GigaDevice Semiconductor Inc.

GD32C2x1xx
Arm® Cortex®-M23 32-bit MCU

for GD32C231xx

User Manual

Revision 1.0

(Jun 2025)

Table of Contents

List of Figures

List of Tables

1. System and memory architecture

The GD32C2x1 series are 32-bit general-purpose microcontrollers based on the Arm® Cortex®-M23 processor. The Arm® Cortex®-M23 processor includes AHB buses. All memory accesses of the Arm® Cortex®-M23 processor are executed on the AHB buses according to the different purposes and the target memory spaces. The memory organization uses a ARMv8M architecture, pre-defined memory map and up to 4 GB of memory space, making the system flexible and extendable.

1.1. Arm® Cortex®-M23 processor

The Arm® Cortex®-M23 processor is an energy-efficient processor with a very low gate count. It is intended to be used for microcontroller and deeply embedded applications that require an area-optimized processor. It offers significant benefits to developers, including:

- A simple architecture that is easy to learn and program.
- Ultra-low power, energy-efficient operation.
- Excellent code density.
- Deterministic, high-performance interrupt handling.
- Upward compatibility with Cortex-M processor family.

The processor delivers high energy efficiency through a small but powerful instruction set and extensively optimized design, providing high-end processing hardware including a single-cycle multiplier and a 17-cycle divider.

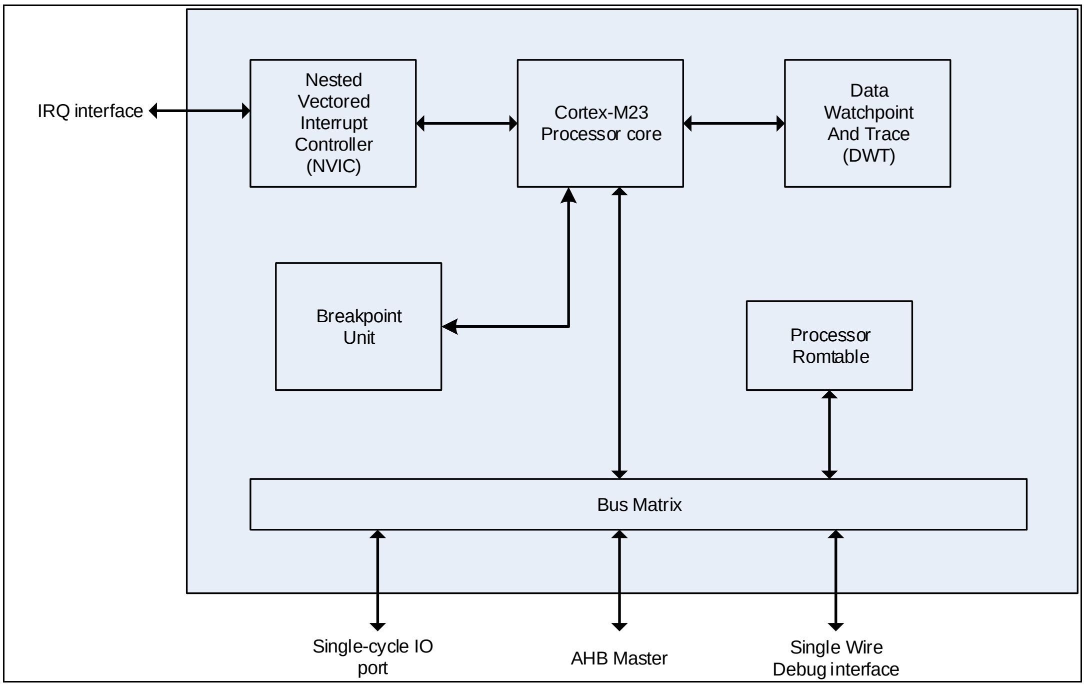
The Arm® Cortex®-M23 processor closely integrates a configurable Nested Vectored Interrupt Controller (NVIC), to deliver industry-leading interrupt performance.

Some system peripherals listed below are also provided by Cortex®-M23:

- Low latency, high-speed peripheral I/O port
- A Vector Table Offset Register
- Breakpoint unit
- Data Watchpoint
- Serial Wire Debug Port

The following figure shows the Arm® Cortex®-M23 processor block diagram. For more information, refer to the Arm® Cortex®-M23 Technical Reference Manual.

Figure 1-1. The structure of the Arm® Cortex®-M23 processor


IRQ interface
Nested Vectored Interrupt Controller (NVIC)
Cortex-M23 Processor core
Data Watchpoint And Trace (DWT)
Breakpoint Unit
Processor Romtable
Bus Matrix
Single-cycle IO port
AHB Master
Single Wire Debug interface

1.2. System architecture

The Bus Matrix is implemented in the GD32C2x1 devices, which manages the access arbitration between masters and Round Robin algorithm is used in arbitration. The bus matrix provides access from a master to a slave, enabling concurrent access and efficient operation even when several high-speed peripherals work simultaneously. A 32-bit multilayer bus is implemented in the devices, which enables parallel access paths between multiple masters and slaves in the system. The multilayer bus consists of an AHB interconnect matrix, two AHB bus. The interconnection relationship of the AHB interconnect matrix is shown below. In the following table, “1” indicates the corresponding master is able to access the corresponding slave through the AHB interconnect matrix, while the blank means the corresponding master cannot access the corresponding slave through the AHB interconnect matrix. This architecture is shown in ***Table 1-1. Bus Interconnection Matrix***.

Table 1-1. Bus Interconnection Matrix

	SBUS	DMA
FMC	1	1
SRAM	1	1
AHB1	1	1
AHB2	1	1

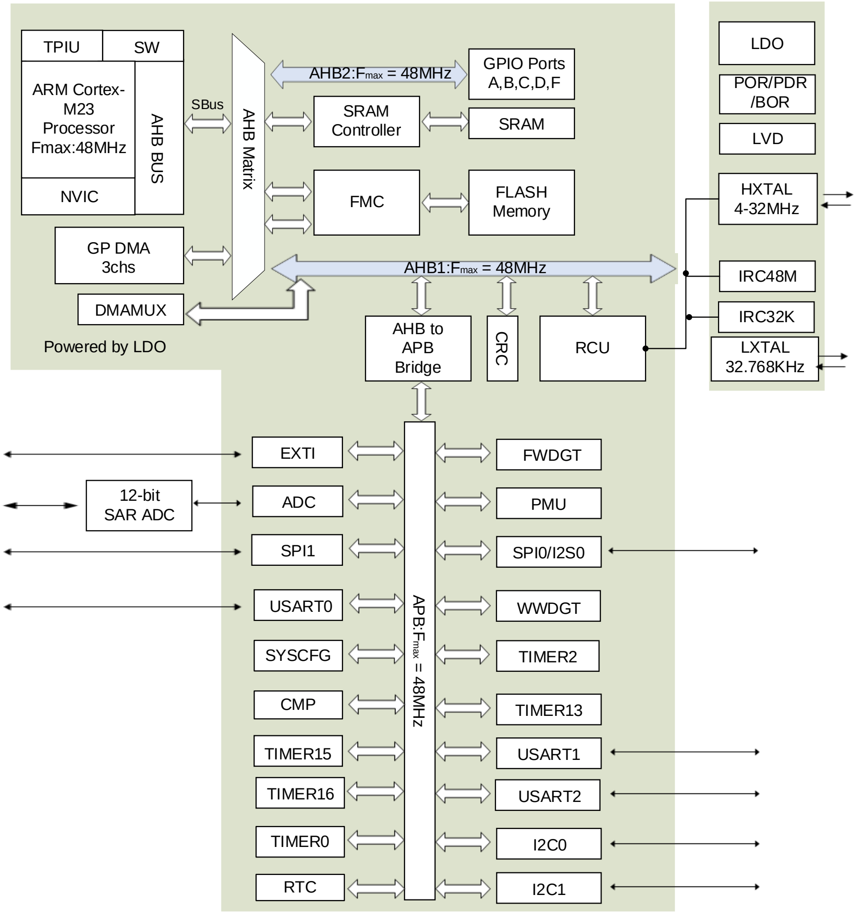
As is shown above, there are two masters connected with the AHB interconnect matrix, including SBUS and DMA. CPU SBUS connects the system bus of the Cortex®-M23 core (peripheral bus) to a bus matrix that manages the arbitration between the core and the DMA. DMA bus connects the AHB master interface of the DMA to the bus matrix that manages the access of CPU and DMA to SRAM, Flash memory and AHB/APB peripherals.

There are also several slaves connected with the AHB interconnect matrix, including FMC, SRAM, AHB1, AHB2. FMC is the bus interface of the flash memory controller. SRAM is on-chip static random access memories. AHB1 is the AHB bus connected with all of the AHB1 slaves and AHB-to-APB bridges. AHB2 is the AHB bus connected with AHB2 slaves. While AHB-to-APB bridges are the two APB buses connected with all of the APB slaves. The two APB buses connect with all the APB peripherals. APB is limited to 48Mhz.

The system architecture of GD32C2x1 series is shown in the following figure. The AHB matrix based on AMBA 5 AHB-LITE is a multi-layer AHB, which enables parallel access paths between multiple masters and slaves in the system. Two masters on the AHB matrix, including AHB bus of the Arm® Cortex®-M23 core and DMA. The AHB matrix consists of four slaves, including the flash memory controller, internal SRAM, AHB1 and AHB2.

The AHB2 connects with the GPIO ports. The AHB1 connects with the AHB peripherals including AHB-to-APB bridge which provide full synchronous connections between the AHB1 and the APB bus. The APB bus connect with all the APB peripherals.

Figure 1-2. Series system architecture of GD32C2x1 series



1.3. Memory map

Program memory, data memory, registers and I/O ports are organized within the same linear 4-Gbyte address space which is the maximum address range of the Arm® Cortex®-M23 since it has a 32-bit bus address width. Additionally, a pre-defined memory map is provided by the Arm® Cortex®-M23 processor to reduce the software complexity of repeated implementation of different device vendors. However, some regions are used by the Arm® Cortex®-M23 system peripherals. The following figure shows the memory map of GD32C2x1 series, including Code, SRAM, peripheral, and other pre-defined regions. Each peripheral of either type is allocated 1KB of space. This allows simplifying the address decoding for each peripheral.

