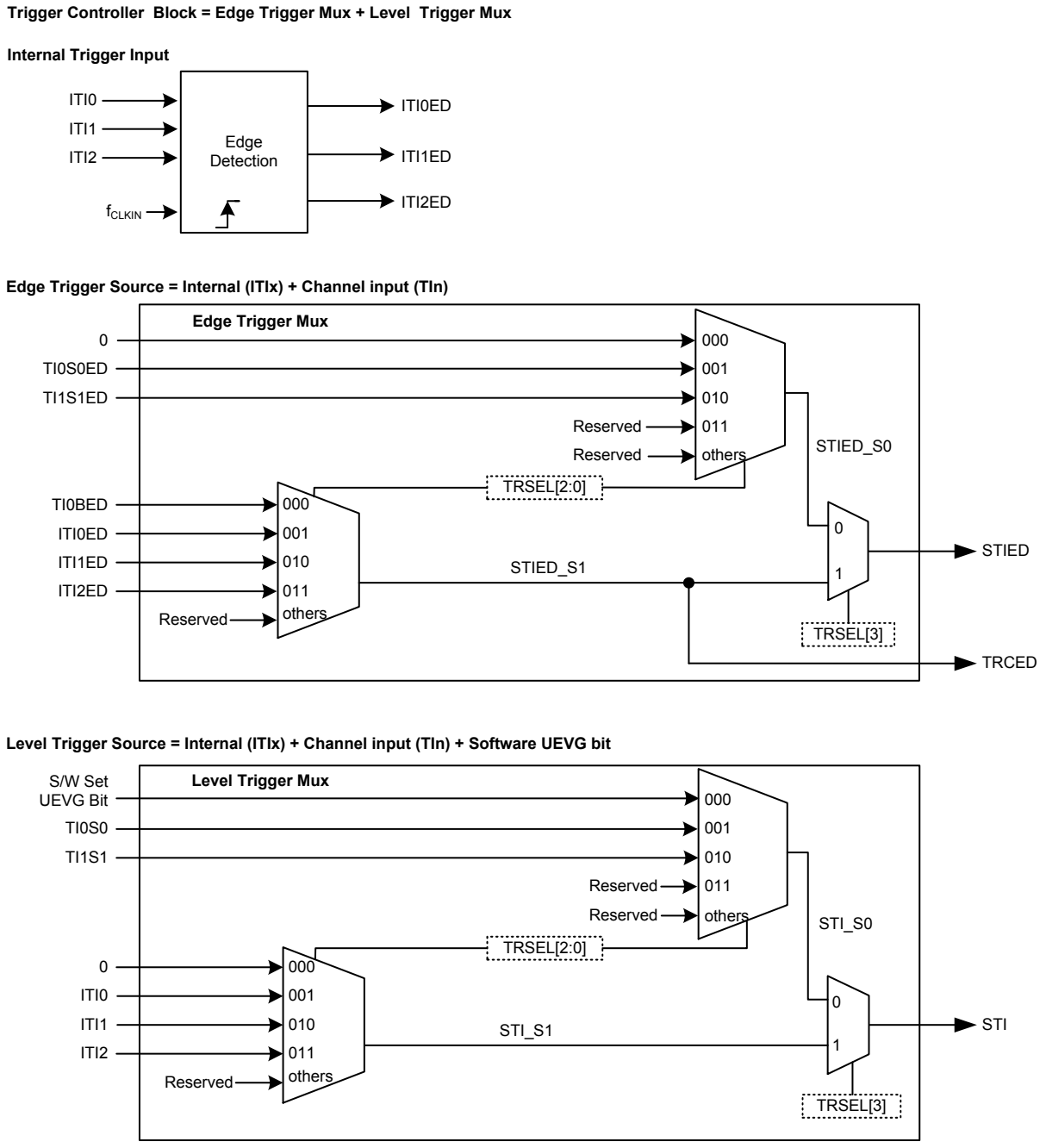


Figure 36. Trigger Controller Block

Slave Controller

The GPTM can be synchronized with an external trigger in several modes including the Restart mode, the Pause mode and the Trigger mode which can be selected by the SMSEL field in the MDCFR register. The trigger input of these modes comes from the STI signal which is selected by the TRSEL field in the TRCFR register. The operation modes in the Slave Controller are described in the accompanying sections.

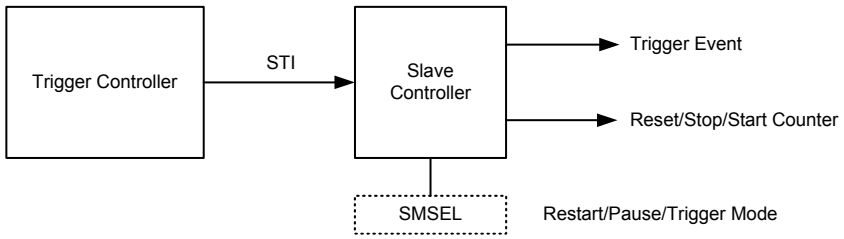


Figure 37. Slave Controller Diagram

Restart Mode

The counter and its prescaler can be reinitialized in response to a rising edge of the STI signal. When an STI rising edge occurs, the update event software generation bit named UEVG will automatically be asserted by hardware and the trigger event flag will also be set. Then the counter and prescaler will be reinitialized. Although the UEVG bit is set to 1 by hardware, the update event does not really occur. It depends upon whether the update event disable control bit UEVDIS is set to 1 or not. If the UEVDIS is set to 1 to disable the update event to occur, there will no update event be generated, however the counter and prescaler are still reinitialized when the STI rising edge occurs. If the UEVDIS bit in the CNTCFR register is cleared to enable the update event to occur, an update event will be generated together with the STI rising edge, then all the preloaded registers will be updated.

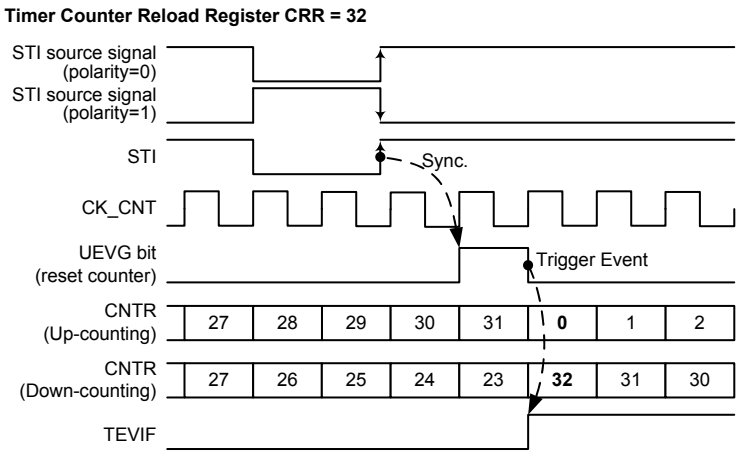


Figure 38. GPTM in Restart Mode

Pause Mode

In the Pause Mode, the selected STI input signal level is used to control the counter start / stop operation. The counter starts to count when the selected STI signal is at a high level and stops counting when the STI signal is changed to a low level, here the counter will maintain its present value and will not be reset. Since the Pause function depends upon the STI level to control the counter stop / start operation, the selected STI trigger signal can not be derived from the TI0BED signal.

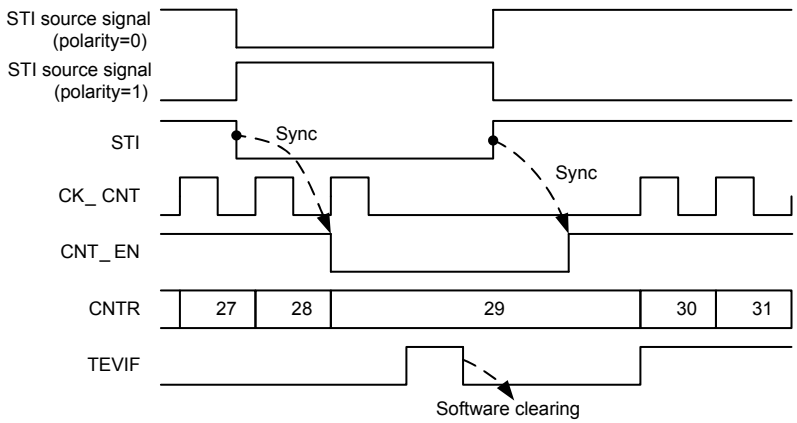


Figure 39. GPTM in Pause Mode

Trigger Mode

After the counter is disabled to count, the counter can resume counting when an STI rising edge signal occurs. When an STI rising edge occurs, the counter will start to count from the current value in the counter. Note that if the STI signal is selected to be derived from the UEVG bit software trigger, the counter will not resume counting. When software triggering using the UEVG bit is selected as the STI source signal, there will be no clock pulse generated which can be used to make the counter resume counting. Note that the STI signal is only used to enable the counter to resume counting and has no effect on controlling the counter to stop counting.

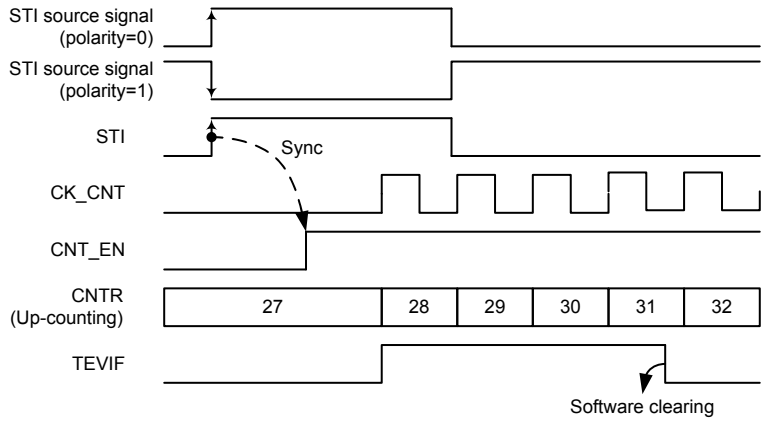


Figure 40. GPTM in Trigger Mode

Master Controller

The GPTMs and MCTMs can be linked together internally for timer synchronization or chaining. When one GPTM is configured to be in the Master Mode, the GPTM Master Controller will generate a Master Trigger Output (MTO) signal which includes a reset, a start, a stop signal or a clock source which is selected by the MMSEL field in the MDCFR register to trigger or drive another GPTM or MCTM, if exists, which is configured in the Slave Mode.

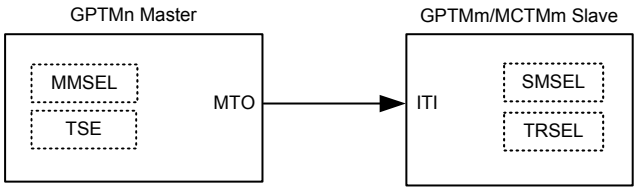


Figure 41. Master GPTMn and Slave GPTMm / MCTMm Connection

The Master Mode Selection bits, MMSEL, in the MDCFR register are used to select the MTO source for synchronizing another slave GPTM or MCTM if exists.

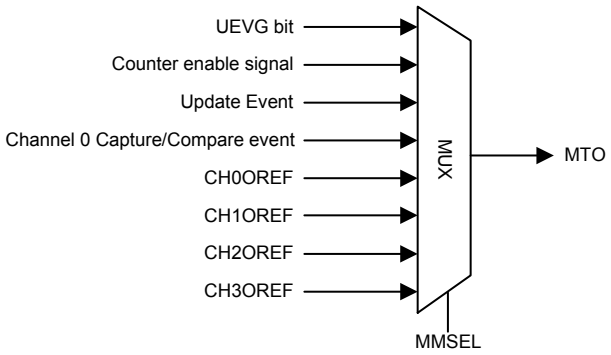


Figure 42. MTO Selection

For example, setting the MMSEL field to 0x5 is to select the CH1OREF signal as the MTO signal to synchronize another slave GPTM or MCTM. For a more detailed description, refer to the related MMSEL field definitions in the MDCFR register.

Channel Controller

The GPTM has four independent channels which can be used as capture inputs or compare match outputs. Each capture input or compare match output channel is composed of a preload register and a shadow register. Data access of the APB bus is always implemented by reading / writing the preload register.

When used in the input capture mode, the counter value is captured into the CHxCCR shadow register first and then transferred into the CHxCCR preload register when the capture event occurs.

When used in the compare match output mode, the contents of the CHxCCR preload register is copied into the associated shadow register, the counter value is then compared with the register value.

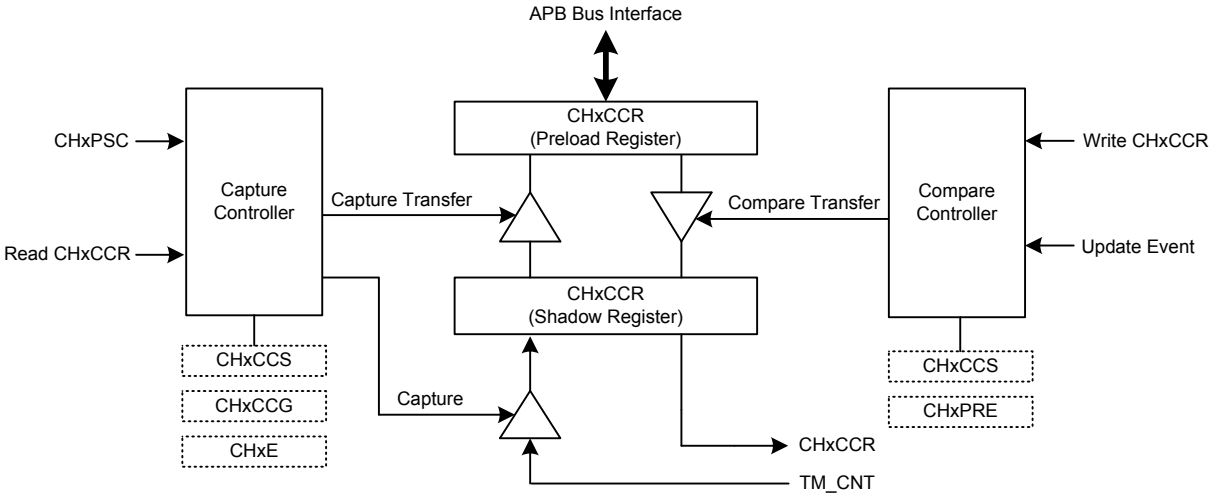


Figure 43. Capture / Compare Block Diagram

Capture Counter Value Transferred to CHxCCR

When the channel is used as a capture input, the counter value is captured into the Channel Capture / Compare Register (CHxCCR) when an effective input signal transition occurs. Once the capture event occurs, the CHxCCIF flag in the INTSR register is set accordingly. If the CHxCCIF bit is already set, i.e., the flag has not yet been cleared by software, and another capture event on this channel occurs, the corresponding channel Over-Capture flag, named CHxOCF, will be set.

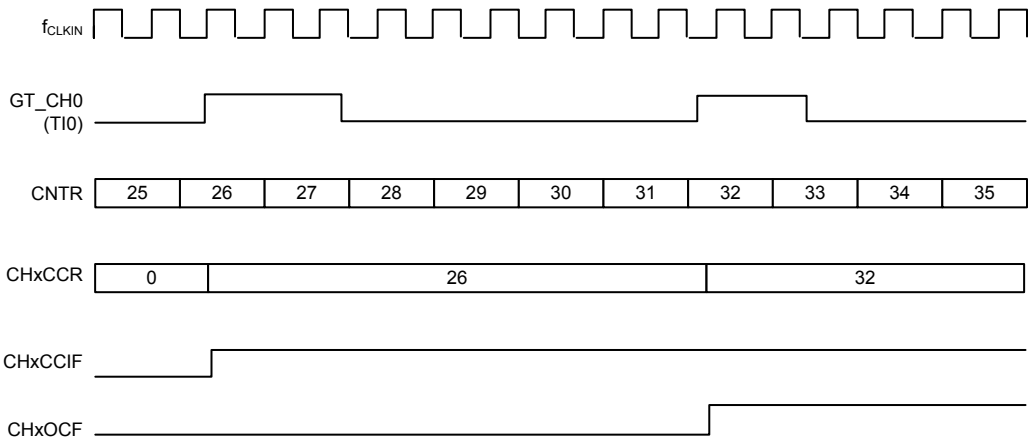


Figure 44. Input Capture Mode

Pulse Width Measurement

The input capture mode can be also used for pulse width measurement from signals on the GT_CHx pins, TIx. The following example shows how to configure the GPTM operated in the input capture mode to measure the high pulse width and the input period on the GT_CH0 pin using channel 0 and channel 1. The basic steps are shown as follows.

- Configure the capture channel 0 (CH0CCS = 0x1) to select the TI0 signal as the capture input.
- Configure the CH0P bit to 0 to choose the rising edge of the TI0 input as the active polarity.
- Configure the capture channel 1 (CH1CCS = 0x2) to select the TI0 signal as the capture input.
- Configure the CH1P bit to 1 to choose the falling edge of the TI0 input as the active polarity.
- Configure the TRSEL bits to 0x1 to select TI0S0 as the trigger input.
- Configure the Slave controller to operate in the Restart mode by setting the SMSEL field in the MDCFR register to 0x4
- Enable the input capture mode by setting the CH0E and CH1E bits in the CHCTR register to 1.

As the following diagram shows, the high pulse width on the GT_CH0 pin will be captured into the CH1CCR register while the input period will be captured into the CH0CCR register after input capture operation.

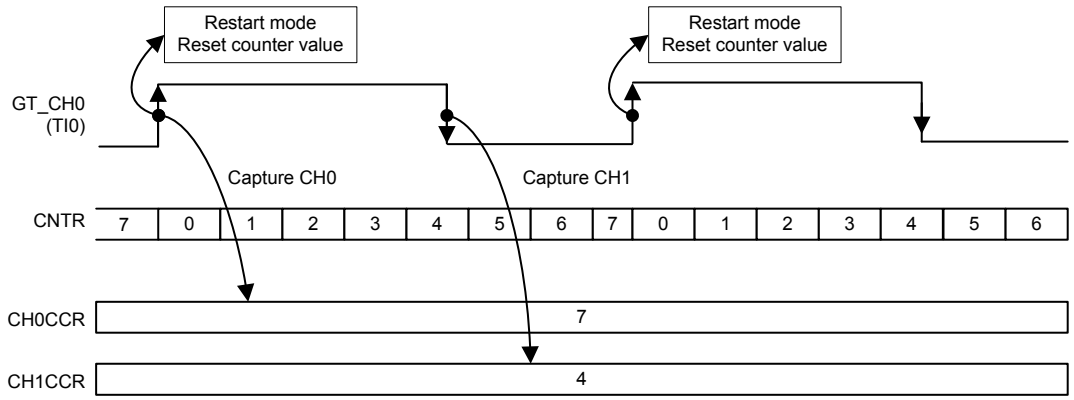


Figure 45. PWM Pulse Width Measurement Example

Input Stage

The input stage consists of a digital filter, a channel polarity selection, edge detection and a channel prescaler. The channel 0 input signal (TI0) can be chosen to come from the GT_CH0 signal or the Excusive-OR function of the GT_CH0, GT_CH1 and GT_CH2 signals. The channel input signal (TIx) is sampled by a digital filter to generate a filtered input signal TIxFP. Then the channel polarity and the edge detection block can generate a TIxS0ED or TIxS1ED signal for the input capture function. The effective input event number can be set by the channel capture input source prescaler setting field (CHxPSC).

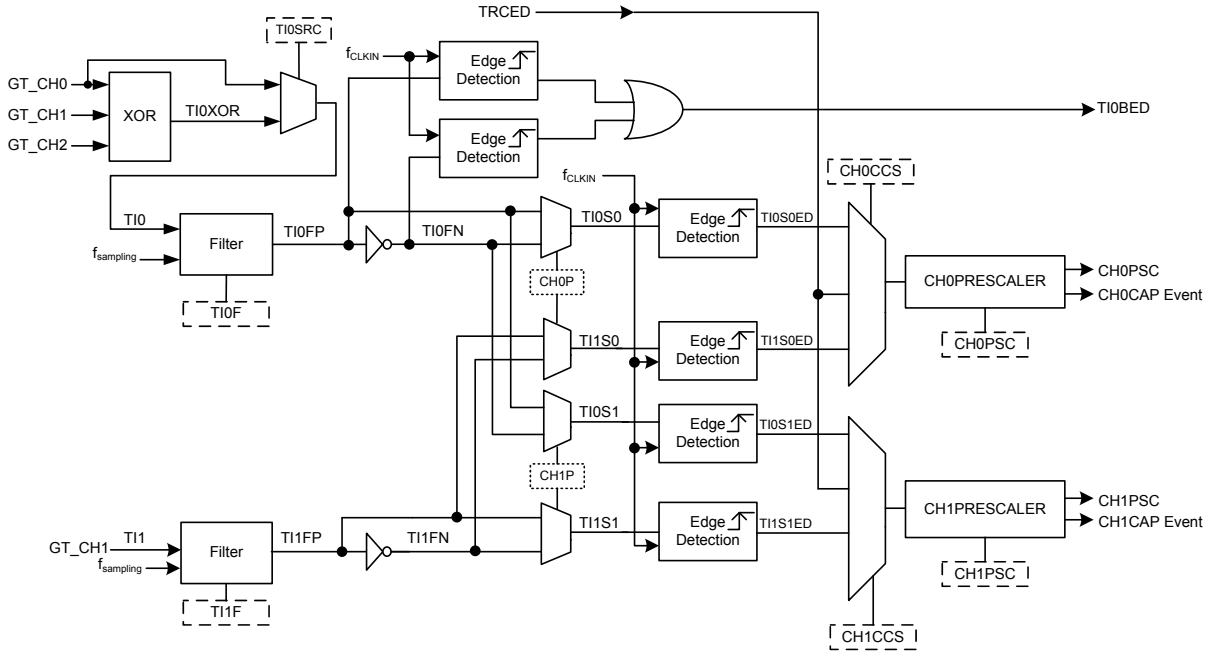


Figure 46. Channel 0 and Channel 1 Input Stages

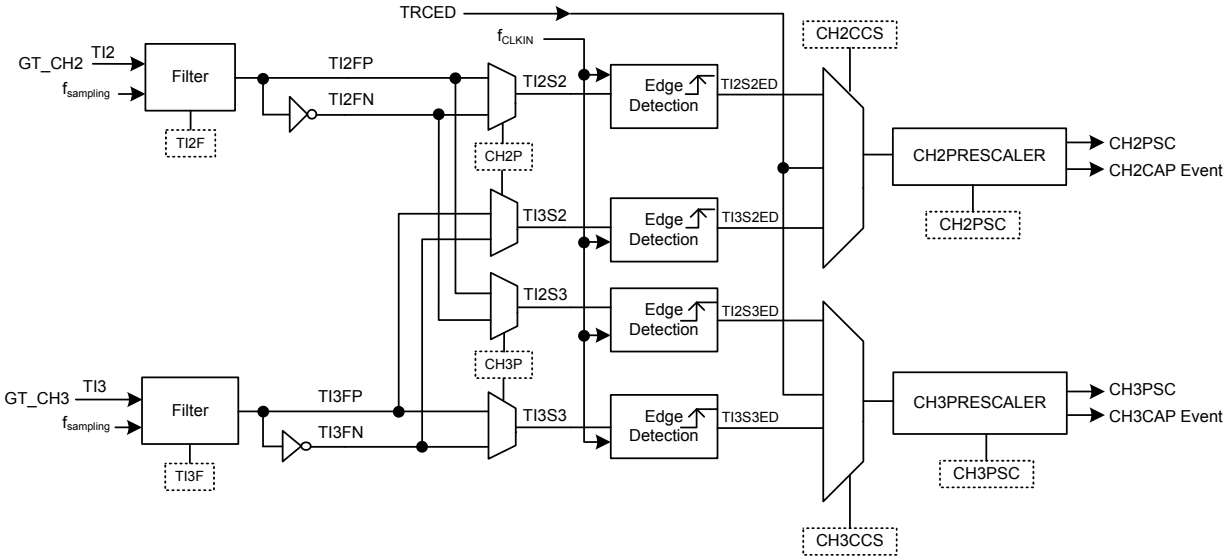


Figure 47. Channel 2 and Channel 3 Input Stages

Digital Filter

The digital filters are embedded in the input stage for the GT_CH0 ~ GT_CH3 pins respectively. The digital filter in the GPTM is an N-event counter where N refers to how many valid transitions are necessary to output a filtered signal. The N value can be 0, 2, 4, 5, 6 or 8 according to the user selection for each filter.

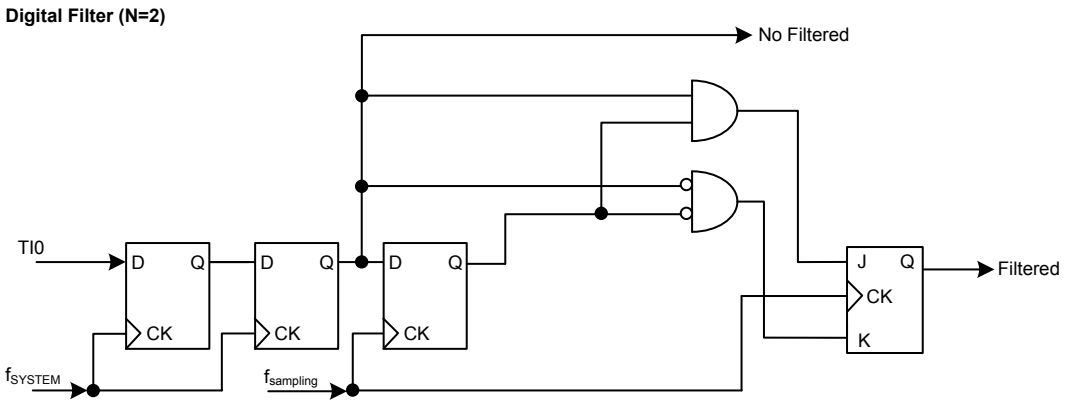


Figure 48. TI0 Digital Filter Diagram with N = 2

Quadrature Decoder

The Quadrature Decoder function uses two quadrantal inputs TI0 and TI1 derived from the GT_CH0 and GT_CH1 pins respectively to interact to generate the counter value. The DIR bit is modified by hardware automatically during each input source transition. The input source can be either TI0 only, TI1 only or both TI0 and TI1, the selection made by setting the SMSEL field to 0x01, 0x02 or 0x03. The mechanism for changing the counter direction is shown in the following table. The Quadrature decoder can be regarded as an external clock with a directional selection. This means that the counter counts continuously in the interval between 0 and the counter-reload value. Therefore, users must configure the CRR register before the counter starts to count.

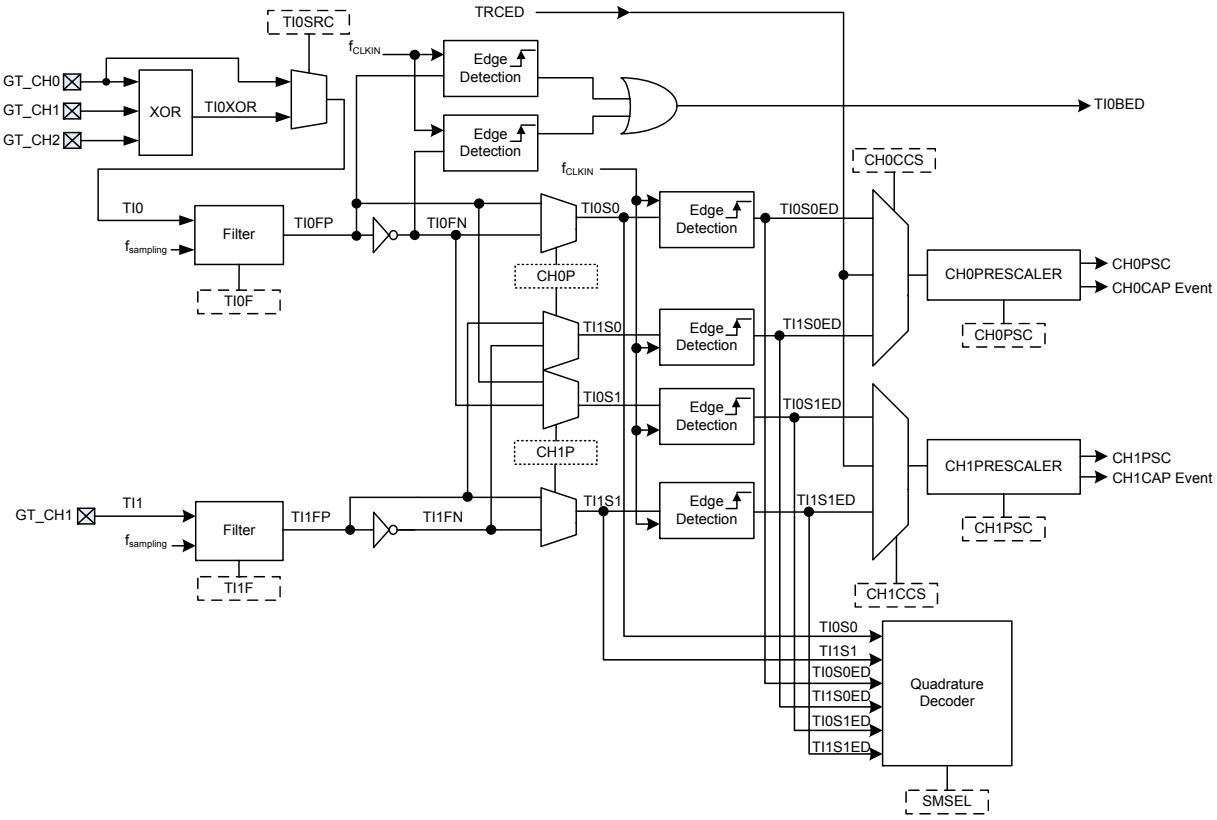


Figure 49. Input Stage and Quadrature Decoder Block Diagram

Table 29. Counting Direction and Encoding Signals

Counting Mode	Level	TI0S0		TI1S1	
		Rising	Falling	Rising	Falling
Counting on TI0 only (SMSEL = 0x1)	TI1S1 = High	Down	Up	—	—
	TI1S1 = Low	Up	Down	—	—
Counting on TI1 only (SMSEL = 0x2)	TI0S0 = High	—	—	Up	Down
	TI0S0 = Low	—	—	Down	Up
Counting on TI0 and TI1 (SMSEL = 0x3)	TI1S1 = High	Down	Up	X	X
	TI1S1 = Low	Up	Down	X	X
	TI0S0 = High	X	X	Up	Down
	TI0S0 = Low	X	X	Down	Up

Note: "—" → "no counting"; "X" → impossible

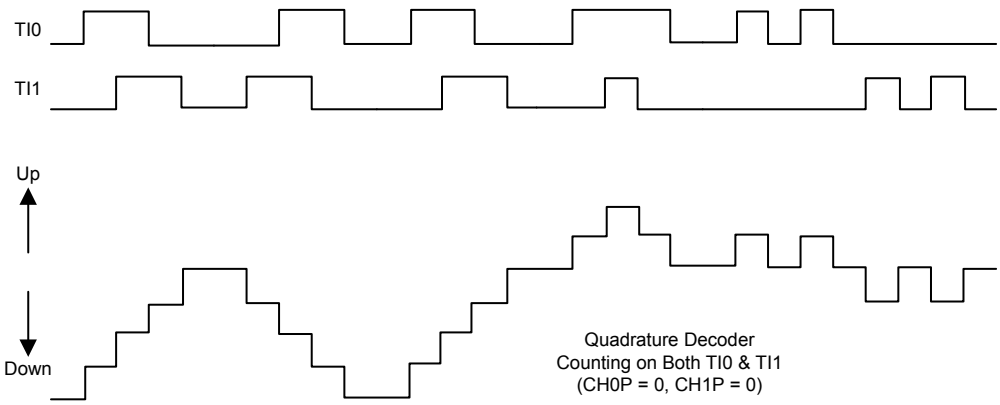


Figure 50. Both TI0 and TI1 Quadrature Decoder Counting

Output Stage

The GPTM has four channels for compare match, single pulse or PWM output function. The channel output GT_CHxO is controlled by the CHxOM, CHxP and CHxE bits in the corresponding CHxOCFR, CHPOLR and CHCTR registers.

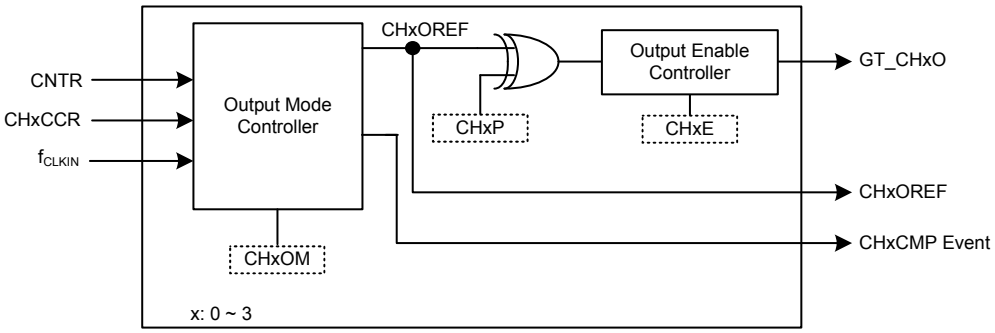


Figure 51. Output Stage Block Diagram

Channel Output Reference Signal

When the GPTM is used in the compare match output mode, the CHxOREF signal (Channel x Output Reference signal) is defined by the CHxOM bit setup. The CHxOREF signal has several types of output function which defines what happens to the output when the counter value matches the contents of the CHxCCR register. In addition to the low, high and toggle CHxOREF output types; there are also PWM mode 1 and PWM mode 2 outputs. In these modes, the CHxOREF signal level is changed according to the count direction and the relationship between the counter value and the CHxCCR content. There are also two modes which will force the output into an inactive or active state irrespective of the CHxCCR content or counter values. With regard to a more detailed description refer to the relative bit definition. The accompanying Table 30 shows a summary of the output type setup.

Table 30. Compare Match Output Setup

CHxOM Value	Compare Match Level
0x0	No change
0x1	Clear Output to 0
0x2	Set Output to 1
0x3	Toggle Output
0x4	Force Inactive Level
0x5	Force Active Level
0x6	PWM Mode 1
0x7	PWM Mode 2

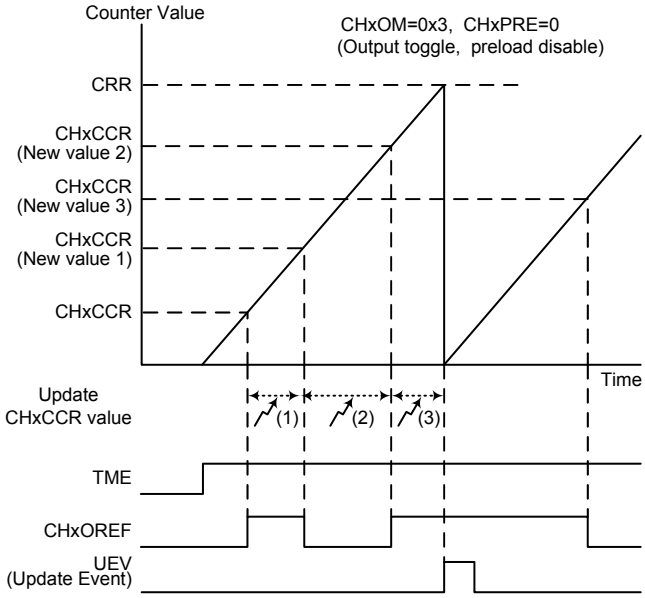


Figure 52. Toggle Mode Channel Output Reference Signal – CHxPRE = 0

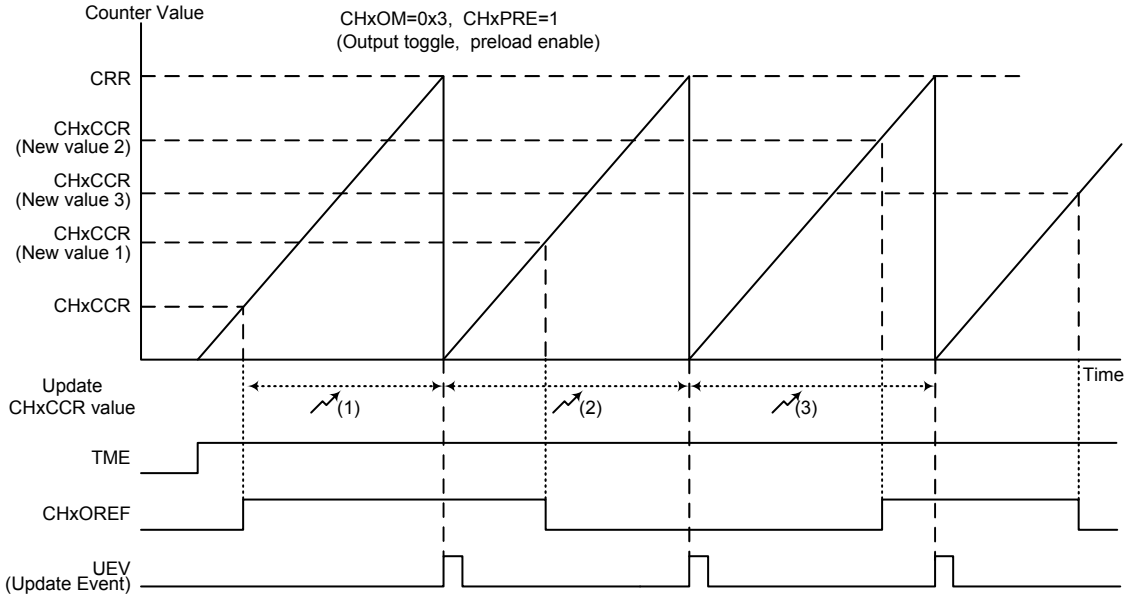


Figure 53. Toggle Mode Channel Output Reference Signal – CHxPRE = 1

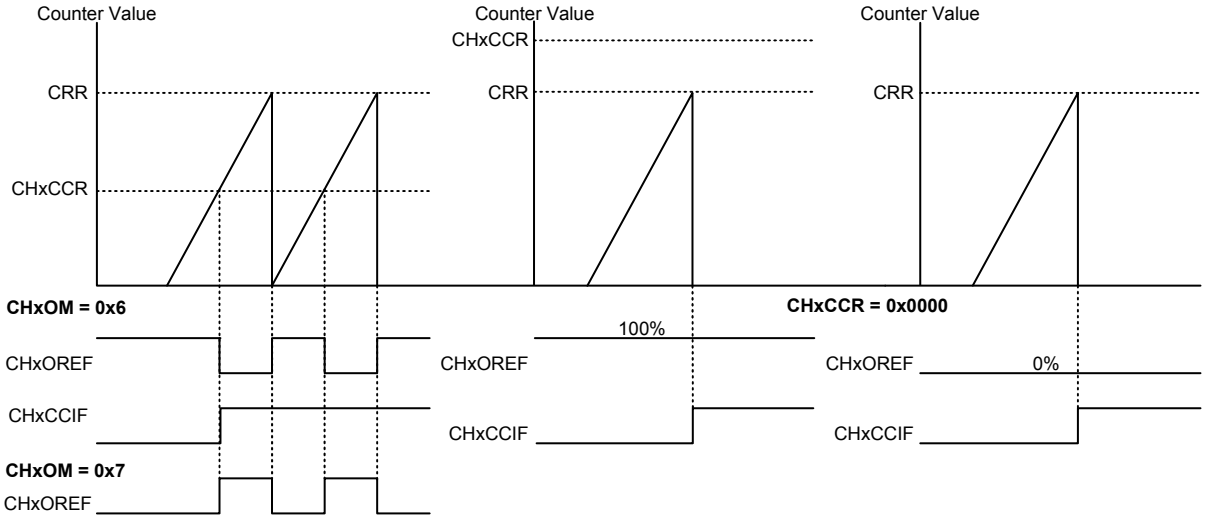


Figure 54. PWM Mode Channel Output Reference Signal and Counter in Up-counting Mode

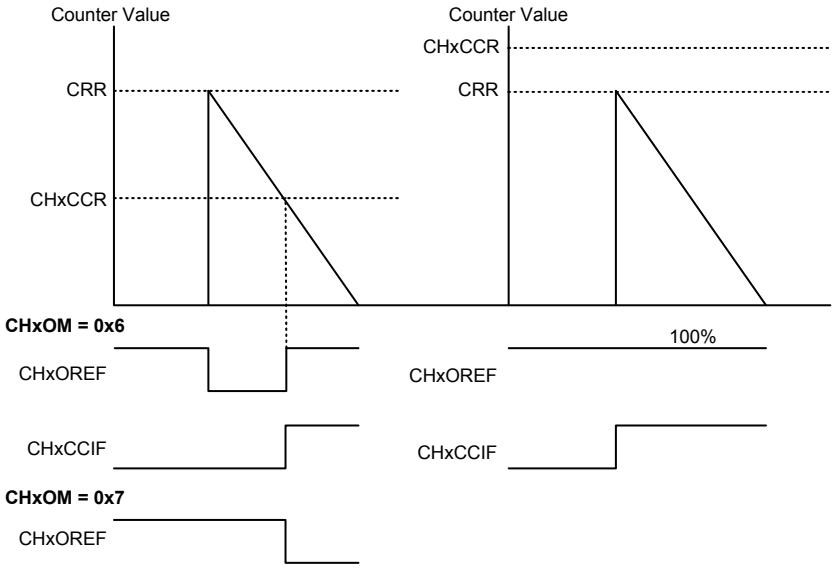


Figure 55. PWM Mode Channel Output Reference Signal and Counter in Down-counting Mode

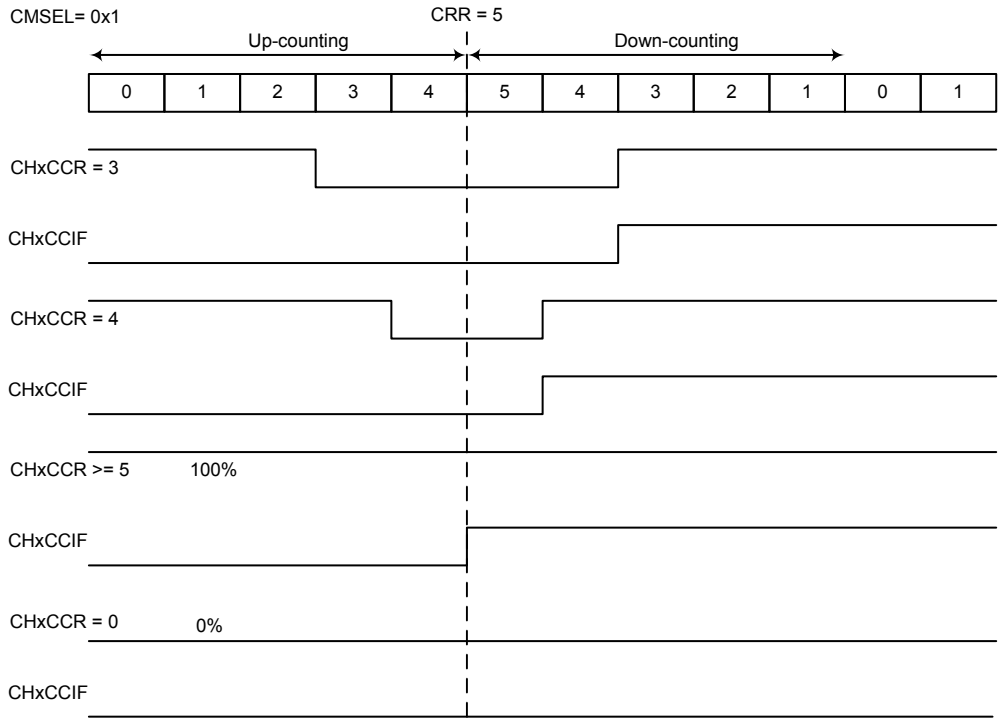


Figure 56. PWM Mode Channel Output Reference Signal and Counter in Centre-aligned Mode

Update Management

The Update event is used to update the CRR, the PSCR and the CHxCCR values from the actual registers to the corresponding shadow registers. An update event occurs when the counter overflows or underflows, the software update control bit is triggered or an update event from the slave controller is generated.

The UEVDIS bit in the CNTCFR register can determine whether the update event occurs or not. When the update event occurs, the corresponding update event interrupt will be generated depending upon whether the update event interrupt generation function is enabled or not by configuring the UGDIS bit in the CNTCFR register. For more detail description, refer to the UEVDIS and UGDIS bit definition in the CNTCFR register.

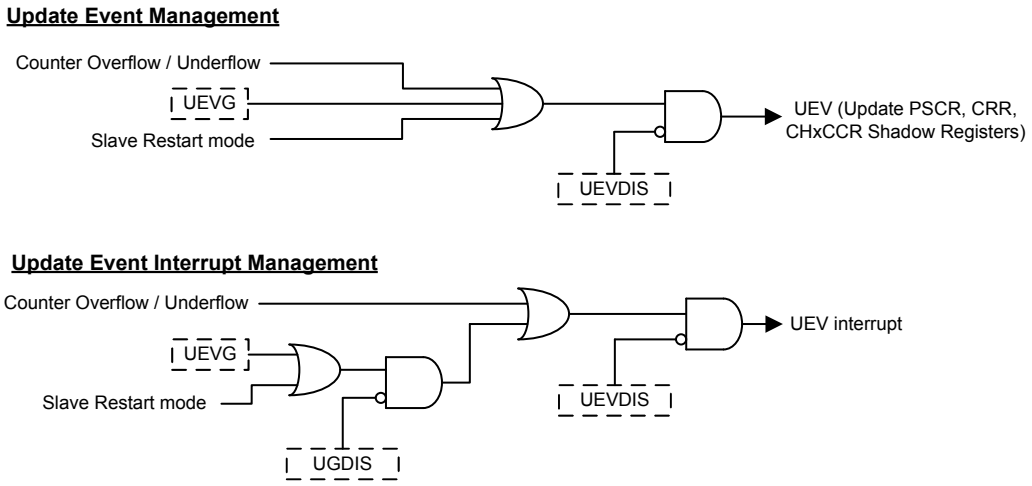


Figure 57. Update Event Setting Diagram

Single Pulse Mode

Once the timer is set to operate in the single pulse mode, it is not necessary to set the timer enable bit TME in the CTR register to 1 to enable the counter. The trigger to generate a pulse can be sourced from the STI signal rising edge or by setting the TME bit to 1 using software. Setting the TME bit to 1 or a trigger from the STI signal rising edge can generate a pulse and then keep the TME bit at a high state until the update event occurs or the TME bit is written to 0 by software. If the TME bit is cleared to 0 using software, the counter will be stopped and its value held. If the TME bit is automatically cleared to 0 by a hardware update event, the counter will be reinitialized.

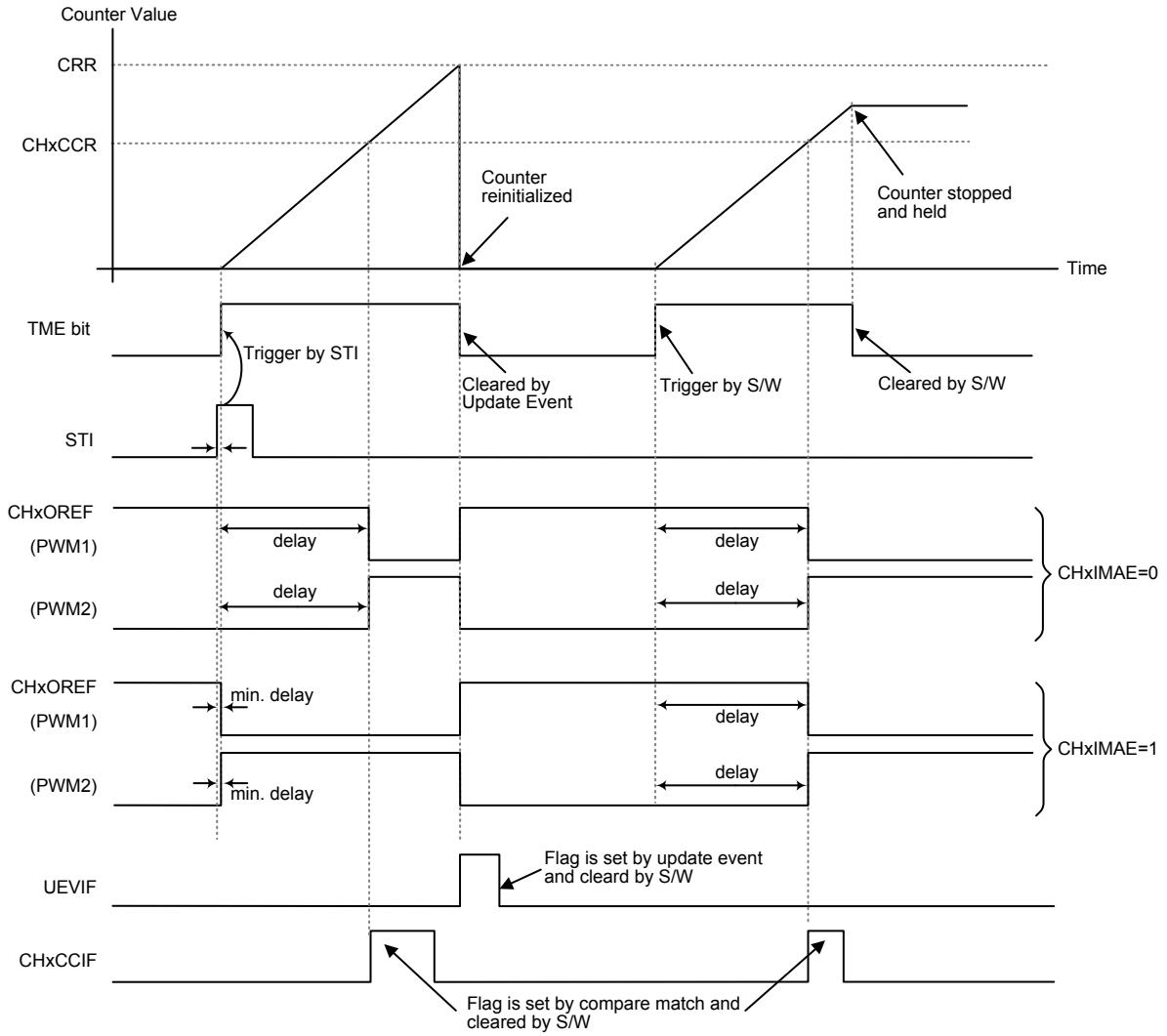


Figure 58. Single Pulse Mode

In the Single Pulse mode, the STI active edge which sets the TME bit to 1 will enable the counter. However, there exist several clock delays to perform the comparison result between the counter value and the CHxCCR value. In order to reduce the delay to a minimum value, the user can set the CHxIMAE bit in each CHxOCFR register. After an STI rising edge trigger occurs in the single pulse mode, the CHxOREF signal will immediately be forced to the state which the CHxOREF signal will change to as the compare match event occurs without taking the comparison result into account. The CHxIMAE bit is available only when the output channel is configured to operate in the PWM1 or PWM2 output mode and the trigger source is derived from the STI signal.

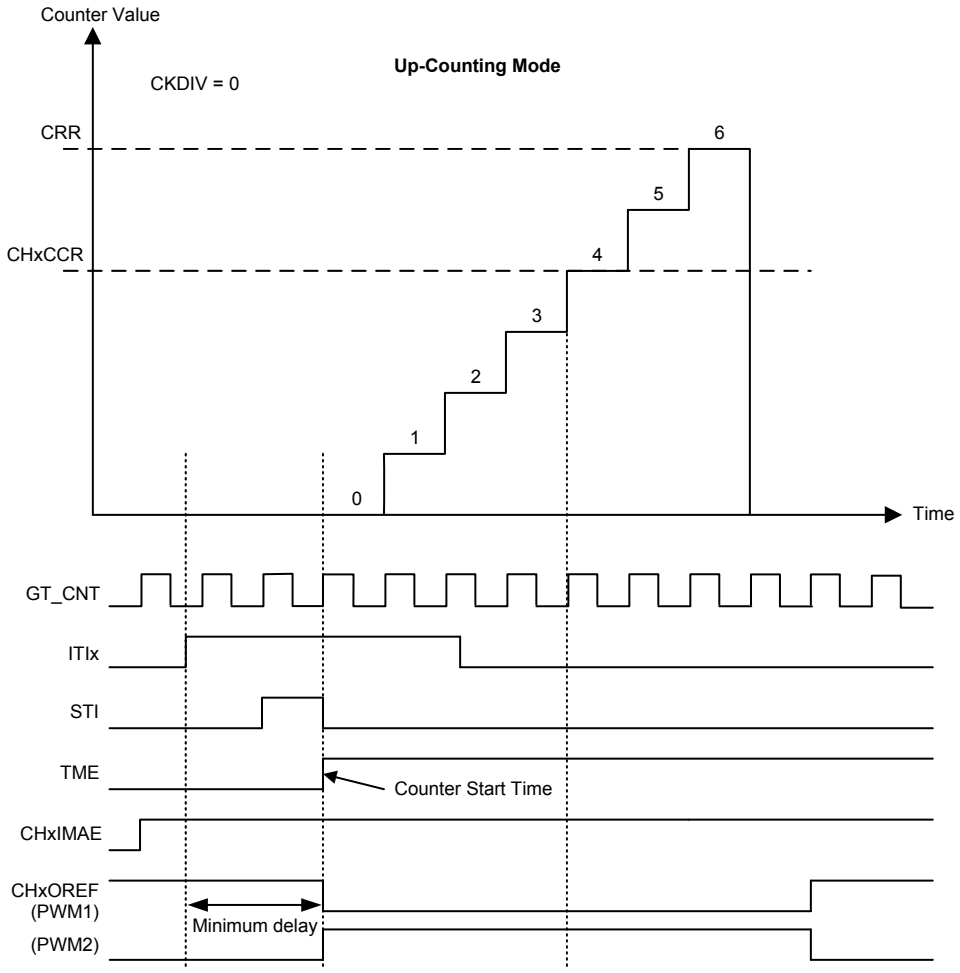


Figure 59. Immediate Active Mode Minimum Delay

Asymmetric PWM Mode

Asymmetric PWM mode allows two center-aligned PWM signals to be generated with a programmable phase shift. While the PWM frequency is determined by the value of the CRR register, the duty cycle and the phase-shift are determined by the CHxCCR and CHxACR register. When the counter is counting up, the PWM uses the value in CHxCCR as up-count compare value. When the counter is into counting down stage, the PWM uses the value in CHxACR as down-count compare value. The Figure 60 is shown as an example for asymmetric PWM mode in center-aligned counting mode.

Note: Asymmetric PWM mode can only be operated in center-aligned counting mode.

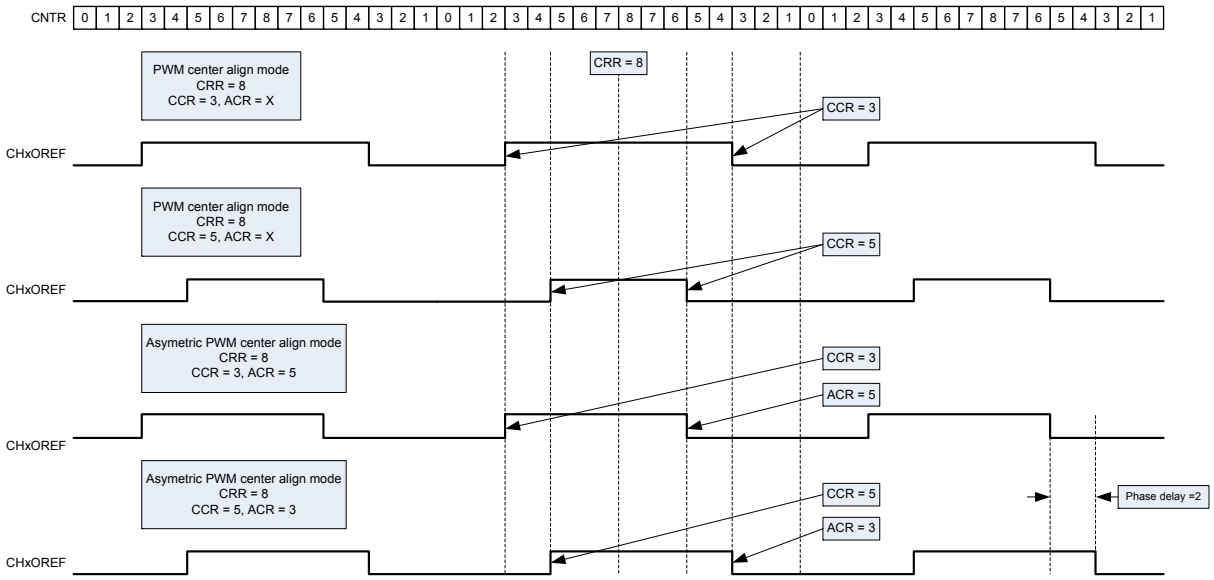


Figure 60. Asymmetric PWM Mode versus Center-aligned Counting Mode

Timer Interconnection

The timers can be internally connected together for timer chaining or synchronization. This can be implemented by configuring one timer to operate in the Master mode while configuring another timer to be in the Slave mode. The following figures present several examples of trigger selection for the master and slave modes.

Using One Timer to Enable / Disable Another Timer Start or Stop Counting

- Configure GPTM0 as the master mode to send its channel 0 Output Reference signal CH0OREF as a trigger output (MMSEL = 0x4).
- Configure GPTM0 CH0OREF waveform.
- Configure GPTM1 to receive its input trigger source from the GPTM0 trigger output (TRSEL = 0x9).
- Configure GPTM1 to operate in the pause mode (SMSEL = 0x5).
- Enable GPTM1 by writing '1' to the TME bit.
- Enable GPTM0 by writing '1' to the TME bit.

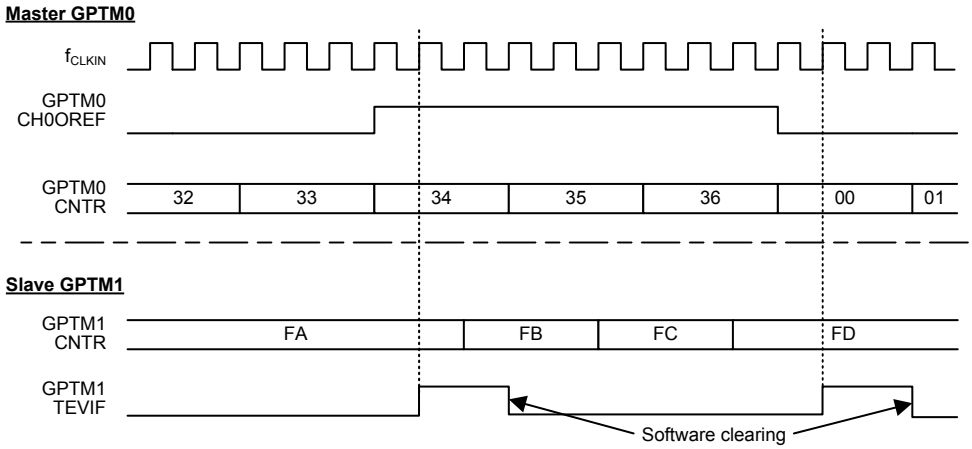


Figure 61. Pausing GPTM1 using the GPTM0 CH0OREF Signal

Using One Timer to Trigger Another Timer Start Counting

- Configure GPTM0 to operate in the master mode to send its Update Event UEV as the trigger output (MMSEL = 0x2).
- Configure the GPTM0 period by setting the CRR register.
- Configure GPTM1 to get the input trigger source from the GPTM0 trigger output (TRSEL = 0x9).
- Configure GPTM1 to be in the slave trigger mode (SMSEL = 0x6).
- Start GPTM0 by writing '1' to the TME bit.

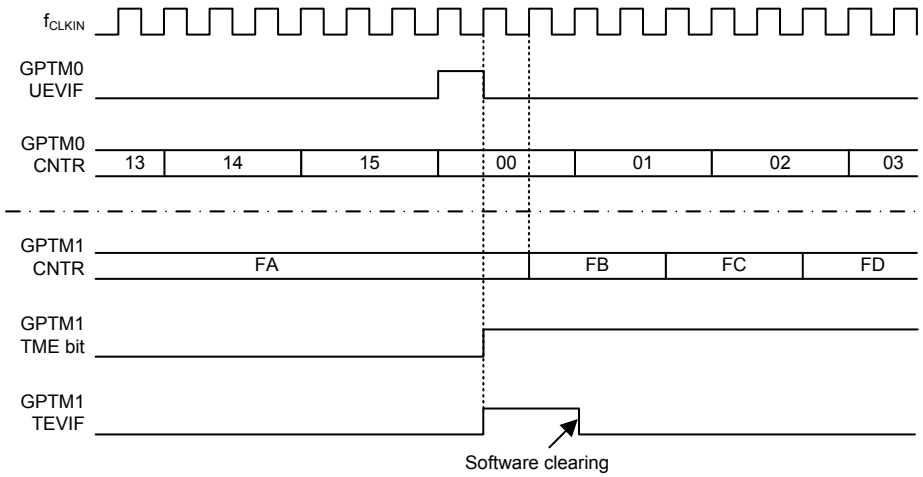


Figure 62. Triggering GPTM1 with GPTM0 Update Event

Starting Two Timers Synchronously in Response to an External Trigger

- Configure GPTM0 to operate in the master mode to send its enable signal as a trigger output (MMSEL = 0x1).
- Configure GPTM0 slave mode to receive its input trigger source from GTn_CH0 pin (TRSEL = 0x1).
- Configure GPTM0 to be in the slave trigger mode (SMSEL = 0x6).
- Enable the GPTM0 master timer synchronization function by setting the TSE bit in the MDCFR register to 1 to synchronize the slave timer.
- Configure GPTM1 to receive its input trigger source from the GPTM0 trigger output (TRSEL = 0x9).
- Configure GPTM1 to be in the slave trigger mode (SMSEL = 0x6).

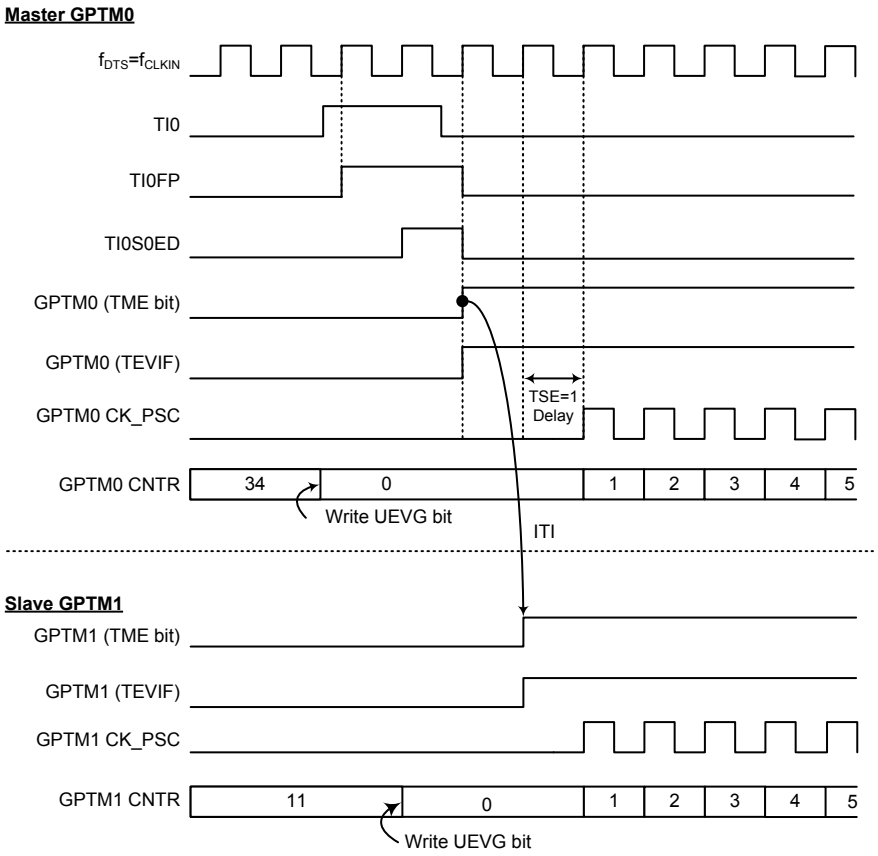


Figure 63. Trigger GPTM0 and GPTM1 with the GPTM0 CH0 Input

Trigger ADC Start

To interconnect to the Analog-to-Digital Converter, the GPTM can output the MTO signal or the channel compare match output signal CHxOREF (x = 0 ~ 3) to be used as an Analog-to-Digital Converter input trigger signal.

PDMA Request

The GPTM supports the interface for PDMA data transfer. There are certain events which can generate the PDMA requests if the corresponding enable control bits are set to 1 to enable the PDMA access. These events are the GPTM update events, trigger events and channel capture / compare events. When the PDMA request is generated from the GPTM channel, it can be derived from the channel capture / compare event or the GPTM update event selected by the channel PDMA selection bit, CHCCDS, for all channels. For more detailed PDMA configuring information, refer to the corresponding section in the PDMA chapter.

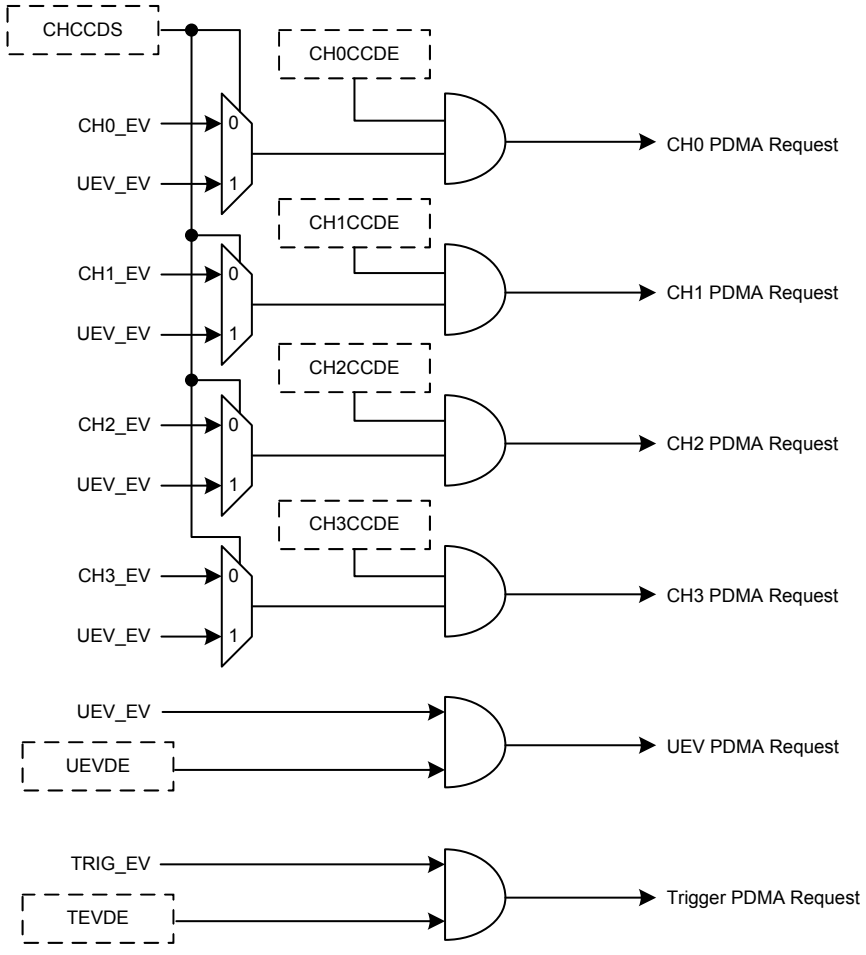


Figure 64. GPTM PDMA Mapping Diagram

Register Map

The following table shows the GPTM registers and reset values.

Table 31. GPTM Register Map

Register	Offset	Description	Reset Value
CNTCFR	0x000	Timer Counter Configuration Register	0x0000_0000
MDCFR	0x004	Timer Mode Configuration Register	0x0000_0000
TRCFR	0x008	Timer Trigger Configuration Register	0x0000_0000
CTR	0x010	Timer Control Register	0x0000_0000
CH0ICFR	0x020	Channel 0 Input Configuration Register	0x0000_0000
CH1ICFR	0x024	Channel 1 Input Configuration Register	0x0000_0000
CH2ICFR	0x028	Channel 2 Input Configuration Register	0x0000_0000
CH3ICFR	0x02C	Channel 3 Input Configuration Register	0x0000_0000
CH0OCFR	0x040	Channel 0 Output Configuration Register	0x0000_0000
CH1OCFR	0x044	Channel 1 Output Configuration Register	0x0000_0000
CH2OCFR	0x048	Channel 2 Output Configuration Register	0x0000_0000
CH3OCFR	0x04C	Channel 3 Output Configuration Register	0x0000_0000
CHCTR	0x050	Channel Control Register	0x0000_0000
CHPOLR	0x054	Channel Polarity Configuration Register	0x0000_0000
DICTR	0x074	Timer PDMA / Interrupt Control Register	0x0000_0000
EVGR	0x078	Timer Event Generator Register	0x0000_0000
INTSR	0x07C	Timer Interrupt Status Register	0x0000_0000
CNTR	0x080	Timer Counter Register	0x0000_0000
PSCR	0x084	Timer Prescaler Register	0x0000_0000
CRR	0x088	Timer Counter Reload Register	0x0000_FFFF
CH0CCR	0x090	Channel 0 Capture / Compare Register	0x0000_0000
CH1CCR	0x094	Channel 1 Capture / Compare Register	0x0000_0000
CH2CCR	0x098	Channel 2 Capture / Compare Register	0x0000_0000
CH3CCR	0x09C	Channel 3 Capture / Compare Register	0x0000_0000
CH0ACR	0x0A0	Channel 0 Asymmetric Compare Register	0x0000_0000
CH1ACR	0x0A4	Channel 1 Asymmetric Compare Register	0x0000_0000
CH2ACR	0x0A8	Channel 2 Asymmetric Compare Register	0x0000_0000
CH3ACR	0x0AC	Channel 3 Asymmetric Compare Register	0x0000_0000

Register Descriptions

Timer Counter Configuration Register – CNTCFR

This register specifies the GPTM counter configuration.

Offset: 0x000
Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved							DIR
Type/Reset								RW 0
	23	22	21	20	19	18	17	16
	Reserved							CMSEL
Type/Reset							RW 0	RW 0
	15	14	13	12	11	10	9	8
	Reserved							CKDIV
Type/Reset							RW 0	RW 0
	7	6	5	4	3	2	1	0
	Reserved						UGDIS	UEVDIS
Type/Reset							RW 0	RW 0

Bits	Field	Descriptions
[24]	DIR	Counting Direction 0: Count-up 1: Count-down Note: This bit is read only when the Timer is configured to be in the Center-aligned mode or when used as a Quadrature decoder.
[17:16]	CMSEL	Counter Mode Selection 00: Edge-aligned mode. Normal up-counting and down-counting available for this mode. Counting direction is defined by the DIR bit. 01: Center-aligned mode 1. The counter counts up and down alternatively. The compare match interrupt flag is set during the count-down period. 10: Center-aligned mode 2. The counter counts up and down alternatively. The compare match interrupt flag is set during the count-up period. 11: Center-aligned mode 3. The counter counts up and down alternatively. The compare match interrupt flag is set during the count-up and count-down period.
[9:8]	CKDIV	Clock Division These two bits define the frequency ratio between the timer clock (f_{CLKIN}) and the dead-time clock (f_{DTS}). The dead-time clock is also used for digital filter sampling clock. 00: $f_{DTS} = f_{CLKIN}$ 01: $f_{DTS} = f_{CLKIN} / 2$ 10: $f_{DTS} = f_{CLKIN} / 4$ 11: Reserved
[1]	UGDIS	Update event interrupt generation disable control 0: Any of the following events will generate an update PDMA request or interrupt - Counter overflow / underflow - Setting the UEVG bit - Update generation through the slave mode 1: Only counter overflow / underflow generates an update PDMA request or interrupt

Bits	Field	Descriptions
[0]	UEVDIS	Update event Disable control 0: Enable the update event request by one of following events: - Counter overflow / underflow - Setting the UEVG bit - Update generation through the slave mode 1: Disable the update event (However the counter and the prescaler are reinitialized if the UEVG bit is set or if a hardware restart is received from the slave mode)

Timer Mode Configuration Register – MDCFR

This register specifies the GPTM master and slave mode selection and single pulse mode.

Offset: 0x004

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved							SPMSET
Type/Reset								RW 0
	23	22	21	20	19	18	17	16
	Reserved					MMSEL		
Type/Reset						RW 0	RW 0	RW 0
	15	14	13	12	11	10	9	8
	Reserved					SMSEL		
Type/Reset						RW 0	RW 0	RW 0
	7	6	5	4	3	2	1	0
	Reserved							TSE
Type/Reset								RW 0

Bits	Field	Descriptions
[24]	SPMSET	Single Pulse Mode Setting 0: Counter counts normally irrespective of whether the update event occurred or not 1: Counter stops counting at the next update event and then the TME bit is cleared by hardware

Bits	Field	Descriptions
[18:16]	MMSEL	Master Mode Selection Master mode selection is used to select the MTO signal source which is used to synchronize the other slave timer.

MMSEL [2:0]	Mode	Descriptions
000	Reset Mode	The MTO in the Reset mode is an output derived from one of the following cases: 1. Software setting UEVG bit 2. The STI trigger input signal which will be output on the MTO signal line when the Timer is used in the slave Restart mode
001	Enable Mode	The Counter Enable signal is used as the trigger output.
010	Update Mode	The update event is used as the trigger output according to one of the following cases when the UEVDIS bit is cleared to 0: 1. Counter overflow / underflow 2. Software setting UEVG 3. Slave trigger input when used in slave restart mode
011	Capture / Compare Mode	When a Channel 0 capture or compare match event occurs, it will generate a positive pulse used as the master trigger output.
100	Compare Mode 0	The Channel 0 Output reference signal named CH0OREF is used as the trigger output.
101	Compare Mode 1	The Channel 1 Output reference signal named CH1OREF is used as the trigger output.
110	Compare Mode 2	The Channel 2 Output reference signal named CH2OREF is used as the trigger output.
111	Compare Mode 3	The Channel 3 Output reference signal named CH3OREF is used as the trigger output.

Bits	Field	Descriptions
[10:8]	SMSEL	Slave Mode Selection

SMSEL [2:0]	Mode	Descriptions
000	Disable mode	The prescaler is clocked directly by the internal clock.
001	Quadrature Decoder mode 1	The counter uses the clock pulse generated from the interaction between the TI0 and TI1 signals to drive the counter prescaler. A transition of the TI0 edge is used in this mode depending upon the TI1 level.
010	Quadrature Decoder mode 2	The counter uses the clock pulse generated from the interaction between the TI0 and TI1 signals to drive the counter prescaler. A transition of the TI1 edge is used in this mode depending upon the TI0 level.
011	Quadrature Decoder mode 3	The counter uses the clock pulse generated from the interaction between the TI0 and TI1 signals to drive the counter prescaler. A transition of one channel edge is used in the quadrature decoder mode 3 depending upon the other channel level.
100	Restart Mode	The counter value restarts from 0 or the CRR shadow register value depending upon the counter mode on the rising edge of the STI signal. The registers will also be updated.
101	Pause Mode	The counter starts to count when the selected trigger input STI is high. The counter stops counting on the instant, not being reset, when the STI signal changes its state to a low level. Both the counter start and stop control are determined by the STI signal.
110	Trigger Mode	The counter starts to count from the original value in the counter on the rising edge of the selected trigger input STI. Only the counter start control is determined by the STI signal.
111	STIED	The rising edge of the selected trigger signal STI will clock the counter.

[0]	TSE	Timer Synchronization Enable 0: No action 1: Master timer (current timer) will generate a delay to synchronize its slave timer through the MTO signal.

Timer Trigger Configuration Register – TRCFR

This register specifies the trigger source selection of GPTM.

Offset: 0x008

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	22	21	20	19	18	17	16
	Reserved							
Type/Reset								
	15	14	13	12	11	10	9	8
	Reserved							
Type/Reset								
	7	6	5	4	3	2	1	0
	Reserved				TRSEL			
Type/Reset					RW 0	RW 0	RW 0	RW 0

Bits	Field	Descriptions
[3:0]	TRSEL	Trigger Source Selection These bits are used to select the trigger input (STI) for counter synchronization. 0000: Software Trigger by setting the UEVG bit 0001: Filtered input of channel 0 (TI0S0) 0010: Filtered input of channel 1 (TI1S1) 0011: Reserved 1000: Channel 0 Edge Detector (TI0BED) 1001: Internal Timing Module Trigger 0 (ITI0) 1010: Internal Timing Module Trigger 1 (ITI1) 1011: Internal Timing Module Trigger 2 (ITI2) Others: Reserved Note: These bits must be updated only when they are not in use, i.e. the slave mode is disabled by setting the SMSEL field to 0x00.

Table 32. GPTM Internal Trigger Connection

Slave Timing Module	ITI0	ITI1	ITI2
GPTM0	GPTM1	MCTM0	MCTM1
GPTM1	GPTM0	MCTM0	MCTM1

Timer Counter Register – CTR

This register specifies the timer enable bit (TME), CRR buffer enable bit (CRBE) and Channel PDMA selection bit (CHCCDS).

Offset: 0x010

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	22	21	20	19	18	17	16
	Reserved							CHCCDS
Type/Reset								RW 0
	15	14	13	12	11	10	9	8
	Reserved							
Type/Reset								
	7	6	5	4	3	2	1	0
	Reserved						CRBE	TME
Type/Reset							RW 0	RW 0

Bits	Field	Descriptions
[16]	CHCCDS	Channel PDMA event selection 0: Channel PDMA request derived from the channel capture / compare event. 1: Channel PDMA request derived from the Update event.
[1]	CRBE	Counter Reload register Buffer Enable 0: Counter reload register can be updated immediately 1: Counter reload register can not be updated until the update event occurs
[0]	TME	Timer Enable bit 0: GPTM off 1: GPTM on – GPTM functions normally When the TME bit is cleared to 0, the counter is stopped and the GPTM consumes no power in any operation mode except for the single pulse mode and the slave trigger mode. In these two modes the TME bit can automatically be set to 1 by hardware which permits all the GPTM registers to function normally.

Channel 0 Input Configuration Register – CH0ICFR

This register specifies the channel 0 input mode configuration.

Offset: 0x020

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	TI0SRC	Reserved						
Type/Reset	RW 0							
	23	22	21	20	19	18	17	16
	Reserved				CH0PSC		CH0CCS	
Type/Reset					RW 0	RW 0	RW 0	RW 0
	15	14	13	12	11	10	9	8
	Reserved							
Type/Reset								
	7	6	5	4	3	2	1	0
	Reserved				TI0F			
Type/Reset					RW 0	RW 0	RW 0	RW 0

Bits	Field	Descriptions
[31]	TI0SRC	Channel 0 Input Source TI0 Selection 0: The GT_CH0 pin is connected to channel 0 input TI0 1: The XOR operation output of the GT_CH0, GT_CH1 and GT_CH2 pins are connected to the channel 0 input TI0
[19:18]	CH0PSC	Channel 0 Capture Input Source Prescaler Setting These bits define the effective events of the channel 0 capture input. Note that the prescaler is reset once the Channel 0 Capture / Compare Enable bit, CH0E, in the Channel Control register named CHCTR is cleared to 0. 00: No prescaler, channel 0 capture input signal is chosen for each active event 01: Channel 0 Capture input signal is chosen for every 2 events 10: Channel 0 Capture input signal is chosen for every 4 events 11: Channel 0 Capture input signal is chosen for every 8 events
[17:16]	CH0CCS	Channel 0 Capture / Compare Selection 00: Channel 0 is configured as an output 01: Channel 0 is configured as an input derived from the TI0 signal 10: Channel 0 is configured as an input derived from the TI1 signal 11: Channel 0 is configured as an input which comes from the TRCED signal derived from the Trigger Controller Note: The CH0CCS field can be accessed only when the CH0E bit is cleared to 0.

Bits	Field	Descriptions
[3:0]	TI0F	Channel 0 Input Source TI0 Filter Setting These bits define the frequency divided ratio used to sample the TI0 signal. The Digital filter in the GPTM is an N-event counter where N is defined as how many valid transitions are necessary to output a filtered signal. 0000: No filter, the sampling clock is f_{SYSTEM} 0001: $f_{SAMPLING} = f_{CLKIN}$, N = 2 0010: $f_{SAMPLING} = f_{CLKIN}$, N = 4 0011: $f_{SAMPLING} = f_{CLKIN}$, N = 8 0100: $f_{SAMPLING} = f_{DTS} / 2$, N = 6 0101: $f_{SAMPLING} = f_{DTS} / 2$, N = 8 0110: $f_{SAMPLING} = f_{DTS} / 4$, N = 6 0111: $f_{SAMPLING} = f_{DTS} / 4$, N = 8 1000: $f_{SAMPLING} = f_{DTS} / 8$, N = 6 1001: $f_{SAMPLING} = f_{DTS} / 8$, N = 8 1010: $f_{SAMPLING} = f_{DTS} / 16$, N = 5 1011: $f_{SAMPLING} = f_{DTS} / 16$, N = 6 1100: $f_{SAMPLING} = f_{DTS} / 16$, N = 8 1101: $f_{SAMPLING} = f_{DTS} / 32$, N = 5 1110: $f_{SAMPLING} = f_{DTS} / 32$, N = 6 1111: $f_{SAMPLING} = f_{DTS} / 32$, N = 8

Channel 1 Input Configuration Register – CH1ICFR

This register specifies the channel 1 input mode configuration.

Offset: 0x024

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	22	21	20	19	18	17	16
	Reserved				CH1PSC		CH1CCS	
Type/Reset					RW 0	RW 0	RW 0	RW 0
	15	14	13	12	11	10	9	8
	Reserved							
Type/Reset								
	7	6	5	4	3	2	1	0
	Reserved				TI1F			
Type/Reset					RW 0	RW 0	RW 0	RW 0

Bits	Field	Descriptions
[19:18]	CH1PSC	Channel 1 Capture Input Source Prescaler Setting These bits define the effective events of the channel 1 capture input. Note that the prescaler is reset once the Channel 1 Capture / Compare Enable bit, CH1E, in the Channel Control register named CHCTR is cleared to 0. 00: No prescaler, channel 1 capture input signal is chosen for each active event 01: Channel 1 Capture input signal is chosen for every 2 events 10: Channel 1 Capture input signal is chosen for every 4 events 11: Channel 1 Capture input signal is chosen for every 8 events

Bits	Field	Descriptions
[17:16]	CH1CCS	Channel 1 Capture / Compare Selection 00: Channel 1 is configured as an output 01: Channel 1 is configured as an input derived from the TI1 signal 10: Channel 1 is configured as an input derived from the TI0 signal 11: Channel 1 is configured as an input which comes from the TRCED signal derived from the Trigger Controller Note: The CH1CCS field can be accessed only when the CH1E bit is cleared to 0.
[3:0]	TI1F	Channel 1 Input Source TI1 Filter Setting These bits define the frequency divided ratio used to sample the TI1 signal. The Digital filter in the GPTM is an N-event counter where N is defined as how many valid transitions are necessary to output a filtered signal. 0000: No filter, the sampling clock is f_{SYSTEM} 0001: $f_{SAMPLING} = f_{CLKIN}$, N = 2 0010: $f_{SAMPLING} = f_{CLKIN}$, N = 4 0011: $f_{SAMPLING} = f_{CLKIN}$, N = 8 0100: $f_{SAMPLING} = f_{DTS}$ / 2, N = 6 0101: $f_{SAMPLING} = f_{DTS}$ / 2, N = 8 0110: $f_{SAMPLING} = f_{DTS}$ / 4, N = 6 0111: $f_{SAMPLING} = f_{DTS}$ / 4, N = 8 1000: $f_{SAMPLING} = f_{DTS}$ / 8, N = 6 1001: $f_{SAMPLING} = f_{DTS}$ / 8, N = 8 1010: $f_{SAMPLING} = f_{DTS}$ / 16, N = 5 1011: $f_{SAMPLING} = f_{DTS}$ / 16, N = 6 1100: $f_{SAMPLING} = f_{DTS}$ / 16, N = 8 1101: $f_{SAMPLING} = f_{DTS}$ / 32, N = 5 1110: $f_{SAMPLING} = f_{DTS}$ / 32, N = 6 1111: $f_{SAMPLING} = f_{DTS}$ / 32, N = 8

Channel 2 Input Configuration Register – CH2ICFR

This register specifies the channel 2 input mode configuration.