Table 1-2. Memory map of GD32C2x1 series

Pre-defined Regions	Bus	ADDRESS	Peripherals
		0xE000 0000 - 0xE00F FFFF	Cortex M23 internal peripherals
External Device		0xA000 0000 - 0xDFFF FFFF	Reserved
External RAM		0x60000000 - 0x9FFFFFFF	Reserved
Peripherals	AHB2	0x5004 0000 - 0x5FFF FFFF	Reserved
		0x5000 0000 - 0x5003 FFFF	Reserved
		0x4800 1800 - 0x4FFF FFFF	Reserved
		0x4800 1400 - 0x4800 17FF	GPIOF
		0x4800 1000 - 0x4800 13FF	Reserved
		0x4800 0C00 - 0x4800 0FFF	GPIOD
		0x4800 0800 - 0x4800 0BFF	GPIOC
		0x4800 0400 - 0x4800 07FF	GPIOB
		0x4800 0000 - 0x4800 03FF	GPIOA
	AHB1	0x4003 8400 - 0x47FF FFFF	Reserved
		0x4003 8000 - 0x4003 83FF	Reserved
		0x4002 4000 - 0x4003 7FFF	Reserved
		0x4002 3400 - 0x4002 3FFF	Reserved
		0x4002 3000 - 0x4002 33FF	CRC
		0x4002 2400 - 0x4002 2FFF	Reserved
		0x4002 2000 - 0x4002 23FF	FMC
		0x4002 1400 - 0x4002 1FFF	Reserved
		0x4002 1000 - 0x4002 13FF	RCU
		0x4002 0C00 - 0x4002 0FFF	Reserved
		0x4002 0800 - 0x4002 0BFF	DMAMUX
		0x4002 0400 - 0x4002 07FF	Reserved
		0x4002 0000 - 0x4002 03FF	DMA
	APB	0x4001 8000 - 0x4001 FFFF	Reserved
		0x4001 7C00 - 0x4001 7FFF	CMP
		0x4001 7800 - 0x4001 7BFF	Reserved

Pre-defined Regions	Bus	ADDRESS	Peripherals
		0x4001 7400 - 0x4001 77FF	Reserved
		0x4001 7000 - 0x4001 73FF	Reserved
		0x4001 6C00 - 0x4001 6FFF	Reserved
		0x4001 6800 - 0x4001 6BFF	Reserved
		0x4001 5C00 - 0x4001 67FF	Reserved
		0x4001 5800 - 0x4001 5BFF	DBG
		0x4001 5400 - 0x4001 57FF	Reserved
		0x4001 5000 - 0x4001 53FF	Reserved
		0x4001 4C00 - 0x4001 4FFF	Reserved
		0x4001 4800 - 0x4001 4BFF	TIMER16
		0x4001 4400 - 0x4001 47FF	TIMER15
		0x4001 3C00 - 0x4001 43FF	Reserved
		0x4001 3800 - 0x4001 3BFF	USART0
		0x4001 3400 - 0x4001 37FF	Reserved
		0x4001 3000 - 0x4001 33FF	SPI0/I2S0
		0x4001 2C00 - 0x4001 2FFF	TIMER0
		0x4001 2800 - 0x4001 2BFF	Reserved
		0x4001 2400 - 0x4001 27FF	ADC
		0x4001 2000 - 0x4001 23FF	Reserved
		0x4001 1C00 - 0x4001 1FFF	Reserved
		0x4001 1800 - 0x4001 1BFF	Reserved
		0x4001 1400 - 0x4001 17FF	Reserved
		0x4001 1000 - 0x4001 13FF	Reserved
		0x4001 0C00 - 0x4001 0FFF	Reserved
		0x4001 0800 - 0x4001 23FF	Reserved
		0x4001 0400 - 0x4001 07FF	EXTI
		0x4001 0000 - 0x4001 03FF	SYSCFG
		0x4000 C000 - 0x4000 FFFF	Reserved
		0x4000 7400 - 0x4000 BFFF	Reserved
		0x4000 7000 - 0x4000 73FF	PMU
		0x4000 5C00 - 0x4000 6FFF	Reserved
		0x4000 5800 - 0x4000 5BFF	I2C1
		0x4000 5400 - 0x4000 57FF	I2C0
		0x4000 4C00 - 0x4000 53FF	Reserved
		0x4000 4800 - 0x4000 4BFF	USART2
		0x4000 4400 - 0x4000 47FF	USART1
		0x4000 3C00 - 0x4000 43FF	Reserved
		0x4000 3800 - 0x4000 3BFF	SPI1
		0x4000 3400 - 0x4000 37FF	Reserved
		0x4000 3000 - 0x4000 33FF	FWDGT
		0x4000 2C00 - 0x4000 2FFF	WWDGT

Pre-defined Regions	Bus	ADDRESS	Peripherals
		0x4000 2800 - 0x4000 2BFF	RTC
		0x4000 2400 - 0x4000 27FF	Reserved
		0x4000 2000 - 0x4000 23FF	TIMER13
		0x4000 0800 - 0x4000 1FFF	Reserved
		0x4000 0400 - 0x4000 07FF	TIMER2
		0x4000 0000 - 0x4000 03FF	Reserved
SRAM		0x2000 3000 - 0x3FFF FFFF	Reserved
		0x2000 0000 - 0x2000 2FFF	SRAM(12KB)
Code		0x1FFF 7880 - 0x1FFF FFFF	Reserved
		0x1FFF 7800 - 0x1FFF 787F	Option bytes(128B)
		0x1FFF 7400 - 0x1FFF 77FF	Reserved
		0x1FFF 7000 - 0x1FFF 73FF	OTP bytes(1KB)
		0x1FFF 0C00 - 0x1FFF 6FFF	Reserved
		0x1FFF 0000 - 0x1FFF 0BFF	System memory(3KB)
		0x0801 0000 - 0x1FFE FFFF	Reserved
		0x0800 0000 - 0x0800 FFFF	Flash memory(64KB)
		0x0000 0000 - 0x07FF FFFF	Aliased to Flash or system memory

1.3.1. On-chip SRAM memory

The GD32C2x1 series contain up to 12KB of on-chip SRAM which starts at the address 0x2000 0000. It supports byte, half-word (16 bits), and word (32 bits) accesses.

ECC

When reading and writing SRAM, it supports 7-bit ECC function. It can correct 1 bit error and detect multiple bits (two bits) error.

It must be written before reading SRAM, otherwise it may cause ECC error. Unaligned read operations will be performed in accordance with 32-bit read operations. Non-aligned write operations will produce a read-modify-write process. For example, when 16-bit data is written into SRAM, firstly the another 16-bit data is read out from the SRAM, and the 16-bit that need to be written are combined to a 32-bit data, and finally the 32-bit data is written into the SRAM together. Therefore, when initializing SRAM, it can only be written in a 32-bit width.

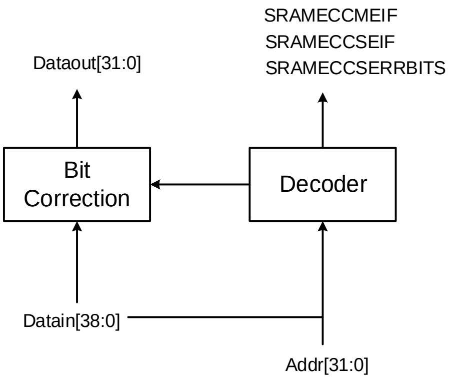
The ECC module is composed of an encoder and a decoder.

Encoder: When performing a SRAM write operation, a 7-bit ECC code will be generated and written into the SRAM together with the data.

Decoder: When performing a SRAM read operation, it uses the same algorithm as the encoder to decode and generate a 7-bit ECC code. The ECC code includes ECC error status and information which specific bit of the 32-bit data has single bit error.

The decoder is shown in the figure ***Figure 1-3. ECC decoder*** below:

Figure 1-3. ECC decoder



EEIC

The EEIC (ECC Error Interrupt Control) module provides the function of ECC error status management and ECC interrupt configuration.

Enable SRAM_ECCEN in Option byte and then by setting the SYSCFG_CFG1 register can realize the ECC error detection of SRAM.

Single bit error correction event

When a single-bit error correction event is detected in SRAM, EEIC：

(1) The ECCSEIF bit in SYSCFG_STAT register will be set. Software can clear it by writing 1.

(2) The ECCEADDR[11:0] bits in SYSCFG_CFG2 register records the address where the single-bit error correction event occurred.

Two bits non-correction error event

When a two bits non-correction error event is detected in SRAM，EEIC：

(1) The ECCMEIF bit in SYSCFG_STAT register will be set. Software can clear it by writing 1.

(2) The ECCEADDR[11:0] bits in SYSCFG_CFG2 records the address where the two bits non-correction error event occurred.

Single bit error correction interrupt

Set the ECCSEIE bit in SYSCFG_CFG2 register. When a single-bit error correctable event is detected, a NMI interrupt will be generated.

Two bits non-correction error interrupt

Set the ECCMEIE bit in SYSCFG_CFG2 register. When a two bits error non-correction event is detected, a NMI interrupt will be generated.

1.3.2. On-chip Flash memory

The devices provide high-density on-chip flash memory, which is structured as follows:

- Up to 64KB of main Flash memory and 2KB data Flash memory.
- Option bytes to configure the device.

Refer to ***Flash memory controller (FMC)*** for more details.

1.4. Boot configuration

Flash empty check

When configuring the BOOT0 pin for startup from the main Flash, programming the original device becomes more convenient by checking the MFPE bit in the FMC_WS register. When the MFPE bit is set, the device is considered empty, and the bootloader starts from the System memory, allowing Flash programming at this point. During the loading of option bytes, if the content at address 0x0800 0000 is read as 0xFFFF FFFF, the MFPE flag will be set; otherwise, it will be cleared. After programming the original device, the MFPE flag can be cleared by either a power reset or setting the OBRLD bit in the FMC_CTL register, enabling the device to execute user code after a system reset. The MFPE flag can also be modified by software.

Note: If the device is programmed for the first time but option bytes are not reloaded, the device will still choose System memory as the boot area after a system reset.

Boot mode

The GD32C2x1 series provide three kinds of boot sources which can be selected by the BOOT0 pin and boot configuration bits BOOTLK, nBOOT1, SWBT0 and nBOOT0 in the User option byte. The details are shown in the following table. When SWBT0=0, the value on the BOOT0 pin is latched on the 4th rising edge of CK_SYS after a reset. It is up to the user to set the boot mode configuration after a power-on reset or a system reset to select the required boot source. When SWBT0=0, once the boot mode configuration have been sampled, they are free and can be used for other purposes.

Table 1-3. Boot modes

Selected boot area	Boot mode configuration				
	BOOTLK	nBOOT1 bit	BOOT0 pin	SWBT0 bit	nBOOT0 bit
Main Flash memory	0	x	0	0	x
System memory	0	1	1	0	x
Embedded SRAM	0	0	1	0	x
Main Flash memory	0	x	x	1	1
System memory	0	1	x	1	0
Embedded SRAM	0	0	x	1	0
Main Flash memory	1	x	x	x	x

After power-on sequence or a system reset, the Arm® Cortex®-M23 processor fetches the topof-stack value from address 0x0000 0000 and the base address of boot code from 0x0000 0004 in sequence. Then, it starts executing code from the base address of boot code.

According to the selected boot source, either the main flash memory (original memory space beginning at 0x0800 0000) or the system memory (original memory space beginning at 0x1FFF 0000) is aliased in the boot memory space which begins at the address 0x0000 0000. When the on-chip SRAM whose memory space is beginning at 0x2000 0000 is selected as the boot source, in the application initialization code, you have to relocate the vector table in SRAM using the NVIC exception table and offset register.

The embedded boot loader is located in the System memory, which is used to reprogram the Flash memory. For GD32C2x1 series, the boot loader can be activated through one of the following interfaces: USART0/1, I2C0.

Ability to use the BOOT_LOCK bit to force boot from a unique entry point in the Main Flash memory regardless of others mode configuration.

- Remap configuration

  When the boot mode configurations are selected, the software can configure the memory remap function through BOOT_MODE bits in memory remap configuration register (SYSCFG_CFG0). Memories including main flash memory, system memory, embedded SRAM can be remapped.

1.5. System configuration controller (SYSCFG)

The main purposes of the system configuration controller (SYSCFG) are the following:

- Enabling/disabling I2C Fast Mode Plus on some I/O ports
- Remapping of some I/O ports
- Managing the external interrupt line connection to the GPIOs

1.6. System configuration registers

SYSCFG base address: 0x4001 0000

1.6.1. System configuration register 0 (SYSCFG_CFG0)

Address offset: 0x00

Reset value: 0x0000 000X (X indicates BOOT_MODE[1:0] may be any value according to the BOOT0 pin and boot configuration bits nBOOT1, SWBT0 and nBOOT0 in the user option byte after reset)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved							PC14FMPEN	PA10FMPEN	PA9FMPEN	Reserved	I2C0_FMP	PB9FMPEN	PB8FMPEN	PB7FMPEN	PB6FMPEN

							rw	rw	rw		rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved											PA12_RMP	PA11_RMP	Reserved	BOOT_MODE[1:0]	
											rw	rw		rw	

Bits	Fields	Descriptions
31:25	Reserved	Must be kept at reset value
24	PC14FMPEN	I2C Fm+ mode on PC14 pin enable This bit controls I2C Fm+ mode, the speed control of the pin is bypassed. 0: Disable Fm+ mode on the PC14 pin 1: Enable Fm+ mode on the PC14 pin
23	PA10FMPEN	I2C Fm+ mode on PA10 pin enable This bit controls I2C Fm+ mode, the speed control of the pin is bypassed. 0: Disable Fm+ mode on the PA10 pin 1: Enable Fm+ mode on the PA10 pin
22	PA9FMPEN	I2C Fm+ mode on PA9 pin enable This bit controls I2C Fm+ mode, the speed control of the pin is bypassed. 0: Disable Fm+ mode on the PA9 pin 1: Enable Fm+ mode on the PA9 pin
21	Reserved	Must be kept at reset value
20	I2C0_FMP	Fast Mode Plus (FM+) enable for I2C0 This bit is set and cleared by software. It enables I2C FM+ driving capability on I/O ports configured as I2C0 through GPIOx_AFSELx registers. 0: Disable 1: Enable With this bit in disable state, the I2C FM+ driving capability on I/O ports configured as I2C0 can be enabled through their corresponding I2Cx_FMP bit. When I2C FM+ is enabled, the speed control is ignored.
19	PB9FMPEN	I2C Fm+ mode on PB9 pin enable This bit controls I2C Fm+ mode, the speed control of the pin is bypassed. 0: Disable Fm+ mode on the PB9 pin 1: Enable Fm+ mode on the PB9 pin
18	PB8FMPEN	I2C Fm+ mode on PB8 pin enable This bit controls I2C Fm+ mode, the speed control of the pin is bypassed. 0: Disable Fm+ mode on the PB8 pin 1: Enable Fm+ mode on the PB8 pin
17	PB7FMPEN	I2C Fm+ mode on PB7 pin enable This bit controls I2C Fm+ mode, the speed control of the pin is bypassed. 0: Disable Fm+ mode on the PB7 pin

		1: Enable Fm+ mode on the PB7 pin
16	PB6FMPEN	I2C Fm+ mode on PB6 pin enable This bit controls I2C Fm+ mode, the speed control of the pin is bypassed. 0: Disable Fm+ mode on the PB6 pin 1: Enable Fm+ mode on the PB6 pin
15:5	Reserved	Must be kept at reset value
4	PA12_RMP	PA12 pin remapping This bit is set and cleared by software. When set, it remaps the PA12 pin to operate as PA10 GPIO port, instead as PA12 GPIO port. 0: No remap (PA12) 1: Remap (PA10)
3	PA11_RMP	PA11 pin remapping This bit is set and cleared by software. When set, it remaps the PA11 pin to operate as PA9 GPIO port, instead as PA11 GPIO port. 0: No remap (PA11) 1: Remap (PA9)
2	Reserved	Must be kept at reset value
1:0	BOOT_MODE[1:0]	Boot mode (Refer to ***Boot configuration*** for details) Bit0 is mapping to the BOOT0 pin; the value of bit1 is the nBOOT1 bit value. x0: Boot from the main flash 01: Boot from the system flash memory 11: Boot from the embedded SRAM

1.6.2. EXTI sources selection register 0 (SYSCFG_EXTISS0)

Address offset: 0x08

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved		EXTI3_SS [1:0]		Reserved		EXTI2_SS [1:0]		Reserved		EXTI1_SS [1:0]		Reserved		EXTI0_SS [1:0]	
		rw				rw				rw				rw	

Bits	Fields	Descriptions
31:14	Reserved	Must be kept at reset value
13:12	EXTI3_SS[1:0]	EXTI 3 sources selection 00: PA3 pin 01: PB3 pin 10: PD3 pin

		11: PF3 pin
11:10	Reserved	Must be kept at reset value
9:8	EXTI2_SS[1:0]	EXTI 2 sources selection 00: PA2 pin 01: PB2 pin 10: PD2 pin 11: PF2 pin
7:6	Reserved	Must be kept at reset value
5:4	EXTI1_SS[1:0]	EXTI 1 sources selection 00: PA1 pin 01: PB1 pin 10: PD1 pin 11: PF1 pin
3:2	Reserved	Must be kept at reset value
1:0	EXTI0_SS[1:0]	EXTI 0 sources selection 00: PA0 pin 01: PB0 pin 10: PD0 pin 11: PF0 pin

1.6.3. EXTI sources selection register 1 (SYSCFG_EXTISS1)

Address offset: 0x0C

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved		EXTI7_SS [1:0]		Reserved		EXTI6_SS [1:0]		Reserved			EXTI5_SS	Reserved			EXTI4_SS
		rw				rw					rw				rw

Bits	Fields	Descriptions
31:14	Reserved	Must be kept at reset value
13:12	EXTI7_SS[1:0]	EXTI 7 sources selection 00: PA7 pin 01: PB7 pin 10: PC7 pin 11: reserved
11:10	Reserved	Must be kept at reset value

9:8	EXTI6_SS[1:0]	EXTI 6 sources selection 00: PA6 pin 01: PB6 pin 10: PC6 pin 11: reserved
7:6	Reserved	Must be kept at reset value
5:4	EXTI5_SS	EXTI 5 sources selection 0: PA5 pin 1: PB5 pin
3:1	Reserved	Must be kept at reset value
0	EXTI4_SS	EXTI 4 sources selection 0: PA4 pin 1: PB4 pin

1.6.4. EXTI sources selection register 2 (SYSCFG_EXTISS2)

Address offset: 0x10

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved			EXTI11_SS	Reserved			EXTI10_SS	Reserved			EXTI9_SS	Reserved			EXTI8_SS
			rw				rw				rw				rw

Bits	Fields	Descriptions
31:13	Reserved	Must be kept at reset value
12	EXTI11_SS	EXTI 11 sources selection 0: PA11 pin 1: PB11 pin
11:9	Reserved	Must be kept at reset value
8	EXTI10_SS	EXTI 10 sources selection 0: PA10 pin 1: PB10 pin
7:5	Reserved	Must be kept at reset value
4	EXTI9_SS	EXTI 9 sources selection 0: PA9 pin

		1: PB9 pin
3:1	Reserved	Must be kept at reset value
0	EXTI8_SS	EXTI 8 sources selection 0: PA8 pin 1: PB8 pin

1.6.5. EXTI sources selection register 3 (SYSCFG_EXTISS3)

Address offset: 0x14

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved		EXTI15_SS [1:0]		Reserved		EXTI14_SS [1:0]		Reserved		EXTI13_SS [1:0]		Reserved			EXTI12_SS
		rw				rw				rw					rw

Bits	Fields	Descriptions
31:14	Reserved	Must be kept at reset value
13:12	EXTI15_SS[1:0]	EXTI 15 sources selection 00: PA15 pin 01: PB15 pin 10: PC15 pin 11: reserved
11:10	Reserved	Must be kept at reset value
9:8	EXTI14_SS[1:0]	EXTI 14 sources selection 00: PA14 pin 01: PB14 pin 10: PC14 pin 11: reserved
7:6	Reserved	Must be kept at reset value
5:4	EXTI13_SS[1:0]	EXTI 13 sources selection 00: PA13 pin 01: PB13 pin 10: PC13 pin 11: reserved
3:1	Reserved	Must be kept at reset value
0	EXTI12_SS	EXTI 12 sources selection

		0: PA12 pin 1: PB12 pin

1.6.6. System configuration register 1 (SYSCFG_CFG1)

Address offset: 0x18

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved														SRAM_ECC_LOCK	LOCKUP_LOCK
														rw	rw

Bits	Fields	Descriptions
31:2	Reserved	Must be kept at reset value
1	SRAM_ECC_LOCK	SRAM ECC check error lock This bit is set by software and cleared by a system reset. 0: The SRAM ECC check error is disconnected from the break input of TIMER0/15/16 1: The SRAM ECC check error is connected from the break input of TIMER0/15/16.
0	LOCKUP_LOCK	Cortex-M23 LOCKUP output lock This bit is set by software and cleared by a system reset. 0: The Cortex-M23 LOCKUP output is disconnected from the break input of TIMER0/15/16. 1: The Cortex-M23 LOCKUP output is connected from the break input of TIMER0/15/16.