Offset: 0x028

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	22	21	20	19	18	17	16
	Reserved				CH2PSC		CH2CCS	
Type/Reset					RW 0	RW 0	RW 0	RW 0
	15	14	13	12	11	10	9	8
	Reserved							
Type/Reset								
	7	6	5	4	3	2	1	0
	Reserved				TI2F			
Type/Reset					RW 0	RW 0	RW 0	RW 0

Bits	Field	Descriptions
[19:18]	CH2PSC	Channel 2 Capture Input Source Prescaler Setting These bits define the effective events of the channel 2 capture input. Note that the prescaler is reset once the Channel 2 Capture / Compare Enable bit, CH2E, in the Channel Control register named CHCTR is cleared to 0. 00: No prescaler, channel 2 capture input signal is chosen for each active event 01: Channel 2 Capture input signal is chosen for every 2 events 10: Channel 2 Capture input signal is chosen for every 4 events 11: Channel 2 Capture input signal is chosen for every 8 events
[17:16]	CH2CCS	Channel 2 Capture / Compare Selection 00: Channel 2 is configured as an output 01: Channel 2 is configured as an input derived from the TI2 signal 10: Channel 2 is configured as an input derived from the TI3 signal 11: Channel 2 is configured as an input which comes from the TRCED signal derived from the Trigger Controller Note: The CH2CCS field can be accessed only when the CH2E bit is cleared to 0

Bits	Field	Descriptions
[3:0]	TI2F	Channel 2 Input Source TI2 Filter Setting These bits define the frequency divided ratio used to sample the TI2 signal. The Digital filter in the GPTM is an N-event counter where N is defined as how many valid transitions are necessary to output a filtered signal. 0000: No filter, the sampling clock is f_{SYSTEM} 0001: $f_{SAMPLING} = f_{CLKIN}$, N = 2 0010: $f_{SAMPLING} = f_{CLKIN}$, N = 4 0011: $f_{SAMPLING} = f_{CLKIN}$, N = 8 0100: $f_{SAMPLING} = f_{DTS}$ / 2, N = 6 0101: $f_{SAMPLING} = f_{DTS}$ / 2, N = 8 0110: $f_{SAMPLING} = f_{DTS}$ / 4, N = 6 0111: $f_{SAMPLING} = f_{DTS}$ / 4, N = 8 1000: $f_{SAMPLING} = f_{DTS}$ / 8, N = 6 1001: $f_{SAMPLING} = f_{DTS}$ / 8, N = 8 1010: $f_{SAMPLING} = f_{DTS}$ / 16, N = 5 1011: $f_{SAMPLING} = f_{DTS}$ / 16, N = 6 1100: $f_{SAMPLING} = f_{DTS}$ / 16, N = 8 1101: $f_{SAMPLING} = f_{DTS}$ / 32, N = 5 1110: $f_{SAMPLING} = f_{DTS}$ / 32, N = 6 1111: $f_{SAMPLING} = f_{DTS}$ / 32, N = 8

Channel 3 Input Configuration Register – CH3ICFR

This register specifies the channel 3 input mode configuration.

Offset: 0x02C

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	22	21	20	19	18	17	16
	Reserved				CH3PSC		CH3CCS	
Type/Reset					RW 0	RW 0	RW 0	RW 0
	15	14	13	12	11	10	9	8
	Reserved							
Type/Reset								
	7	6	5	4	3	2	1	0
	Reserved				TI3F			
Type/Reset					RW 0	RW 0	RW 0	RW 0

Bits	Field	Descriptions
[19:18]	CH3PSC	Channel 3 Capture Input Source Prescaler Setting These bits define the effective events of the channel 3 capture input. Note that the prescaler is reset once the Channel 3 Capture / Compare Enable bit, CH3E, in the Channel Control register named CHCTR is cleared to 0. 00: No prescaler, channel 3 capture input signal is chosen for each active event 01: Channel 3 Capture input signal is chosen for every 2 events 10: Channel 3 Capture input signal is chosen for every 4 events 11: Channel 3 Capture input signal is chosen for every 8 events

Bits	Field	Descriptions
[17:16]	CH3CCS	Channel 3 Capture / Compare Selection 00: Channel 3 is configured as an output 01: Channel 3 is configured as an input derived from the TI3 signal 10: Channel 3 is configured as an input derived from the TI2 signal 11: Channel 3 is configured as an input which comes from the TRCED signal derived from the Trigger Controller Note: The CH3CCS field can be accessed only when the CH3E bit is cleared to 0
[3:0]	TI3F	Channel 3 Input Source TI3 Filter Setting These bits define the frequency divided ratio used to sample the TI3 signal. The Digital filter in the GPTM is an N-event counter where N is defined as how many valid transitions are necessary to output a filtered signal. 0000: No filter, the sampling clock is f_{SYSTEM} 0001: $f_{SAMPLING} = f_{CLKIN}$, N = 2 0010: $f_{SAMPLING} = f_{CLKIN}$, N = 4 0011: $f_{SAMPLING} = f_{CLKIN}$, N = 8 0100: $f_{SAMPLING} = f_{DTS}$ / 2, N = 6 0101: $f_{SAMPLING} = f_{DTS}$ / 2, N = 8 0110: $f_{SAMPLING} = f_{DTS}$ / 4, N = 6 0111: $f_{SAMPLING} = f_{DTS}$ / 4, N = 8 1000: $f_{SAMPLING} = f_{DTS}$ / 8, N = 6 1001: $f_{SAMPLING} = f_{DTS}$ / 8, N = 8 1010: $f_{SAMPLING} = f_{DTS}$ / 16, N = 5 1011: $f_{SAMPLING} = f_{DTS}$ / 16, N = 6 1100: $f_{SAMPLING} = f_{DTS}$ / 16, N = 8 1101: $f_{SAMPLING} = f_{DTS}$ / 32, N = 5 1110: $f_{SAMPLING} = f_{DTS}$ / 32, N = 6 1111: $f_{SAMPLING} = f_{DTS}$ / 32, N = 8

Channel 0 Output Configuration Register – CH0OCFR

This register specifies the channel 0 output mode configuration.

Offset: 0x040
Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	22	21	20	19	18	17	16
	Reserved							
Type/Reset								
	15	14	13	12	11	10	9	8
	Reserved							CH0OM[3]
Type/Reset								RW 0
	7	6	5	4	3	2	1	0
		Reserved	CH0IMAE	CH0PRE	Reserved	CH0OM[2:0]		
Type/Reset			RW 0	RW 0		RW 0	RW 0	RW 0

Bits	Field	Descriptions
[5]	CH0IMAE	Channel 0 Immediate Active Enable 0: No action 1: Single pulse Immediate Active Mode is enabled The CH0OREF will be forced to the compare matched level immediately after an available trigger event occurs irrespective of the result of the comparison between the CNTR and the CH0CCR values. The effective duration ends automatically at the next overflow or underflow event. Note: The CH0IMAE bit is available only if the channel 0 is configured to be operated in the PWM mode 1 or the PWM mode 2.
[4]	CH0PRE	Channel 0 Capture / Compare Register (CH0CCR) Preload Enable 0: CH0CCR preload function is disabled. The CH0CCR register can be immediately assigned a new value when the CH0PRE bit is cleared to 0 and the updated CH0CCR value is used immediately. 1: CH0CCR preload function is enabled. The new CH0CCR value will not be transferred to its shadow register until the update event occurs.
[8][2:0]	CH0OM[3:0]	Channel 0 Output Mode Setting These bits define the functional types of the output reference signal CH0OREF. 0000: No Change 0001: Output 0 on compare match 0010: Output 1 on compare match 0011: Output toggles on compare match 0100: Force inactive – CH0OREF is forced to 0 0101: Force active – CH0OREF is forced to 1 0110: PWM mode 1 - During up-counting, channel 0 has an active level when CNTR < CH0CCR or otherwise has an inactive level. - During down-counting, channel 0 has an inactive level when CNTR > CH0CCR or otherwise has an active level. 0111: PWM mode 2 - During up-counting, channel 0 is has an inactive level when CNTR < CH0CCR or otherwise has an active level. - During down-counting, channel 0 has an active level when CNTR > CH0CCR or otherwise has an inactive level. 1110: Asymmetric PWM mode 1 - During up-counting, channel 0 has an active level when CNTR < CH0CCR or otherwise has an inactive level. - During down-counting, channel 0 has an inactive level when CNTR > CH0ACR or otherwise has an active level. 1111: Asymmetric PWM mode 2 - During up-counting, channel 0 has an inactive level when CNTR < CH0CCR or otherwise has an active level. - During down-counting, channel 0 has an active level when CNTR > CH0ACR or otherwise has an inactive level Note: When channel 0 is used as asymmetric PWM output mode, the Counter Mode Selection bit in Counter Configuration Register must be configured as Center-aligned Counting mode (CMSEL = 0x1 / 0x2 / 0x3)

Channel 1 Output Configuration Register – CH1OCFR

This register specifies the channel 1 output mode configuration.

Offset: 0x044

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	22	21	20	19	18	17	16
	Reserved							
Type/Reset								
	15	14	13	12	11	10	9	8
	Reserved							CH1OM[3]
Type/Reset								RW 0
	7	6	5	4	3	2	1	0
		Reserved	CH1IMAE	CH1PRE	Reserved	CH1OM[2:0]		
Type/Reset			RW 0	RW 0		RW 0	RW 0	RW 0

Bits	Field	Descriptions
[5]	CH1IMAE	Channel 1 Immediate Active Enable 0: No action 1: Single pulse Immediate Active Mode is enabled The CH1OREF will be forced to the compare matched level immediately after an available trigger event occurs irrespective of the result of the comparison between the CNTR and the CH1CCR values. The effective duration ends automatically at the next overflow or underflow event. Note: The CH1IMAE bit is available only if the channel 1 is configured to be operated in the PWM mode 1 or the PWM mode 2.
[4]	CH1PRE	Channel 1 Capture / Compare Register (CH1CCR) Preload Enable 0: CH1CCR preload function is disabled. The CH1CCR register can be immediately assigned a new value when the CH1PRE bit is cleared to 0 and the updated CH1CCR value is used immediately. 1: CH1CCR preload function is enabled The new CH1CCR value will not be transferred to its shadow register until the update event occurs.

Bits	Field	Descriptions
[8][2:0]	CH1OM[3:0]	Channel 1 Output Mode Setting These bits define the functional types of the output reference signal CH1OREF. 0000: No Change 0001: Output 0 on compare match 0010: Output 1 on compare match 0011: Output toggles on compare match 0100: Force inactive – CH1OREF is forced to 0 0101: Force active – CH1OREF is forced to 1 0110: PWM mode 1 - During up-counting, channel 1 has an active level when CNTR < CH1CCR or otherwise has an inactive level. - During down-counting, channel 1 has an inactive level when CNTR > CH1CCR or otherwise has an active level. 0111: PWM mode 2 - During up-counting, channel 1 has an inactive level when CNTR < CH1CCR or otherwise has an active level. - During down-counting, channel 1 has an active level when CNTR > CH1CCR or otherwise has an inactive level. 1110: Asymmetric PWM mode 1 - During up-counting, channel 1 has an active level when CNTR < CH1CCR or otherwise has an inactive level. - During down-counting, channel 1 has an inactive level when CNTR > CH1ACR or otherwise has an active level. 1111: Asymmetric PWM mode 2 - During up-counting, channel 1 has an inactive level when CNTR < CH1CCR or otherwise has an active level. - During down-counting, channel 1 has an active level when CNTR > CH1ACR or otherwise has an inactive level Note: When channel 1 is used as asymmetric PWM output mode, the Counter Mode Selection bit in Counter Configuration Register must be configured as Center-aligned Counting mode (CMSEL = 0x1 / 0x2 / 0x3)

Channel 2 Output Configuration Register – CH2OCFR

This register specifies the channel 2 output mode configuration.

Offset: 0x048

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	22	21	20	19	18	17	16
	Reserved							
Type/Reset								
	15	14	13	12	11	10	9	8
	Reserved							CH2OM[3]
Type/Reset								RW 0
	7	6	5	4	3	2	1	0
		Reserved	CH2IMAE	CH2PRE	Reserved	CH2OM[2:0]		
Type/Reset			RW 0	RW 0		RW 0	RW 0	RW 0

Bits	Field	Descriptions
[5]	CH2IMAE	Channel 2 Immediate Active Enable 0: No action 1: Single pulse Immediate Active Mode is enabled The CH2OREF will be forced to the compare matched level immediately after an available trigger event occurs irrespective of the result of the comparison between the CNTR and the CH2CCR values. The effective duration ends automatically at the next overflow or underflow event. Note: The CH2IMAE bit is available only if the channel 2 is configured to be operated in the PWM mode 1 or the PWM mode 2.
[4]	CH2PRE	Channel 2 Capture / Compare Register (CH2CCR) Preload Enable 0: CH2CCR preload function is disabled. The CH2CCR register can be immediately assigned a new value when the CH2PRE bit is cleared to 0 and the updated CH2CCR value is used immediately. 1: CH2CCR preload function is enabled The new CH2CCR value will not be transferred to its shadow register until the update event occurs.

Bits	Field	Descriptions
[8][2:0]	CH2OM[3:0]	Channel 2 Output Mode Setting These bits define the functional types of the output reference signal CH2OREF. 0000: No Change 0001: Output 0 on compare match 0010: Output 1 on compare match 0011: Output toggles on compare match 0100: Force inactive – CH2OREF is forced to 0 0101: Force active – CH2OREF is forced to 1 0110: PWM mode 1 - During up-counting, channel 2 has an active level when CNTR < CH2CCR or otherwise has an inactive level. - During down-counting, channel 2 has an inactive level when CNTR > CH2CCR or otherwise has an active level. 0111: PWM mode 2 - During up-counting, channel 2 has an inactive level when CNTR < CH2CCR or otherwise has an active level. - During down-counting, channel 2 has an active level when CNTR > CH2CCR or otherwise has an inactive level. 1110: Asymmetric PWM mode 1 - During up-counting, channel 2 has an active level when CNTR < CH2CCR or otherwise has an inactive level. - During down-counting, channel 2 has an inactive level when CNTR > CH2ACR or otherwise has an active level. 1111: Asymmetric PWM mode 2 - During up-counting, channel 2 has an inactive level when CNTR < CH2CCR or otherwise has an active level. - During down-counting, channel 2 has an active level when CNTR > CH2ACR or otherwise has an inactive level Note: When channel 2 is used as asymmetric PWM output mode, the Counter Mode Selection bit in Counter Configuration Register must be configured as Center-aligned Counting mode (CMSEL = 0x1 / 0x2 / 0x3)

Channel 3 Output Configuration Register – CH3OCFR

This register specifies the channel 3 output mode configuration.

Offset: 0x04C

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	22	21	20	19	18	17	16
	Reserved							
Type/Reset								
	15	14	13	12	11	10	9	8
	Reserved							CH3OM[3]
Type/Reset								RW 0
	7	6	5	4	3	2	1	0
		Reserved	CH3IMAE	CH3PRE	Reserved	CH3OM[2:0]		
Type/Reset			RW 0	RW 0		RW 0	RW 0	RW 0

Bits	Field	Descriptions
[5]	CH3IMAE	Channel 3 Immediate Active Enable 0: No action 1: Single pulse Immediate Active Mode is enabled The CH3OREF will be forced to the compare matched level immediately after an available trigger event occurs irrespective of the result of the comparison between the CNTR and the CH3CCR values. The effective duration ends automatically at the next overflow or underflow event. Note: The CH3IMAE bit is available only if the channel 3 is configured to be operated in the PWM mode 1 or the PWM mode 2.
[4]	CH3PRE	Channel 3 Capture / Compare Register (CH3CCR) Preload Enable 0: CH3CCR preload function is disabled. The CH3CCR register can be immediately assigned a new value when the CH3PRE bit is cleared to 0 and the updated CH3CCR value is used immediately. 1: CH3CCR preload function is enabled The new CH3CCR value will not be transferred to its shadow register until the update event occurs.

Bits	Field	Descriptions
[8][2:0]	CH3OM[3:0]	Channel 3 Output Mode Setting These bits define the functional types of the output reference signal CH3OREF 0000: No Change 0001: Output 0 on compare match 0010: Output 1 on compare match 0011: Output toggles on compare match 0100: Force inactive – CH3OREF is forced to 0 0101: Force active – CH3OREF is forced to 1 0110: PWM mode 1 - During up-counting, channel 3 has an active level when CNTR < CH3CCR or otherwise has an inactive level. - During down-counting, channel 3 has an inactive level when CNTR > CH3CCR or otherwise has an active level. 0111: PWM mode 2 - During up-counting, channel 3 has an inactive level when CNTR < CH3CCR or otherwise has an active level. - During down-counting, channel 3 has an active level when CNTR > CH3CCR or otherwise has an inactive level 1110: Asymmetric PWM mode 1 - During up-counting, channel 3 has an active level when CNTR < CH3CCR or otherwise has an inactive level. - During down-counting, channel 3 has an inactive level when CNTR > CH3ACR or otherwise has an active level. 1111: Asymmetric PWM mode 2 - During up-counting, channel 3 has an inactive level when CNTR < CH3CCR or otherwise has an active level. - During down-counting, channel 3 has an active level when CNTR > CH3ACR or otherwise has an inactive level Note: When channel 3 is used as asymmetric PWM output mode, the Counter Mode Selection bit in Counter Configuration Register must be configured as Center-aligned Counting mode (CMSEL = 0x1 / 0x2 / 0x3)

Channel Control Register – CHCTR

This register contains the channel capture input or compare output function enable control bits.

Offset: 0x050

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	22	21	20	19	18	17	16
	Reserved							
Type/Reset								
	15	14	13	12	11	10	9	8
	Reserved							
Type/Reset								

	7	6	5	4	3	2	1	0
	Reserved	CH3E	Reserved	CH2E	Reserved	CH1E	Reserved	CH0E
Type/Reset		RW 0		RW 0		RW 0		RW 0

Bits	Field	Descriptions
[6]	CH3E	Channel 3 Capture / Compare Enable - Channel 3 is configured as an input (CH3CCS = 0x1 / 0x2 / 0x3) 0: Input Capture Mode disabled 1: Input Capture Mode enabled - Channel 3 is configured as an output (CH3CCS = 0x0) 0: Off – Channel 3 output signal CH3O is not active 1: On – Channel 3 output signal CH3O generated on the corresponding output pin
[4]	CH2E	Channel 2 Capture / Compare Enable - Channel 2 is configured as an input (CH2CCS = 0x1 / 0x2 / 0x3) 0: Input Capture Mode disabled 1: Input Capture Mode enabled - Channel 2 is configured as an output (CH2CCS = 0x0) 0: Off – Channel 2 output signal CH2O is not active 1: On – Channel 2 output signal CH2O generated on the corresponding output pin
[2]	CH1E	Channel 1 Capture / Compare Enable - Channel 1 is configured as an input (CH1CCS = 0x1 / 0x2 / 0x3) 0: Input Capture Mode disabled 1: Input Capture Mode enabled - Channel 1 is configured as an output (CH1CCS = 0x0) 0: Off – Channel 1 output signal CH1O is not active 1: On – Channel 1 output signal CH1O generated on the corresponding output pin
[0]	CH0E	Channel 0 Capture / Compare Enable - Channel 0 is configured as an input (CH0CCS = 0x1 / 0x2 / 0x3) 0: Input Capture Mode disabled 1: Input Capture Mode enabled - Channel 0 is configured as an output (CH0CCS = 0x0) 0: Off – Channel 0 output signal CH0O is not active 1: On – Channel 0 output signal CH0O generated on the corresponding output pin

Channel Polarity Configuration Register – CHPOLR

This register contains the channel capture input or compare output polarity control.

Offset: 0x054

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	22	21	20	19	18	17	16
	Reserved							
Type/Reset								
	15	14	13	12	11	10	9	8
	Reserved							
Type/Reset								

	7	6	5	4	3	2	1	0
	Reserved	CH3P	Reserved	CH2P	Reserved	CH1P	Reserved	CH0P
Type/Reset		RW 0		RW 0		RW 0		RW 0

Bits	Field	Descriptions
[6]	CH3P	Channel 3 Capture / Compare Polarity - When Channel 3 is configured as an input (CH3CCS = 0x1 / 0x2 / 0x3) 0: Capture event occurs on a Channel 3 rising edge 1: Capture event occurs on a Channel 3 falling edge - When Channel 3 is configured as an output (CH3CCS = 0x0) 0: Channel 3 Output active high 1: Channel 3 Output active low
[4]	CH2P	Channel 2 Capture / Compare Polarity - When Channel 2 is configured as an input (CH2CCS = 0x1 / 0x2 / 0x3) 0: Capture event occurs on a Channel 2 rising edge 1: Capture event occurs on a Channel 2 falling edge - When Channel 2 is configured as an output (CH2CCS = 0x0) 0: Channel 2 Output active high 1: Channel 2 Output active low
[2]	CH1P	Channel 1 Capture / Compare Polarity - When Channel 1 is configured as an input (CH1CCS = 0x1 / 0x2 / 0x3) 0: Capture event occurs on a Channel 1 rising edge 1: Capture event occurs on a Channel 1 falling edge - When Channel 1 is configured as an output (CH1CCS = 0x0) 0: Channel 1 Output active high 1: Channel 1 Output active low
[0]	CH0P	Channel 0 Capture / Compare Polarity - When Channel 0 is configured as an input (CH0CCS = 0x1 / 0x2 / 0x3) 0: Capture event occurs on a Channel 0 rising edge 1: Capture event occurs on a Channel 0 falling edge - When Channel 0 is configured as an output (CH0CCS = 0x0) 0: Channel 0 Output active high 1: Channel 0 Output active low

Timer PDMA / Interrupt Control Register – DICTR

This register contains the timer PDMA and interrupt enable control bits.

Offset: 0x074

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved					TEVDE	Reserved	UEVDE
Type/Reset						RW 0		RW 0
	23	22	21	20	19	18	17	16
	Reserved				CH3CCDE	CH2CCDE	CH1CCDE	CH0CCDE
Type/Reset					RW 0	RW 0	RW 0	RW 0
	15	14	13	12	11	10	9	8
	Reserved					TEVIE	Reserved	UEVIE
Type/Reset						RW 0		RW 0
	7	6	5	4	3	2	1	0
	Reserved				CH3CCIE	CH2CCIE	CH1CCIE	CH0CCIE
Type/Reset					RW 0	RW 0	RW 0	RW 0

Bits	Field	Descriptions
[26]	TEVDE	Trigger event PDMA Request Enable 0: Trigger PDMA request disabled 1: Trigger PDMA request enabled
[24]	UEVDE	Update event PDMA Request Enable 0: Update event PDMA request disabled 1: Update event PDMA request enabled
[19]	CH3CCDE	Channel 3 Capture / Compare PDMA Request Enable 0: Channel 3 PDMA request disabled 1: Channel 3 PDMA request enabled
[18]	CH2CCDE	Channel 2 Capture / Compare PDMA Request Enable 0: Channel 2 PDMA request disabled 1: Channel 2 PDMA request enabled
[17]	CH1CCDE	Channel 1 Capture / Compare PDMA Request Enable 0: Channel 1 PDMA request disabled 1: Channel 1 PDMA request enabled
[16]	CH0CCDE	Channel 0 Capture / Compare PDMA Request Enable 0: Channel 0 PDMA request disabled 1: Channel 0 PDMA request enabled
[10]	TEVIE	Trigger event Interrupt Enable 0: Trigger event interrupt disabled 1: Trigger event interrupt enabled
[8]	UEVIE	Update event Interrupt Enable 0: Update event interrupt disabled 1: Update event interrupt enabled
[3]	CH3CCIE	Channel 3 Capture / Compare Interrupt Enable 0: Channel 3 interrupt disabled 1: Channel 3 interrupt enabled
[2]	CH2CCIE	Channel 2 Capture / Compare Interrupt Enable 0: Channel 2 interrupt disabled 1: Channel 2 interrupt enabled

Bits	Field	Descriptions
[1]	CH1CCIE	Channel 1 Capture / Compare Interrupt Enable 0: Channel 1 interrupt disabled 1: Channel 1 interrupt enabled
[0]	CH0CCIE	Channel 0 Capture / Compare Interrupt Enable 0: Channel 0 interrupt disabled 1: Channel 0 interrupt enabled

Timer Event Generator Register – EVGR

This register contains the software event generation bits.

Offset: 0x078

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	22	21	20	19	18	17	16
	Reserved							
Type/Reset								
	15	14	13	12	11	10	9	8
	Reserved					TEVG	Reserved	UEVG
Type/Reset						WO 0		WO 0
	7	6	5	4	3	2	1	0
	Reserved				CH3CCG	CH2CCG	CH1CCG	CH0CCG
Type/Reset					WO 0	WO 0	WO 0	WO 0

Bits	Field	Descriptions
[10]	TEVG	Trigger Event Generation The trigger event TEV can be generated by setting this bit. It is cleared by hardware automatically. 0: No action 1: TEVIF flag is set
[8]	UEVG	Update Event Generation The update event UEV can be generated by setting this bit. It is cleared by hardware automatically. 0: No action 1: Reinitialize the counter The counter value returns to 0 or the CRR preload value, depending on the counter mode in which the current timer is being used. An update operation of any related registers will also be performed. For more detail descriptions, refer to the corresponding section.

Bits	Field	Descriptions
[3]	CH3CCG	Channel 3 Capture / Compare Generation A Channel 3 capture / compare event can be generated by setting this bit. It is cleared by hardware automatically. 0: No action 1: Capture / compare event is generated on channel 3 If Channel 3 is configured as an input, the counter value is captured into the CH3CCR register and then the CH3CCIF bit is set. If Channel 3 is configured as an output, the CH3CCIF bit is set.
[2]	CH2CCG	Channel 2 Capture / Compare Generation A Channel 2 capture / compare event can be generated by setting this bit. It is cleared by hardware automatically. 0: No action 1: Capture / compare event is generated on channel 2 If Channel 2 is configured as an input, the counter value is captured into the CH2CCR register and then the CH2CCIF bit is set. If Channel 2 is configured as an output, the CH2CCIF bit is set.
[1]	CH1CCG	Channel 1 Capture / Compare Generation A Channel 1 capture / compare event can be generated by setting this bit. It is cleared by hardware automatically. 0: No action 1: Capture / compare event is generated on channel 1 If Channel 1 is configured as an input, the counter value is captured into the CH1CCR register and then the CH1CCIF bit is set. If Channel 1 is configured as an output, the CH1CCIF bit is set.
[0]	CH0CCG	Channel 0 Capture / Compare Generation A Channel 0 capture / compare event can be generated by setting this bit. It is cleared by hardware automatically. 0: No action 1: Capture / compare event is generated on channel 0 If Channel 0 is configured as an input, the counter value is captured into the CH0CCR register and then the CH0CCIF bit is set. If Channel 0 is configured as an output, the CH0CCIF bit is set.

Timer Interrupt Status Register – INTSR

This register stores the timer interrupt status.

Offset: 0x07C

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	22	21	20	19	18	17	16
	Reserved							
Type/Reset								
	15	14	13	12	11	10	9	8
	Reserved					TEVIF	Reserved	UEVIF
Type/Reset						RW 0		RW 0
	7	6	5	4	3	2	1	0
	CH3OCF	CH2OCF	CH1OCF	CH0OCF	CH3CCIF	CH2CCIF	CH1CCIF	CH0CCIF
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0

Bits	Field	Descriptions
[10]	TEVIF	Trigger Event Interrupt Flag This flag is set by hardware on a trigger event and is cleared by software. 0: No trigger event occurs 1: Trigger event occurs
[8]	UEVIF	Update Event Interrupt Flag. This bit is set by hardware on an update event and is cleared by software. 0: No update event occurs 1: Update event occurs Note: The update event is derived from the following conditions: - The counter overflows or underflows - The UEVG bit is asserted - A restart trigger event occurs from the slave trigger input
[7]	CH3OCF	Channel 3 Over-Capture Flag This flag is set by hardware and cleared by software. 0: No over-capture event is detected 1: Capture event occurs again when the CH3CCIF bit is already set and it is not yet cleared by software
[6]	CH2OCF	Channel 2 Over-Capture Flag This flag is set by hardware and cleared by software. 0: No over-capture event is detected 1: Capture event occurs again when the CH2CCIF bit is already set and it is not cleared yet by software
[5]	CH1OCF	Channel 1 Over-Capture Flag This flag is set by hardware and cleared by software. 0: No over-capture event is detected 1: Capture event occurs again when the CH1CCIF bit is already set and it is not cleared yet by software.

Bits	Field	Descriptions
[4]	CH0OCF	Channel 0 Over-Capture Flag This flag is set by hardware and cleared by software. 0: No over-capture event is detected 1: Capture event occurs again when the CH0CCIFbit is already set and it is not yet cleared by software.
[3]	CH3CCIF	Channel 3 Capture / Compare Interrupt Flag - Channel 3 is configured as an output: 0: No match event occurs 1: The contents of the counter CNTR have matched the contents of the CH3CCR register. This flag is set by hardware when the counter value matches the CH3CCR value except in the center-aligned mode. It is cleared by software. - Channel 3 is configured as an input: 0: No input capture occurs 1: Input capture occurs This bit is set by hardware on a capture event. It is cleared by software or by reading the CH3CCR register.
[2]	CH2CCIF	Channel 2 Capture / Compare Interrupt Flag - Channel 2 is configured as an output: 0: No match event occurs 1: The contents of the counter CNTR have matched the contents of the CH2CCR register This flag is set by hardware when the counter value matches the CH2CCR value except in the center-aligned mode. It is cleared by software. - Channel 2 is configured as an input: 0: No input capture occurs 1: Input capture occurs. This bit is set by hardware on a capture event. It is cleared by software or by reading the CH2CCR register.
[1]	CH1CCIF	Channel 1 Capture / Compare Interrupt Flag - Channel 1 is configured as an output: 0: No match event occurs 1: The contents of the counter CNTR have matched the contents of the CH1CCR register This flag is set by hardware when the counter value matches the CH1CCR value except in the center-aligned mode. It is cleared by software. - Channel 1 is configured as an input: 0: No input capture occurs 1: Input capture occurs This bit is set by hardware on a capture event. It is cleared by software or by reading the CH1CCR register.

Bits	Field	Descriptions
[0]	CH0CCIF	Channel 0 Capture / Compare Interrupt Flag - Channel 0 is configured as an output: 0: No match event occurs 1: The contents of the counter CNTR have matched the content of the CH0CCR register This flag is set by hardware when the counter value matches the CH0CCR value except in the center-aligned mode. It is cleared by software. - Channel 0 is configured as an input: 0: No input capture occurs 1: Input capture occurs This bit is set by hardware on a capture event. It is cleared by software or by reading the CH0CCR register.

Timer Counter Register – CNTR

This register stores the timer counter value.

Offset: 0x080

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	22	21	20	19	18	17	16
	Reserved							
Type/Reset								
	15	14	13	12	11	10	9	8
	CNTV							
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0
	7	6	5	4	3	2	1	0
	CNTV							
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0

Bits	Field	Descriptions
[15:0]	CNTV	Counter Value

Timer Prescaler Register – PSCR

This register specifies the timer prescaler value to generate the counter clock.

Offset: 0x084

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	22	21	20	19	18	17	16
	Reserved							
Type/Reset								
	15	14	13	12	11	10	9	8
	PSCV							
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0
	7	6	5	4	3	2	1	0
	PSCV							
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0

Bits	Field	Descriptions
[15:0]	PSCV	Prescaler Value These bits are used to specify the prescaler value to generate the counter clock frequency f_{CK_CNT}. $f_{CK\_CNT} = \frac{f_{CK\_PSC}}{PSCV[15:0] + 1}$, where the f_{CK_PSC} is the prescaler clock source.

Timer Counter Reload Register – CRR

This register specifies the timer counter reload value.

Offset: 0x088

Reset value: 0x0000_FFFF

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	22	21	20	19	18	17	16
	Reserved							
Type/Reset								
	15	14	13	12	11	10	9	8
	CRV							
Type/Reset	RW 1	RW 1	RW 1	RW 1	RW 1	RW 1	RW 1	RW 1
	7	6	5	4	3	2	1	0
	CRV							
Type/Reset	RW 1	RW 1	RW 1	RW 1	RW 1	RW 1	RW 1	RW 1

Bits	Field	Descriptions
[15:0]	CRV	Counter Reload Value The CRV is the reload value which is loaded into the actual counter register.

Channel 0 Capture / Compare Register – CH0CCR

This register specifies the timer channel 0 capture / compare value.

Offset: 0x090

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	22	21	20	19	18	17	16
	Reserved							
Type/Reset								
	15	14	13	12	11	10	9	8
	CH0CCV							
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0
	7	6	5	4	3	2	1	0
	CH0CCV							
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0

Bits	Field	Descriptions
[15:0]	CH0CCV	Channel 0 Capture / Compare Value - When Channel 0 is configured as an output The CH0CCR value is compared with the counter value and the comparison result is used to trigger the CH0OREF output signal. - When Channel 0 is configured as an input The CH0CCR register stores the counter value captured by the last channel 0 capture event.

Channel 1 Capture / Compare Register – CH1CCR

This register specifies the timer channel 1 capture / compare value.

Offset: 0x094

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	22	21	20	19	18	17	16
	Reserved							
Type/Reset								
	15	14	13	12	11	10	9	8
	CH1CCV							
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0
	7	6	5	4	3	2	1	0
	CH1CCV							
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0

Bits	Field	Descriptions
[15:0]	CH1CCV	Channel 1 Capture / Compare Value - When Channel 1 is configured as an output The CH1CCR value is compared with the counter value and the comparison result is used to trigger the CH1OREF output signal. - When Channel 1 is configured as an input The CH1CCR register stores the counter value captured by the last channel 1 capture event.

Channel 2 Capture / Compare Register – CH2CCR

This register specifies the timer channel 2 capture / compare value.

Offset: 0x098

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	22	21	20	19	18	17	16
	Reserved							
Type/Reset								
	15	14	13	12	11	10	9	8
	CH2CCV							
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0
	7	6	5	4	3	2	1	0
	CH2CCV							
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0

Bits	Field	Descriptions
[15:0]	CH2CCV	Channel 2 Capture / Compare Value - When Channel 2 is configured as an output The CH2CCR value is compared with the counter value and the comparison result is used to trigger the CH2OREF output signal. - When Channel 2 is configured as an input The CH2CCR register stores the counter value captured by the last channel 2 capture event.

Channel 3 Capture / Compare Register – CH3CCR

This register specifies the timer channel 3 capture / compare value.

Offset: 0x09C

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	22	21	20	19	18	17	16
	Reserved							
Type/Reset								
	15	14	13	12	11	10	9	8
	CH3CCV							
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0
	7	6	5	4	3	2	1	0
	CH3CCV							
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0

Bits	Field	Descriptions
[15:0]	CH3CCV	Channel 3 Capture / Compare Value - When Channel 3 is configured as an output The CH3CCR value is compared with the counter value and the comparison result is used to trigger the CH3OREF output signal. - When Channel 3 is configured as an input The CH3CCR register stores the counter value captured by the last channel 3 capture event.

Channel 0 Asymmetric Compare Register – CH0ACR

This register specifies the timer channel 0 asymmetric compare value.

Offset: 0x0A0

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	22	21	20	19	18	17	16
	Reserved							
Type/Reset								
	15	14	13	12	11	10	9	8
	CH0ACV							
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0
	7	6	5	4	3	2	1	0
	CH0ACV							
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0

Bits	Field	Descriptions
[15:0]	CH0ACV	Channel 0 Asymmetric Compare Value When channel 0 is configured as asymmetric PWM mode and the counter is counting down, the value written is this register will be compared to the counter.

Channel 1 Asymmetric Compare Register – CH1ACR

This register specifies the timer channel 1 asymmetric compare value.

Offset: 0x0A4

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	22	21	20	19	18	17	16
	Reserved							
Type/Reset								
	15	14	13	12	11	10	9	8
	CH1ACV							
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0
	7	6	5	4	3	2	1	0
	CH1ACV							
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0

Bits	Field	Descriptions
[15:0]	CH1ACV	Channel 1 Asymmetric Compare Value When channel 1 is configured as asymmetric PWM mode and the counter is counting down, the value written is this register will be compared to the counter.

Channel 2 Asymmetric Compare Register – CH2ACR

This register specifies the timer channel 2 asymmetric compare value.

Offset: 0x0A8

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	22	21	20	19	18	17	16
	Reserved							
Type/Reset								
	15	14	13	12	11	10	9	8
	CH2ACV							
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0
	7	6	5	4	3	2	1	0
	CH2ACV							
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0

Bits	Field	Descriptions
[15:0]	CH2ACV	Channel 2 Asymmetric Compare Value When channel 2 is configured as asymmetric PWM mode and the counter is counting down, the value written is this register will be compared to the counter.

Channel 3 Asymmetric Compare Register – CH3ACR

This register specifies the timer channel 3 asymmetric compare value.

Offset: 0x0AC

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	22	21	20	19	18	17	16
	Reserved							
Type/Reset								
	15	14	13	12	11	10	9	8
	CH3ACV							
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0
	7	6	5	4	3	2	1	0
	CH3ACV							
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0

Bits	Field	Descriptions
[15:0]	CH3ACV	Channel 3 Asymmetric Compare Value When channel 3 is configured as asymmetric PWM mode and the counter is counting down, the value written is this register will be compared to the counter.

14 Basic Function Timer (BFTM)

Introduction

The Basic Function Timer Module, BFTM, is a 32-bit up-counting counter designed to measure time intervals, generate one shot pulses or generate repetitive interrupts. The BFTM can operate in two modes which are repetitive and one shot modes. The repetitive mode restarts the counter at each compare match event which is generated by the internal comparator. The BFTM also supports a one shot mode which will force the counter to stop counting when a compare match event occurs.

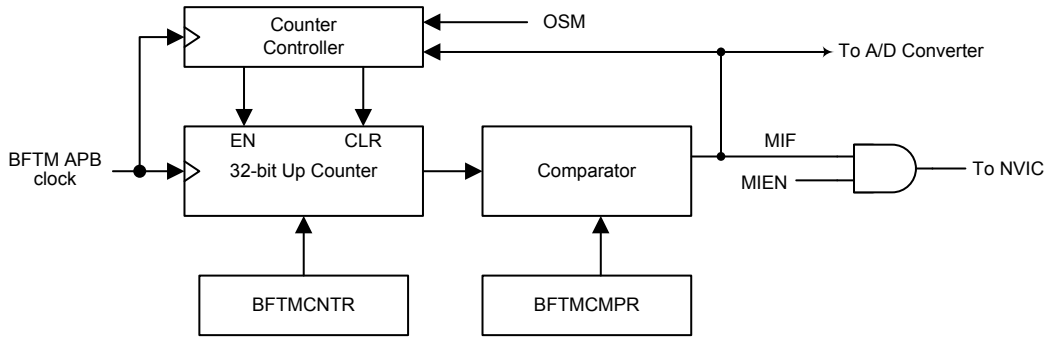


Figure 65. BFTM Block Diagram

Features

- 32-bit up-counting counter
- Compare Match function
- Includes debug mode
- Clock source: BFTM APB clock
- Counter value can be R/W on the fly
- One shot mode: counter stops counting when compare match occurs
- Repetitive mode: counter restarts when compare match occurs
- Compare Match interrupt enable/disable control

Functional Description

The BFTM is a 32-bit up-counting counter which is driven by the BFTM APB clock, PCLK. The counter value can be changed or read at any time even when the timer is counting. The BFTM supports two operating modes known as the repetitive mode and one shot mode allowing the measurement of time intervals or the generation of periodic time durations.

Repetitive Mode

The BFTM counts up from zero to a specific compare value which is pre-defined by the BFTMCMPR register. When the BFTM operates in the repetitive mode and the counter reaches a value equal to the specific compare value in the BFTMCMPR register, the timer will generate a compare match event signal, MIF. When this occurs, the counter will be reset to 0 and resume its counting operation. When the MIF signal is generated, a BFTM compare match interrupt will also be generated periodically if the compare match interrupt is enabled by setting the corresponding interrupt control bit, MIEN, to 1. The counter value will remain unchanged and the counter will stop counting if it is disabled by clearing the CEN bit to 0.

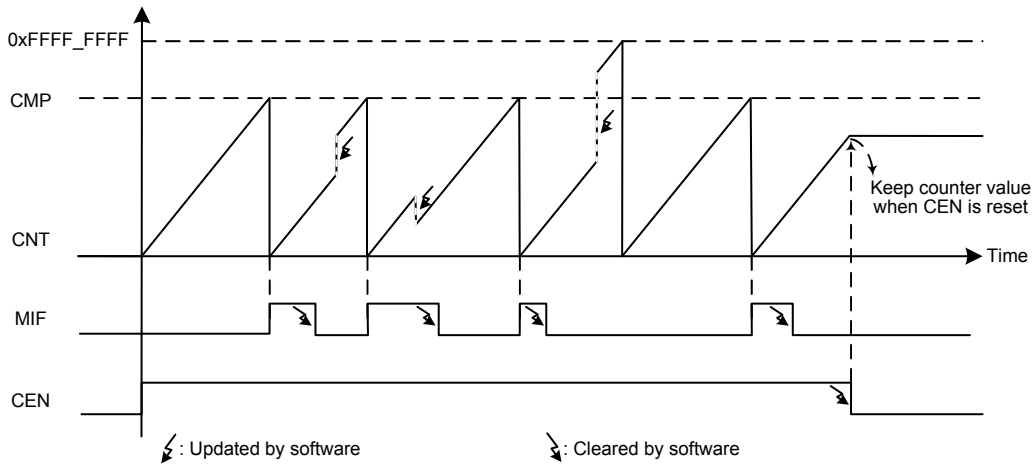


Figure 66. BFTM – Repetitive Mode

One Shot Mode

By setting the OSM bit in BFTMCR register to 1, the BFTM will operate in the one shot mode. The BFTM starts to count when the CEN bit is set to 1 by the application program. The counter value will remain unchanged if the CEN bit is cleared to 0 by the application program. However, the counter value will be reset to 0 and stop counting when the CEN bit is cleared automatically to 0 by the internal hardware when a counter compare match event occurs.

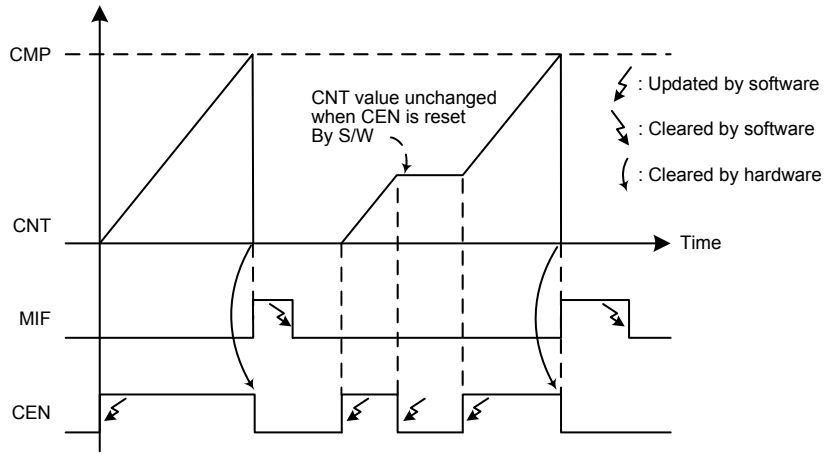


Figure 67. BFTM – One Shot Mode

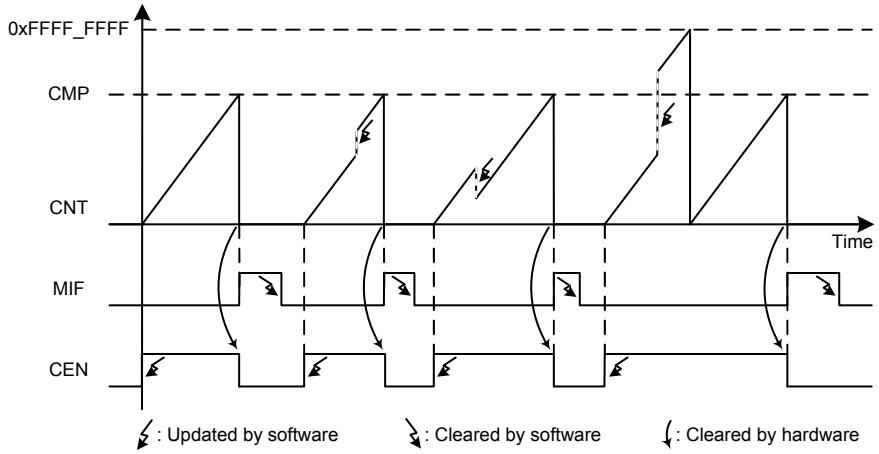


Figure 68. BFTM – One Shot Mode Counter Updating

Trigger ADC Start

When a BFTM compare match event occurs, a compare match interrupt flag, MIF, will be generated which can be used as an A/D Converter input trigger source.

Register Map

The following table shows the BFTM registers and their reset values.

Table 33. BFTM Register Map

Register	Offset	Description	Reset Value
BFTMCR	0x000	BFTM Control Register	0x0000_0000
BFTMSR	0x004	BFTM Status Register	0x0000_0000
BFTMCNTR	0x008	BFTM Counter Value Register	0x0000_0000
BFTMCMPR	0x00C	BFTM Compare Value Register	0xFFFF_FFFF

Register Descriptions

BFTM Control Register – BFTMCR

This register specifies the overall BFTM control bits.

Offset: 0x000

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	22	21	20	19	18	17	16
	Reserved							
Type/Reset								
	15	14	13	12	11	10	9	8
	Reserved							
Type/Reset								
	7	6	5	4	3	2	1	0
	Reserved					CEN	OSM	MIEN
Type/Reset						RW 0	RW 0	RW 0

Bits	Field	Descriptions
[2]	CEN	BFTM Counter Enable Control 0: BFTM is disabled 1: BFTM is enabled When this bit is set to 1, the BFTM counter will start to count. The counter will stop counting and the counter value will remain unchanged when the CEN bit is cleared to 0 by the application program regardless of whether it is in the repetitive or one shot mode. However, in the one shot mode, the counter will stop counting and be reset to 0 when the CEN bit is cleared to 0 by the timer hardware circuitry which results from a compare match event.
[1]	OSM	BFTM One Shot Mode Selection 0: Counter operates in repetitive mode 1: Counter operates in one shot mode
[0]	MIEN	BFTM Compare Match Interrupt Enable Control 0: Compare Match Interrupt is disabled 1: Compare Match Interrupt is enabled

BFTM Status Register – BFTMSR

This register specifies the BFTM status.

Offset: 0x004

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	22	21	20	19	18	17	16
	Reserved							
Type/Reset								
	15	14	13	12	11	10	9	8
	Reserved							
Type/Reset								
	7	6	5	4	3	2	1	0
	Reserved							MIF
Type/Reset								W0C 0

Bits	Field	Descriptions
[0]	MIF	BFTM Compare Match Interrupt Flag 0: No compare match event occurs 1: Compare match event occurs When the counter value, CNT, is equal to the compare register value, CMP, a compare match event will occur and the corresponding interrupt flag, MIF will be set. The MIF bit is cleared to 0 by writing a data “0”.

BFTM Counter Value Register – BFTMCNTR

This register specifies the BFTM counter value.

Offset: 0x008

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	CNT							
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0
	23	22	21	20	19	18	17	16
	CNT							
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0
	15	14	13	12	11	10	9	8
	CNT							
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0
	7	6	5	4	3	2	1	0
	CNT							
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0

Bits	Field	Descriptions
[31:0]	CNT	BFTM Counter Value Register A 32-bit BFTM counter value is stored in this field which can be read or written on the fly.

BFTM Compare Value Register – BFTMCMPR

The register specifies the BFTM compare value.

Offset: 0x00C

Reset value: 0xFFFF_FFFF

	31	30	29	28	27	26	25	24
	CMP							
Type/Reset	RW 1	RW 1	RW 1	RW 1	RW 1	RW 1	RW 1	RW 1
	23	22	21	20	19	18	17	16
	CMP							
Type/Reset	RW 1	RW 1	RW 1	RW 1	RW 1	RW 1	RW 1	RW 1
	15	14	13	12	11	10	9	8
	CMP							
Type/Reset	RW 1	RW 1	RW 1	RW 1	RW 1	RW 1	RW 1	RW 1
	7	6	5	4	3	2	1	0
	CMP							
Type/Reset	RW 1	RW 1	RW 1	RW 1	RW 1	RW 1	RW 1	RW 1

Bits	Field	Descriptions
[31:0]	CMP	BFTM Compare Value Register This register specifies a 32-bit BFTM compare value which is used for comparison with the BFTM counter value.

15 Single-Channel Timer (SCTM)

Introduction

The Single-Channel Timer consists of one 16-bit up-counter, one 16-bit Capture / Compare Register (CCR), one 16-bit Counter-Reload Register (CRR) and several control / status registers. It can be used for a variety of purposes including general timer, input signal pulse width measurement or output waveform generation such as single pulse generation or PWM output.

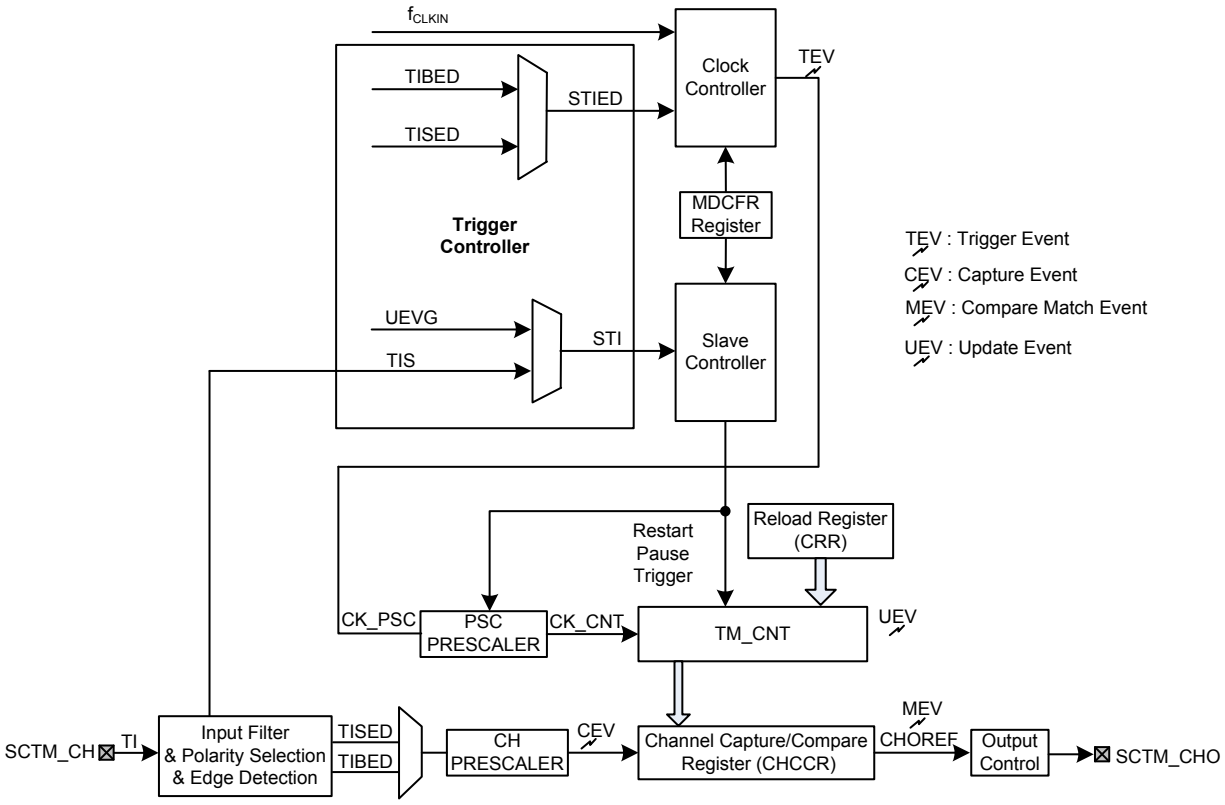


Figure 69. SCTM Block Diagram

Features

- 16-bit auto-reload up-counter
- 16-bit programmable prescaler that allows division of the counter clock frequency by any factor between 1 and 65536
- Single channel for
  - Input Capture function
  - Compare Match Output
  - PWM Waveform Output
- Interrupt generation with the following events
  - Update event
  - Trigger event
  - Input capture event
  - Output compare match event

Functional Descriptions

Counter Mode

Up-Counting

The counter counts continuously from 0 to the counter-reload value, which is defined in the CRR register. Once the counter reaches the counter-reload value, the Timer Module generates an overflow event and the counter restarts to count once again from 0. This action will continue repeatedly. When the update event is generated by setting the UEVG bit in the EVGR register to 1, the counter value will also be initialized to 0.

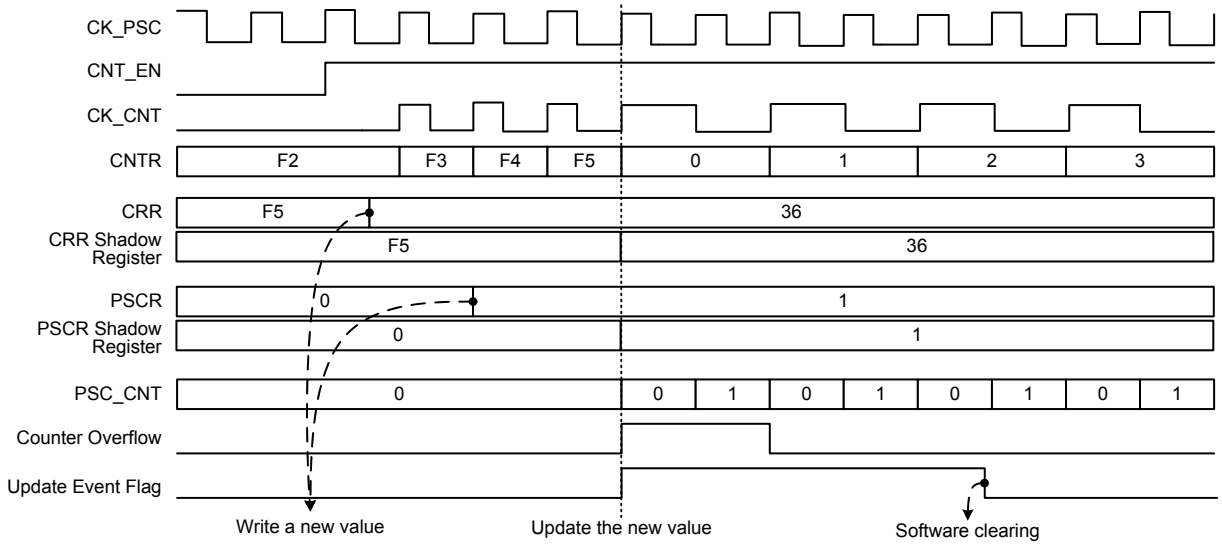


Figure 70. Up-counting Example

Clock Controller

The following describes the Timer Module clock controller which determines the clock source of the internal prescaler counter.

- Internal APB clock f_{CLKIN}

  The default internal clock source is the APB clock f_{CLKIN} used to drive the counter prescaler when the slave mode is disabled. When the slave mode selection bits SMSEL are set to 0x4, 0x5 or 0x6, the internal APB clock f_{CLKIN} is the counter prescaler driving clock source. If the slave mode controller is enabled by setting SMSEL field in the MDCFR register to 0x7, the prescaler is clocked by other clock sources selected by the TRSEL field in the TRCFR register and described as follows.

- STIED

  The counter prescaler can count during each rising edge of the STI signal. This mode can be selected by setting the SMSEL field to 0x7 in the MDCFR register. Here the counter will act as an event counter. The input event, known as STI here, can be selected by setting the TRSEL field to an available value except the value of 0x0. When the STI signal is selected as the clock source, the internal edge detection circuitry will generate a clock pulse during each STI signal rising edge to drive the counter prescaler. It is important to note that if the TRSEL field is set to 0x0 to select the software UEVG bit as the trigger source, then when the SMSEL field is set to 0x7, the counter will be updated instead of counting.

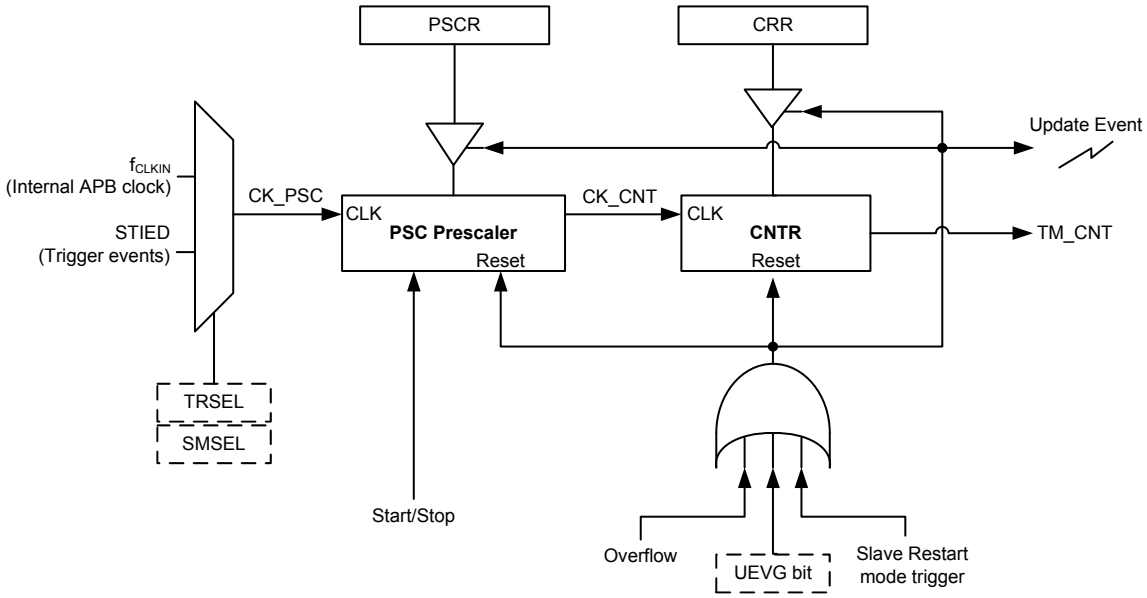


Figure 71. SCTM Clock Selection Source

Trigger Controller

The trigger controller is used to select the trigger source and setup the trigger level or edge trigger condition. For the internal trigger input, it can be selected by the Trigger Selection bits TRSEL in the TRCFR register. For all the trigger sources except the UEVG bit software trigger, the internal edge detection circuitry will generate a clock pulse at each trigger signal rising edge to stimulate some SCTM functions which are triggered by a trigger signal rising edge.

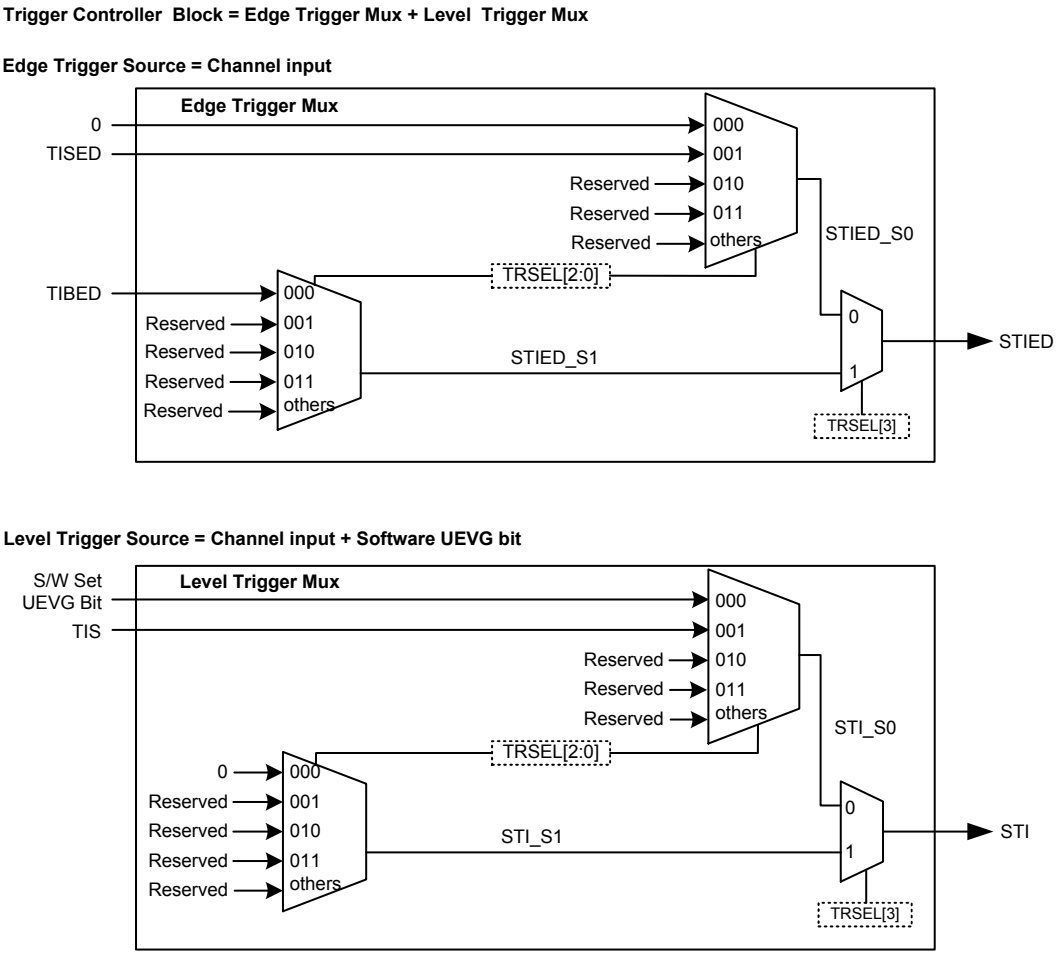


Figure 72. Trigger Control Block

Slave Controller

The SCTM can be synchronized with an external trigger in several modes including the Restart mode, the Pause mode and the Trigger mode which is selected by the SMSEL field in the MDCFR register. The trigger input of these modes comes from the STI signal which is selected by the TRSEL field in the TRCFR register. The operation modes in the Slave Controller are described in the accompanying sections.

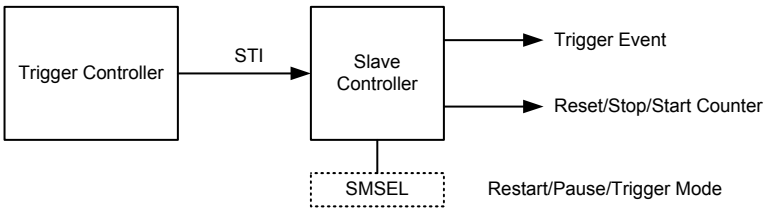


Figure 73. Slave Controller Diagram

Restart Mode

The counter and its prescaler can be reinitialized in response to a rising edge of the STI signal. When an STI rising edge occurs, the update event software generation bit named UEVG will automatically be asserted by hardware and the trigger event flag will also be set. Then the counter and prescaler will be reinitialized. Although the UEVG bit is set to 1 by hardware, the update event does not really occur. It depends upon whether the update event disable control bit UEVDIS is set to 1 or not. If the UEVDIS is set to 1 to disable the update event to occur, there will no update event be generated, however the counter and prescaler are still reinitialized when the STI rising edge occurs. If the UEVDIS bit in the CNTCFR register is cleared to enable the update event to occur, an update event will be generated together with the STI rising edge, then all the preloaded registers will be updated.

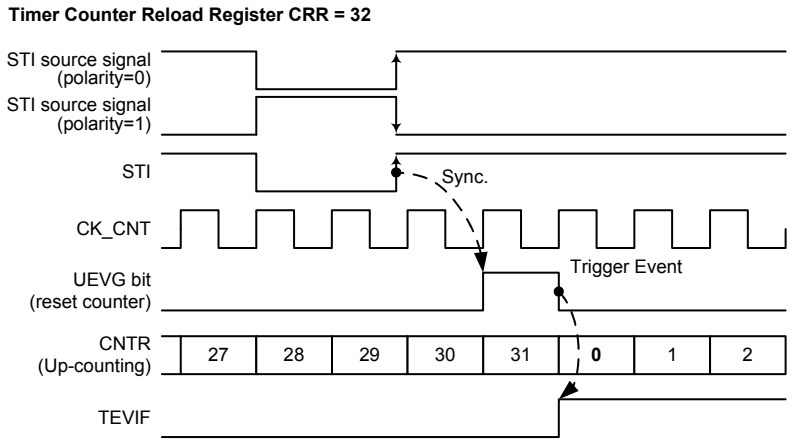


Figure 74. SCTM in Restart Mode

Pause Mode

In the Pause Mode, the selected STI input signal level is used to control the counter start / stop operation. The counter starts to count when the selected STI signal is at a high level and stops counting when the STI signal is changed to a low level, here the counter will maintain its present value and will not be reset. Since the Pause function depends upon the STI level to control the counter stop / start operation, the selected STI trigger signal can not be derived from the TIBED signal.

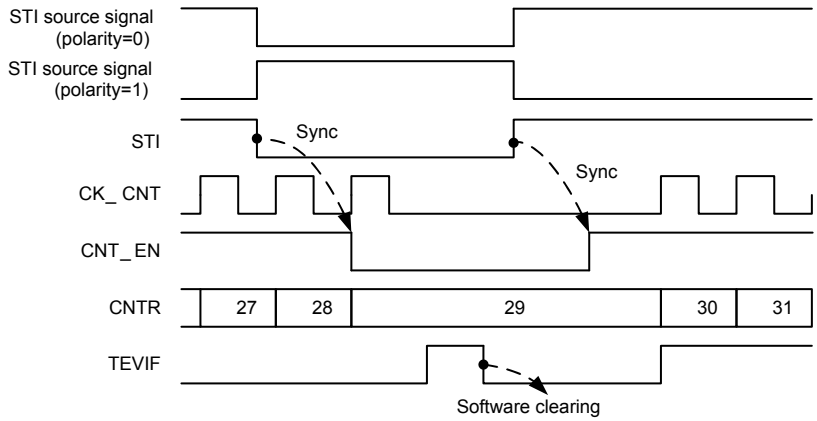


Figure 75. SCTM in Pause Mode

Trigger Mode

After the counter is disabled to count, the counter can resume counting when an STI rising edge signal occurs. When an STI rising edge occurs, the counter will start to count from the current value in the counter. Note that if the STI signal is selected to be derived from the UEVG bit software trigger, the counter will not resume counting. When software triggering using the UEVG bit is selected as the STI source signal, there will be no clock pulse generated which can be used to make the counter resume counting. Note that the STI signal is only used to enable the counter to resume counting and has no effect on controlling the counter to stop counting.

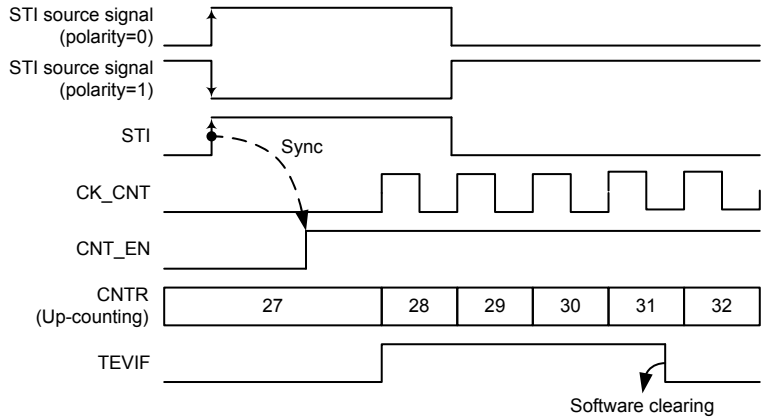


Figure 76. SCTM in Trigger Mode

Channel Controller

The SCTM channel can be used as the capture input or compare match output. The capture input or compare match output channel is composed of a preload register and a shadow register. Data access of the APB bus is always implemented by reading / writing the preload register.

When used in the input capture mode, the counter value is captured into the CHCCR shadow register first and then transferred into the CHCCR preload register when the capture event occurs.

When used in the compare match output mode, the contents of the CHCCR preload register is copied into the associated shadow register; the counter value is then compared with the register value.

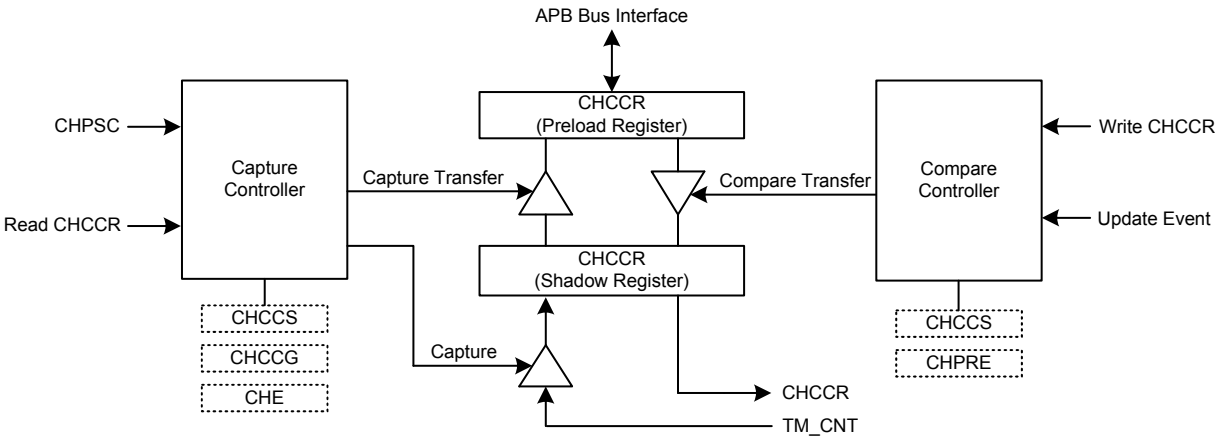


Figure 77. Capture / Compare Block Diagram

Capture Counter Value Transferred to CHCCR

When the channel is used as a capture input, the counter value is captured into the Channel Capture / Compare Register (CHCCR) when an effective input signal transition occurs. Once the capture event occurs, the CHCCIF flag in the INTSR register is set accordingly. If the CHCCIF bit is already set, i.e., the flag has not yet been cleared by software, and another capture event on this channel occurs, the corresponding channel Over-Capture flag, named CHOCF, will be set.

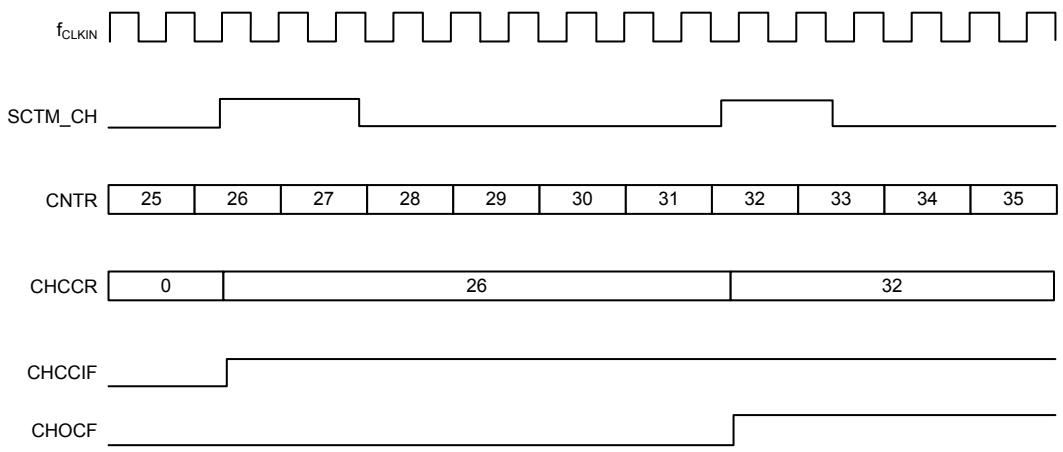


Figure 78. Input Capture Mode

Input Stage

The input stage consists of a digital filter, a channel polarity selection, edge detection and a channel prescaler. The channel input signal (TI) is sampled by a digital filter to generate a filtered input signal TIFP. Then the channel polarity and the edge detection block can generate a TISED signal for the input capture function. The effective input event number can be set by the channel capture input source prescaler setting field (CHPSC).

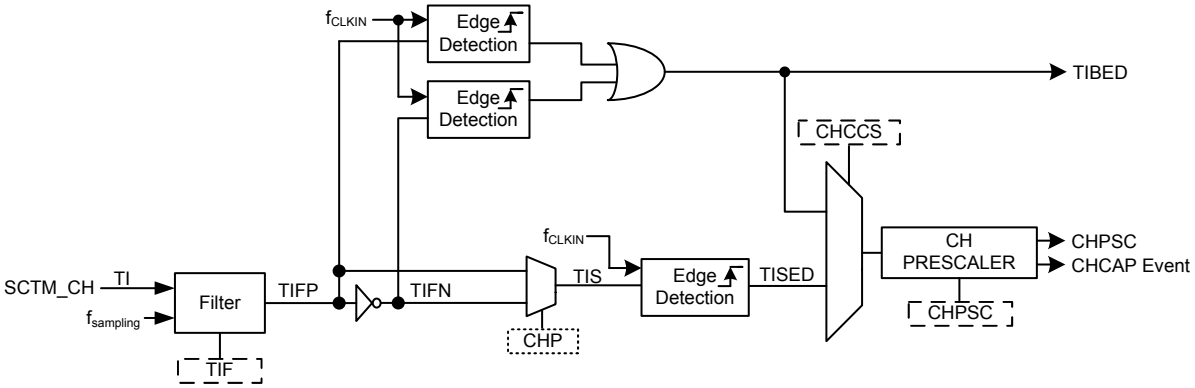


Figure 79. Channel Input Stages

Digital Filter

The digital filter is embedded in the channel input stage. The digital filter in the SCTM is an N-event counter where N refers to how many valid transitions are necessary to output a filtered signal. The N value can be 0, 2, 4, 5, 6 or 8 according to the user selection for this digital filter.

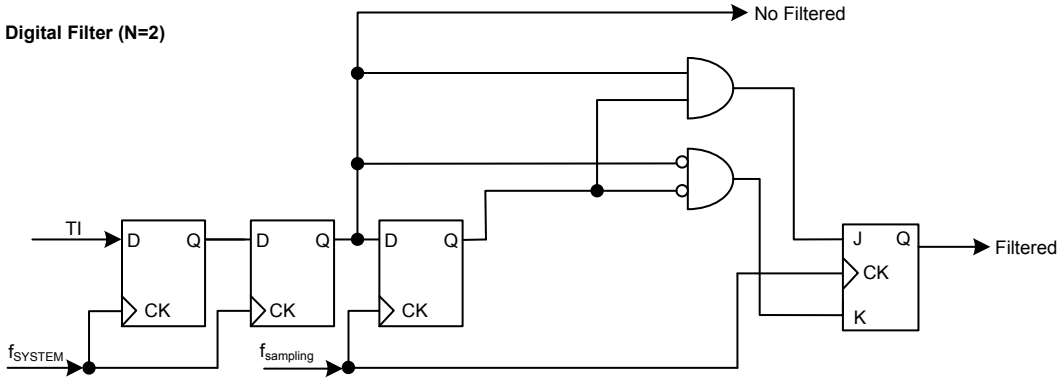


Figure 80. TI Digital Filter Diagram with N = 2

Output Stage

The SCTM output has function for compare match, single pulse or PWM output. The channel output SCTM_CHO is controlled by the CHOM, CHP and CHE bits in the corresponding CHOCFR, CHPOLR and CHCTR registers.

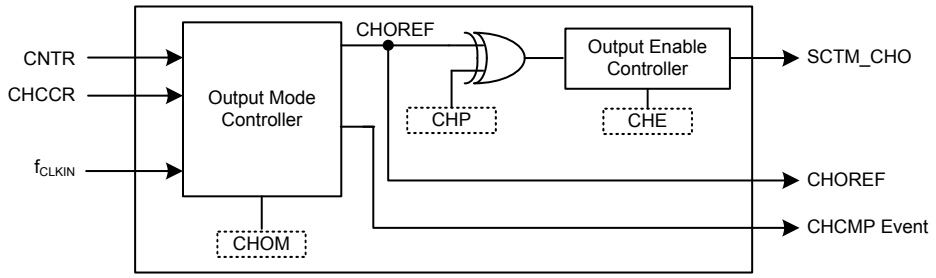


Figure 81. Output Stage Block Diagram

Channel Output Reference Signal

When the SCTM is used in the compare match output mode, the CHOREF signal (Channel Output Reference signal) is defined by the CHOM bit setup. The CHOREF signal has several types of output function which defines what happens to the output when the counter value matches the contents of the CHCCR register. In addition to the low, high and toggle CHOREF output types, there are also PWM mode 1 and PWM mode 2 outputs. In these modes, the CHOREF signal level is changed according to the relationship between the counter value and the CHCCR content. There are also two modes which will force the output into an inactive or active state irrespective of the CHCCR content or counter values. With regard to a more detailed description refer to the relative bit definition. The accompanying Table 34 shows a summary of the output type setup.

Table 34. Compare Match Output Setup

CHOM Value	Compare Match Level
0x0	No change
0x1	Clear Output to 0
0x2	Set Output to 1
0x3	Toggle Output
0x4	Force Inactive Level
0x5	Force Active Level
0x6	PWM Mode 1
0x7	PWM Mode 2

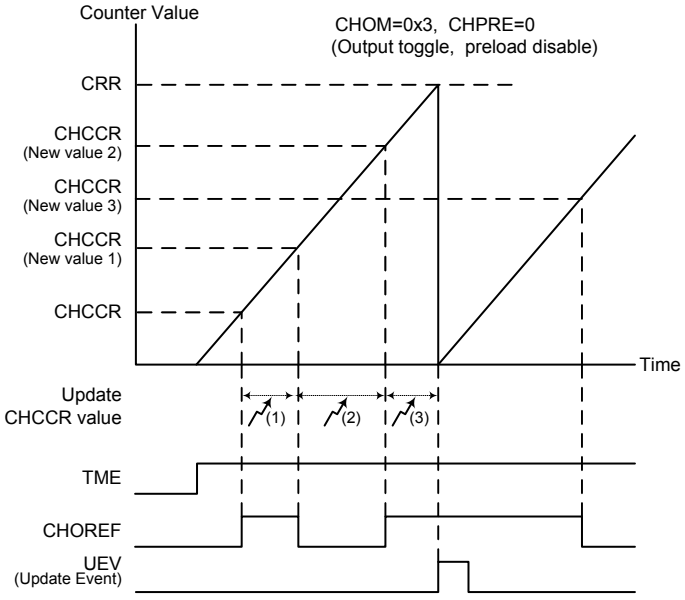


Figure 82. Toggle Mode Channel Output Reference Signal – CHPRE = 0

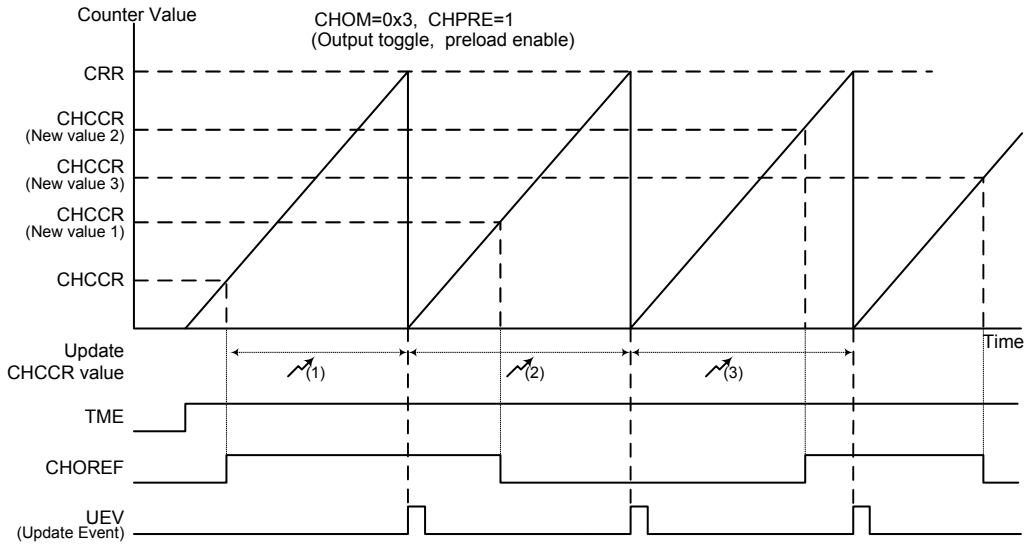


Figure 83. Toggle Mode Channel Output Reference Signal – CHPRE = 1

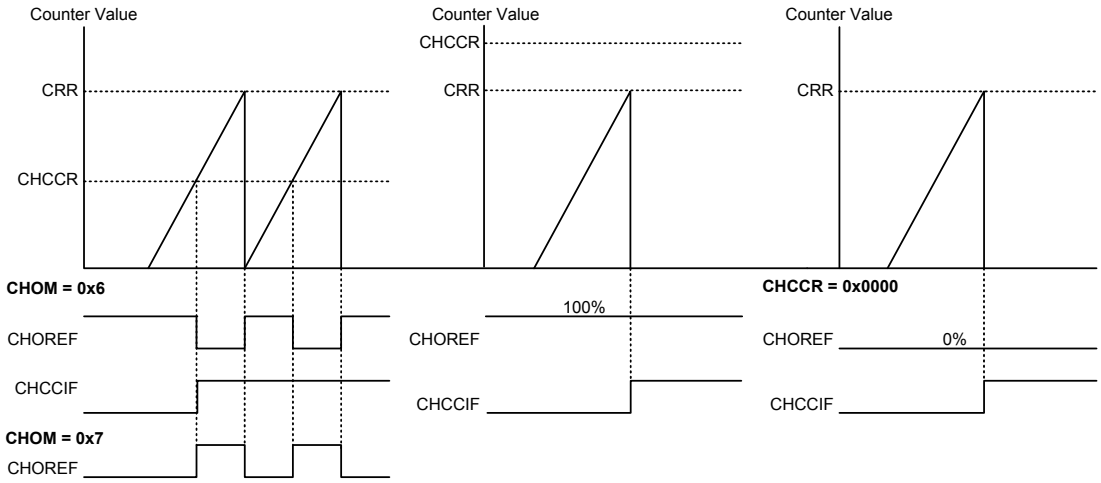


Figure 84. PWM Mode Channel Output Reference Signal

Update Management

The Update event is used to update the CRR, the PSCR and the CHCCR values from the actual registers to the corresponding shadow registers. An update event will occur when the counter overflows, the software update control bit is triggered or an update event from the slave controller is generated.

The UEVDIS bit in the CNTCFR register can determine whether the update event occurs or not. When the update event occurs, the corresponding update event interrupt will be generated depending upon whether the update event interrupt generation function is enabled or not by configuring the UGDIS bit in the CNTCFR register. For more detailed description, refer to the UEVDIS and UGDIS bit definition in the CNTCFR register.