1.6.7. System status register (SYSCFG_STAT)

Address offset: 0x1C

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved														ECCSEIF	ECCMEIF
														rc_w1	rc_w1

Bits	Fields	Descriptions

31:2	Reserved	Must be kept at reset value
1	ECCSEIF	SRAM single bit correction event flag The software can clear it by writing 0. 0: No SRAM single bit correction event is detected. 1: SRAM single bit correction event is detected.
0	ECCMEIF	SRAM two bits non-correction event flag The software can clear it by writing 1. 0: No SRAM non-correction event is detected. 1: SRAM non-correction event is detected.

1.6.8. SYSCFG configuration register 2 (SYSCFG_CFG2)

Address offset: 0x28

Reset value: 0x0000 0007

A NMI interrupt is generated when a NMI event occurs and the interrupt signal is enabled.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ECCEADDR[11:0]												Reserved			
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ECCSERRBITS[5:0]						Reserved					ECCSEIE	ECCMEIE	HXTALCS _IE	LXTALCS_ IE	Reserved
r											rw	rw	rw	rw	

Bits	Fields	Descriptions
31:20	ECCEADDR[11:0]	Indicates the last ECC event on SRAM occurred.
19:16	Reserved	Must be kept at reset value.
15:10	ECCSERRBITS[5:0]	Indicates the error bit Which one bit has an ECC single-bit correctable error 0: no error 1: bit 0 … 31: bit 31
9:5	Reserved	Must be kept at reset value.
4	ECCSEIE	Single bit correction error interrupt enable This bit can be used to enable single bit correction event flag connection to NVIC. 0: Disable. 1: Enable.
3	ECCMEIE	Multi-bits (two bits) non-correction error NMI interrupt enable This bit can be used to enable two bits non-correction event flag

		0: Disable. 1: Enable.
2	HXTALCS_IE	HXTAL clock stuck interrupt (NMI / EXTI) enable 0:Disable. 1:Enable.
1	LXTALCS_IE	LXTAL clock stuck interrupt (NMI / EXTI) enable 0:Disable. 1:Enable.
0	Reserved	Must be kept at reset value.

1.6.9. IRQ latency register (SYSCFG_CPU_IRQ_LAT)

Address offset: 0x100

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								IRQ_LATENCY[7:0]							
								rw							

Bits	Fields	Descriptions
31:8	Reserved	Must be kept at reset value
7:0	IRQ_LATENCY[7:0]	IRQ_LATENCY specifies the minimum number of cycles between an interrupt that becomes pended in the NVIC, and the vector fetch for that interrupt being issued on the AHB-Lite interface. If IRQ_LATENCY is set to 0, interrupts are taken as quickly as possible. For non-zero values, the Cortex-M23 processor ensures that a minimum of IRQ_LATENCY+1 HCLK cycles exist between an interrupt becoming pended in the NVIC and the vector fetch for the interrupt being performed.

1.6.10. TIMERx configuration register 0 (SYSCFG_TIMERxCFG0, x=0, 2)

Address offset: 0x110 for TIMER0

Address offset: 0x118 for TIMER2

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved	TSCFG7[2:0]			Reserved	TSCFG6[2:0]			Reserved	TSCFG5 [2:0]			Reserved	TSCFG4 [2:0]		

		rw				rw				rw				rw	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	TSCFG3 [2:0]			Reserved	TSCFG2 [2:0]			Reserved	TSCFG1 [2:0]			Reserved	TSCFG0 [2:0]		
		rw				rw				rw				rw	

Bits	Fields	Descriptions
31	Reserved	Must be kept at reset value
30:28	TSCFG7[2:0]	Restart event mode configuration The counter is reinitialized and started, the shadow registers are updated on the rising edge of the selected trigger input when these bits are not 0. 000: Restart + event mode disable 001: Internal trigger input 0 (ITI0) 010: Internal trigger input 2 (ITI2) 011: Internal trigger input 3 (ITI3) 100: CI0 edge flag (CI0F_ED) 101: The filtered output of channel 0 input (CI0FE0) 110: The filtered output of channel 1 input (CI1FE1) 111: The filtered output of external trigger input (ETIFP)
27	Reserved	Must be kept at reset value
26:24	TSCFG6[2:0]	External clock mode 0 configuration The counter counts on the rising edges of the selected trigger when these bits are not 0. 000: External clock mode 0 disable 001: Internal trigger input 0 (ITI0) 010: Internal trigger input 2 (ITI2) 011: Internal trigger input 3 (ITI3) 100: CI0 edge flag (CI0F_ED) 101: The filtered output of channel 0 input (CI0FE0) 110: The filtered output of channel 1 input (CI1FE1) 111: The filtered output of external trigger input (ETIFP)
23	Reserved	Must be kept at reset value
22:20	TSCFG5[2:0]	Event mode configuration A rising edge of the trigger input enables the counter. 000: Event mode disable 001: Internal trigger input 0 (ITI0) 010: Internal trigger input 2 (ITI2) 011: Internal trigger input 3 (ITI3) 100: CI0 edge flag (CI0F_ED) 101: The filtered output of channel 0 input (CI0FE0) 110: The filtered output of channel 1 input (CI1FE1) 111: The filtered output of external trigger input (ETIFP)

19	Reserved	Must be kept at reset value
18:16	TSCFG4[2:0]	Pause mode configuration The trigger input enables the counter clock when it is high and disables the counter when it is low when these bits are not 0. 000: Pause mode disable 001: Internal trigger input 0 (ITI0) 010: Internal trigger input 2 (ITI2) 011: Internal trigger input 3 (ITI3) 100: Reserved 101: The filtered output of channel 0 input (CI0FE0) 110: The filtered output of channel 1 input (CI1FE1) 111: The filtered output of external trigger input (ETIFP)
15	Reserved	Must be kept at reset value
14:12	TSCFG3[2:0]	Restart mode configuration The counter is reinitialized and the shadow registers are updated on the rising edge of the selected trigger input when these bits are not 0. 000: Restart mode disable 001: Internal trigger input 0 (ITI0) 010: Internal trigger input 2 (ITI2) 011: Internal trigger input 3 (ITI3) 100: CI0 edge flag (CI0F_ED) 101: The filtered output of channel 0 input (CI0FE0) 110: The filtered output of channel 1 input (CI1FE1) 111: The filtered output of external trigger input (ETIFP)
11	Reserved	Must be kept at reset value
10:8	TSCFG2[2:0]	Quadrature decoder mode 2 configuration 000: Quadrature decoder mode 2 disable Others: The counter counts on both CI0FE0 and CI1FE1 edges, while the direction depends on each other
7	Reserved	Must be kept at reset value
6:4	TSCFG1[2:0]	Quadrature decoder mode 1 configuration 000: Quadrature decoder mode 1 disable Others: The counter counts on CI1FE1 edge, while the direction depends on CI0FE0 level
3	Reserved	Must be kept at reset value
2:0	TSCFG0[2:0]	Quadrature decoder mode 0 configuration 000: Quadrature decoder mode 0 disable Others: The counter counts on CI0FE0 edge, while the direction depends on CI1FE1 level.

1.6.11. TIMERx configuration register 1 (SYSCFG_TIMERxCFG1, x=0, 2)

Address offset: 0x114 for TIMER0
Address offset: 0x11C for TIMER2
Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved													TSCFG8[2:0]		
													rw		

Bits	Fields	Descriptions
31:3	Reserved	Must be kept at reset value
2:0	TSCFG8[2:0]	Internal trigger input source configuration 000: Reserved 001: Internal trigger input 0 (ITI0) 010: Internal trigger input 2 (ITI2) 011: Internal trigger input 3 (ITI3) 100: CI0 edge flag (CI0F_ED)

1.7. Device electronic signature

The device electronic signature contains memory density information and the 96-bit unique device ID. It is stored in the information block of the Flash memory. The 96-bit unique device ID is unique for any device. It can be used as serial numbers, or part of security keys, etc.

1.7.1. Memory density information

Base address: 0x1FFF 0BE0
The value is factory programmed and can never be altered by user.

This register has to be accessed by word(32-bit)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SRAM_DENSITY[15:0]															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FLASH_DENSITY[15:0]															
r															

Bits	Fields	Descriptions
31:16	SRAM_DENSITY [15:0]	SRAM density The value indicates the on-chip SRAM density of the device in Kbytes. Example: 0x0008 indicates 8 Kbytes.
15:0	FLASH_DENSITY [15:0]	Flash memory density The value indicates the Flash memory density of the device in Kbytes. Example: 0x0020 indicates 32 Kbytes.

1.7.2. Unique device ID (96 bits)

Base address: 0x1FFF 0BE8

The value is factory programmed and can never be altered by user.

This register has to be accessed by word(32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
UNIQUE_ID[31:16]															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
UNIQUE_ID[15:0]															
r															

Bits	Fields	Descriptions
31:0	UNIQUE_ID[31:0]	Unique device ID

Base address: 0x1FFF 0BEC

The value is factory programmed and can never be altered by user.

This register has to be accessed by word(32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
UNIQUE_ID[63:48]															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
UNIQUE_ID[47:32]															
r															

Bits	Fields	Descriptions
31:0	UNIQUE_ID[63:32]	Unique device ID

Base address: 0x1FFF 0BF0

The value is factory programmed and can never be altered by user.

This register has to be accessed by word(32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
UNIQUE_ID[95:80]															

r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
UNIQUE_ID[79:64]															
r															

Bits	Fields	Descriptions
31:0	UNIQUE_ID[96:64]	Unique device ID

2. Flash memory controller (FMC)

2.1. Overview

The flash memory controller, FMC, provides all the necessary functions for the on-chip flash memory. A little waiting time is needed while CPU executes instructions stored in the main flash. It also provides page erase, mass erase, program operations, read and write protection, and security mechanisms for flash memory.

2.2. Characteristics

- Up to 64KB of main flash memory for instruction and data. Up to 1KB OTP. 3KB information block.
  - Main flash block: 64KB
  - Information block: 3KB
  - OTP block: 1KB
  - Option bytes block: 128B
- 0~1 waiting time when CPU executes instructions and read data.
- Pre-fetch buffer to speed read operations.
- Instruction Cache: 2 cache lines of 64 bits on ICode (16-byte RAM)
- Double word programming, page erase and mass erase operation.
- 1KB OTP (one-time program) block used for user data storage.
- 128B option bytes block for user application requirements.
- Option bytes are uploaded to the option byte control registers when the system is reset.
- Flash security protection to prevent illegal code / data access.
- Provides two execute-only dedicated code read protection (DCRP) area.
- Provides page erase / program protection (WP) function.
- Provides secure user area (SCR).

2.3. Function overview

2.3.1. Flash memory architecture

The flash memory consists of up to 64 KB main flash, which is organized into 1 x 64 pages with 1KB capacity, 3 KB information block for the boot loader. Each page of main flash memory can be erased individually. ***Table 2-1. Base address and size of flash memory*** shows the base address and size of flash memory.

Table 2-1. Base address and size of flash memory

Block	Name	Address	size(bytes)
Main Flash Block	Page 0	0x0800 0000 - 0x0800 03FF	1KB

Block	Name	Address	size(bytes)
	Page 1	0x0800 0400 - 0x0800 07FF	1KB
	Page 2	0x0800 0800 - 0x0800 0BFF	1KB
	. . .	. . .	. . .
	Page 63	0x0800 FC00 - 0x0800 FFFF	1KB
Information Block	Bootloader[1]	0x1FFF 0000 - 0x1FFF 0BFF	3KB
Option byte Block	Option bytes	0x1FFF 7800 - 0x1FFF 78FF	128B
One-time program Block	OTP bytes[2]	0x1FFF 7000~0x1FFF 73FF	1KB

Note: 1. The Information Block stores the boot loader. This block cannot be programmed or erased by user.

2. The OTP (one-time programmable) has 1 KB (128 double word) to be written by user. The OTP data can be written only once and cannot be erased. If any bit has been set 0, the whole double word cannot be written anymore.

2.3.2. Main flash empty check

The flash interface provides Main flash empty check mechanism, which is achieved by checking whether the first location of Main flash is programmed. And the result of this empty check status can be read from the MFPE bit in FMC_WS register. In conjunction with the boot0 and boot1 information which is used to determine where the system has to boot from. It prevents the system to boot from Main flash memory area when no user code has been programmed. Software can modify the Main flash memory empty status by writing an appropriate value to MFPE bit in FMC_WS register.

2.3.3. Read operations

The flash can be addressed directly as a common memory space. Any instruction fetches and the data access from the flash are through CBUS.

Wait state added

The WSCNT bits in the FMC_WS register needs to be configured correctly depend on the AHB clock frequency when reading the flash memory. The relation between WSCNT and AHB clock frequency is show as the ***Table 2-2. The relation between WSCNT and AHB clock frequency*** ,

Table 2-2. The relation between WSCNT and AHB clock frequency

AHB clock frequency	WSCNT configured
<= 24MHz	0 (0 wait state added)
<= 48MHz	1 (1 wait state added)

If system reset occurs, the AHB clock frequency is 12MHz and the WSCNT is 0.

Note:

1. If it is necessary to increase the AHB clock frequency, firstly, configure the WSCNT bits according to the target AHB clock frequency. Then, increase the AHB clock frequency to the target frequency. It is forbidden to increase the AHB clock frequency before configuring the WSCNT.

2. If it is necessary to decrease the AHB clock frequency, firstly, decrease the target AHB clock frequency. Then configure the WSCNT bits according the target AHB clock frequency. It is forbidden to configure the WSCNT bits before decrease the AHB clock frequency.

Considering that the wait state is added, the read efficiency is low (such as add 1 wait state when 48MHz). In order to speed up the read access, there are some functions performed as below.

Current buffer

The current buffer is always enabled. Each time read from flash memory, 64-bit data will be get and store in current buffer. The CPU only need 32-bit or 16-bit buffer in each read operation. So in the case of sequential code, the next data can get from current buffer without repeat fetch from flash memory.

Pre-fetch buffer

The pre-fetch buffer is enabled by setting the PFEN bit in the FMC_WS register. In the case of sequential code, when CPU executes the current buffer data (64-bit), 32-bit needs at least 2 clocks and 16-bit needs at least 4 clocks. In this case, pre-fetch the data of next double-word address from flash memory and store to Pre-fetch buffer. So when the CPU finishes the current buffer and needs execute the next data, the pre-fetch buffer hits.

ICODE cache

Instruction cache is enabled by set the ICEN bit in the FMC_WS register. The cache has 16 bytes, and it is organized by 2 cache lines. Each cache line has 64 bits.

If the data is in the instruction cache (cache hit), the CPU read data from cache without any wait state. If the data is not in the instruction cache (cache miss), nor in current buffer / pre-fetch buffer, the cache line fetch data from the flash memory and copied it into the instruction cache. If all cache line filled, LRU (least recently used) policy is used to replace the cache line.

2.3.4. Unlock the FMC_CTL register and FMC_OBCTL register

After reset, the FMC_CTL register is not accessible in write mode, and the LK bit in FMC_CTL register is 1. An unlocking sequence consists of two write operations to the FMC_KEY register can open the access to the FMC_CTL register. The two write operations are writing 0x45670123 and 0xCDEF89AB to the FMC_KEY register. After the two write operations, the LK bit in FMC_CTL register is cleared by hardware. The software can lock the FMC_CTL again by setting the LK bit in FMC_CTL register. If there are some wrong operations on the FMC_KEY register, the LK bit in FMC_CTL register will be set, and the FMC_CTL register will be locked, then it will generate a bus error.

The FMC_OBCTL register, OBRLD bit and OBSTART bit in FMC_CTL is also protected by FMC_OBKEY register. The unlocking sequence includes two write operations, which are orderly writing 0x0819 2A3B and 0x4C5D 6E7F to FMC_OBKEY register, then hardware reset the OBLK bit in FMC_CTL register to 0. The software can set OBLK bit to 1 to protect the FMC_OBCTL register, OBRLD bit and OBSTART bit in FMC_CTL register again

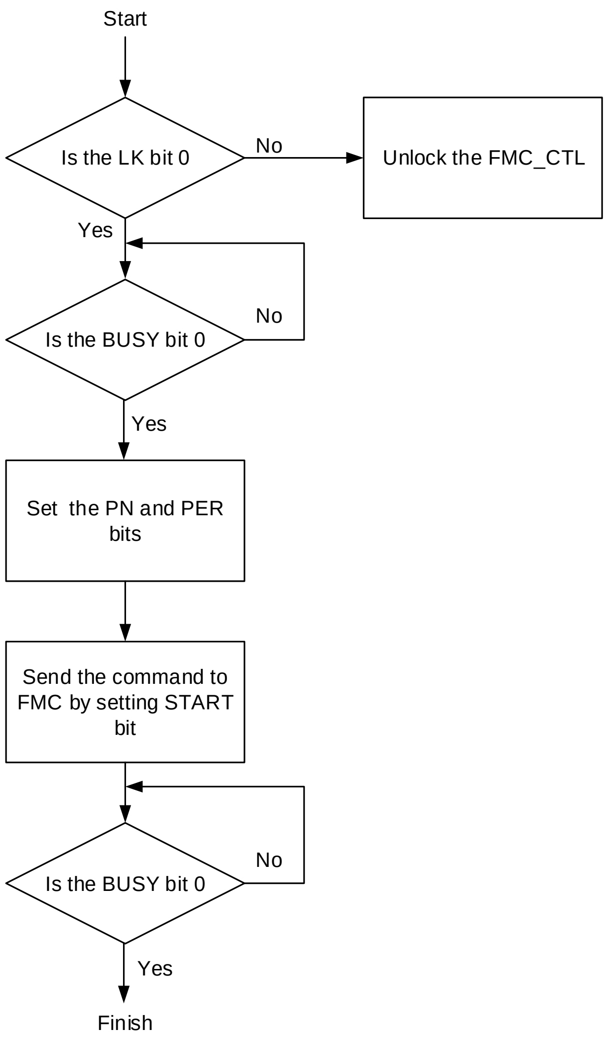
2.3.5. Page erase

The FMC provides a page erase function which is used for initializing the contents of a main flash memory page to a high state. 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.

- Unlock the FMC_CTL register if necessary.
- Check the BUSY bit in FMC_STAT register to confirm that no flash memory operation is in progress (BUSY equal to 0). Otherwise, wait until the operation has been finished.
- Select the page to erase (PN).
- Write the page erase command into PER bit in FMC_CTL register.
- Send the START command to the FMC by setting the START bit in FMC_CTL register.
- Wait until all the operations have been completed by checking the value of the BUSY bit in FMC_STAT register.
- Read and verify the page if required using a CBUS access.

After this operation is successfully executed, the ENDF in FMC_STAT register will be set. An interrupt will be triggered by FMC if the ENDIE bit in the FMC_CTL register is set. Note that a correct target page number must be confirmed. Otherwise, the software may run out of control if the target erase page is being used for fetching codes or accessing data. The FMC will not provide any notification when it occurs. Additionally, the page erase operation will be ignored on protected pages. Flash operation error interrupt will be triggered by the FMC if the ERRIE bit in the FMC_CTL register is set. The software can check the WPERR bit in the FMC_STAT register to detect this condition in the interrupt handler. The ***Figure 2-1. Process of page erase operation*** shows the page erase operation flow.