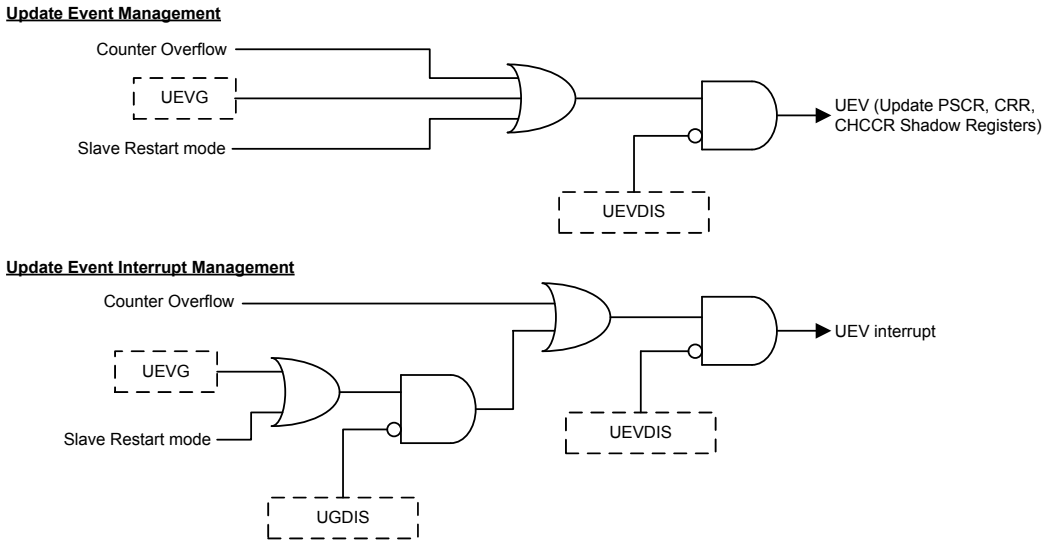


Figure 85. Update Event Setting Diagram

Register Map

The following table shows the SCTM registers and reset values.

Table 35. SCTM Register Map

Register	Offset	Description	Reset Value
CNTCFR	0x000	Timer Counter Configuration Register	0x0000_0000
MDCFR	0x004	Timer Mode Configuration Register	0x0000_0000
TRCFR	0x008	Timer Trigger Configuration Register	0x0000_0000
CTR	0x010	Timer Control Register	0x0000_0000
CHICFR	0x020	Channel Input Configuration Register	0x0000_0000
CHOCFR	0x040	Channel Output Configuration Register	0x0000_0000
CHCTR	0x050	Channel Control Register	0x0000_0000
CHPOLR	0x054	Channel Polarity Configuration Register	0x0000_0000
DICTR	0x074	Timer Interrupt Control Register	0x0000_0000
EVGR	0x078	Timer Event Generator Register	0x0000_0000
INTSR	0x07C	Timer Interrupt Status Register	0x0000_0000
CNTR	0x080	Timer Counter Register	0x0000_0000
PSCR	0x084	Timer Prescaler Register	0x0000_0000
CRR	0x088	Timer Counter Reload Register	0x0000_FFFF
CHCCR	0x090	Channel Capture / Compare Register	0x0000_0000

Register Descriptions

Timer Counter Configuration Register – CNTCFR

This register specifies the SCTM counter configuration.

Offset: 0x000

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	22	21	20	19	18	17	16
	Reserved							
Type/Reset								
	15	14	13	12	11	10	9	8
	Reserved						CKDIV	
Type/Reset							RW 0	RW 0
	7	6	5	4	3	2	1	0
	Reserved						UGDIS	UEVDIS
Type/Reset							RW 0	RW 0

Bits	Field	Descriptions
[9:8]	CKDIV	Clock Division These two bits define the frequency ratio between the timer clock (f_{CLKIN}) and the dead-time clock (f_{DTS}). The dead-time clock is also used for digital filter sampling clock. 00: $f_{DTS} = f_{CLKIN}$ 01: $f_{DTS} = f_{CLKIN} / 2$ 10: $f_{DTS} = f_{CLKIN} / 4$ 11: Reserved
[1]	UGDIS	Update event interrupt generation disable control 0: Any of the following events will generate an update interrupt - Counter overflow - Setting the UEVG bit - Update generation through the slave mode 1: Only counter overflow generates an update interrupt
[0]	UEVDIS	Update event Disable control 0: Enable the update event request by one of following events: - Counter overflow - Setting the UEVG bit - Update generation through the slave mode 1: Disable the update event (However the counter and the prescaler are reinitialized if the UEVG bit is set or if a hardware restart is received from the slave mode)

Timer Mode Configuration Register – MDCFR

This register specifies the SCTM slave mode selection.

Offset: 0x004

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	22	21	20	19	18	17	16
	Reserved							
Type/Reset								
	15	14	13	12	11	10	9	8
	Reserved					SMSEL		
Type/Reset						RW 0	RW 0	RW 0
	7	6	5	4	3	2	1	0
	Reserved							
Type/Reset								

Bits	Field	Descriptions
[10:8]	SMSEL	Slave Mode Selection

SMSEL [2:0]	Mode	Descriptions
000	Disable mode	The prescaler is clocked directly by the internal clock.
100	Restart Mode	The counter value restarts from 0 or the CRR shadow register value depending upon the counter mode on the rising edge of the STI signal. The registers will also be updated.
101	Pause Mode	The counter starts to count when the selected trigger input STI is high. The counter stops counting on the instant, not being reset, when the STI signal changes its state to a low level. Both the counter start and stop control are determined by the STI signal.
110	Trigger Mode	The counter starts to count from the original value in the counter on the rising edge of the selected trigger input STI. Only the counter start control is determined by the STI signal.
111	STIED	The rising edge of the selected trigger signal STI will clock the counter.
Others	Reserved	

Timer Trigger Configuration Register – TRCFR

This register specifies the trigger source selection of SCTM.

Offset: 0x008

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	22	21	20	19	18	17	16
	Reserved							
Type/Reset								
	15	14	13	12	11	10	9	8
	Reserved							
Type/Reset								
	7	6	5	4	3	2	1	0
	Reserved				TRSEL			
Type/Reset					RW 0	RW 0	RW 0	RW 0

Bits	Field	Descriptions
[3:0]	TRSEL	Trigger Source Selection These bits are used to select the trigger input (STI) for counter synchronization. 0000: Software Trigger by setting the UEVG bit 0001: Filtered input of channel (TIS) 0011: Reserved 1000: Channel both edge detector (TIBED) Others: Reserved Note: These bits must be updated only when they are not in use, i.e. the slave mode is disabled by setting the SMSEL field to 0x0.

Timer Counter Register – CTR

This register specifies the timer enable bit (TME), CRR buffer enable bit (CRBE).

Offset: 0x010

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	22	21	20	19	18	17	16
	Reserved							
Type/Reset								
	15	14	13	12	11	10	9	8
	Reserved							
Type/Reset								
	7	6	5	4	3	2	1	0
	Reserved						CRBE	TME
Type/Reset							RW 0	RW 0

Bits	Field	Descriptions
[1]	CRBE	Counter-Reload register Buffer Enable 0: Counter reload register can be updated immediately 1: Counter reload register cannot be updated until the update event occurs
[0]	TME	Timer Enable bit 0: SCTM off 1: SCTM on – SCTM functions normally When the TME bit is cleared to 0, the counter is stopped and the SCTM consumes no power in any operation mode except for the single pulse mode and the slave trigger mode. In these two modes the TME bit can automatically be set to 1 by hardware which permits all the SCTM registers to function normally.

Channel Input Configuration Register – CHICFR

This register specifies the channel input mode configuration.

Offset: 0x020

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	22	21	20	19	18	17	16
	Reserved				CHPSC		CHCCS	
Type/Reset					RW 0	RW 0	RW 0	RW 0
	15	14	13	12	11	10	9	8
	Reserved							
Type/Reset								
	7	6	5	4	3	2	1	0
	Reserved				TIF			
Type/Reset					RW 0	RW 0	RW 0	RW 0

Bits	Field	Descriptions
[19:18]	CHPSC	Channel Capture Input Source Prescaler Setting These bits define the effective events of the channel capture input. Note that the prescaler is reset once the Channel Capture / Compare Enable bit, CHE, in the Channel Control register named CHCTR is cleared to 0. 00: No prescaler, channel capture input signal is chosen for each active event. 01: Channel Capture input signal is chosen for every 2 events. 10: Channel Capture input signal is chosen for every 4 events. 11: Channel Capture input signal is chosen for every 8 events.
[17:16]	CHCCS	Channel Capture/Compare Selection. 00: Channel is configured as an output. 01: Channel is configured as an input derived from the TI signal. 10: Reserved. 11: Channel is configured as an input which comes from the TIBED signal. Note: The CHCCS field can be accessed only when the CHE bit is cleared to 0.

Bits	Field	Descriptions
[3:0]	TIF	Channel Input Source TI Filter Setting These bits define the frequency divided ratio used to sample the TI signal. The Digital filter in the SCTM is an N-event counter where N is defined as how many valid transitions are necessary to output a filtered signal. 0000: No filter, the sampling clock is f_{SYSTEM}. 0001: $f_{SAMPLING} = f_{CLKIN}$, N = 2 0010: $f_{SAMPLING} = f_{CLKIN}$, N = 4 0011: $f_{SAMPLING} = f_{CLKIN}$, N = 8 0100: $f_{SAMPLING} = f_{DTS}$ / 2, N = 6 0101: $f_{SAMPLING} = f_{DTS}$ / 2, N = 8 0110: $f_{SAMPLING} = f_{DTS}$ / 4, N = 6 0111: $f_{SAMPLING} = f_{DTS}$ / 4, N = 8 1000: $f_{SAMPLING} = f_{DTS}$ / 8, N = 6 1001: $f_{SAMPLING} = f_{DTS}$ / 8, N = 8 1010: $f_{SAMPLING} = f_{DTS}$ / 16, N = 5 1011: $f_{SAMPLING} = f_{DTS}$ / 16, N = 6 1100: $f_{SAMPLING} = f_{DTS}$ / 16, N = 8 1101: $f_{SAMPLING} = f_{DTS}$ / 32, N = 5 1110: $f_{SAMPLING} = f_{DTS}$ / 32, N = 6 1111: $f_{SAMPLING} = f_{DTS}$ / 32, N = 8

Channel Output Configuration Register – CHOCFR

This register specifies the channel output mode configuration.

Offset: 0x040

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	22	21	20	19	18	17	16
	Reserved							
Type/Reset								
	15	14	13	12	11	10	9	8
	Reserved							
Type/Reset								
	7	6	5	4	3	2	1	0
	Reserved			CHPRE	Reserved	CHOM		
Type/Reset				RW 0		RW 0	RW 0	RW 0

Bits	Field	Descriptions
[4]	CHPRE	Channel Capture / Compare Register (CHCCR) Preload Enable 0: CHCCR preload function is disabled. The CHCCR register can be immediately assigned a new value when the CHPRE bit is cleared to 0 and the updated CHCCR value is used immediately. 1: CHCCR preload function is enabled. The new CHCCR value will not be transferred to its shadow register until the update event occurs.
[2:0]	CHOM	Channel Output Mode Setting These bits define the functional types of the output reference signal CHOREF. 000: No Change 001: Output 0 on compare match 010: Output 1 on compare match 011: Output toggles on compare match 100: Force inactive – CHOREF is forced to 0 101: Force active – CHOREF is forced to 1 110: PWM mode 1 - During up-counting, channel has an active level when CNTR < CHCCR or otherwise has an inactive level. 111: PWM mode 2 - During up-counting, channel is has an inactive level when CNTR < CHCCR or otherwise has an active level.

Channel Control Register – CHCTR

This register contains the channel capture input or compare output function enable control bit.

Offset: 0x050

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	22	21	20	19	18	17	16
	Reserved							
Type/Reset								
	15	14	13	12	11	10	9	8
	Reserved							
Type/Reset								
	7	6	5	4	3	2	1	0
	Reserved							CHE
Type/Reset								RW 0

Bits	Field	Descriptions
[0]	CHE	Channel Capture / Compare Enable - Channel is configured as an input (CHCCS = 0x1 / 0x3) 0: Input Capture Mode disabled 1: Input Capture Mode enabled - Channel is configured as an output (CHCCS = 0x0) 0: Off – Channel output signal CHO is not active 1: On – Channel output signal CHO generated on the corresponding output pin

Channel Polarity Configuration Register – CHPOLR

This register contains the channel capture input or compare output polarity control.

Offset: 0x054

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	22	21	20	19	18	17	16
	Reserved							
Type/Reset								
	15	14	13	12	11	10	9	8
	Reserved							
Type/Reset								
	7	6	5	4	3	2	1	0
	Reserved							CHP
Type/Reset								RW 0

Bits	Field	Descriptions
[0]	CHP	Channel Capture / Compare Polarity - When Channel is configured as an input (CHCCS = 0x1 / 0x3) 0: Capture event occurs on a Channel rising edge 1: Capture event occurs on a Channel falling edge - When Channel is configured as an output (CHCCS = 0x0) 0: Channel Output active high 1: Channel Output active low

Timer Interrupt Control Register – DICTR

This register contains the timer interrupt enable control bits.

Offset: 0x074

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	22	21	20	19	18	17	16
	Reserved							
Type/Reset								
	15	14	13	12	11	10	9	8
	Reserved					TEVIE	Reserved	UEVIE
Type/Reset						RW 0		RW 0
	7	6	5	4	3	2	1	0
	Reserved							CHCCIE
Type/Reset								RW 0

Bits	Field	Descriptions
[10]	TEVIE	Trigger event Interrupt Enable 0: Trigger event interrupt disabled 1: Trigger event interrupt enabled
[8]	UEVIE	Update event Interrupt Enable 0: Update event interrupt disabled 1: Update event interrupt enabled
[0]	CHCCIE	Channel Capture / Compare Interrupt Enable 0: Channel interrupt disabled 1: Channel interrupt enabled

Timer Event Generator Register – EVGR

This register contains the software event generation bits.

Offset: 0x078

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	22	21	20	19	18	17	16
	Reserved							
Type/Reset								
	15	14	13	12	11	10	9	8
	Reserved					TEVG	Reserved	UEVG
Type/Reset						WO 0		WO 0
	7	6	5	4	3	2	1	0
	Reserved							CHCCG
Type/Reset								WO 0

Bits	Field	Descriptions
[10]	TEVG	Trigger Event Generation The trigger event TEV can be generated by setting this bit. It is cleared by hardware automatically. 0: No action 1: TEVIF flag is set
[8]	UEVG	Update Event Generation The update event UEV can be generated by setting this bit. It is cleared by hardware automatically. 0: No action 1: Reinitialize the counter If this bit is set, the counter value returns to 0 or the CRR preload value, depending on the counter mode in which the current timer is being used. An update operation of any related registers will also be performed. For more detail descriptions, refer to the corresponding section.
[0]	CHCCG	Channel Capture / Compare Generation A Channel capture / compare event can be generated by setting this bit. It is cleared by hardware automatically. 0: No action 1: Capture / compare event is generated on channel. If the channel is configured as an input, the counter value is captured into the CHCCR register and then the CHCCIF bit is set. If the channel is configured as an output, the CHCCIF bit is set.

Timer Interrupt Status Register – INTSR

This register stores the timer interrupt status.

Offset: 0x07C

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	22	21	20	19	18	17	16
	Reserved							
Type/Reset								
	15	14	13	12	11	10	9	8
	Reserved					TEVIF	Reserved	UEVIF
Type/Reset						RW 0		RW 0
	7	6	5	4	3	2	1	0
	Reserved			CHOCF	Reserved			CHCCIF
Type/Reset				RW 0				RW 0

Bits	Field	Descriptions
[10]	TEVIF	Trigger Event Interrupt Flag This flag is set by hardware on a trigger event and is cleared by software. 0: No trigger event occurs 1: Trigger event occurs
[8]	UEVIF	Update Event Interrupt Flag. This bit is set by hardware on an update event and is cleared by software. 0: No update event occurs 1: Update event occurs Note: The update event is derived from the following conditions: - The counter overflow - The UEVG bit is asserted - A restart trigger event occurs from the slave trigger input
[4]	CHOCF	Channel Over-Capture Flag This flag is set by hardware and cleared by software. 0: No over-capture event is detected 1: Capture event occurs again when the CHCCIF bit is already set and it is not yet cleared by software.
[0]	CHCCIF	Channel Capture / Compare Interrupt Flag - Channel is configured as an output: 0: No match event occurs 1: The contents of the counter CNTR have matched the contents of the CHCCR register This flag is set by hardware when the counter value matches the CHCCR value. It is cleared by software. - Channel is configured as an input: 0: No input capture occurs 1: Input capture occurs This bit is set by hardware on a capture event. It is cleared by software or by reading the CHCCR register.

Timer Counter Register – CNTR

This register stores the timer counter value.

Offset: 0x080

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	22	21	20	19	18	17	16
	Reserved							
Type/Reset								
	15	14	13	12	11	10	9	8
	CNTV							
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0
	7	6	5	4	3	2	1	0
	CNTV							
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0

Bits	Field	Descriptions
[15:0]	CNTV	Counter Value.

Timer Prescaler Register – PSCR

This register specifies the timer prescaler value to generate the counter clock.

Offset: 0x084

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	22	21	20	19	18	17	16
	Reserved							
Type/Reset								
	15	14	13	12	11	10	9	8
	PSCV							
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0
	7	6	5	4	3	2	1	0
	PSCV							
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0

Bits	Field	Descriptions
[15:0]	PSCV	Prescaler Value These bits are used to specify the prescaler value to generate the counter clock frequency f_{CK_CNT}. $f_{CK\_CNT} = \frac{f_{CK\_PSC}}{PSCV[15:0] + 1}$, where the f_{CK_PSC} is the prescaler clock source.

Timer Counter Reload Register – CRR

This register specifies the timer counter reload value.

Offset: 0x088

Reset value: 0x0000_FFFF

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	22	21	20	19	18	17	16
	Reserved							
Type/Reset								
	15	14	13	12	11	10	9	8
	CRV							
Type/Reset	RW 1	RW 1	RW 1	RW 1	RW 1	RW 1	RW 1	RW 1
	7	6	5	4	3	2	1	0
	CRV							
Type/Reset	RW 1	RW 1	RW 1	RW 1	RW 1	RW 1	RW 1	RW 1

Bits	Field	Descriptions
[15:0]	CRV	Counter Reload Value The CRV is the reload value which is loaded into the actual counter register.

Channel Capture / Compare Register – CHCCR

This register specifies the timer channel capture / compare value.

Offset: 0x090

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
	Reserved							
Type/Reset								
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
	CHCCV							
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
	CHCCV							
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0

Bits	Field	Descriptions
[15:0]	CHCCV	Channel Capture / Compare Value - When Channel is configured as an output The CHCCR value is compared with the counter value and the comparison result is used to trigger the CHOREF output signal. - When Channel is configured as an input The CHCCR register stores the counter value captured by the last channel capture event.

16 Real Time Clock (RTC)

Introduction

The Real Time Clock, RTC, circuitry includes the APB interface, a 32-bit up-counter, a control register, a prescaler, a compare register and a status register. Most of the RTC circuits are located in the V_{DD} Power Domain, as shown shaded in the accompanying figure, except for the APB interface. The APB interface is located in the V_{DD15} domain. Therefore, it is necessary to be isolated from the ISO signal that comes from the power control unit when the V_{DD15} domain is powered off, i.e., when the device enters the Power-Down mode. The RTC counter is used as a wakeup timer to let the system resume from the Power-Down mode. The detailed RTC function will be described in the following sections.

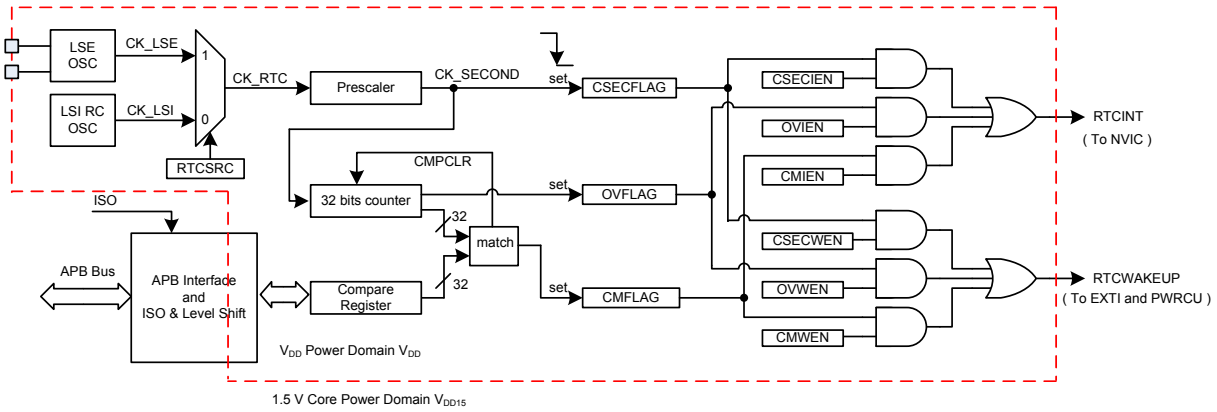


Figure 86. RTC Block Diagram

Features

- 32-bit up counter for counting elapsed time
- Programmable clock prescaler
  - Division factor: 1, 2, 4, 8…, 32768
- 32-bit compare register for alarm usage
- RTC clock source
  - LSE oscillator clock
  - LSI oscillator clock
- Three RTC Interrupt / wakeup settings
  - RTC second clock interrupt / wakeup
  - RTC compare match interrupt / wakeup
  - RTC counter overflow interrupt / wakeup
- The RTC interrupt / wakeup event can work together with power management to wake up the chip from power saving mode

Functional Descriptions

RTC Related Register Reset

The RTC registers can only be reset by either a V_{DD} Power Domain power on reset, POR, or by a V_{DD} Power Domain software reset by setting the PWRST bit in the PWRCR register. Other reset events have no effect to clear the RTC registers.

Reading RTC Register

The RTC control logic and the related registers are powered by the V_{DD} supply voltage. Therefore, the RTC circuitry remains operational in the Power-Down mode where V_{DD15} is powered off. Only the APB bus, which is located in the V_{DD15} domain, is interconnected to the circuits located in the V_{DD} domain using level shift circuitry and isolated by the ISO signals when the V_{DD15} supply voltage is powered off. The isolation function must be disabled by setting the VDDISO bit to 1 in the LPCR register as described in the Clock Control Unit before accessing the RTC registers using the APB bus.

Low Speed Clock Configuration

The default RTC clock source, CK_RTC, is derived from the LSI oscillator. The CK_RTC clock can be derived from either the external 32,768 Hz crystal oscillator, named the LSE oscillator, or the internal 32 kHz RC oscillator named the LSI oscillator, by setting the RTCSRC bit in the RTCCR register. A prescaler is provided to divide the CK_RTC by a ratio ranged from 2^0 to 2^{15} determined by the RPRE [3:0] field. For instance, setting the prescaler value RPRE [3:0] to 0x0F will generate an exact 1 Hz CK_SECOND clock if the CK_RTC clock frequency is equal to 32,768 Hz. The LSI and LSE oscillators can be enabled by the LSIEN and LSEEN control bits in the RTCCR register respectively. In addition, the LSE oscillator startup mode can be selected by configuring the LSESM bit in the RTCCR register. This enables the LSE oscillator to have either a shorter startup time or a lower power consumption, both of which are traded off depending upon specific application requirements. An example of the startup time and the power consumption for different startup modes are shown in the accompanying table for reference.

Table 36. LSE Startup Mode Operating Current and Startup Time

Startup Mode	LSESM Setting in the RTCCR Register	Operating Current	Startup Time
Normal startup	0	2.0 μA	Above 500 ms
Fast startup	1	3.5 μA	Below 300 ms

@ V_{DD} = 3.3 V and LSE clock = 32,768 Hz; these values are only for reference, actual values are dependent on the specification of the external 32.768 kHz crystal.

RTC Counter Operation

The RTC provides a 32-bit up-counter which increments at the falling edge of the CK_SECOND clock and whose value can be read from the RTCCNT register asynchronously via the APB bus. A 32-bit compare register, RTCCMP, is provided to store the specific value to be compared with the RTCCNT content. This is used to define a pre-determined time interval. When the RTCCNT register content is equal to the RTCCMP register value, the match flag CMFLAG in the RTCSR register will be set by hardware and an interrupt or wakeup event can be sent according to the corresponding enable bits in the RTCIWEN register. The RTC counter will be either reset to zero or keep counting when the compare match event occurs, dependent upon the CMPCLR bit in the RTCCR register. For example, if the RPRE [3:0] is set to 0x0F, the RTCCMP register content is set to a decimal value of 60 and the CMPCLR bit is set to 1, then the CMFLAG bit will be set every minute. In addition, the OVFLAG bit in the RTCSR register will be set when the RTC counter overflows. A read operation on the RTCSR register clears the status flags including the CSECFLAG, CMFLAG and OVFLAG bits.

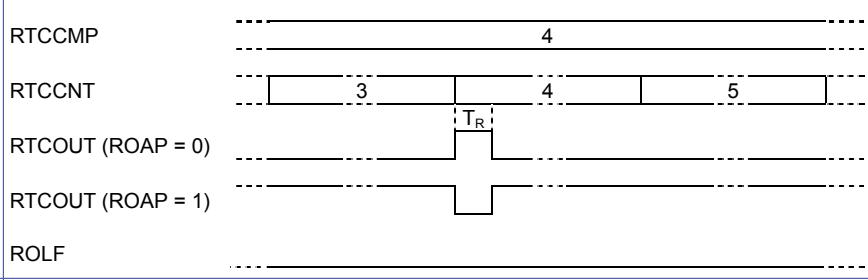
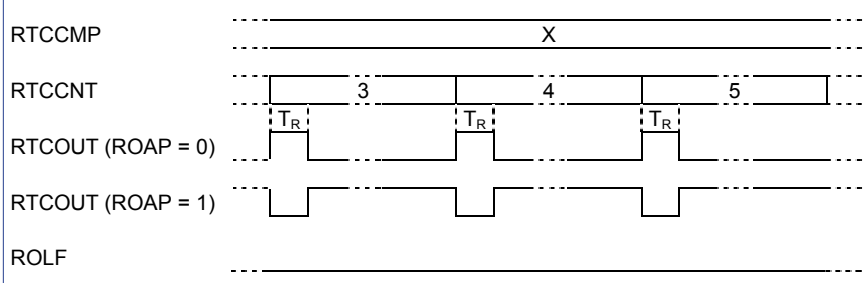
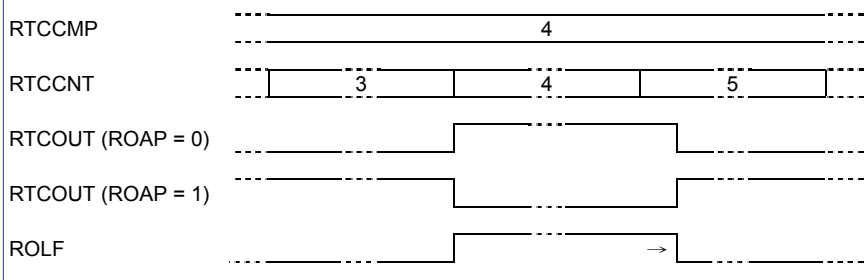
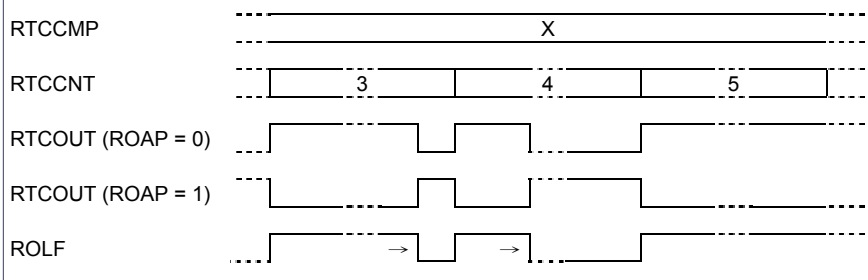
Interrupt and Wakeup Control

The falling edge of the CK_SECOND clock causes the CSECFLAG bit in the RTCSR register to be set and generates an interrupt if the corresponding interrupt enable bit, CSECIEN, in the RTCIWEN register is set. The wakeup event can also be generated to wake up the HSI / HSE oscillators, the PLL circuitry, the LDO and the CPU core if the corresponding wakeup enable bit CSECWEN is set. When the RTC counter overflows or a compare match event occurs, it will generate an interrupt or a wake up event determined by the corresponding interrupt or wakeup enable control bits, OVIEN / OVWEN or CMIEN / CMWEN bits, in the RTCIWEN register. Refer to the related register definitions for more details.

RTCOUT Output Pin Configuration

The following table shows RTCOUT output format according to the mode, polarity and event selection setting.

Table 37. RTCOUT Output Mode and Active Level Setting

ROWM	ROES	RTCOUT Output Waveform
0 (Pulse mode)	0 Compare match	
	1 Second clock	
1 (Level mode)	0 Compare match	
	1 Second clock	

T_R: RTCOUT output pulse time = 1 / f_{CK_RTC}
→: Clear by software reading ROLF bit

Register Map

The following table shows the RTC registers and reset values. Note all the registers in this unit are located at the V_{DD} power domain.

Table 38. RTC Register Map

Register	Offset	Description	Reset Value
RTCCNT	0x000	RTC Counter Register	0x0000_0000
RTCCMP	0x004	RTC Compare Register	0x0000_0000
RTCCR	0x008	RTC Control Register	0x0000_0F04
RTCSR	0x00C	RTC Status Register	0x0000_0000
RTCIWEN	0x010	RTC Interrupt and Wakeup Enable Register	0x0000_0000

Register Descriptions

RTC Counter Register – RTCCNT

This register defines a 32-bit up-counter which is incremented by the CK_SECOND clock.

Offset: 0x000

Reset value: 0x0000_0000 (Reset by V_{DD} Power Domain reset only)

	31	30	29	28	27	26	25	24
	RTCCNTV							
Type/Reset	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
	RTCCNTV							
Type/Reset	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
	RTCCNTV							
Type/Reset	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
	RTCCNTV							
Type/Reset	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0

Bits	Field	Descriptions
[31:0]	RTCCNTV	RTC Counter The current value of the RTC counter is returned when reading the RTCCNT register. The RTCCNT register is updated during the falling edge of the CK_SECOND. This register is reset by one of the following conditions: - V_{DD} Power Domain software reset – Set the PWRST bit in the PWRCR register - V_{DD} Power Domain power on reset – POR - Compare match (RTCCNT = RTCCMP) when CMPCLR = 1 (in the RTCCR register) - RTCEN bit changed from 0 to 1

RTC Compare Register – RTCCMP

This register defines a specific value to be compared with the RTC counter value.

Offset: 0x004

Reset value: 0x0000_0000 (Reset by V_{DD} Power Domain reset only)

	31	30	29	28	27	26	25	24
	RTCCMPV							
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0
	23	22	21	20	19	18	17	16
	RTCCMPV							
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0
	15	14	13	12	11	10	9	8
	RTCCMPV							
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0
	7	6	5	4	3	2	1	0
	RTCCMPV							
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0

Bits	Field	Descriptions
[31:0]	RTCCMPV	RTC Compare Match Value A match condition happens when the value in the RTCCNT register is equal to RTCCMP value. An interrupt can be generated if the CMIEN bit in the RTCIWEN register is set. When the CMPCLR bit in the RTCCR register is set to 0 and a match condition happens, the CMFLAG bit in the RTCSR register is set while the value in the RTCCNT register is not affected and will continue to count until overflow. When the CMPCLR bit is set to 1 and a match condition happens, the CMFLAG bit in the RTCSR register is set and the RTCCNT register will be reset to zero and then the counter continues to count.

RTC Control Register – RTCCR

This register specifies a range of RTC circuitry control bits.

Offset: 0x008

Reset value: 0x0000_0F04 (Reset by V_{DD} Power Domain reset only)

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	22	21	20	19	18	17	16
	Reserved			ROLF	ROAP	ROWM	ROES	ROEN
Type/Reset				RC 0	RW 0	RW 0	RW 0	RW 0
	15	14	13	12	11	10	9	8
	Reserved				RPRE			
Type/Reset					RW 1	RW 1	RW 1	RW 1
	7	6	5	4	3	2	1	0
		Reserved	LSESM	CMPCLR	LSEEN	LSIEN	RTCSRC	RTCEN
Type/Reset			RW 0	RW 0	RW 0	RW 1	RW 0	RW 0

Bits	Field	Descriptions
[20]	ROLF	RTCOUT Level Mode Flag 0: RTCOUT Output is inactive 1: RTCOUT Output is holding as active level Set by hardware when in the level mode (ROWM = 1) and an RTCOUT output event occurred. Cleared by software reading this flag. The RTCOUT signal will return to the inactive level after software has read this bit.
[19]	ROAP	RTCOUT Output Active Polarity 0: Active level is high 1: Active level is low
[18]	ROWM	RTCOUT Output Waveform Mode 0: Pulse mode The output pulse duration is one RTC clock (CK_RTC) period. 1: Level mode The RTCOUT signal will remain at an active level until the ROLF bit is cleared by software reading the ROLF bit.
[17]	ROES	RTCOUT Output Event Selection 0: RTC compare match is selected 1: RTC second clock (CK_SECOND) event is selected The ROES bit can be used to select whether the RTCOUT signal is output on the RTCOUT pin when a RTC compare match event or the RTC second clock (CK_SECOND) event occurs.
[16]	ROEN	RTCOUT Output Pin Enable 0: Disable RTCOUT output pin 1: Enable RTCOUT output pin When the ROEN bit is set to 1, the RTCOUT signal will be at an active level once a RTC compare match or the RTC second clock (CK_SECOND) event occurs. The active polarity and output waveform mode can be configured by the ROAP and ROWM bits respectively. When the ROEN bit is cleared to 0, the RTCOUT pin will be in a floating state.

Bits	Field	Descriptions
[11:8]	RPRE	RTC Clock Prescaler Select CK_SECOND = CK_RTC / 2^{RPRE} 0000: CK_SECOND = CK_RTC / 2^0 0001: CK_SECOND = CK_RTC / 2^1 0010: CK_SECOND = CK_RTC / 2^2 … 1111: CK_SECOND = CK_RTC / 2^{15}
[5]	LSESM	LSE oscillator Startup Mode 0: Normal startup and requires less operating power 1: Fast startup but requires higher operating current
[4]	CMPCLR	Compare Match Counter Clear 0: 32-bit RTC counter is not affected when compare match condition occurs 1: 32-bit RTC counter is cleared when compare match condition occurs
[3]	LSEEN	LSE oscillator Enable Control 0: LSE oscillator is disabled 1: LSE oscillator is enabled
[2]	LSIEN	LSI oscillator Enable Control 0: LSI oscillator is disabled 1: LSI oscillator is enabled The LSIEN bit default value is 1 which means the LSI oscillator is enabled automatically after the V_{DD} Power Domain is powered up. Note: After the V_{DD} power domain is powered on, the internal LSI RC oscillator will start to oscillate. The frequency range of the LSI oscillator is shown in the LSI oscillator electrical characteristics in the datasheet. The device also provides a production trim value to obtain a more accurate oscillation frequency. The procedure is to disable the LSI oscillator and then enable it again after the V_{DD} power domain is powered on. After the trimming procedure has completed, the system will automatically load the production trim value to the frequency trimming circuit of the LSI RC oscillator.
[1]	RTCSRC	RTC Clock Source Selection 0: LSI oscillator is selected as the RTC clock source 1: LSE oscillator is selected as the RTC clock source
[0]	RTCEN	RTC Enable Control 0: RTC is disabled 1: RTC is enabled

RTC Status Register – RTCSR

This register stores the counter flags.

Offset: 0x00C

Reset value: 0x0000_0000 (Reset by V_{DD} Power Domain reset and RTCEN bit change from 1 to 0)

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	22	21	20	19	18	17	16
	Reserved							
Type/Reset								
	15	14	13	12	11	10	9	8
	Reserved							
Type/Reset								
	7	6	5	4	3	2	1	0
	Reserved					OVFLAG	CMFLAG	CSECFLAG
Type/Reset						RC 0	RC 0	RC 0

Bits	Field	Descriptions
[2]	OVFLAG	Counter Overflow Flag 0: Counter overflow is not occurred since the last RTCSR register read operation 1: Counter overflow has occurred since the last RTCSR register read operation This bit is set by hardware when the counter value in the RTCCNT register changes from 0xFFFF_FFFF to 0x0000_0000 and cleared by read operation. This bit is suggested to read in the RTC IRQ handler and should be taken care when software polling is used.
[1]	CMFLAG	Compare Match Condition Flag 0: Compare match condition is not occurred since the last RTCSR register read operation 1: Compare match condition has occurred since the last RTCSR register read operation. This bit is set by hardware on the CK_SECOND clock falling edge when the RTCCNT register value is equal to the RTCCMP register content. It is cleared by software reading this bit. This bit is suggested for access in the corresponding RTC interrupt routine – do not use software polling during software free running.
[0]	CSECFLAG	CK_SECOND Occurrence Flag 0: CK_SECOND is not occurred since the last RTCSR register read operation 1: CK_SECOND has occurred since the last RTCSR register read operation This bit is set by hardware on the CK_SECOND clock falling edge. It is cleared by software reading this bit. This bit is suggested for access in the corresponding RTC interrupt routine – do not use software polling during software free running.

RTC Interrupt and Wakeup Enable Register – RTCIWEN

This register contains the interrupt and wakeup enable bits.

Offset: 0x010

Reset value: 0x0000_0000 (Reset by V_{DD} Power Domain reset only)

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	22	21	20	19	18	17	16
	Reserved							
Type/Reset								
	15	14	13	12	11	10	9	8
	Reserved					OVWEN	CMWEN	CSECWEN
Type/Reset						RW 0	RW 0	RW 0
	7	6	5	4	3	2	1	0
	Reserved					OVIEN	CMIEN	CSECIEN
Type/Reset						RW 0	RW 0	RW 0

Bits	Field	Descriptions
[10]	OVWEN	Counter Overflow Wakeup Enable 0: Counter overflow wakeup is disabled 1: Counter overflow wakeup is enabled
[9]	CMWEN	Compare Match Wakeup Enable 0: Compare match wakeup is disabled 1: Compare match wakeup is enabled
[8]	CSECWEN	Counter Clock CK_SECOND Wakeup Enable 0: Counter Clock CK_SECOND wakeup is disabled 1: Counter Clock CK_SECOND wakeup is enabled
[2]	OVIEN	Counter Overflow Interrupt Enable 0: Counter Overflow Interrupt is disabled 1: Counter Overflow Interrupt is enabled
[1]	CMIEN	Compare Match Interrupt Enable 0: Compare Match Interrupt is disabled 1: Compare Match Interrupt is enabled
[0]	CSECIEN	Counter Clock CK_SECOND Interrupt Enable 0: Counter Clock CK_SECOND Interrupt is disabled 1: Counter Clock CK_SECOND Interrupt is enabled

17 Watchdog Timer (WDT)

Introduction

The Watchdog timer is a hardware timing circuitry that can be used to detect a system lock-up due to software trapped in a deadlock. The Watchdog timer can be operated in a reset mode. The Watchdog timer will generate a reset when the counter counts down to a zero value. Therefore, the software should reload the counter value before a Watchdog timer underflow occurs. In addition, a reset is also generated if the software reloads the counter before it reaches a delta value. That means that the Watchdog timer prevents a software deadlock that continuously triggers the Watchdog, the reload must occur when the Watchdog timer value has a value within a limited window of 0 and WDTD. The Watchdog timer counter can be stopped when the processor is in the debug or sleep mode. The register write protection function can be enabled to prevent an unexpected change in the Watchdog timer configuration.

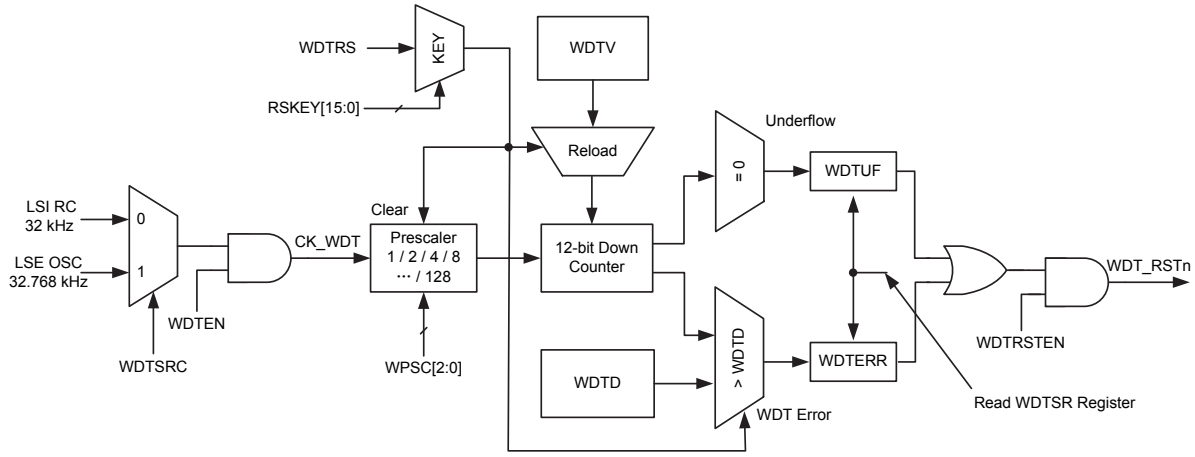


Figure 87. Watchdog Timer Block Diagram

Features

- Clock source from either internal 32 kHz RC oscillator (LSI) or external 32,768 Hz oscillator (LSE)
- Can be independently setup to keep running or to stop when entering the Sleep mode or Deep-Sleep1 mode
- 12-bit down-counter with 3-bit prescaler structure
- Provides reset to the system
- Limited reload window setup function for custom Watchdog timer reload times
- Watchdog Timer may be stopped when the processor is in the debug
- Reload lock key to prevent unexpected operation
- Configuration register write protection function for counter value, reset enable, delta value and prescaler value

Functional Description

The Watchdog timer is formed from a 12-bit count-down counter and a fixed 3-bit prescaler. The largest time-out period is 16 seconds, using the LSE or LSI clock and a 1/128 maximum prescaler value.

The Watchdog timer configuration setup includes programmable counter reload value, reset enable, window value and prescaler value. These configurations are set using the WDTMR0 and WDTMR1 registers which must be properly programmed before the Watchdog timer starts counting. In order to prevent unexpected write operations to those configurations, a register write protection function can be enabled by writing any value, other than 0x35CA to PROTECT[15:0], in the WDTPR register. A value of 0x35CA can be written to PROTECT[15:0] to disable the register write protection function before accessing any configuration register. A read operation on PROTECT[0] can obtain the enable / disable status of the register write protection function.

During normal operation, the Watchdog timer counter should be reloaded before it underflows to prevent the generation of a Watchdog reset. The 12-bit count-down counter can be reloaded with the required Watchdog Timer Counter Value (WDTV) by first setting the WDTRS bit to 1 with the correct key, which is 0x5FA0 in the WDTCR register.

If a software deadlock occurs during a Watchdog timer reload routine, the reload operation will still go ahead and therefore the software deadlock cannot be detected. To prevent this situation from occurring, the reload operation must be executed in such a way that the value of the Watchdog timer counter is limited within a delta value (WDTD). If the Watchdog timer counter value is greater than the delta value and a reload operation is executed, a Watchdog Timer error will occur. The Watchdog timer error will generate a Watchdog reset if the related functional control is enabled. Additionally, the above features can be disabled by programming a WDTD value greater than or equal to the WDTV value.

The WDTERR and WDTUF flags in the WDTSR register will be set respectively when the Watchdog timer underflows or when a Watchdog timer error occurs. A system reset or write-one operation on the WDTSR register clears the WDTERR and WDTUF flags.

The watchdog timer uses two clocks: PCLK and CK_WDT. The PCLK clock is used for APB access to the watchdog registers. The CK_WDT clock is used for the Watchdog timer functionality and counting. There is some synchronization logic between these two clock domains.

When the system enters the Sleep mode or Deep-Sleep1 mode, the Watchdog timer counter will either continue to count or stop depending on the WDTSHLT bits in the WDTMR0 register. However, the Watchdog Timer will always stop when the system is in the Deep-Sleep2 mode. When the Watchdog stops counting, the count value is retained so that it continues counting after the system is woken up from these three Sleep modes. A Watchdog reset will occur any time when the Watchdog timer is running and when it has an operating clock source. When the system enters the debug mode, the Watchdog timer counter will either continue to count or stop depending on the DBWDT bit (in the MCUDBGCR register) in the Clock Control Unit.

The Watchdog timer should be used in the following manners:

- Set the Watchdog timer reload value (WDTV) and reset in the WDTMR0 register.
- Set the Watchdog timer delta value (WDTD) and prescaler in the WDTMR1 register.
- Start the Watchdog timer by writing to the WDTCR register with WDTRS = 1 and RSKEY = 0x5FA0.
- Write to the WDTPR register to lock all the Watchdog timer registers except for WDTCR and WDTPR.
- The Watchdog timer counter should be reloaded again within the delta value (WDTD).

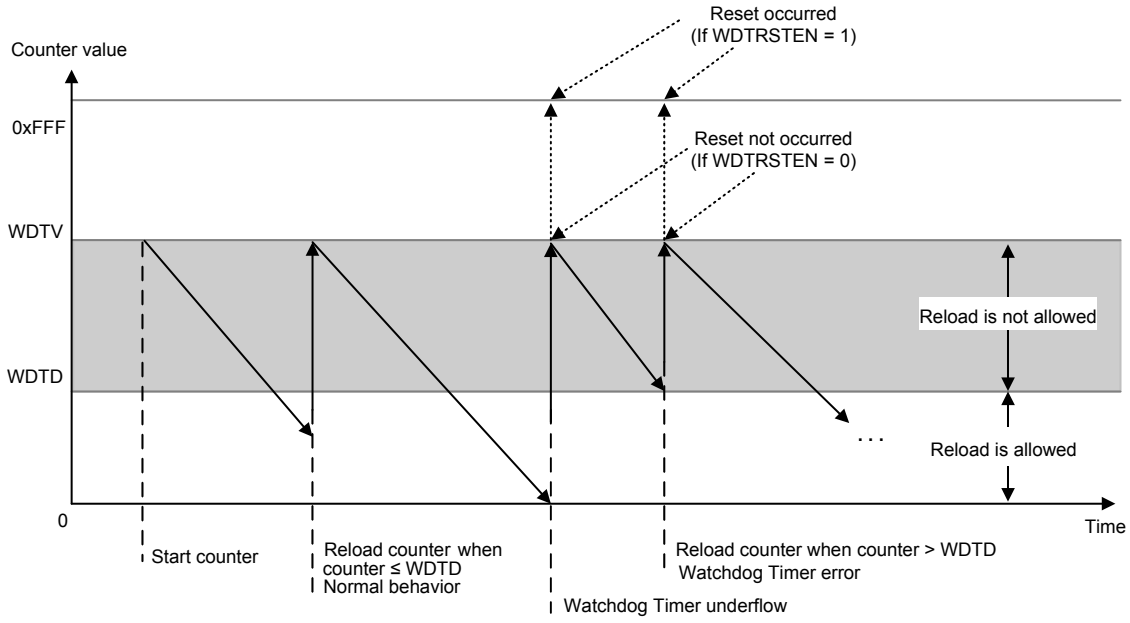


Figure 88. Watchdog Timer Behavior

Register Map

The following table shows the Watchdog Timer registers and reset values.

Table 39. Watchdog Timer Register Map

Register	Offset	Description	Reset Value
WDTCR	0x000	Watchdog Timer Control Register	0x0000_0000
WDTMR0	0x004	Watchdog Timer Mode Register 0	0x0000_0FFF
WDTMR1	0x008	Watchdog Timer Mode Register 1	0x0000_7FFF
WDTSR	0x00C	Watchdog Timer Status Register	0x0000_0000
WDTPR	0x010	Watchdog Timer Protection Register	0x0000_0000
WDTCSR	0x018	Watchdog Timer Clock Selection Register	0x0000_0000

Register Descriptions

Watchdog Timer Control Register – WDTCR

This register is used to reload the Watchdog timer.

Offset: 0x000

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	RSKEY							
Type/Reset	WO 0	WO 0	WO 0	WO 0	WO 0	WO 0	WO 0	WO 0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
	RSKEY							
Type/Reset	WO 0	WO 0	WO 0	WO 0	WO 0	WO 0	WO 0	WO 0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
	Reserved							
Type/Reset								
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
	Reserved							WDTRS
Type/Reset								WO 0

Bits	Field	Descriptions
[31:16]	RSKEY	Watchdog Timer Reload Lock Key The RSKEY [15:0] bits should be written with a 0x5FA0 value to enable the WDT reload operation function. Writing any other value except 0x5FA0 in this field will abort the write operation.
[0]	WDTRS	Watchdog Timer Reload 0: No effect 1: Reload Watchdog Timer This bit is used to reload the Watchdog timer counter as a WDTV value which is stored in the WDTMR0 register. It is set to 1 by software and cleared to 0 by hardware automatically.

Watchdog Timer Mode Register 0 – WDTMR0

This register specifies the Watchdog timer counter reload value and reset enable control.

Offset: 0x004

Reset value: 0x0000_0FFF

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	22	21	20	19	18	17	16
	Reserved							WDTEN
Type/Reset								RW 0
	15	14	13	12	11	10	9	8
		WDTSHLT	WDTRSTEN	Reserved	WDTV			
Type/Reset	RW 0	RW 0	RW 0		RW 1	RW 1	RW 1	RW 1
	7	6	5	4	3	2	1	0
	WDTV							
Type/Reset	RW 1	RW 1	RW 1	RW 1	RW 1	RW 1	RW 1	RW 1

Bits	Field	Descriptions
[16]	WDTEN	Watchdog Timer Running Enable 0: Watchdog timer is disabled 1: Watchdog timer is enabled to run When the Watchdog timer is disabled, the counter will be reset to its hardware default condition. When the WDTEN bit is set, the Watchdog timer will be reloaded with the WDTV value and count down.
[15:14]	WDTSHLT	Watchdog Timer Sleep Halt 00: The Watchdog runs when the system is in the Sleep mode or Deep-Sleep1 mode 01: The Watchdog runs when the system is in the Sleep mode and halts in Deep-Sleep1 mode 10 or 11: The Watchdog halts when the system is in the Sleep mode and Deep-Sleep1 mode Note that the Watchdog timer always halts when the system is in Deep-Sleep2 mode. The Watchdog stops counting when the WDTSHLT bits are properly configured in the Sleep mode or Deep-Sleep1 mode, the count value is retained so that it continues counting after the system wakes up from these three Sleep modes. If a Watchdog reset occurs in Sleep mode or Deep-Sleep1 mode, it will wake up the device.
[13]	WDTRSTEN	Watchdog Timer Reset Enable 0: A Watchdog Timer underflow or error has no effect on the reset of system 1: A Watchdog Timer underflow or error triggers a Watchdog timer system reset
[11:0]	WDTV	Watchdog Timer Counter Value WDTV defines the value loaded into the 12-bit Watchdog down-counter.

Watchdog Timer Mode Register 1 – WDTMR1

This register specifies the Watchdog delta value and the prescaler selection.

Offset: 0x008

Reset value: 0x0000_7FFF

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	22	21	20	19	18	17	16
	Reserved							
Type/Reset								
	15	14	13	12	11	10	9	8
	Reserved	WPSC			WDTD			
Type/Reset		RW 1	RW 1	RW 1	RW 1	RW 1	RW 1	RW 1
	7	6	5	4	3	2	1	0
	WDTD							
Type/Reset	RW 1	RW 1	RW 1	RW 1	RW 1	RW 1	RW 1	RW 1

Bits	Field	Descriptions
[14:12]	WPSC	Watchdog Timer Prescaler Selection 000: 1 / 1 001: 1 / 2 010: 1 / 4 011: 1 / 8 100: 1 / 16 101: 1 / 32 110: 1 / 64 111: 1 / 128
[11:0]	WDTD	Watchdog Timer Delta Value Define the permitted range to reload the Watchdog timer. If the Watchdog timer counter value is less than or equal to WDTD, writing to the WDTCR register with WDTRS = 1 and RSKEY = 0x5FA0 will reload the timer. If the Watchdog Timer value is greater than WDTD, then writing WDTCR with WDTRS = 1 and RSKEY = 0x5FA0 will cause a Watchdog Timer error. This feature can be disabled by programming a WDTD value greater then or equal to the WDTV value.

Watchdog Timer Status Register – WDTSR

This register specifies the Watchdog timer status.

Offset: 0x00C

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	22	21	20	19	18	17	16
	Reserved							
Type/Reset								
	15	14	13	12	11	10	9	8
	Reserved							
Type/Reset								
	7	6	5	4	3	2	1	0
	Reserved						WDTERR	WDTUF
Type/Reset							WC 0	WC 0

Bits	Field	Descriptions
[1]	WDTERR	Watchdog Timer Error 0: No Watchdog timer error has occurred since the last read of this register 1: A Watchdog timer error has occurred since the last read of this register Note: A reload operation when the Watchdog timer counter value is larger than WDTD causes a Watchdog timer error. Note that this bit is a write-one-clear flag.
[0]	WDTUF	Watchdog timer Underflow 0: No Watchdog timer underflow has occurred since the last read of this register 1: A Watchdog timer underflow has occurred since the last read of this register Note that this bit is a write-one-clear flag.

Watchdog Timer Protection Register – WDTPR

This register specifies the Watchdog timer protect key configuration.

Offset: 0x010

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	22	21	20	19	18	17	16
	Reserved							
Type/Reset								
	15	14	13	12	11	10	9	8
	PROTECT							
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0
	7	6	5	4	3	2	1	0
	PROTECT							
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0

Bits	Field	Descriptions
[15:0]	PROTECT	Watchdog Timer Register Protection For write operation: 0x35CA: Disable the Watchdog timer register write protection Others: Enable the Watchdog timer register write protection For read operation: 0x0000: Watchdog timer register write protection is disabled 0x0001: Watchdog timer register write protection is enabled This register is used to enable / disable the Watchdog timer configuration register write protection function. All configuration registers become read only except for WDTCR and WDTPR when the register write protection is enabled. Additionally, the read operation of PROTECT[0] can obtain the enable / disable status of the register write protection function.

Watchdog Timer Clock Selection Register – WDTCSR

This register specifies the Watchdog timer clock source selection and lock configuration.

Offset: 0x018

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	22	21	20	19	18	17	16
	Reserved							
Type/Reset								
	15	14	13	12	11	10	9	8
	Reserved							
Type/Reset								
	7	6	5	4	3	2	1	0
	Reserved			WDTLOCK	Reserved			WDTSRC
Type/Reset				RW 0				RW 0

Bits	Field	Descriptions
[4]	WDTLOCK	Watchdog Timer Lock Mode 0: This bit is only set to 0 on any reset. It cannot be cleared by software 1: This bit is set once only by software and locks the Watchdog timer function Software can set this bit to 1 at any time. Once the WDTLOCK bit is set, the function and registers of the Watchdog timer cannot be modified or disabled, including the Watchdog timer clock source, and only waits for a system reset to disable the lock mode.
[0]	WDTSRC	Watchdog Timer Clock Source Selection 0: Internal 32 kHz RC oscillator clock selected (LSI) 1: External 32.768 kHz crystal oscillator clock selected (LSE) Select using software to control the Watchdog timer clock source.

18 Inter-Integrated Circuit (I²C)

Introduction

The I²C Module is an internal circuit allowing communication with an external I²C interface which is an industry standard two-wire serial interface used for connection to external hardware. These two serial lines are known as a serial data line, SDA, and a serial clock line, SCL. The I²C module provides three data transfer rates: (1) 100 kHz in the Standard mode, (2) 400 kHz in the Fast mode and (3) 1MHz in the Fast mode plus. The SCL period generation registers are used to setup different kinds of duty cycle implementation for the SCL pulse.

The SDA line which is connected to the whole I²C bus is a bidirectional data line between the master and slave devices used for the transmission and reception of data. The I²C module also has an arbitration detection function to prevent the situation where more than one master attempts to transmit data on the I²C bus at the same time.

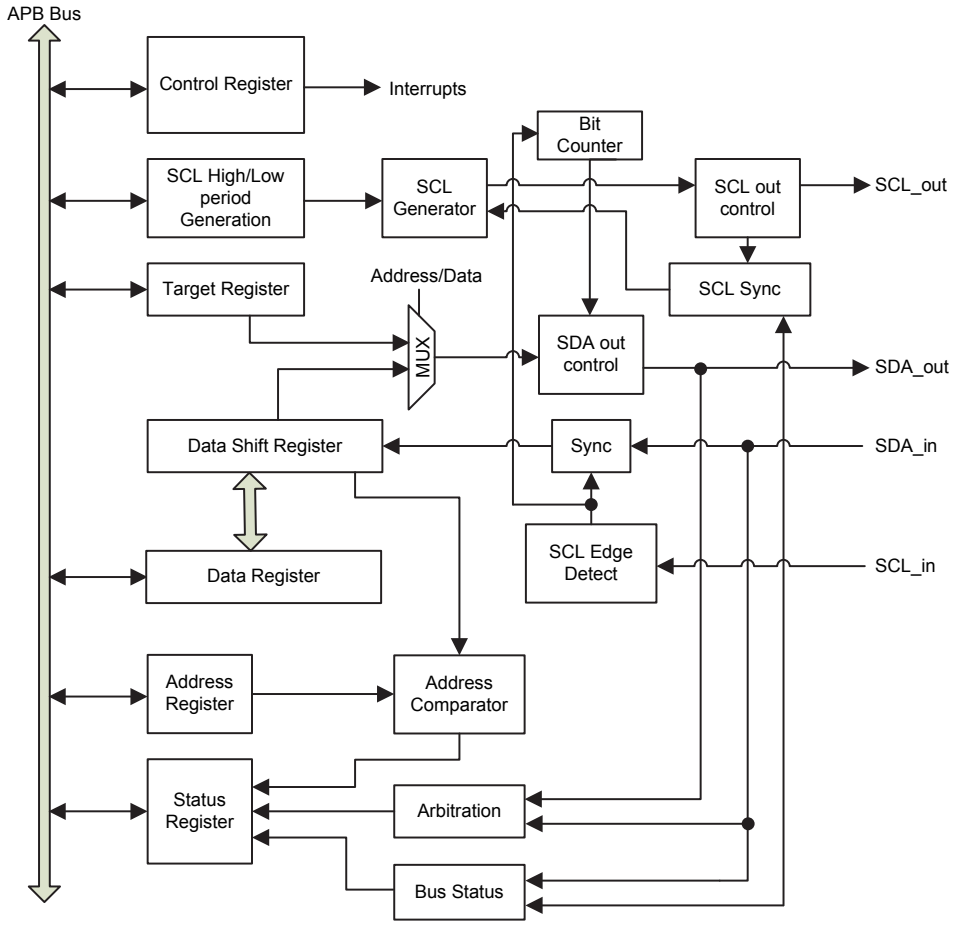


Figure 89. I²C Module Block Diagram

Features

- Two-wire I²C serial interface
  - Serial data line (SDA) and serial clock (SCL)
- Multiple speed modes
  - Standard mode – 100 kHz
  - Fast mode – 400 kHz
  - Fast mode plus – 1 MHz
- Bidirectional data transfer between master and slave
- Multi-master bus – no central master
  - The same interface can act as Master or Slave
- Arbitration among simultaneously transmitting masters without corrupting serial data on the bus
- Clock synchronization
  - Allow devices with different bit rates to communicate via one serial bus
- Supports 7-bit and 10-bit addressing mode and general call addressing.
- Multiple slave addresses using address mask function
- Timeout function
- Supports PDMA Interface

Functional Descriptions

Two-Wire Serial Interface

The I²C module has two external lines, the serial data SDA and serial clock SCL lines, to carry information between the interconnected devices connected to the bus. The SCL and SDA lines are both bidirectional and must be connected to a pull-high resistor. When the I²C bus is in the free or idle state, both pins are at a high level to perform the required wired-AND function for multiple connected devices.

START and STOP Conditions

A master device can initialize a transfer by sending a START signal and terminate the transfer with a STOP signal. A START signal is usually referred to as the “S” bit, which is defined as a High to Low transition on the SDA line while the SCL line is high. A STOP signal is usually referred to as the “P” bit, which is defined as a Low to High transition on the SDA line while SCL is high.

A repeated START signal, which is denoted as the “Sr” bit, is functionally identical to the normal START condition. A repeated START signal allows the I²C interface to communicate with another slave device or with the same device but in a different transfer direction without releasing the I²C bus control.

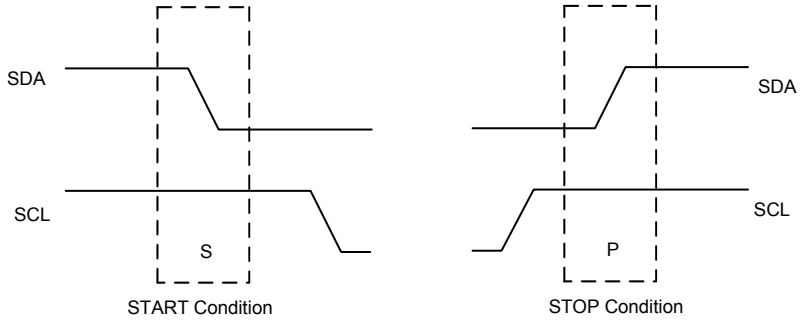


Figure 90. START and STOP Condition

Data Validity

The data on the SDA line must be stable during the high period of the SCL clock. The SDA data state can only be changed when the clock signal on the SCL line is in a low state.

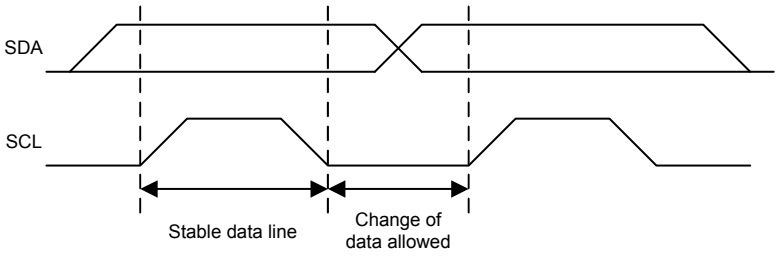


Figure 91. Data Validity

Addressing Format

The I²C interface starts to transfer data after the master device has sent the address to confirm the targeted slave device. The address frame is sent just after the START signal by the master device. The addressing mode selection bit named ADRM in the I2CCR register should be defined to choose either the 7-bit or 10-bit addressing mode.

7-bit Address Format

The 7-bit address format is composed of the 7-bit length slave address, which the master device wants to communicate, with a R/$\overline{W}$ bit and an ACK bit. The R/$\overline{W}$ bit defines the direction of the data transfer.

R/$\overline{W}$ = 0 (Write): The master transmits data to the addressed slave.

R/$\overline{W}$ = 1 (Read): The master receives data from the addressed slave.

The slave address can be assigned through the ADDR field in the I2CADDR register. The slave device sends back the acknowledge bit (ACK) if its slave address matches the transmitted address sent by master.

Note that it is forbidden to own the same address for two slave devices.

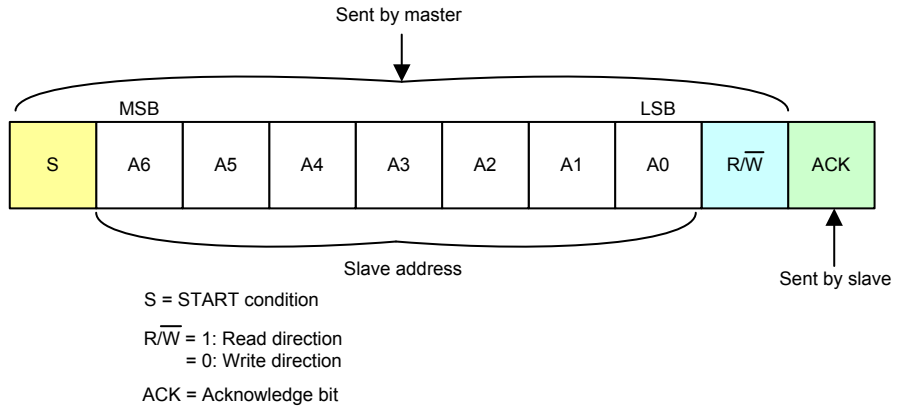


Figure 92. 7-bit Addressing Mode

10-bit Address Format

In order to prevent address clashes, due to the limited range of the 7-bit addresses, a new 10-bit address scheme has been introduced. This enhancement can be mixed with the 7-bit addressing mode which increases the available address range about ten times. For the 10-bit addressing mode, the first two bytes after a START signal include a header byte and an address byte that usually determines which slave will be selected by the master. The header byte is composed of a leading "11110", the 10th and 9th bits of the slave address. The second byte is the remaining 8 bits of the slave device address.

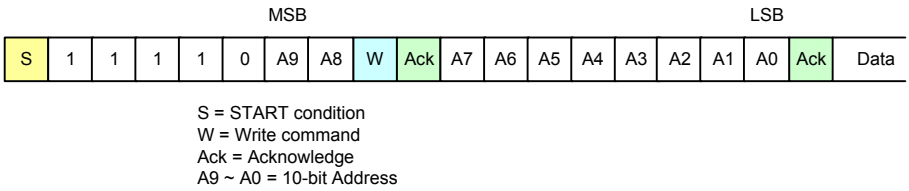


Figure 93. 10-bit Addressing Write Transmit Mode

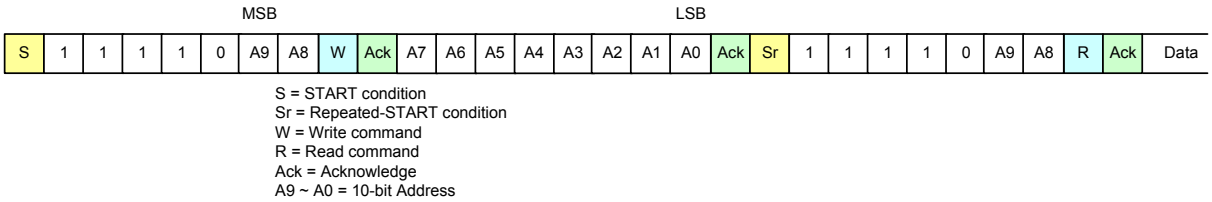


Figure 94. 10-bits Addressing Read Receive Mode

Data Transfer and Acknowledge

Once the slave device address has been matched, the data can be transmitted to or received from the slave device according to the transfer direction specified by the R/$\overline{W}$ bit. Each byte is followed by an acknowledge bit on the 9th SCL clock.

If the slave device returns a Not Acknowledge (NACK) signal to the master device, the master device can generate a STOP signal to terminate the data transfer or generate a repeated START signal to restart the transfer.

If the master device sends a Not Acknowledge (NACK) signal to the slave device, the slave device should release the SDA line for the master device to generate a STOP signal to terminate the transfer.

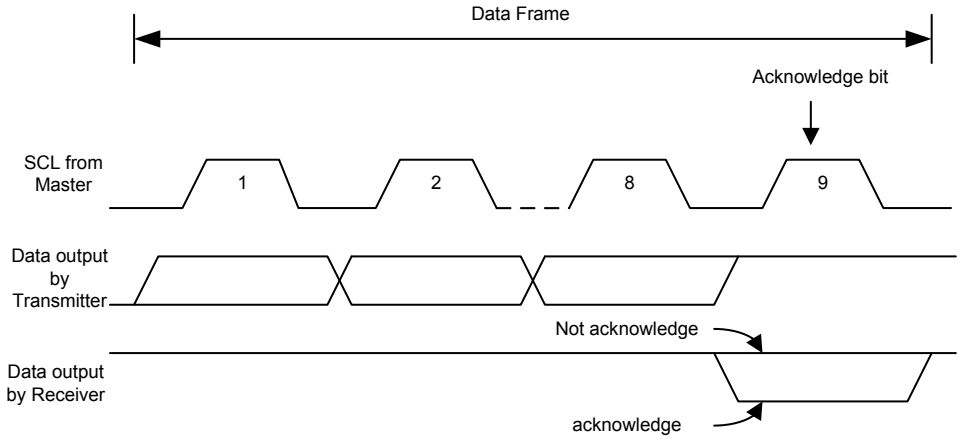


Figure 95. I²C Bus Acknowledge

Clock Synchronization

Only one master device can generate the SCL clock under normal operation. However when there is more than one master trying to generate the SCL clock, the clock should be synchronized so that the data output can be compared. Clock synchronization is performed using the wired-AND connection of the I²C interface to the SCL line.

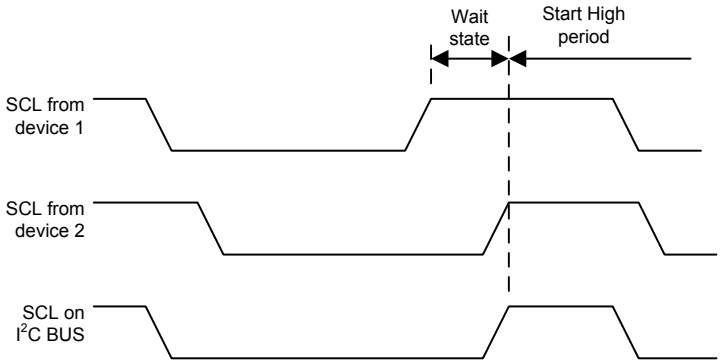


Figure 96. Clock Synchronization during Arbitration

Arbitration

A master may start a transfer only if the I²C bus line is in the free or idle mode. If two or more masters generate a START signal at approximately the same time, an arbitration procedure will occur.

Arbitration takes place on the SDA line and can continue for many bits. The arbitration procedure gives a higher priority to the device that transmits serial data with a binary low bit (logic low). Other master devices which want to transmit binary high bits (logic high) will lose the arbitration. As soon as a master loses the arbitration, the I²C module will set the ARBLOS bit in the I2CSR register and generate an interrupt if the interrupt enable bit, ARBLOSIE, in the I2CIER register is set to 1. Meanwhile, it stops sending data and listens to the bus in order to detect an I²C stop signal. When the stop signal is detected, the master which has lost the arbitration may try to access the bus again.

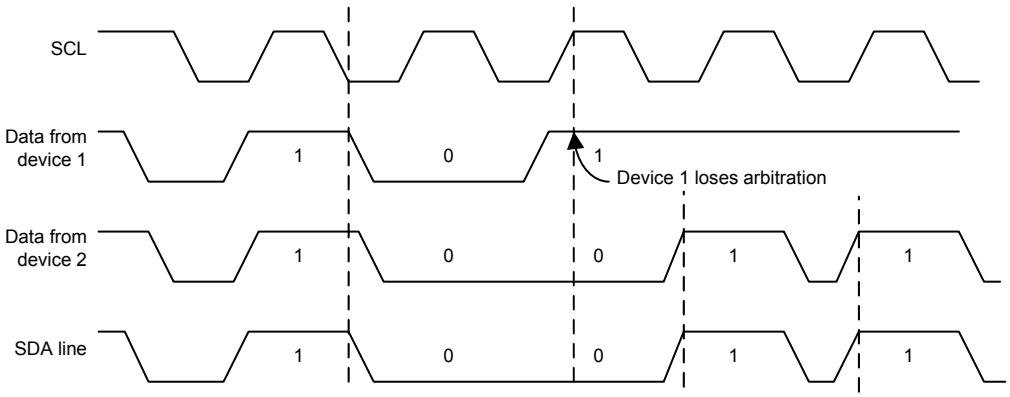


Figure 97. Two Master Arbitration Procedure

General Call Addressing

The general call addressing function can be used to address all the devices connected to the I²C bus. The master device can activate the general call function by writing a value "00" into the TAR field and setting the RWD bit to 0 in the I2CTAR register on the addressing frame.

The device can support the general call addressing function by setting the corresponding enable control bit GCEN to 1. If the GCEN bit is set to 1 to support the general call addressing, the AA bit in the I2CCR register should also be set to 1 to send an acknowledge signal back when the device receives an address frame with a value of 00H. When this condition occurs, the general call flag, GCS, will be set to 1, but the ADRS flag will not be set.

Bus Error

If an unpredictable START or STOP condition occurs when the data is being transferred on the I²C bus, it will be considered as a bus error and the transferring data will be aborted. When a bus error event occurs, the relevant bus error flag BUSERR in the I2CSR register will set to 1 and both the SDA and SCL lines are released. The BUSERR flag should be cleared by writing a 1 to it to initiate the I²C module to an idle state.

Address Mask Enable

The I²C module provides an address mask function for users to decide which address bit can be ignored during the comparison with the address frame sent from the master. The ADRS flag will be asserted when the unmasked address bits and the address frame sent from the master are matched. Note that this function is only available in the slave mode.

For instance, the user sets a data transfer with the 7-bit addressing mode together with the I2CADDMR register value as 0x05h and the I2CADDR register value as 0x55h, this means if an address which is sent by an I²C master on the bus is equal to 0x50h, 0x51h, 0x54h or 0x55h, the I²C slave address will all be considered to be matched and the ADRS flag in the I2CSR register will be asserted after the address frame.

Address Snoop

The Address Snoop register, I2CADDSR, is used to monitor the calling address on the I²C bus during the whole data transfer operation no matter if the I²C module operates as a master or a slave device. Note that the I2CADDSR register is a read only register and each calling address on the I²C bus will be stored in the I2CADDSR register automatically even if the I²C device is not addressed.

Operation Mode

The I²C module can operate in the following modes:

- Master Transmitter
- Master Receiver
- Slave Transmitter
- Slave Receiver

The I²C module operates in the slave mode by default. The interface will switch to the master mode automatically after generating a START signal.

Master Transmitter Mode

Start Condition

Users write the target slave device address and communication direction into the I2CTAR register after setting the I2CEN bit in the I2CCR register. The STA flag in the I2CSR register is set by hardware after a start condition occurs. In order to send the following address frame, the STA flag must be cleared to 0 if it has been set to 1. The STA flag is cleared by reading the I2CSR register.

Address Frame

The ADRS flag in the I2CSR register will be set after the address frame is sent by the master device and the acknowledge signal from the address matched slave device is received. In order to send the following data frame, the ADRS flag must be cleared to 0 if it has been set to 1. The ADRS bit is cleared by reading the I2CSR register.

Data Frame

The data to be transmitted to the slave device must be transferred to the I2CDR register.

The TXDE bit in the I2CSR register is set to indicate that the I2CDR register is empty, which results in the SCL line being held at a logic low state. New data must then be transferred to the I2CDR register to continue the data transfer process. Writing a data into the I2CDR register will clear the TXDE flag.

Close / Continue Transmission

After transmitting the last data byte, the STOP bit in the I2CCR register can be set to terminate the transmission or re-assign another slave device by configuring the I2CTAR register to restart a new transfer.

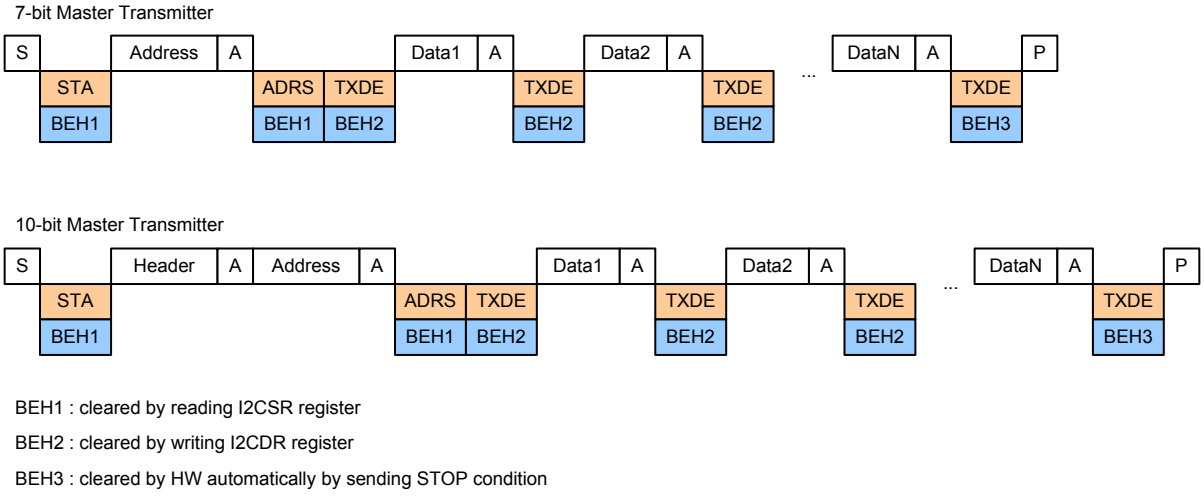


Figure 98. Master Transmitter Timing Diagram

Master Receiver Mode

Start Condition

The target slave device address and communication direction must be written into the I2CTAR register. The STA flag in the I2CSR register is set by hardware after a start condition occurs. In order to send the following address frame, the STA flag must be cleared to 0 if it has been set to 1. The STA flag is cleared by reading the I2CSR register.

Address Frame

In the 7-bit addressing mode: The ADRS flag is set after the address frame is sent by the master device and the acknowledge signal from the address matched slave device is received. In order to receive the following data frame, the ADRS bit must be cleared to 0 if it has been set to 1. The ADRS bit is cleared after reading the I2CSR register.

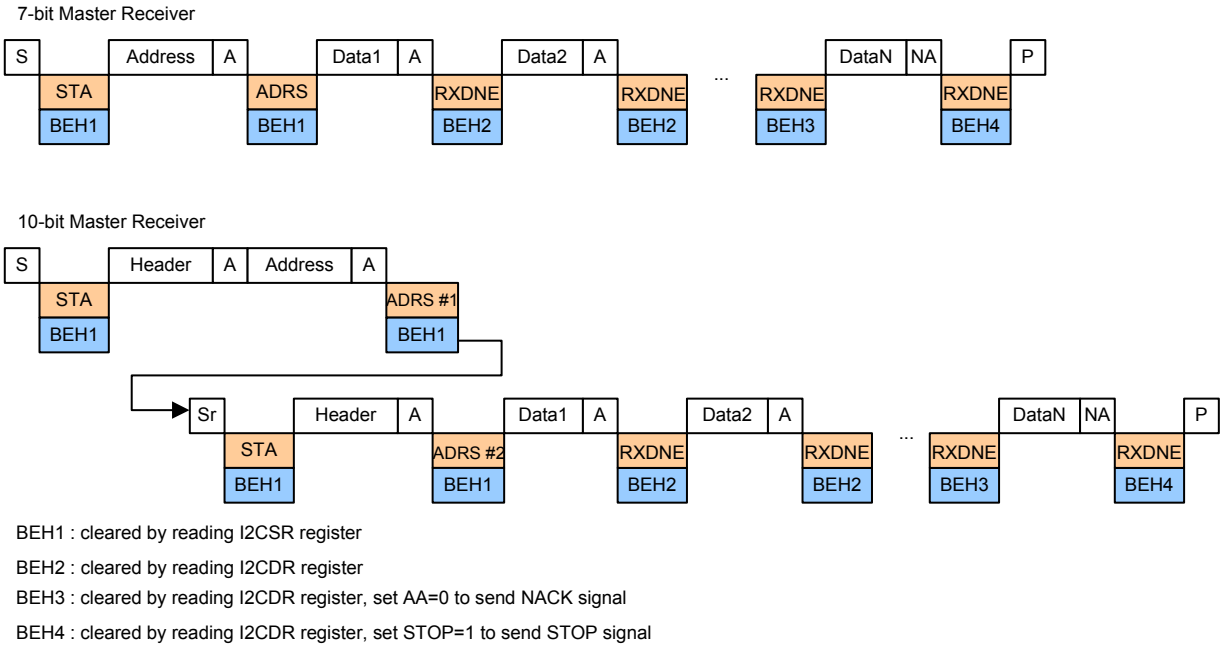
In the 10-bit addressing mode: The ADRS bit in the I2CSR register will be set twice in the 10-bit addressing mode. The first time the ADRS bit is set is when the 10-bit address is sent and the acknowledge signal from the slave device is received. The second time the ADRS bit is set is when the header byte is sent and the slave acknowledge signal is received. In order to receive the following data frame, the ADRS bit must be cleared to 0 if it has been set to 1. The ADRS bit is cleared after reading the I2CSR register. The detailed master receiver mode timing diagram is shown in the following figure.

Data Frame

In the master receiver mode, data is transmitted from the slave device. Once a data is received by the master device, the RXDNE flag in the I2CSR register is set but it will not hold the SCL line. However, if the device receives a complete new data byte and the RXDNE flag has already been set to 1, the RXBF bit in the I2CSR register will be set to 1 and the SCL line will be held at a logic low state. When this situation occurs, data from the I2CDR register should be read to continue the data transfer process. The RXDNE flag can be cleared after reading the I2CDR register.

Close / Continue Transmission

The master device needs to reset the AA bit in the I2CCR register to send a NACK signal to the slave device before the last data byte transfer has been completed. After the last data byte has been received from the slave device, the master device will hold the SCL line at a logic low state following after a NACK signal sent by the master device to the slave device. The STOP bit can be set to terminate the data transfer process or re-assign the I2CTAR register to restart a new transfer.



BEH1 : cleared by reading I2CSR register

BEH2 : cleared by reading I2CDR register

BEH3 : cleared by reading I2CDR register, set AA=0 to send NACK signal

BEH4 : cleared by reading I2CDR register, set STOP=1 to send STOP signal

Figure 99. Master Receiver Timing Diagram

Slave Transmitter Mode

Address Frame

In the 7-bit addressing mode, the ADRS bit in the I2CSR register is set after the slave device receives the calling address which matches with the slave device address. In the 10-bit addressing mode, the ADRS bit is set for the first time when the first header byte and the second address byte are both matched. Note that when the second header byte is also matched, the ADRS bit will be set again. After the ADRS bit has been set to 1, it must be cleared to 0 to continue the data transfer process. The ADRS bit is cleared after reading the I2CSR register.

Data Frame

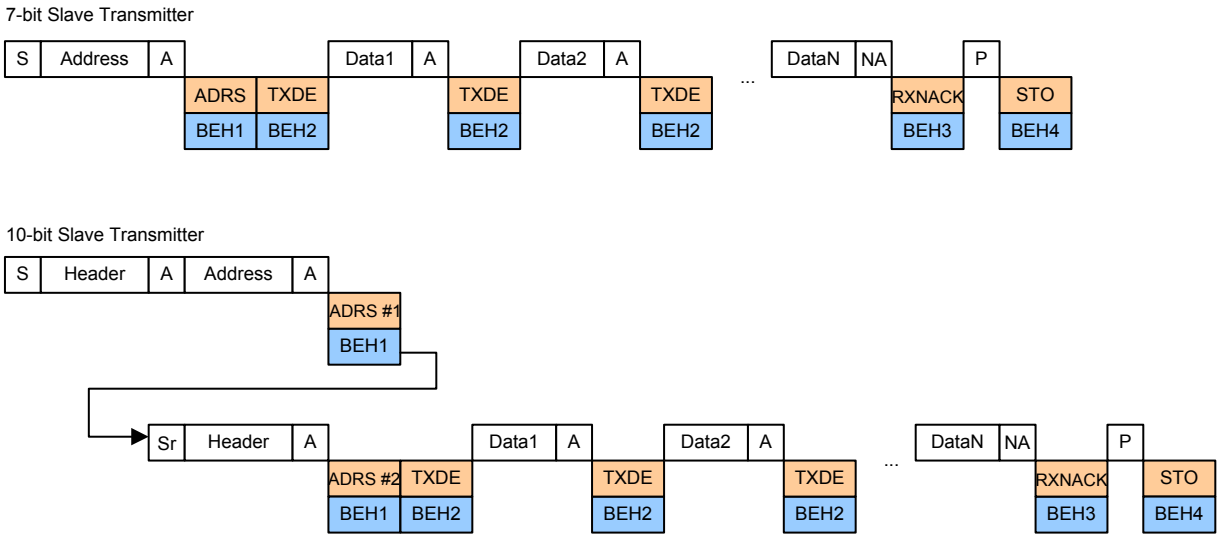
In the Slave transmitter mode, the TXDE bit is set to indicate that the I2CDR is empty, which results in the SCL line being held at a logic low state. New transmission data must then be written into the I2CDR register to continue the data transfer process. Writing a data into the I2CDR register will clear the TXDE bit.

Receive Not-Acknowledge

When the slave device receives a Not-Acknowledge signal, the RXNACK bit in the I2CSR Register is set but it will not hold the SCL line. Writing "1" to RXNACK will clear the RXNACK flag.

STOP Condition

When the slave device detects a STOP condition, the STO bit in the I2CSR register is set to indicate that the I^2C interface transmission is terminated. Reading the I2CSR register can clear the STO flag.



BEH1 : cleared by reading I2CSR register

BEH2 : cleared by writing I2CDR register

BEH3 : cleared by writing 1 clear for RXNACK flag, TXDE is not set when NACK is received.

BEH4 : cleared by reading I2CSR register

Figure 100. Slave Transmitter Timing Diagram

Slave Receiver Mode

Address Frame

The ADRS bit in the I2CSR register is set after the slave device receives the calling address which matches with the slave device address. After the ADRS bit has been set to 1, it must be cleared to 0 to continue the data transfer process. The ADRS flag is cleared after reading the I2CSR register.

Data Frame

In the slave receiver mode, the data is transmitted from the master device. Once a data byte is received by the slave device, the RXDNE flag in the I2CSR register is set but it will not hold the SCL line. However, if the device receives a complete new data byte and the RXDNE bit has been set to 1, the RXBF bit in the I2CSR register will be set to 1 and the SCL line will be held at a logic low state. When this situation occurs, data from the I2CDR register should be read to continue the data transfer process. The RXDNE flag bit can be cleared after reading the I2CDR register.

STOP Condition

When the slave device detects a STOP condition, the STO flag bit in the I2CSR register is set to indicate that the I^2C interface transmission is terminated. Reading the I2CSR register can clear the STO flag bit.

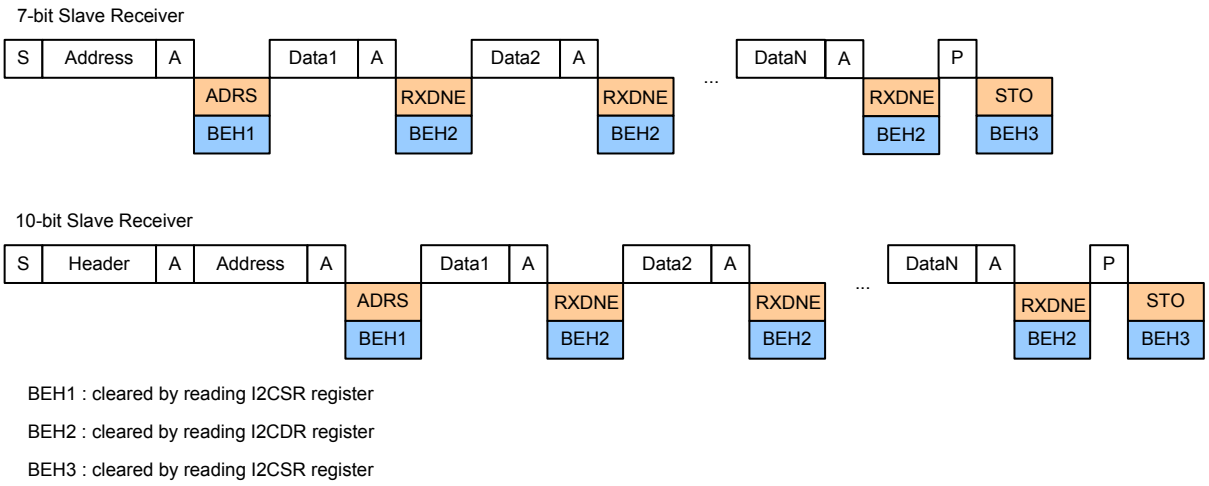


Figure 101. Slave Receiver Timing Diagram

Conditions of Holding SCL Line

The following conditions will cause the SCL line to be held at a logic low state by hardware resulting in all the I²C transfers being stopped. Data transfer will be continued after the creating conditions are eliminated.

Table 40. Conditions of Holding SCL line

Type	Condition	Description	Eliminated
Flag	TXDE	I²C is used in transmitted mode and I2CDR register needs to have data to transmit. (Note: TXDE won't be assert after receiving a NACK)	Master case: Writing data to I2CDR register Set TAR Set STOP Slave case: Writing data to I2CDR register
	GCS	I²C is addressed as slave through general call	Reading I2CSR register
	ADRS	Master: I²C is sent over address frame and is returned an ACK from slave (Note: Reference Fig.98 and Fig.99) Slave: I²C is addressed as slave device (Note: Reference Fig.100 and Fig.101)	Reading I2CSR register
	STA	Master sends a START signal	Reading I2CSR register
	RXBF	Received a complete new data and meanwhile the RXDNE flag has been set already before.	Reading I2CDR register
Event	Master receives NACK	No matter in address or data frame, once received a NACK signal will hold SCL line in master mode.	Set TAR Set STOP
	Master sends NACK used in receive mode	Occurred when receiving the last data byte in Master receive mode (Note: Reference Fig.99, and RXNACK flag won't be asserted in this case)	Set TAR Set STOP

I²C Timeout Function

In order to reduce the occurrence of I²C lockup problem due to the reception of erroneous clock source, a timeout function is provided. If the I²C bus clock source is not received for a certain timeout period, then a corresponding I²C timeout flag will be asserted. This timeout period is determined by a 16-bit down-counting counter with a programmable preload value. The timeout counter is driven by the I²C timeout clock, f_{I2CTO}, which is specified by the timeout prescaler field in the I2CTOUT register. The TOUT field in the I2CTOUT register is used to define the timeout counter preload value. The timeout function is enabled by setting the ENTOUT bit in the I2CCR register. The timeout counter will start to count down from the preloaded value if the ENTOUT bit is set to 1 and one of the following conditions occurs:

- The I²C master module sends a START signal.
- The I²C slave module detects a START signal.
- The RXBF, TXDE, RXDNE, RXNACK, GCS or ADRS flags is asserted.

The timeout counter will stop counting when the ENTOUT bit is cleared. However, the counter will also stop counting when one of the conditions, listed as follows, occurs:

- The I²C slave module is not addressed.
- The I²C slave module detects a STOP signal.
- The I²C master module sends a STOP signal.
- The ARBLOS or BUSERR flag in the I2CSR register are asserted.

If the timeout counter underflows, the corresponding timeout flag, TOUTF, in the I2CSR register will be set to 1 and a timeout interrupt will be generated if the relevant interrupt is enabled.

PDMA Interface

The PDMA interface is integrated in the I²C module. The PDMA function can be enabled by setting the TXDMAE or RXDMAE bit to 1 in the transmitter or receiver mode respectively. When the data register is empty in the transmitter mode and the TXDMAE bit is set to 1, the PMDA function will be activated to move data from a certain memory location into the I²C data register. Similarly, when the data register is not empty in the receiver mode and the RXDMAE bit is set to 1, the PDMA function will also be activated to move data from the I²C data register to a specific memory location.

The DMA NACK control bit, DMANACK, is used to determine whether the NACK signal is sent or not when the I²C module operates in the master receiver mode and the PDMA function is enabled. If the DMANACK bit is set to 1 and the data has all been received and moved using the PDMA interface, a NACK signal will automatically be sent out to properly terminate the data transfer.

For a more detailed description on the PDMA configurations, refer to the PDMA chapter.

Register Map

The following table shows the I²C registers and reset values.

Table 41. I²C Register Map

Register	Offset	Description	Reset Value
I2CCR	0x000	I²C Control Register	0x0000_2000
I2CIER	0x004	I²C Interrupt Enable Register	0x0000_0000
I2CADDR	0x008	I²C Address Register	0x0000_0000
I2CSR	0x00C	I²C Status Register	0x0000_0000
I2CSHPGR	0x010	I²C SCL High Period Generation Register	0x0000_0000
I2CSLPGR	0x014	I²C SCL Low Period Generation Register	0x0000_0000
I2CDR	0x018	I²C Data Register	0x0000_0000
I2CTAR	0x01C	I²C Target Register	0x0000_0000
I2CADDMR	0x020	I²C Address Mask Register	0x0000_0000
I2CADDSR	0x024	I²C Address Snoop Register	0x0000_0000
I2CTOUT	0x028	I²C Timeout Register	0x0000_0000

Register Descriptions

I²C Control Register – I2CCR

This register specifies the corresponding I²C function enable control.

Offset: 0x000 (0)
Reset value: 0x0000_2000

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	22	21	20	19	18	17	16
	Reserved							
Type/Reset								
	15	14	13	12	11	10	9	8
	SEQFILTER		COMBFILTEREN	ENTOUT	Reserved	DMANACK	RXDMAE	TXDMAE
Type/Reset	RW 0	RW 0	RW 1	RW 0		RW 0	RW 0	RW 0
	7	6	5	4	3	2	1	0
	ADRM	Reserved			I2CEN	GCEN	STOP	AA
Type/Reset	RW 0				RW 0	RW 0	RW 0	RW 0

Bits	Field	Descriptions
[15:14]	SEQFILTER	SDA or SCL Input Sequential Filter Configuration Bits 00: Sequential filter is disabled 01: 1 PCLK glitch filter 1x: 2 PCLK glitch filter Note: This setting would affect the frequency of SCL. Detail is described in I2CSLPGR register.
[13]	COMBFILTEREN	SDA or SCL Input Combinational Filter Enable Bit 0: Combinational filter is disabled 1: Combinational filter is enabled
[12]	ENTOUT	I²C Timeout Function Enable Control 0: Timeout Function is disabled 1: Timeout Function is enabled This bit is used to enable or disable the I²C timeout function. When the I2CEN bit is cleared to 0, the ENTOUT bit will be automatically cleared to 0 by hardware. It is recommended that users have to properly configure the PSC and TOUT fields in the I2CTOUT register before the timeout counter starts to count by setting the ENOUT bit to 1.
[10]	DMANACK	DMA Mode NACK Control 0: No operation 1: The I²C master receiver module sends a NACK signal automatically after receiving the last byte from the slave transmitter in the DMA mode When the I2CEN bit is cleared to 0, the DMANACK bit is automatically cleared to 0 by hardware.

Bits	Field	Descriptions
[9]	RXDMAE	DMA Mode RX Request Enable Control 0: RX DMA request is disabled 1: RX DMA request is enabled If the data register is not empty in the receiver mode and the RXDMAE bit is set to 1, the relevant PDMA channel will be activated to move the data from the data register to a specific location which is defined in the corresponding PDMA register. When the I2CEN bit is cleared to 0, the RXDMAE bit is automatically cleared to 0 by hardware.
[8]	TXDMAE	DMA Mode TX Request Enable Control 0: TX DMA request is disabled 1: TX DMA request is enabled If the data register is empty in the transmitter mode and the TXDMAE bit is set to 1, the relevant PDMA channel will be activated to move the data from a specific location defined in the related PDMA register to the data register. When the I2CEN bit is cleared to 0, the TXDMAE bit is automatically cleared to 0 by hardware.
[7]	ADRM	Addressing Mode 0: 7-bit addressing mode 1: 10-bit addressing mode When the I^2C master / slave module operates in the 7-bit addressing mode, it can only send out and respond to a 7-bit address and vice versa. When the I2CEN bit is disabled, the ADRM bit is automatically cleared to 0 by hardware.
[3]	I2CEN	I^2C Interface Enable 0: I^2C interface is disabled 1: I^2C interface is enabled
[2]	GCEN	General Call Enable 0: General call is disabled 1: General call is enabled When the device receives the calling address with a value of 0x00 and if both the GCEN and the AA bits are set to 1, then the I^2C interface is addressed as a slave and the GCS bit in the I2CSR register is set to 1. When the I2CEN bit is cleared to 0, the GCEN bit is automatically cleared to 0 by hardware.
[1]	STOP	STOP Condition Control 0: No action 1: Send a STOP condition in master mode This bit is set to 1 by software to generate a STOP condition and automatically cleared to 0 by hardware. The STOP bit is only available for the master device.
[0]	AA	Acknowledge Bit 0: Send a Not Acknowledge (NACK) signal after a byte is received 1: Send an Acknowledge (ACK) signal after a byte is received When the I2CEN bit is cleared to 0, the AA bit is automatically cleared to 0 by hardware.

I²C Interrupt Enable Register – I2CIER

This register specifies the corresponding I²C interrupt enable bits.

Offset: 0x004

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	22	21	20	19	18	17	16
	Reserved					RXBFIE	TXDEIE	RXDNEIE
Type/Reset						RW 0	RW 0	RW 0
	15	14	13	12	11	10	9	8
	Reserved				TOUTIE	BUSERRIE	RXNACKIE	ARBLOSIE
Type/Reset					RW 0	RW 0	RW 0	RW 0
	7	6	5	4	3	2	1	0
	Reserved				GCSIE	ADRSIE	STOIE	STAIE
Type/Reset					RW 0	RW 0	RW 0	RW 0

Bits	Field	Descriptions
[18]	RXBFIE	RX Buffer Full Interrupt Enable Bit 0: Interrupt is disabled 1: Interrupt is enabled When the I2CEN bit in the I2CCR register is cleared to 0, this bit is cleared to 0 by hardware.
[17]	TXDEIE	Data Register Empty Interrupt Enable Bit in Transmitter Mode 0: Interrupt is disabled 1: Interrupt is enabled When the I2CEN bit in the I2CCR register is cleared to 0, this bit is cleared to 0 by hardware.
[16]	RXDNEIE	Data Register Not Empty Interrupt Enable Bit in Received Mode 0: Interrupt is disabled 1: Interrupt is enabled When the I2CEN bit in the I2CCR register is cleared to 0, this bit is cleared to 0 by hardware.
[11]	TOUTIE	Timeout Interrupt Enable Bit 0: Interrupt is disabled 1: Interrupt is enabled When the I2CEN bit in the I2CCR register is cleared to 0, this bit is cleared to 0 by hardware.
[10]	BUSERRIE	Bus Error Interrupt Enable Bit 0: Interrupt is disabled 1: Interrupt is enabled When the I2CEN bit in the I2CCR register is cleared to 0, this bit is cleared to 0 by hardware.
[9]	RXNACKIE	Received Not Acknowledge Interrupt Enable Bit 0: Interrupt is disabled 1: Interrupt is enabled When the I2CEN bit in the I2CCR register is cleared to 0, this bit is cleared to 0 by hardware.

Bits	Field	Descriptions
[8]	ARBLOSIE	Arbitration Loss Interrupt Enable Bit in the I²C multi-master mode 0: Interrupt is disabled 1: Interrupt is enabled When the I2CEN bit in the I2CCR register is cleared to 0, this bit is cleared to 0 by hardware.
[3]	GCSIE	General Call Slave Interrupt Enable Bit 0: Interrupt is disabled 1: Interrupt is enabled When the I2CEN bit in the I2CCR register is cleared to 0, this bit is cleared to 0 by hardware.
[2]	ADRSIE	Slave Address Match Interrupt Enable Bit 0: Interrupt is disabled 1: Interrupt is enabled When the I2CEN bit in the I2CCR register is cleared to 0, this bit is cleared to 0 by hardware.
[1]	STOIE	STOP Condition Detected Interrupt Enable Bit 0: Interrupt is disabled 1: Interrupt is enabled When the I2CEN bit in the I2CCR register is cleared to 0, this bit is cleared to 0 by hardware. The bit is used for the I²C slave mode only.
[0]	STAIE	START Condition Transmit Interrupt Enable Bit 0: Interrupt is disabled 1: Interrupt is enabled When the I2CEN bit in the I2CCR register is cleared to 0, this bit is cleared to 0 by hardware. The bit is used for the I²C master mode only.

I²C Address Register – I2CADDR

This register specifies the I²C device address.

Offset: 0x008

Reset value: 0x0000_0000

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

Bits	Field	Descriptions
[9:0]	ADDR	Device Address The register indicates the I²C device address. When the I²C device is used in the 7-bit addressing mode, only the ADDR[6:0] bits will be compared with the received address sent from the I²C master device.

I²C Status Register – I2CSR

This register contains the I²C operation status.

Offset: 0x00C

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	22	21	20	19	18	17	16
		Reserved	TXNRX	MASTER	BUSBUSY	RXBF	TXDE	RXDNE
Type/Reset			RO 0	RO 0	RO 0	RO 0	RO 0	RO 0
	15	14	13	12	11	10	9	8
	Reserved				TOUTF	BUSERR	RXNACK	ARBLOS
Type/Reset					WC 0	WC 0	WC 0	WC 0
	7	6	5	4	3	2	1	0
	Reserved				GCS	ADRS	STO	STA
Type/Reset					RC 0	RC 0	RC 0	RC 0

Bits	Field	Descriptions
[21]	TXNRX	Transmitter / Receiver Mode 0: Receiver mode 1: Transmitter mode Read only bit.
[20]	MASTER	Master Mode 0: I²C is in the slave mode or idle 1: I²C is in the master mode The I²C interface is switched as a master device on the I²C bus when the I2CTAR register is assigned and the I²C bus is idle. The MASTER bit is cleared by hardware when software disables the I²C bus by clearing the I2CEN bit to 0 or sends a STOP condition to the I²C bus or the bus error is detected. This bit is set and cleared by hardware and is a read only bit.
[19]	BUSBUSY	Bus Busy 0: I²C bus is idle 1: I²C bus is busy The I²C interface hardware starts to detect the I²C bus status if the interface is enabled by setting the I2CEN bit to 1. It is set to 1 when the SDA or SCL signal is detected to have a logic low state and cleared when a STOP condition is detected.
[18]	RXBF	Buffer Full Flag in Receiver Mode 0: Data buffer is not full 1: Data buffer is full This bit is set when the data register I2CDR has already stored a data byte and meanwhile the data shift register also has been received a complete new data byte. The RXBF bit is cleared by software reading the I2CDR register.

Bits	Field	Descriptions
[17]	TXDE	Data Register Empty Using in Transmitter Mode 0: Data register I2CDR is not empty 1: Data register I2CDR is empty This bit is set when the I2CDR register is empty in the Transmitter mode. Note that the TXDE bit will be set after the address frame is being transmitted to inform that the data to be transmitted should be loaded into the I2CDR register. The TXDE bit is cleared by software writing data to the I2CDR register in both the master and slave mode or cleared automatically by hardware after setting the STOP signal to terminate the data transfer or setting the I2CTAR register to restart a new data transfer in the master mode.
[16]	RXDNE	Data Register Not Empty in Receiver Mode 0: Data register I2CDR is empty 1: Data register I2CDR is not empty This bit is set when the I2CDR register is not empty in the receiver mode. The RXDNE bit is cleared by software reading the data byte from the I2CDR register.
[11]	TOUTF	Timeout Counter Underflow Flag 0: No timeout counter underflow occurred 1: Timeout counter underflow occurred Writing “1” to this bit will clear the TOUTF flag.
[10]	BUSERR	Bus Error Flag 0: No bus error has occurred 1: Bus error has occurred This bit is set by hardware when the I^2C interface detects a misplaced START or STOP condition in a transfer process. Writing a “1” to this bit will clear the BUSERR flag. In Master Mode: Once the Bus Error event occurs, both the SDA and SCL lines are released by hardware and the BUSERR flag is asserted. The application software has to clear the BUSERR flag before the next address byte is transmitted. In Slave Mode: Once a misplaced START or STOP condition has been detected by the slave device, the software must clear the BUSERR flag before the next address byte is received.
[9]	RXNACK	Received Not Acknowledge Flag 0: Acknowledge is returned from receiver 1: Not Acknowledge is returned from receiver The RXNACK bit indicates that the not Acknowledge signal is received in master or slave transmitter mode. Writing “1” to this bit will clear the RXNACK flag.
[8]	ARBLOS	Arbitration Loss Flag 0: No arbitration loss is detected 1: Bit arbitration loss is detected This bit is set by hardware on the current clock which the I^2C interface loses the bus arbitration to another master during the address or data frame transmission. Writing “1” to this bit will clear the ARBLOS flag. Once the ARBLOS flag is asserted by hardware, the ARBLOS flag must be cleared before the next transmission.
[3]	GCS	General Call Slave Flag 0: No general call slave occurs 1: I^2C interface is addressed by a general call command When the I^2C interface receives an address with a value of 0x00 or 0x000 in the 7-bit or 10-bit addressing mode, if both the GCEN and the AA bit are set to 1, then it is switched as a general call slave. This flag is cleared automatically after being read.

Bits	Field	Descriptions
[2]	ADRS	Address Transmit (master mode) / Address Receive (slave mode) Flag Address Sent in Master Mode 0: Address frame has not been transmitted 1: Address frame has been transmitted For the 7-bit addressing mode, this bit is set after the master device receives the address frame acknowledge bit sent from the slave device. For the 10-bit addressing mode, this bit is set after receiving the acknowledge bits of the first header byte and the second address. Address Matched in Slave Mode 0: I^2C interface is not addressed 1: I^2C interface is addressed as slave When the I^2C interface has received the calling address that matches the address defined in the I2CADDR register together with the AA bit being set to 1 in the I2CCR register, it will be switched to a slave mode. This flag is cleared automatically after the I2CSR register has been read.
[1]	STO	STOP Condition Detected Flag 0: No STOP condition detected 1: STOP condition detected in slave mode This bit is only available for the slave mode and is cleared automatically after the I2CSR register is read.
[0]	STA	START Condition Transmit 0: No START condition detected 1: START condition is transmitted in master mode This bit is only available for the master mode and is cleared automatically after the I2CSR register is read.

I²C SCL High Period Generation Register – I2CSHPGR

This register specifies the I²C SCL clock high period interval.

Offset: 0x010

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	22	21	20	19	18	17	16
	Reserved							
Type/Reset								
	15	14	13	12	11	10	9	8
	SHPG							
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0
	7	6	5	4	3	2	1	0
	SHPG							
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0

Bits	Field	Descriptions
[15:0]	SHPG	SCL Clock High Period Generation High period duration setting $SCL_{HIGH} = T_{PCLK} \times (SHPG + d)$ where T_{PCLK} is the APB bus peripheral clock (PCLK) period, and d value depends on the setting of the SEQFILTER field in the I²C Control Register (I2CCR). If SEQFILTER = 00, d = 6 If SEQFILTER = 01, d = 8 If SEQFILTER = 10 or 11, d = 9

I²C SCL Low Period Generation Register – I2CSLPGR

This register specifies the I²C SCL clock low period interval.

Offset: 0x014

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	22	21	20	19	18	17	16
	Reserved							
Type/Reset								
	15	14	13	12	11	10	9	8
	SLPG							
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0
	7	6	5	4	3	2	1	0
	SLPG							
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0

Bits	Field	Descriptions
[15:0]	SLPG	SCL Clock Low Period Generation Low period duration setting $SCL_{LOW} = T_{PCLK} \times (SLPG + d)$ where T_{PCLK} is the APB bus peripheral clock (PCLK) period, and d value depends on the setting of the SEQFILTER field in the I²C Control Register (I2CCR). If SEQFILTER = 00, d = 6 If SEQFILTER = 01, d = 8 If SEQFILTER = 10 or 11, d = 9

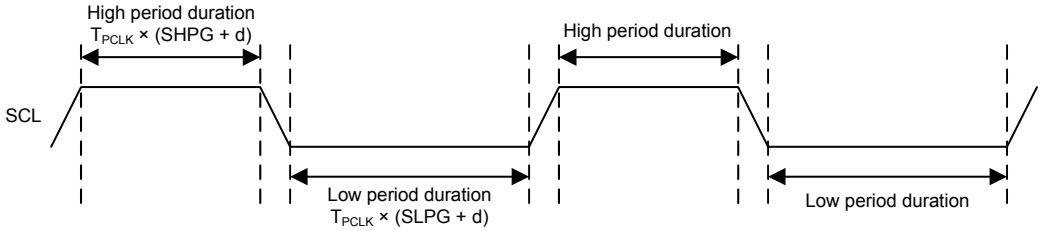


Figure 102. SCL Timing Diagram

Table 42. I²C Clock Setting Example

I²C Clock	$T_{SCL} = T_{PCLK} \times [(SHPG + d) + (SLPG + d)]$ (where d = 6) SHPG + SLPG Value at PCLK		
	8 MHz	24 MHz	48 MHz
100 kHz (Standard Mode)	68	228	468
400 kHz (Fast Mode)	8	48	108
1 MHz (Fast-Mode Plus)	x	12	36

I²C Data Register – I2CDR

This register specifies the data to be transmitted or received by the I²C module.

Offset: 0x018

Reset value: 0x0000_0000

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

Bits	Field	Descriptions
[7:0]	DATA	I²C Data Register For the transmitter mode, a data byte which is transmitted to a slave device can be assigned to these bits. The TXDE flag is cleared if the application software assigns new data to the I2CDR register. For the receiver mode, a data byte is received bit by bit from MSB to LSB through the I²C interface and stored in the data shift register. Once the acknowledge bit is given, the data shift register value is delivered into the I2CDR register if the RXDNE flag is equal to 0.

I²C Target Register – I2CTAR

This register specifies the target device address to be communicated.

Offset: 0x01C

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	22	21	20	19	18	17	16
	Reserved							
Type/Reset								
	15	14	13	12	11	10	9	8
	Reserved					RWD	TAR	
Type/Reset						RW 0	RW 0	RW 0
	7	6	5	4	3	2	1	0
	TAR							
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0

Bits	Field	Descriptions
[10]	RWD	Read or Write Direction 0: Write direction to target slave address 1: Read direction from target slave address If this bit is set to 1 in the 10-bit master receiver mode, the I²C interface will initiate a byte with a value of 11110XX0b in the first header frame and then continue to deliver a byte with a value of 11110XX1b in the second header frame by hardware automatically.
[9:0]	TAR	Target Slave Address The I²C interface will assign a START signal and send a target slave address automatically once the data is written to this register. When the system wants to send a repeated START signal to the I²C bus, the timing is suggested to set the I2CTAR register after a byte transfer is completed. It is not allowed to set TAR in the address frame. I2CTAR[9:7] is not available under the 7-bit addressing mode.

I²C Address Mask Register – I2CADDMR

This register specifies which bit of the I²C address is masked and not compared with corresponding bit of the received address frame.

Offset: 0x020

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	22	21	20	19	18	17	16
	Reserved							
Type/Reset								
	15	14	13	12	11	10	9	8
	Reserved						ADDMR	
Type/Reset							RW 0	RW 0
	7	6	5	4	3	2	1	0
	ADDMR							
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0

Bits	Field	Descriptions
[9:0]	ADDMR	Address Mask Control Bit The ADDMR[i] is used to specify whether the i^{th} bit of the ADDR in the I2CADDR register is masked and is compared with the received address frame or not on the I²C bus. The register is only used for the I²C slave mode only. 0: i^{th} bit of the ADDR is compared with the address frame on the I²C bus 1: i^{th} bit of the ADDR is masked and not compared with the address frame on the I²C bus

I²C Address Snoop Register – I2CADDSR

This register is used to indicate the address frame value appeared on the I²C bus.

Offset: 0x024

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	22	21	20	19	18	17	16
	Reserved							
Type/Reset								
	15	14	13	12	11	10	9	8
	Reserved						ADDSR	
Type/Reset							RO 0	RO 0
	7	6	5	4	3	2	1	0
	ADDSR							
Type/Reset	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0

Bits	Field	Descriptions
[9:0]	ADDSR	Address Snoop Once the I2CEN bit is enabled, the calling address value on the I²C bus will automatically be loaded into this ADDSR field.

I²C Timeout Register – I2CTOUT

This register specifies the I²C Timeout counter preload value and clock prescaler ratio.

Offset: 0x028

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	22	21	20	19	18	17	16
	Reserved					PSC		
Type/Reset						RW 0	RW 0	RW 0
	15	14	13	12	11	10	9	8
	TOUT							
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0
	7	6	5	4	3	2	1	0
	TOUT							
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0

Bits	Field	Descriptions
[18:16]	PSC	I²C Timeout Counter Prescaler Selection This PSC field is used to specify the I²C timeout counter clock frequency, f_{I2CTO}. The timeout clock frequency is obtained using the formula. $f_{I2CTO} = \frac{f_{PCLK}}{2^{PSC}}$ PSC = 0 → $f_{I2CTO} = f_{PCLK} / 2^0 = f_{PCLK}$ PSC = 1 → $f_{I2CTO} = f_{PCLK} / 2^1 = f_{PCLK} / 2$ PSC = 2 → $f_{I2CTO} = f_{PCLK} / 2^2 = f_{PCLK} / 4$ … PSC = 7 → $f_{I2CTO} = f_{PCLK} / 2^7 = f_{PCLK} / 128$
[15:0]	TOUT	I²C Timeout Counter Preload Value The TOUT field is used to define the counter preloaded value The counter value is reloaded when any of the following conditions occurs: 1. The RXBF, TXDE, RXDNE, RXNACK, GCS or ADRS flag in the I2CSR register is asserted. 2. The I²C master module sends a START signal. 3. The I²C slave module detects a START signal. The counter stops counting when any of the following conditions occurs: 1. The I²C slave device is not addressed. 2. The I²C master module sends a STOP signal. 3. The I²C slave module detects a STOP signal. 4. The ARBLOS or BUSERR flag in the I2CSR register is asserted.

19 Serial Peripheral Interface (SPI)

Introduction

The Serial Peripheral Interface, SPI, provides an SPI protocol data transmit and receive functions in both master or slave mode. The SPI interface uses 4 pins, among which are serial data input and output lines MISO and MOSI, the clock line SCK, and the slave select line SEL. One SPI device acts as a master who controls the data flow using the SEL and SCK signals to indicate the start of the data communication and the data sampling rate. To receive the data bits, the streamlined data bits which range from 1 bit to 16 bits specified by the DFL field in the SPICR1 register are latched on a specific clock edge and stored in the data register or in the RX FIFO. Data transmission is carried in a similar way but with the reverse sequence. The mode fault detection provides a capability for multi-master applications.

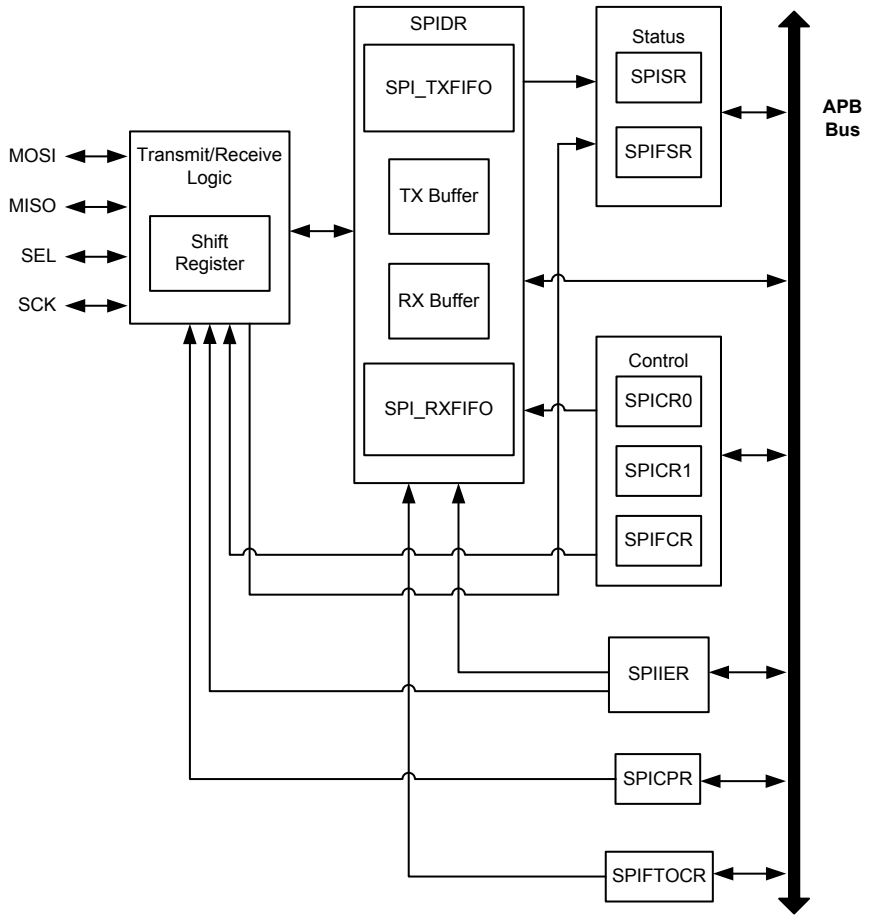


Figure 103. SPI Block Diagram

Features

- Master or slave mode
- Master mode speed up to $f_{PCLK}/2$
- Slave mode speed up to $f_{PCLK}/3$
- Programmable data frame length up to 16 bits
- FIFO Depth: 8 levels
- MSB or LSB first shift selection
- Programmable slave select high or low active polarity
- Multi-master and multi-slave operation
- Master mode supports the dual output read mode of SPI series NOR Flash
- Four error flags with individual interrupt
  - Read overrun
  - Write collision
  - Mode fault
  - Slave abort

Function Descriptions

Master Mode

Each data frame can range from 1 to 16 bits in data length. The first bit of the transmitted data can be either an MSB or LSB determined by the FIRSTBIT bit in the SPICR1 register. The SPI module is configured as a master or a slave by setting the MODE bit in the SPICR1 register. When the MODE bit is set, the SPI module is configured as a master and will generate the serial clock on the SCK pin. The data stream will transmit data in the shift register to the MOSI pin on the serial clock edge. The SEL pin is active during the full data transmission. When the SELAP bit in the SPICR1 register is set, the SEL pin is active high during the complete data transactions. When the SELM bit in the SPICR1 register is set, the SEL pin will be driven by the hardware automatically and the time interval between the active SEL edge and the first edge of SCK is equal to half an SCK period.

Slave Mode

In the slave mode, the SCK pin acts as an input pin and the serial clock will be derived from the external master device. The SEL pin also acts as an input. When the SELAP bit is cleared to 0, the SEL signal is active low during the full data stream reception. When the SELAP bit is set to 1, the SEL signal will be active high during the full data stream reception.

Note: For the slave mode, the APB clock, known as f_{PCLK}, must be at least 3 times faster than the external SCK clock input frequency.

SPI Serial Frame Format

The SPI interface format is based on the Clock Polarity, CPOL, and the Clock Phase, CPHA, configurations.

- Clock Polarity Bit – CPOL
  When the Clock Polarity bit is cleared to 0, the SCK line idle state is LOW. When the Clock Polarity bit is set to 1, the SCK line idle state is HIGH.
- Clock Phase Bit – CPHA
  When the Clock Phase bit is cleared to 0, the data is sampled on the first SCK clock transition. When the Clock Phase bit is set to 1, the data is sampled on the second SCK clock transition.

There are four formats contained in the SPI interface. Table 43 shows how to configure these formats by setting the FORMAT field in the SPICR1 register.

Table 43. SPI Interface Format Setup

FORMAT [2:0]	CPOL	CPHA
001	0	0
010	0	1
110	1	0
101	1	1
Others	Reserved	

CPOL = 0, CPHA = 0

In this format, the received data is sampled on the SCK line rising edge while the transmitted data is changed on the SCK line falling edge. In the master mode, the first bit is driven when data is written into the SPIDR Register. In the slave mode, the first bit is driven when the SEL signal goes to an active level. Figure 104 shows the single byte data transfer timing of this format.

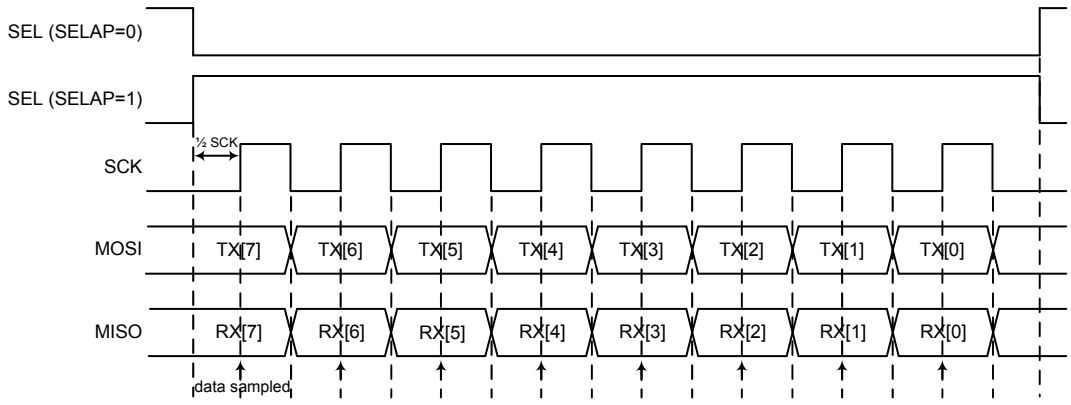


Figure 104. SPI Single Byte Transfer Timing Diagram – CPOL = 0, CPHA = 0

Figure 105 shows the continuous data transfer timing diagram of this format. Note that the SEL signal must change to an inactive level between each data frame.

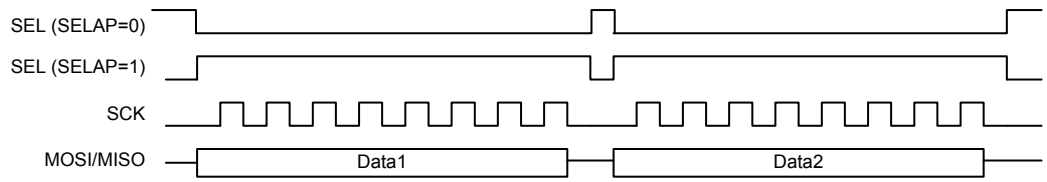


Figure 105. SPI Continuous Data Transfer Timing Diagram – CPOL = 0, CPHA = 0

CPOL = 0, CPHA = 1

In this format, the received data is sampled on the SCK line falling edge while the transmitted data is changed on the SCK line rising edge. In the master mode, the first bit is driven when data is written into the SPIDR register. In the slave mode, the first bit is driven at the first SCK clock rising edge. Figure 106 shows the single data byte transfer timing.

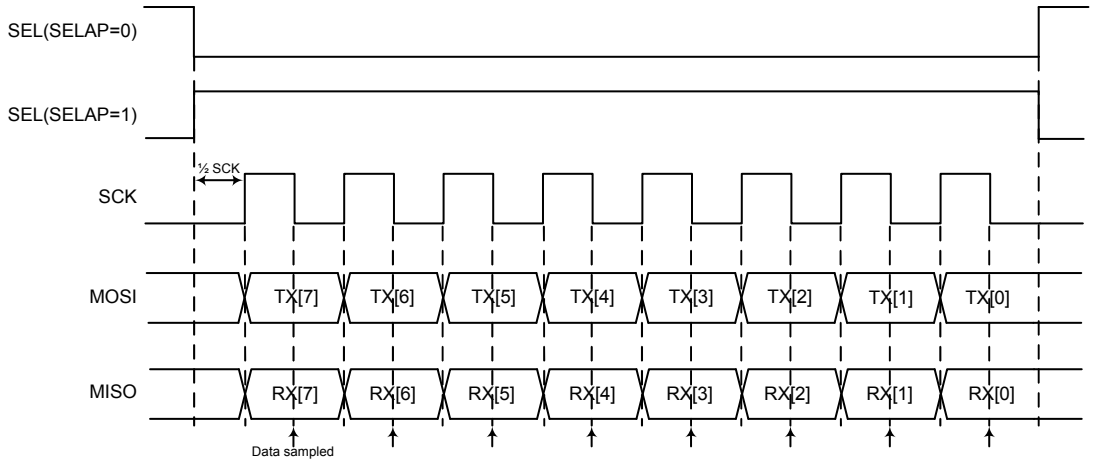


Figure 106. SPI Single Byte Transfer Timing Diagram – CPOL = 0, CPHA = 1

Figure 107 shows the continuous data transfer diagram timing. Note that the SEL signal must remain active until the last data transfer has completed.

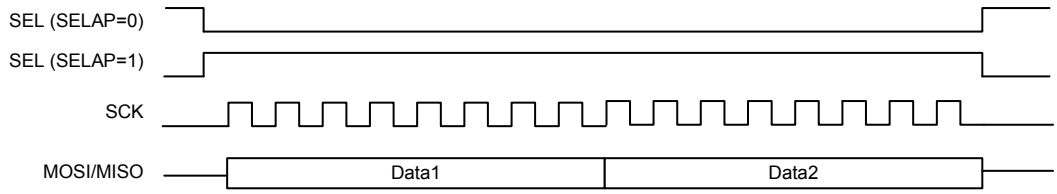


Figure 107. SPI Continuous Transfer Timing Diagram – CPOL = 0, CPHA = 1

CPOL = 1, CPHA = 0

In this format, the received data is sampled on the SCK line falling edge while the transmitted data is changed on the SCK line rising edge. In the master mode, the first bit is driven when data is written into the SPIDR register. In the slave mode, the first bit is driven when the SEL signal changes to an active level. Figure 108 shows the single byte transfer timing of this format.

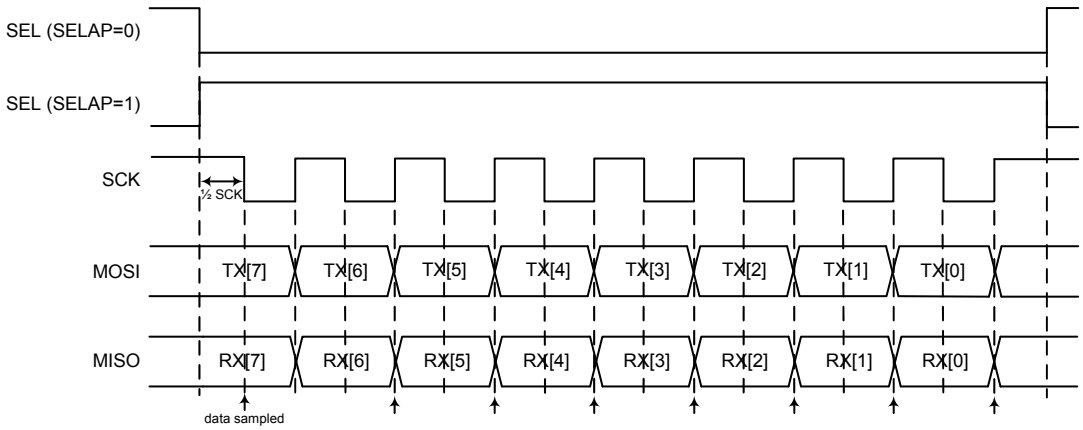


Figure 108. SPI Single Byte Transfer Timing Diagram – CPOL = 1, CPHA = 0

Figure 109 shows the continuous data transfer timing of this format. Note that the SEL signal must change to an inactive level between each data frame.

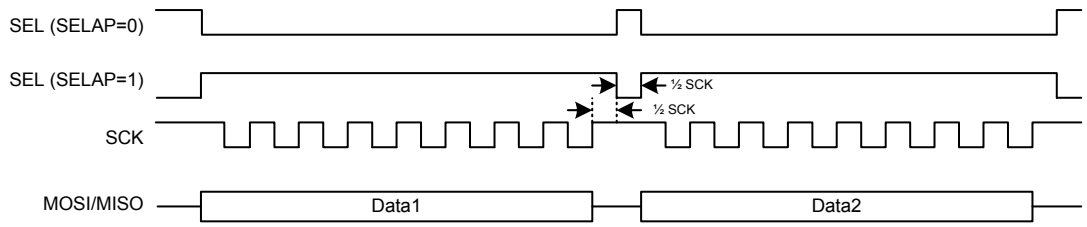


Figure 109. SPI Continuous Transfer Timing Diagram – CPOL = 1, CPHA = 0

CPOL = 1, CPHA = 1

In this format, the received data is sampled on the SCK line rising edge while the transmitted data is changed on the SCK line falling edge. In the master mode, the first bit is driven when data is written into the SPIDR register. In the slave mode, the first bit is driven at the first SCK falling edge. Figure 110 shows the single byte transfer timing of this format.

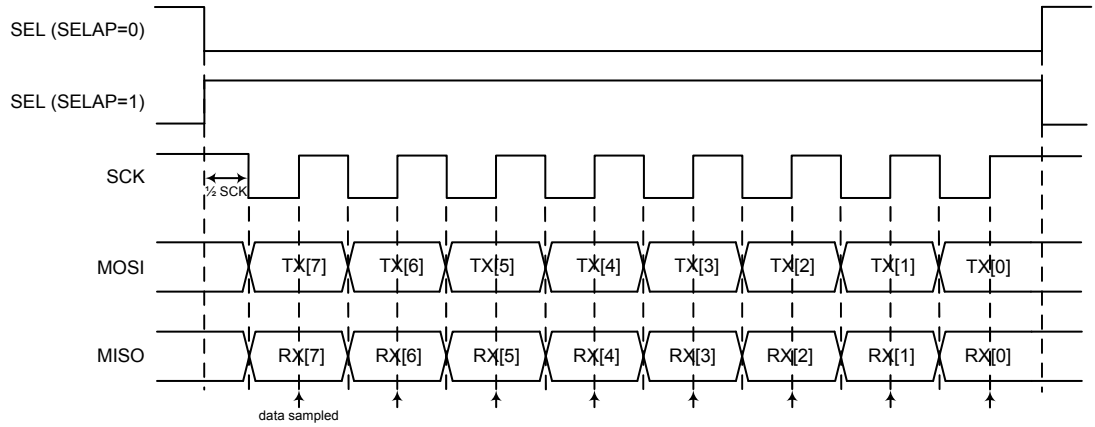


Figure 110. SPI Single Byte Transfer Timing Diagram – CPOL = 1, CPHA = 1

Figure 111 shows the continuous data transfer timing of this format. Note that the SEL signal must remain active until the last data transfer has completed.

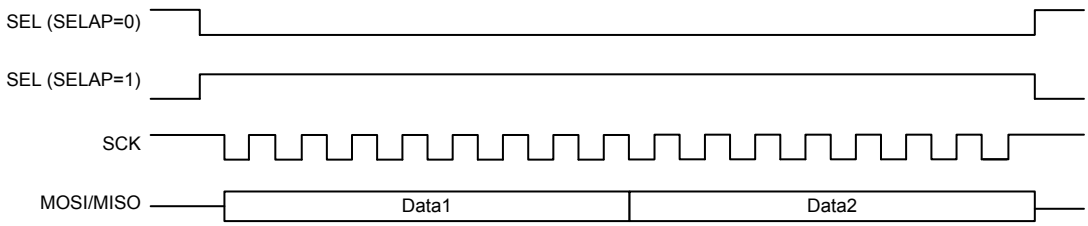


Figure 111. SPI Continuous Transfer Timing Diagram – CPOL = 1, CPHA = 1

Status Flags

TX Buffer Empty – TXBE

This TXBE flag is set when the TX buffer is empty in the non-FIFO mode or when the TX FIFO data length is equal to or less than the TX FIFO threshold level as defined by the TXFTLS field in the SPIFCR register in the FIFO mode. The following data to be transmitted can then be loaded into the buffer again. After this, the TXBE flag will be reset when the TX buffer already contains new data in the non-FIFO mode or when the TX FIFO data length is greater than the TX FIFO threshold level determined by the TXFTLS bits in the FIFO mode.

Transmission Register Empty – TXE

This TXE flag is set when both the TX buffer and the TX shift registers are empty. It will be reset when the TX buffer or the TX shift register contains new transmitted data.

RX Buffer Not Empty – RXBNE

This RXBNE flag is set when there is valid received data in the RX buffer in the non-FIFO mode or the RX FIFO data length is equal to or greater than the RX FIFO threshold level as defined by the RXFTLS field in the SPIFCR register in the SPI FIFO mode. This flag will be automatically cleared by hardware when the received data have been read out from the RX buffer totally in the non-FIFO mode or when the RX FIFO data length is less than the RX FIFO threshold level set in the RXFTLS field.

Time Out Flag – TO

The time out function is only available in the SPI FIFO mode and is disabled by loading a zero value into the TOC field in the Time Out Counter register. The timeout counter will start counting if the SPI RX FIFO is not empty, once data is read from the SPIDR register or new data is received, the timeout counter will be reset to 0 and count again. When the timeout counter value is equal to the value specified by the TOC field in the SPIFTOCR register, the TO flag will be set. The flag is cleared by writing 1 to this bit.

Mode Fault – MF

The mode fault flag can be used to detect SPI bus usage in the SPI multi-master mode. For the multi-master mode, the SPI module is configured as a master device and the SEL signal is setup as an input signal. The mode fault flag is set when the SPI SEL pin is suddenly changed to an active level by another SPI master. This means that another SPI master is requesting to use the SPI bus. Therefore, when an SPI mode fault occurs, it will force the SPI module to operate in the slave mode and also disable all of the SPI interface signals to avoid SPI bus signal collisions. For the same reason, if the SPI master wants to transfer data, it also needs to inform other SPI masters by driving its SEL signal to an active state. The detailed configuration diagram for the SPI multi-master mode is shown in the following figure.

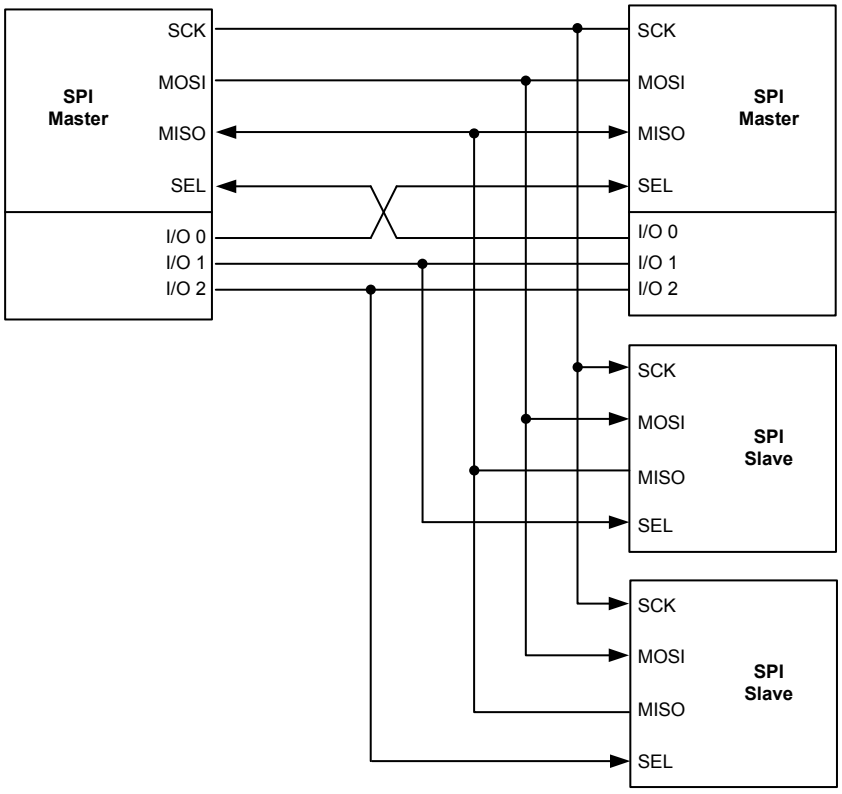


Figure 112. SPI Multi-Master Slave Environment

Table 44. SPI Mode Fault Trigger Conditions

Mode Fault	Descriptions
Trigger condition	1. SPI Master mode 2. SELOEN = 0 in the SPICR0 register – SEL pin is configured to be the input mode 3. SEL signal changes to an active level when driven by the external SPI master
SPI behavior	1. Mode fault flag is set. 2. The SPIEN bit in the SPICR0 register is reset. This disables the SPI interface and blocks all output signals from the device. 3. The MODE bit in the SPICR1 register is reset. This forces the device into slave mode.

Table 45. SPI Master Mode SEL Pin Status

	SEL as Input – SELOEN = 0		SEL as Output – SELOEN = 1	
Multi-master	Support		Not support	
SPI SEL control signal	Use Another GPIO to replace the SEL pin function		SEL pin in hardware or software control mode - using SELM setting	
Continuous transfer	Case 1	Case 2	Case 1	Case 2
	Not support	Support	Hardware control	Hardware or software control

Case 1: SEL signal must be inactive between each data transfer.
Case 2: SEL signal will not to be inactive until the last data frame has finished.
Note: When the SPI is in the slave mode, the SEL signal is always an input and not affected by the SELOEN bit in the SPICR0 register.

Write Collision – WC

The following conditions will assert the Write Collision Flag.

- The FIFOEN bit in the SPIFCR register is cleared
  The write collision flag is asserted when new data is written into the SPIDR register while both the TX buffer and the shift register are already full. Any new data written into the TX buffer will be lost.
- The FIFOEN bit in the SPIFCR register is set
  The write collision flag is asserted to indicate that new data is written into the SPIDR register while both the TX FIFO and the TX shift register are already full. Any new data written into the TX FIFO will be lost.

Read Overrun – RO

- The FIFOEN bit in the SPIFCR register is cleared
  The read overrun flag is asserted to indicate that both the RX shift register and the RX buffer are already full, if one more data is received. This will result in the newly received data not being shifted into the SPI shift register. As a result the latest received data will be lost.
- The FIFOEN bit in the SPIFCR register is set
  The read overrun flag is set to indicate that the RX shift register and the RX FIFO are both full, if one more data is received. This means that the latest received data can not be shifted into the SPI shift register. As a result the latest received data will be lost.

Slave Abort – SA

In the SPI slave mode, the slave abort flag is set to indicate that the SEL pin suddenly changed to an inactive state during the reception of a data frame transfer. The data frame length is set by the DFL field in the SPICR1 register.

PDMA Interface

The PDMA interface is integrated in the SPI module. The PDMA function can be enabled by setting the TXDMAE or RXDMAE bit to 1 in the transmitter or receiver mode respectively. When the transmit buffer empty flag, TXBE, is asserted and the TXDMAE bit is set to 1, the PMDA function will be activated to move data from the memory location that users designated into the SPI data register or the TX FIFO until the TXBE flag is cleared to 0. The TXBE flag will be asserted when the transmit buffer is empty in the non-FIFO mode or the data contained in the TX FIFO is equal to or less than the level defined by the TXFTLS field in the FIFO mode.

Similarly, when the receive buffer not empty flag, RXBNE, is asserted and the RXDMAE bit is set to 1, the PDMA function will be activated to move data from the SPI data register or the RX FIFO to the memory location that users designated until the RXBNE flag is cleared to 0. The RXBNE flag will be asserted when the receive buffer is not empty in the non-FIFO mode or the data contained in the RX FIFO is equal to or greater than the level defined by the RXFTLS field in the FIFO mode.

For a more detailed description on the PDMA configurations, refer to the PDMA chapter.

Register Map

The following table shows the SPI registers and their reset values.

Table 46. SPI Register Map

Register	Offset	Description	Reset Value
SPICR0	0x000	SPI Control Register 0	0x0000_0000
SPICR1	0x004	SPI Control Register 1	0x0000_0000
SPIIER	0x008	SPI Interrupt Enable Register	0x0000_0000
SPICPR	0x00C	SPI Clock Prescaler Register	0x0000_0000
SPIDR	0x010	SPI Data Register	0x0000_0000
SPISR	0x014	SPI Status Register	0x0000_0003
SPIFCR	0x018	SPI FIFO Control Register	0x0000_0000
SPIFSR	0x01C	SPI FIFO Status Register	0x0000_0000
SPIFTOCR	0x020	SPI FIFO Time Out Counter Register	0x0000_0000

Register Descriptions

SPI Control Register 0 – SPICR0

This register specifies the SEL control and the SPI enable bits.

Offset: 0x000

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	22	21	20	19	18	17	16
	Reserved							
Type/Reset								
	15	14	13	12	11	10	9	8
	SELHT				GUADT			
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0
	7	6	5	4	3	2	1	0
	GUADTEN	DUALEN	Reserved	SSELC	SELOEN	RXDMAE	TXDMAE	SPIEN
Type/Reset	RW 0	RW 0		RW 0	RW 0	RW 0	RW 0	RW 0

Bits	Field	Descriptions
[15:12]	SELHT	Chip Select Hold Time 0x0: 1/2 SCK 0x1: 1 SCK 0x2: 3/2 SCK 0x3: 2 SCK .... Note that SELHT is for master mode only.

Bits	Field	Descriptions
[11:8]	GUADT	Guard Time GUADTEN = 1 0x0: 1 SCK 0x1: 2 SCK 0x2: 3 SCK ... Note that GUADT is for master mode only.
[7]	GUADTEN	Guard Time Enable 0: Guard Time is 1/2 SCK 1: When this bit is set, Guard time can be controlled by GUADT Note that GUADTEN is for master mode only.
[6]	DUALEN	Dual Port Enable 0: Dual port is disabled 1: Dual port is enabled The control bit is used to support the dual output read mode of the series SPI NOR Flash. When this bit is set and the MOSI signal will change the direction from output to input and receive the series data stream. That means the DUALEN control bit is only for master mode.
[4]	SSELC	Software Slave Select Control 0: Set the SEL output to an inactive state 1: Set the SEL output to an active state The application software can setup the SEL output to an active or inactive state by configuring the SSELC bit. The active level is configured by the SELAP bit in the SPICR1 register. Note that the SSELC bit is only available when the SELOEN bit is set to 1 for enabling the SEL output meanwhile the SELM bit is cleared to 0 for controlling the SEL signal by software. Otherwise, the SSELC bit has no effect.
[3]	SELOEN	Slave Select Output Enable 0: Set the SEL signal to the input mode for multi-master mode 1: Set the SEL signal to the output mode for slave select The SELOEN is only available in the master mode to setup the SEL signal as an input or output signal. When the SEL signal is configured to operate in the output mode, it is used as a slave select signal in either the hardware or software mode according to the SELM bit setting in the SPICR1 register. The SEL signal is used for mode fault detection in the multi-master environment when it is configured to operate in the input mode.
[2]	RXDMAE	RX PDMA request enable 0: SPI RX path PDMA request is disabled 1: SPI RX path PDMA request is enabled
[1]	TXDMAE	TX PDMA request enable 0: SPI TX path PDMA request is disabled 1: SPI TX path PDMA request is enabled
[0]	SPIEN	SPI Enable 0: SPI interface is disabled 1: SPI interface is enabled

SPI Control Register 1 – SPICR1

This register specifies the SPI parameters including the data length, the transfer format, the SEL active polarity / mode, the LSB / MSB control and the master / slave mode.

Offset: 0x004

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	22	21	20	19	18	17	16
	Reserved							
Type/Reset								
	15	14	13	12	11	10	9	8
	Reserved	MODE	SELM	FIRSTBIT	SELAP	FORMAT		
Type/Reset		RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0
	7	6	5	4	3	2	1	0
	Reserved				DFL			
Type/Reset					RW 0	RW 0	RW 0	RW 0

Bits	Field	Descriptions
[14]	MODE	Master or Slave Mode 0: Slave mode 1: Master mode
[13]	SELM	Slave Select Mode 0: SEL signal is controlled by software – asserted or de-asserted by the SSELC bit 1: SEL signal is controlled by hardware – generated automatically by the SPI hardware Note that the SELM bit is available for master mode only – MODE = 1.
[12]	FIRSTBIT	LSB or MSB Transmitted First 0: MSB is transmitted first 1: LSB is transmitted first
[11]	SELAP	Slave Select Active Polarity 0: SEL signal is active low 1: SEL signal is active high

Bits	Field	Descriptions
[10:8]	FORMAT	SPI Data Transfer Format These three bits are used to determine the data transfer format of the SPI interface

FORMAT [2:0]	CPOL	CPHA
001	0	0
010	0	1
110	1	0
101	1	1
Others	Reserved	

CPOL: Clock Polarity
0: SCK Idle state is low
1: SCK Idle state is high
CPHA: Clock Phase
0: Data is captured on the first SCK clock edge
1: Data is captured on the second SCK clock edge

Bits	Field	Descriptions
[3:0]	DFL	Data Frame Length Selects the data transfer frame from 1 bit to 16 bits. 0x1: 1 bit 0x2: 2 bits … 0xF: 15 bits 0x0: 16 bits

SPI Interrupt Enable Register – SPIIER

This register contains the corresponding SPI interrupt enable control bit.

Offset: 0x008

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	22	21	20	19	18	17	16
	Reserved							
Type/Reset								
	15	14	13	12	11	10	9	8
	Reserved							
Type/Reset								

	7	6	5	4	3	2	1	0
	TOIEN	SAIEN	MFIEN	ROIEN	WCIEN	RXBNEIEN	TXEIEN	TXBEIEN
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0

Bits	Field	Descriptions
[7]	TOIEN	Time Out Interrupt Enable 0: Disable 1: Enable
[6]	SAIEN	Slave Abort Interrupt Enable 0: Disable 1: Enable
[5]	MFIEN	Mode Fault Interrupt Enable 0: Disable 1: Enable
[4]	ROIEN	Read Overrun Interrupt Enable 0: Disable 1: Enable
[3]	WCIEN	Write Collision Interrupt Enable 0: Disable 1: Enable
[2]	RXBNEIEN	RX Buffer Not Empty Interrupt Enable 0: Disable 1: Enable Generates an interrupt request when the RXBNE flag is set and when RXBNEIEN is set. In the FIFO mode, the interrupt request being generated depends upon the RX FIFO trigger level setting
[1]	TXEIEN	TX Empty Interrupt Enable 0: Disable 1: Enable The TX register empty interrupt request will be generated when the TXE flag and the TXEIEN bit are set

Bits	Field	Descriptions
[0]	TXBEIEN	TX Buffer Empty Interrupt Enable 0: Disable 1: Enable The TX buffer empty interrupt request will be generated when the TXBE flag and the TXBEIEN bit are set. In the FIFO mode, the interrupt request being generated depends upon the TX FIFO trigger level setting.

SPI Clock Prescaler Register – SPICPR

This register specifies the SPI clock prescaler ratio.

Offset: 0x00C

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	22	21	20	19	18	17	16
	Reserved							
Type/Reset								
	15	14	13	12	11	10	9	8
	CP							
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0
	7	6	5	4	3	2	1	0
	CP							
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0

Bits	Field	Descriptions
[15:0]	CP	SPI Clock Prescaler The SPI clock (SCK) is determined by the following equation: $f_{SCK} = f_{PCLK} / (2 \times (CP + 1))$, where the CP ranges is from 0 to 65535 Note: For the SPI slave mode, the system clock (f_{PCLK}) must be at least 3 times faster than the external SPI SCK input.

SPI Data Register – SPIDR

This register stores the SPI received or transmitted Data.

Offset: 0x010

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	22	21	20	19	18	17	16
	Reserved							
Type/Reset								
	15	14	13	12	11	10	9	8
	DR							
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0
	7	6	5	4	3	2	1	0
	DR							
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0

Bits	Field	Descriptions
[15:0]	DR	Data Register The SPI data register is used to store the serial bus transmitted or received data. In the non-FIFO mode, writing data into the SPI data register will also load the data into the data transmission buffer, known as the TX buffer. Reading data from the SPI data register will return the data held in the data received buffer, named RX buffer.

SPI Status Register – SPISR

This register contains the relevant SPI status.

Offset: 0x014

Reset value: 0x0000_0003

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	22	21	20	19	18	17	16
	Reserved							
Type/Reset								
	15	14	13	12	11	10	9	8
	Reserved							BUSY
Type/Reset								RO 0
	7	6	5	4	3	2	1	0
	TO	SA	MF	RO	WC	RXBNE	TXE	TXBE
Type/Reset	WC 0	WC 0	WC 0	WC 0	WC 0	RO 0	RO 1	RO 1

Bits	Field	Descriptions
[8]	BUSY	SPI Busy flag 0: SPI not busy 1: SPI busy In the master mode, this flag is reset when the TX buffer and TX shift register are both empty and is set when the TX buffer or the TX shift register are not empty. In the slave mode, this flag is set when SEL changes to an active level and is reset when SEL changes to an inactive level.
[7]	TO	Time Out flag 0: No RX FIFO time out 1: RX FIFO time out has occurred Write 1 to clear it. Once the timeout counter value is equal to the TOC field setting in the SPIFTOCR register, the time out flag will be set and an interrupt will be generated if the TOIEN bit in the SPIIER register is enabled. This bit is cleared by writing 1 Note: This Time Out flag function is only available in the SPI FIFO mode.
[6]	SA	Slave Abort flag 0: No slave abort 1: Slave abort has occurred This bit is set by hardware and cleared by writing 1.
[5]	MF	Mode Fault flag 0: No mode fault 1: Mode fault has occurred This bit is set by hardware and cleared by writing 1.
[4]	RO	Read Overrun flag 0: No read overrun 1: Read overrun has occurred This bit is set by hardware and cleared by writing 1.

Bits	Field	Descriptions
[3]	WC	Write Collision flag 0: No write collision 1: Write collision has occurred This bit is set by hardware and cleared by writing 1.
[2]	RXBNE	Receive Buffer Not Empty flag 0: RX buffer is empty 1: RX buffer not empty This bit indicates the RX buffer status in the non-FIFO mode. It is also used to indicate if the RX FIFO trigger level has been reached in the FIFO mode. This bit will be cleared when the SPI RX buffer is empty in the non-FIFO mode or if the number of data contained in RX FIFO is less than the trigger level which is specified by the RXFTLS field in the SPIFCR register in the SPI FIFO mode.
[1]	TXE	Transmission Register Empty flag 0: TX buffer or TX shift register is not empty 1: TX buffer and TX shift register both are empty
[0]	TXBE	Transmit Buffer Empty flag 0: TX buffer is not empty 1: TX buffer is empty In the FIFO mode, this bit, if set, indicates that the number of data contained in TX FIFO is equal to or less than the trigger level specified by the TXFTLS field in the SPIFCR register.

SPI FIFO Control Register – SPIFCR

This register contains the related SPI FIFO control including the FIFO enable control and the FIFO trigger level selections.

Offset: 0x018

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	22	21	20	19	18	17	16
	Reserved							
Type/Reset								
	15	14	13	12	11	10	9	8
	Reserved					FIFOEN		Reserved
Type/Reset						RW 0		
	7	6	5	4	3	2	1	0
	RXFTLS				TXFTLS			
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0

Bits	Field	Descriptions
[10]	FIFOEN	FIFO Enable 0: FIFO disable 1: FIFO enable This bit cannot be set or reset when the SPI interface is in transmitting.
[7:4]	RXFTLS	RX FIFO Trigger Level Select 0000: Trigger level is 0 0001: Trigger level is 1 ... 1000: Trigger level is 8 Others: Reserved The RXFTLS field is used to specify the RX FIFO trigger level. When the number of data contained in the RX FIFO is equal to or greater than the trigger level defined by the RXFTLS field, the RXBNE flag will be set
[3:0]	TXFTLS	TX FIFO Trigger Level Select 0000: Trigger level is 0 0001: Trigger level is 1 ... 1000: Trigger level is 8 Others: Reserved The TXFTLS field is used to specify the TX FIFO trigger level. When the number of data contained in the TX FIFO is equal to or less than the trigger level defined by the TXFTLS field, the TXBE flag will be set.

SPI FIFO Status Register – SPIFSR

This register contains the relevant SPI FIFO status.

Offset: 0x01C

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	22	21	20	19	18	17	16
	Reserved							
Type/Reset								
	15	14	13	12	11	10	9	8
	Reserved							
Type/Reset								
	7	6	5	4	3	2	1	0
	RXFS				TXFS			
Type/Reset	RO	RO	RO	RO	RO	RO	RO	RO
	0	0	0	0	0	0	0	0

Bits	Field	Descriptions
[7:4]	RXFS	RX FIFO Status 0000: RX FIFO empty 0001: RX FIFO contains 1 data ... 1000: RX FIFO contains 8 data Others: Reserved
[3:0]	TXFS	TX FIFO Status 0000: TX FIFO empty 0001: TX FIFO contains 1 data … 1000: TX FIFO contains 8 data Others: Reserved

SPI FIFO Time Out Counter Register – SPIFTOCR

This register stores the SPI RX FIFO timeout counter value.

Offset: 0x020

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	22	21	20	19	18	17	16
	Reserved							
Type/Reset								
	15	14	13	12	11	10	9	8
	TOC							
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0
	7	6	5	4	3	2	1	0
	TOC							
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0

Bits	Field	Descriptions
[15:0]	TOC	Time Out Counter The timeout counter starts to count from 0 after the SPI RX FIFO receives a data, and reset the counter value once the data is read from the SPIDR register by software or another new data is received. If the FIFO does not receive new data or the software does not read data from the SPIDR register the timeout counter value will continuously increase. When the timeout counter value is equal to the TOC setting value, the TO flag in the SPISR register will be set and an interrupt will be generated if the TOIEN bit in the SPIIEN register is set. The timeout counter will be stopped when the RX FIFO is empty. The SPI FIFO timeout function can be disabled by setting the TOC field to zero. The timeout counter is driven by the system APB clock, named f_{PCLK}.

20 Quad Serial Peripheral Interface (QSPI)

Introduction

The Quad Serial Peripheral Interface, QSPI, provides an SPI protocol data transmit and receive functions in both master and slave mode. The QSPI interface uses 6 pins, among which are serial data input and output lines MOSI / SIO0 and MISO / SIO1, SIO2, SIO3, the clock line SCK, and the slave select line SEL. One QSPI device acts as a master who controls the data flow using the SEL and SCK signals to indicate the start of the data communication and the data sampling rate. To receive the data bits, the streamlined data bits which range from 1 bit to 16 bits specified by the DFL field in the QSPICR1 register are latched on a specific clock edge and stored in the data register or in the RX FIFO. Data transmission is carried in a similar way but with the reverse sequence. The mode fault detection provides a capability for multi-master applications.

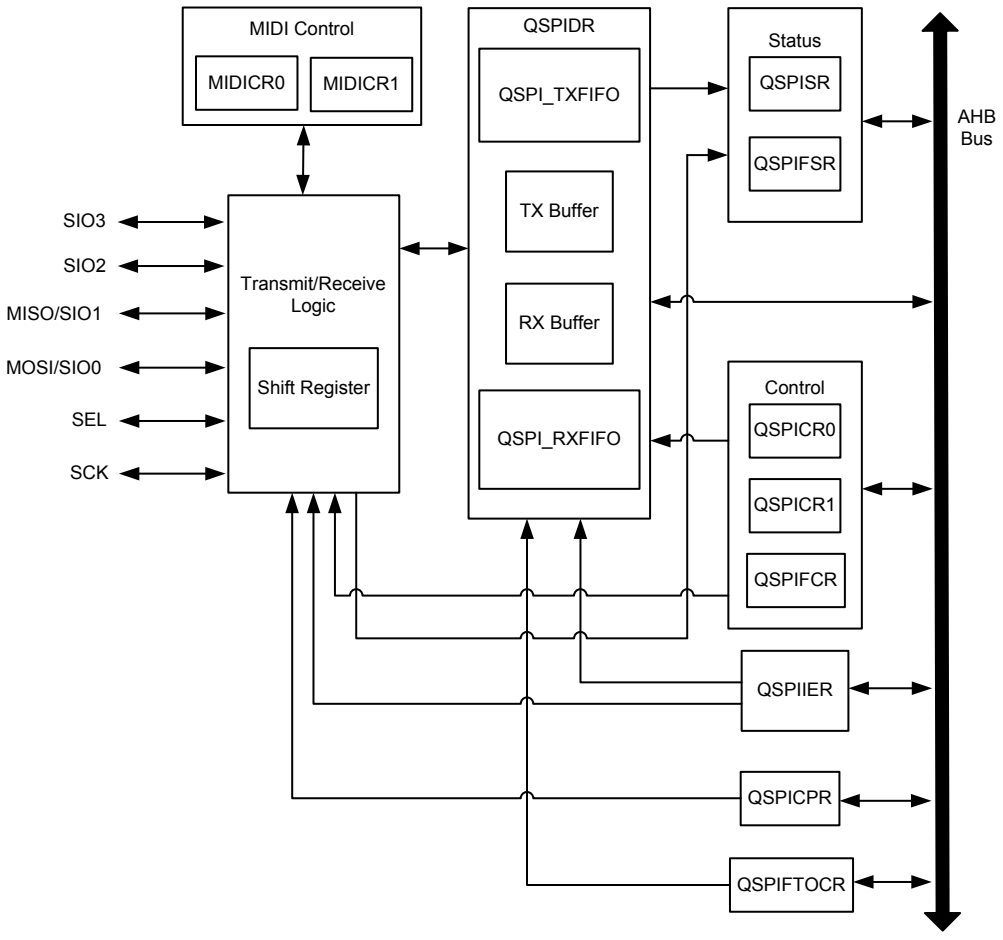


Figure 113. QSPI Block Diagram

Features

- Master or slave mode
- Master mode speed up to $f_{HCLK}/2$
- Slave mode speed up to $f_{HCLK}/3$
- Programmable data frame length up to 16 bits
- FIFO Depth: 8 levels
- MSB or LSB first shift selection
- Programmable slave select high or low active polarity
- Multi-master and multi-slave operation
- Master mode supports the dual / quad output read mode of QSPI series NOR Flash
- Four error flags with individual interrupt
  - Read overrun
  - Write collision
  - Mode fault
  - Slave abort
- Supports PDMA interface

Function Descriptions

Master Mode

Each data frame can range from 1 bit to 16 bits in data length. The first bit of the transmitted data can be either an MSB or LSB determined by the FIRSTBIT bit in the QSPICR1 register. The QSPI module is configured as a master or a slave by setting the MODE bit in the QSPICR1 register. When the MODE bit is set, the QSPI module is configured as a master and will generate the serial clock on the SCK pin. The data stream will transmit data in the shift register to the MOSI pin on the serial clock edge. The SEL pin is active during the full data transmission. When the SELAP bit in the QSPICR1 register is set, the SEL pin is active high during the complete data transactions. When the SELM bit in the QSPICR1 register is set, the SEL pin will be driven by the hardware automatically and the time interval between the active SEL edge and the first edge of SCK is equal to one half an SCK period.

Slave Mode

In the slave mode, the SCK pin acts as an input pin and the serial clock will be derived from the external master device. The SEL pin also acts as an input. When the SELAP bit is cleared to 0, the SEL signal is active low during the full data stream reception. When the SELAP bit is set to 1, the SEL signal will be active high during the full data stream reception.

Note: For the slave mode, the AHB clock, known as f_{HCLK}, must be at least 3 times faster than the external SCK clock input frequency.

QSPI Serial Frame Format

The QSPI interface format is based on the Clock Polarity, CPOL, and the Clock Phase, CPHA, configurations.

- Clock Polarity Bit – CPOL
  When the Clock Polarity bit is cleared to 0, the SCK line idle state is LOW. When the Clock Polarity bit is set to 1, the SCK line idle state is HIGH.

- Clock Phase Bit – CPHA
  When the Clock Phase bit is cleared to 0, the data is sampled on the first SCK clock transition. When the Clock Phase bit is set to 1, the data is sampled on the second SCK clock transition.

  There are four formats contained in the QSPI interface.Table 47 shows how to configure these formats by setting the FORMAT field in the QSPICR1 register.

Table 47. QSPI Interface Format Setup

FORMAT [2:0]	CPOL	CPHA
001	0	0
010	0	1
110	1	0
101	1	1
Others	Reserved	

CPOL = 0, CPHA = 0

In this format, the received data is sampled on the SCK line rising edge while the transmitted data is changed on the SCK line falling edge. In the master mode, the first bit is driven when data is written into the QSPIDR Register. In the slave mode, the first bit is driven when the SEL signal goes to an active level. Figure 114 shows the single byte data transfer timing of this format.

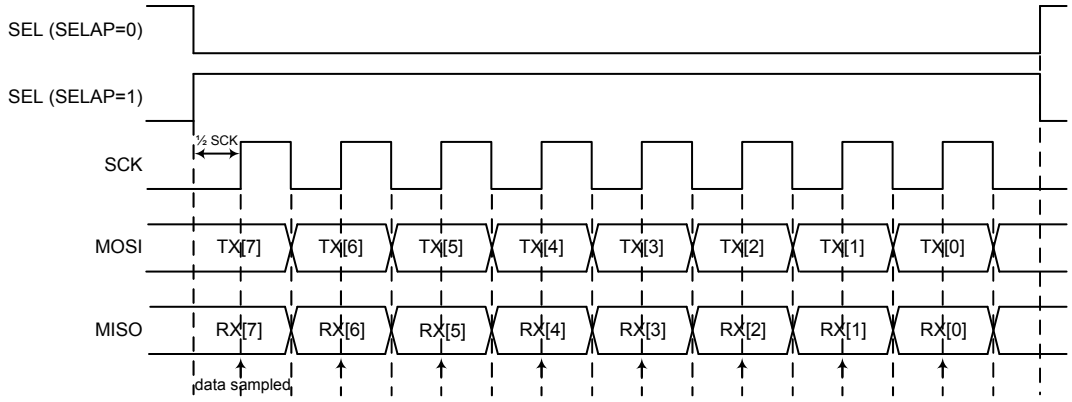


Figure 114. QSPI Single Byte Transfer Timing Diagram – CPOL = 0, CPHA = 0

Figure 115 shows the continuous data transfer timing diagram of this format. Note that the SEL signal must change to an inactive level between each data frame.

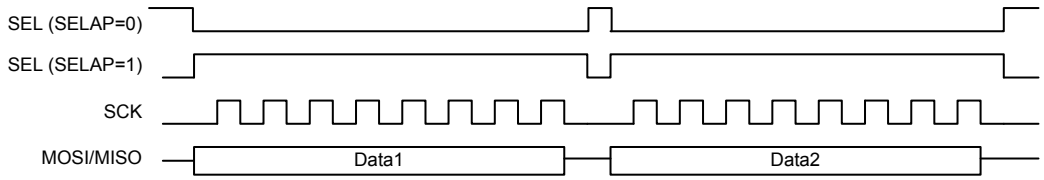


Figure 115. QSPI Continuous Transfer Timing Diagram – CPOL = 0, CPHA = 0

CPOL = 0, CPHA = 1

In this format, the received data is sampled on the SCK line falling edge while the transmitted data is changed on the SCK line rising edge. In the master mode, the first bit is driven when data is written into the QSPIDR register. In the slave mode, the first bit is driven at the first SCK clock rising edge. Figure 116 shows the single data byte transfer timing.

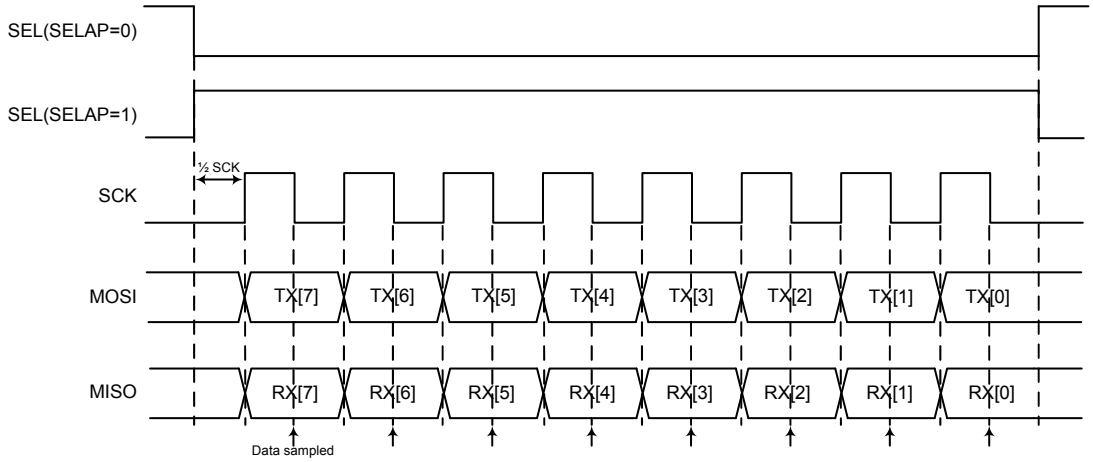


Figure 116. QSPI Single Byte Transfer Timing Diagram – CPOL = 0, CPHA = 1

Figure 117 shows the continuous data transfer diagram timing. Note that the SEL signal must remain active until the last data transfer has completed.

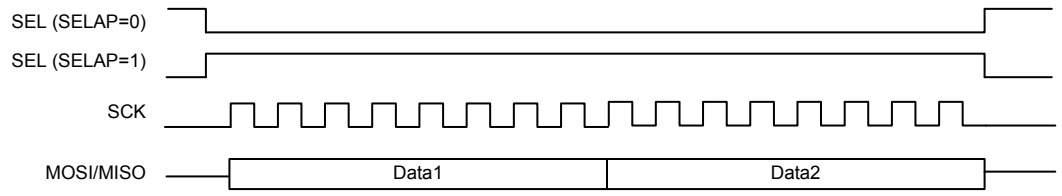


Figure 117. QSPI Continuous Transfer Timing Diagram – CPOL = 0, CPHA = 1

CPOL = 1, CPHA = 0

In this format, the received data is sampled on the SCK line falling edge while the transmitted data is changed on the SCK line rising edge. In the master mode, the first bit is driven when data is written into the QSPIDR register. In the slave mode, the first bit is driven when the SEL signal changes to an active level. Figure 118 shows the single byte transfer timing of this format.

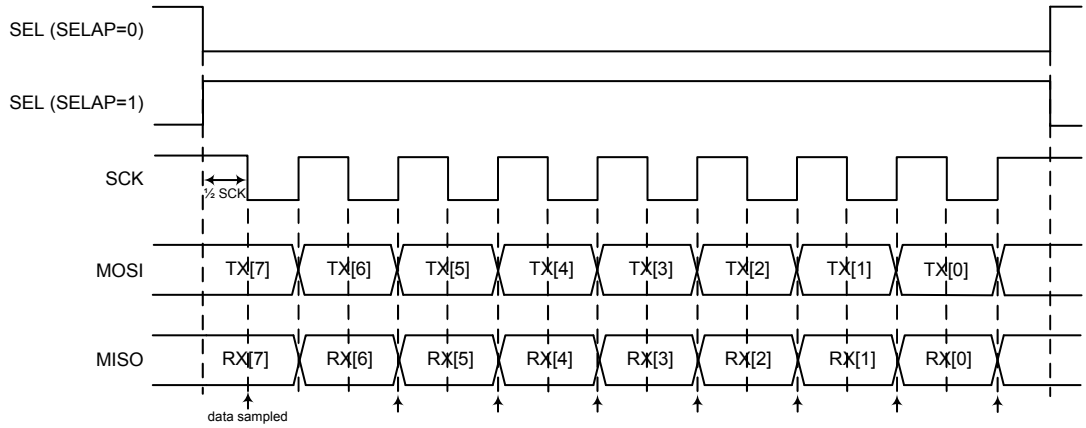


Figure 118. QSPI Single Byte Transfer Timing Diagram – CPOL = 1, CPHA = 0

Figure 119 shows the continuous data transfer timing of this format. Note that the SEL signal must change to an inactive level between each data frame.

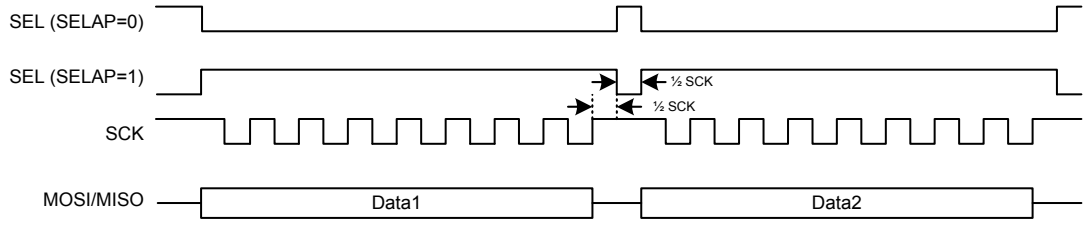


Figure 119. QSPI Continuous Transfer Timing Diagram – CPOL = 1, CPHA = 0

CPOL = 1, CPHA = 1

In this format, the received data is sampled on the SCK line rising edge while the transmitted data is changed on the SCK line falling edge. In the master mode, the first bit is driven when data is written into the QSPIDR register. In the slave mode, the first bit is driven at the first SCK falling edge. Figure 120 shows the single byte transfer timing of this format.