Figure 2-1. Process of page erase operation



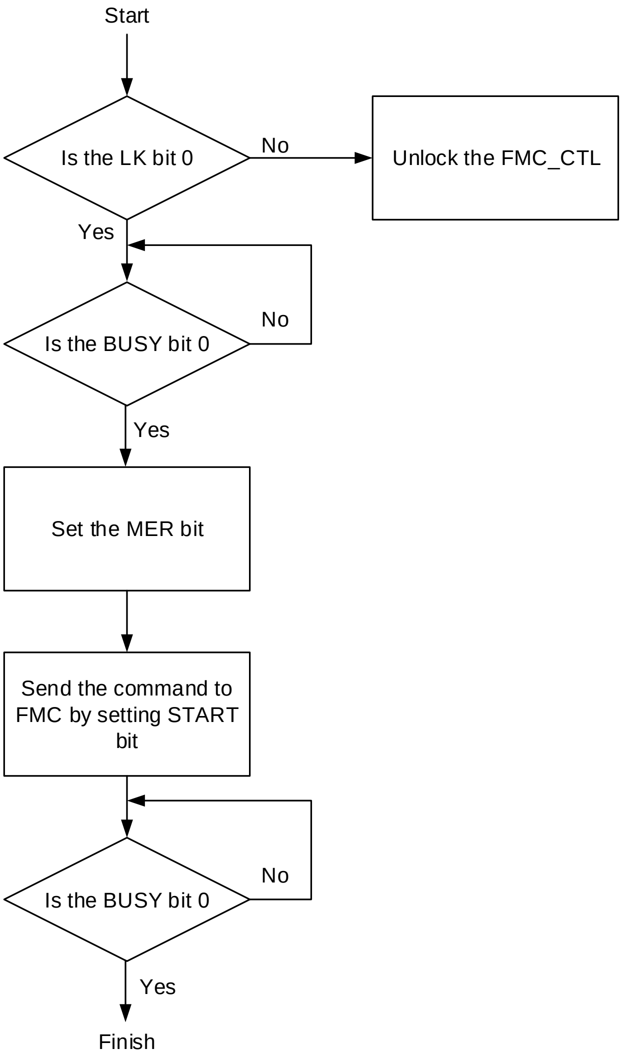
2.3.6. Mass erase

The FMC provides a complete erase function which is used for initializing the main flash block contents. The following steps show the mass erase register access sequence.

- Unlock the FMC_CTL register if necessary.
- Check the BUSY bit in FMC_STAT register to confirm that no flash memory operation is in progress (BUSY equal to 0). Otherwise, wait until the operation has finished.
- Write the mass erase command into MER bit in FMC_CTL register.
- Send the mass erase command to the FMC by setting the START bit in FMC_CTL register.
- Wait until all the operations have finished by checking the value of the BUSY bit in FMC_STAT register.
- Read and verify the flash memory if required using a SBUS access.

When the operation is successfully executed, the ENDF in FMC_STAT register is set, and an interrupt will be triggered by FMC if the ENDIE bit in the FMC_CTL register is set. Since all flash data will be reset to a value of 0xFFFF FFFF, the mass erase operation can be implemented using a program that runs in SRAM or by using the debugging tool to access the FMC registers directly. Additionally, the mass erase operation will be ignored if any page is erase / program protected. In this condition, a flash operation error interrupt will be triggered by the FMC if the ERRIE bit in the FMC_CTL register is set. The software can check the WPERR bit in the FMC_STAT register to detect this condition in the interrupt handler. The ***Figure 2-2. Process of mass erase operation*** indicates the mass erase operation flow.

Figure 2-2. Process of mass erase operation



2.3.7. Main flash programming

The FMC provides a double-word programming (2 x 32 bits) function which is used to modify the main flash memory. The following steps show the register access sequence of the programming operation.

- Unlock the FMC_CTL register if necessary.
- Check the BUSY bit in FMC_STAT register to confirm that no flash memory operation is

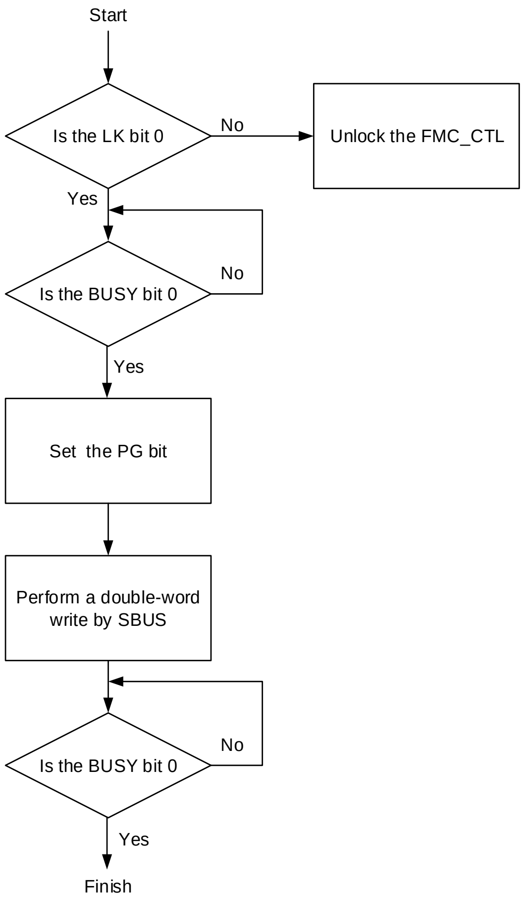
in progress (BUSY equal to 0). Otherwise, wait until the operation has finished.

- Set the PG bit in FMC_CTL register.
- Write the data to be programed with desired absolute address (0x08XX XXXX).
  The SBUS write twice to form a 64-bit data and then the 64-bit data program to flash memory. The data to be programed must double-word alignment.
- Wait until all the operations have completed by checking the value of the BUSY bit in FMC_STAT register.
- Read and verify the flash memory if required using a SBUS access.

When the operation is successfully executed, the ENDF in FMC_STAT register is set, and an interrupt will be triggered by FMC if the ENDIE bit in the FMC_CTL register is set. Note that before the double word programming operation you should check the address that it has been erased. If the address has not been erased, PGERR bit will set when programming the address, except if programming 0x0. Additionally, the program operation will be ignored on protected pages and WPERR bit in FMC_STAT is set. In these conditions, a flash operation error interrupt will be triggered by the FMC if the ERRIE bit in the FMC_CTL register is set. The software can check the PGERR, PGSERR, PGMERR, PGAERR and RPERR bits in the FMC_STAT register to detect which condition occurred in the interrupt handler.

The ***Figure 2-3. Process of word program operation*** displays the word programming operation flow.

Figure 2-3. Process of word program operation



Note: Reading the flash should be avoided when a program / erase operation is ongoing.

If the program / erase operation is interrupted by a power down, reset, ect, the contents in flash will not be guaranteed and leave in an indeterminate state. So appropriate measures should be taken to avoid data loss by interrupt of program / erase.

2.3.8. Main flash fast programming

The FMC provides a fast programming function which is used to modify the main flash memory contents. A row (16 words = 64 bytes) could be programmed to main flash memory in this mode to reduce the page programming time by eliminating the need for verifying the flash locations before they are programmed and to avoid rising and falling time of high voltage for each word. During fast programming, the flash memory clock(HCLK) frequency must be at least 8MHz.

The following steps show the register access sequence of the programming operation.

- Perform a page or mass erase. If not, PGSERR is set
- Unlock the FMC_CTL register if necessary.
- Check the BUSY bit in FMC_STAT register to confirm that no flash memory operation is in progress (BUSY equals to 0). Otherwise, wait until the operation has finished.
- Set the FSTPG bit in FMC_CTL register.
- Write the one row data(16 words) to be programed with desired absolute address (0x08XX XXXX).
- Wait until all the operations have been finished by checking the value of the BUSY bit in FMC_STAT register.

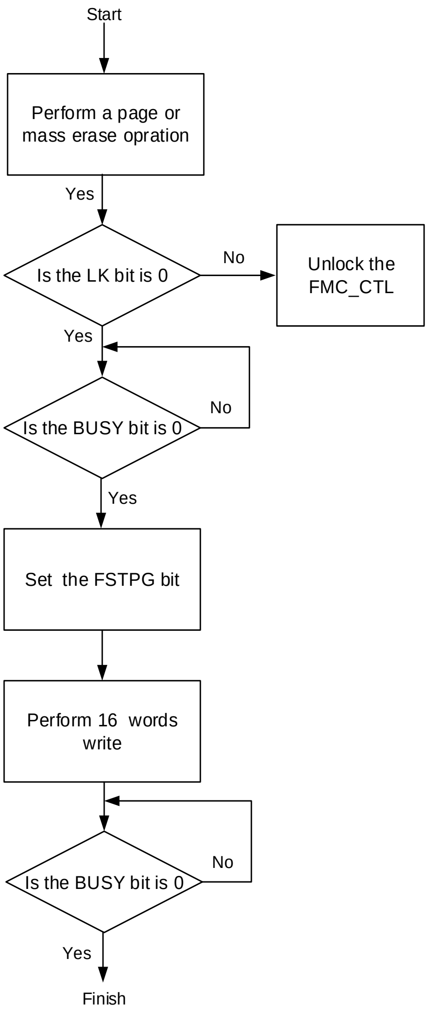
In fast programming: if the data to program doesn’t belong to the same row than the previous programmed words, or the address to program is not greater than the previous one, the PGAERR and FSTPERR bit in FMC_STAT will be set, and the fast programming request will be turned to normal programming to complete the writing request. When the operation is successfully executed, the ENDF in FMC_STAT register is set, and an interrupt will be generated if the ENDIE bit in the FMC_CTL register is set.

In sequential fast programming, if the row is confirmed erased, repeat all the steps to fast program next row. If there are no programming requests anymore, clear the FSTPG bit in FMC_CTL register.

The program operation will be ignored on erase / program protected pages and WPERR bit in FMC_STAT will be set.

In these conditions, a flash operation error interrupt will be generated if the ERRIE bit in the FMC_CTL register is set. The software can check the PGSERR / PGAERR / WPERR / PGERR bit in the FMC_STAT register to detect which condition occurred in the interrupt handler. ***Figure 2-4. Process of fast programming operation*** shows the fast programming operation flow.

Figure 2-4. Process of fast programming operation



Note: 1. The 16 words must be written successively.

2. The 16 words must be aligned.

3. Because the fast program does not check 0xFF in flash macro by hardware, the software must check 0xFF first and don't program one row twice or more between 2 erases. If program one row twice or more between 2 erases, unpredictable result may occur.

4. The fast programming's code is running in SRAM. If it is running in flash, due to the reading access to flash code, the fast programming request will be turned to normal programming.

2.3.9. OTP programming

The OTP programming operation flow is as same as the main flash programming. The OTP block can only be programed once and cannot be erased.

Note: It must ensure the OTP programming sequence completely without any unexpected interrupt, such as system reset or power down. If unexpected interrupt occurs, there is very little probability of corrupting the data stored in flash memory.

2.3.10. Option bytes

Option bytes description

The option bytes registers are reloaded to relevant block of flash memory after each system reset or OBRLD bit set in FMC_CTL register, and the option bytes work. The option complement bytes are the opposite of option bytes. When option bytes reload, if the option complement bytes and option bytes does not match, the OBERR bit in FMC_STAT register is set, and the option byte is set to 0xFF. The ***Table 2-3. Option bytes*** is the detail of option bytes.