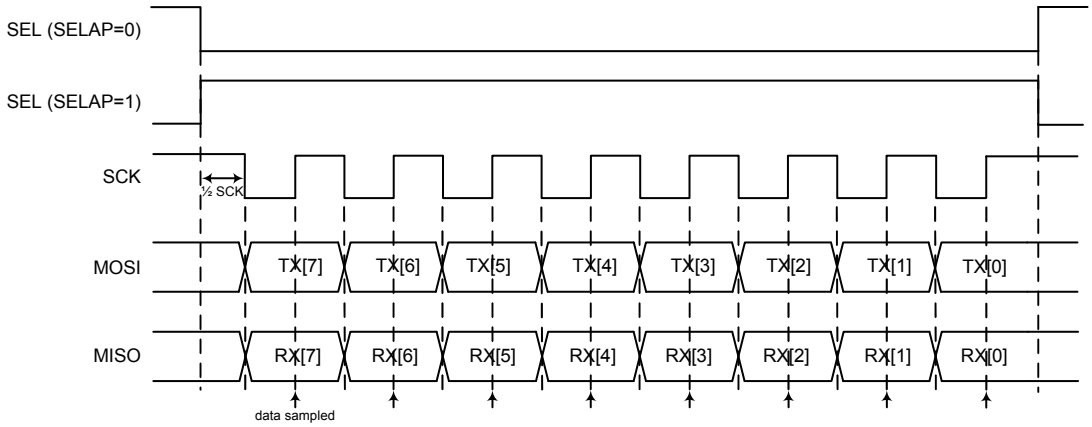


Figure 120. QSPI Single Byte Transfer Timing Diagram – CPOL = 1, CPHA = 1

Figure 121 shows the continuous data transfer timing of this format. Note that the SEL signal must remain active until the last data transfer has completed.

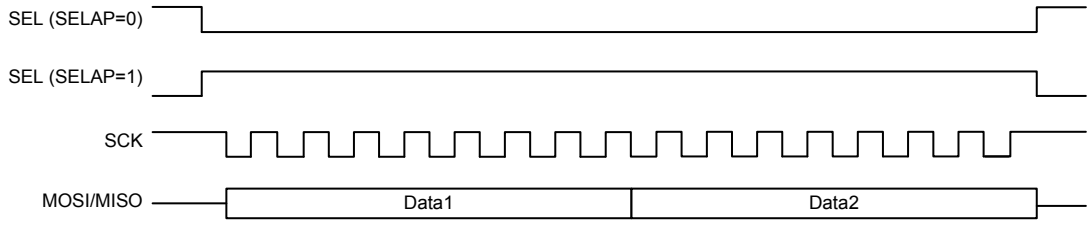


Figure 121. QSPI Continuous Transfer Timing Diagram – CPOL = 1, CPHA = 1

Status Flags

TX Buffer Empty – TXBE

This TXBE flag is set when the TX buffer is empty in the non-FIFO mode or when the TX FIFO data length is equal to or less than the TX FIFO threshold level as defined by the TXFTLS field in the QSPIFCR register in the FIFO mode. The following data to be transmitted can then be loaded into the buffer again. After this, the TXBE flag will be reset when the TX buffer already contains new data in the non-FIFO mode or when the TX FIFO data length is greater than the TX FIFO threshold level determined by the TXFTLS bits in the FIFO mode.

Transmission Register Empty – TXE

This TXE flag is set when both the TX buffer and the TX shift registers are empty. It will be reset when the TX buffer or the TX shift register contains new transmitted data.

RX Buffer Not Empty – RXBNE

This RXBNE flag is set when there is valid received data in the RX buffer in the non-FIFO mode or the RX FIFO data length is equal to or greater than the RX FIFO threshold level as defined by the RXFTLS field in the QSPIFCR register in the QSPI FIFO mode. This flag will be automatically cleared by hardware when the received data have been read out from the RX buffer totally in the non-FIFO mode or when the RX FIFO data length is less than the RX FIFO threshold level set in the RXFTLS field.

Time Out – TO

The time out function is only available in the QSPI FIFO mode and is disabled by loading a zero value into the TOC field in the Time Out Counter register. The timeout counter will start counting if the QSPI RX FIFO is not empty, once data is read from the QSPIDR register or new data is received, the timeout counter will be reset to 0 and count again. When the timeout counter value is equal to the value specified by the TOC field in the QSPIFTOCR register, the TO flag will be set. The flag is cleared by writing 1 to this bit.

Mode Fault – MF

The mode fault flag can be used to detect QSPI bus usage in the QSPI multi-master mode. For the multi-master mode, the QSPI module is configured as a master device and the SEL signal is setup as an input signal. The mode fault flag is set when the QSPI SEL pin is suddenly changed to an active level by another QSPI master. This means that another QSPI master is requesting to use the QSPI bus. Therefore, when a QSPI mode fault occurs, it will force the QSPI module to operate in the slave mode and also disable all of the QSPI interface signals to avoid QSPI bus signal collisions. For the same reason, if the QSPI master wants to transfer data, it also needs to inform other QSPI masters by driving its SEL signal to an active state. The detailed configuration diagram for the QSPI multi-master mode is shown in the following figure.

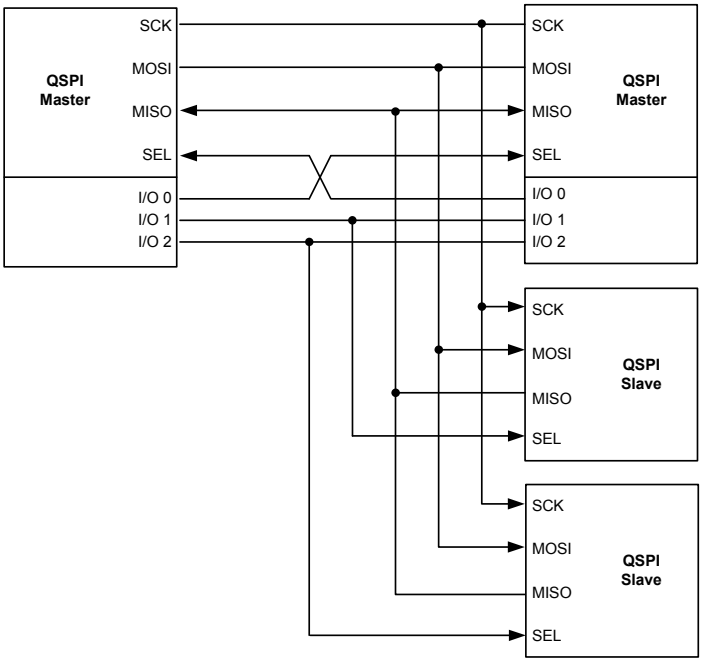


Figure 122. QSPI Multi-Master Slave Environment

Table 48. QSPI Mode Fault Trigger Conditions – for 1-bit serial mode only

Mode Fault	Descriptions
Trigger condition	QSPI Master mode SELOEN = 0 in the QSPICR0 register – SEL pin is configured to be the input mode SEL signal changes to an active level when driven by the external QSPI master
QSPI behavior	Mode fault flag is set. The QSPIEN bit in the QSPICR0 register is reset. This disables the QSPI interface and blocks all output signals from the device. The MODE bit in the QSPICR1 register is reset. This forces the device into slave mode.

Table 49. QSPI Master Mode SEL Pin Status

	SEL as Input – SELOEN = 0		SEL as Output – SELOEN = 1	
Multi-master	Support		Not support	
QSPI SEL control signal	Use another GPIO to replace the SEL pin function		SEL pin in hardware or software control mode - using SELM setting	
Continuous transfer	Case 1	Case 2	Case 1	Case 2
	Not support	Support	Hardware control	Hardware or software control

Case 1: SEL signal must be inactive between each data transfer.

Case 2: SEL signal will not to be inactive until the last data frame has finished.

Note: When the QSPI is in the slave mode, the SEL signal is always an input and not affected by the SELOEN bit in the QSPICR0 register.

Write Collision – WC

The following conditions will assert the Write Collision Flag.

- The FIFOEN bit in the QSPIFCR register is cleared
  The write collision flag is asserted when new data is written into the QSPIDR register while both the TX buffer and the shift register are already full. Any new data written into the TX buffer will be lost.
- The FIFOEN bit in the QSPIFCR register is set
  The write collision flag is asserted to indicate that new data is written into the QSPIDR register while both the TX FIFO and the TX shift register are already full. Any new data written into the TX FIFO will be lost.

Read Overrun – RO

- The FIFOEN bit in the QSPIFCR register is cleared
  The read overrun flag is asserted to indicate that both the RX shift register and the RX buffer are already full, if one more data is received. This will result in the newly received data not being shifted into the QSPI shift register. As a result the latest received data will be lost.
- The FIFOEN bit in the QSPIFCR register is set
  The read overrun flag is set to indicate that the RX shift register and the RX FIFO are both full, if one more data is received. This means that the latest received data can not be shifted into the QSPI shift register. As a result the latest received data will be lost.

Slave Abort – SA

In the QSPI slave mode, the slave abort flag is set to indicate that the SEL pin suddenly changed to an inactive state during the reception of a data frame transfer. The data frame length is set by the DFL field in the QSPICR1 register.

Dual SPI Mode – Generic

The QSPI also supports a Dual mode (2-bit) by setting the DUALEN bit to 1 and clearing the QUADEN to 0. The hardware configuration is similar to the one in the serial mode, the difference is that in the Dual mode there are two lines used for data transmission, which means that data is transmitted and received in two lines. The Dual mode only supports master mode.

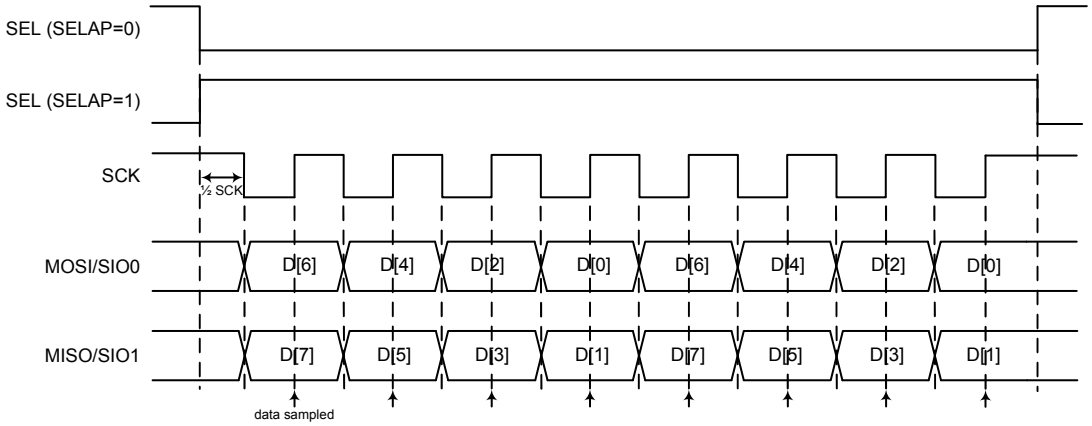


Figure 123. QSPI Dual Mode Bit Sequence – CPOL = 1, CPHA = 1, DFL = 0x8 (16-bit Data in total)

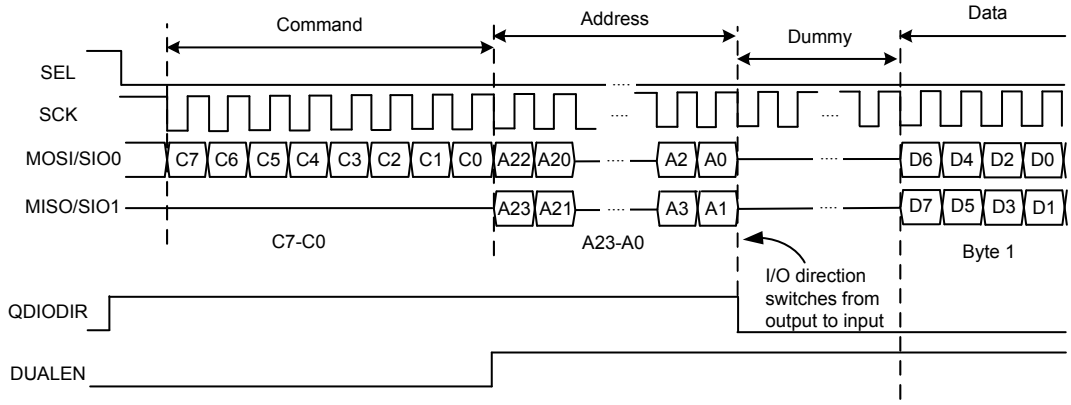


Figure 124. QSPI Dual Mode Data Read Example – CPOL = 1, CPHA = 1

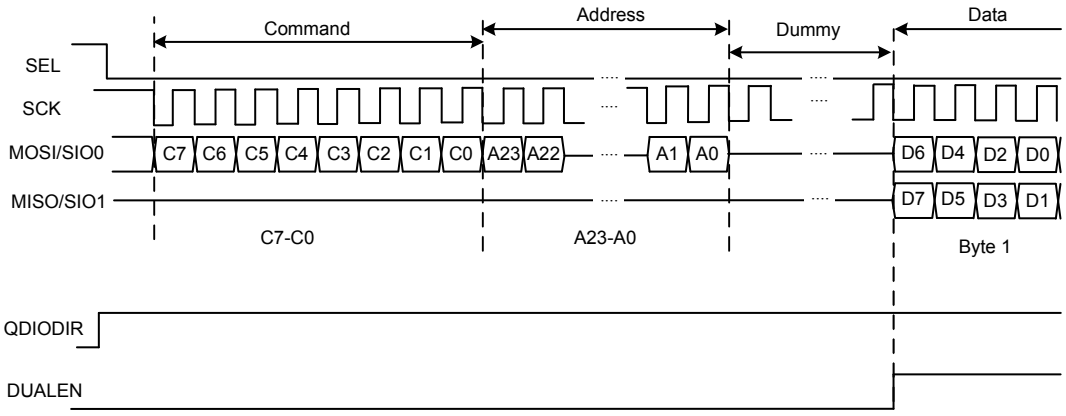


Figure 125. QSPI Dual Mode Data Write Example – CPOL = 1, CPHA = 1

Quad SPI Mode – Generic

The QSPI also supports a Quad mode (4-bit) by setting the QUADEN bit to 1 and clearing the DUALEN bit to 0. The hardware configuration is similar to the one in the serial mode, the difference is that in the Quad mode there are four lines used for data transmission, which means that data is transmitted and received in four lines. The Quad mode only supports master mode.

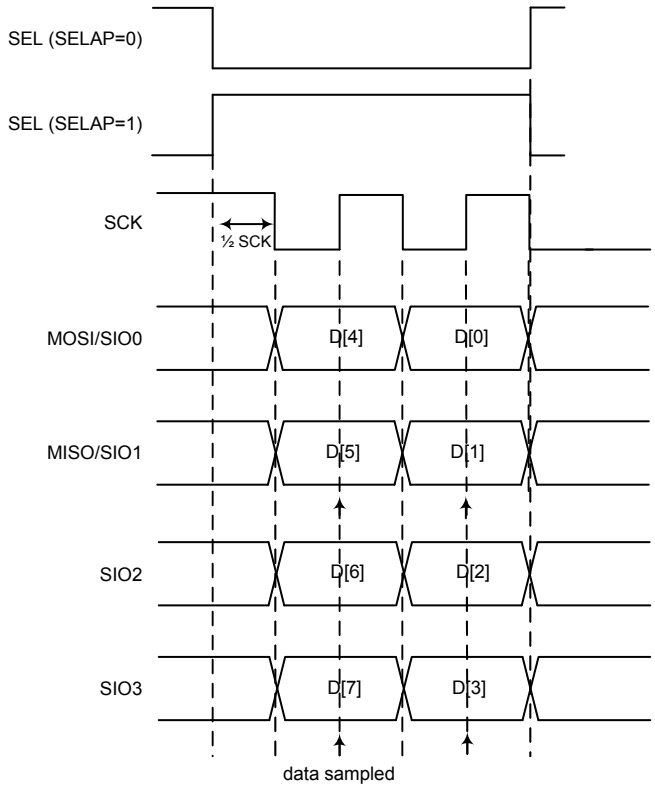


Figure 126. QSPI Quad Mode Bit Sequence – CPOL = 1, CPHA = 1, DFL = 0x2 (8-bit Data in Total)

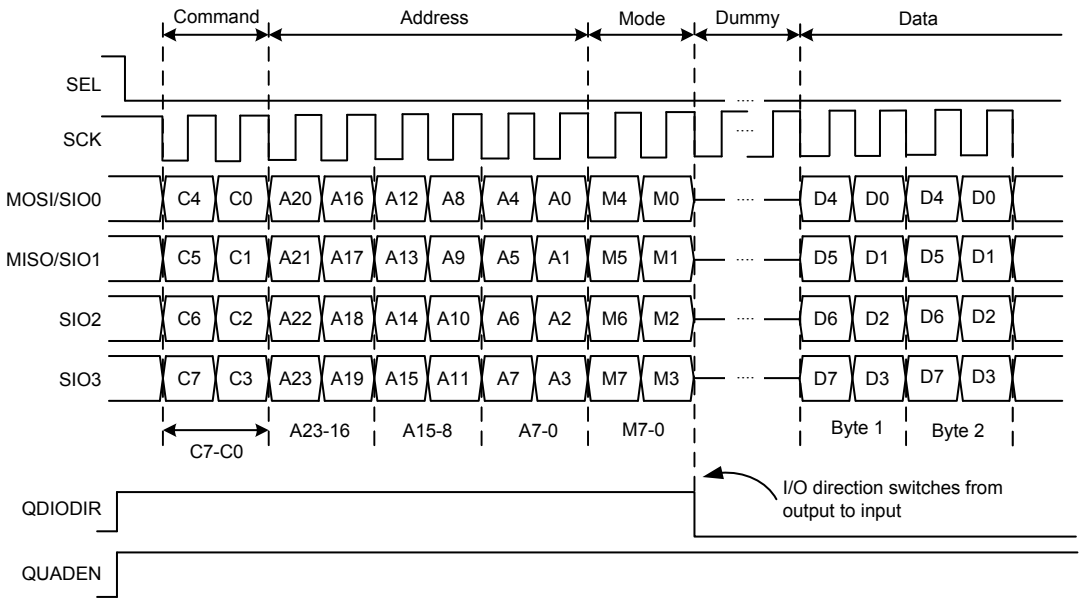


Figure 127. QSPI Quad Mode Data Read Example – CPOL = 1, CPHA = 1

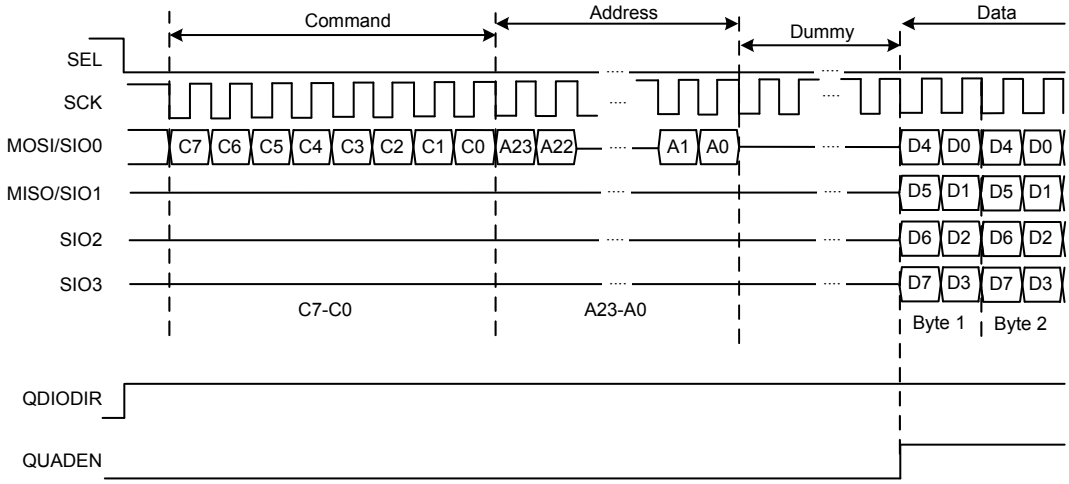


Figure 128. QSPI Quad Mode Data Write Example – CPOL = 1, CPHA = 1

Dual SPI Mode – MIDI Mode

When cooperated with an MIDI Engine, the QSPI can support a Dual mode (2-bit) if the SMDSEL[1:0] field value is set to "01". The QSPI communicates with MIDI Engine by properly configuring the MIDICR0 and MIDICR1 control Registers. Note that the MIDICEN must be set to 1.

The QSPI data transmission sequence is composed of five fields including Command, Address, Mode, Dummy and Data. The Address field content comes from the MIDI engine while the Data field content will be transmitted to the MIDI engine. It should be noted that when cooperated with an MIDI engine, the QSPI only supports Master mode and data read operation. In addition, only the configuration of CPOL = 0 and CPHA = 0 is available when QSPI is in the MIDI mode.

Table 50. QSPI Field Definition – MIDI Mode

	Command Field	Address Field	Mode Field	Dummy Field	Data Field
Cycle Count	CMDFL[Note]	ADFL[Note]	MDFL[Note]	DMFL[Note]	DATFL[Note]
Content	CMDVALUE[Note]	From MIDI Engine	MDVALUE[Note]	—	To MIDI Engine

Note: The actual value should be based on an external set of QSPI Flash Memory timing specifications.

Table 51. QSPI Dual Mode Setting – MIDI Mode

Mode	SMDSEL[1:0]	QDIOEN	2-bit Dual Mode Start Field
DIOR	01	1	Start from Address field
DOR	01	0	Start from Data field

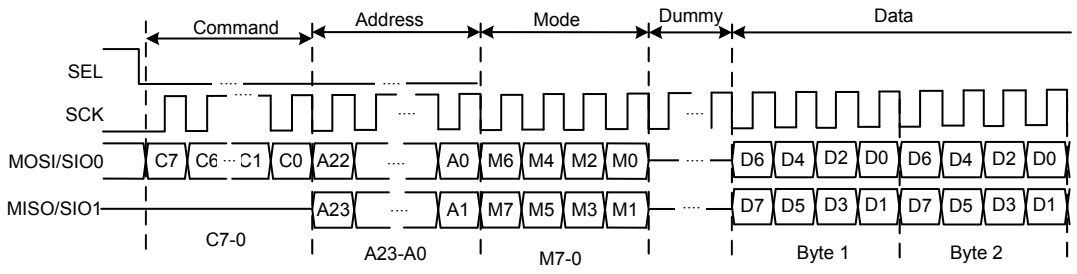


Figure 129. QSPI DIOR Mode Example

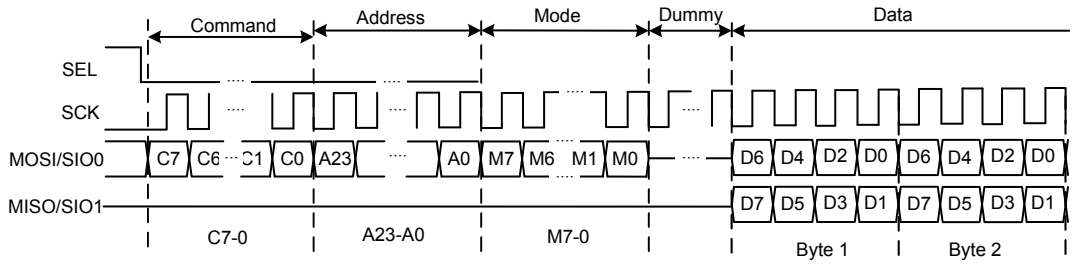


Figure 130. QSPI DOR Mode Example

Quad SPI Mode – MIDI Mode

When cooperated with MIDI Engine, the QSPI can support a Quad mode (4-bit) if the SMDSEL[1:0] field value is set to "10". The QSPI communicates with MIDI Engine by properly configuring the MIDICR0 and MIDICR1 control Registers. Note that the MIDICEN must be set to 1.

The Quad SPI mode operation is similar to the Dual SPI mode except that the data bit is 4-bit format in the Quad mode.

Table 52. QSPI Quad Mode Setting – MIDI Mode

Mode	SMDSEL[1:0]	QDIOEN	4-bit Quad Mode Start Field
QIOR	10	1	Start from Address field
QOR	10	0	Start from Data field
QPI	11	—	Start from Command field

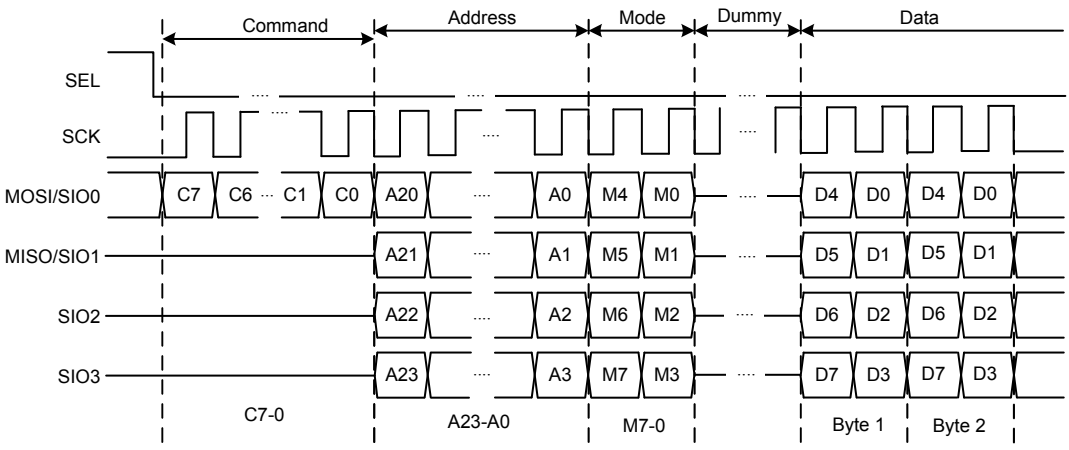


Figure 131. QSPI QIOR Mode Example

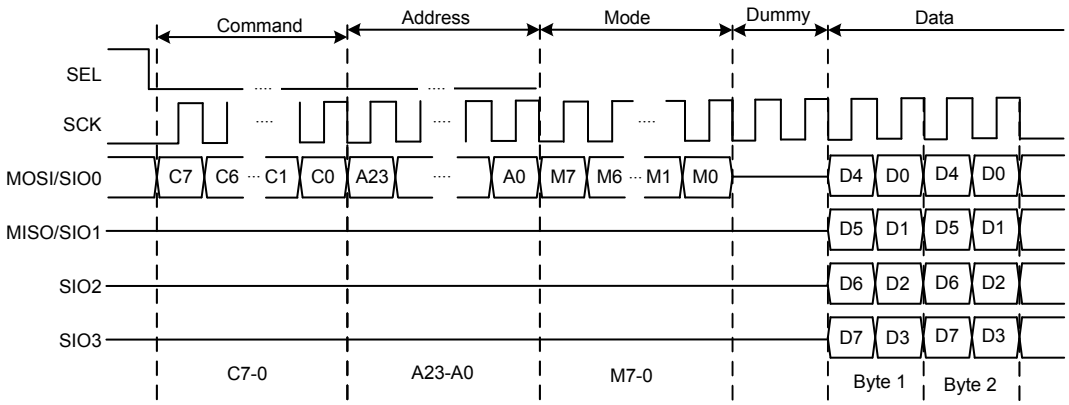


Figure 132. QSPI QOR Mode Example

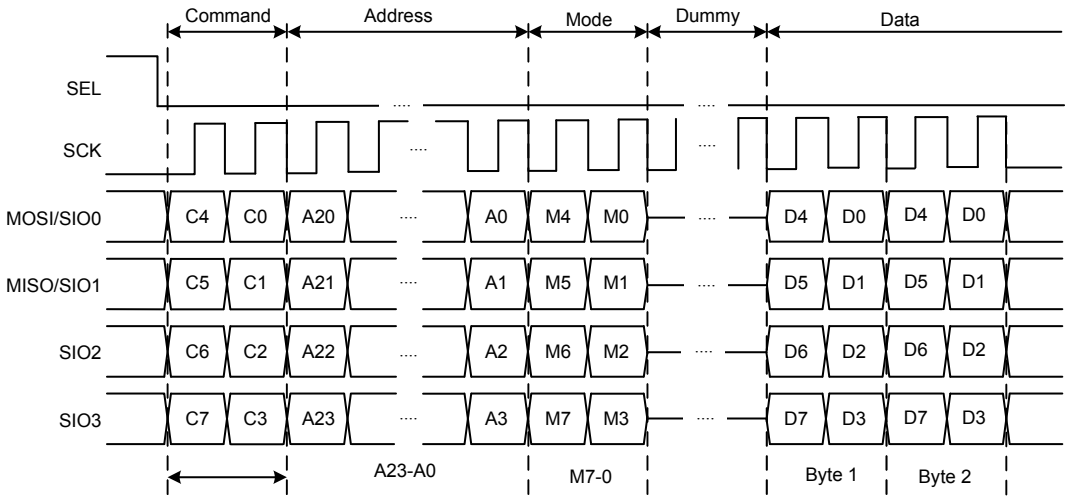


Figure 133. QSPI QPI Mode Example

PDMA Interface

The PDMA interface is integrated in the QSPI module. The PDMA function can be enabled by setting the TXDMAE or RXDMAE bit to 1 in the transmitter or receiver mode respectively. When the transmit buffer empty flag, TXBE, is asserted and the TXDMAE bit is set to 1, the PMDA function will be activated to move data from the memory location that users designated into the QSPI data register or the TX FIFO until the TXBE flag is cleared to 0. The TXBE flag will be asserted when the transmit buffer is empty in the non-FIFO mode or the data contained in the TX FIFO is equal to or less than the level defined by the TXFTLS field in the FIFO mode.

Similarly, when the receive buffer not empty flag, RXBNE, is asserted and the RXDMAE bit is set to 1, the PDMA function will be activated to move data from the QSPI data register or the RX FIFO to the memory location that users designated until the RXBNE flag is cleared to 0. The RXBNE flag will be asserted when the receive buffer is not empty in the non-FIFO mode or the data contained in the RX FIFO is equal to or greater than the level defined by the RXFTLS field in the FIFO mode.

For more detailed description on the PDMA configurations, refer to the PDMA chapter.

Register Map

The following table shows the QSPI registers and their reset values.

Table 53. QSPI Register Map

Register	Offset	Description	Reset Value
QSPICR0	0x000	QSPI Control Register 0	0x0000_0000
QSPICR1	0x004	QSPI Control Register 1	0x0000_0000
QSPIIER	0x008	QSPI Interrupt Enable Register	0x0000_0000
QSPICPR	0x00C	QSPI Clock Prescaler Register	0x0000_0000
QSPIDR	0x010	QSPI Data Register	0x0000_0000
QSPISR	0x014	QSPI Status Register	0x0000_0003
QSPIFCR	0x018	QSPI FIFO Control Register	0x0000_0000
QSPIFSR	0x01C	QSPI FIFO Status Register	0x0000_0000
QSPIFTOCR	0x020	QSPI FIFO Time Out Counter Register	0x0000_0000
MIDICR0	0x040	MIDI Control Register 0	0x0000_0000
MIDICR1	0x044	MIDI Control Register 1	0x0000_0000

Register Descriptions

QSPI Control Register 0 – QSPICR0

This register specifies the SEL control and the QSPI enable bits.

Offset: 0x000

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	22	21	20	19	18	17	16
	Reserved						QUADEN	QDIODIR
Type/Reset							RW 0	RW 0
	15	14	13	12	11	10	9	8
	SELHT				GUADT			
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0
	7	6	5	4	3	2	1	0
	GUADTEN	DUALEN	LOOPBACK	SSELC	SELOEN	RXDMAE	TXDMAE	QSPIEN
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0

Bits	Field	Descriptions
[17]	QUADEN	Quad Port Enable 0: Quad port is disabled 1: Quad port is enabled The control bit is used to support the quad output read / write mode of the series QSPI NOR Flash. When this bit is set, the SIO0 to SIO3 signals will change the direction according to QDIODIR to send or receive the series data stream. The QUADEN control bit is for master mode only.

Bits	Field	Descriptions
[16]	QDIODIR	I/O Direction in Dual or Quad mode 0: Input direction 1: Output direction The control bit is used to change the SIOn direction in dual or quad mode. When this bit is set, SIO0 to SIO3 will change to output direction to send data. On the contrary, when this bit is cleared to zero, SIO0 to SIO3 will change to input direction to receive data.
[15:12]	SELHT	Chip Select Hold Time 0x0: 1/ 2 SCK 0x1: 1 SCK 0x2: 3/2 SCK 0x3: 2 SCK .... Note that SELHT is for master mode only.
[11:8]	GUADT	Guard Time GUADTEN = 1 0x0: 1 SCK 0x1: 2 SCK 0x2: 3 SCK ... Note that GUADT is for master mode only.
[7]	GUADTEN	Guard Time Enable 0: Guard Time is 1/2 SCK 1: When this bit is set, Guard time can be controlled by GUADT Note that GUADTEN is for master mode only.
[6]	DUALEN	Dual Port Enable 0: Dual port is disabled 1: Dual port is enabled The control bit is used to support the dual output read / write mode of the series QSPI NOR Flash. When this bit is set and the MOSI signal will change the direction according to QDIODIR and send or receive the series data stream. That means the DUALEN control bit is only for master mode.
[5]	LOOPBACK	Loop Back Test Mode
[4]	SSELC	Software Slave Select Control 0: Set the SEL output to an inactive state 1: Set the SEL output to an active state The application software can setup the SEL output to an active or inactive state by configuring the SSELC bit. The active level is configured by the SELAP bit in the QSPICR1 register. Note that the SSELC bit is only available when the SELOEN bit is set to 1 for enabling the SEL output meanwhile the SELM bit is cleared to 0 for controlling the SEL signal by software. Otherwise, the SSELC bit has no effect.
[3]	SELOEN	Slave Select Output Enable 0: Set the SEL signal to the input mode for multi-master mode 1: Set the SEL signal to the output mode for slave select The SELOEN is only available in the master mode to setup the SEL signal as an input or output signal. When the SEL signal is configured to operate in the output mode, it is used as a slave select signal in either the hardware or software mode according to the SELM bit setting in the QSPICR1 register. The SEL signal is used for mode fault detection in the multi-master environment when it is configured to operate in the input mode.

Bits	Field	Descriptions
[2]	RXDMAE	RX PDMA request enable 0: QSPI RX path PDMA request is disabled 1: QSPI RX path PDMA request is enabled
[1]	TXDMAE	TX PDMA request enable 0: QSPI TX path PDMA request is disabled 1: QSPI TX path PDMA request is enabled
[0]	QSPIEN	QSPI Enable 0: QSPI interface is disabled 1: QSPI interface is enabled

QSPI Control Register 1 – QSPICR1

This register specifies the QSPI parameters including the data length, the transfer format, the SEL active polarity / mode, the LSB / MSB control and the master / slave mode.

Offset: 0x004

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	22	21	20	19	18	17	16
	Reserved							
Type/Reset								
	15	14	13	12	11	10	9	8
	Reserved	MODE	SELM	FIRSTBIT	SELAP	FORMAT		
Type/Reset		RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0
	7	6	5	4	3	2	1	0
	Reserved				DFL			
Type/Reset					RW 0	RW 0	RW 0	RW 0

Bits	Field	Descriptions
[14]	MODE	Master or Slave Mode 0: Slave mode 1: Master mode
[13]	SELM	Slave Select Mode 0: SEL signal is controlled by software – asserted or de-asserted by the SSELC bit 1: SEL signal is controlled by hardware – generated automatically by the QSPI hardware Note that SELM bit is available for master mode only – MODE = 1.
[12]	FIRSTBIT	LSB or MSB Transmitted First 0: MSB is transmitted first 1: LSB is transmitted first
[11]	SELAP	Slave Select Active Polarity 0: SEL signal is active low 1: SEL signal is active high

Bits	Field	Descriptions
[10:8]	FORMAT	QSPI Data Transfer Format These three bits are used to determine the data transfer format of the QSPI interface

FORMAT [2:0]	CPOL	CPHA
001	0	0
010	0	1
110	1	0
101	1	1
Others	Reserved	

CPOL: Clock Polarity
0: SCK Idle state is low
1: SCK Idle state is high
CPHA: Clock Phase
0: Data is captured on the first SCK clock edge
1: Data is captured on the second SCK clock edge

[3:0] DFL

Data Frame Length
Selects the data transfer frame from 1 bit to 16 bits.
0x1: 1 bit
0x2: 2 bits
…
0xF: 15 bits
0x0: 16 bits
Notes:
1. The total number of data bits is determined by the DFL field configuration together with the selected QSPI device transmission mode.
Taking the 8-bit data transmission for example, the DFL setting can be figured out as follows
In the Serial mode: Data frame length = 8/1 = 8, DFL = 0x8;
In the Dual mode: Data frame length = 8/2 = 4, DFL = 0x4;
In the Quad mode: Data frame length = 8/4 = 2, DFL = 0x2.
2. Only 0x4, 0x8 and 0x0 are available when DUALEN = 1.
3. Only 0x2, 0x4 and 0x8 are available when QUADEN = 1.

QSPI Interrupt Enable Register – QSPIIER

This register contains the corresponding QSPI interrupt enable control bit.

Offset: 0x008

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	22	21	20	19	18	17	16
	Reserved							
Type/Reset								
	15	14	13	12	11	10	9	8
	Reserved							
Type/Reset								

	7	6	5	4	3	2	1	0
	TOIEN	SAIEN	MFIEN	ROIEN	WCIEN	RXBNEIEN	TXEIEN	TXBEIEN
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0

Bits	Field	Descriptions
[7]	TOIEN	Time Out Interrupt Enable 0: Disable 1: Enable
[6]	SAIEN	Slave Abort Interrupt Enable 0: Disable 1: Enable
[5]	MFIEN	Mode Fault Interrupt Enable 0: Disable 1: Enable
[4]	ROIEN	Read Overrun Interrupt Enable 0: Disable 1: Enable
[3]	WCIEN	Write Collision Interrupt Enable 0: Disable 1: Enable
[2]	RXBNEIEN	RX Buffer Not Empty Interrupt Enable 0: Disable 1: Enable Generates an interrupt request when the RXBNE flag is set and when RXBNEIEN is set. In the FIFO mode, the interrupt request being generated depends upon the RX FIFO trigger level setting.
[1]	TXEIEN	TX Empty Interrupt Enable 0: Disable 1: Enable The TX register empty interrupt request will be generated when the TXE flag and the TXEIEN bit are set.

Bits	Field	Descriptions
[0]	TXBEIEN	TX Buffer Empty Interrupt Enable 0: Disable 1: Enable The TX buffer empty interrupt request will be generated when the TXBE flag and the TXBEIEN bit are set. In the FIFO mode, the interrupt request being generated depends upon the TX FIFO trigger level setting.

QSPI Clock Prescaler Register – QSPICPR

This register specifies the QSPI clock prescaler ratio.

Offset: 0x00C

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	22	21	20	19	18	17	16
	Reserved							
Type/Reset								
	15	14	13	12	11	10	9	8
	CP							
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0
	7	6	5	4	3	2	1	0
	CP							
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0

Bits	Field	Descriptions
[15:0]	CP	QSPI Clock Prescaler The QSPI clock (SCK) is determined by the following equation: $f_{SCK} = f_{HCLK} / (2 \times (CP + 1))$, where the CP ranges is from 0 to 65535 Note: For the QSPI slave mode, the system clock (f_{HCLK}) must be at least 3 times faster than the external QSPI SCK input.

QSPI Data Register – QSPIDR

This register stores the QSPI received or transmitted Data.

Offset: 0x010

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	22	21	20	19	18	17	16
	Reserved							
Type/Reset								
	15	14	13	12	11	10	9	8
	DR							
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0
	7	6	5	4	3	2	1	0
	DR							
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0

Bits	Field	Descriptions
[15:0]	DR	Data Register The QSPI data register is used to store the serial bus transmitted or received data. In the non-FIFO mode, writing data into the QSPI data register will also load the data into the data transmission buffer, known as the TX buffer. Reading data from the QSPI data register will return the data held in the data received buffer, named RX buffer.

QSPI Status Register – QSPISR

This register contains the relevant QSPI status.

Offset: 0x014

Reset value: 0x0000_0003

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	22	21	20	19	18	17	16
	Reserved							
Type/Reset								
	15	14	13	12	11	10	9	8
	Reserved							BUSY
Type/Reset								RO 0
	7	6	5	4	3	2	1	0
	TO	SA	MF	RO	WC	RXBNE	TXE	TXBE
Type/Reset	WC 0	WC 0	WC 0	WC 0	WC 0	RO 0	RO 1	RO 1

Bits	Field	Descriptions
[8]	BUSY	QSPI Busy flag 0: QSPI not busy 1: QSPI busy In the master mode, this flag is reset when the TX buffer and TX shift register are both empty and is set when the TX buffer or the TX shift register are not empty. In the slave mode, this flag is set when SEL changes to an active level and is reset when SEL changes to an inactive level.
[7]	TO	Time Out flag 0: No RX FIFO time out 1: RX FIFO time out has occurred Once the timeout counter value is equal to the TOC field setting in the QSPIFTOCR register, the time out flag will be set and an interrupt will be generated if the TOIEN bit in the QSPIIER register is enabled. This bit is cleared by writing 1 Note: This Time Out flag function is only available in the QSPI FIFO mode.
[6]	SA	Slave Abort flag 0: No slave abort 1: Slave abort has occurred This bit is set by hardware and cleared by writing 1.
[5]	MF	Mode Fault flag 0: No mode fault 1: Mode fault has occurred This bit is set by hardware and cleared by writing 1.
[4]	RO	Read Overrun flag 0: No read overrun 1: Read overrun has occurred This bit is set by hardware and cleared by writing 1.
[3]	WC	Write Collision flag 0: No write collision 1: Write collision has occurred This bit is set by hardware and cleared by writing 1.
[2]	RXBNE	Receive Buffer Not Empty flag 0: RX buffer is empty 1: RX buffer is not empty This bit indicates the RX buffer status in the non-FIFO mode. It is also used to indicate if the RX FIFO trigger level has been reached in the FIFO mode. This bit will be cleared when the QSPI RX buffer is empty in the non-FIFO mode or if the number of data contained in RX FIFO is less than the trigger level which is specified by the RXFTLS field in the QSPIFCR register in the QSPI FIFO mode.
[1]	TXE	Transmission Register Empty flag 0: TX buffer or TX shift register is not empty 1: TX buffer and TX shift register both are empty
[0]	TXBE	Transmit Buffer Empty flag 0: TX buffer is not empty 1: TX buffer is empty In the FIFO mode, this bit, if set, indicates that the number of data contained in TX FIFO is equal to or less than the trigger level specified by the TXFTLS field in the QSPIFCR register.

QSPI FIFO Control Register – QSPIFCR

This register contains the related QSPI FIFO control including the FIFO enable control and the FIFO trigger level selections.

Offset: 0x018

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	22	21	20	19	18	17	16
	Reserved							
Type/Reset								
	15	14	13	12	11	10	9	8
	Reserved					FIFOEN		Reserved
Type/Reset						RW 0		
	7	6	5	4	3	2	1	0
	RXFTLS				TXFTLS			
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0

Bits	Field	Descriptions
[10]	FIFOEN	FIFO Enable 0: FIFO disable 1: FIFO enable This bit cannot be set or reset when the QSPI interface is in transmitting.
[7:4]	RXFTLS	RX FIFO Trigger Level Select 0000: Trigger level is 0 0001: Trigger level is 1 ... 1000: Trigger level is 8 Others: Reserved The RXFTLS field is used to specify the RX FIFO trigger level. When the number of data contained in the RX FIFO is equal to or greater than the trigger level defined by the RXFTLS field, the RXBNE flag will be set
[3:0]	TXFTLS	TX FIFO Trigger Level Select 0000: Trigger level is 0 0001: Trigger level is 1 ... 1000: Trigger level is 8 Others: Reserved The TXFTLS field is used to specify the TX FIFO trigger level. When the number of data contained in the TX FIFO is equal to or less than the trigger level defined by the TXFTLS field, the TXBE flag will be set.

QSPI FIFO Status Register – QSPIFSR

This register contains the relevant QSPI FIFO status.

Offset: 0x01C

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	22	21	20	19	18	17	16
	Reserved							
Type/Reset								
	15	14	13	12	11	10	9	8
	Reserved							
Type/Reset								
	7	6	5	4	3	2	1	0
	RXFS				TXFS			
Type/Reset	RO	RO	RO	RO	RO	RO	RO	RO
	0	0	0	0	0	0	0	0

Bits	Field	Descriptions
[7:4]	RXFS	RX FIFO Status 0000: RX FIFO empty 0001: RX FIFO contains 1 data ... 1000: RX FIFO contains 8 data Others: Reserved
[3:0]	TXFS	TX FIFO Status 0000: TX FIFO empty 0001: TX FIFO contains 1 data … 1000: TX FIFO contains 8 data Others: Reserved

QSPI FIFO Time Out Counter Register – QSPIFTOCR

This register stores the QSPI RX FIFO timeout counter value.

Offset: 0x020

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	22	21	20	19	18	17	16
	Reserved							
Type/Reset								
	15	14	13	12	11	10	9	8
	TOC							
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0
	7	6	5	4	3	2	1	0
	TOC							
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0

Bits	Field	Descriptions
[15:0]	TOC	Time Out Counter The timeout counter starts to count from 0 after the QSPI RX FIFO receives a data, and reset the counter value once the data is read from the QSPIDR register by software or another new data is received. If the FIFO does not receive new data or the software does not read data from the QSPIDR register the timeout counter value will continuously increase. When the timeout counter value is equal to the TOC setting value, the TO flag in the QSPISR register will be set and an interrupt will be generated if the TOIEN bit in the QSPIIEN register is set. The timeout counter will be stopped when the RX FIFO is empty. The QSPI FIFO timeout function can be disabled by setting the TOC field to zero. The timeout counter is driven by the system AHB clock, named f_{HCLK}

MIDI Control Register 0 – MIDICR0

This register stores the MIDI control and enable bits.

Offset: 0x040

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved				CMDFL			
Type/Reset					RW 0	RW 0	RW 0	RW 0
	23	22	21	20	19	18	17	16
	MIDICEN	QDIOEN	ADFL					
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0
	15	14	13	12	11	10	9	8
	MDFL				DMFL			
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0
	7	6	5	4	3	2	1	0
		SMDSEL	DATFL					
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0

Bits	Field	Descriptions
[27:24]	CMDFL	QSPI Flash Command Field Length 0x0: Disable 0x1: 1 cycle 0x2: 2 cycles … 0x8: 8 cycles … These bits are used to specify the number of clock cycles in the Flash command field. When the number of clock cycle is equal to 0, this field is disabled.
[23]	MIDICEN	MIDI Controller Enable 0: Disable 1: Enable The control bit is used to enable the MIDI controller and is only for Master mode.

Bits	Field	Descriptions
[22]	QDIOEN	QSPI Flash Dual / Quad Mode Enable for Address, Mode and Dummy Fields 0: Disable 1: Enable If this bit is set, the QSPI Flash Address field, Mode field and Dummy field would also transmit in Dual / Quad mode. Otherwise, only the Data field would transmit in Dual / Quad mode. Dual / Quad mode can be selected by the SMDSEL field. Examples: If the QDIOEN bit is cleared to 0 and the SMDSEL bit field value is set to “01” to select the Dual mode, the transmission mode is DOR, which means data lines is 1-line in the initial, and then will be 2-line format starting from the Data field. If the QDIOEN bit is set to 1 and the SMDSEL bit field value is set to “01” to select the Dual mode, the transmission mode is DIOR, which means data lines is 1-line in the initial, and then will be 2-line starting from the Address field. If the QDIOEN bit is cleared to 0 and the SMDSEL bit field value is set to “10” to select the Quad mode, the transmission mode is QOR, which means data lines is 1-line in the initial, and then will be 4-line starting from the Data field. If the QDIOEN bit is set to 1 and the SMDSEL bit field value is set to “10” to select the Quad mode, the transmission mode is QIOR, which means data lines is 1-line in the initial, and then will be 4-line starting from the Address field.
[21:16]	ADFL	QSPI Flash Address Field Length 0x0: Disable 0x1: 1 cycle 0x2: 2 cycles … 0x18: 24 cycles … These bits are used to specify the number of clock cycles in the address field. When the number of clock cycle is equal to 0, this field is disabled.
[15:12]	MDFL	QSPI Flash Mode Field Length 0x0: Disable 0x1: 1 cycle 0x2: 2 cycles … 0x8: 8 cycles … These bits are used to specify the number of clock cycles in the mode field. When the number of clock cycle is equal to 0, this field is disabled.
[11:8]	DMFL	QSPI Flash Dummy Field Length 0x0: Disable 0x1: 1 cycle 0x2: 2 cycles … 0x8: 8 cycles … These bits are used to specify the number of clock cycles in the dummy field. When the number of clock cycle is equal to 0, this field is disabled.
[7:6]	SMDSEL	QSPI Flash Mode selection 00: Serial mode 01: Dual mode 10: Quad mode 11: QPI mode

Bits	Field	Descriptions
[5:0]	DATFL	QSPI Flash Data Field Length 0x0: Disable 0x1: 1 cycle 0x2: 2 cycles … 0x20: 32 cycles … These bits are used to specify the number of clock cycles in the data field. When the number of clock cycle is equal to 0, this field is disabled.

MIDI Control Register 1 – MIDICR1

This register stores the QSPI Flash Command field and Mode field values.

Offset: 0x044
Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	22	21	20	19	18	17	16
	Reserved							
Type/Reset								
	15	14	13	12	11	10	9	8
	CMDVALUE							
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0
	7	6	5	4	3	2	1	0
	MDVALUE							
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0

Bits	Field	Descriptions
[15:8]	CMDVALUE	QSPI Flash Command Field Value This bit field is used to store the serial NOR or NAND Flash command. When the MIDI controller enable bit is set, the command would be sent by the MIDI controller automatically. The Flash Command field length should be configured appropriately. If the Flash Command field length is set to 0, this value would be ignored.
[7:0]	MDVALUE	QSPI Flash Mode Field Value This bit field is used to store the mode bit field value of Flash. When the MIDI controller enable bit is set, the value would be sent after the address field by the MIDI controller.

21 Universal Synchronous Asynchronous Receiver Transmitter (USART)

Introduction

The Universal Synchronous Asynchronous Receiver Transceiver, USART, provides a flexible full duplex data exchange using synchronous or asynchronous transfer. The USART is used to translate data between parallel and serial interfaces, and is also commonly used for RS232 standard communication. The USART peripheral function supports a variety of interrupts.

The USART module includes an 8-byte transmit FIFO, TX FIFO, and a 8-byte receive FIFO, RX FIFO. Software can detect a USART error status by reading USART Status & Interrupt Flag Register, USRSIFR. The status includes the condition of the transfer operations as well as several error conditions resulting from Parity, Overrun, Framing and Break events.

The USART includes a programmable baud rate generator which is capable of dividing the USART clock of the CK_APB (CK_USART) to produce a baud rate clock for the USART transmitter and receiver.

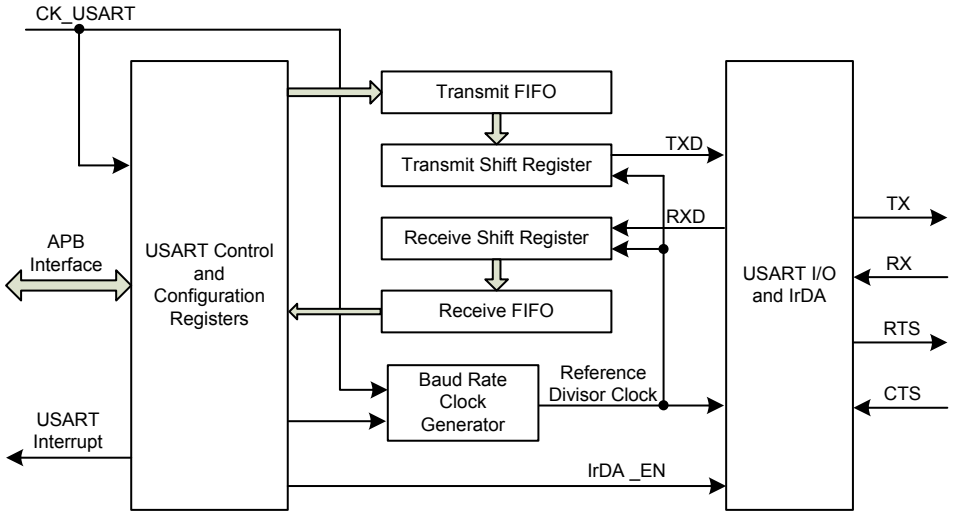


Figure 134. USART Block Diagram

Features

- Supports both asynchronous and clocked synchronous serial communication modes
- Full Duplex Communication Capability
- Programming baud rate clock frequency up to (f_{PCLK}/16) MHz for asynchronous mode and (f_{PCLK}/8) MHz for synchronous mode
- IrDA SIR encoder and decoder
  - Support of normal 3/16 bit duration and low-power (1.41 ~ 2.23 μs) durations
- Supports RS485 mode with output enable
- Auto hardware flow control mode – RTS, CTS
- Fully programmable serial communication functions including:
  - Word length: 7, 8 or 9-bit character
  - Parity: Even, odd or no-parity bit generation and detection
  - Stop bit: 1 or 2 stop bits generation
  - Bit order: LSB-first or MSB-first transfer
- Error detection: Parity, overrun and frame error
- FIFO:
  - Receive FIFO: 8 × 9 bits (max. 9 data bits)
  - Transmit FIFO: 8 × 9 bits (max. 9 data bits)
- Supports PDMA Interface

Function Descriptions

Serial Data Format

The USART module performs a parallel-to-serial conversion on data that is written to the transmit FIFO registers and then sends the data with the following format: Start bit, 7 ~ 9 LSB / MSB first data bits, optional Parity bit and finally 1 ~ 2 Stop bits. The Start bit has the opposite polarity of the data line idle state. The Stop bit is the same as the data line idle state and provides a delay before the next start situation. Both the Start and Stop bits are used for data synchronization during the asynchronous data transmission.

The USART module also performs a serial-to-parallel conversion on the data that is read from the receive FIFO registers. It will first check the Parity bit and will then look for a Stop bit. If the Stop bit is not found, the USART module will consider the entire word transmission to have failed and respond with a Framing Error.

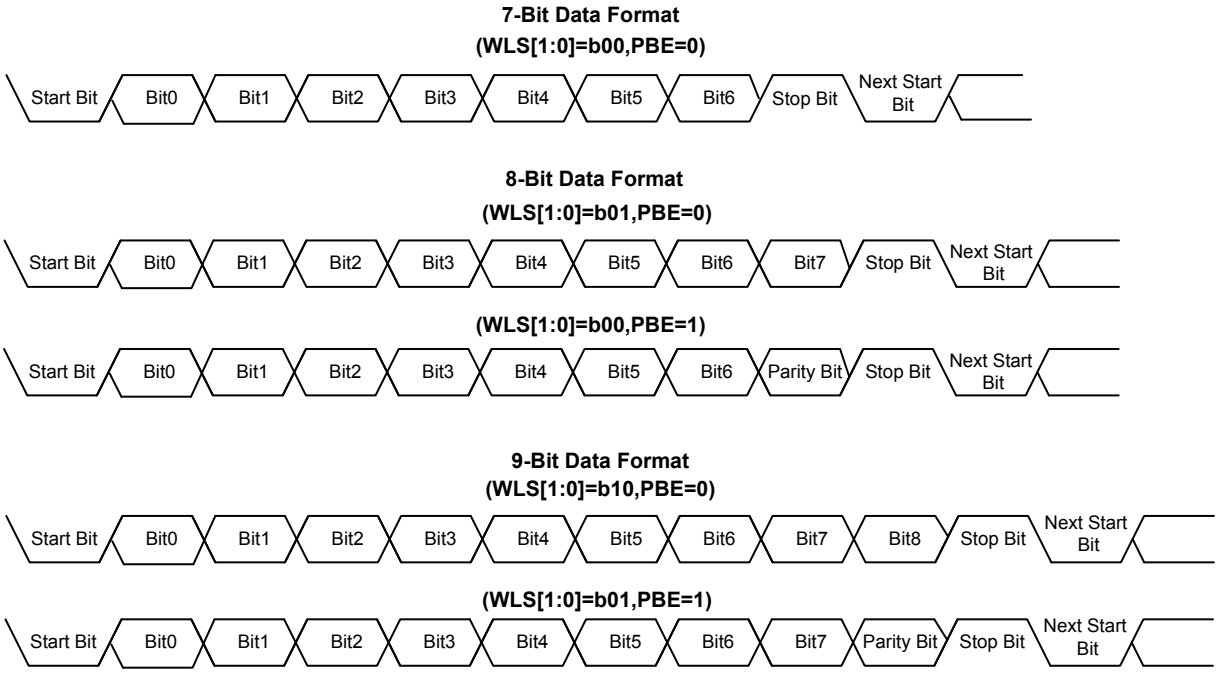


Figure 135. USART Serial Data Format

Baud Rate Generation

The baud rate for the USART receiver and transmitter are both set with the same values. The baud rate divisor, BRD, has the following relationship with the USART clock which is known as CK_USART.

$$\text{Baud Rate Clock} = \text{CK\_USART} / \text{BRD}$$

Where CK_USART clock is the APB clock connected to the USART while the BRD range is from 16 to 65535 for asynchronous mode and 8 to 65535 for synchronous mode.

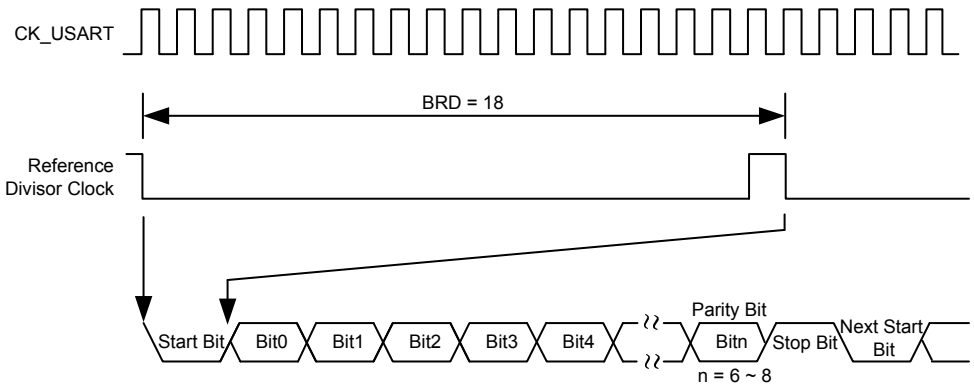


Figure 136. USART Clock CK_USART and Data Frame Timing

Table 54. Baud Rate Deviation Error Calculation – CK_USART = 40 MHz

Baud rate		CK_USART = 40 MHz		
No.	Kbps	Actual	BRD	Deviation Error Rate
1	2.4	2.4	16667	0.00%
2	9.6	9.6	4167	-0.01%
3	19.2	19.2	2083	0.02%
4	57.6	57.6	694	0.06%
5	115.2	115.3	347	0.06%
6	230.4	229.9	174	-0.22%
7	460.8	459.8	87	-0.22%
8	921.6	930.2	43	0.94%
9	2250	2222.2	18	-1.23%
10	3000	3076.9	13	2.56%

Table 55. Baud Rate Deviation Error Calculation – CK_USART = 48 MHz

Baud rate		CK_USART = 48 MHz		
No	Kbps	Actual	BRD	Deviation Error Rate
1	2.4	2.4	20000	0.00%
2	9.6	9.6	5000	0.00%
3	19.2	19.2	2500	0.00%
4	57.6	57.6	833	0.04%
5	115.2	115.1	417	-0.08%
6	230.4	230.8	208	0.16%
7	460.8	461.5	104	0.16%
8	921.6	923.1	52	0.16%
9	2250	2285.7	21	1.59%
10	3000	3000	16	0.00%

Hardware Flow Control

The USART supports the hardware flow control function which is enabled by setting the HFCEN bit in the USRCR register to 1. It is possible to control the serial data flow between 2 USART devices by using the CTS input and the RTS output. The Figure 137 shows the connection diagram in this mode. The hardware flow control function is categorized into two types. One is the RTS flow control function and the other is the CTS flow control function.

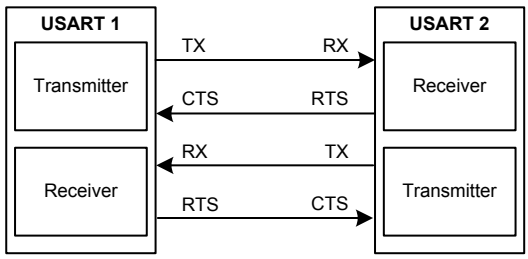


Figure 137. Hardware Flow Control between 2 USARTs

RTS Flow Control

In the RTS flow control, the USART RTS pin is active with a logic low state when the receive data register is empty. It means that the receiver is ready to receive a new data. When the RX FIFO reaches the trigger level which is specified by configuring the RXTL field in the USRFCR register, the USART RTS pin is inactive with a logic high state. Figure 138 shows the example of RTS flow control.

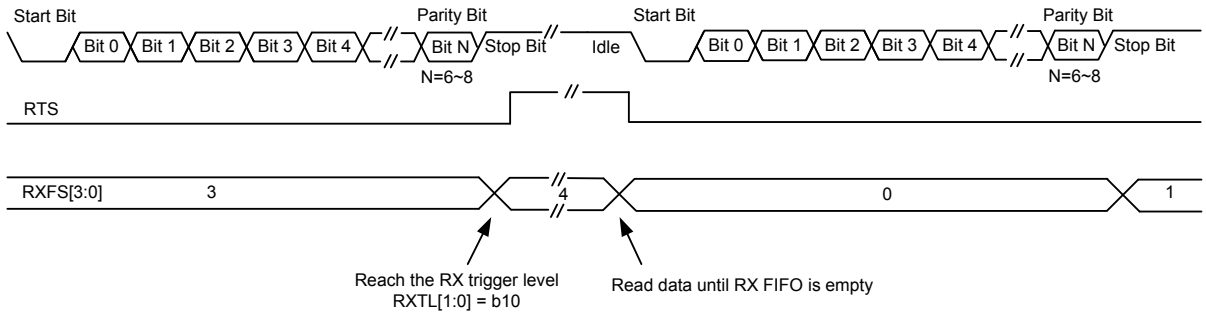


Figure 138. USART RTS Flow Control

CTS Flow Control

If the hardware flow control function is enabled, the URTXEN bit in the USRCR register will be controlled by the USART CTS input signal. If the USART CTS pin is forced to a logic low state, the URTXEN bit will automatically be set to 1 to enable the data transmission. However, if the USART CTS pin is forced to a logic high state, the URTXEN bit will be cleared to 0 and then the data transmission will also be disabled.

When the USART CTS pin is forced to a logic high state during a data transmission period, the current data transmission will be continued until the stop bit is completed. The Figure 139 shows an example of communication with CTS flow control..

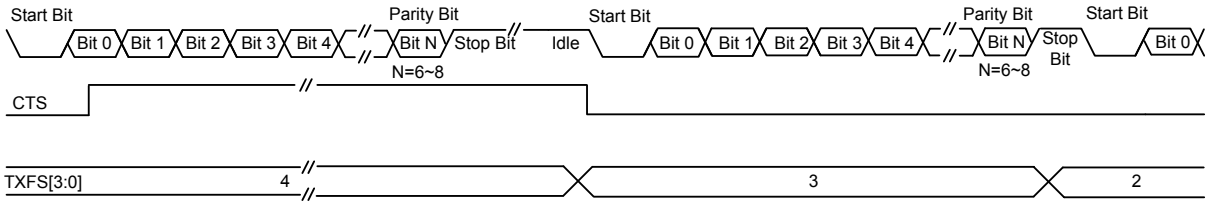


Figure 139. USART CTS Flow Control

IrDA

The USART IrDA mode is provided for half-duplex point-to-point wireless communication.

The USART module includes an integrated modulator and demodulator which allow a wireless communication using infrared transceivers. The transmitter specifies a logic data '0' as a 'high' pulse and a logic data '1' as a 'low' level while the receiver specifies a logic data '0' as a 'low' pulse and a logic data '1' as 'high' level in the IrDA mode.

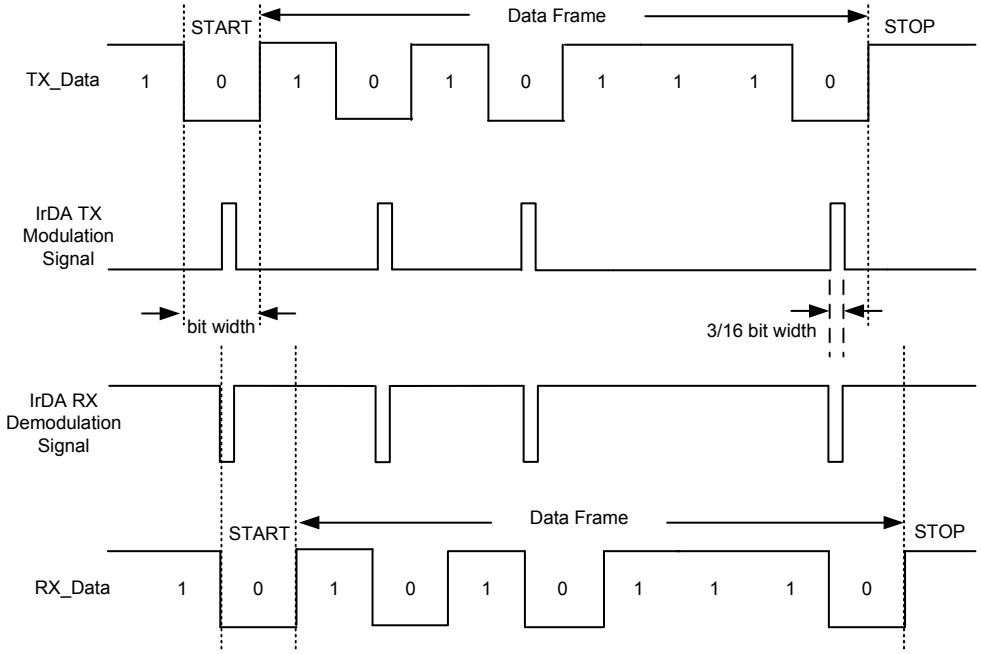


Figure 140. IrDA Modulation and Demodulation

The IrDA mode provides two operation modes, one is the normal mode and the other is the low-power mode.

IrDA Normal Mode

For the IrDA normal mode, the width of each transmitted pulse generated by the transmitter modulator is specified as 3/16 of the baud rate clock period. The receiver pulse width for the IrDA receiver demodulator is based on the IrDA receive debounce filter which is implement using an 8-bit down-counting counter. The debounce filter counter value is specified by the IrDAPSC field in the IrDACR register. When a falling edge is detected on the receiver pin, the debounce filter counter starts to count down, driven by the CK_USART clock. If a rising edge is detected on the receiver pin, the counter stops counting and is reloaded with the IrDAPSC value. When a low pulse falling edge on the receiver pin is detected and then before the debounce filter has counted down to zero, a rising edge is also detected, then this low pulse will be considered as glitch noise and will be discarded. If a low pulse falling edge appears on the receiver pin but no rising edge is detected before the debounce counter reaches 0, then the input is regarded as a valid data "0" for this bit duration. The IrDAPSC value must be set to be greater than or equal to 0x01, then the IrDA receiver demodulation operation can function properly. The IrDAPSC value can be adjusted to meet the USART baud rate setting to filter the IrDA received glitch noise of which the width is smaller than the prescaler setting duration.

IrDA Low-Power Mode

In the IrDA low-power mode, the transmitted IrDA pulse width generated by the transmitter modulator is not kept at 3/16 of the baud rate clock period. Instead, the pulse width is fixed and is calculated by the following formula. The transmitted pulse width can be adjusted by the IrDAPSC field to meet the minimum pulse width specification of the external IrDA Receiver device.

$$T_{IrDA\_L} = 3 \times IrDAPSC / CK\_USART$$

Note: T_{IrDA_L} is the transmitted pulse width in the low-power mode.
The IrDAPSC filed is the IrDA prescaler value in the IrDA Control Register IrDACR.

The debounce behavior in the IrDA low-power receiving mode is similar to the IrDA normal mode. For glitch detection, the low pulse of which the pulse width is shorter than 1 × (IrDAPSC / CK_USART) should be discarded in the IrDA receiver demodulation. A valid low data is accepted if its low pulse width is greater than 2 × (IrDAPSC / CK_USART) duration.

The IrDA physical layer specification specifies a minimum delay with a value of 10 ms between the transmission and reception switch; and this IrDA receiver set-up time also should be managed by the software.

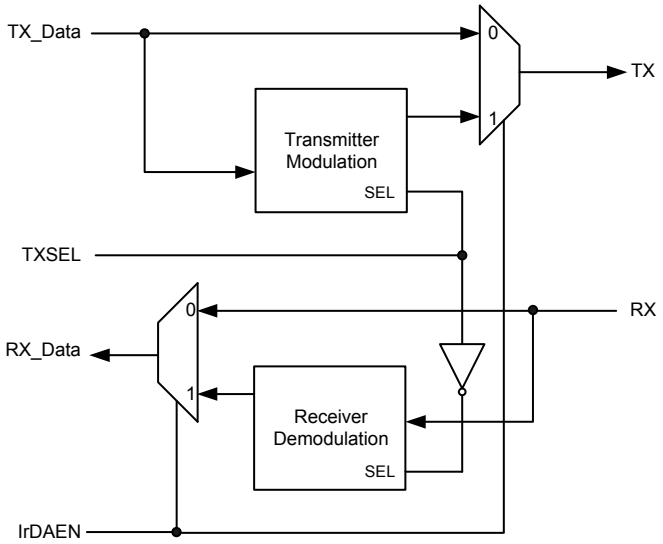


Figure 141. USART I/O and IrDA Block Diagram

RS485 Mode

The RS485 mode of the USART provides the data transmission on the interface transmitted over a 2-wire twisted pair bus. The RS485 transceiver interprets the voltage levels of the differential signals with respect to a third common voltage. Without this common reference, the transceiver may interpret the differential signals incorrectly. This enhances the noise rejection capabilities of the RS485 interface. The USART RTS pin is used to control the external RS485 transceiver whose polarity can be selected by configuring the TXENP bit in the RS485 Control Register, named RS485CR, when the USART operates in the RS485 mode.

RS485 Auto Direction Mode – AUD

When the RS485 mode is configured as a master transmitter, it will operate in the Auto Direction Mode, AUD. In the AUD mode the polarity of the USART RTS pin is configurable according to the TXENP bit in the RS485 Control Register in the RS485 mode. This pin can be used to control the external RS485 transceiver to enable the transmitter.

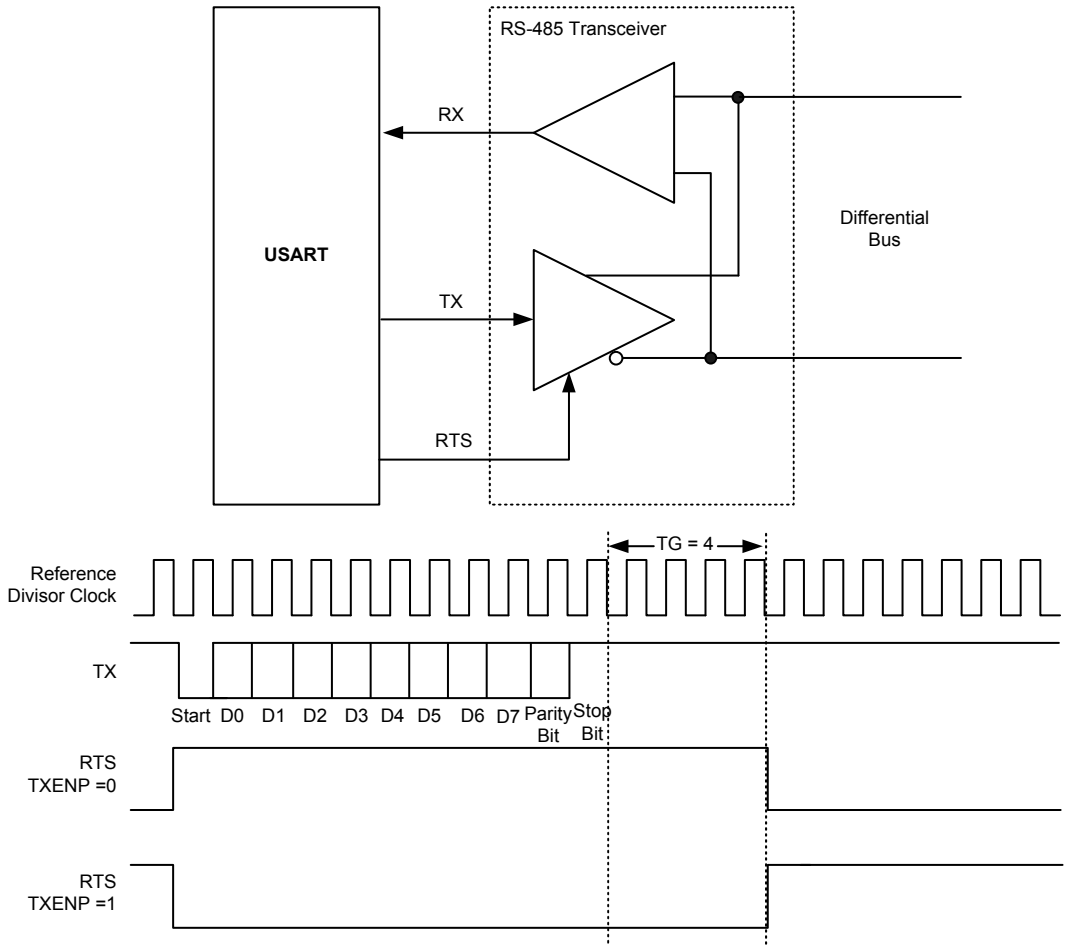


Figure 142. RS485 Interface and Waveform

RS485 Normal Multi-drop Operation Mode – NMM

When the RS485 mode is configured as an addressable slave, it will operate in the Normal Multi-drop Operation Mode, NMM. This mode is enabled when the RSNMM field is set in the RS485CR register. Regardless of the URRXEN value in the USRCR register, all the received data with a parity bit "0" will be ignored until the first address byte is detected with a parity bit "1" and then the received address byte will be stored in the RX FIFO. Once the first address data is detected and stored in the RX FIFO, the RSADD flag in the USRSIFR register will be set and generate an interrupt if the RSADDIE bit in the USRIER register is set to 1. Application software can determine whether the receiver is enabled or disabled to accept the following data by configuring the URRXEN bit. When the receiver is enabled by setting the URRXEN bit to 1, all received data will be stored in the RX FIFO. Otherwise, all received data will be ignored if the receiver is disabled by clearing the URRXEN bit to 0.

RS485 Auto Address Detection Operation Mode – AAD

Except in the Normal Multi-drop Operation Mode, the RS485 mode can operate in the Auto Address Detection Operation Mode, AAD, when it is configured as an addressable slave. This mode is enabled by setting the RSAAD filed to 1 in the RS485CR register. The receiver will detect the address frame with a parity bit "1" and then compare the received address data with the ADDMATCH field value which is a programmable 8-bit address value specified in the RS485CR register. If the address data matches the ADDMATCH value, it will be stored in the RX FIFO and the URRXEN bit will be automatically set. When the receiver is enabled, all received data will be stored in the RX FIFO until the next address frame does not match the ADDMATCH value and then the receiver will be automatically disabled. After the receiver is enabled, software can disable the receiver by setting the URRXEN bit to '0'.

Synchronous Master Mode

The data is transmitted in a full-duplex style in the USART Synchronous Master Mode, i.e., data transmission and reception both occur at the same time and only support master mode. The USART CTS pin is the synchronous USART transmitter clock output. In this mode, no clock pulses will be sent to the CTS pin during the start bit, parity bit and stop bit duration. The CPS bit in the Synchronous Control Register SYNCR, can be used to determine whether data is captured on the first or the second clock edge. The CPO bit in the SYNCR can be used to configure the clock polarity in the USART Synchronous Mode idle state. Detailed timing information is shown in Figure 144.

In the USART synchronous Mode, the USART CTS / SCK clock output pin is only used to transmit the data to slave device. If the transmission data register USRDR, is written with valid data, the USART synchronous mode will automatically transmit this data with the corresponding clock output and the USART receiver will also receive data on the RX pin. Otherwise the receiver will not obtain synchronous data if no data is transmitted.

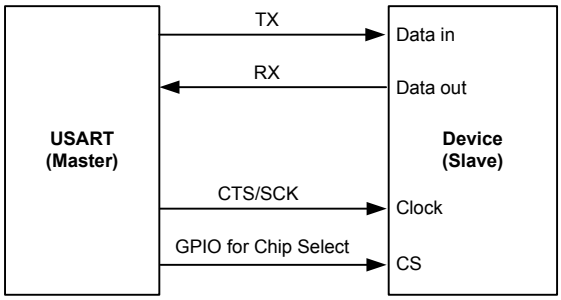


Figure 143. USART Synchronous Transmission Example

Note: The USART supports the synchronous master mode only: it cannot receive or send data related to an input clock. The USART CTS / SCK clock is always an output.

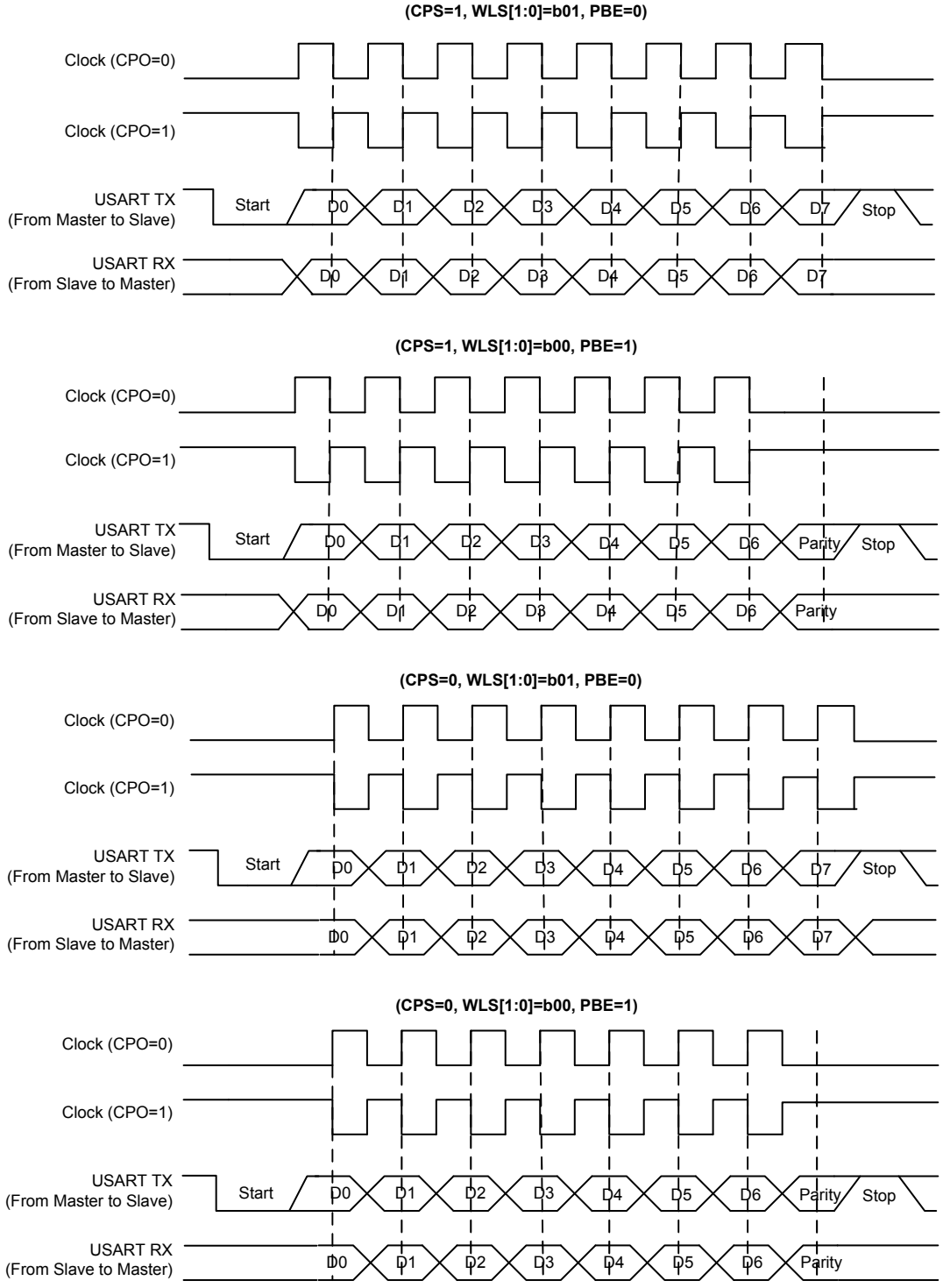


Figure 144. 8-bit Format USART Synchronous Waveform

Interrupts and Status

The USART can generate interrupts when the following event occurs and corresponding interrupt enable bits are set:

- Receive FIFO time-out interrupt: An interrupt will be generated when the USART receive FIFO is not empty and does not receive a new data package during the specified time-out interval.
- Receiver line status interrupts: The interrupts will be generated when the USART receiver overrun error, parity error, framing error and break events occur.
- Transmit FIFO threshold level interrupt: An interrupt will be generated when the data to be transmitted in the USART Transmit FIFO is less than the specified threshold level.
- Transmit complete interrupt: An interrupt will be generated when the Transmit FIFO is empty and the content of the transmit shift register (TSR) is also completely shifted.
- Receive FIFO threshold level interrupt: An interrupt will be generated when the FIFO received data amount has reached the specified threshold level.

PDMA Interface

The PDMA interface is integrated in the USART. The PDMA function can be enabled by setting the TXDMAEN or RXDMAEN bit in the USRCR register to 1 in the transmit or receive mode respectively. When the data to be transmitted in the USART Transmit FIFO is less than the TX FIFO threshold level specified by the TXTL field in the USRFCR register and the TXDMAEN bit is set to 1, the PDMA function will be activated to move data from a source location into the USART TX FIFO.

Similarly, when the received data amount in the receive FIFO is equal to the RX FIFO threshold level specified by the RXTL field in the USRFCR register and the RXDMAEN bit is set to 1, the PDMA function will be activated to move data from the USART RX FIFO to a specific destination location. For a mode detailed description on the PDMA configurations, refer to the PDMA chapter.

Register Map

The following table shows the USART registers and reset values.

Table 56. USART Register Map

Register	Offset	Description	Reset Value
USRDR	0x000	USART Data Register	0x0000_0000
USRCR	0x004	USART Control Register	0x0000_0000
USRFCR	0x008	USART FIFO Control Register	0x0000_0000
USRIER	0x00C	USART Interrupt Enable Register	0x0000_0000
USRSIFR	0x010	USART Status & Interrupt Flag Register	0x0000_0180
USRTPR	0x014	USART Timing Parameter Register	0x0000_0000
IrDACR	0x018	USART IrDA Control Register	0x0000_0000
RS485CR	0x01C	USART RS485 Control Register	0x0000_0000
SYNCR	0x020	USART Synchronous Control Register	0x0000_0000
USRDLR	0x024	USART Divider Latch Register	0x0000_0010
USRTSTR	0x028	USART Test Register	0x0000_0000

Register Descriptions

USART Data Register – USRDR

The register is used to access the USART transmitted and received FIFO data.