Table 2-3. Option bytes

Address	Name	Factory value	Description
0x1fff 7800	OB_SPC	0xA5	option byte Security Protection Code 0xA5: No protection 0xCC: Protection level high Any value except 0xA5 or 0xCC: Protection level low.
0x1fff 7801	OB_USER[7:0]	0xFE	[7]: Reserved [6]: nRST_STDBY 0: Generates a reset if entering standby mode 1: No reset if entering standby mode [5]: nRST_DPSLP 0: Generates a reset if entering Deep-sleep mode 1: No reset if entering Deep-sleep mode [4:3]: BORF_TH 00: BOR falling level 1 with threshold around 2.0 V 01: BOR falling level 2 with threshold around 2.2 V 10: BOR falling level 3 with threshold around 2.5 V 11: BOR falling level 4 with threshold around 2.8 V [2:1]: BORR_TH 00: BOR rising level 1 with threshold around 2.1 V 01: BOR rising level 1 with threshold around 2.3 V 10: BOR rising level 1 with threshold around 2.6 V 11: BOR rising level 1 with threshold around 2.9 V [0]: BORST_EN 0: Brown out reset disable, power-on reset defined by POR/PDR levels 1: Brown out reset enable, values of BORR_TH and BORF_TH taken into account
0x1fff 7802	OB_USER[15:8]	0xFF	[7]: Reserved [6]: SRAM_ECCEN

Address	Name	Factory value	Description
			0: SRAM ECC enable 1: SRAM ECC disable [5]: HXTAL_REMAP 0: Remapping enable 1: Remapping disable [4]: Reserved [3]: nWWDG_HW 0: Hardware window watchdog 1: Software window watchdog [2:1]: Reserved [0]: nFWDG_HW 0: Hardware free watchdog 1: Software free watchdog
0x1fff 7803	OB_USER [23:16]	0xFF	[7:5]: Reserved [4:3]: NRST_MDSEL 00: Reserved 01: A low level on the NRST pin can reset system, internal reset cannot drive NRST pin. 10: NRST pin function as normal GPIO, and only internal RESET. 11: NRST pin configure as input/output mode. [2]: nBOOT0 0: BOOT0 signal is 1 1: BOOT0 signal is 0 [1]: nBOOT1 0: BOOT1 signal is 1 1: BOOT1 signal is 0 Together with the BOOT0 signal, this bit selects boot mode [0]: SWBT0 0: BOOT0 signal depends on PA14/BOOT0 pin 1: BOOT0 signal depends on the option bit nBOOT0
0x1fff 7804	OB_SPC_N	0x5A	OB_SPC complement value
0x1fff 7805	OB_USER_N [7:0]	0x01	OB_USER complement value bit 7 to 0
0x1fff 7806	OB_USER_N [15:8]	0x00	OB_USER complement value bit 15 to 8
0x1fff 7807	OB_USER_N [23:16]	0x00	OB_USER complement value bit 23 to 16
0x1fff 7808	DCRP0_SADDR	0xFF	[6:0]: DCRP area 0 start address
0x1fff 780c	DCRP0_SADDR _N	0x00	DCRP0_SADDR complement value bit 6 to 0

Address	Name	Factory value	Description
0x1fff 7810	DCRP0_EADDR	0x00	[6:0]: DCRP area 0 end address
0x1fff 7813	DCRP_EREN	0xFF	[7]: 0: DCRP area is not erased when a SPC value is decreased from value 1 to value 0 1: DCRP area is erased when a SPC value is decreased from value 1 to value 0 [6:0]: Reserved
0x1fff 7814	DCRP0_EADDR_N	0xFF	DCRP0_EADDR complement value bit 6 to 0
0x1fff 7817	DCRP_EREN_N	0x00	[7]: DCRP_EREN complement value bit 7 [6:0]: Reserved
0x1fff 7818	WP0_SADDR	0xFF	[5:0]: the first page of WP area 0
0x1fff 781a	WP0_EADDR	0x00	[5:0]: the last page of WP area 0
0x1fff 781c	WP0_SADDR_N	0x00	WP0_SADDR complement value bit 5 to 0
0x1fff 781e	WP0_EADDR_N	0xFF	WP0_EADDR complement value bit 5 to 0
0x1fff 7820	WP1_SADDR	0xFF	[5:0]: the first page of WP area 1
0x1fff 7822	WP1_EADDR	0x00	[5:0]: the last page of WP area 1
0x1fff 7824	WP1_SADDR_N	0x00	WP1_SADDR complement value bit 5 to 0
0x1fff 7826	WP1_EADDR_N	0xFF	WP1_EADDR complement value bit 5 to 0
0x1fff 7828	DCRP1_SADDR	0xFF	[6:0]: DCRP area 1 start address
0x1fff 782c	DCRP1_SADDR_N	0x00	DCRP1_SADDR complement value bit 6 to 0
0x1fff 7830	DCRP1_EADDR	0x00	[6:0] DCRP1_EADDR DCRP area 1 end address
0x1fff 7834	DCRP1_EADDR_N	0xFF	[6:0]: DCRP1_EADDR complement value bit 6 to 0
0x1fff 7870	SCR_PAGE_CNT	0x00	[6:0]: Configure the number of pages in secure user area.
0x1fff 7872	BOOTLK	0x00	[0]: This bit is set to force boot from user flash area. 0: Support flash, ram and system boot 1: Only boot from main flash
0x1fff 7874	SCR_PAGE_CNT_N	0xFF	[6:0]: SCR_PAGE_CNT complement value bit 6 to 0
0x1fff 7876	BOOTLK_N	0xFF	BOOTLK complement value bit 0

Option bytes modify

The following steps show the modify sequence.

- Unlock the FMC_CTL register if necessary.
- Check the BUSY bit in the FMC_STAT register to confirm that no flash memory operation is in progress (BUSY equals to 0). Otherwise, wait until the operation has finished.
- Unlock the OBLK bit in the FMC_CTL register by writing the right sequence to the FMC_OBKEY register.

- Wait until the OBLK bit is cleared in the FMC_CTL register.
- Write the desired option bytes in the desired option byte registers.
- Set the OBSTART bit in the FMC_CTL register to send the option byte change command.
- Wait until all the operations have been finished by checking the value of the BUSY bit in the FMC_STAT register.
- Launch a system power on / down reset (or exit from Standby mode) or set the OBRLD bit in FMC_CTL register to load the option bytes.

Note:

1. Once a modification of one option byte is performed, the user options bytes will be automatically erased first. When the operation is successfully executed, the ENDF in FMC_STAT register is set, and an interrupt will be triggered by FMC if the ENDIE bit in the FMC_CTL register is set.

2. Bits 63:32 is the complement byte of Bits 31:0. Every option bit has its complement bits in the same word. The option byte can be written into the corresponding register only when the option byte matches with its complement.

3. If the option byte does not match with its complement, the OBERR bit in FMC_STAT register is set. Option bytes are forced into the values below:
All the OB_USER bytes are forced at 0xFF except for BORST_EN is at 0(brown out reset disable). The status of WP pages is “No protection”. The status of DCRP is “All area protected”. BOOTLK is at 1(only boot from main flash)

2.3.11. Dedicated code read protection area (DCRP)

In the main flash block, FMC can define two executable-only areas, allowing only instruction transactions from the system, but not data access. The DCRP areas have a 512-byte subpage granularity.

Note: Users need to compile their native code accordingly, using the execution-only option when apply execute-only area function.

The DCRP area x (x= 0, 1) are defined by a start address offset and an end address offset:

- From base address + [DCRPx_SADDR x 0x200] (included) to base address + address [(DCRPx_EADDR+ 1) x 0x200] (exclude). The minimum DCRP area size is two DCRP subpages 2 x 512 bytes.

For example, to protect by DCRP from the address 0x0800 0A00 (included) to the address 0x0800 17FF (included), the option bytes must be set as follows:

- FMC_DCRPx_SADDR = 0x05.
- FMC_DCRPx_EADDR = 0x0B.

When executing code in this area, the debug events will be ignored. Only CPU can access DCRP area, using only instruction fetch transactions. In all other cases, access to the DCRP area is illegal. For example, read operations will trigger a RPERR flag in FMC_STAT register, and write operations will be ignored and a WPERR in FMC_STAT register is set.

A valid DCRP area is erase-protected. Pages which located in this area cannot be erased. If a valid DCRP area is setting, mass erase cannot be performed unless erase is performed during the SPC level low to no protection or protection-removed mass erase.

Only CPU can modify the DCRP area definition bits. If DCRP area is valid (not empty), during SPC level low to no protection demotion, if DCRP_EREN bits in FMC_DCRP_EADDR0 is set to 0, the contents of the DCRP area will not be erased, otherwise the contents will be erased.

2.3.12. Erase / program protection (WP)

The FMC provides page erase/program protection functions to prevent inadvertent operations on the flash memory. The page erase or program will not be accepted by the FMC on protected pages. If the page erase or program command is sent to the FMC on a protected page, then the WPERR bit in the FMC_STAT register will be set by the FMC. The WP area are defined by a start address offset and an end address offset. The page protection function can be individually enabled by configuring the WP address registers: FMC_WPx (x = 0,1).

Two WP areas can be defined in the main flash with a granularity of 1 Kbytes. The WP area is defined by:

- From base address + [(WPx_SADDR) x 0x400] (include) to base address + address [(WPx_EADDR + 1) x 0x400] (exclude).

For example, to protect by WP from the address 0x0800 2000 (included) to the address 0x0800 2FFF (included), the option bytes must be set as follows:

- WPx_SADDR = 0x08.
- WPx_EADDR = 0x0B.

Erase/program protected pages cannot either be deleted or programmed. Therefore, if a page is erase/program protected, mass erase cannot be performed.

If the security protection (SPC) level is set to high, the WP area cannot be modified, else WP area can be modified without any restrictions

Note: DCRP or secure user area is erase/program protected.

2.3.13. Data flash erase / program protection (DFWP)

The FMC provides data flash page erase/program protection functions to prevent inadvertent operations on the flash memory. The page erase or program will not be accepted by the FMC on protected data flash pages. If the page erase or program command is sent to the FMC on a protected page, then the WPERR bit in the FMC_STAT register will be set by the FMC. Each data flash page protection can be individually enabled by configuring the DFPxWP(x= 0…3) bits in FMC_DFWP register.