Offset: 0x000

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	22	21	20	19	18	17	16
	Reserved							
Type/Reset								
	15	14	13	12	11	10	9	8
	Reserved							DB
Type/Reset								RW 0
	7	6	5	4	3	2	1	0
	DB							
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0

Bits	Field	Descriptions
[8:0]	DB	Reading data via this receiver buffer register will return the data from the receive FIFO. The receive FIFO has a capacity of up to 8 × 9 bits. By reading this register, the USART will return a 7, 8 and 9-bit received data. The DB field bit 8 is valid for 9-bit mode only and is fixed at 0 for the 8-bit mode. For the 7-bits mode, the DB[6:0] field contains the available bits. Writing data to this buffer register will load data into the Transmit FIFO. The Transmit FIFO has a capacity of up to 8 × 9 bits. By writing to this register, the USART will send out 7, 8 or 9-bit transmitted data. The DB field bit 8 is valid for the 9-bit mode only and will be ignored for the 8-bit mode. For the 7-bit mode, the DB[6:0] field contains the available bits.

USART Control Register – USRCR

The register specifies the serial parameters such as data length, parity and stop bit for the USART. It also contains the USART enable control bits together with the USART mode and data transfer mode selections.

Offset: 0x004

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	22	21	20	19	18	17	16
	Reserved							
Type/Reset								
	15	14	13	12	11	10	9	8
	RTS	BCB	SPE	EPE	PBE	NSB		WLS
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0
	7	6	5	4	3	2	1	0
	RXDMAEN	TXDMAEN	URRXEN	URTXEN	HFCEN	TRSM		MODE
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0

Bits	Field	Descriptions
[15]	RTS	Request-To-Send Signal 0: Drive USART RTS pin to logic 1 1: Drive USART RTS pin to logic 0 Note that the RTS bit is used to control the USART RTS pin status when the HFCEN bit is reset. When the HFCEN bit is set, this RTS bit is read only, which indicates the pin status that is controlled by hardware flow control function.
[14]	BCB	Break Control Bit When this bit is set 1, the serial data output on the USART TX pin will be forced to the Spacing State (logic 0). This bit acts only on USART TX output pin and has no effect on the transmitter logic.
[13]	SPE	Stick Parity Enable 0: Disable stick parity 1: Stick Parity bit is transmitted This bit is only available when the PBE bit is set to 1. If both the PBE and SPE bits are set to 1 and the EPE bit is cleared to 0, the transmitted parity bit will be stuck to 1. However, when the PBE and SPE bits are set to 1 and also the EPE bit is set to 1, the transmitted parity bit will be stuck to 0.
[12]	EPE	Even Parity Enable 0: Odd number of logic 1's are transmitted or checked in the data word and parity bits 1: Even number of logic 1's are transmitted or checked in the data word and parity bits This bit is only available when PBE is set to 1.

Bits	Field	Descriptions
[11]	PBE	Parity Bit Enable 0: Parity bit is not generated (transmitted data) or checked (received data) during transfer 1: Parity bit is generated or checked during transfer Note: When the WLS field is set to “10” to select the 9-bit data format, writing to the PBE bit has no effect.
[10]	NSB	Number of “STOP bit” 0: One “ STOP bit” is generated in the transmitted data 1: Two “STOP bit” is generated when 8-bit or 9-bit word length is selecte
[9:8]	WLS	Word Length Select 00: 7 bits 01: 8 bits 10: 9 bits 11: Reserved
[7]	RXDMAEN	USART RX DMA Enable 0: Disable 1: Enable
[6]	TXDMAEN	USART TX DMA Enable 0: Disable 1: Enable
[5]	URRXEN	USART RX Enable 0: Disable 1: Enable
[4]	URTXEN	USART TX Enable 0: Disable 1: Enable
[3]	HFCEN	Hardware Flow Control Function Enable 0: Disable 1: Enable
[2]	TRSM	Transfer Mode Selection This bit is used to select the data transfer protocol. 0: LSB first 1: MSB first
[1:0]	MODE	USART Mode Selection 00: Normal operation 01: IrDA 10: RS485 11: Synchronous

USART FIFO Control Register – USRFCR

This register specifies the USART FIFO control and configurations including threshold level and reset function together with the USART FIFO status.

Offset: 0x008

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved				RXFS			
Type/Reset					RO 0	RO 0	RO 0	RO 0
	23	22	21	20	19	18	17	16
	Reserved				TXFS			
Type/Reset					RO 0	RO 0	RO 0	RO 0
	15	14	13	12	11	10	9	8
	Reserved							
Type/Reset								
	7	6	5	4	3	2	1	0
	RXTL		TXTL		Reserved		RXR	TXR
Type/Reset	RW 0	RW 0	RW 0	RW 0			WO 0	WO 0

Bits	Field	Descriptions
[27:24]	RXFS	RX FIFO Status The RXFS field shows the current number of data contained in the RX FIFO. 0000: RX FIFO is empty 0001: RX FIFO contains 1 data ... 1000: RX FIFO contains 8 data Others: Reserved
[19:16]	TXFS	TX FIFO Status The TXFS field shows the current number of data contained in the TX FIFO. 0000: TX FIFO is empty 0001: TX FIFO contains 1 data ... 1000: TX FIFO contains 8 data Others: Reserved
[7:6]	RXTL	RX FIFO Threshold Level Setting 00: 1 byte 01: 2 bytes 10: 4 bytes 11: 6 bytes The RXTL field defines the RX FIFO trigger level.
[5:4]	TXTL	TX FIFO Threshold Level Setting 00: 0 byte 01: 2 bytes 10: 4 bytes 11: 6 bytes The TXTL field determines the TX FIFO trigger level.
[1]	RXR	RX FIFO Reset Setting this bit will generate a reset pulse to reset the RX FIFO which will empty the RX FIFO, i.e., the RX pointer will be reset to 0 after a reset signal. This bit returns to 0 automatically after the reset pulse is generated.

Bits	Field	Descriptions
[0]	TXR	TX FIFO Reset Setting this bit will generate a reset pulse to reset the TX FIFO which will empty the TX FIFO, i.e., the TX pointer will be reset to 0 after a reset signal. This bit returns to 0 automatically after the reset pulse is generated.

USART Interrupt Enable Register – USRIER

This register is used to enable the related USART interrupt function. The USART module generates interrupts to the controller when the corresponding events occur and the corresponding interrupt enable bits are set.

Offset: 0x00C

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	22	21	20	19	18	17	16
	Reserved							
Type/Reset								
	15	14	13	12	11	10	9	8
	Reserved						CTSIE	RXTOIE
Type/Reset							RW 0	RW 0
	7	6	5	4	3	2	1	0
	RSADDIE	BIE	FEIE	PEIE	OEIE	TXCIE	TXDEIE	RXDRIE
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0

Bits	Field	Descriptions
[9]	CTSIE	CTS Clear-To-Send Interrupt Enable 0: Disable interrupt 1: Enable interrupt If this bit is set, an interrupt will be generated when the CTSC bit is set in the USRSIFR register.
[8]	RXTOIE	Receive FIFO Time-Out Interrupt Enable 0: Disable interrupt 1: Enable interrupt If this bit is set, an interrupt will be generated when the RXTOF bit is set in the USRSIFR register.
[7]	RSADDIE	RS485 Address Detection Interrupt Enable 0: Disable interrupt 1: Enable interrupt If this bit is set, an interrupt will be generated when the RSADD bit is set in the USRSIFR register.
[6]	BIE	Break Interrupt Enable 0: Disable interrupt 1: Enable interrupt If this bit is set, an interrupt will be generated when the BII bit is set in the USRSIFR register.

Bits	Field	Descriptions
[5]	FEIE	Framing Error Interrupt Enable 0: Disable interrupt 1: Enable interrupt If this bit is set, an interrupt will be generated when the FEI bit is set in the URSIFR register.
[4]	PEIE	Parity Error Interrupt Enable 0: Disable interrupt 1: Enable interrupt If this bit is set, an interrupt will be generated when the PEI bit is set in the USRSIFR register.
[3]	OEIE	Overrun Error Interrupt Enable 0: Disable interrupt 1: Enable interrupt If this bit is set, an interrupt will be generated when the OEI bit is set in the USRSIFR register.
[2]	TXCIE	Transmit Complete Interrupt Enable 0: Disable interrupt 1: Enable interrupt If this bit is set, an interrupt will be generated when the TXC bit is set in the USRSIFR register.
[1]	TXDEIE	Transmit Data Empty Interrupt Enable 0: Disable interrupt 1: Enable interrupt If this bit is set, an interrupt will be generated when the TXDE bit is set in the USRSIFR register.
[0]	RXDRIE	Receive Data Ready Interrupt Enable 0: Disable interrupt 1: Enable interrupt If this bit is set, an interrupt will be generated when the RXDR bit is set in the USRSIFR register.

USART Status & Interrupt Flag Register – USRSIFR

This register contains the corresponding USART status.

Offset: 0x010

Reset value: 0x0000_0180

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	22	21	20	19	18	17	16
	Reserved							
Type/Reset								
	15	14	13	12	11	10	9	8
	Reserved				CTSS	CTSC	RSADD	TXC
Type/Reset					RO 0	WC 0	WC 0	RO 1
	7	6	5	4	3	2	1	0
	TXDE	RXTOF	RXDR	BII	FEI	PEI	OEI	RXDNE
Type/Reset	RO 1	WC 0	RO 0	WC 0	WC 0	WC 0	WC 0	RO 0

Bits	Field	Descriptions
[11]	CTSS	CTS Clear-To-Send Status 0: CTS pin is inactive 1: CTS pin is active and kept at a logic low state
[10]	CTSC	CTS Status Change Flag This bit will be set whenever the CTS input pin status is changed and an Interrupt will be generated if the CTSIE = 1 in the USRIER register. Writing 1 to this bit clears the flag.
[9]	RSADD	RS485 Address Detection 0: Address is not detected 1: Address is detected This bit will be set to 1 when the receiver detects the address. An interrupt will be generated if RSADDIE = 1 in the USRIER register. Writing 1 to this bit clears the flag. Note: This bit is only used in the RS485 mode by setting the MODE field in the USRCR register.
[8]	TXC	Transmit Complete 0: Either transmit FIFO (TX FIFO) or transmit shift register (TSR) is not empty 1: Both the TX FIFO and TSR register are empty An interrupt will be generated if TXCIE = 1 in the USRIER register. This bit is cleared by a write to the USRDR register with new data.
[7]	TXDE	Transmit Data FIFO Empty 0: TX FIFO level is higher than threshold 1: TX FIFO level is less than threshold The TXDE bit will be set when transmit FIFO level is less than the transmit FIFO threshold level setting which is set by the TXTL field in the USRFCR register. This bit will be cleared when the new data is written into the USRDR register and the TX FIFO level is higher than threshold setting.

Bits	Field	Descriptions
[6]	RXTOF	Receive FIFO Time-Out Flag 0: RX FIFO Time-Out does not occur 1: RX FIFO Time-Out occurs The RXTOF bit will be set if the receive FIFO is not empty and no activities have occurred in the receive FIFO during the time-out duration specified by the RXTOC field. Writing 1 to this bit clears the flag.
[5]	RXDR	Receive FIFO Ready Flag 0: RX FIFO level is less than threshold 1: RX FIFO level is higher than threshold The RXDR bit will be set when the FIFO received data amount reaches the specified threshold level which is set by the RXTL field in the USRFCR register. This bit will be cleared when the data is read from the USRDR register and the RX FIFO level is less than threshold setting.
[4]	BII	Break Interrupt Indicator This bit will be set to 1 whenever the received data input is held in the “spacing state” (logic 0) for longer than a full word transmission time, which is the total time of “start bit” + data bits + “parity” + “stop bits” duration. Writing 1 to this bit clears the flag.
[3]	FEI	Framing Error Indicator This bit will be set to 1 whenever the received character does not have a valid “stop bit”, which means, the stop bit following the last data bit or parity bit is detected as logic 0. Writing 1 to this bit clears the flag.
[2]	PEI	Parity Error Indicator This bit will be set to 1 whenever the received character does not have a valid “parity bit”. Writing 1 to this bit clears the flag.
[1]	OEI	Overrun Error Indicator An overrun error will occur only after the RX FIFO is full and when the next character has been completely received in the RX shift register. The character in the shift register will be overwritten if a new character is received in the RX shift register after an overrun event occurs, but the data in the RX FIFO will not be overwritten. The OEI bit is used to indicate the overrun event as soon as it happens. Writing 1 to this bit clears the flag.
[0]	RXDNE	RX FIFO Data Not Empty 0: RX FIFO is empty 1: RX FIFO contains at least 1 received data word

USART Timing Parameter Register – USRTPR

This register contains the USART timing parameters including the transmitter time guard parameters and the receive FIFO time-out value together with the RX FIFO time-out function enable control.

Offset: 0x014

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	22	21	20	19	18	17	16
	Reserved							
Type/Reset								
	15	14	13	12	11	10	9	8
	TG							
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0
	7	6	5	4	3	2	1	0
	RXTOEN	RXTOC						
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0

Bits	Field	Descriptions
[15:8]	TG	Transmitter Time Guard The transmitter time guard counter is driven by the baud rate clock. When the TX FIFO transmits data, the counter will be reset and then starts to count after a word transmission has completed. Only when the counter content is equal to the TG value, are further word transmission transactions allowed.
[7]	RXTOEN	Receive FIFO Time-Out Counter Enable 0: Receive FIFO Time-Out Counter is disabled 1: Receive FIFO Time-Out Counter is enabled
[6:0]	RXTOC	Receive FIFO Time-Out Counter Compare Value The RX FIFO time-out counter is driven by the baud rate clock. When the RX FIFO receives new data, the counter will be reset and then starts to count. Once the time-out counter content is equal to the time-out counter compare value RXTOC, a receive FIFO time-out interrupt, RXTOI, will be generated if the RXTOIE bit in the USRIER register is set to 1. New received data or the empty RX FIFO after being read will clear the RX FIFO time-out counter.

USART IrDA Control Register – IrDACR

This register is used to control the IrDA mode of USART.

Offset: 0x018

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	22	21	20	19	18	17	16
	Reserved							
Type/Reset								
	15	14	13	12	11	10	9	8
	IrDAPSC							
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0
	7	6	5	4	3	2	1	0
		Reserved	RXINV	TXINV	LB	TXSEL	IrDALP	IrDAEN
Type/Reset			RW 0	RW 0	RW 0	RW 0	RW 0	RW 0

Bits	Field	Descriptions
[15:8]	IrDAPSC	IrDA Prescaler value This field contains the 8-bit debounce prescaler value. The debounce count-down counter is driven by the USART clock, named as CK_USART. The counting period is specified by the IrDAPSC field. The IrDAPSC field must be set to a value equal to or greater than 0x01 for normal debounce counter operation. If the pulse width is less than the duration specified by the IrDAPSC field, the pulse will be considered as glitch noise and discarded. 00000000: Reserved – can not be used. 00000001: CK_USART clock divided by 1 00000010: CK_USART clock divided by 2 00000011: CK_USART clock divided by 3 ...
[5]	RXINV	RX Signal Inverse Control 0: No inversion 1: RX input signal is inversed
[4]	TXINV	TX Signal Inverse Control 0: No inversion 1: TX output signal is inversed
[3]	LB	IrDA Loop Back Mode 0: Disable IrDA loop back mode 1: Enable IrDA loop back mode for self testing.
[2]	TXSEL	Transmit Select 0: Enable IrDA receiver 1: Enable IrDA transmitter
[1]	IrDALP	IrDA Low Power Mode Select the IrDA operation mode. 0: Normal mode 1: IrDA low power mode

Bits	Field	Descriptions
[0]	IrDAEN	IrDA Enable control 0: Disable IrDA mode 1: Enable IrDA mode

USART RS485 Control Register – RS485CR

This register is used to control the RS485 mode of USART.

Offset: 0x01C

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	22	21	20	19	18	17	16
	Reserved							
Type/Reset								
	15	14	13	12	11	10	9	8
	ADDMATCH							
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0
	7	6	5	4	3	2	1	0
	Reserved					RSAAD	RSNMM	TXENP
Type/Reset						RW 0	RW 0	RW 0

Bits	Field	Descriptions
[15:8]	ADDMATCH	RS485 Auto Address Match value The field contains the address match value for the RS485 auto address detection operation mode.
[2]	RSAAD	RS485 Auto Address Detection Operation Mode Control 0: Disable 1: Enable
[1]	RSNMM	RS485 Normal Multi-drop Operation Mode Control 0: Disable 1: Enable
[0]	TXENP	USART RTS / TXE Pin Polarity 0: RTS / TXE is active high in the RS485 transmission mode. 1: RTS / TXE is active low in the RS485 transmission mode.

USART Synchronous Control Register – SYNCR

This register is used to control the USART synchronous mode.

Offset: 0x020

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	22	21	20	19	18	17	16
	Reserved							
Type/Reset								
	15	14	13	12	11	10	9	8
	Reserved							
Type/Reset								
	7	6	5	4	3	2	1	0
	Reserved				CPO	CPS	Reserved	CLKEN
Type/Reset					RW 0	RW 0		RW 0

Bits	Field	Descriptions
[3]	CPO	Clock Polarity 0: CTS / SCK pin idle state is low. 1: CTS / SCK pin idle state is high. Selects the polarity of the clock output on the USART CTS / SCK pin in the synchronous mode. Works in conjunction with the CPS bit to specify the desired clock idle state.
[2]	CPS	Clock Phase 0: Data is captured on the first clock edge. 1: Data is captured on the second clock edge. This bit allows the user to select the phase of the clock output on the USART CTS / SCK pin in the synchronous mode. Works in conjunction with the CPO bit to determine the data capture edge.
[0]	CLKEN	Clock Enable 0: CTS / SCK pin disabled 1: CTS / SCK pin enabled Enable / disable the USART CTS / SCK pin.

USART Divider Latch Register – USRDLR

The register is used to determine the USART clock divided ratio to generate the appropriate baud rate.

Offset: 0x024

Reset value: 0x0000_0010

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	22	21	20	19	18	17	16
	Reserved							
Type/Reset								
	15	14	13	12	11	10	9	8
	BRD							
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0
	7	6	5	4	3	2	1	0
	BRD							
Type/Reset	RW 0	RW 0	RW 0	RW 1	RW 0	RW 0	RW 0	RW 0

Bits	Field	Descriptions
[15:0]	BRD	Baud Rate Divider The 16 bits define the USART clock divider ratio. Baud Rate = CK_USART / BRD Where the CK_USART clock is the clock connected to the USART module. BRD = 16 ~ 65535 for asynchronous mode BRD = 8 ~ 65535 for synchronous mode.

USART Test Register – USRTSTR

This register controls the USART debug mode.

Offset: 0x028

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	22	21	20	19	18	17	16
	Reserved							
Type/Reset								
	15	14	13	12	11	10	9	8
	Reserved							
Type/Reset								
	7	6	5	4	3	2	1	0
	Reserved						LBM	
Type/Reset							RW 0	RW 0

Bits	Field	Descriptions
[1:0]	LBM	Loopback Test Mode Select 00: Normal Operation 01: Reserved 10: Automatic Echo Mode 11: Loopback Mode

22 Universal Asynchronous Receiver Transmitter (UART)

Introduction

The Universal Asynchronous Receiver Transceiver, UART, provides a flexible full duplex data exchange using asynchronous transfer. The UART is used to translate data between parallel and serial interfaces, and is also commonly used for RS232 standard communication. The UART peripheral function supports a variety of interrupts.

The UART module includes a transmit data register TDR and transmit shift register TSR, and a receive data register RDR and receive shift register RSR. Software can detect a UART error status by reading the UART Status & Interrupt Flag Register, URSIFR. The status includes the condition of the transfer operations as well as several error conditions resulting from Parity, Overrun, Framing and Break events.

The UART includes a programmable baud rate generator which is capable of dividing the UART clock of the CK_APB (CK_UART) to produce a baud rate clock for the UART transmitter and receiver.

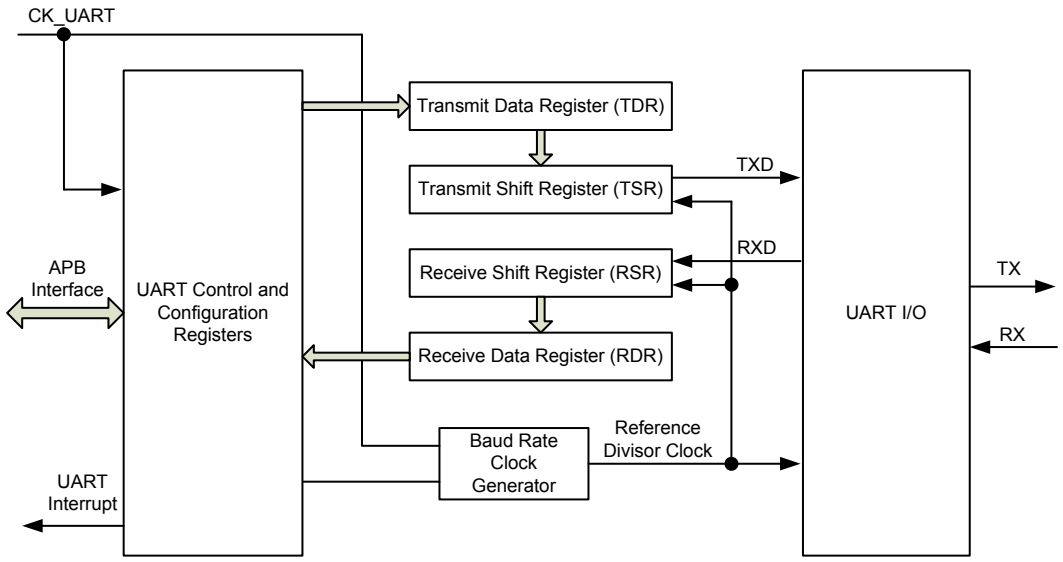


Figure 145. UART Block Diagram

Features

- Supports asynchronous serial communication modes
- Full Duplex Communication Capability
- Programming baud rate clock frequency up to (f_{PCLK}/16) MHz
- Fully programmable serial communication functions including:
  - Word length: 7, 8 or 9-bit character
  - Parity: Even, odd or no-parity bit generation and detection
  - Stop bit: 1 or 2 stop bits generation
  - Bit order: LSB-first or MSB-first transfer
- Error detection: Parity, overrun and frame error
- Supports PDMA Interface

Function Descriptions

Serial Data Format

The UART module performs a parallel-to-serial conversion on data that is written to the transmit data register and then sends the data with the following format: Start bit, 7 ~ 9 LSB / MSB first data bits, optional Parity bit and finally 1 ~ 2 Stop bits. The Start bit has the opposite polarity of the data line idle state. The Stop bit is the same as the data line idle state and provides a delay before the next start situation. Both the Start and Stop bits are used for data synchronization during the asynchronous data transmission.

The UART module also performs a serial-to-parallel conversion on the data that is read from the receive data register. It will first check the Parity bit and will then look for a Stop bit. If the Stop bit is not found, the UART module will consider the entire word transmission to have failed and respond with a Framing Error.

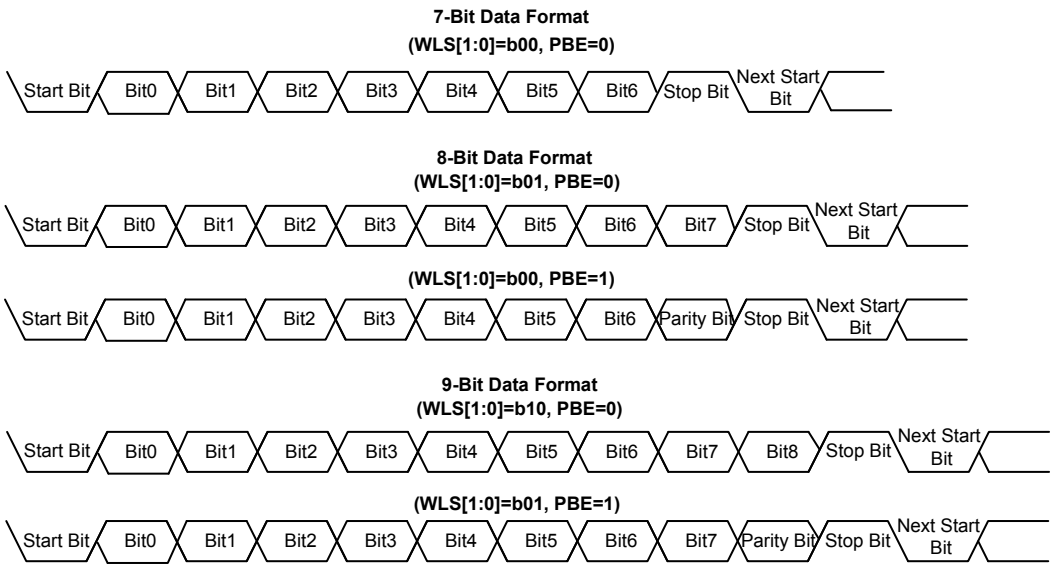


Figure 146. UART Serial Data Format

Baud Rate Generation

The baud rate for the UART receiver and transmitter are both set with the same values. The baud-rate divisor, BRD, has the following relationship with the UART clock which is known as CK_UART.

Baud Rate Clock = CK_UART / BRD

Where CK_UART clock is the APB clock connected to the UART while the BRD range is from 16 to 65535.

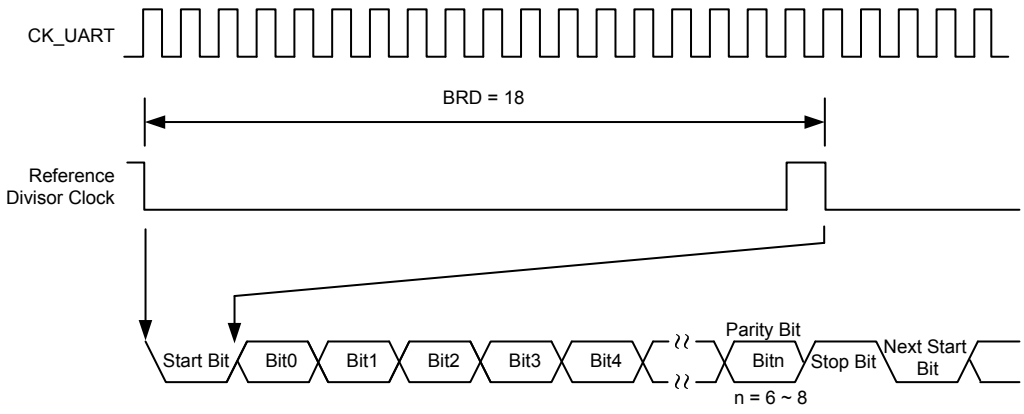


Figure 147. UART Clock CK_UART and Data Frame Timing

Table 57. Baud Rate Deviation Error Calculation – CK_UART = 40 MHz

Baud Rate		CK_UART = 40 MHz		
No.	Kbps	Actual	BRD	Deviation Error Rate
1	2.4	2.4	16667	0.00%
2	9.6	9.6	4167	-0.01%
3	19.2	19.2	2083	0.02%
4	57.6	57.6	694	0.06%
5	115.2	115.3	347	0.06%
6	230.4	229.9	174	-0.22%
7	460.8	459.8	87	-0.22%
8	921.6	930.2	43	0.94%
9	2250	2222.2	18	-1.23%
10	3000	3076.9	13	2.56%

Table 58. Baud Rate Deviation Error Calculation – CK_UART = 48 MHz

Baud Rate		CK_UART = 48 MHz		
No.	Kbps	Actual	BRD	Deviation Error Rate
1	2.4	2.4	20000	0.00%
2	9.6	9.6	5000	0.00%
3	19.2	19.2	2500	0.00%
4	57.6	57.6	833	0.04%
5	115.2	115.1	417	-0.08%
6	230.4	230.8	208	0.16%
7	460.8	461.5	104	0.16%
8	921.6	923.1	52	0.16%
9	2250	2285.7	21	1.59%
10	3000	3000	16	0.00%

Interrupts and Status

The UART can generate interrupts when the following event occurs and the corresponding interrupt enable bits are set:

- Receiver line status interrupts: The interrupts are generated when the overrun error, parity error, framing error or break event occurs for the UART receiver.
- Transmit data register empty interrupt: An interrupt is generated when the content of the transmit data register is transferred to the transmit shift register (TSR).
- Transmit complete interrupt: An interrupt is generated when the transmit data register (TDR) is empty and the content of the transmit shift register (TSR) is also completely shifted.
- Receive data ready interrupt: An interrupt is generated when the content of the receive shift register RDR has been transferred to the URDR register and is ready to read.

PDMA Interface

The PDMA interface is integrated in the UART. The PDMA function can be enabled by setting the TXDMAEN or RXDMAEN bit in the URCR register to 1 in the transmit or receive mode respectively. When the UART transmit data register TDR is empty and the TXDMAEN bit is set to 1, the PDMA function will be activated to move data from a source location into the UART transmit data register TDR.

Similarly, when the received data has been in the UART receive data register RDR and the RXDMAEN bit is set to 1, the PDMA function will be activated to move data from the UART receive data register RDR to a specific destination location. For a mode detailed description on the PDMA configurations, refer to the PDMA chapter.

Register Map

The following table shows the UART registers and reset values.

Table 59. UART Register Map

Register	Offset	Description	Reset Value
URDR	0x000	UART Data Register	0x0000_0000
URCR	0x004	UART Control Register	0x0000_0000
URIER	0x00C	UART Interrupt Enable Register	0x0000_0000
URSIFR	0x010	UART Status & Interrupt Flag Register	0x0000_0180
URDLR	0x024	UART Divider Latch Register	0x0000_0010
URTSTR	0x028	UART Test Register	0x0000_0000

Register Descriptions

UART Data Register – URDR

The register is used to access the UART transmitted and received data.

Offset: 0x000
Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	22	21	20	19	18	17	16
	Reserved							
Type/Reset								
	15	14	13	12	11	10	9	8
	Reserved							DB
Type/Reset								RW 0
	7	6	5	4	3	2	1	0
	DB							
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0

Bits	Field	Descriptions
[8:0]	DB	By reading this register, the UART will return a 7, 8 and 9-bit received data. The DB field bit 8 is valid for the 9-bit mode only and is fixed at 0 for the 8-bit mode. For the 7-bit mode, the DB[6:0] contains the available bits. By writing to this register, the UART will send out 7, 8 or 9-bit transmitted data. The DB field bit 8 is valid for the 9-bit mode only and will be ignored for the 8-bit mode. For the 7-bit mode, the DB[6:0] field contains the available bits.

UART Control Register – URCR

The register specifies the serial parameters such as data length, parity and stop bit for the UART.

Offset: 0x004

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	22	21	20	19	18	17	16
	Reserved							
Type/Reset								
	15	14	13	12	11	10	9	8
	Reserved	BCB	SPE	EPE	PBE	NSB	WLS	
Type/Reset		RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0
	7	6	5	4	3	2	1	0
	RXDMAEN	TXDMAEN	URRXEN	URTXEN	Reserved	TRSM		Reserved
Type/Reset	RW 0	RW 0	RW 0	RW 0		RW 0		

Bits	Field	Descriptions
[14]	BCB	Break Control Bit When this bit is set to 1, the serial data output on the UART TX pin will be forced to the Spacing State (logic 0). This bit acts only on the UART TX output pin and has no effect on the transmitter logic.
[13]	SPE	Stick Parity Enable 0: Disable stick parity 1: Stick Parity bit is transmitted This bit is only available when the PBE bit is set to 1. If both the PBE and SPE bits are set to 1 and the EPE bit is cleared to 0, the transmitted parity bit will be stuck to 1. However, when the PBE and SPE bits are set to 1 and also the EPE bit is set to 1, the transmitted parity bit will be stuck to 0.
[12]	EPE	Even Parity Enable 0: Odd number of logic 1's are transmitted or checked in the data word and parity bits 1: Even number of logic 1's are transmitted or checked in the data word and parity bits This bit is only available when PBE is set to 1.
[11]	PBE	Parity Bit Enable 0: Parity bit is not generated (transmitted data) and checked (receive data) during transfer 1: Parity bit is generated and checked during transfer Note: When the WLS field is set to "10" to select the 9-bit data format, writing to the PBE bit has no effect.
[10]	NSB	Number of "STOP bit" 0: One " STOP bit" is generated in the transmitted data 1: Two "STOP bit" is generated when 8- and 9-bit word length is selected

Bits	Field	Descriptions
[9:8]	WLS	Word Length Select 00: 7 bits 01: 8 bits 10: 9 bits 11: Reserved
[7]	RXDMAEN	UART RX DMA Enable 0: Disable 1: Enable
[6]	TXDMAEN	UART TX DMA Enable 0: Disable 1: Enable
[5]	URRXEN	UART RX Enable 0: Disable 1: Enable
[4]	URTXEN	UART TX Enable 0: Disable 1: Enable
[2]	TRSM	Transfer Mode Selection This bit is used to select the data transfer protocol. 0: LSB first 1: MSB first

UART Interrupt Enable Register – URIER

This register is used to enable the related UART interrupt function. The UART module generates interrupts to the controller when the corresponding events occur and the corresponding interrupt enable bits are set.

Offset: 0x00C

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	22	21	20	19	18	17	16
	Reserved							
Type/Reset								
	15	14	13	12	11	10	9	8
	Reserved							
Type/Reset								

	7	6	5	4	3	2	1	0
	Reserved	BIE	FEIE	PEIE	OEIE	TXCIE	TXDEIE	RXDRIE
Type/Reset		RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0

Bits	Field	Descriptions
[6]	BIE	Break Interrupt Enable 0: Disable interrupt 1: Enable interrupt If this bit is set, an interrupt is generated when the break interrupt is enabled and the BII bit is set in the URSIFR register.
[5]	FEIE	Framing Error Interrupt Enable 0: Disable interrupt 1: Enable interrupt If this bit is set, an interrupt is generated when the framing error interrupt is enabled and the FEI bit is set in the URSIFR register.
[4]	PEIE	Parity Error Interrupt Enable 0: Disable interrupt 1: Enable interrupt If this bit is set, an interrupt is generated when the parity error interrupt is enabled and the PEI bit is set in the URSIFR register.
[3]	OEIE	Overrun Error Interrupt Enable 0: Disable interrupt 1: Enable interrupt If this bit is set, an interrupt is generated when the overrun error interrupt is enabled and the OEI bit is set in the URSIFR register.
[2]	TXCIE	Transmit Complete Interrupt Enable 0: Disable interrupt 1: Enable interrupt If this bit is set, an interrupt is generated when the transmit complete interrupt is enabled and the TXC bit is set in the URSIFR register.

Bits	Field	Descriptions
[1]	TXDEIE	Transmit Data Register Empty Interrupt Enable 0: Disable interrupt 1: Enable interrupt If this bit is set, an interrupt is generated when the transmit data register empty interrupt is enabled and the TXDE bit is set in the URSIFR register.
[0]	RXDRIE	Receive Data Ready Interrupt Enable 0: Disable interrupt 1: Enable interrupt If this bit is set, an interrupt is generated when the receive data ready interrupt is enabled and the RXDR bit is set in the URSIFR register.

UART Status & Interrupt Flag Register – URSIFR

This register contains the corresponding UART status.

Offset: 0x010

Reset value: 0x0000_0180

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	22	21	20	19	18	17	16
	Reserved							
Type/Reset								
	15	14	13	12	11	10	9	8
	Reserved							TXC
Type/Reset								RO 1
	7	6	5	4	3	2	1	0
	TXDE	Reserved	RXDR	BII	FEI	PEI	OEI	Reserved
Type/Reset	RO 1		RO 0	WC 0	WC 0	WC 0	WC 0	

Bits	Field	Descriptions
[8]	TXC	Transmit Complete 0: Either the transmit data register (TDR) or transmit shift register (TSR) is not empty 1: Both the transmit data register (TDR) and transmit shift register (TSR) are empty An interrupt is generated if TXCIE = 1 in the URIER register. This bit is cleared by a write to the URDR register with new data.
[7]	TXDE	Transmit Data Register Empty 0: Transmit data register is not empty 1: Transmit data register is empty The TXDE bit is set by hardware when the content of the transmit data register is transferred to the transmit shift register (TSR). An interrupt is generated if TXEIE = 1 in the URIER register. This bit is cleared by a write to the URDR register with a new data.

Bits	Field	Descriptions
[5]	RXDR	RX Data Ready 0: Receive data register is empty 1: Received data in the receive data register is ready to read This bit is set by hardware when the content of the receive shift register RDR has been transferred to the URDR register. An interrupt is generated if RXDRIE = 1 in the URIER register. It is cleared by a read to the URDR register.
[4]	BII	Break Interrupt Indicator This bit is set to 1 whenever the received data input is held in the “spacing state” (logic 0) for longer than a full character transmission time, which is the total time of “start bit” + “data bits” + “parity” + “stop bits” duration. Writing 1 to this bit clears the flag.
[3]	FEI	Framing Error Indicator This bit is set 1 whenever the received character does not have a valid “stop bit”, which means the stop bit following the last data bit or parity bit is detected as logic 0. Writing 1 to this bit clears the flag.
[2]	PEI	Parity Error Indicator This bit is set to 1 whenever the received character does not have a valid “parity bit”. Writing 1 to this bit clears the flag.
[1]	OEI	Overrun Error Indicator An overrun error will occur only after the receive data register is full and when the next character has been completely received in the receive shift register. The character in the receive shift register will be overwritten when a new character is received in the receive shift register after an overrun event occurs, but the data in the receive shift register will not be transferred to the receive data register. The OEI bit is used to indicate the overrun event as soon as it happens. Writing 1 to this bit clears the flag.

UART Divider Latch Register – URDLR

The register is used to determine the UART clock divided ratio to generate the appropriate baud rate.

Offset: 0x024

Reset value: 0x0000_0010

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	22	21	20	19	18	17	16
	Reserved							
Type/Reset								
	15	14	13	12	11	10	9	8
	BRD							
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0
	7	6	5	4	3	2	1	0
	BRD							
Type/Reset	RW 0	RW 0	RW 0	RW 1	RW 0	RW 0	RW 0	RW 0

Bits	Field	Descriptions
[15:0]	BRD	Baud Rate Divider The 16 bits define the UART clock divider ratio. Baud Rate = CK_UART / BRD Where the CK_UART clock is the clock connected to the UART module. BRD = 16 ~ 65535 for UART mode

UART Test Register – URTSTR

This register controls the UART debug mode.

Offset: 0x028

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	22	21	20	19	18	17	16
	Reserved							
Type/Reset								
	15	14	13	12	11	10	9	8
	Reserved							
Type/Reset								
	7	6	5	4	3	2	1	0
	Reserved						LBM	
Type/Reset							RW 0	RW 0

Bits	Field	Descriptions
[1:0]	LBM	Loopback Test Mode Select 00: Normal Operation 01: Reserved 10: Automatic Echo Mode 11: Loopback Mode

23 USB Device Controller (USB)

Introduction

The USB device controller is compliant with the USB 2.0 full-speed specification. There is one control endpoint know as Endpoint 0 and seven configurable endpoints (EP1 ~ EP7). A 1024-byte EP_SRAM is used for the endpoint buffers. Each endpoint buffer size is programmable by corresponding registers, which provides maximum flexibility for various applications. The integrated USB full-speed transceiver helps to minimize overall system complexity and cost. The USB also contains the suspend and resume features to meet low-power consumption requirement. The accompanying figure shows the USB block diagram.

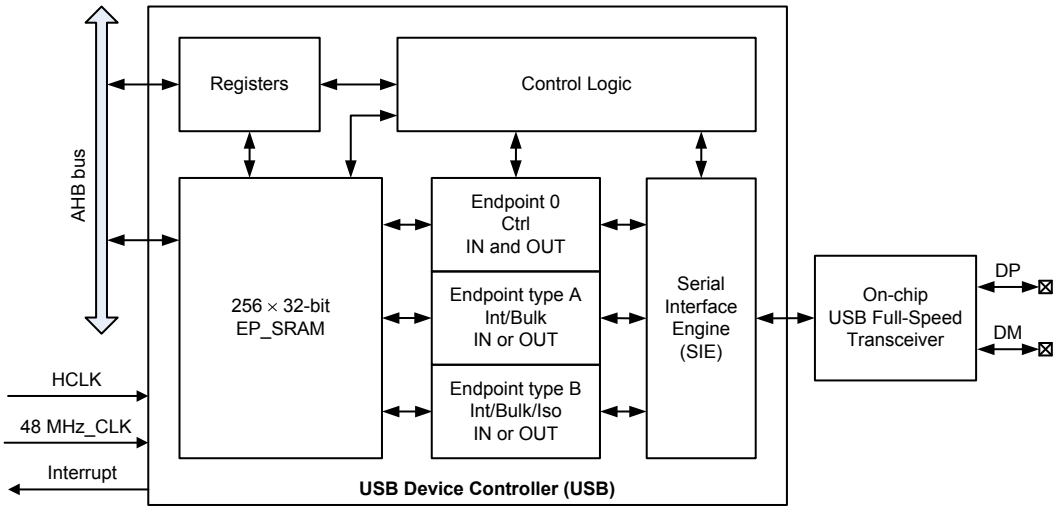


Figure 148. USB Block Diagram

Features

- Complies with USB 2.0 full-speed (12 Mbps) specification
- Fully integrated USB full-speed transceiver
- 1 control endpoint (EP0) for control transfer
- 3 single-buffered endpoints (EP1 ~ EP3) for bulk and interrupt transfer
- 4 double-buffered endpoints (EP4 ~ EP7) for bulk, interrupt and isochronous transfer
- 1,024 bytes EP_SRAM used as endpoint data buffers

Functional Descriptions

Endpoints

The USB Endpoint 0 is the only bidirectional endpoint dedicated to USB control transfer. The device also contains seven unidirectional endpoints for other USB transfer types. There are three endpoints (EP1 ~ EP3) which supports a single buffering function which is used for Bulk and Interrupt IN or OUT data transfer. There are four other endpoints (EP4 ~ EP7) which support single or double buffering functions for Bulk, Interrupt and Isochronous IN or OUT data transfer. The address of the seven unidirectional endpoints (EP1 ~ EP7) can be configured by the application software. The following table lists the endpoint characteristics.

Table 60. Endpoint Characteristics

Endpoint Number	Number Address	Transfer Type	Direction	Buffer Type
0	Fixed	Control	IN and OUT	Single buffering
1 ~ 3	Configurable	Interrupt / Bulk	IN or OUT	Single buffering
4 ~ 7	Configurable	Interrupt / Bulk / Isochronous	IN or OUT	Single or Double buffering

EP_SRAM

The USB controller contains a dedicated memory space, EP_SRAM, which is used for the USB endpoint buffers. The EP_SRAM, which is connected to the APB bus, can be accessed by the MCU and PDMA. The EP_SRAM base address is 0x400A_A000 with an offset which ranges from 0x000 to 0x3FF. The EP_SRAM first two words are reserved for Endpoint 0 to temporarily store the 8-byte SETUP data. Therefore the valid start address of the endpoint buffer should start from 0400A_A008 and align to a 4-byte boundary. Each endpoint buffer size is programmable. The following table lists the maximum USB endpoint buffer size which is compliant with USB 2.0 full-speed device specification.

Table 61. USB Data Types and Buffer Size

Transfer Type	Direction	Supported Buffer Size (Byte)	Bandwidth	CRC	Retrying
Control	Bidirectional	8, 16, 32, 64	Not guaranteed	Yes	Automatic
Bulk	Unidirectional	8, 16, 32, 64	Not guaranteed	Yes	Yes
Interrupt	Unidirectional	≤ 64	Not guaranteed	Yes	Yes
Isochronous	Unidirectional	< 512	Guaranteed	Yes	No

In the following endpoint buffer allocation example, the Endpoint “4” is configured as a double-buffered Bulk IN endpoint while the Endpoint “5” is configured as a double-buffered Bulk OUT endpoint. Each endpoint buffer size is set to 64-bytes.

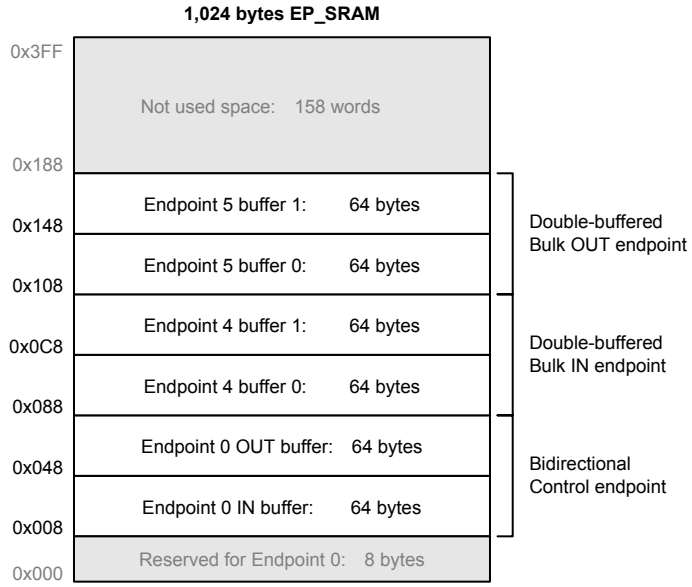


Figure 149. Endpoint Buffer Allocation Example

Serial Interface Engine – SIE

The Serial Interface Engine, SIE, which is connected to the USB full-speed transceiver and internal USB control circuitry provides a temporal buffer for the transmitted and received data. The SIE also decodes the SE0 signal, SE1 signal, J-state, K-state, USB RESET event and End of Packet event signals, EOP, when the USB module receives data, transmits data or transmits the resume signal for remote control. The SIE detects the number of SOF packets and generates the SOF interrupt signal to the USB control circuitry which includes data format conversion from parallel to serial or serial to parallel. It also includes CRC checking and generation, PID decoder, bit-stuffing and debit-stuffing functions.

Double-Buffering

The double buffering function is recommended to be enabled when the corresponding endpoint is specified to be used for Isochronous transfer or high throughput Bulk transfer. The double buffering function stores the preceding data packet sent by the USB host in a simple buffer for the MCU to process and the hardware will ensure that it continues to receive the current data packet in the other buffer during a OUT transaction and vice versa. Using a double buffering function can achieve the highest possible data transfer rate. The details regarding double buffering usage is provided in the corresponding UDBTG and MDBTG control bit description in the USBEPnCSR register where the denotation n ranges from 4 to 7.

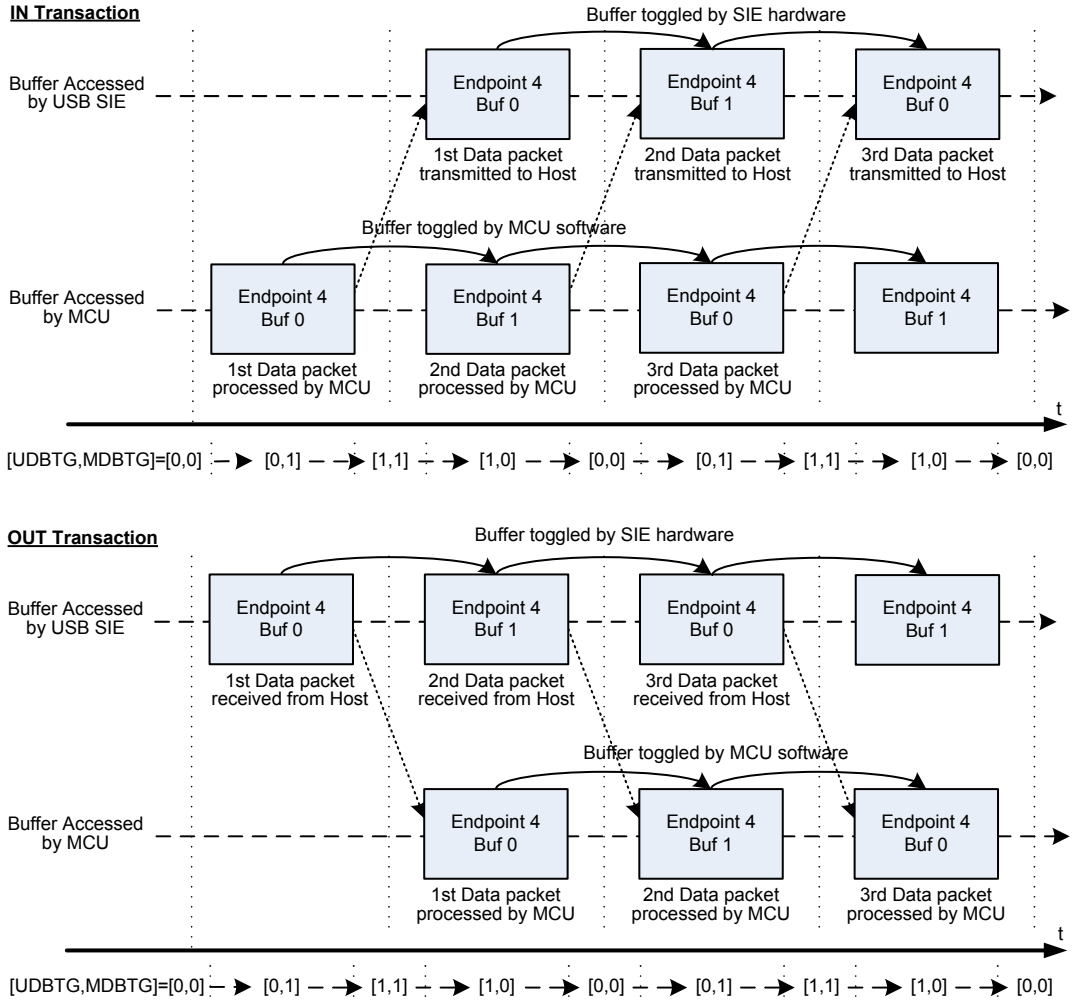


Figure 150. Double-buffering Operation Example

Suspend Mode and Wake-up

According to USB specifications, the device must enter the suspend mode after a 3 ms bus idle time. When the USB device enters the suspend mode, the current from the USB bus must not be greater than 500 μA to meet the specification suspend mode current requirements. The USB control circuitry will generate a suspend interrupt if the bus is in the idle state for 3 ms. Here the software should set the LPMODE and PDWN bits in the USBCSR register to 1. The LPMODE bit is used to determine whether the USB controller enters the low power mode or not by holding the USB bus in a reset condition while the PDWN bit is used to determine if the integrated USB full-speed transceiver is turned off or not.

There are two ways for the USB host to wake up the USB device, one is to send a USB reset signal, SE0, and the other is to send a USB resume signal known as the K-state. After a wake-up signal, regardless of whether a SE0 signal or a K-state is detected, the USB device will be woken up.

Remote Wake-up

As the USB device has a remote wake-up function, it can wake up the USB host by sending a resume request signal by setting the GENRSM bit in the USBCSR register to 1. Once the USB host receives the remote wake-up signal from the USB device, it will send a resume signal to the USB device.

Register Map

The following table shows the USB registers and reset values.

Table 62. USB Register Map

Register	Offset	Description	Reset Value
USBCSR	0x000	USB Control and Status Register	0x0000_00X6
USBIER	0x004	USB Interrupt Enable Register	0x0000_0000
USBISR	0x008	USB Interrupt Status Register	0x0000_0000
USBFCR	0x00C	USB Frame Count Register	0x0000_0000
USBDEVAR	0x010	USB Device Address Register	0x0000_0000
USBEP0CSR	0x014	USB Endpoint 0 Control and Status Register	0x0000_0002
USBEP0IER	0x018	USB Endpoint 0 Interrupt Enable Register	0x0000_0000
USBEP0ISR	0x01C	USB Endpoint 0 Interrupt Status Register	0x0000_0000
USBEP0TCR	0x020	USB Endpoint 0 Transfer Count Register	0x0000_0000
USBEP0CFGR	0x024	USB Endpoint 0 Configuration Register	0x8000_0002
USBEP1CSR	0x028	USB Endpoint 1 Control and Status Register	0x0000_0002
USBEP1IER	0x02C	USB Endpoint 1 Interrupt Enable Register	0x0000_0000
USBEP1ISR	0x030	USB Endpoint 1 Interrupt Status Register	0x0000_0000
USBEP1TCR	0x034	USB Endpoint 1 Transfer Count Register	0x0000_0000
USBEP1CFGR	0x038	USB Endpoint 1 Configuration Register	0x1000_03FF
USBEP2CSR	0x03C	USB Endpoint 2 Control and Status Register	0x0000_0002
USBEP2IER	0x040	USB Endpoint 2 Interrupt Enable Register	0x0000_0000
USBEP2ISR	0x044	USB Endpoint 2 Interrupt Status Register	0x0000_0000
USBEP2TCR	0x048	USB Endpoint 2 Transfer Count Register	0x0000_0000
USBEP2CFGR	0x04C	USB Endpoint 2 Configuration Register	0x1000_03FF
USBEP3CSR	0x050	USB Endpoint 3 Control and Status Register	0x0000_0002
USBEP3IER	0x054	USB Endpoint 3 Interrupt Enable Register	0x0000_0000

Register	Offset	Description	Reset Value
USBEP3ISR	0x058	USB Endpoint 3 Interrupt Status Register	0x0000_0000
USBEP3TCR	0x05C	USB Endpoint 3 Transfer Count Register	0x0000_0000
USBEP3CFGR	0x060	USB Endpoint 3 Configuration Register	0x1000_03FF
USBEP4CSR	0x064	USB Endpoint 4 Control and Status Register	0x0000_0002
USBEP4IER	0x068	USB Endpoint 4 Interrupt Enable Register	0x0000_0000
USBEP4ISR	0x06C	USB Endpoint 4 Interrupt Status Register	0x0000_0000
USBEP4TCR	0x070	USB Endpoint 4 Transfer Count Register	0x0000_0000
USBEP4CFGR	0x074	USB Endpoint 4 Configuration Register	0x1000_03FF
USBEP5CSR	0x078	USB Endpoint 5 Control and Status Register	0x0000_0002
USBEP5IER	0x07C	USB Endpoint 5 Interrupt Enable Register	0x0000_0000
USBEP5ISR	0x080	USB Endpoint 5 Interrupt Status Register	0x0000_0000
USBEP5TCR	0x084	USB Endpoint 5 Transfer Count Register	0x0000_0000
USBEP5CFGR	0x088	USB Endpoint 5 Configuration Register	0x1000_03FF
USBEP6CSR	0x08C	USB Endpoint 6 Control and Status Register	0x0000_0002
USBEP6IER	0x090	USB Endpoint 6 Interrupt Enable Register	0x0000_0000
USBEP6ISR	0x094	USB Endpoint 6 Interrupt Status Register	0x0000_0000
USBEP6TCR	0x098	USB Endpoint 6 Transfer Count Register	0x0000_0000
USBEP6CFGR	0x09C	USB Endpoint 6 Configuration Register	0x1000_03FF
USBEP7CSR	0x0A0	USB Endpoint 7 Control and Status Register	0x0000_0002
USBEP7IER	0x0A4	USB Endpoint 7 Interrupt Enable Register	0x0000_0000
USBEP7ISR	0x0A8	USB Endpoint 7 Interrupt Status Register	0x0000_0000
USBEP7TCR	0x0AC	USB Endpoint 7 Transfer Count Register	0x0000_0000
USBEP7CFGR	0x0B0	USB Endpoint 7 Configuration Register	0x1000_03FF

Register Descriptions

USB Control and Status Register – USBCSR

This register specifies the USB control bits and USB data line status.

Offset: 0x000

Reset value: 0x0000_00X6

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	22	21	20	19	18	17	16
	Reserved							
Type/Reset								
	15	14	13	12	11	10	9	8
	Reserved				DPWKEN	DPPUEN	SRAMRSTC	ADRSET
Type/Reset					RW 0	RW 0	RW 0	RW 0
	7	6	5	4	3	2	1	0
	RXDM	RXDP	GENRSM	Reserved	LPMODE	PDWN	FRES	Reserved
Type/Reset	RO X	RO X	RW 0		RW 0	RW 1	RW 1	

Bits	Field	Descriptions
[11]	DPWKEN	DP Wake Up Enable 0: Disable DP wake up 1: Enable DP wake up
[10]	DPPUEN	DP Pull Up Enable 0: Disable DP pull up 1: Enable DP pull up
[9]	SRAMRSTC	EP_SRAM reset condition 0: Reset EP_SRAM when (DP, DM) = (0,0) 1: User can access EP_SRAM in spite of (DP, DM) state
[8]	ADRSET	This bit is used to determine when the USB SIE updates the device address with the value of the USBDEVAR register. 0: The SIE updates the device address immediately after an address is written into the USBDEVAR register. 1: The SIE updates the device address after the USB Host has successfully read the data from the device by the IN operation. This bit is cleared by the SIE after the device address is updated.
[7]	RXDM	Received DM Line Status This bit is used to observe the status of DM data line status at the end of suspend routines to determine whether a wakeup event has occurred.
[6]	RXDP	Received DP Line Status This bit is used to observe the status of DP data line status at the end of suspend routines to determine whether a wakeup event has occurred.
[5]	GENRSM	Resume Request Generation Control This bit is used to generate a resume request which is sent to the USB host by writing 1 into this bit location. The USB remote wakeup function is always enabled. This bit will be cleared to 0 after a resume signal, sent by the USB host, has been received.

Bits	Field	Descriptions
[3]	LPMODE	Low-power Mode Control This bit is used to determine the USB operating mode. Setting this bit will force the USB to enter the low-power mode. When USB bus traffic, known as a wakeup event, is detected by the hardware, this bit should be cleared by software. 0: Exit the Low-power mode 1: Enter the Low-power mode
[2]	PDWN	Power Down Mode Control Setting this bit will power down the full-speed USB PHY transceiver. This will disconnect the USB PHY transceiver from the USB bus. 0: Exit the Power-Down mode 1: Enter the Power-Down mode
[1]	FRES	Force USB Reset Control This bit is used to reset the USB circuitry. Setting this bit will force the USB into a reset state until the software clears it. A USB reset interrupt will be generated if the corresponding interrupt enable bit in the USBIER register is set to 1. All related USB registers are reset to their default values. 0: Release USB reset 1: Force USB reset

Table 63. Resume Event Detection

[RXDP, RXDM] Status	Wakeup Event	Required Resume Software Action
00	Root reset	None
10	None (noise on bus)	Go back to suspend mode
01	Root resume	None
11	Not allowed (noise on bus)	Go back to suspend mode

USB Interrupt Enable Register – USBIER

This register specifies the USB interrupt enable control.

Offset: 0x004

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	22	21	20	19	18	17	16
	Reserved							
Type/Reset								
	15	14	13	12	11	10	9	8
	EP7IE	EP6IE	EP5IE	EP4IE	EP3IE	EP2IE	EP1IE	EP0IE
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0
	7	6	5	4	3	2	1	0
		Reserved	ESOFIE	SUSPIE	RSMIE	URSTIE	SOFIE	UGIE
Type/Reset			RW 0	RW 0	RW 0	RW 0	RW 0	RW 0

Bits	Field	Descriptions
[15:8]	EPnIE	Endpoint n Interrupt Enable Control (n = 0 ~ 7) 0: Disable interrupt 1: Enable interrupt
[5]	ESOFIE	Expected Start of Frame (ESOF) Interrupt Enable Control 0: Disable ESOF interrupt 1: Enable ESOF interrupt
[4]	SUSPIE	Suspend Interrupt Enable Control 0: Disable suspend interrupt 1: Enable suspend interrupt
[3]	RSMIE	Resume Interrupt Enable Control 0: Disable Resume interrupt 1: Enable Resume interrupt
[2]	URSTIE	USB Reset Interrupt Enable Control 0: Disable USB Reset interrupt 1: Enable USB Reset interrupt
[1]	SOFIE	Start of Frame (SOF) Interrupt Enable Control 0: Disable SOF interrupt 1: Enable SOF interrupt
[0]	UGIE	USB Global Interrupt Enable Control 0: Disable USB Global interrupt 1: Enable USB Global interrupt This bit must be set to 1 to enable the corresponding USB interrupt function. If this bit is cleared to 0, the relevant USB interrupt will not be generated. However, the corresponding interrupt flags will still be asserted.

USB Interrupt Status Register – USBISR

This register specifies the USB interrupt status.

Offset: 0x008

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	22	21	20	19	18	17	16
	Reserved							
Type/Reset								
	15	14	13	12	11	10	9	8
	EP7IF	EP6IF	EP5IF	EP4IF	EP3IF	EP2IF	EP1IF	EP0IF
Type/Reset	WC 0	WC 0	WC 0	WC 0	WC 0	WC 0	WC 0	WC 0
	7	6	5	4	3	2	1	0
	Reserved	ESOFIF	SUSPIF	RSMIF	URSTIF	SOFIF	Reserved	
Type/Reset		RW 0	WC 0	WC 0	WC 0	WC 0		

Bits	Field	Descriptions
[15:8]	EPnIF	Endpoint n Interrupt Flag (n = 0 ~ 7) This bit is set by the hardware to indicate the generation of relevant endpoint interrupt. Writing 1 into this bit to clear it. It is important to note that the interrupt flag can only be cleared when the endpoint interrupt status bit in the USBEPnISR register is equal to 0.
[6]	ESOFIF	Expected Start of Frame Interrupt Flag This bit is set by the hardware when an SOF packet is expected to be received. The USB host sends an SOF (Start of Frame) packet each millisecond. If the USB device hardware does not receive it properly, an ESOF interrupt will be generated when the ESOFIE bit in the USBIER register is set to 1. If three consecutive ESOF interrupts are generated, which means that the SOF packet has been missed 3 times, the SUSPIF will be set to 1. This bit will be set to 1 when the missing SOF packets occur if the timer is not yet locked. This bit can be read or written. However, only 0 can be written into this bit. Writing 1 has no effect.
[5]	SUSPIF	Suspend Interrupt Flag This bit is set by the hardware when no data transfer has occurred for 3 ms, indicating that a suspend request has been sent from the USB host. The suspend condition check is enabled immediately after a USB reset. This bit is cleared to 0 by writing 1.
[4]	RSMIF	Resume Interrupt Flag This bit is set by the hardware. When this bit is set to 1, this means that a device resume has occurred. This bit is cleared to 0 by writing 1.

Bits	Field	Descriptions
[3]	URSTIF	USB Reset Interrupt Flag This bit is set by the hardware when the USB reset has been detected. When a USB reset occurs, the internal protocol state machine will be reset and an USB reset interrupt will be generated if the URSTIE bit in the USBIER register is set to 1. Data reception and transmission are disabled until the URSTIF bit is cleared to 0. The USB configuration related registers (USBCSR, USBIER, USBISR, USBFCR and USBDEVAR) will not be reset by a USB reset event except for the USB device address (USBDEVAR), this is to ensure that a USB reset interrupt can be safely excited and any data transactions immediately followed by the USB reset can be completely accessed by the software. Therefore the microcontroller must properly reset these registers. The USB endpoint related registers (USBEPnCSR, USBEPnISR and USBEPnTCR) are also reset by a USB reset event, however, the endpoint configuration (USBEPnCFGR) and interrupt enable (USBEPnIER) registers are not affected by the USB reset event and will remain unchanged. This bit is cleared to 0 by writing 1.
[2]	SOFIF	SOF Interrupt Flag This bit is set by the hardware when a start-of-frame packet has been received. This bit is cleared to 0 by writing 1.

USB Frame Count Register – USBFCR

This register specifies the lost Start-of-Frame number and the USB frame count.

Offset: 0x00C

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	22	21	20	19	18	17	16
	Reserved						LSOF	SOFLCK
Type/Reset						RO 0	RO 0	RO 0
	15	14	13	12	11	10	9	8
	Reserved						FRNUM	
Type/Reset						RO 0	RO 0	RO 0
	7	6	5	4	3	2	1	0
	FRNUM							
Type/Reset	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0

Bits	Field	Descriptions
[18:17]	LSOF	Lost Start-of-Frame number These bits are written and incremented by 1 by the hardware each time the ESOFIF bit is set. It is used to count the number of lost SOF packets. When a SOF packet has been received, these bits are cleared.
[16]	SOFLCK	Start-of-Frame Lock Flag This bit is set by the hardware when SOF packets have been received before the frame timer times out. Once this flag is set to 1, the frame number which is sent from the USB host will be loaded into the Frame Number field in the USBFCR register. If there is no SOF packet has been received during the 1ms frame time duration, this bit will be cleared to 0.
[10:0]	FRNUM	rame Number This field stores the frame number received from the USB host.

USB Device Address Register – USBDEVAR

This register specifies the USB device address.

Offset: 0x010

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	22	21	20	19	18	17	16
	Reserved							
Type/Reset								
	15	14	13	12	11	10	9	8
	Reserved							
Type/Reset								
	7	6	5	4	3	2	1	0
	Reserved	DEVA						
Type/Reset		RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0

Bits	Field	Descriptions
[6:0]	DEVA	Device Address This field is used to specify the USB device address. This field is cleared when a USB reset event occurs.

USB Endpoint 0 Control and Status Register – USBEP0CSR

This register specifies the Endpoint 0 control and status.

Offset: 0x014

Reset value: 0x0000_0002

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	22	21	20	19	18	17	16
	Reserved							
Type/Reset								
	15	14	13	12	11	10	9	8
	Reserved							
Type/Reset								

	7	6	5	4	3	2	1	0
		Reserved	STLRX	NAKRX	DTGRX	STLTX	NAKTX	DTGTX
Type/Reset			RW 0	RW 0	RW 0	RW 0	RW 1	RW 0

Bits	Field	Descriptions
[5]	STLRX	STALL Status for reception (OUT) transfer This bit is set to 1 by the application software and then returns a STALL signal in the handshake phase of an OUT transaction if a functional error is detected. This means that a control request delivered from the USB host is not supported by the USB device. The STALL status is cleared by the hardware circuitry when a new SETUP token is received. This bit can be read and written and can only be toggled by writing 1.
[4]	NAKRX	NAK Status for reception (OUT) transfer This bit is toggled from 0 to 1 by the hardware circuitry, which will result in a NAK signal in the handshake phase of an OUT transaction after an ACK signal has been transmitted. This means that the USB device will be temporarily unable to accept data from the USB host. Therefore, more time will be required for the received data to be properly processed. This bit can be read and written and can only be toggled by writing 1.
[3]	DTGRX	Data Toggle Status for reception (OUT) transfer This bit contains the expected value of the data toggle bit (0 = DATA0, 1 = DTAT1) for the next data packet to be received. When the current valid data packet is received and the corresponding ACK signal is sent to the USB host by the USB device, the hardware circuitry will toggle this bit and the device will be ready to receive the next data packet. For Endpoint 0, the hardware circuitry will toggle this bit to 1 after the SETUP token is received as Endpoint 0 is addressed. This bit can also be toggled by the software to initialize its value for certain applications. This bit can be read and written and can only be toggled by writing 1.
[2]	STLTX	STALL Status for transmission (IN) transfer This bit is set to 1 by the application software and then returns a STALL signal in response to an IN token if a functional error is detected. This means that the USB device is unable to transmit data. The STALL status is cleared by the hardware circuitry when a new SETUP token is received. This bit can be read and written and can only be toggled by writing 1.

Bits	Field	Descriptions
[1]	NAKTX	NAK Status for transmission (IN) transfer This bit is toggled from 0 to 1 by the hardware circuitry, which will result in a NAK signal in the handshake phase of an IN transaction after an ACK signal has been received. It indicates that the USB device is temporarily unable to transmit data to the USB host. Therefore, there will be more time for the application software to properly prepare the data to be transmitted. This bit can be read and written and can only be toggled by writing 1.
[0]	DTGTX	Data Toggle Status for transmission (IN) transfer This bit contains the required value of the data toggle bit (0 = DATA0, 1 = DATA1) for the next data packet to be transmitted. When the current data packet is transmitted by the USB device and the corresponding ACK signal sent by the USB host is received, the hardware circuitry will toggle this bit and the next data packet will be transmitted. For Endpoint 0, the hardware circuitry will toggle this bit to 1 after the SETUP token is received as Endpoint 0 is addressed. This bit can also be toggled by the software to initialize its value for certain applications. This bit can be read and written and can only be toggled by writing 1.

USB Endpoint 0 Interrupt Enable Register – USBEP0IER

This register specifies the Endpoint 0 interrupt control bits.

Offset: 0x018

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	22	21	20	19	18	17	16
	Reserved							
Type/Reset								
	15	14	13	12	11	10	9	8
	Reserved				ZLRXIE	SDERIE	SDRXIE	STRXIE
Type/Reset					RW 0	RW 0	RW 0	RW 0
	7	6	5	4	3	2	1	0
	UERIE	STLIE	NAKIE	IDTXIE	ITRXIE	ODOVIE	ODRXIE	OTRXIE
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0

Bits	Field	Descriptions
[11]	ZLRXIE	Zero Length Data Received Interrupt Enable Control 0: Disable interrupt 1: Enable interrupt
[10]	SDERIE	SETUP Data Error Interrupt Enable Control 0: Disable interrupt 1: Enable interrupt
[9]	SDRXIE	SETUP Data Received Interrupt Enable Control 0: Disable interrupt 1: Enable interrupt

Bits	Field	Descriptions
[8]	STRXIE	SETUP Token Received Interrupt Enable Control 0: Disable interrupt 1: Enable interrupt
[7]	UERIE	USB Error Interrupt Enable Control 0: Disable interrupt 1: Enable interrupt
[6]	STLIE	STALL Transmitted Interrupt Enable Control 0: Disable interrupt 1: Enable interrupt
[5]	NAKIE	NAK Transmitted Interrupt Enable Control 0: Disable interrupt 1: Enable interrupt
[4]	IDTXIE	IN Data Transmitted Interrupt Enable Control 0: Disable interrupt 1: Enable interrupt
[3]	ITRXIE	IN Token Received Interrupt Enable Control 0: Disable interrupt 1: Enable interrupt
[2]	ODOVIE	OUT Data Buffer Overrun Interrupt Enable Control 0: Disable interrupt 1: Enable interrupt
[1]	ODRXIE	OUT Data Received Interrupt Enable Control 0: Disable interrupt 1: Enable interrupt
[0]	OTRXIE	OUT Token Received Interrupt Enable Control 0: Disable interrupt 1: Enable interrupt

USB Endpoint 0 Interrupt Status Register – USBEP0ISR

This register specifies the Endpoint 0 interrupt status.

Offset: 0x01C

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	22	21	20	19	18	17	16
	Reserved							
Type/Reset								
	15	14	13	12	11	10	9	8
	Reserved				ZLRXIF	SDERIF	SDRXIF	STRXIF
Type/Reset					WC 0	WC 0	WC 0	WC 0
	7	6	5	4	3	2	1	0
	UERIF	STLIF	NAKIF	IDTXIF	ITRXIF	ODOVIF	ODRXIF	OTRXIF
Type/Reset	WC 0	WC 0	WC 0	WC 0	WC 0	WC 0	WC 0	WC 0

Bits	Field	Descriptions
[11]	ZLRXIF	Zero Length Data Received Interrupt Flag This bit is set by the hardware when a zero length data packet is received. This bit is cleared by hardware when a SETUP Token is received or by writing 1.
[10]	SDERIF	SETUP Data Error Interrupt Flag This bit is set by the hardware when the SETUP data packet length is not 8 bytes. This bit is cleared by hardware when a SETUP Token is received or by writing 1.
[9]	SDRXIF	SETUP Data Received Interrupt Flag This bit is set by the hardware when a SETUP data packet from the USB host has been received. This bit is cleared by the hardware when a SETUP Token is received or by writing 1. If the received SETUP data is not accessed by the application software before the next SETUP packet is received, the SETUP data buffer will be overwritten.
[8]	STRXIF	SETUP Token Received Interrupt Flag This bit is set by the hardware when a SETUP token is received and is cleared by writing 1.
[7]	UERIF	USB Error Interrupt Flag This bit is set by the hardware when an error occurs during the Endpoint 0 transaction. This bit is cleared by hardware when a SETUP Token is received or by writing 1.
[6]	STLIF	STALL Transmitted Interrupt Flag This bit is set by the hardware when a STALL signal is sent in response to an IN or OUT transaction. This bit is cleared by hardware when a SETUP Token is received or by writing 1.
[5]	NAKIF	NAK Transmitted Interrupt Flag This bit is set by the hardware when a NAK signal is sent in response to an IN or OUT transaction. This bit is cleared by hardware when a SETUP Token is received or by writing 1.

Bits	Field	Descriptions
[4]	IDTXIF	IN Data Transmitted Interrupt Flag This bit is set by the hardware when a data packet is transmitted to and then an ACK signal is received from the USB host. This bit is cleared by hardware when a SETUP Token is received or by writing 1.
[3]	ITRXIF	IN Token Received Interrupt Flag This bit is set by the hardware when the IN token is received from the USB host. This bit is cleared by hardware when a SETUP Token is received or by writing 1.
[2]	ODOVIF	OUT Data Buffer Overrun Interrupt Flag This bit is set by the hardware when the number of received data bytes is larger than the endpoint buffer size. This bit is cleared by hardware when a SETUP Token is received or by writing 1.
[1]	ODRXIF	OUT Data Received Interrupt Flag This bit is set by the hardware when a data packet is successfully received from the USB host and then an ACK signal is sent to the USB host. This bit is cleared by hardware when a SETUP Token is received or by writing 1.
[0]	OTRXIF	OUT Token Received Interrupt Flag This bit is set by the hardware when the OUT token is received from the USB host. This bit is cleared by hardware when a SETUP Token is received or by writing 1.

USB Endpoint 0 Transfer Count Register – USBEP0TCR

This register specifies the Endpoint 0 data transfer byte count.

Offset: 0x020

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	22	21	20	19	18	17	16
	Reserved	RXCNT						
Type/Reset		RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0
	15	14	13	12	11	10	9	8
	Reserved							
Type/Reset								
	7	6	5	4	3	2	1	0
	Reserved	TXCNT						
Type/Reset		RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0

Bits	Field	Descriptions
[22:16]	RXCNT	Reception Byte Count The bit field contains the number of data bytes received by Endpoint 0 in the preceding SETUP transaction.
[6:0]	TXCNT	Transmission Byte Count The bit field contains the number of data bytes to be transmitted by Endpoint 0 in the next IN token. If the value of this field is zero, it indicates that a zero length packet will be sent.

USB Endpoint 0 Configuration Register – USBEP0CFGR

This register specifies the Endpoint 0 configurations.

Offset: 0x024

Reset value: 0x8000_0002

	31	30	29	28	27	26	25	24
	EPEN	Reserved			EPADR			
Type/Reset	RO 1				RO 0	RO 0	RO 0	RO 0
	23	22	21	20	19	18	17	16
	Reserved							EPLEN
Type/Reset								RW 0
	15	14	13	12	11	10	9	8
	EPLEN							EPBUFA
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0
	7	6	5	4	3	2	1	0
	EPBUFA							
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 1	RW 0

Bits	Field	Descriptions
[31]	EPEN	Endpoint Enable Control This bit is always set to 1 by the hardware circuitry to always enable Endpoint 0.
[27:24]	EPADR	Endpoint Address This field is always set to 0 by the hardware circuitry.
[16:10]	EPLEN	Endpoint Buffer Length This field is used to specify the control transfer packet size which can be 8, 16, 32 or 64 bytes as defined in the USB full-speed standard specification.
[9:0]	EPBUFA	Endpoint Buffer Address This field is used to specify the star address of the Endpoint 0 buffer allocated in the EP_SRAM. It starts from 0x008 and should be aligned to 4-byte boundary. Start address of EP0 IN buffer = EPBUFA Start address of EP0 OUT buffer = EPBUFA + EPLEN

USB Endpoint 1 ~ 3 Control and Status Register – USBEPnCSR, n = 1 ~ 3

This register specifies the Endpoint 1 ~ 3 control and status bit.

Offset: 0x028 (n = 1), 0x03C (n = 2), 0x050 (n = 3)

Reset value: 0x0000_0002

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	22	21	20	19	18	17	16
	Reserved							
Type/Reset								
	15	14	13	12	11	10	9	8
	Reserved							
Type/Reset								

	7	6	5	4	3	2	1	0
		Reserved	STLRX	NAKRX	DTGRX	STLTX	NAKTX	DTGTX
Type/Reset			RW 0	RW 0	RW 0	RW 0	RW 1	RW 0

Bits	Field	Descriptions
[5]	STLRX	STALL bit for reception transfers This bit is set to 1 by the application software if a functional error has been detected. This bit can be read and written and can only be toggled by writing 1. It can also be toggled by the software to initialize the value under certain conditions.
[4]	NAKRX	NAK bit for reception transfers This bit is toggled from 0 to 1 by the hardware circuitry, which will result in a NAK signal in the handshake phase of an OUT transaction after an ACK signal has been transmitted. It means that the USB device will be temporarily unable to accept data from the USB host until the received data is properly processed. This bit can be read and written and can be only toggled by writing 1.
[3]	DTGRX	Data Toggle bit for reception transfers This bit contains the expected value of the data toggle bit (0 = DATA0, 1 = DATA1) for the next data packet to be received. When the current valid data packet is received and the corresponding ACK signal is sent to the USB host by the USB device, the hardware circuitry will toggle this bit and the device will be ready to receive the next data packet. This bit can be read and written and can only be toggled by writing 1. This bit can also be toggled by the software to initialize its value under certain conditions.
[2]	STLTX	STALL bit for transmission transfers This bit is set to 1 by the application software if a functional error has been detected. This bit can be read and written and can be only toggled by writing 1. It can also be toggled by the software to initialize its value under certain conditions.
[1]	NAKTX	NAK bit for transmission transfers This bit is toggled from 0 to 1 by the hardware circuitry, which will result in a NAK signal in the handshake phase of an IN transaction after an ACK signal has been received. It means that the USB device will be temporarily unable to transmit data packet until the data to be transmitted is appropriately prepared by the application software. This bit can be read and written and can be only toggled by writing 1.

Bits	Field	Descriptions
[0]	DTGTX	Data Toggle bit for transmission transfers. This bit contains the required value of the data toggle bit (0 = DATA0, 1 = DATA1) for the next data packet to be transmitted. When the current data packet is transmitted by the USB device and the corresponding ACK signal sent from the USB host is received, the hardware circuitry will toggle this bit and then the next data packet will be transmitted. This bit can be read and written and can only be toggled by writing 1. It can also be toggled by the software to initialize its value under certain conditions.

USB Endpoint 1 ~ 3 Interrupt Enable Register – USBEPnIER, n = 1 ~ 3

This register specifies the Endpoint 1 ~ 3 interrupt enable control bits.

Offset: 0x02C (n = 1), 0x040 (n = 2), 0x054 (n = 3)

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	22	21	20	19	18	17	16
	Reserved							
Type/Reset								
	15	14	13	12	11	10	9	8
	Reserved							
Type/Reset								

	7	6	5	4	3	2	1	0
	UERIE	STLIE	NAKIE	IDTXIE	ITRXIE	ODOVIE	ODRXIE	OTRXIE
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0

Bits	Field	Descriptions
[7]	UERIE	USB Error Interrupt Enable Control. 0: Disable interrupt 1: Enable interrupt
[6]	STLIE	STALL Transmitted Interrupt Enable Control. 0: Disable interrupt 1: Enable interrupt
[5]	NAKIE	NAK Transmitted Interrupt Enable Control. 0: Disable interrupt 1: Enable interrupt
[4]	IDTXIE	IN Data Transmitted Interrupt Enable Control. 0: Disable interrupt 1: Enable interrupt
[3]	ITRXIE	IN Token Received Interrupt Enable Control. 0: Disable interrupt 1: Enable interrupt
[2]	ODOVIE	OUT Data Buffer Overrun Interrupt Enable Control 0: Disable interrupt 1: Enable interrupt
[1]	ODRXIE	OUT Data Received Interrupt Enable Control 0: Disable interrupt 1: Enable interrupt
[0]	OTRXIE	OUT Token Received Interrupt Enable Control 0: Disable interrupt 1: Enable interrupt

USB Endpoint 1 ~ 3 Interrupt Status Register – USBEPnISR, n = 1 ~ 3

This register specifies the Endpoint 1 ~ 3 interrupt status.

Offset: 0x030 (n = 1), 0x044 (n = 2), 0x058 (n = 3)

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	22	21	20	19	18	17	16
	Reserved							
Type/Reset								
	15	14	13	12	11	10	9	8
	Reserved							
Type/Reset								

	7	6	5	4	3	2	1	0
	UERIF	STLIF	NAKIF	IDTXIF	ITRXIF	ODOVIF	ODRXIF	OTRXIF
Type/Reset	WC 0	WC 0	WC 0	WC 0	WC 0	WC 0	WC 0	WC 0

Bits	Field	Descriptions
[7]	UERIF	USB Error Interrupt Flag. This bit is set by the hardware when an error occurs during the transaction. Writing 1 into this status bit will clear it to 0.
[6]	STLIF	STALL Transmitted Interrupt Flag. This bit is set by hardware circuitry when a STALL signal is sent in response to an IN or OUT token and is cleared to 0 by writing 1.
[5]	NAKIF	NAK Transmitted Interrupt Flag. This bit is set by hardware circuitry when a NAK signal is sent in response to an IN or OUT token and is cleared to 0 by writing 1.
[4]	IDTXIF	IN Data Transmitted Interrupt Flag. This bit is set by hardware circuitry when a data packet is successfully transmitted to the host in response to an IN token and an ACK signal is received. Writing 1 into this status bit will clear it to 0.
[3]	ITRXIF	IIN Token Received Interrupt Flag This bit is set by the hardware circuitry when the endpoint receives an IN token from the host and is cleared to 0 by writing 1.
[2]	ODOVIF	OUT Data Buffer Overrun Interrupt Flag This bit is set by the hardware circuitry when the received data byte count is larger than the corresponding endpoint OUT data buffer size. Writing 1 into this status bit will clear it to 0.
[1]	ODRXIF	OUT Data Received Interrupt Flag This bit is set by the hardware circuitry when a data packet is successfully received from the host for an OUT token and when an endpoint n ACK signal is sent to the host. Writing 1 into this status bit will clear it to 0.
[0]	OTRXIF	OUT Token Received Interrupt Flag This bit is set by the hardware circuitry when the endpoint receives an OUT token from the host and is cleared to 0 by writing 1.

USB Endpoint 1 ~ 3 Transfer Count Register – USBEPnTCR, n = 1 ~ 3

This register specifies the Endpoint 1 ~ 3 transfer byte count.

Offset: 0x034 (n = 1), 0x048 (n = 2), 0x05C (n = 3)

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	22	21	20	19	18	17	16
	Reserved							
Type/Reset								
	15	14	13	12	11	10	9	8
	Reserved							TCNT
Type/Reset								RW 0
	7	6	5	4	3	2	1	0
	TCNT							
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0

Bits	Field	Descriptions
[8:0]	TCNT	Transfer Byte Count This field contains the number of bytes received by the endpoint n in the preceding OUT transaction or the number of bytes to be transmitted by the endpoint n in the next IN transaction.

USB Endpoint 1 ~ 3 Configuration Register – USBEPnCFGR, n = 1 ~ 3

This register specifies the Endpoint 1 ~ 3 configurations.

Offset: 0x038 (n = 1), 0x04C (n = 2), 0x060 (n = 3)

Reset value: 0x1000_03FF

	31	30	29	28	27	26	25	24
	EPEN	Reserved	EPTYPE	EPDIR	EPADR			
Type/Reset	RW 0		RW 0	RW 1	RW 0	RW 0	RW 0	RW 0
	23	22	21	20	19	18	17	16
	Reserved							EPLEN
Type/Reset								RW 0
	15	14	13	12	11	10	9	8
	EPLEN							EPBUFA
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 1	RW 1
	7	6	5	4	3	2	1	0
	EPBUFA							
Type/Reset	RW 1	RW 1	RW 1	RW 1	RW 1	RW 1	RW 1	RW 1

Bits	Field	Descriptions
[31]	EPEN	Enable Control 0: Disable the endpoint n 1: Enable the endpoint n
[29]	EPTYPE	Transfer Type This bit is set to 0 by the hardware circuitry to specify that the endpoint n transfer type is an Interrupt or Bulk transfer type.
[28]	EPDIR	Transfer Direction 0: OUT 1: IN
[27:24]	EPADR	Endpoint Address The EPADR field value can be assigned by the application software to specify the address of the endpoint n. It is important to note that this EPADR field should not be set to 0; otherwise, the endpoint will be disabled.
[16:10]	EPLEN	Buffer Length This field is used to specify the endpoint n data packet size. The field value must be word-aligned to a 4-byte boundary. The maximum size in this field can be 64 bytes which is the maximum payload as defined in the USB full-speed standard specification. Note that the EPLEN value should not be assigned to 0 which will result in the endpoint being disabled.
[9:0]	EPBUFA	Endpoint Buffer Address This field is used to specify the endpoint n data buffer start address which ranges from 0x008 to 0x3FC in the EP_SRAM which has a capacity of 1024 bytes and whose field value must be a multiple of 4.

USB Endpoint 4 ~ 7 Control and Status Register – USBEPnCSR, n = 4 ~ 7

This register specifies the Endpoint 4 ~ 7 control and status bits.

Offset: 0x064 (n = 4), 0x078 (n = 5), 0x08C (n = 6), 0x0A0 (n = 7)

Reset value: 0x0000_0002

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	22	21	20	19	18	17	16
	Reserved							
Type/Reset								
	15	14	13	12	11	10	9	8
	Reserved							
Type/Reset								

	7	6	5	4	3	2	1	0
	UDBTG	MDBTG	STLRX	NAKRX	DTGRX	STLTX	NAKTX	DTGTX
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 1	RW 0

Bits	Field	Descriptions
[7]	UDBTG	USB Double Buffer Toggle bit The UDBTG and MDBTG bits are used to indicate which data buffer is accessed by the USB SIE hardware and which data buffer is accessed by the MCU software if the double buffering function is enabled. The UDBTG bit will be toggled by the SIE hardware circuitry after the current buffer operation is complete. After the UDBTG bit is toggled by the SIE, an NAK signal will be sent automatically to the USB host by the hardware circuitry. Therefore, the data transfer will be stopped temporarily until the data in the other buffer has been properly setup after which the MDBTG bit is toggled by the MCU application software. The following tables show the double buffering operation and the UDBTG and MDBTG bit status for an IN or OUT transaction.

Transaction Type	UDBTG	MDBTG	Buffer read by SIE	Buffer written by MCU
IN	0	0	None*	EP_BUF0
	0	1	EP_BUF0	EP_BUF1
	1	1	None*	EP_BUF1
	1	0	EP_BUF1	EP_BUF0

Transaction Type	UDBTG	MDBTG	Buffer written by SIE	Buffer read by MCU
OUT	0	0	None*	EP_BUF0
	0	1	EP_BUF0	EP_BUF1
	1	1	None*	EP_BUF1
	1	0	EP_BUF1	EP_BUF0

* Means the USB device sends a NAK signal to the USB host using the hardware circuitry.

The UDBTG and MDBTG bits setting procedure for the double buffering function is shown in the following example:

[UDBTG, MDBTG] = [0, 0] → [0, 1] → [1, 1] → [1, 0] → [0, 0] → [0, 1] → [1, 1] → [1, 0] →...

Bits	Field	Descriptions
[6]	MDBTG	MCU Double Buffer Toggle bit The MDBTG bit is used to indicate which data buffer is accessed by the MCU if the double buffering function is enabled. It can be toggled to switch to the other buffer by the MCU application software after the data in the current buffer accessed by the MCU has been properly setup. The double buffering operation together with the UDBTG and MDBTG bits are shown in the preceding two tables for the UDBTG bit definition
[5]	STLRX	STALL bit for reception transfers This bit is set to 1 by the application software if a functional error has been detected. This bit can be read and written and can only be toggled by writing 1. It can also be toggled by software to initialize its value under certain conditions.
[4]	NAKRX	NAK bit for reception transfers This bit is toggled from 0 to 1 by the hardware circuitry, which will result in a NAK signal in the handshake phase of an OUT transaction after an ACK signal has been transmitted. It means that the USB device will be temporarily unable to accept data from the USB host until the received data is properly processed. If the endpoint is defined as an Isochronous transfer type, this bit is not available for usage. The hardware will not change the NAKRX bit status after a complete transaction. This bit can be read and written and can be only toggled by writing 1.
[3]	DTGRX	Data Toggle bit for reception transfers If the endpoint is not used for Isochronous transfer, this bit is available for usage. This bit contains the expected value of the data toggle bit (0 = DATA0, 1 = DATA1) for the next data packet to be received. When the current valid data packet is received and the corresponding ACK signal is sent to the USB host by the USB device, the hardware circuitry will toggle this bit and the device will be ready to receive the next data packet. If the endpoint is defined as an Isochronous transfer type, this bit is not used since no data toggling is used and only the DATA0 packet will be transferred for normal Isochronous transfers. This bit can be read and written and can only be toggled by writing 1. This bit can also be toggled by the software to initialize its value under certain conditions.
[2]	STLTX	STALL bit for transmission transfers This bit is set to 1 by the application software if there a functional error has been detected. This bit can be read and written and can be only toggled by writing 1. It can be toggled by the software to initialize its value under certain conditions.
[1]	NAKTX	NAK bit for transmission transfers This bit is toggled from 0 to 1 by the hardware circuitry, which will result in a NAK signal in the handshake phase of an IN transaction after an ACK signal has been received. It means that the USB device will be temporarily unable to transmit a data packet until the data to be transmitted is properly setup by the application software. If the endpoint is defined as an Isochronous transfer type, then this bit is not available for usage. The hardware will not change the NAKTX bit status after a complete transaction. This bit can be read and written and can be only toggled by writing 1. It can also be toggled by the software to initialize its value under certain conditions.

Bits	Field	Descriptions
[0]	DTGTX	Data Toggle bit for transmission transfers If the endpoint is not used for Isochronous transfer, this bit is available for usage. This bit contains the required value of the data toggle bit (0 = DATA0, 1 = DATA1) for the next data packet to be transmitted. When the current data packet is transmitted by the USB device and the corresponding ACK signal sent from the USB host is received, the hardware circuitry will toggle this bit and then the next data packet will be transmitted. If the endpoint is used for Isochronous transfer, this bit is not used since no data toggling is used and only the DATA0 packet will be transferred for normal Isochronous transfer. This bit can be read and written and can only be toggled by writing 1. It can also be toggled by the software to initialize its value under certain conditions.

USB Endpoint 4 ~ 7 Interrupt Enable Register – USBEPnIER, n = 4 ~ 7

This register specifies the Endpoint 4 ~ 7 interrupt enable control bits.

Offset: 0x068 (n = 4), 0x07C (n = 5), 0x090 (n = 6), 0x0A4 (n = 7)

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	22	21	20	19	18	17	16
	Reserved							
Type/Reset								
	15	14	13	12	11	10	9	8
	Reserved							
Type/Reset								

	7	6	5	4	3	2	1	0
	UERIE	STLIE	NAKIE	IDTXIE	ITRXIE	ODOVIE	ODRXIE	OTRXIE
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0

Bits	Field	Descriptions
[7]	UERIE	USB Error Interrupt Enable Control 0: Disable interrupt 1: Enable interrupt
[6]	STLIE	STALL Transmitted Interrupt Enable Control 0: Disable interrupt 1: Enable interrupt
[5]	NAKIE	NAK Transmitted Interrupt Enable Control 0: Disable interrupt 1: Enable interrupt
[4]	IDTXIE	IN Data Transmitted Interrupt Enable Control 0: Disable interrupt 1: Enable interrupt
[3]	ITRXIE	IN Token Received Interrupt Enable Control 0: Disable interrupt 1: Enable interrupt
[2]	ODOVIE	OUT Data Buffer Overrun Interrupt Enable Control 0: Disable interrupt 1: Enable interrupt
[1]	ODRXIE	OUT Data Received Interrupt Enable Control 0: Disable interrupt 1: Enable interrupt
[0]	OTRXIE	OUT Token Received Interrupt Enable Control 0: Disable interrupt 1: Enable interrupt

USB Endpoint 4 ~ 7 Interrupt Status Register – USBEPnISR, n = 4 ~ 7

This register specifies the Endpoint 4 ~ 7 interrupt status.

Offset: 0x06C (n = 4), 0x080 (n = 5), 0x094 (n = 6), 0x0A8 (n = 7)

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	22	21	20	19	18	17	16
	Reserved							
Type/Reset								
	15	14	13	12	11	10	9	8
	Reserved							
Type/Reset								

	7	6	5	4	3	2	1	0
	UERIF	STLIF	NAKIF	IDTXIF	ITRXIF	ODOVIF	ODRXIF	OTRXIF
Type/Reset	WC 0	WC 0	WC 0	WC 0	WC 0	WC 0	WC 0	WC 0

Bits	Field	Descriptions
[7]	UERIF	USB Error Interrupt flag This bit is set by the hardware circuitry when an error occurs during the transaction. Writing 1 into this status bit will clear it to 0.
[6]	STLIF	STALL Transmitted Interrupt flag This bit is set by the hardware circuitry when a STALL signal is sent in response to an IN or OUT token and is cleared to 0 by writing 1.
[5]	NAKIF	NAK Transmitted Interrupt flag This bit is set by the hardware circuitry when a NAK signal is sent in response to an IN or OUT token and is cleared to 0 by writing 1.
[4]	IDTXIF	IN Data Transmitted Interrupt flag This bit is set by the hardware circuitry when a data packet is successfully transmitted to the host in response to an IN token and an ACK signal is received. Writing 1 into this status bit will clear it to 0.
[3]	ITRXIF	IN Token Received Interrupt flag This bit is set by the hardware circuitry when the endpoint receives an IN token from the host and is cleared to 0 by writing 1.
[2]	ODOVIF	OUT Data Buffer Overrun Interrupt flag This bit is set by the hardware circuitry when the received data byte count is larger than the endpoint OUT data buffer size. Writing 1 into this status bit will clear it to 0.
[1]	ODRXIF	OUT Data Received Interrupt flag This bit is set by the hardware circuitry when a data packet is successfully received from the host for an OUT-token and an ACK signal is sent to the host. Writing 1 into this status bit will clear it to 0.
[0]	OTRXIF	OUT Token Received Interrupt flag This bit is set by the hardware circuitry when the endpoint receives an OUT token from the host and is cleared to 0 by writing 1.

USB Endpoint 4 ~ 7 Transfer Count Register – USBEPnTCR, n = 4 ~ 7

This register specifies the Endpoint 4 ~ 7 transfer byte count.

Offset: 0x070 (n = 4), 0x084 (n = 5), 0x098 (n = 6), 0x0AC (n = 7)

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved						TCNT1	
Type/Reset							RW 0	RW 0
	23	22	21	20	19	18	17	16
	TCNT1							
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0
	15	14	13	12	11	10	9	8
	Reserved						TCNT0	
Type/Reset							RW 0	RW 0
	7	6	5	4	3	2	1	0
	TCNT0							
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0

Bits	Field	Descriptions
[25:16]	TCNT1	Buffer 1 Transfer Byte Count This bit field contains the number of data bytes received by the endpoint n buffer 1 in the preceding OUT transaction or the number of data bytes to be transmitted by the endpoint n buffer 1 in the next IN transaction.
[9:0]	TCNT0	Buffer 0 Transfer Byte Count This bit field contains the number of data bytes received by the endpoint n buffer 0 in the preceding OUT transaction or the number of data bytes to be transmitted by the endpoint n buffer 0 in the next IN transaction. Only the TCNT0 field is used for the endpoint data transfer count when the endpoint is configured as a single-buffering transfer type.

USB Endpoint 4 ~ 7 Configuration Register – USBEPnCFGR, n = 4 ~ 7

This register specifies the Endpoint 4 ~ 7 configurations.

Offset: 0x074 (n = 4), 0x088 (n = 5), 0x09C (n = 6), 0x0B0 (n = 7)

Reset value: 0x1000_03FF

	31	30	29	28	27	26	25	24
	EPEN	Reserved	EPTYPE	EPDIR	EPADR			
Type/Reset	RW 0		RW 0	RW 1	RW 0	RW 0	RW 0	RW 0
	23	22	21	20	19	18	17	16
	SDBS	Reserved			EPLEN			
Type/Reset	RW 0				RW 0	RW 0	RW 0	RW 0
	15	14	13	12	11	10	9	8
	EPLEN						EPBUFA	
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 1	RW 1
	7	6	5	4	3	2	1	0
	EPBUFA							
Type/Reset	RW 1	RW 1	RW 1	RW 1	RW 1	RW 1	RW 1	RW 1

Bits	Field	Descriptions
[31]	EPEN	Enable Control 0: Disable the endpoint n 1: Enable the endpoint n
[29]	EPTYPE	Transfer Type 0: Interrupt or Bulk transfer type 1: Isochronous transfer type
[28]	EPDIR	Transfer Direction 0: OUT 1: IN
[27:24]	EPADR	Endpoint Address The EPADR field can be configured by the application software to specify the address of endpoint n. It is important to note that this EPADR field should not be set to 0; otherwise, the endpoint n will be disabled.
[23]	SDBS	Single-Buffering or Double-Buffering Selection 0: Single-buffering 1: Double-buffering If the SDBS bit is set to 1, the endpoint buffer size is twice that of the EPLEN value: - Endpoint Buffer 0 start address is EPBUFA - Endpoint Buffer 1 start address is (EPBUFA + EPLEN)
[19:10]	EPLEN	Buffer Length This field is used to specify the endpoint n data packet size whose field value must be word-aligned to a 4-byte boundary. Note that the endpoint will be disabled if the EPLEN value is assigned to 0.
[9:0]	EPBUFA	Buffer Address This field is used to specify the endpoint n data buffer start address which ranges from 0x008 to 0x3FC in the EP_SRAM which has a capacity of 1024 bytes where the endpoint transfer data is stored. Note that the buffer start address value must be a multiple of 4.

24 Peripheral Direct Memory Access (PDMA)

Introduction

The Peripheral Direct Memory Access circuitry, PDMA, provides 6 unidirectional channels for dedicated peripherals to implement the peripheral-to-memory and memory-to-peripheral data transfer. The memory-to-memory data transfer such as the FLASH-to-SRAM or SRAM-to-SRAM type is also supported and requested by the application program. Each PDMA channel configuration is independent. The PDMA channel transfer is split into multiple block transactions and the size of a block is equal to the block length multiplied by the data width.

Features

- 6 unidirectional PDMA channels
- Memory-to-peripheral, peripheral-to-memory and memory-to-memory data transfer
- 8-bit, 16-bit and 32-bit width data transfer
- Software and hardware requested data transfer with configurable channel priority
- Linear address, circular address and fixed address modes
- 4 transfer event flags – Transfer complete, Half Transfer, Block End and Transfer Error
- Auto-Reload function

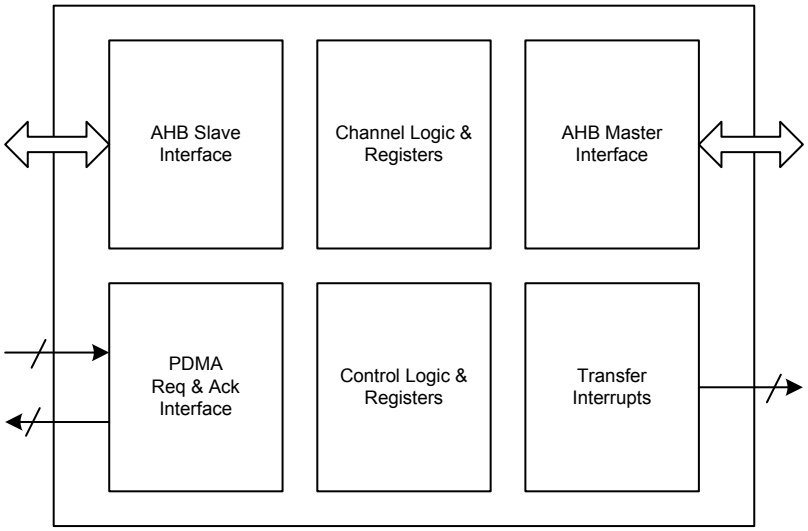


Figure 151. PDMA Block Diagram

Functional Description

AHB Master

The PDMA is an AHB master connected to other AHB peripherals such as the FLASH memory, the SRAM memory and the AHB-to-APB bridges through the bus-matrix. The CPU and PDMA can access different AHB slaves at the same time via the bus-matrix.

PDMA Channel

There are 6 unidirectional PDMA channels used to support data transfer between the peripherals and the memory. The configuration and operation of each PDMA channel is independent. For a bidirectional transfer application, two PDMA channels are required. Each PDMA channel is designed to support the dedicated multiple peripherals with the same registers. Therefore, one PDMA channel only can service one peripheral at the same time. The related registers of the PDMA channel are limited to be accessed with 32-bit operation; otherwise a system hard fault event will occur.

PDMA Request Mapping

The multiple requests from the peripherals (ADC, SPI , I^2C, USART and so on) are simply logically ANDed before entering the PDMA, that means that only one request must be enabled at a time in each PDMA channel. Refer to Figure 152: PDMA Request Mapping Architecture and detailed peripheral IP requests mapping table is shown as the Table 64. The peripheral DMA requests can be independently activated / de-activated by programming the DMA control bit in the registers of the corresponding peripheral.

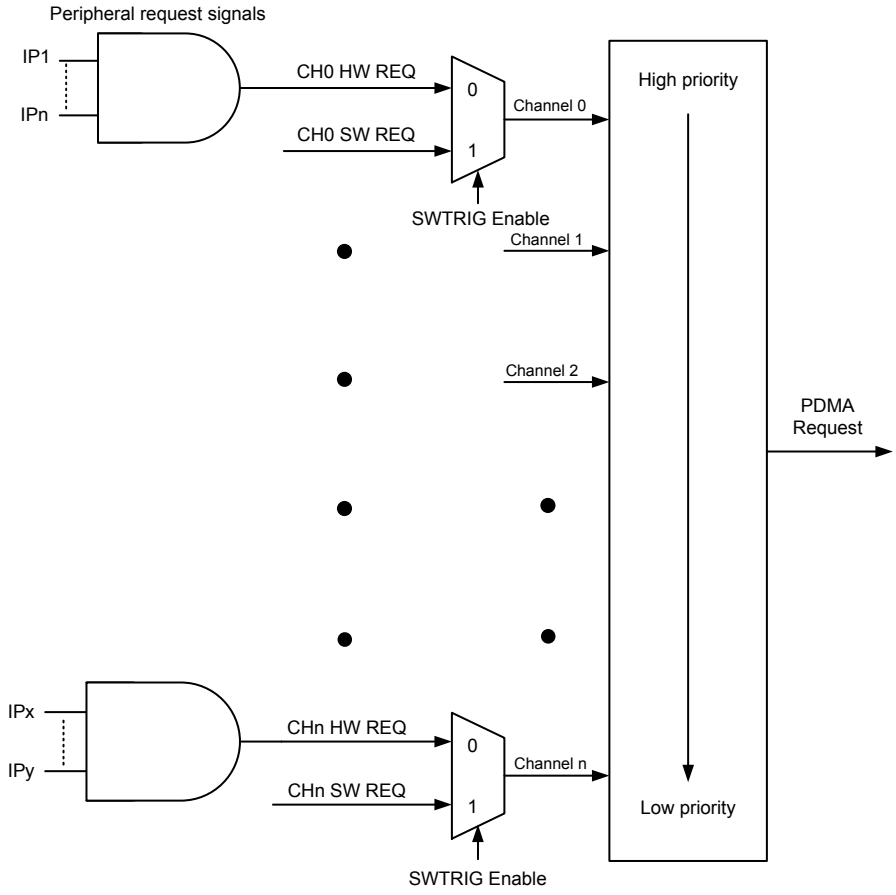


Figure 152. PDMA Request Mapping Architecture

Table 64. PDMA Channel Assignments

IP (x = 0,1)	PDMA Channel Number					
	CH0	CH1	CH2	CH3	CH4	CH5
ADC	ADC					
SPI, QSPI	QSPI_RX	QSPI_TX			SPI_RX	SPI_TX
USART	USR_RX	USR_TX				
UART			UR_RX	UR_TX		
I^2C			I2C_RX		I2C_TX	
GPTM	GT_CH1 GT_CH3	GT_CH2 GT_UEV	GT_CH0 GT_TRIG			
I^2S		I2S_RX	I2S_TX			
MIDI				MIDI_IN	MIDI_OUT	

Channel transfer

A PDMA channel transfer is split into multiple block transactions with PDMA arbitration occurring at the end of each block transaction. Although these channel transfers can all be activated, there is only one block transaction being transferred through the bus at a time. The channel transfer sequence depends upon the channel priority setting of each PDMA channel. The total transfer size is calculated from the block transaction count and block size. The block size is equal to the product of the block length and data bit width. For an efficient transfer, it is recommended that the block length is set as a multiple of 4.

The total transfer data size calculation is shown as the following equation:

A PDMA channel total transfer data size = Block transaction count × (Block length × Data width)

Channel Priority

The PDMA provides four priority levels, known as very high, high, medium and low, which can be configured by the application software. The PDMA also provides two methods to determine the channel priority. One is determined by application software configuration and the other is determined by the fixed hardware channel number. The PDMA arbitration processor will first check the software configuring channel priority level used to request the PDMA to provide the data transfer services. If more than one channel has the same priority, the channel with a smaller channel number will have priority over one with a larger channel number after arbitration.

Note that the highest priority channel will not occupy the PDMA service all the time when other lower priority channel requests are pending. The highest priority channel will be skipped for one block transaction time duration after one block transaction is complete. Then a block transaction requested by the second priority channel will be performed. After a block transaction of the second priority channel is complete, the PDMA arbitration processor will re-check all of the requested channel priority with the exception of the second priority channel since the second priority channel will be excluded after the end of a block transaction. Therefore, a block data transaction of the higher priority channel will be serviced and this channel will be excluded from the priority arbitration at the end of the block transaction. The PDMA will keep transferring the data using the method described above until all of the requested channel data transfer is complete. Refer to the accompanying figure for an example which shows the PDMA channel arbitration and scheduling.

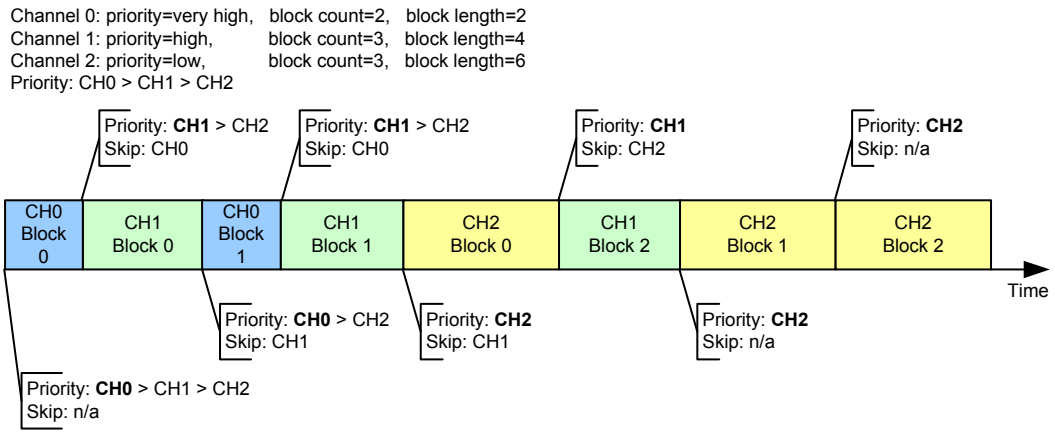


Figure 153. PDMA Channel Arbitration and Scheduling Example

Transfer Request

For a peripheral-to-memory or memory-to-peripheral transfer, one peripheral hardware request will trigger one block transaction of the dedicated PDMA channel. However, a complete data transfer of the relevant dedicated PDMA channel will be triggered when a software request occurs. It is recommended that the PDMA channel is configured to have a lower priority level and a smaller block length which is requested by the software for memory-to-memory data copy applications.

Address Mode

The PDMA provides three kinds of address modes which are the linear address, circular address and fixed address modes. These different address modes are used to support different kinds of source and destination address arrangements. The following table shows the detailed address mode combinations.

Table 65. PDMA Address Modes

Source Address Mode	Destination Address Mode
Linear Increment / Decrement Address	Linear Increment / Decrement Address
Linear Increment / Decrement Address	Circular Increment / Decrement Address
Linear Increment / Decrement Address	Fixed Address
Circular Increment / Decrement Address	Linear Increment / Decrement Address
Circular Increment / Decrement Address	Circular Increment / Decrement Address
Fixed Address	Linear Increment / Decrement Address
Fixed Address	Fixed Address

Linear Address Mode

After data is transferred, the current address will be increased or decreased by 1, 2 or 4 depending upon the data bit width setting.

Circular Address Mode

After data is transferred, the current address will be increased or decreased by 1, 2 or 4 depending upon the data bit width setting. When a block transaction is complete, the current address is loaded with the configured start address.

Fixed Address Mode

After data is transferred, the current address remains unchanged.

Auto-Reload

When the auto-reload control bit, AUTORLn, in the PDMA channel n control register PDMACHnCR is set, both the channel n current address and the channel n current transfer size will be automatically reloaded with the corresponding start value after the current PDMA channel data transfer has totally completed. The channel n will still be activated and the next relative PDMA request can be serviced without any re-configuration using the application software.

Transfer Interrupt

There are five transfer events during which the interrupts can be asserted for each PDMA channel. These are the block transaction end (BE), half transfer (HT), transfer complete (TC), transfer error (TE) and global transfer event (GE). Setting the corresponding control bits in the PDMA interrupt enable register PDMAIER will enable the relevant interrupt events. The global interrupt event, GE, will be generated if any of the four interrupt events including the BE, HT, TC or TE occurs. Clearing the BE, HT, TC or TE event flag will also clear the GE flag. Clearing the GE flag will automatically clear all other event flags. The TE interrupt event will occur when the PDMA accesses a system reserved address space or when the PDMA receives a request but the corresponding transfer size setting is equal to zero.

Register Map

The following table shows the PDMA registers and the reset values.

Table 66. PDMA Register Map

Register	Offset	Description	Reset Value
PDMA Channel 0 Registers			
PDMACH0CR	0x000	PDMA Channel 0 Control Register	0x0000_0000
PDMACH0SADR	0x004	PDMA Channel 0 Source Address Register	0x0000_0000
PDMACH0DADR	0x008	PDMA Channel 0 Destination Address Register	0x0000_0000
PDMACH0TSR	0x010	PDMA Channel 0 Transfer Size Register	0x0000_0000
PDMACH0CTSR	0x014	PDMA Channel 0 Current Transfer Size Register	0x0000_0000
PDMA Channel 1 Registers			
PDMACH1CR	0x018	PDMA Channel 1 Control Register	0x0000_0000
PDMACH1SADR	0x01C	PDMA Channel 1 Source Address Register	0x0000_0000
PDMACH1DADR	0x020	PDMA Channel 1 Destination Address Register	0x0000_0000
PDMACH1TSR	0x028	PDMA Channel 1 Transfer Size Register	0x0000_0000
PDMACH1CTSR	0x02C	PDMA Channel 1 Current Transfer Size Register	0x0000_0000
PDMA Channel 2 Registers			
PDMACH2CR	0x030	PDMA Channel 2 Control Register	0x0000_0000
PDMACH2SADR	0x034	PDMA Channel 2 Source Address Register	0x0000_0000
PDMACH2DADR	0x038	PDMA Channel 2 Destination Address Register	0x0000_0000
PDMACH2TSR	0x040	PDMA Channel 2 Transfer Size Register	0x0000_0000
PDMACH2CTSR	0x044	PDMA Channel 2 Current Transfer Size Register	0x0000_0000
PDMA Channel 3 Registers			
PDMACH3CR	0x048	PDMA Channel 3 Control Register	0x0000_0000
PDMACH3SADR	0x04C	PDMA Channel 3 Source Address Register	0x0000_0000
PDMACH3DADR	0x050	PDMA Channel 3 Destination Address Register	0x0000_0000
PDMACH3TSR	0x058	PDMA Channel 3 Transfer Size Register	0x0000_0000
PDMACH3CTSR	0x05C	PDMA Channel 3 Current Transfer Size Register	0x0000_0000
PDMA Channel 4 Registers			
PDMACH4CR	0x060	PDMA Channel 4 Control Register	0x0000_0000
PDMACH4SADR	0x064	PDMA Channel 4 Source Address Register	0x0000_0000
PDMACH4DADR	0x068	PDMA Channel 4 Destination Address Register	0x0000_0000
PDMACH4TSR	0x070	PDMA Channel 4 Transfer Size Register	0x0000_0000
PDMACH4CTSR	0x074	PDMA Channel 4 Current Transfer Size Register	0x0000_0000

Register	Offset	Description	Reset Value
PDMA Channel 5 Registers			
PDMACH5CR	0x078	PDMA Channel 5 Control Register	0x0000_0000
PDMACH5SADR	0x07C	PDMA Channel 5 Source Address Register	0x0000_0000
PDMACH5DADR	0x080	PDMA Channel 5 Destination Address Register	0x0000_0000
PDMACH5TSR	0x088	PDMA Channel 5 Transfer Size Register	0x0000_0000
PDMACH5CTSR	0x08C	PDMA Channel 5 Current Transfer Size Register	0x0000_0000
PDMA Global Register			
PDMAISR	0x120	PDMA Interrupt Status Register	0x0000_0000
PDMAISCR	0x128	PDMA Interrupt Status Clear Register	0x0000_0000
PDMAIER	0x130	PDMA Interrupt Enable Register	0x0000_0000

Register Descriptions

PDMA Channel n Control Register – PDMACHnCR, n = 0 ~ 5

This register is used to specify the PDMA channel n data transfer configuration.

Offset: 0x000 (0), 0x018 (1), 0x030 (2), 0x048 (3), 0x060 (4), 0x078 (5)

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	22	21	20	19	18	17	16
	Reserved							
Type/Reset								
	15	14	13	12	11	10	9	8
	Reserved				AUTORLn	FIXAENn	CHnPRI	
Type/Reset					RW 0	RW 0	RW 0	RW 0
	7	6	5	4	3	2	1	0
	SRCAMODn	SRCAINCn	DSTAMODn	DSTAINCn		DWIDTHn	SWTRIGn	CHnEN
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0

Bits	Field	Descriptions
[11]	AUTORLn	Channel n Auto Reload Enable Control 0: Disable Auto Reload function 1: Enable Auto Reload function If this bit is set to 1 to enable the auto-reload function, both the channel n current address and the channel n current transfer size will be reloaded with the relevant start value and the PDMA channel n will still be activated when a transfer is complete. If this bit is cleared to 0, the channel n current address and the channel n current transfer size will remain unchanged and the PDMA channel n will be disabled after a transfer completion.
[10]	FIXAENn	Channel n Fixed Address Enable control 0: Disable fixed address function in the circular address mode 1: Enable fixed address function in the circular address mode Note that this bit is only available when the source or destination address mode is set to be in the circular address mode. For example, the source address mode is set as in the linear address mode and the destination address mode is set as in the circular mode. If this bit is set to enable the fixed address function, then the source address mode will still be in the linear address but the destination address mode will be in the fixed address mode instead of the circular address mode.
[9:8]	CHnPRI	Channel n Priority 00: Low 01: Medium 10: High 11: Very high The CHnPRI field is used to configure the channel priority using the application program. If there are more than one channel which have the same software configured priority level, the channel with the smaller channel number will have priority to transfer one block of data after the arbitration.

Bits	Field	Descriptions
[7]	SRCAMODn	Channel n Source Address Mode selection 0: Linear address mode 1: Circular address mode In the linear address mode, the current source address value can be incremented or decremented, determined by the SRCAINCn bit value during a complete transfer. In the circular address mode, the current source address value can be incremented or decremented which is also determined by the SRCAINCn bit value during a block transfer and will be loaded with the lower 16-bit value of the PDMACHnSADR register, which will be regarded as the current source address when a block transaction has completed.
[6]	SRCAINCn	Channel n Source Address Increment control 0: Increment 1: Decrement This bit is used to determine whether the current source address is increased or decreased during a complete transfer in the linear address mode or a block transfer in the circular address mode.
[5]	DSTAMODn	Channel n Destination Address Mode selection 0: Linear address mode 1: Circular address mode In linear address mode, the current destination address value can be incremented or decremented, determined by the DSTAINCn bit value during a complete transfer. In the circular address mode, the current destination address value can be incremented or decremented which is also determined by the DSTAINCn bit value during a block transfer and will be loaded with the lower 16-bit value of the PDMACHnDADR register, which will be regarded as the current destination address when a block transfer has completed.
[4]	DSTAINCn	Channel n Destination Address Increment Control 0: Increment 1: Decrement This bit is used to determine if the current destination address is increased or decreased during a complete transfer in the linear address mode or a block transfer in the circular address mode.
[3:2]	DWIDTHn	Data Bit Width selection 00: 8-bit 01: 16-bit 10: 32-bit 11: Reserved The field is used to select the data bit width of the corresponding PDMA channel n.
[1]	SWTRIGn	Software Trigger control 0: No operation 1: Software triggered transfer request Setting this bit will generate a memory-to-memory software transfer request on the corresponding PDMA channel n. It is automatically cleared when a transfer has completely finished.

Bits	Field	Descriptions
[0]	CHnEN	Channel n Enable control 0: Disable the PDMA channel n 1: Enable the PDMA channel n Setting this bit will enable a software or hardware transfer request on the PDMA channel n. It is automatically cleared by hardware when a transfer has completed with the auto-reload function being disabled. However, if the AUTORLn bit is set to 1 to enable the auto-reload function, this bit will be remain high to enable the PDMA channel n function for the next transfer request instead of automatically being cleared by hardware after a transfer has finished.

PDMA Channel n Source Address Register – PDMACHnSADR, n = 0 ~ 5

This register specifies the source address of the PDMA channel n.

Offset: 0x004 (0), 0x01C (1), 0x034 (2), 0x04C (3), 0x064 (4), 0x07C (5)

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	SADRn							
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0
	23	22	21	20	19	18	17	16
	SADRn							
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0
	15	14	13	12	11	10	9	8
	SADRn							
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0
	7	6	5	4	3	2	1	0
	SADRn							
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0

Bits	Field	Descriptions
[31:0]	SADRn	Channel n Source Address The register is used to specify the 32-bit source address of the PDMA channel n.

PDMA Channel n Destination Address Register – PDMACHnDADR, n = 0 ~ 5

This register specifies the destination address of the PDMA channel n.

Offset: 0x008 (0), 0x020 (1), 0x038 (2), 0x050 (3), 0x068 (4), 0x080 (5)

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	DADRn							
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0
	23	22	21	20	19	18	17	16
	DADRn							
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0
	15	14	13	12	11	10	9	8
	DADRn							
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0
	7	6	5	4	3	2	1	0
	DADRn							
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0

Bits	Field	Descriptions
[31:0]	DADRn	Channel n Destination Address The register is used to specify the 32-bit destination address of the PDMA channel n.

PDMA Channel n Transfer Size Register – PDMACHnTSR, n = 0 ~ 5

This register is used to specify the block transaction count and block transaction length.

Offset: 0x010 (0), 0x028 (1), 0x040 (2), 0x058 (3), 0x070 (4), 0x088 (5)

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	BLKCNTn							
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0
	23	22	21	20	19	18	17	16
	BLKCNTn							
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0
	15	14	13	12	11	10	9	8
	Reserved							
Type/Reset								
	7	6	5	4	3	2	1	0
	BLKLENn							
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0

Bits	Field	Descriptions
[31:16]	BLKCNTn	Channel n Block Transaction Count BLKCNTn represents the number of block transactions for a channel n complete transfer. The capacity of a complete transfer is the product of the BLKCNTn and BLKLENn values. The maximum BLKCNTn value is 65,535.
[7:0]	BLKLENn	Channel n Block Length The BLKLENn represents the length of a data block. The data width is defined by the DWIDTHn field in the PDMACHnCR register. The maximum BLKLENn value is 255.

PDMA Channel n Current Transfer Size Register – PDMACHnCTSR, n = 0 ~ 5

This register is used to indicate the current block transaction count.

Offset: 0x014 (0), 0x02C (1), 0x044 (2), 0x05C (3), 0x074 (4), 0x08C (5)

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	CBLKCNTn							
Type/Reset	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
	CBLKCNTn							
Type/Reset	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
	Reserved							
Type/Reset								
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
	Reserved							
Type/Reset								

Bits	Field	Descriptions
[31:16]	CBLKCNTn	Channel n Current Block Count The CBLKCNTn field is a 16-bit read-only value indicating the number of data blocks that remain to be transferred. After a data block has transferred completely, the CBLKCNTn value will be decremented by 1. Writing a new value to the BLKCNTn field in the PDMACHnTSR register will update the CBLKCNTn field value.

PDMA Interrupt Status Register – PDMAISR

This register is used to indicate the corresponding interrupt status of the PDMA channel 0 ~ 5.

Offset: 0x120

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved		TEISTA5	TCISTA5	HTISTA5	BEISTA5	GEISTA5	TEISTA4
Type/Reset			RO 0	RO 0	RO 0	RO 0	RO 0	RO 0
	23	22	21	20	19	18	17	16
	TCISTA4	HTISTA4	BEISTA4	GEISTA4	TEISTA3	TCISTA3	HTISTA3	BEISTA3
Type/Reset	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0
	15	14	13	12	11	10	9	8
	GEISTA3	TEISTA2	TCISTA2	HTISTA2	BEISTA2	GEISTA2	TEISTA1	TCISTA1
Type/Reset	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0
	7	6	5	4	3	2	1	0
	HTISTA1	BEISTA1	GEISTA1	TEISTA0	TCISTA0	HTISTA0	BEISTA0	GEISTA0
Type/Reset	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0

Bits	Field	Descriptions
[29], [24], [19], [14], [9], [4]	TEISTAn	Channel n Transfer Error Interrupt Status (n = 0 ~ 5) 0: No Transfer Error occurs 1: Transfer Error occurs This bit is set by hardware and is cleared by writing a “1” into the corresponding interrupt status clear bit in the PDMAISCR register. A Transfer error will occur when the PDMA accesses a system reserved address space or the PDMA receives a request but when the corresponding transfer capacity is equal to zero.
[28], [23], [18], [13], [8], [3]	TCISTAn	Channel n Transfer Complete Interrupt Status (n = 0 ~ 5) 0: No Transfer Completion Occurs 1: Transfer Completion Occurs This bit is set by hardware and is cleared by writing a “1” into the corresponding interrupt status clear bit in the PDMAISCR register. The Transfer Completion event will occur when the PDMA has completed a data transfer task.
[27], [22], [17], [12], [7], [2]	HTISTAn	Channel n Half Transfer Interrupt Status (n = 0 ~ 5) 0: No Half Transfer Event Occurs 1: Half Transfer Event Occurs This bit is set by hardware and is cleared by writing a “1” into the corresponding interrupt status clear bit in the PDMAISCR register. A Half Transfer event will occur when the PDMA has completed half of the data transfer task.
[26], [21], [16], [11], [6], [1]	BEISTAn	Channel n Block Transaction End Interrupt Status (n = 0 ~ 5) 0: No Block Transaction End Event Occurs 1: Block Transaction End Event Occurs This bit is set by hardware and is cleared by writing a “1” into the corresponding interrupt status clear bit in the PDMAISCR register. A Block Transaction End event will occur when the PDMA completes a data block transaction task.

Bits	Field	Descriptions
[25], [20], [15], [10], [5], [0]	GEISTAn	Channel n Global Transfer Interrupt Status (n = 0 ~ 5) 0: No TE, TC, HT or BE event occurs 1: TE, TC, HT, or BE event occurs This bit is set by hardware and is cleared by writing a "1" into the corresponding interrupt status clear bit, GEICLRn, in the PDMAISCR register. A Global Transfer Event will occur if any of the BE, HT, TC or TE events occur. Also clearing any of the BE, HT, TC or TE event interrupt flags will clear the GE interrupt flag. Note that if a "1" is written into the GEICLRn bit in the PDMAISCR register to clear the GE interrupt flag, the BE, HT, TC and TE event interrupt flags will also be cleared to 0 together with the GE interrupt status flag.

PDMA Interrupt Status Clear Register – PDMAISCR

This register is used to clear the corresponding interrupt status bits in the PDMAISR Register.

Offset: 0x128

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved		TEICLR5	TCICLR5	HTICLR5	BEICLR5	GEICLR5	TEICLR4
Type/Reset			RW 0	RW 0	RW 0	RW 0	RW 0	RW 0
	23	22	21	20	19	18	17	16
	TCICLR4	HTICLR4	BEICLR4	GEICLR4	TEICLR3	TCICLR3	HTICLR3	BEICLR3
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0
	15	14	13	12	11	10	9	8
	GEICLR3	TEICLR2	TCICLR2	HTICLR2	BEICLR2	GEICLR2	TEICLR1	TCICLR1
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0
	7	6	5	4	3	2	1	0
	HTICLR1	BEICLR1	GEICLR1	TEICLR0	TCICLR0	HTICLR0	BEICLR0	GEICLR0
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0

Bits	Field	Descriptions
[29], [24], [19], [14], [9], [4]	TEICLRn	Channel n Transfer Error Interrupt Status Clear (n = 0 ~ 5) 0: No Operation 1: Clear the corresponding TEISTAn bit in the PDMAISR register Writing a "1" into the TEICLRn bit will clear the TEISTAn status bit in the PDMAISR register. This bit will be automatically cleared to 0 after a "1" is written.
[28], [23], [18], [13], [8], [3]	TCICLRn	Channel n Transfer Complete Interrupt Status Clear (n = 0 ~ 5) 0: No Operation 1: Clear the corresponding TCISTAn bit in the PDMAISR register Writing a "1" into the TCICLRn bit will clear the TCISTAn status bit in the PDMAISR register. This bit will be automatically cleared to 0 after a "1" is written.
[27], [22], [17], [12], [7], [2]	HTRICLRn	Channel n Half Transfer Interrupt Status Clear (n = 0 ~ 5) 0: No Operation 1: Clear the corresponding HTISTAn bit in the PDMAISR register Writing a "1" into the HTRICLRn bit will clear the HTISTAn status bit in the PDMAISR register. This bit will be automatically cleared to 0 after a "1" is written.

Bits	Field	Descriptions
[26], [21], [16], [11], [6], [1]	BEICLRn	Channel n Block Transaction End Interrupt Status Clear (n = 0 ~ 5) 0: No Operation 1: Clear the corresponding BEISTAn bit in the PDMAISR register Writing a "1" into the BEICLRn bit will clear the BEISTAn status bit in the PDMAISR register. This bit will automatically be cleared to 0 after a data "1" is written.
[25], [20], [15], [10], [5], [0]	GEICLRn	Channel n Global Transfer Event Interrupt Status Clear (n = 0 ~ 5) 0: No Operation 1: Clear the corresponding TEISTAn, TCISTAn, HTISTAn, BEISTAn and GEISTAn bits in the PDMAISR register Writing a "1" into the GEICLRn bit will clear the GEISTAn status bit together with the TEISTAn, TCISTAn, HTISTAn and BEISTAn bits in the PDMAISR register. This bit will be automatically cleared to 0 after a "1" is written.

PDMA Interrupt Enable Register – PDMAIER

This register is used to enable or disable the related interrupts of the PDMA channel 0 ~ 5.

Offset: 0x130

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved		TEIE5	TCIE5	HTIE5	BEIE5	GEIE5	TEIE4
Type/Reset			RW 0	RW 0	RW 0	RW 0	RW 0	RW 0
	23	22	21	20	19	18	17	16
	TCIE4	HTIE4	BEIE4	GEIE4	TEIE3	TCIE3	HTIE3	BEIE3
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0
	15	14	13	12	11	10	9	8
	GEIE3	TEIE2	TCIE2	HTIE2	BEIE2	GEIE2	TEIE1	TCIE1
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0
	7	6	5	4	3	2	1	0
	HTIE1	BEIE1	GEIE1	TEIE0	TCIE0	HTIE0	BEIE0	GEIE0
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0

Bits	Field	Descriptions
[29], [24], [19], [14], [9], [4]	TEIEn	Channel n Transfer Error Interrupt Enable control (n = 0 ~ 5) 0: Transfer Error interrupt is disabled 1: Transfer Error interrupt is enabled This bit is set and cleared by software.
[28], [23], [18], [13], [8], [3]	TCIEn	Channel n Transfer Complete Interrupt Enable control (n = 0 ~ 5) 0: Transfer Completion interrupt is disabled 1: Transfer Completion interrupt is enabled This bit is set and cleared by software.
[27], [22], [17], [12], [7], [2]	HTIEn	Channel n Half Transfer Interrupt Enable control (n = 0 ~ 5) 0: Half Transfer interrupt is disabled 1: Half Transfer interrupt is enabled This bit is set and cleared by software.
[26], [21], [16], [11], [6], [1]	BEIEn	Channel n Block Transaction End Interrupt Enable control (n = 0 ~ 5) 0: Block Transaction End interrupt is disabled 1: Block Transaction End interrupt is enabled This bit is set and cleared by software.
[25], [20], [15], [10], [5], [0]	GEIEn	Channel n Global Transfer Event Interrupt Enable control (n = 0 ~ 5) 0: Global Transfer Event interrupt is disabled 1: Global Transfer Event interrupt is enabled This bit is set and cleared by software.

25 Inter-IC Sound (I²S)

Introduction

The I²S is a synchronous communication interface that can be used as a master or slave to exchange data with other audio peripherals, such as ADCs or DACs. The I²S supports a variety of data formats. In addition to the stereo I²S-justified, Left-justified and Right-justified modes, there are mono PCM modes with 8 / 16 / 24 / 32-bit sample size. When the I²S operates in the master mode, then when using the fractional divider, it can provide an accurate sampling frequency output and support the rate control function and fine-tuning of the output frequency to avoid system problems caused by the cumulative frequency error between different devices.

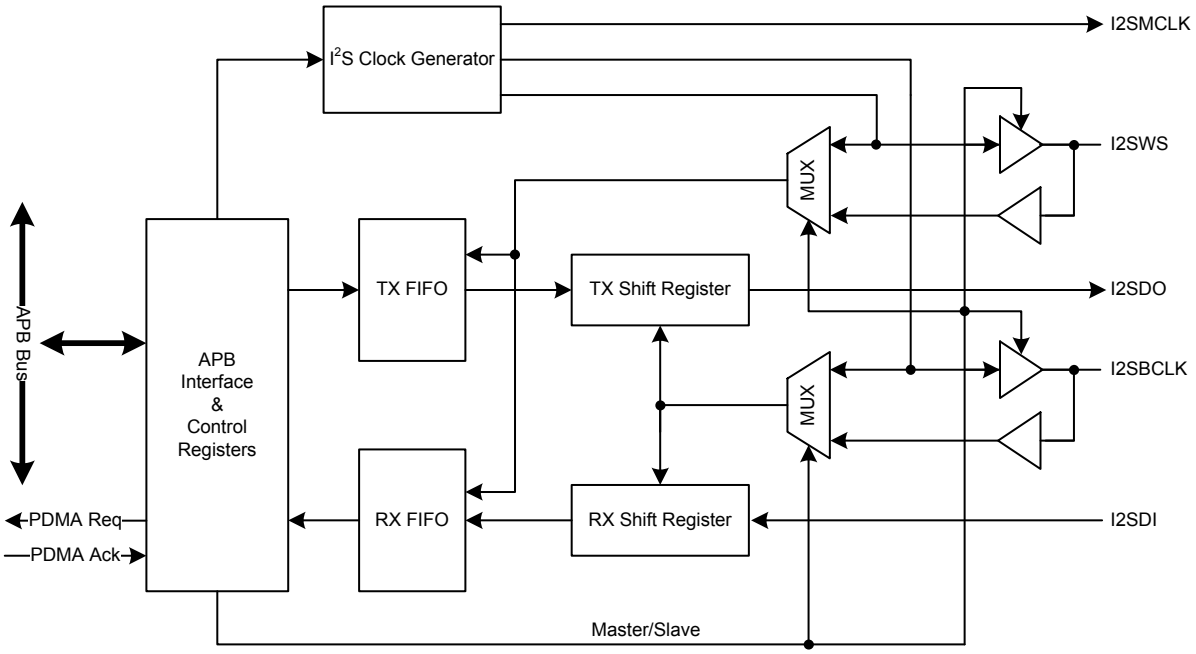


Figure 154. I²S Block Diagram

Features

- Master or slave mode
- Mono and stereo
- I²S-justified, Left-justified, and Right-justified mode
- 8 / 16 / 24 / 32-bit sample size with 32-bit channel extended
- 8 × 32-bit TX & RX FIFO with PDMA supported
- 8-bit Fractional Clock Divider with rate control

Functional Description

I²S Master and Slave Mode

The I²S can operate in slave or master mode. Within the I²S module the difference between these modes lies in the word select (WS) signal which determines the timing of data transmissions.

- In the master mode, the word select signal is generated internally by a clock rate generator.
- In the slave mode, the word select signal is input on the I2S_WS pin.
- When an I²S bus is enabled, the word select, bit clock signals are sent continuously by the bus master.
- The mute control bit will place the transmit channel in a mute condition. When the mute mode is enabled, the transmit channel FIFO operates normally, but the output data stream is discarded and replaced by zeroes. This bit does not affect the receive channel so data reception can occur normally.

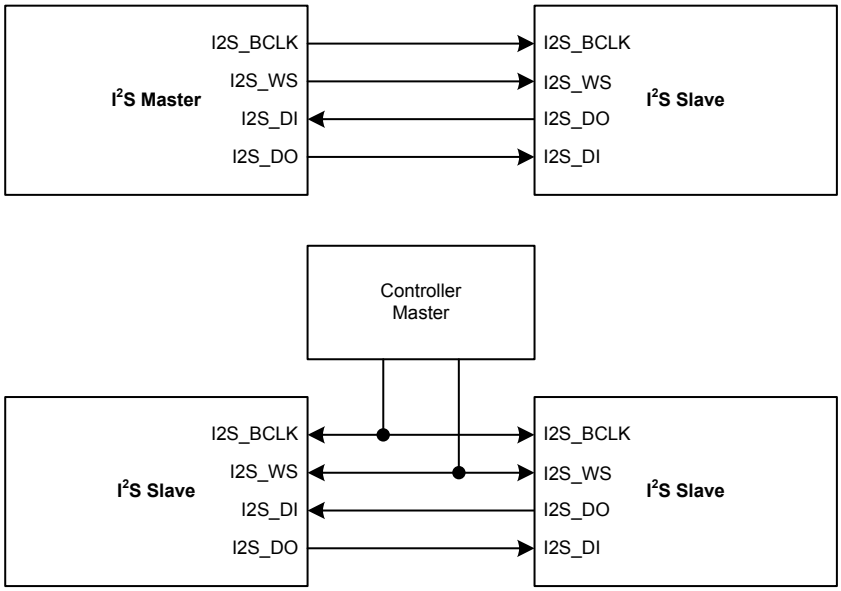


Figure 155. Simple I²S Master / Slave Configuration

I²S Clock Rate Generator

The main (I2S_MCLK) and bit clock (I2S_BCLK) rates for the I²S are determined by the values in the I2SCDR register. The required I²S bit clock rate setting depends on the desired audio sample rate desired, the format (stereo / mono) used and the data size. The main clock rate (I2S_MCLK) is generated using a fractional rate divider which is a divided down PCLK frequency of the I²S. Values of the numerator (X) and the denominator (Y) must be chosen to produce a frequency twice that of the main clock (I2S_MCLK). The output frequency of the divider is divided by 2 in order to get the duty cycle of the output clock more even. The I²S clock generator block diagram is shown in Figure 156. The equation for the fractional rate divider is:

$$I^2S\_MCLK = 1 / 2 \times PCLK \times (X / Y), \text{ and } 2X \leq Y, X \neq 0, Y \neq 0$$

$$I^2S\_BCLK = I^2S\_MCLK / N, N = \{1, 2, ..., 64\}$$

Because the fractional rate divider is a fully digital implementation function, the divider output clock transitions are synchronous with the input source clock. Therefore, the fractional rate divider will generate some jitter with some divider settings. Users should make note of this phenomenon when choosing the X and Y setup values. It is possible to avoid jitter entirely by choosing fractions such that X divides evenly into Y. For example, 2/4, 2/6, 3/9, 1/N, etc.

The tables below show the recommended setup values to reduce clock jitter for different source clocks and sample rates.

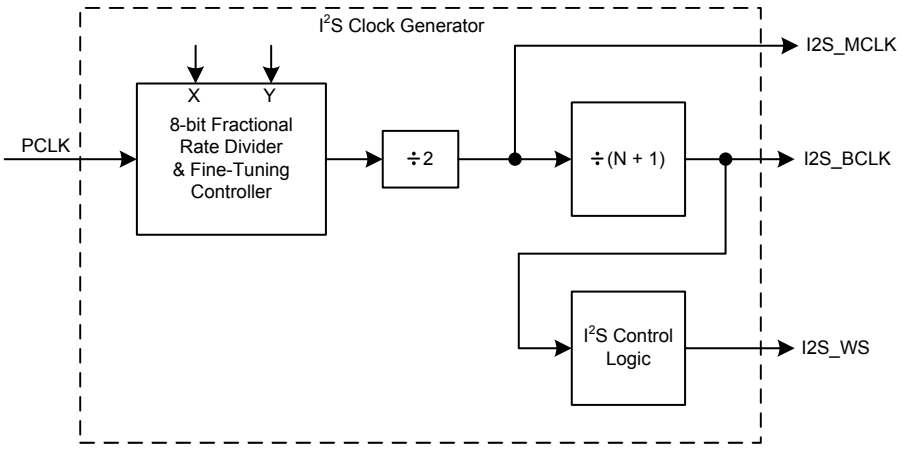


Figure 156. I²S Clock Generator Diagram

Table 67. Recommend F_S List @ 8 MHz PCLK

	512 F_S		384 F_S		256 F_S		192 F_S		128 F_S		64 F_S		32 F_S		16 F_S	
Fs (Hz)	X	Y	X	Y	X	Y	X	Y	X	Y	X	Y	X	Y	X	Y
8,000	—	—	48	125	32	125	24	125	16	125	8	125	4	125	2	125
11,025	—	—	—	—	85	241	59	223	45	255	21	238	9	204	3	136
12,000	—	—	—	—	48	125	36	125	24	125	12	125	6	125	3	125
16,000	—	—	—	—	—	—	48	125	32	125	16	125	8	125	4	125
22,050	—	—	—	—	—	—	—	—	85	241	45	255	21	238	9	204
24,000	—	—	—	—	—	—	—	—	48	125	24	125	12	125	6	125
32,000	—	—	—	—	—	—	—	—	—	—	32	125	16	125	8	125
44,100	—	—	—	—	—	—	—	—	—	—	85	241	45	255	21	238
48,000	—	—	—	—	—	—	—	—	—	—	48	125	24	125	12	125
96,000	—	—	—	—	—	—	—	—	—	—	—	—	48	125	24	125
192,000	—	—	—	—	—	—	—	—	—	—	—	—	—	—	48	125

Table 68. Recommend F_S List @ 48 MHz PCLK

	512 F_S		384 F_S		256 F_S		192 F_S		128 F_S		64 F_S		32 F_S		16 F_S	
Fs (Hz)	X	Y	X	Y	X	Y	X	Y	X	Y	X	Y	X	Y	X	Y
8,000	18	211	8	125	9	211	4	125	5	234	1	94	1	188	—	—
11,025	2	17	3	34	1	17	3	68	1	34	1	68	1	136	—	—
12,000	16	125	12	125	8	125	6	125	4	125	2	125	1	125	1	250
16,000	43	252	16	125	18	211	8	125	9	211	5	234	1	94	1	188
22,050	4	17	3	17	2	17	3	34	1	17	1	34	1	68	1	136
24,000	32	125	24	125	16	125	12	125	8	125	4	125	2	125	1	125
32,000	71	208	32	125	43	252	16	125	18	211	9	211	5	234	1	94
44,100	119	253	85	241	4	17	3	17	2	17	1	17	1	34	1	68
48,000	—	—	48	125	32	125	24	125	16	125	8	125	4	125	2	125
96,000	—	—	—	—	—	—	48	125	32	125	16	125	8	125	4	125
192,000	—	—	—	—	—	—	—	—	—	—	32	125	16	125	8	125

I²S Interface Format

I²S-justified Stereo Mode

The standard I²S-justified mode is where the Most Significant Bit (MSB) of the stereo audio sample data is available on the second rising edge of the BCLK clock following a WS signal transition. In the stereo mode, a low WS state indicates left channel data and a high state indicates right channel data. Figure 157 and Figure 158 show the standard I²S-justified stereo mode format.

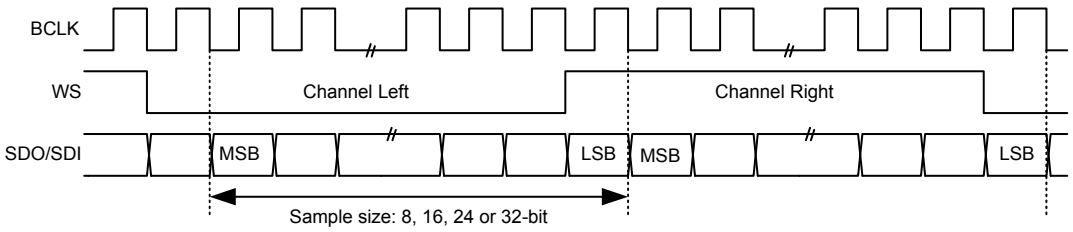


Figure 157. I²S-justified Stereo Mode Waveforms

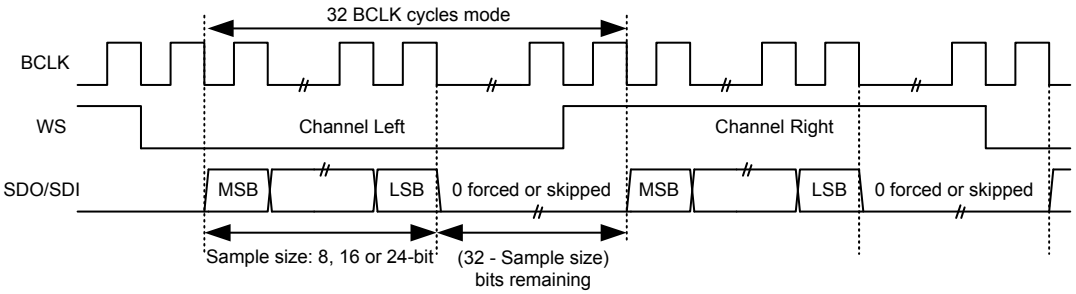


Figure 158. I²S-justified Stereo Mode Waveforms (32-bit Channel Enabled)

Left-justified Stereo Mode

Left-Justified mode is where the Most Significant Bit (MSB) of the stereo audio sample data is available on the first rising edge of BCLK following a WS transition. Figure 159 and Figure 160 are shown with a left I²S-justified stereo mode format.

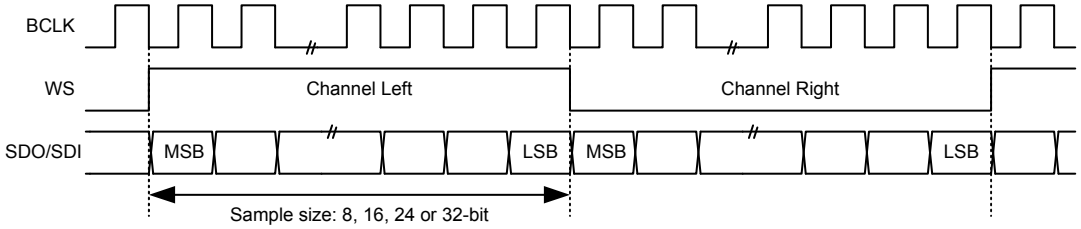


Figure 159. Left-justified Stereo Mode Waveforms

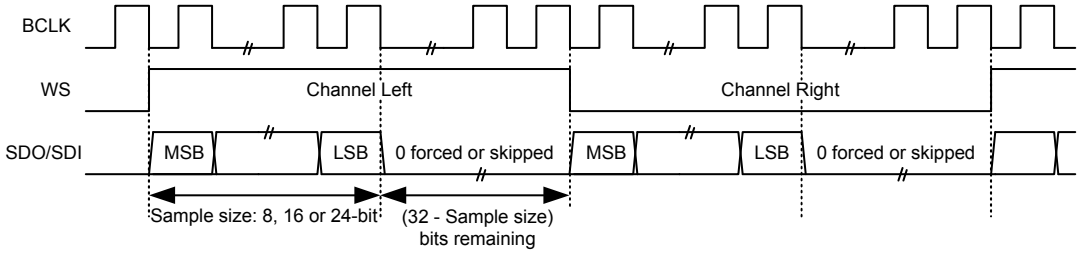


Figure 160. Left-justified Stereo Mode Waveforms (32-bit Channel Enabled)

Right-justified Stereo Mode

Right-Justified mode is where the Least Significant Bit (LSB) of the stereo audio sample data is available on the rising edge of BCLK preceding a WS transition and where the MSB is transmitted first. Figure 161 and Figure 162 show a right I²S-justified stereo mode format.

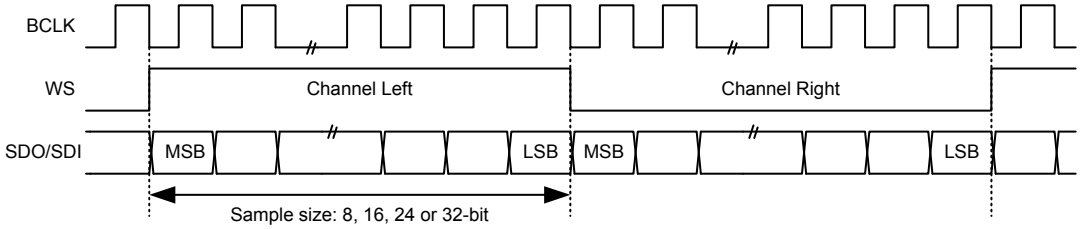


Figure 161. Right-justified Stereo Mode Waveforms

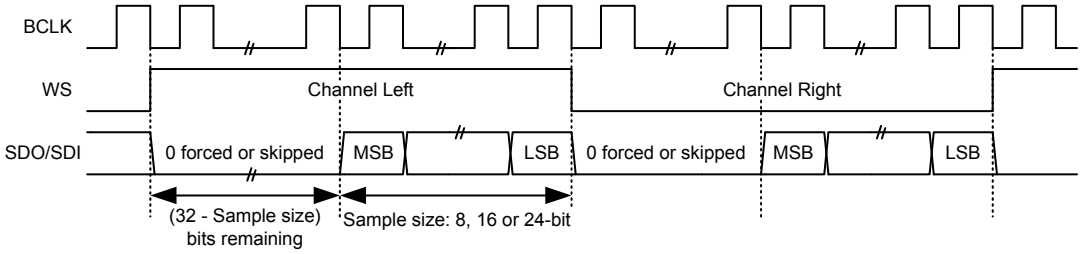


Figure 162. Right-justified Stereo Mode Waveforms (32-bit Channel Enabled)

I²S-justified Mono Mode

In the I²S-justified mono mode, the Most Significant Bit (MSB) of the mono audio sample data is available on the second rising edge of the BCLK clock following a falling edge on the WS signal. Figure 163 and Figure 164 show an I²S-justified mono mode format.

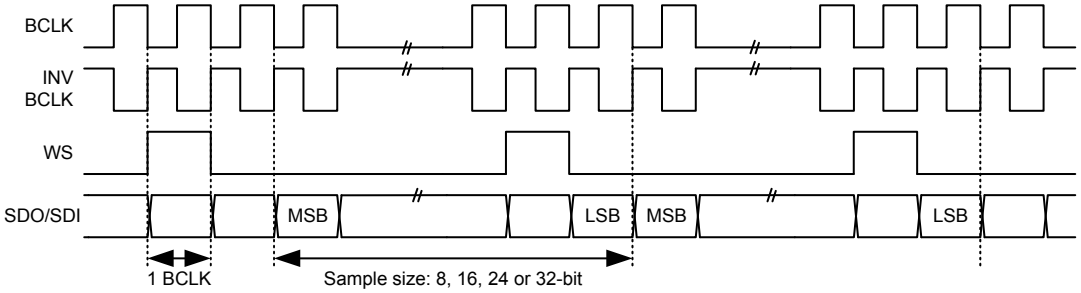


Figure 163. I²S-justified Mono Mode Waveforms

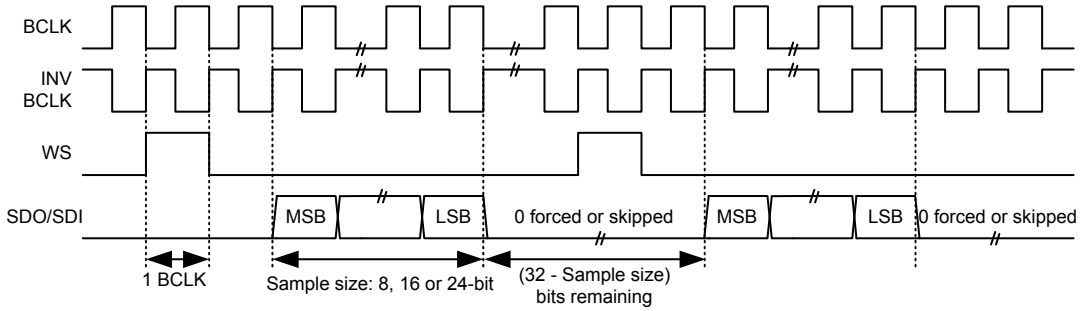


Figure 164. I²S-justified Mono Mode Waveforms (32-bit Channel Enabled)

Left-justified Mono Mode

In the left-justified mono mode, the Most Significant Bit (MSB) of the mono audio sample data is available on the first rising edge of the BCLK clock following a falling edge on the WS signal. Figure 165 and Figure 166 show a left-justified mono mode format.

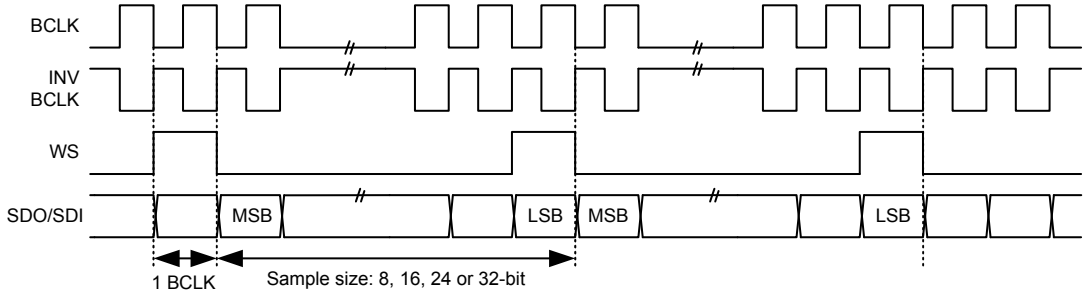


Figure 165. Left-justified Mono Mode Waveforms

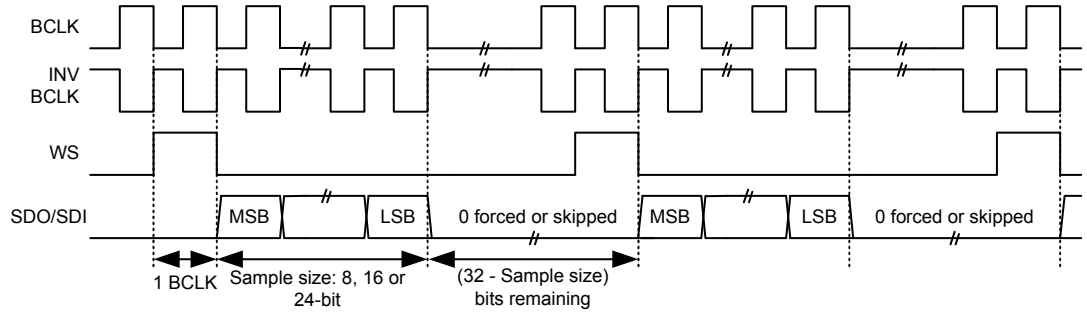


Figure 166. Left-justified Mono Mode Waveforms (32-bit Channel Enabled)

Right-justified Mono Mode

In the right-justified mono mode, the Least Significant Bit (LSB) of the mono audio sample data is available on the last rising edge of the BCLK clock preceding a rising edge on the WS signal. Figure 167 and Figure 168 show the right-justified mono mode format.

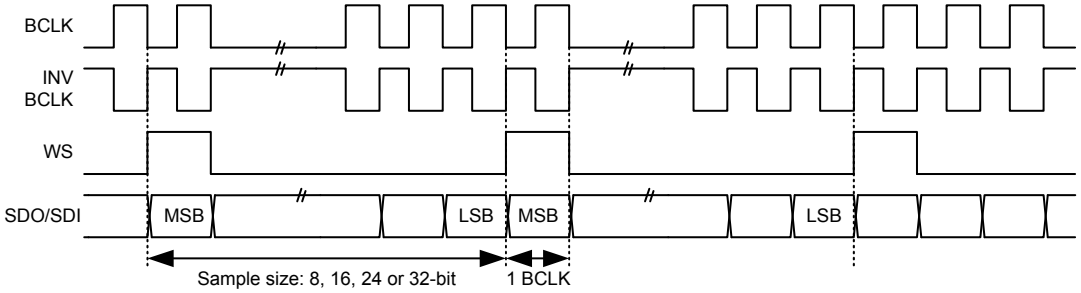


Figure 167. Right-justified Mono Mode Waveforms

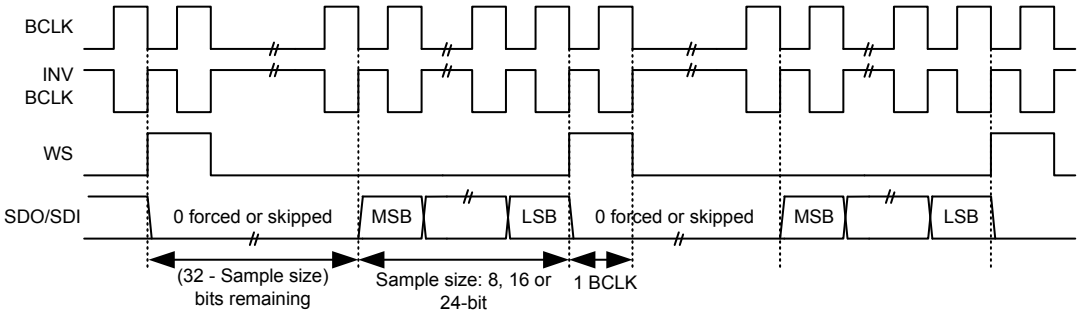


Figure 168. Right-justified Mono Mode Waveforms (32-bit Channel Enabled)

I²S-justified Repeat Mode

In the I²S-justified repeat mode, the Most Significant Bit (MSB) of the mono audio sample data is available on the second rising edge of the BCLK clock following a WS signal transition. In this mode the same data is transmitted twice, once when WS is low and again when WS is high. Figure 169 and Figure 170 show the I²S-justified repeat mode format.

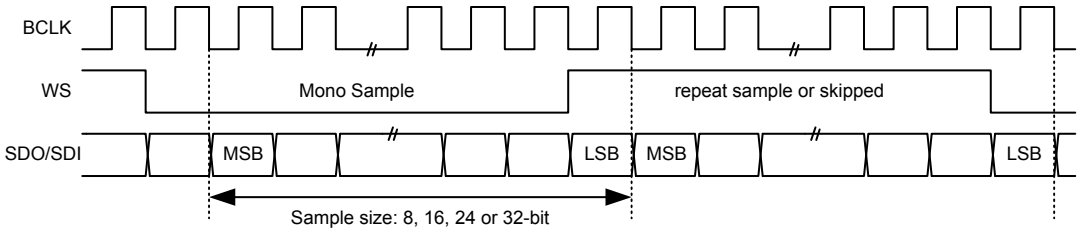


Figure 169. I²S-justified Repeat Mode Waveforms

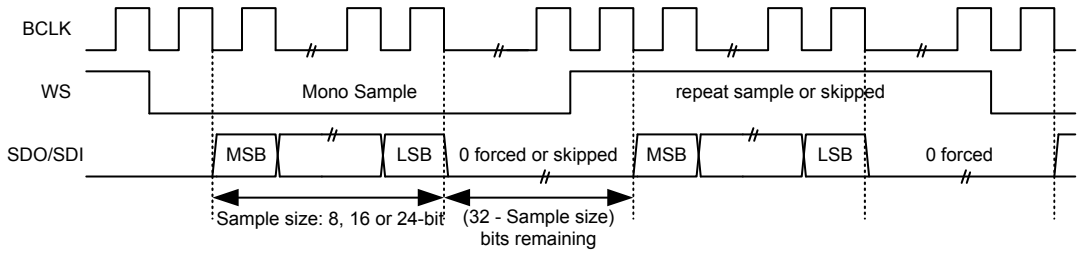


Figure 170. I²S-justified Repeat Mode Waveforms (32-bit Channel Enabled)

FIFO Control and Arrangement

The I²S handles audio data for transmission and reception and is performed via the FIFO controller. Each transmitted or received FIFO has a depth of 8 words (8 × 32-bit) and can buffer the data. The format is dependent upon the stereo / mono mode and sample size setting. The detailed FIFO data content format is shown in Figure 171. The FIFO controller consists of comparators which compare the current FIFO levels with configurable depth settings. The current level of the TX or RX FIFO status can be seen in the TXFS and RXFS fields of the I²S status register (I2SSR).

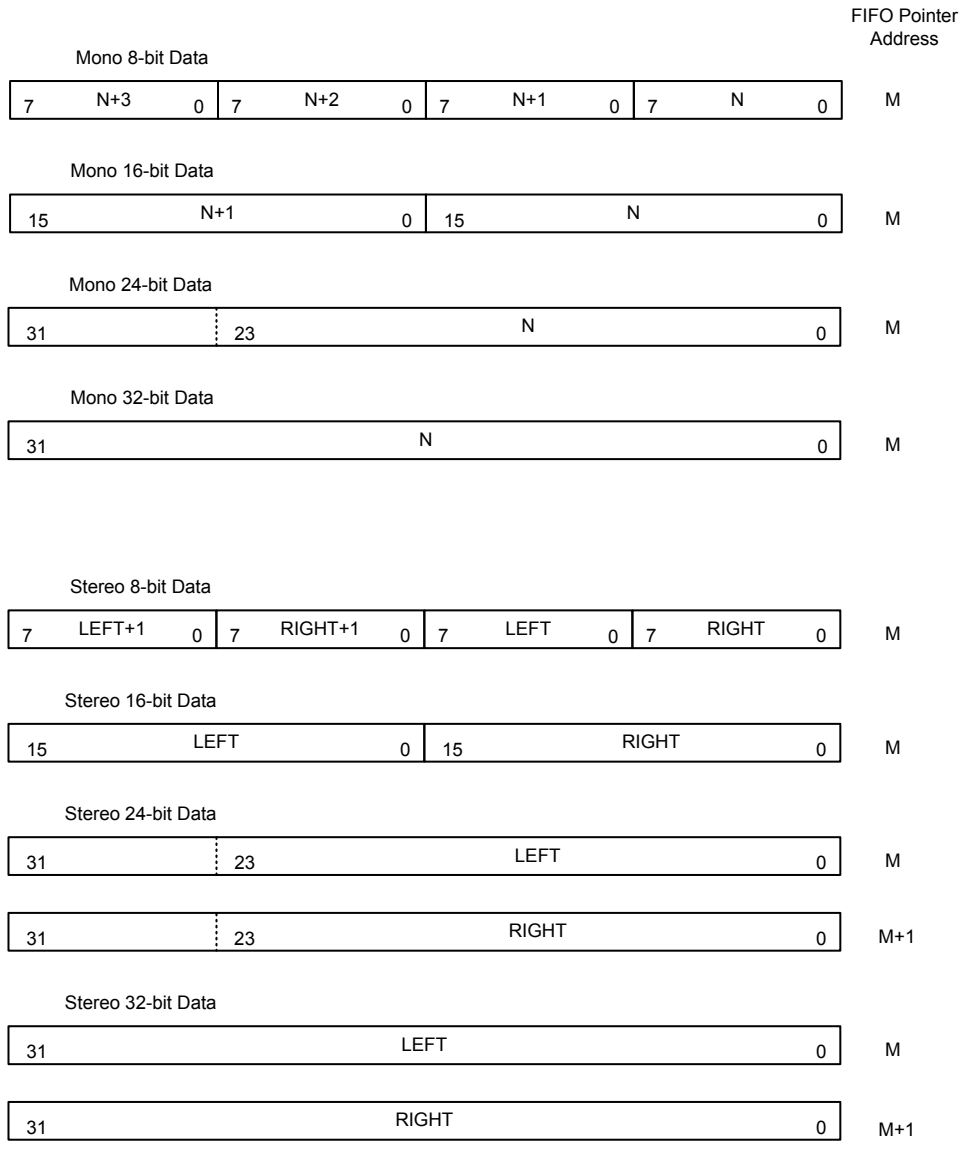


Figure 171. FIFO Data Content Arrangement for Various Modes

PDMA and Interrupt

When the level of received data in the RX FIFO is equal to or greater than the level defined by the RXFTLS field in the I²S FIFO control register (I2SFCR), the relative RXFTL flag will be set and then an I²S RX PDMA request will be generated. A CPU interrupt will be generated if the enable bit of the I²S RX PDMA request or the RX FIFO trigger level interrupt is asserted. When the level of transmitted data in the TX FIFO is equal to or less than the level defined by the TXFTLS field in the I²S FIFO control register (I2SFCR), the relative TXFTL flag will be set and a I²S TX PDMA request will be generated. A CPU interrupt will be generated if the enable bit of the I²S TX PDMA request or TX FIFO trigger level interrupt is asserted.

The I²S transmitter and receiver have separate PDMA requests and can be assigned to two different PDMA channels. When a PDMA request is enabled for the I²S transmitter (TXDMAEN = 1) then this will automatically request that data is transferred to the assigned I²S TX PDMA channel whenever TX FIFO space is available and TXFTL is active. When a PDMA request is enabled for the receiver (RXDMAEN = 1) then this will automatically request the data transfers to the I²S RX PDMA channel whenever data is present in the receive FIFO and when RXFTL is active.

Register Map

The following table shows the I²S registers and reset values.

Table 69. I²S Register Map

Register	Offset	Description	Reset Value
I2SCR	0x000	I²S Control Register	0x0000_0000
I2SIER	0x004	I²S Interrupt Enable Register	0x0000_0000
I2SCDR	0x008	I²S Clock Divider Register	0x0000_0000
I2STXDR	0x00C	I²S TX Data Register	0x0000_0000
I2SRXDR	0x010	I²S RX Data Register	0x0000_0000
I2SFCR	0x014	I²S FIFO Control Register	0x0000_0000
I2SSR	0x018	I²S Status Register	0x0000_0809
I2SRCNTR	0x01C	I²S Rate Counter Value Register	0x0000_0000

Register Descriptions

I²S Control Register – I2SCR

This register specifies the corresponding I²S function enable control.

Offset: 0x000
Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	22	21	20	19	18	17	16
	Reserved				MCKINV	BCKINV	RCSEL	RCEN
Type/Reset					RW 0	RW 0	RW 0	RW 0
	15	14	13	12	11	10	9	8
	CLKDEN	RXDMAEN	TXDMAEN	TXMUTE	CHANNEL	REPEAT	MCLKEN	BITEXT
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0
	7	6	5	4	3	2	1	0
		FORMAT		SMPSIZE	MS	RXEN	TXEN	I2SEN
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0

Bits	Field	Descriptions
[19]	MCKINV	MCLK Inverse Enable 0: Disable 1: Enable
[18]	BCKINV	BCLK Inverse Enable 0: Disable 1: Enable
[17]	RCSEL	Rate Control Select (master only) 0: Slower 1: Faster
[16]	RCEN	Rate Control Enable (master only) 0: Disable 1: Enable
[15]	CLKDEN	Clock Divider Enable (master only) 0: Disable 1: Enable The clock divider can be used to generate the MCLK and BCLK clock of the I²S interface for master mode.
[14]	RXDMAEN	RX PDMA Request Enable 0: Disable 1: Enable
[13]	TXDMAEN	TX PDMA Request Enable 0: Disable 1: Enable
[12]	TXMUTE	TX Mute Enable 0: Disable 1: Enable

Bits	Field	Descriptions
[11]	CHANNEL	Stereo or Mono 0: Stereo 1: Mono Note: This bit should be configured when I²S is disabled.
[10]	REPEAT	Repeat Mode 0: Disable 1: Enable This mode is for I²S-justified stereo configuration only, transmitting the mono data on both channels and receiving just the left channel data and ignoring the right. Enabling the repeat mode will reset the CHANNEL bit automatically. Note: This bit should be configured when the I²S is disabled.
[9]	MCLKEN	MCLK Output Enable (master only) 0: Disable 1: Enable Note: This bit should be configured when the I²S is disabled.
[8]	BITEXT	32-bit Channel Enable 0: Disable 1: Enable Setting this bit will force the channel size to 32-bits. If the sample size is 8 / 16 / 24-bits, the remaining bits will be forced to 0 in the TX and ignored in the RX. Note: This bit should be configured when the I²S is disabled.
[7:6]	FORMAT	Data Format 00: I²S-justified 01: Left-justified 10: Right-justified 11: Reserved Note: This bit should be configured when the I²S is disabled.
[5:4]	SMPSIZE	Sample Size 00: 8-bit 01: 16-bit 10: 24-bit 11: 32-bit Note: This bit should be configured when the I²S is disabled.
[3]	MS	Master or Slave Mode 0: Master 1: Slave Note: This bit should be configured when the I²S is disabled.
[2]	RXEN	RX Enable 0: Disable 1: Enable
[1]	TXEN	TX Enable 0: Disable 1: Enable
[0]	I2SEN	I²S Enable 0: Disable 1: Enable

I²S Interrupt Enable Register – I2SIER

This register contains the corresponding I²S interrupt enable bits.

Offset: 0x004

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	22	21	20	19	18	17	16
	Reserved							
Type/Reset								
	15	14	13	12	11	10	9	8
	Reserved							
Type/Reset								

	7	6	5	4	3	2	1	0
	Reserved	RXOVIEN	RXUDIEN	RXFTLIEN	Reserved	TXOVIEN	TXUDIEN	TXFTLIEN
Type/Reset		RW 0	RW 0	RW 0		RW 0	RW 0	RW 0

Bits	Field	Descriptions
[6]	RXOVIEN	RX FIFO Overflow Interrupt Enable 0: Disable 1: Enable
[5]	RXUDIEN	RX FIFO Underflow Interrupt Enable 0: Disable 1: Enable
[4]	RXFTLIEN	RX FIFO Trigger Level Interrupt Enable 0: Disable 1: Enable
[2]	TXOVIEN	TX FIFO Overflow Interrupt Enable 0: Disable 1: Enable
[1]	TXUDIEN	TX FIFO Underflow Interrupt Enable 0: Disable 1: Enable
[0]	TXFTLIEN	TX FIFO Trigger Level Interrupt Enable 0: Disable 1: Enable

I²S Clock Divider Register – I2SCDR

This register specifics the I²S clock divider ratio.

Offset: 0x008

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	22	21	20	19	18	17	16
	N_DIV							
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0
	15	14	13	12	11	10	9	8
	X_DIV							
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0
	7	6	5	4	3	2	1	0
	Y_DIV							
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0

Bits	Field	Descriptions
[23:16]	N_DIV	N divider for BCLK Note: This bit should be configured when the I²S is disabled.
[15:8]	X_DIV	X divider for MCLK (X ! = 0) && (Y ! = 0) && (X / Y ≥ 1 / 255) && (X / Y ≤ 127 / 255) Note: This bit should be configured when the I²S is disabled.
[7:0]	Y_DIV	Y divider for MCLK (X ! = 0) && (Y ! = 0) && (X / Y ≥ 1 / 255) && (X / Y ≤ 127 / 255) Note: This bit should be configured when the I²S is disabled.

I²S TX Data Register – I2STXDR

This register is used to specify the I²S transmitted data.

Offset: 0x00C

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	TXDR							
Type/Reset	WO 0	WO 0	WO 0	WO 0	WO 0	WO 0	WO 0	WO 0
	23	22	21	20	19	18	17	16
	TXDR							
Type/Reset	WO 0	WO 0	WO 0	WO 0	WO 0	WO 0	WO 0	WO 0
	15	14	13	12	11	10	9	8
	TXDR							
Type/Reset	WO 0	WO 0	WO 0	WO 0	WO 0	WO 0	WO 0	WO 0
	7	6	5	4	3	2	1	0
	TXDR							
Type/Reset	WO 0	WO 0	WO 0	WO 0	WO 0	WO 0	WO 0	WO 0

Bits	Field	Descriptions
[31:0]	TXDR	TX Data Register

I²S RX Data Register – I2SRXDR

This register is used to store the I²S received data.

Offset: 0x010

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	RXDR							
Type/Reset	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0
	23	22	21	20	19	18	17	16
	RXDR							
Type/Reset	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0
	15	14	13	12	11	10	9	8
	RXDR							
Type/Reset	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0
	7	6	5	4	3	2	1	0
	RXDR							
Type/Reset	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0

Bits	Field	Descriptions
[31:0]	RXDR	RX Data Register

I²S FIFO Control Register – I2SFCR

This register contains the related I²S FIFO control bits.

Offset: 0x014

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	22	21	20	19	18	17	16
	Reserved							
Type/Reset								
	15	14	13	12	11	10	9	8
	Reserved						RXFRST	TXFRST
Type/Reset							RW 0	RW 0
	7	6	5	4	3	2	1	0
	RXFTLS				TXFTLS			
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0

Bits	Field	Descriptions
[9]	RXFRST	RX FIFO Reset Set this bit to reset the RX FIFO.
[8]	TXFRST	TX FIFO Reset Set this bit to reset the TX FIFO.
[7:4]	RXFTLS	RX FIFO Trigger Level Select 0000: Trigger level is 0 0001: Trigger level is 1 ... 0111: Trigger level is 7 1xxx: Trigger level is 8 When the data contained in the RX FIFO is equal to or greater than the level defined by the RXFTLS field, the RXFTL flag will be set.
[3:0]	TXFTLS	TX FIFO Trigger Level Select 0000: Trigger level is 0 0001: Trigger level is 1 ... 0111: Trigger level is 7 1xxx: Trigger level is 8 When the data contained in the TX FIFO is equal to or less than the level defined by the TXFTLS field, the TXFTL flag will be set.

I²S Status Register – I2SSR

This register contains the relevant I²S status.

Offset: 0x018

Reset value: 0x0000_0809

	31	30	29	28	27	26	25	24
	RXFS				TXFS			
Type/Reset	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0
	23	22	21	20	19	18	17	16
	Reserved					CLKRDY	TXBUSY	CHS
Type/Reset						RO 0	RO 0	RO 0
	15	14	13	12	11	10	9	8
	Reserved			RXFFUL	RXFEMT	RXFOV	RXFUD	RXFTL
Type/Reset				RO 0	RO 1	WC 0	WC 0	WC 0
	7	6	5	4	3	2	1	0
	Reserved			TXFFUL	TXFEMT	TXFOV	TXFUD	TXFTL
Type/Reset				RO 0	RO 1	WC 0	WC 0	WC 1

Bits	Field	Descriptions
[31:28]	RXFS	RX FIFO Status 0000: RX FIFO empty 0001: RX FIFO contains 1 data ... 1000: RX FIFO contains 8 data Others: Reserved
[27:24]	TXFS	TX FIFO Status 0000: TX FIFO empty 0001: TX FIFO contains 1 data … 1000: TX FIFO contains 8 data Others: Reserved
[18]	CLKRDY	Clock Divider Output Ready Flag 0: Not ready 1: Ready
[17]	TXBUSY	TX Busy Flag 0: Not busy 1: Busy
[16]	CHS	Channel Status 0: Left channel 1: Right channel
[12]	RXFFUL	RX FIFO Full Flag 0: RX FIFO not full 1: RX FIFO full
[11]	RXFEMT	RX FIFO Empty Flag 0: RX FIFO not empty 1: RX FIFO empty

Bits	Field	Descriptions
[10]	RXFOV	RX FIFO Overflow Flag 0: RX FIFO not overflow 1: RX FIFO overflow This bit is set by hardware and cleared by writing 1.
[9]	RXFUD	RX FIFO Underflow Flag 0: RX FIFO not underflow 1: RX FIFO underflow This bit is set by hardware and cleared by writing 1.
[8]	RXFTL	RX FIFO Trigger Level Flag 0: Data in the RX FIFO is less than the trigger level 1: Data in the RX FIFO is equal to or higher than the trigger level This bit is set by hardware and cleared by writing 1.
[4]	TXFFUL	TX FIFO Full Flag 0: TX FIFO not full 1: TX FIFO full
[3]	TXFEMT	TX FIFO Empty Flag 0: TX FIFO not empty 1: TX FIFO empty
[2]	TXFOV	TX FIFO Overflow Flag 0: TX FIFO not overflow 1: TX FIFO overflow This bit is set by hardware and cleared by writing 1.
[1]	TXFUD	TX FIFO Underflow Flag 0: TX FIFO not underflow 1: TX FIFO underflow This bit is set by hardware and cleared by writing 1.
[0]	TXFTL	TX FIFO Trigger Level Flag 0: Data in the TX FIFO is higher than the trigger level 1: Data in the TX FIFO is equal to or less than the trigger level This bit is set by hardware and cleared by writing 1.

I²S Rate Counter Value Register – I2SRCNTR

This register specifics the I²S rate control counter value.

Offset: 0x01C

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	22	21	20	19	18	17	16
	Reserved				RCNTR			
Type/Reset					RW 0	RW 0	RW 0	RW 0
	15	14	13	12	11	10	9	8
	RCNTR							
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0
	7	6	5	4	3	2	1	0
	RCNTR							
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0

Bits	Field	Descriptions
[19:0]	RCNTR	Rate Counter Value This value must be equal to or higher than Y_DIV for useful rate fine-tuning control.

26 Music Synthesis Engine (MSE) – MIDI Engine

Introduction

The Music Synthesis Engine, MIDI Engine, is a synthesizer which integrates a wavetable synthesis function. It can operate up to 32 channels of wavetable synthesis at a single time and provide a stereo output. When the internal control registers are written to, the wavetable synthesizer will automatically output the dedicated PCM code using its hardware synthesis engine.

The data on BL3 ~ BL0 and FR11 ~ FR0 are used to define the output speed of the PCM file, i.e. it can be used to generate the tone scale. When FR11 ~ FR0 is 0x800 and BL3 ~ BL0 is 0x6, each sampled data of the PCM code will be sent out sequentially. The MIDI Engine accepts three waveform formats – 8, 12 and 16 bits. The WBS[1:0] bits are used to define the sample format of each PCM code.

The repeat number is used to define the address which is the repeat point of the sample, RE14 ~ RE0 is used to calculate the repeat address of the PCM code. The process for setting RE14 ~ RE0 is to write the 2's complement of the repeat length to RE14 ~ RE0, with the highest carry ignored. The MIDI Engine will get the repeat address by adding the RE14 ~ RE0 to the address of the end code, then jump to the address to repeat this range.

The device provides independent volume control, the stereo output volume is controlled by VL9 ~ VL0 and VR9 ~ VR0 respectively. The MIDI Engine provides 1024 levels of controllable volume where 0x000 is the maximum and 0x3FF is the minimum output volume.

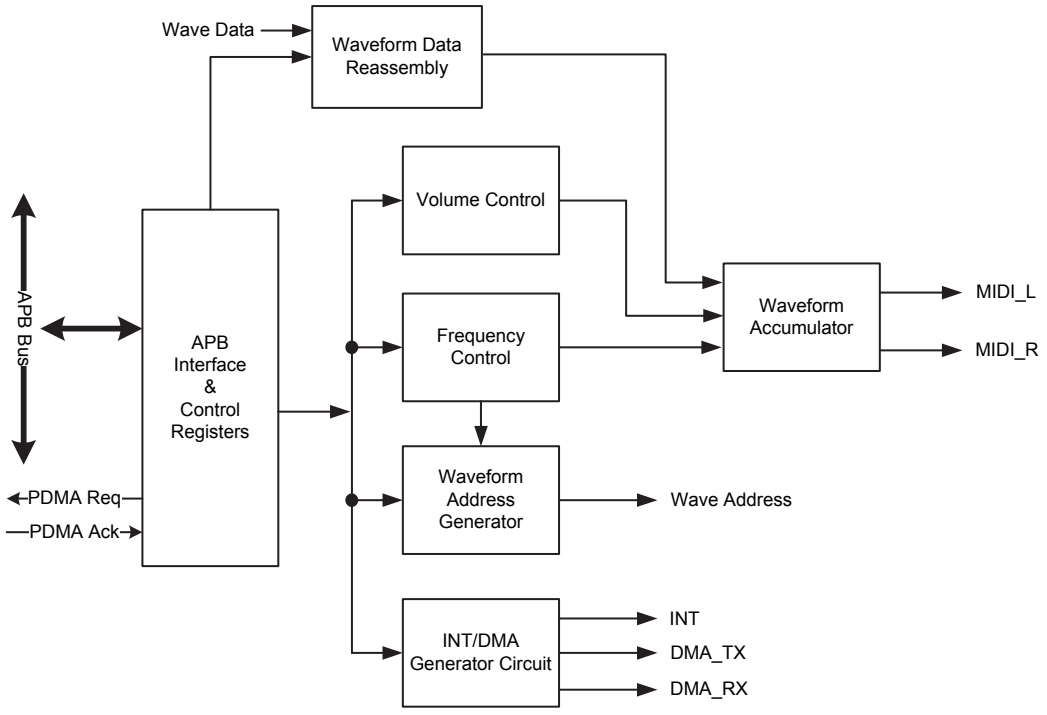


Figure 172. MIDI Engine Block Diagram

Features

- Up to 32 simultaneous sounds
- 10-bit Volume Control
- Output sampling frequency of up to 50 kHz
- Waveform data lengths of 8, 12 or 16 bits
- Stereo output
- Supports Repeat Loop Play
- Supports PDMA

Functional Descriptions

Change Parameter Selection

The channel-related registers, including the CHAN register, FREQ register, VOL register, ST_ADDR register, RE_NUM register and END_ADDR register, are used to determine the parameters in which channel will be changed.

The corresponding register parameters are as follows:

FREQ register: BL[3:0], FR[11:0]

VOL register: A_R, ENV[1:0], VL[9:0], VR[9:0]

ST_ADDR register: ST[18:0]

RE_NUM register: WBS[1:0], RE[14:0]

END_ADDR register: EA[23:0]

The data written into the channel-related registers will only be stored in the register and not be simultaneously updated to the channel immediately. The actual update operation is implemented using the CHAN register, where the CH[4:0] bit field is used to determine which channel will be updated. In this register the ST, VM and FR bits are used to determine which parameters are to be changed with the corresponding register value being changed when the relevant bit is high. For example, the FREQ register will be updated when the FR bit is high and the VOL register will be updated when the VM bit is high while the ST_ADDR, RE_NUM and END_ADDR registers will be updated when the ST bit is high.

The channel-related parameters will be updated to the specified channel only when writing to the CHAN register. For example, channel 1 will be updated if the CH[4:0] bit field value is 1, channel 2 will be updated if the CH[4:0] bit field value is 2 and so on. Which parameters are updated is determined by the ST, VM and FR bits. After all the parameters are properly configured, the CH[4:0] field should be setup to determine which channel will be updated. For these reasons the CHAN register is always the last register to be written to.

Output Frequency

The BL[3:0] bit field is used to control the octave and the FR[11:0] bit field determines the tone.

The sound recorded in the ROM will be scanned out point by point if the {BL[3:0], FR[11:0]} bit field value is 0x6800, which is the so-called the original sound playback.

Taking 0x6800 as an example:

BL = 0x6 is defined as the standard octave. Decrementing the BL bit field value will create a lower octave (frequency divided by 2) while incrementing the BL bit field value will create a higher octave (frequency multiplied by 2).

The BL bit field value ranges from 0x0 to 0xB, but note that values from 0xC to 0xF should be avoided as these values will stop the waveform from moving forward.

The FR bit field value is 0x800. The tone will increase in scale when the FR bit field value is incremented and vice versa. For example, suppose that a sound of 800H is the central C of the piano, then the value of $0x800 \times \sqrt[12]{2} = 0x800 \times 1.059 = 0x879$ will make a sound which is a semitone higher than central C. Selecting greater FR bit field values will result in better tone accuracies, as the value ratio is closer to the standard musical scale ratio.

Waveform Start Address

The ST[18:0] bit field is used to define the waveform start address. The waveform address space in the device contains 24 lines, namely WA[23:0], where WA stands for Waveform Address. The maximum accessible memory is 16 M×16-bit.

For an 8-bit waveform, WA[23:0] = (ST[18:0] << 5), the 8-bit waveform must be placed at an address which is divisible by 0x20 to get the correct waveform start address.

For a 12-bit waveform, WA[23:0] = (ST[18:0] << 5) × (3/2), the 12-bit waveform must be placed at an address which is divisible by 0x30 to get the correct waveform start address.

For a 16-bit waveform, WA[23:0] = (ST[18:0] << 5) × 2, the 16-bit waveform must be placed at an address which is divisible by 0x40 to get the correct waveform start address.

The ST[18:0] field value calculation is shown below:

For an 8-bit waveform, divide the waveform start point address by 0x20, the result is the Start Address, e.g. for a waveform start point address = 0x20, then ST = (0x20 / 0x20) = 1.

For a 12-bit waveform, divide the waveform start point address by 0x30, the result is the Start Address, e.g. for a waveform start point address = 0x30, then ST = (0x30 / 0x30) = 1.

For a 16-bit waveform, divide the waveform start point address by 40h, the result is the Start Address, e.g. for a waveform start point address = 0x40, then ST = (0x40 / 0x40) = 1.

The physical circuit will automatically perform the relevant calculation:

Examples:

For an 8-bit waveform, if ST[18:0] = 1, the circuit will implement WA = (1 << 5), namely the address WA = 0x20, to get the first waveform point.

For a 12-bit waveform, if ST[18:0] = 1, the circuit will implement WA = (1 << 5) × (3/2), namely the address WA = 0x30, to get the first waveform point.

For a 16-bit waveform, if ST[18:0] = 1, the circuit will implement WA = (1 << 5) × (2), namely the address WA = 0x40, to get the first waveform point.

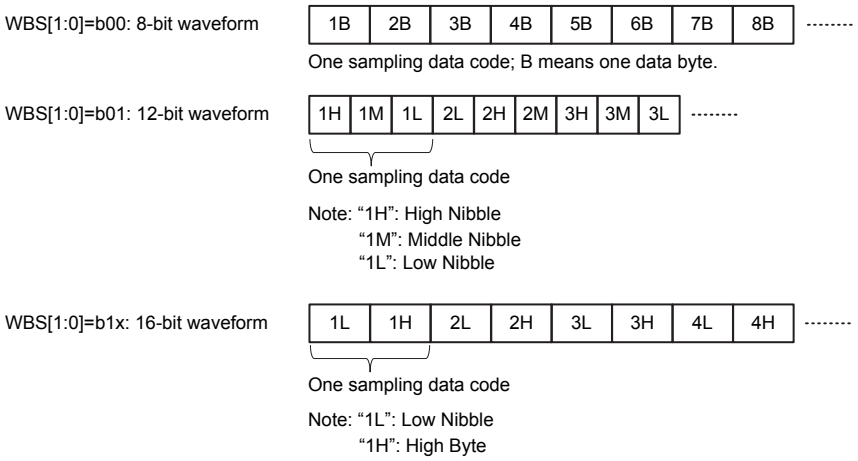
Waveform Format Definition

The WBS[1:0] bit field is used to determine the waveform storage format: 16-bit, 12-bit or 8-bit.

WBS = b1x: 16-bit waveform.

WBS = b01: 12-bit waveform.

WBS = b00: 8-bit waveform.



Waveform Repeat Number Definition

The RE[14:0] bit field is used to define the waveform repeat address.

The RE field value is the 2’complement of the desired total loop point number. For example, if the desired loop point number is 128, which gives a 2’complement value of 0x7F80, then RE[14:0] should be assigned a value of 0x7F80.

Instead of placing a return code in the waveform content, the device judges the address for waveform repeat implementation. However, the first waveform point value of the desired loop area should be placed at the end of the waveform. In using this structure, the waveform can join at any point.

When the waveform moves to the return address, the waveform address will be added to the RE[14:0] field value. As the RE field value is negative, the waveform will roll back after which the waveform moves forward again. Therefore a circulation of the waveform in a specific area can be implemented.

Waveform End Address

The EA[23:0] bit field defines the waveform end address, which is used for waveform circulation. There are 24 lines for waveform address space in the device, namely WA[23:0], where WA stands for Waveform Address.

For examples:

The 8-bit waveform will roll back at the address WA[23:0] = (EA[23:0]).

The 12-bit waveform will roll back at the address WA[23:0] = (EA[23:0]) × (3/2).

The 16-bit waveform will roll back at the address WA[23:0] = (EA[23:0]) × 2.

The EA[23:0] field value calculation is shown as follows:

For an 8-bit waveform, the EA field value is the waveform end point address, e.g. if the waveform end point address = 0x20, then EA[23:0] = 0x20

For a 12-bit waveform, divide the waveform end point address by 3/2, the result is the EA field value, e.g. if the waveform end point address = 0x30, then EA[23:0] = (0x30 / 3×2) = 0x20

For a 16-bit waveform, divide the waveform end point address by 2, the result is the EA field value, e.g. if the waveform end point address = 0x40h, then EA[23:0] = (0x40/2) = 0x20

The physical circuit will calculate the actual return address automatically.

Examples:

For an 8-bit waveform, if EA[23:0] = 0x20, the circuit will perform a waveform roll back at the address WA = 0x20 × 1 = 0x20

For a 12-bit waveform, if EA[23:0] = 0x20, the circuit will perform a waveform roll back at the address WA = 0x20 × 3/2 = 0x30

For a 16-bit waveform, if EA[23:0] = 0x20, the circuit will perform a waveform roll back at the address WA = 0x20 × 2 = 0x40

More detailed information about the ST, RE and EA field is described in the following examples.

Term Descriptions

Logic Address: This is the address that is used for calculation in the circuit, it makes no distinction between 8, 12 or 16 bit waveforms.

Physical Address: This is the actual external address, which will be changed in scale according to the 8, 12 or 16-bit waveforms. For a 12-bit waveform the address is 1.5 times while for 16-bit waveform the address is 2 times.

ST, RE and EA Example Description – 8-bit Waveform

Table 70. 8-bit Music Code

Logic Address – Hex	100	101	102	103	104	105	106	107	108
Physical Address – Hex	100	101	102	103	104	105	106	107	108
Code – Hex	12	34	56	78	9A	BC	DE	F0	9A

Suppose that the physical address 0x104 ~ 0x107 is the desired loop area, ST = (0x100 / 0x20) = 0x8, the desired loop point is 4, whose 2'complement value is 0x7FFC, therefore RE = 0x7FFC, EA = 0x108. Note that the EA address must be filled with the identical code with the Loop start point. This means that code in 0x108 and 0x104 must be the same.

ST, RE and EA Example Description – 16-bit Waveform

Table 71. 16-bit Music Code

Logic Address – Hex	100	101	102	103	104	105	106	107	108
Physical Address – Hex	200	202	204	206	208	20A	20C	20E	210
Code – Hex	1234	5678	9ABC	DEF0	2345	6789	ABCD	EF01	2345

Suppose that the physical address 0x208 ~ 0x20E is the desired loop area, ST = (0x200 / 0x40) = 0x8, the desired loop point is 4, whose 2'complement value is 0x7FFC, therefore RE = 0x7FFC, EA = 0x210 / 2 = 0x108. Note that the EA address must be filled with the identical code with the Loop start point. This means that code in the 0x210 and 0x208 must be the same.

Volume Control

Volume control is implement using the 10-bit left channel volume control bits VL[9:0] and the 10-bit right channel volume control bits VR[9:0]. Here the field value is inversely proportional to the volume.

Taking VL as an example, however VR operates in the same way.

The volume calculation equation is shown as follows:

$$\text{Volume} = 20\log\left[\frac{128 - \text{VL}[5:0]}{128} \times \frac{1}{2^{\text{VL}[9:6]}}\right] \text{dB}$$

For example, if VL[5:0] = b000001, VL[9:6] = b0000, then the result ≈ −0.068 dB and so on.

A_R and ENV[1:0]

In the 16-channel mode, XTAL = 12 MHz, T160 = 5.86 kHz = 0.17 ms; T320 = 2.93 kHz = 0.34 ms; T640 = 1.46 kHz = 0.68 ms.

A_R = 0 or 1, ENV[1:0] = b11, the VM[2:0] is determined by programs and will not be incremented or decremented automatically, where the VM[2:0] is the 3-bit LSB of the VL or VR, namely VL[2:0] or VR[2:0].

Clear A_R to 0, which means Release, and set ENV[1:0] to "b10". VM[2:0] will be incremented by one at every time T640.

If the VM[2:0] initial value is "0", the value will stop at "1" when it is increased to "1", the sequence is "0111...";

If the VM[2:0] initial value is "1", the value will stop at "1", the sequence is "1111...";

If the VM[2:0] initial value is “2”, the value will stop at “3” when it is increased to “3”, the sequence is “2333...”;

If the VM[2:0] initial value is “3”, the value will stop at “3”, the sequence is “3333...”;

If the VM[2:0] initial value is “4”, the value will stop at “5” when it is increased to “5”, the sequence is “4555...”;

If the VM[2:0] initial value is “5”, the value will stop at “5”, the sequence is “5555...”;

If the VM[2:0] initial value is “6”, the value will stop at “7” when it is increased to “7”, the sequence is “6777...”;

If the VM[2:0] initial value is “7”, the value will stop at “7”, the sequence is “7777...”.

Clear A_R to 0, which means Release, and set ENV[1:0] to “b01”. VM[2:0] will be incremented by one at every time T320.

If the VM[2:0] initial value is “0”, the value will stop at “3” when it is increased to “3”, the sequence is “012333...”;

If the VM[2:0] initial value is “1”, the value will stop at “3” when it is increased to “3”, the sequence is “123333...”;

If the VM[2:0] initial value is “2”, the value will stop at “3” when it is increased to “3”, the sequence is “233333...”;

If the VM[2:0] initial value is “3”, the value will stop at “3”, the sequence is “333333...”;

If the VM[2:0] initial value is “4”, the value will stop at “7” when it is increased to “7”, the sequence is “456777...”;

If the VM[2:0] initial value is “5”, the value will stop at “7” when it is increased to “7”, the sequence is “567777...”;

If the VM[2:0] initial value is “6”, the value will stop at “7” when it is increased to “7”, the sequence is “677777...”;

If the VM[2:0] initial value is “7”, the value will stop at “7”, the sequence is “777777...”.

Clear A_R to 0, which means Release, and set ENV[1:0] to “b00”. VM[2:0] will be incremented by one at every time T160.

If the VM[2:0] initial value is “0”, the value will stop at “7” when it is increased to “7”, the sequence is “0123456777...”;

If the VM[2:0] initial value is “1”, the value will stop at “7” when it is increased to “7”, the sequence is “1234567777...”;

If the VM[2:0] initial value is “2”, the value will stop at “7” when it is increased to “7”, the sequence is “2345677777...”;

If the VM[2:0] initial value is “3”, the value will stop at “7” when it is increased to “7”, the sequence is “3456777777...”;

If the VM[2:0] initial value is “4”, the value will stop at “7” when it is increased to “7”, the sequence is “4567777777...”;

If the VM[2:0] initial value is “5”, the value will stop at “7” when it is increased to “7”, the sequence is “5677777777...”;

If the VM[2:0] initial value is “6”, the value will stop at “7” when it is increased to “7”, the sequence is “6777777777...”;

If the VM[2:0] initial value is “7”, the value will stop at “7”, the sequence is “7777777777...”.

Set A_R to 1, which means Attack, and set ENV[1:0] to “b10”. VM[2:0] will be decremented by one at every time T640.

If the VM[2:0] initial value is “7”, the value will stop at “6” when it is reduced to “6”, the sequence is “7666...”;

If the VM[2:0] initial value is “6”, the value will stop at “6”, the sequence is “6666...”;

If the VM[2:0] initial value is “5”, the value will stop at “4” when it is reduced to ”4”, the sequence is “5444...”;

If the VM[2:0] initial value is “4”, the value will stop at “4”, the sequence is “4444...”;

If the VM[2:0] initial value is “3”, the value will stop at “2” when it is reduced to “2”, the sequence is “3222...”;

If the VM[2:0] initial value is “2”, the value will stop at “2”, the sequence is “2222...”;

If the VM[2:0] initial value is “1”, the value will stop at “0” when it is reduced to “0”, the sequence is “1000...”.

If the VM[2:0] initial value is “0”, the value will stop at “0”, the sequence is “0000...”.

Set A_R to 1, which means Attack, and set ENV[1:0] to “b01”. VM[2:0] will be decremented by one at every time T320.

If the VM[2:0] initial value is “7”, the value will stop at “4” when it is reduced to “4”, the sequence is “765444...”;

If the VM[2:0] initial value is “6”, the value will stop at “4” when it is reduced to “4”, the sequence is “654444...”;

If the VM[2:0] initial value is “5”, the value will stop at “4” when it is reduced to “4”, the sequence is “544444...”;

If the VM[2:0] initial value is “4”, the value will stop at “4”, the sequence is “444444...”;

If the VM[2:0] initial value is “3”, the value will stop at “0” when it is reduced to “0”, the sequence is “321000...”;

If the VM[2:0] initial value is “2”, the value will stop at “0” when it is reduced to “0”, the sequence is “210000...”;

If the VM[2:0] initial value is “1”, the value will stop at “0” when it is reduced to “0”, the sequence is “100000...”.

If the VM[2:0] initial value is “0”, the value will stop at “0”, the sequence is “000000...”.

Set A_R to 1, which means Attack, and set ENV[1:0] to “b00”. VM[2:0] will be decremented by one at every time T160.

If the VM[2:0] initial value is “7”, the value will stop at “0” when it is reduced to “0”, the sequence is “7654321000...”;

If the VM[2:0] initial value is “6”, the value will stop at “0” when it is reduced to “0”, the sequence is “6543210000...”;

If the VM[2:0] initial value is “5”, the value will stop at “0” when it is reduced to “0”, the sequence is “5432100000...”;

If the VM[2:0] initial value is “4”, the value will stop at “0” when it is reduced to “0”, the sequence is “4321000000...”;

If the VM[2:0] initial value is “3”, the value will stop at “0” when it is reduced to “0”, the sequence is “3210000000...”;

If the VM[2:0] initial value is “2”, the value will stop at “0” when it is reduced to “0”, the sequence is “2100000000...”;

If the VM[2:0] initial value is “1”, the value will stop at “0” when it is reduced to “0”, the sequence is “1000000000...”.

If the VM[2:0] initial value is “0”, the value will stop at “0”, the sequence is “0000000000...”.

The following example shows how to control the A_R and ENV[1:0]:

These 3 bits is used to process the volume join problem, note that this control function is not necessary if the VM[2:0] is incremented or decremented by 1 every time. The ENV bit values of “b10”, ”b01” and “b00” are used when the volume change becomes more and more sharp. The specific control method is shown as below:

If the sound is being generated in the Attack area, the bit A_R must be “1” to indicate Attack;

If the sound is being generated in the Release area, the bit A_R must be “0” to indicate Release.

For different volume change rates, which is shown by different slopes, then the ENV[1:0] value is listed as below:

If the VM[2:0] is incremented or decremented by 1, assign “b11” to the ENV[1:0];

If the VM[2:0] is incremented or decremented by 2, assign “b10” to the ENV[1:0];

If the VM[2:0] is incremented or decremented by 4, assign “b01” to the ENV[1:0];

If the VM[2:0] is incremented or decremented by 8, assign “b00” to the ENV[1:0].

Take the VM[2:0] increment of 8 as an example to show why this function should be initiated. The VM[2:0] increment gets a lower volume, the envelope of Release area will be discussed.

If the VM[2:0] initial value is 0x000 and the increment is 8, the VM[2:0] sequence will be 0x0, 0x8, 0x10, 0x18…

Clear the A_R to “0” and set the ENV[1:0] to “00” to initiate the control function, in this situation circuit will automatically supplement 1 ~ 7 between 0 and 8, as shown below, the sequence will be a more smooth Release envelope:

0x0, 0x1, 0x2, 0x3, 0x4, 0x5, 0x6, 0x7, 0x8, 0x9, 0xA, 0xB, 0xC, 0xD, 0xE, 0xF, 0x10, 0x11…

Interrupt Enable and Status Definition

The MIDIO_DMAEN bit is used to control the MIDIO DMA function. If it is enabled, the MIDI Engine will send a DMA TX request when processing channel 1. The DMA program will be arranged to automatically read the MIDIL and MIDIR registers and send the data to the external I^2S D/A converter, thus reducing the MCU data read load.

MIDIO_DMAEN = 0: Disable MIDIO DMA

MIDIO_DMAEN = 1: Enable MIDIO DMA

The MIDII_DMAEN bit is used to control the MIDII DMA function. If it is enabled, the MIDI Engine will send a DMA RX request when processing channel 1. The DMA program will be arranged to automatically read the external I^2S microphone, then send the data to the MIDI Engine MCU channel to implement automatic audio mixing, thus reducing the MCU data read load.

MIDII_DMAEN = 0: Disable MIDII DMA

MIDII_DMAEN = 1: Enable MIDII DMA

The INTEN bit is used to control the interrupt function

INTEN = 0: Disable interrupt

INTEN = 1: Enable interrupt

In the following example, XTAL = 12 MHz, channel number = 16. This is the same as the 20, 24, 28, and 32 channel equations. An interrupt generation indicates the following situations:

1. The MIDIL[23:0] and MIDIR[23:0] bit fields are ready. Here users can use the program to read the bit field values and send them to the MCU for further sound processing, which will then be sent to the external I^2S D/A converter. Or users can also enable the DMA function to send the register value to the external I^2S D/A converter directly if MCU sound processing is not required.

2. The SPID[31:0] field is ready. Here the MCU can read the register value as data or for other uses. Data in the SPID[31:0] field can be stored until the next interrupt occurs and will be overwritten if it is not read immediately. Therefore the MCU must read the data within a time period of (1 / 12 MHz) × 16 × 15 = 20 μs after the interrupt is generated.

   Considerations: After the interrupt function has been enabled, SPID[31:0] will not be ready until the second interrupt occurs. It is recommended that users first obtain the SPID[31:0] value by reading the SPIA[23:0] field.

3. If some of the channels are set as MCU channels, it is required to estimate the time consumed by the interrupt occurrence and the time consumed by the MCU sending the values to the registers MCU_CH0 ~ MCU_CH3.

Suppose that channel 11 is set as the MCU channel. Then as the interrupt occurrence time is stored in channel 1, the MCU must provide data within a time period of (1/12 MHz) × 16 × (11 – 1) = 13.3 μs after the interrupt is received, where the time 13.3 μs value is an estimated value. It is suggested that the interrupt should be processed immediately to obtain the correct data read and write. If the MCU channel data to be provided is the same as the last time, the MCU step to obtain the data can be ignored. This will result in a reduction of several instructions and consequently the interrupt processing time will be shorter.

Similar to the above example, change the MCU channel to channel 14. Then the MCU must provide data within a time period of (1 / 12 MHz) × 16 × (14-1) = 17.3 μs after the interrupt occurred. The limit time equation is shown as follows:

T = (1 / 12 MHz) × 16 × distance between the two channels

In the first example, an interrupt occurred in channel 1 so the distance between channel 1 and channel 11 is 10 channels. In the second example, an interrupt occurred in channel 1 so the distance between channel 1 and channel 14 is 13 channels.

MIDIL, MIDIR Registers

These two registers are read only. The MIDIL[23:0] and MIDIR[23:0] field values are the left and right channel sound synthesized by the MIDI Engine.

Users can make implement sound processing on the data read by program, which will then be sent to D/A converter to make variations to the sound, such as sound effects etc.

The MIDIL[23:0] and MIDIR[23:0] data formats are 24-bit 2's complement.

CTRL Register

The MCU_CH0 ~ MCU_CH3 bits are used to provide MCU channel data. Note that only channels 11 ~ 14 (CH = B ~ E) can be used as MCU channels. Their waveform sources are determined by the MCUCHEN[3:0] bits in the CTRL register. The waveform value is provided by the CHXB[15:0] and CHXA[15:0] fields, where X = 0 ~ 3.

If MCUCHEN[0] = 1, waveform in Channel 11 is sourced from the CH0B[15:0] and CH0A[15:0] fields, if MCUCHEN[0] = 0, the waveform is sourced from the external SPI Flash memory.

If MCUCHEN[1] = 1, waveform in Channel 12 is sourced from the CH1B[15:0] and CH1A[15:0] fields, if MCUCHEN[1] = 0, the waveform is sourced from the external SPI Flash memory.

If MCUCHEN[2] = 1, waveform in Channel 13 is sourced from the CH2B[15:0] and CH2A[15:0] fields, if MCUCHEN[2] = 0, the waveform is sourced from the external SPI Flash memory.

If MCUCHEN[3] = 1, waveform in Channel 14 is sourced from the CH3B[15:0] and CH3A[15:0] fields, if MCUCHEN[3] = 0, the waveform is sourced from the external SPI Flash memory.

The CHXA[15:0] and CHXB[15:0] bit field are used as two points for interpolation, CHXA[15:0] is the first point while CHXB[15:0] is the second point.

If the interrupt function is enabled, an interrupt will occur when MIDI Engine is processing the sound in channel 1. After the interrupt signal has been received, the MCU program should maintain the CHXA[15:0] and CHXB[15:0] contents properly to ensure that the MIDI Engine generates the right sound. If data in CHXA[15:0] and CHXB[15:0] to be sent is the same as previously then it is not necessary to send the data again.

If the DMA function is enabled, the MIDI Engine will send a DMA request when processing the sound in channel 1. Programs will transfer data through the DMA automatically without using the interrupt function.

The CHXB[15:0] and CHXA[15:0] data formats are 16-bit 2'complement. The CHXA[15:0] and CHXB[15:0] interpolation function is only used by the MCU to configure the waveform, while the channel operation and calculation still work as usual. Therefore as the channel-related parameters

are available, the volume can still be adjusted. Control bits such as A_R, ENV, VL, VR, etc. are also valid. Additionally, as the interpolation operation of the BL and FR field are still operational, it is recommended to properly control BL and FR as well as using data transmission to implement a point adding effect, thereby improving the waveform smoothness.

The DACDS[2:0] field is used to control the data processing method in each channel before data accumulation, thereby any distortion caused by an overlarge accumulated value can be avoided. The 16-channel mode has 5 choices ranging from "b000" to "b100" while the 20, 24, 28 and 32 channel modes have 6 choices ranging from "b000" to "b101".

16-channel mode:

DACDS = b000 ~ b111, where b000 is the default value and options b101 ~ b111 are the same as option b000.

000: Each channel is distortionless

001: Each channel shift right by 1 bit

010: Each channel shift right by 2 bits

011: Each channel shift right by 3 bits

100: Each channel shift right by 4 bits

20, 24, 28 and 32 channel mode:

DACDS = b000 ~ b111, where b000 is the default value and options b110 ~ b111 are the same as option b000.

000: Each channel is distortionless

001: Each channel shift right by 1 bit

010: Each channel shift right by 2 bits

011: Each channel shift right by 3 bits

100: Each channel shift right by 4 bits

101: Each channel shift right by 5 bits

The MUSIC_EN bit is used to enable / disable the MIDI Engine

MUSIC_EN = 0: Disable the MIDI engine

MUSIC_EN = 1: Enable the MIDI engine

The MUSIC_EN bit must be set high before transferring the parameters, which should be kept high during the whole MIDI Engine operation. When the sound has been fully synthesized and played and there is no more sound from the D/A converter, the MUSIC_EN bit can be cleared to disable the MIDI Engine for power saving.

SPI_RDEN:

SPI_RDEN = 0: Disable the function whereby the MCU reads the SPI Flash memory through MIDI engine channel 0.

SPI_RDEN = 1: Enable the function whereby the MCU reads the SPI Flash memory through MIDI engine channel 0.

SPI_DISLOOP:

SPI_DISLOOP = 1: channel 0 address will be incremented instead of returned when an SPI flash read accesses the end address.

SPI_DISLOOP = 0: the channel 0 address will be returned according to the RE[14:0] field when an SPI flash read accesses the end address.

The CHS[2:0] field is used to select multiple channels to play music simultaneously.

CHS = 0, 5, 6, 7: 16 channels

CHS = 1: 20 channels

CHS = 2: 24 channels

CHS = 3: 28 channels

CHS = 4: 32 channels

SPI_DATA Register

This register is read only and is used to read the external SPI Flash data. The read operation is implemented using the MIDI engine channel 0. ST[18:0] is used for addressing, and works alongside the BL and FR field setting. The address will be incremented automatically for data reading. The SPI_DISLOOP bit in the CTRL register will determine the address behavior when the read operation accesses the end address.

When a data read operation has completed, an interrupt will be generated when the MIDI Engine is processing the music in channel 1. The MCU must then read the data before the next channel 1 music is processed, otherwise the data will be updated automatically.

A specific example of how it is used is shown below:

A. Set SPI_RDEN high to allow the MCU to read the SPI Flash memory through the MIDI Engine

B. Set the {BL[3:0], FR[11:0]} field value to 0x8800, the address will be incremented by 4 each time. Setting the value to 0x6800 and the address will be incremented by 1 while setting the value to 0x7800 and the address will be incremented by 2. As four 8-bit blocks of data are read back each time, it is recommended to set the value to 0x8800.

C. Set the ST field to define the data read start address. When WBS is setup to select the 8-bit mode, according to the equation where WA = ST << 5 as described in the “Waveform Start Address” Section, the ST field value should be set to 0x01. Therefore the first data will be read at the waveform address “0x20”. Refer to the “Waveform Start Address” Section for the corresponding configuration for other WBS setting modes.

D. Set the WBS value according to specific requirements. For the 8-bit mode, four blocks of 8-bit data will be read back each time. For 16-bit, two blocks of 16-bit data will be read back each time. If the WBS field is setup to select the 12-bit mode, a 32-bit block of data will be read back each time and will be assembled as two 12-bit waveforms according to the 12-bit waveform storage rule. This 32-bit data can also be used as data.

E. If the purpose of the read operation is data, the SPI_DISLOOP bit should be set to 1 so that the data read process will not be disturbed by the end address, thus effectively avoiding address hopping.

F. If the purpose of the read operation is for voice or music, which is to be circulated in a certain waveform area, the SPI_DISLOOP bit should be cleared. Therefore the waveform will roll back at the end address according to the RE[14:0] field value. The RE field value should be properly set if the roll back function is required.

G. If channel 0 needs to produce sound and read back the original sound data simultaneously, the volume should be adjusted according to specific requirements. The channel volume should be turned to the minimum value to eliminate noise in channel 0 during the data read back operation.

SPI_ADDR Register

This register is read only. The return value of SPIA[23:0] indicates the address of SPID[31:0].

As the SPI Flash memory is read indirectly by the MIDI Engine, addressing is also implemented using the MIDI Engine addressing mechanism instead of addressing directly, allowing programmers to easily obtain the current SPI Flash address directly.

In addition, after the interrupt function is enabled, SPID[31:0] will not be ready until the second interrupt occurs. Programmers can obtain the SPID[31:0] bit value by reading the SPIA[23:0] field.

Register Map

The following table shows the MIDI Engine registers and reset values.

Table 72. MIDI Engine Register Map

Register	Offset	Description	Reset Value
CHAN	0x000	Channel Number Select Control Register	0x0000_0000
FREQ	0x004	Frequency Setting Register	0x0000_0000
VOL	0x008	Volume Control Register	0x63FF_03FF
ST_ADDR	0x00C	Start Address Register	0x0000_0000
RE_NUM	0x010	Repeat Number Register	0x0000_0000
END_ADDR	0x014	End Address Register	0x0000_0000
IER	0x018	Interrupt / DMA Enable Register	0x0000_0000
SR	0x01C	Status Register	0x0000_0000
MCU_CH0	0x020	MCU Channel 0 Data Register	0x0000_0000
MCU_CH1	0x024	MCU Channel 1 Data Register	0x0000_0000
MCU_CH2	0x028	MCU Channel 2 Data Register	0x0000_0000
MCU_CH3	0x02C	MCU Channel 3 Data Register	0x0000_0000
MIDIL	0x030	Left Channel MIDI Engine Waveform Register	0x0000_0000
MIDIR	0x034	Right Channel MIDI Engine Waveform Register	0x0000_0000
SPI_DATA	0x038	SPI Flash Data Register	0xXXXX_XXXX
SPI_ADDR	0x03C	SPI Flash Address Register	0x00XX_XXXX
CTRL	0x040	Control Register	0x0000_0000

Register Descriptions

Channel Number Select Control Register – CHAN

This register specifies the MIDI engine channel selection and setting.

Offset: 0x000

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	22	21	20	19	18	17	16
	Reserved							
Type/Reset								
	15	14	13	12	11	10	9	8
	Reserved					ST	VM	FR
Type/Reset						RW 0	RW 0	RW 0
	7	6	5	4	3	2	1	0
	Reserved			CH4	CH3	CH2	CH1	CH0
Type/Reset				RW 0	RW 0	RW 0	RW 0	RW 0

Bits	Field	Descriptions
[10]	ST	Update ST_ADDR, RE_NUM and END_ADDR Register Values 0: Not update 1: Update the ST_ADDR, RE_NUM and END_ADDR register values to the specified channel when writing CH[4:0]
[9]	VM	Update VOL Register Value 0: Not update 1: Update the VOL register value to the specified channel when writing CH[4:0]
[8]	FR	Update FREQ Register Value 0: Not update 1: Update the FREQ register value to the specified channel when writing CH[4:0]

Bits	Field	Descriptions
[4:0]	CH	MIDI Engine Channel Selection for the FREQ, VOL, ST_ADDR, RE_NUM and END_ADDR Register Setting 0x0: MIDI Engine CH0 Setting 0x1: MIDI Engine CH1 Setting 0x2: MIDI Engine CH2 Setting 0x3: MIDI Engine CH3 Setting 0x4: MIDI Engine CH4 Setting 0x5: MIDI Engine CH5 Setting 0x6: MIDI Engine CH6 Setting 0x7: MIDI Engine CH7 Setting 0x8: MIDI Engine CH8 Setting 0x9: MIDI Engine CH9 Setting 0xA: MIDI Engine CH10 Setting 0xB: MIDI Engine CH11 Setting 0xC: MIDI Engine CH12 Setting 0xD: MIDI Engine CH13 Setting 0xE: MIDI Engine CH14 Setting 0xF: MIDI Engine CH15 Setting 0x10: MIDI Engine CH16 Setting 0x11: MIDI Engine CH17 Setting 0x12: MIDI Engine CH18 Setting 0x13: MIDI Engine CH19 Setting 0x14: MIDI Engine CH20 Setting 0x15: MIDI Engine CH21 Setting 0x16: MIDI Engine CH22 Setting 0x17: MIDI Engine CH23 Setting 0x18: MIDI Engine CH24 Setting 0x19: MIDI Engine CH25 Setting 0x1A: MIDI Engine CH26 Setting 0x1B: MIDI Engine CH27 Setting 0x1C: MIDI Engine CH28 Setting 0x1D: MIDI Engine CH29 Setting 0x1E: MIDI Engine CH30 Setting 0x1F: MIDI Engine CH31 Setting

Frequency Setting Register – FREQ

This register specifies the channel frequency.

Offset: 0x004

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	22	21	20	19	18	17	16
	Reserved							
Type/Reset								
	15	14	13	12	11	10	9	8
	BL3	BL2	BL1	BL0	FR11	FR10	FR9	FR8
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0
	7	6	5	4	3	2	1	0
	FR7	FR6	FR5	FR4	FR3	FR2	FR1	FR0
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0

Bits	Field	Descriptions
[15:12]	BL	Set the Octave Frequency Point Note: The bit field value ranges from 0x0 to 0xB.
[11:0]	FR	Set the Tone Scale Frequency Point

Volume Control Register – VOL

This register specifies the volume control.

Offset: 0x008

Reset value: 0x63FF_03FF

	31	30	29	28	27	26	25	24
	A_R	ENV1	ENV0	Reserved			VL9	VL8
Type/Reset	RW 0	RW 1	RW 1				RW 1	RW 1
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
	VL7	VL6	VL5	VL4	VL3	VL2	VL1	VL0
Type/Reset	RW 1	RW 1	RW 1	RW 1	RW 1	RW 1	RW 1	RW 1
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
	Reserved						VR9	VR8
Type/Reset							RW 1	RW 1
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
	VR7	VR6	VR5	VR4	VR3	VR2	VR1	VR0
Type/Reset	RW 1	RW 1	RW 1	RW 1	RW 1	RW 1	RW 1	RW 1

Bits	Field	Descriptions
[31]	A_R	Attack / Release Volume Control 0: Release 1: Attack
[30:29]	ENV	Volume Control LSB 3-bit Control Method 11: By software Others: By hardware
[25:16]	VL	Left Channel Volume Control
[9:0]	VR	Right Channel Volume Control

Start Address Register – ST_ADDR

This register specifies the waveform start address.

Offset: 0x00C
Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	22	21	20	19	18	17	16
	Reserved					ST18	ST17	ST16
Type/Reset						RW 0	RW 0	RW 0
	15	14	13	12	11	10	9	8
	ST15	ST14	ST13	ST12	ST11	ST10	ST9	ST8
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0
	7	6	5	4	3	2	1	0
	ST7	ST6	ST5	ST4	ST3	ST2	ST1	ST0
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0

Bits	Field	Descriptions
[18:0]	ST	Waveform Start Address

Repeat Number Register – RE_NUM

This register specifies the waveform repeat length and sample format.

Offset: 0x010
Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	22	21	20	19	18	17	16
	Reserved						WBS1	WBS0
Type/Reset							RW 0	RW 0
	15	14	13	12	11	10	9	8
	Reserved	RE14	RE13	RE12	RE11	RE0	RE9	RE8
Type/Reset		RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0
	7	6	5	4	3	2	1	0
	RE7	RE6	RE5	RE4	RE3	RE2	RE1	RE0
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0

Bits	Field	Descriptions
[17:16]	WBS	Sample Format 00: 8-bit format 01: 12-bit format 1x: 16-bit format
[14:0]	RE	Repeat Length of Waveform (2's Complement)

End Address Register – END_ADDR

This register specifies the waveform end address.

Offset: 0x014

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	22	21	20	19	18	17	16
	EA23	EA22	EA21	EA20	EA19	EA18	EA17	EA16
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0
	15	14	13	12	11	10	9	8
	EA15	EA14	EA13	EA12	EA11	EA10	EA9	EA8
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0
	7	6	5	4	3	2	1	0
	EA7	EA6	EA5	EA4	EA3	EA2	EA1	EA0
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0

Bits	Field	Descriptions
[23:0]	EA	Waveform End Address

Interrupt / DMA Enable Register – IER

This register specifies the interrupt / DMA enable register.

Offset: 0x018

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	22	21	20	19	18	17	16
	Reserved							
Type/Reset								
	15	14	13	12	11	10	9	8
	Reserved							
Type/Reset								
	7	6	5	4	3	2	1	0
	Reserved					MIDIO_DMAEN	MIDII_DMAEN	INTEN
Type/Reset						RW 0	RW 0	RW 0

Bits	Field	Descriptions
[2]	MIDIO_DMAEN	MIDI Output DMA Enable Control 0: Disable the MIDI output DMA 1: Enable the MIDI output DMA

Bits	Field	Descriptions
[1]	MIDII_DMAEN	MIDI Input DMA Enable Control 0: Disable the MIDI input DMA 1: Enable the MIDI input DMA
[0]	INTEN	MIDI Engine Interrupt Enable Control 0: Disable the MIDI interrupt 1: Enable the MIDI interrupt

Status Register – STATUS

This register specifies the status register.

Offset: 0x01C

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	22	21	20	19	18	17	16
	Reserved							
Type/Reset								
	15	14	13	12	11	10	9	8
	Reserved							
Type/Reset								
	7	6	5	4	3	2	1	0
	Reserved							INTF
Type/Reset								RW 0

Bits	Field	Descriptions
[0]	INTF	MIDI Interrupt Flag 0: MIDI interrupt not occur 1: MIDI interrupt occurs

MCU Channel 0 Data Register – MCU_CH0

This register specifies the MCU Channel 0 data input for interpolation.

Offset: 0x020

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	CH0B15	CH0B14	CH0B13	CH0B12	CH0B11	CH0B10	CH0B9	CH0B8
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0
	23	22	21	20	19	18	17	16
	CH0B7	CH0B6	CH0B5	CH0B4	CH0B3	CH0B2	CH0B1	CH0B0
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0
	15	14	13	12	11	10	9	8
	CH0A15	CH0A14	CH0A13	CH0A12	CH0A11	CH0A10	CH0A9	CH0A8
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0
	7	6	5	4	3	2	1	0
	CH0A7	CH0A6	CH0A5	CH0A4	CH0A3	CH0A2	CH0A1	CH0A0
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0

Bits	Field	Descriptions
[31:16]	CH0B	MCU Channel 0 Data Input B (Used for Interpolation)
[15:0]	CH0A	MCU Channel 0 Data Input A (Used for Interpolation)

MCU Channel 1 Data Register – MCU_CH1

This register specifies the MCU Channel 1 data input for interpolation.

Offset: 0x024

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	CH1B15	CH1B14	CH1B13	CH1B12	CH1B11	CH1B10	CH1B9	CH1B8
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0
	23	22	21	20	19	18	17	16
	CH1B7	CH1B6	CH1B5	CH1B4	CH1B3	CH1B2	CH1B1	CH1B0
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0
	15	14	13	12	11	10	9	8
	CH1A15	CH1A14	CH1A13	CH1A12	CH1A11	CH1A10	CH1A9	CH1A8
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0
	7	6	5	4	3	2	1	0
	CH1A7	CH1A6	CH1A5	CH1A4	CH1A3	CH1A2	CH1A1	CH1A0
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0

Bits	Field	Descriptions
[31:16]	CH1B	MCU Channel 1 Data Input B (Used for Interpolation)
[15:0]	CH1A	MCU Channel 1 Data Input A (Used for Interpolation)

MCU Channel 2 Data Register – MCU_CH2

This register specifies the MCU Channel 2 data input for interpolation.

Offset: 0x028

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	CH2B15	CH2B14	CH2B13	CH2B12	CH2B11	CH2B10	CH2B9	CH2B8
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0
	23	22	21	20	19	18	17	16
	CH2B7	CH2B6	CH2B5	CH2B4	CH2B3	CH2B2	CH2B1	CH2B0
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0
	15	14	13	12	11	10	9	8
	CH2A15	CH2A14	CH2A13	CH2A12	CH2A11	CH2A10	CH2A9	CH2A8
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0
	7	6	5	4	3	2	1	0
	CH2A7	CH2A6	CH2A5	CH2A4	CH2A3	CH2A2	CH2A1	CH2A0
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0

Bits	Field	Descriptions
[31:16]	CH2B	MCU Channel 2 Data Input B (Used for Interpolation)
[15:0]	CH2A	MCU Channel 2 Data Input A (Used for Interpolation)

MCU Channel 3 Data Register – MCU_CH3

This register specifies the MCU Channel 3 data input for interpolation.

Offset: 0x02C

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	CH3B15	CH3B14	CH3B13	CH3B12	CH3B11	CH3B10	CH3B9	CH3B8
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0
	23	22	21	20	19	18	17	16
	CH3B7	CH3B6	CH3B5	CH3B4	CH3B3	CH3B2	CH3B1	CH3B0
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0
	15	14	13	12	11	10	9	8
	CH3A15	CH3A14	CH3A13	CH3A12	CH3A11	CH3A10	CH3A9	CH3A8
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0
	7	6	5	4	3	2	1	0
	CH3A7	CH3A6	CH3A5	CH3A4	CH3A3	CH3A2	CH3A1	CH3A0
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0

Bits	Field	Descriptions
[31:16]	CH3B	MCU Channel 3 Data Input B (Used for Interpolation)
[15:0]	CH3A	MCU Channel 3 Data Input A (Used for Interpolation)

MIDI Engine Left Channel Data Output Register – MIDI_L

This register specifies the Left channel MIDI output data.

Offset: 0x030

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	22	21	20	19	18	17	16
	MIDIL23	MIDIL22	MIDIL21	MIDIL20	MIDIL19	MIDIL18	MIDIL17	MIDIL16
Type/Reset	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0
	15	14	13	12	11	10	9	8
	MIDIL15	MIDIL14	MIDIL13	MIDIL12	MIDIL11	MIDIL10	MIDIL9	MIDIL8
Type/Reset	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0
	7	6	5	4	3	2	1	0
	MIDIL7	MIDIL6	MIDIL5	MIDIL4	MIDIL3	MIDIL2	MIDIL1	MIDIL0
Type/Reset	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0

Bits	Field	Descriptions
[23:0]	MIDIL	MIDI Engine Left Channel Output Data

MIDI Engine Right Channel Data Output Register – MIDI_R

This register specifies the Right channel MIDI output data.

Offset: 0x034

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	22	21	20	19	18	17	16
	MIDIR23	MIDIR22	MIDIR21	MIDIR20	MIDIR19	MIDIR18	MIDIR17	MIDIR16
Type/Reset	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0
	15	14	13	12	11	10	9	8
	MIDIR15	MIDIR14	MIDIR13	MIDIR12	MIDIR11	MIDIR10	MIDIR9	MIDIR8
Type/Reset	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0
	7	6	5	4	3	2	1	0
	MIDIR7	MIDIR6	MIDIR5	MIDIR4	MIDIR3	MIDIR2	MIDIR1	MIDIR0
Type/Reset	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0

Bits	Field	Descriptions
[23:0]	MIDIR	MIDI Engine Right Channel Output Data

SPI Flash Data Register – SPI_DATA

This register is used to store the data read from external SPI Flash memory.

Offset: 0x038

Reset value: 0xXXXX_XXXX

	31	30	29	28	27	26	25	24
	SPID31	SPID30	SPID29	SPID28	SPID27	SPID26	SPID25	SPID24
Type/Reset	RO X	RO X	RO X	RO X	RO X	RO X	RO X	RO X
	23	22	21	20	19	18	17	16
	SPID23	SPID22	SPID21	SPID20	SPID19	SPID18	SPID17	SPID16
Type/Reset	RO X	RO X	RO X	RO X	RO X	RO X	RO X	RO X
	15	14	13	12	11	10	9	8
	SPID15	SPID14	SPID13	SPID12	SPID11	SPID10	SPID9	SPID8
Type/Reset	RO X	RO X	RO X	RO X	RO X	RO X	RO X	RO X
	7	6	5	4	3	2	1	0
	SPID7	SPID6	SPID5	SPID4	SPID3	SPID2	SPID1	SPID0
Type/Reset	RO X	RO X	RO X	RO X	RO X	RO X	RO X	RO X

Bits	Field	Descriptions
[31:0]	SPID	The Data Read from External SPI Flash Memory

SPI Flash Address Register – SPI_ADDR

This register is used to store the address of the SPI_DATA.

Offset: 0x03C

Reset value: 0x00XX_XXXX

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	22	21	20	19	18	17	16
	SPIA23	SPIA22	SPIA21	SPIA20	SPIA19	SPIA18	SPIA17	SPIA16
Type/Reset	RO X	RO X	RO X	RO X	RO X	RO X	RO X	RO X
	15	14	13	12	11	10	9	8
	SPIA15	SPIA14	SPIA13	SPIA12	SPIA11	SPIA10	SPIA9	SPIA8
Type/Reset	RO X	RO X	RO X	RO X	RO X	RO X	RO X	RO X
	7	6	5	4	3	2	1	0
	SPIA7	SPIA6	SPIA5	SPIA4	SPIA3	SPIA2	SPIA1	SPIA0
Type/Reset	RO X	RO X	RO X	RO X	RO X	RO X	RO X	RO X

Bits	Field	Descriptions
[23:0]	SPIA	The Address of the Data Read from External SPI Flash Memory

Control Register – CTRL

This register specifies the control bit of the MIDI Engine.

Offset: 0x040

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	22	21	20	19	18	17	16
	Reserved							
Type/Reset								
	15	14	13	12	11	10	9	8
	MCUCHEN3	MCUCHEN2	MCUCHEN1	MCUCHEN0	Reserved	DACDS2	DACDS1	DACDS0
Type/Reset	RW 0	RW 0	RW 0	RW 0		RW 0	RW 0	RW 0
	7	6	5	4	3	2	1	0
	MUSIC_EN	SPI_RDEN	SPI_DISLOOP	Reserved	Reserved	CHS2	CHS1	CHS0
Type/Reset	RW 0	RW 0	RW 0			RW 0	RW 0	RW 0

Bits	Field	Descriptions
[15:12]	MCUCHEN	Channel 11 ~ 14 Waveform Data Source Selection 0: External SPI Flash memory 1: CHxA and CHxB Registers
[10:8]	DACDS	Data Processing Method in Each Channel before Data Accumulation 000: Original sound data 001: Shift right by 1 bit 010: Shift right by 2 bits 011: Shift right by 3 bits 100: Shift right by 4 bits 101: Shift right by 5 bits (Only for 20, 24, 28 and 32 Channels) Others: Original sound data
[7]	MUSIC_EN	MIDI Engine Enable / Disable Control 0: Disable the MIDI Engine 1: Enable the MIDI Engine
[6]	SPI_RDEN	Function Control for the MIDI Engine reading SPI Flash Memory through Channel 0 0: Disable 1: Enable
[5]	SPI_DISLOOP	Behavior when the SPI Flash Memory Data Read Operation accesses the End Address 0: When accessing the end address, the channel 0 address will return according to the RE[14:0] field 1: When accessing the end address, the channel 0 address will be incremented instead of returning
[2:0]	CHS	Select multiple Channel numbers to play music simultaneously 000 / 101 / 110 / 111: 16 channels 001: 20 channels 010: 24 channels 011: 28 channels 100: 32 channels

27 Digital-to-Analog Converter (DAC)

Introduction

The DAC module contains two 16-bit voltage output digital-to-analog converters. An input reference voltage, VDDA, which is shared with ADC, is available. The stereo output pins are the DAC_LCH for L-Channel and the DAC_RCH for R-Channel.

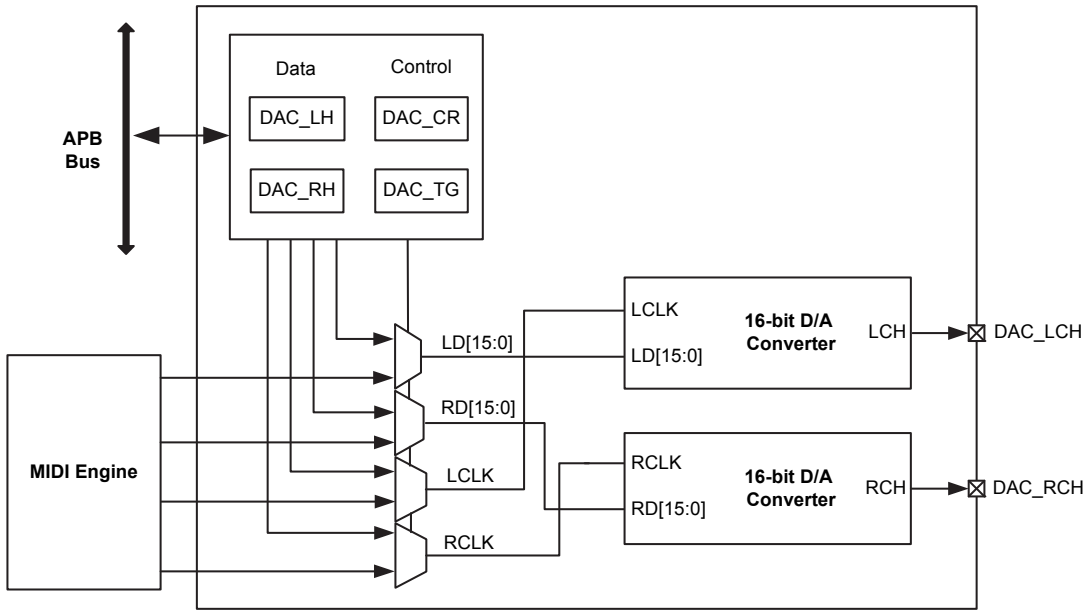


Figure 173. DAC Block Diagram

Features

- Two 16-bit R2R digital-to-analog converters
- Stereo audio output with buffer
- Power down control

Functional Descriptions

The DAC module is composed of two 16-bit D/A Converters with R2R structure for stereo audio application. The DAC module reference voltage is sourced from the analog power supply, VDDA, and can be powered down by clearing the DACCEN bit in the APBCCR1 register to save power. Although this DAC module does not provide general one-to-one digital to analog conversion, it provides a nice and same audio quality regardless of the volume of the voice. This makes the 16-bit DAC module suitable for voice or audio applications. The DAC module voltage is amplified and data in the buffer is output by OPAMP.

The DAC can be selected to be controlled by the MIDI Engine or by software. If it is controlled by software, the stereo voice data located in the DAC_RH and DAC_LH registers can be output to the DAC module output pins DAC_RCH and DAC_LCH separately. The R-Channel DAC data is latched at the rising edge of RSWTG while the L-Channel DAC data is latched at the rising edge of LSWTG.

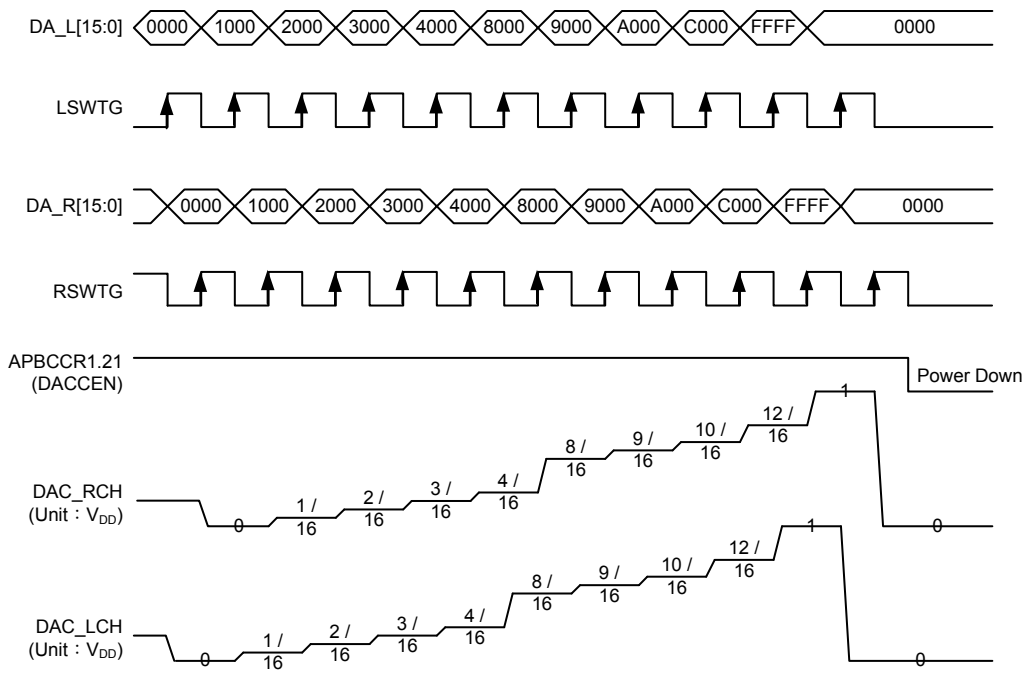


Figure 174. Timing Diagram when DAC is Controlled by Software (SELWR = 0, SELWL = 0)

Register Map

The following table shows the DAC registers and reset values.

Table 73. DAC Register Map

Register	Offset	Description	Reset Value
DAC_CR	0x000	DAC Control Register	0x0000_0000
DAC_RH	0x004	DAC R-channel Data Register	0x0000_0000
DAC_LH	0x008	DAC L-channel Data Register	0x0000_0000
DAC_TG	0x00C	DAC Data Trigger Control Register	0x0000_0000

Register Descriptions

DAC Control Register – DAC_CR

This register contains the DAC data source selection bits.

Offset: 0x000

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	22	21	20	19	18	17	16
	Reserved							
Type/Reset								
	15	14	13	12	11	10	9	8
	Reserved							SELWL
Type/Reset								RW 0
	7	6	5	4	3	2	1	0
	Reserved							SELWR
Type/Reset								RW 0

Bits	Field	Descriptions
[8]	SELWL	Select the L-channel data and control signal source 0: Data from software 1: Data from MIDI Engine
[0]	SELWR	Select the R-channel data and control signal source 0: Data from software 1: Data from MIDI Engine

DAC R-channel Data Register – DAC_RH

This register contains the R-channel data bits.

Offset: 0x004

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	22	21	20	19	18	17	16
	Reserved							
Type/Reset								
	15	14	13	12	11	10	9	8
	DA_R15	DA_R14	DA_R13	DA_R12	DA_R11	DA_R10	DA_R9	DA_R8
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0
	7	6	5	4	3	2	1	0
	DA_R7	DA_R6	DA_R5	DA_R4	DA_R3	DA_R2	DA_R1	DA_R0
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0

Bits	Field	Descriptions
[15:0]	DA_R	R-channel data

DAC L-channel Data Register – DAC_LH

This register contains the L-channel data bits.

Offset: 0x008

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	22	21	20	19	18	17	16
	Reserved							
Type/Reset								
	15	14	13	12	11	10	9	8
	DA_L15	DA_L14	DA_L13	DA_L12	DA_L11	DA_L10	DA_L9	DA_L8
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0
	7	6	5	4	3	2	1	0
	DA_L7	DA_L6	DA_L5	DA_L4	DA_L3	DA_L2	DA_L1	DA_L0
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0

Bits	Field	Descriptions
[15:0]	DA_L	L-channel data

DAC Data Trigger Control Register – DAC_TG

This register contains the DAC output control bits.

Offset: 0x00C

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	22	21	20	19	18	17	16
	Reserved							
Type/Reset								
	15	14	13	12	11	10	9	8
	Reserved							LSWTG
Type/Reset								RW 0
	7	6	5	4	3	2	1	0
	Reserved							RSWTG
Type/Reset								RW 0

Bits	Field	Descriptions
[8]	LSWTG	Software trigger for L-channel DAC data Data in the DAC_LH register will be latched into the L-channel D/A Converter at the rising edge of LSWTG.
[0]	RSWTG	Software trigger for R-channel DAC data Data in the DAC_RH register will be latched into the R-channel D/A Converter at the rising edge of RSWTG.

28 Divider (DIV)

Introduction

In order to enhance MCU performance, a divider is implemented within the device.

Features

- Signed / unsigned 32-bit divider
- Operation in 8 clock cycles, Load in 1 clock cycle.
- Division by zero error flag.

Functional Descriptions

The division and modulus functions of the truncated division are related in the following way:

$$A / B = Q \ldots R$$

Where "A" is Dividend, "B" is Divisor, "Q" is Quotient and "R" is Remainder. Divider requires a software trigger start signal by controlling the "Start" bit in the CR register. The divider calculation complete flag will be set to 1 after 8 clock cycles, however, if the divisor register data is zero during the calculation, the division by zero error flag will be set to 1.

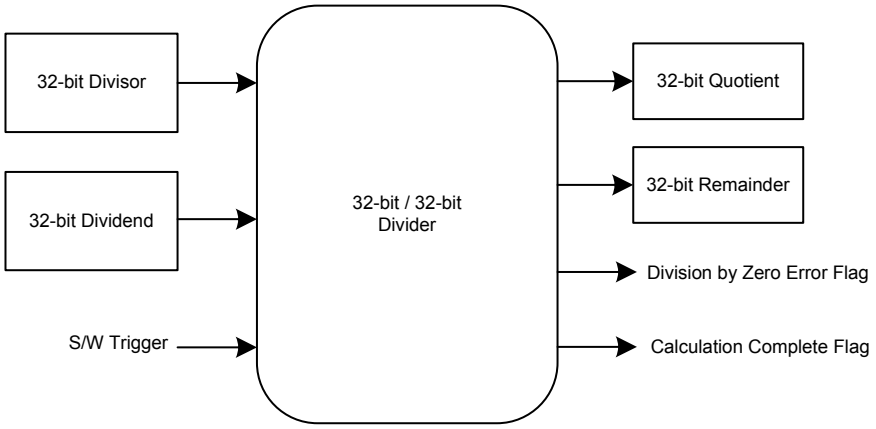


Figure 175. Divider Functional Diagram

Register Map

The following table shows the DIV registers and reset values.

Table 74. DIV Register Map

Register	Offset	Description	Reset Value
CR	0x000	Divider Control Register	0x0000_0000
DDR	0x004	Dividend Data Register	0x0000_0000
DSR	0x008	Divisor Data Register	0x0000_0000
QTR	0x00C	Quotient Data Register	0x0000_0000
RMR	0x010	Remainder Data Register	0x0000_0000

Register Descriptions

Divider Control Register – CR

This register contains the divider calculation complete flag, division by zero error flag and the calculation start control bit

Offset: 0x000

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	22	21	20	19	18	17	16
	Reserved							
Type/Reset								
	15	14	13	12	11	10	9	8
	Reserved							
Type/Reset								
	7	6	5	4	3	2	1	0
	Reserved				COM	ZEF	Reserved	Start
Type/Reset					RO 0	RO 0		RW 0

Bits	Field	Descriptions
[3]	COM	Calculation complete flag 0: Data are invalid 1: New data are valid If this bit is set to 1, it indicates that the divider calculation is completed and data are valid. This bit is cleared to 0 by hardware after the calculation start.
[2]	ZEF	Division by Zero Error Flag 0: Divisor is not zero 1: Divisor is zero This bit is cleared to 0 by hardware after the calculation start.
[0]	Start	Calculation start control bit 0: No operation 1: Start the divider calculation When write 1 to this bit, divider will start calculation.

Dividend Data Register – DDR

The register contains the dividend of the divider.

Offset: 0x004

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	DDR							
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0
	23	22	21	20	19	18	17	16
	DDR							
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0
	15	14	13	12	11	10	9	8
	DDR							
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0
	7	6	5	4	3	2	1	0
	DDR							
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0

Bits	Field	Descriptions
[31:0]	DDR	This bit field is used to store the dividend of the divider calculation.

Divisor Data Register – DSR

The register contains the divisor of the divider.

Offset: 0x008

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	DSR							
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0
	23	22	21	20	19	18	17	16
	DSR							
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0
	15	14	13	12	11	10	9	8
	DSR							
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0
	7	6	5	4	3	2	1	0
	DSR							
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0

Bits	Field	Descriptions
[31:0]	DSR	This bit field is used to store the divisor of the divider calculation.

Quotient Data Register – QTR

The register is used to store the quotient of the divider calculation result.

Offset: 0x00C

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	QTR							
Type/Reset	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0
	23	22	21	20	19	18	17	16
	QTR							
Type/Reset	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0
	15	14	13	12	11	10	9	8
	QTR							
Type/Reset	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0
	7	6	5	4	3	2	1	0
	QTR							
Type/Reset	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0

Bits	Field	Descriptions
[31:0]	QTR	This bit field is used to store the quotient of the divider calculation result.

Remainder Data Register – RMR

The register is used to store the remainder of the divider calculation result.

Offset: 0x010

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	RMR							
Type/Reset	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0
	23	22	21	20	19	18	17	16
	RMR							
Type/Reset	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0
	15	14	13	12	11	10	9	8
	RMR							
Type/Reset	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0
	7	6	5	4	3	2	1	0
	RMR							
Type/Reset	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0

Bits	Field	Descriptions
[31:0]	RMR	This bit field is used to store the remainder of the divider calculation result.

29 Cyclic Redundancy Check (CRC)

Introduction

The CRC (Cyclic Redundancy Check) calculation unit is an error detection technique test algorithm and is used to verify data transmission or storage data correctness. A CRC calculation takes a data stream or a block of data as input and generates a 16-bit or 32-bit output remainder. Ordinarily, a data stream is suffixed by a CRC code and used as a checksum when being sent or stored. Therefore, the received or restored data stream is calculated by the same generator polynomial as described above. If the new CRC code result does not match the one calculated earlier, that means data stream contains a data error.

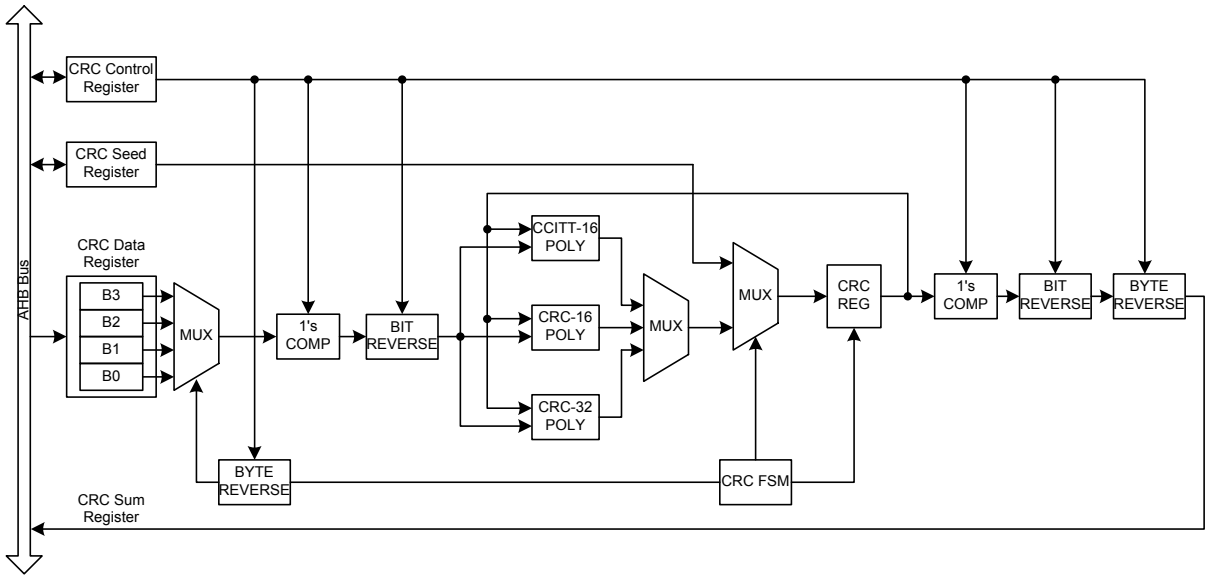


Figure 176. CRC Block Diagram

Features

- Support CRC16 polynomial: 0x8005, $X^{16} + X^{15} + X^2 + 1$
- Support CCITT CRC16 polynomial: 0x1021, $X^{16} + X^{12} + X^5 + 1$
- Support IEEE-802.3 CRC32 polynomial: 0x04C11DB7, $X^{32} + X^{26} + X^{23} + X^{22} + X^{16} + X^{12} + X^{11} + X^{10} + X^8 + X^7 + X^5 + X^4 + X^2 + X + 1$
- Supports 1's complement, byte reverse & bit reverse operation on data and checksum
- Supports byte, half-word & word data size
- Programmable CRC initial seed value
- CRC computation done in 1 AHB clock cycle for 8-bit data and 4 AHB clock cycles for 32-bit data
- Supports PDMA to complete a CRC computation of a block of memory

Function Descriptions

This unit only enables the calculation in the CRC16, CCITT CRC16 and IEEE-802.3 CRC32 polynomials. In this unit, the generator polynomial is fixed to the numeric values for those modes; therefore, the CRC value based on other generator polynomials cannot be calculated.

CRC Computation

The CRC calculation unit has a 32-bit write CRC data register (CRCDR) and a read CRC checksum register (CRCCSR). The CRCDR register is used to input new data (write access) and the CRCCSR register is used to hold the result of the previous CRC calculation (read access). Each write operation to the CRCDR register creates a combination of the previous CRC value (stored in CRCCSR) and the new one. The CRC block diagram is shown as Figure 176. The CRC unit calculates the CRC data register (CRCDR) value byte by byte and the default byte and bit order is big-endian. The CRCDR register can be written by word, right-aligned half-word and right-aligned byte. For the other registers only 32-bit access is allowed. The duration of the computation depends on data width:

- 4 AHB clock cycles for 32-bit data input
- 2 AHB clock cycles for 16-bit data input
- 1 AHB clock cycle for 8-bit data input

Byte and Bit Reversal for CRC Computation

The byte reordering and byte-level bit reversal operation can be occurred before the data is used in the CRC calculation or after the CRC checksum output. They are configurable using the corresponding setting field of the CRCCR register. These operations occur on word or half-word writes. The hardware ignores the DATBYRV bit of the CRCCR register with any byte writes but the bit reversal setting DATBIRV are still applied to the byte. The Figure 177 is shown the byte and bit reversal operation example.

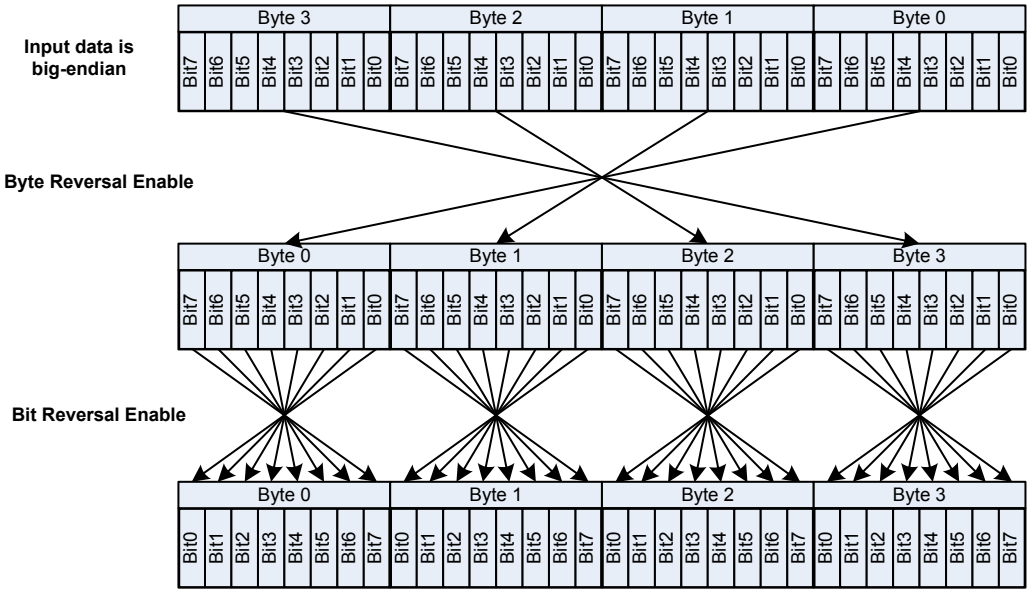


Figure 177. CRC Data Bit and Byte Reversal Example

CRC with PDMA

A PDMA channel with software trigger may be used to transfer data into the CRC unit. If a huge block data needs to be calculated, the recommended PDMA model is to use the PDMA to transfer all available words of data and uses software writes to transfer the other remaining bytes. To write data into the CRC unit, the PDMA should transfer data by word write format from the sou rce location of memory to the CRC data register (CRCDR) in non-incrementing address mode. Then software can write any remaining bytes to the CRC data register (CRCDR) and read the CRC calculation result value from the CRC checksum register (CRCCSR).

Register Map

The following table shows the CRC registers and reset values.

Table 75. CRC Register Map

Register	Offset	Description	Reset Value
CRCCR	0x000	CRC Control Register	0x0000_0000
CRCSDR	0x004	CRC Seed Register	0x0000_0000
CRCCSR	0x008	CRC Checksum Register	0x0000_0000
CRCDR	0x00C	CRC Data Register	0x0000_0000

Register Descriptions

CRC Control Register – CRCCR

This register specifies the corresponding CRC function enable control.

Offset: 0x000

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	Reserved							
Type/Reset								
	23	22	21	20	19	18	17	16
	Reserved							
Type/Reset								
	15	14	13	12	11	10	9	8
	Reserved							
Type/Reset								

	7	6	5	4	3	2	1	0
	SUMCMPL	SUMBYRV	SUMBIRV	DATCMPL	DATBYRV	DATBIRV	POLY	
Type/Reset	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0	RW 0

Bits	Field	Descriptions
[7]	SUMCMPL	1's Complement operation on Checksum Output 0: Disable 1: Enable
[6]	SUMBYRV	Byte Reverse operation on Checksum Output 0: Disable 1: Enable
[5]	SUMBIRV	Bit Reverse operation on Checksum Output 0: Disable 1: Enable
[4]	DATCMPL	1's Complement operation on Data 0: Disable 1: Enable
[3]	DATBYRV	Byte Reverse operation on Data 0: Disable 1: Enable
[2]	DATBIRV	Bit Reverse operation on Data 0: Disable 1: Enable
[1:0]	POLY	CRC polynomial 00: CRC-CCITT (0x1021) 01: CRC-16 (0x8005) 1x: CRC-32 (0x04C11DB7)

CRC Seed Register – CRCSDR

This register is used to specify the CRC seed.

Offset: 0x004

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	SEED							
Type/Reset	WO 0	WO 0	WO 0	WO 0	WO 0	WO 0	WO 0	WO 0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
	SEED							
Type/Reset	WO 0	WO 0	WO 0	WO 0	WO 0	WO 0	WO 0	WO 0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
	SEED							
Type/Reset	WO 0	WO 0	WO 0	WO 0	WO 0	WO 0	WO 0	WO 0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
	SEED							
Type/Reset	WO 0	WO 0	WO 0	WO 0	WO 0	WO 0	WO 0	WO 0

Bits	Field	Descriptions
[31:0]	SEED	CRC Seed Data Put the 16 / 32-bit seed value in this register according to the polynomial setting in the CRCCR register.

CRC Checksum Register – CRCCSR

This register contains the CRC checksum output.

Offset: 0x008

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	CHKSUM							
Type/Reset	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0
	23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
	CHKSUM							
Type/Reset	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
	CHKSUM							
Type/Reset	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
	CHKSUM							
Type/Reset	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0	RO 0

Bits	Field	Descriptions
[31:0]	CHKSUM	CRC Checksum Data Get the CRC 16 / 32-bit checksum result through this register according to the polynomial setting in the CRCCR register after all data are written to the CRCDR register.

CRC Data Register – CRCDR

This register is used to specify the CRC input data.

Offset: 0x00C

Reset value: 0x0000_0000

	31	30	29	28	27	26	25	24
	CRCDATA							
Type/Reset	WO 0	WO 0	WO 0	WO 0	WO 0	WO 0	WO 0	WO 0
	23	22	21	20	19	18	17	16
	CRCDATA							
Type/Reset	WO 0	WO 0	WO 0	WO 0	WO 0	WO 0	WO 0	WO 0
	15	14	13	12	11	10	9	8
	CRCDATA							
Type/Reset	WO 0	WO 0	WO 0	WO 0	WO 0	WO 0	WO 0	WO 0
	7	6	5	4	3	2	1	0
	CRCDATA							
Type/Reset	WO 0	WO 0	WO 0	WO 0	WO 0	WO 0	WO 0	WO 0

Bits	Field	Descriptions
[31:0]	CRCDATA	CRC Input Data Byte, half-word and word writes are allowed. 1's complement, byte reverse and bit reverse operation can be applied.