2.3.14. Security protection (SPC)

The FMC provides a security protection function to prevent illegal code / data access on the flash memory. This function is useful for protecting the software / firmware from illegal users. There are 3 levels for protection:

No protection: when setting OB_SPC byte and its complement value to 0xA55A, no protection performed. The main flash and option bytes block are accessible by all operations.

Low level protection: when setting OB_SPC byte to any value except 0xA5 or 0xCC, the low level protection performed. Note that a power reset should be followed instead of a system reset if the SPC modification has been performed while the debug module is still connected to JTAG/SWD device. Under the low security protection, the main flash can only be accessed by user code. In debug mode, boot from SRAM or boot loader mode, all operations to main flash is forbidden. If a read operation to main flash in debug mode, boot from SRAM or boot loader mode, a bus error will be generated. If a program/erase operation to main flash in debug mode, boot from SRAM or boot from boot loader mode, the WPERR bit in the FMC_STAT register will be set. Option bytes block are accessible by all operations, which can be used to disable the security protection. Back to no protection level by setting OB_SPC byte and its complement value to 0xA55A, then a mass erase for main flash and data flash will be performed. Note if low level protection is configured and no DCRP area is defined, it is mandatory to set DCRP_EREN bit.

High level protection: when set OB_SPC byte and its complement value to 0xCC33, high level protection performed. When this level is programmed, debug mode, boot from SRAM or boot from boot loader mode are disabled. The main flash block is accessible by all operations from user code. The user option bits can be read but cannot be modified. And accesses to the other secured areas are also allowed. The OB_SPC byte cannot be reprogrammed. So if high level protection is programmed, it cannot move back to protection level low or no protection level.

2.3.15. Secure user area (SCR)

In the main flash block, FMC can define secure user areas which can be executed only once at boot, and never again unless a reset occurs.

Secure user areas can isolate secure code from application non-secure code. Secure user areas can be used to protect a custom secure boot library, firmware update code, or a third party secure library. When the secured bit SCR in FMC_CTL register is set, erase operations on the SCR area will set the WPERR bit, write operations will trigger a Hard Fault exception immediately after setting the WPERR bit, read operations to the SCR area will generate a Bus Fault instead of setting the RDERR bit.

The size of the Securable memory area is defined by the SCR_PAGE_CNT[6:0] bits of the FMC_SCR register. It can be modified only in OB_SPC level no protection. Its content is erased upon changing from OB_SPC low level protection to no protection, even if it overlaps with DCRP pages.

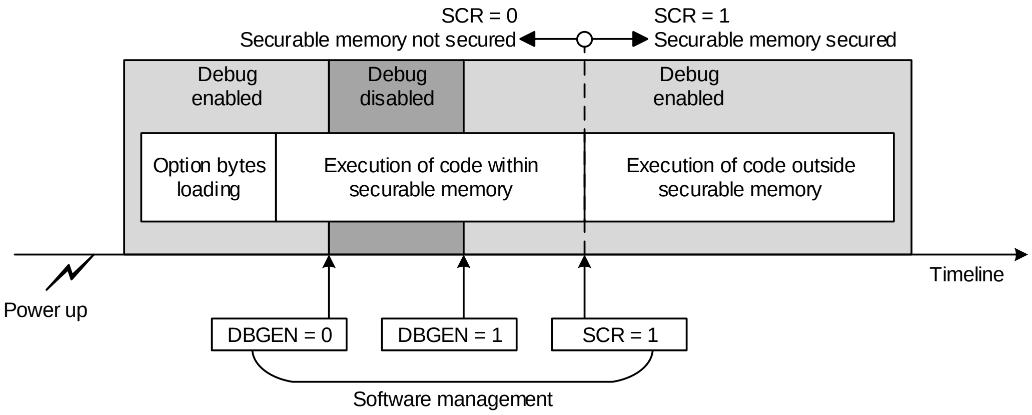
The securable memory area is defined:

- From main flash base address (0x0800 0000) to base address + [SCR_PAGE_CNT x 0x400] (excluded).

2.3.16. Disabling core debug access

The FMC provides a way to temporarily disable the debug access to the core for executing sensitive code or manipulating sensitive data in Securable memory area. The ***Figure 2-5. Example of disabling core debug access*** gives an example of managing DBGEN bit in FMC_WS register and SCR bit in FMC_CTL register.

Figure 2-5. Example of disabling core debug access



2.3.17. Forcing boot from flash memory

The BOOTLK bit in the FMC_SCR register can be configured to force the system to boot from the Main Flash memory. This bit can be reset only when:

1. SPC level is no protection
2. SPC level is low, while no protection level is requested and a full mass-erase is performed.

2.3.18. Interrupt / event flag

Table 2-4. FMC interrupt request

Interrupt event	Event flag	Event flag / Interrupt clearing method	Interrupt enable control bit
End of operation	ENDF	Write ENDF=1	ENDIE
Operation error	OPRERR	Write OPRERR=1	ERRIE
Read protection error	RPERR	Write RPERR=1	RPERRIE
Write protection error	WPERR	Write WPERR=1	N/A
Programming size not match error	PGMERR	Write PGMERR =1	N/A
Programming sequential error	PGERR	Write PGERR=1	N/A
Programming alignment error	PGAERR	Write PGAERR=1	N/A

Interrupt event	Event flag	Event flag / Interrupt clearing method	Interrupt enable control bit
Programming sequence error	PGSERR	Write PGSERR=1	N/A
Fast programming error	FSTPERR	Write FSTPERR =1	N/A

2.4. Register definition

FMC base address: 0x4002 2000

2.4.1. Wait state register (FMC_WS)

Address offset: 0x00

Reset value: 0x000X 0200 (X’ value is 0b010x)，if the chip boots from the Bootloader, the WSCNT will be configured as 0b001.

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved													DBGEN	Reserved	MFPE
													rw		rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				ICRST	Reserved	ICEN	PFEN	Reserved					WSCNT[2:0]		
				rw		rw	rw						rw		

Bits	Fields	Descriptions
31:19	Reserved	Must be kept at reset value
18	DBGEN	Debugger enable This bit is used to enable/disable the debugger by software 0: Debugger disabled 1: Debugger enabled
17	Reserved	Must be kept at reset value
16	MFPE	Main flash programmed or empty flag This bit is used to indicate whether the first location of main flash is programmed or empty 0: Main flash programmed 1: Main flash empty The bit can be set and reset by software
15:12	Reserved	Must be kept at reset value
11	ICRST	IBUS cache reset. This bit can be write only when ICEN is set to 0. 0: No effect 1: IBUS cache reset
10	Reserved	Must be kept at reset value
9	ICEN	IBUS cache enable 0: IBUS cache disable 1: IBUS cache enable

8	PFEN	Pre-fetch enable 0: Pre-fetch disable 1: Pre-fetch enable
7:3	Reserved	Must be kept at reset value
2:0	WSCNT[2:0]	Wait state counter register These bits set and reset by software. 000: 0 wait state added 001: 1 wait state added Others: Reserved

2.4.2. Unlock key register (FMC_KEY)

Address offset: 0x08

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
KEY[31:16]															
w															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
KEY[15:0]															
w															

Bits	Fields	Descriptions
31:0	KEY[31:0]	FMC_CTL unlock register These bits can only be written by software. Write KEY[31:0] with the consecutive keys below to unlock FMC_CTL register.

2.4.3. Option byte unlock key register (FMC_OBKEY)

Address offset: 0x0C

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
OBKEY[31:16]															
w															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OBKEY[15:0]															
w															

Bits	Fields	Descriptions
31:0	OBKEY[31:0]	FMC_CTL option bytes operation unlock register

These bits can only be written by software.

Write OBKEY[31:0] with the consecutive keys below to unlock FMC_OBCTL register.

2.4.4. Status register (FMC_STAT)

Address offset: 0x10

Reset value: 0x0000 0000

This register has to be accessed by word(32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															BUSY
													r		r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OBERR	RPERR	Reserved				FSTPERR	Reserved	PGSERR	PGMERR	PGAERR	WPERR	PGERR	Reserved	OPRERR	ENDF
rc_w1	rc_w1					rc_w1		rc_w1	rc_w1	rc_w1	rc_w1	rc_w1		rc_w1	rc_w1

Bits	Fields	Descriptions
31:17	Reserved	Must be kept at reset value
16	BUSY	The flash is busy bit When the operation is in progress, this bit is set to 1. When the operation is end or an error generated, this bit is clear to 0.
15	OBERR	Option byte read error bit. This bit is set by hardware when the option byte and its complement byte do not match, and the option byte set 0xFF.
14	RPERR	Read protection error flag bit. When access address protected by DCRP or SCR, this bit is set by hardware. This bit can be cleared it by writing 1. 0: No read protection error occurs 1: Read protection error occurs
13:10	Reserved	Must be kept at reset value
9	FSTPERR	Fast programming error flag bit. Set by hardware when a fast programming sequence is interrupted due to an error (alignment, size, write protection or data miss),. The corresponding error bit (PGAERR, PGMERR or WPERR) is set at the same time. This bit can be cleared it by writing 1.
8	Reserved	Must be kept at reset value
7	PGSERR	Program sequence error flag bit. Set by hardware if a write operation to the Flash is performed while not setting PG previously. It is also set by hardware when PGERR, PGMERR, PGAERR or WPERR because a previous programming error occured. This bit can be cleared

		by writing 1. 0: No program sequence error occurs 1: Program sequence error occurs
6	PGMERR	Program size not match error flag bit. This bit is set by hardware when program size is a half-word/word access. The only correct programming size is double word. This bit can be cleared by writing 1.
5	PGAERR	Program alignment error flag bit This bit is set by hardware when DBUS write data is not alignment. This bit can be cleared by writing 1.
4	WPERR	Erase / program protection error flag bit. When an erase/program protection error occurs, this bit is set by hardware. This bit can be cleared by writing 1. 0: No write protection error occurs 1: Write protection error occurs
3	PGERR	Program error flag bit When a double-word address programming to the flash while the value is not 0xFFFF FFFF FFFF FFFF, this bit is set by hardware. This bit can be cleared by writing 1.
2	Reserved	Must be kept at reset value
1	OPRERR	Operation error flag bit If a flash memory programming or erase operation completes unsuccessfully and error interrupts are enabled (ERRIE = 1), it is set by hardware. This bit can be cleared by writing 1.
0	ENDF	End of operation flag bit When the operation executed successful, this bit is set by hardware. This bit can be cleared by writing 1.

2.4.5. Control register (FMC_CTL)

Address offset: 0x14

Reset value: 0xC000 0000

This register can be accessed by byte(8-bit), half-word (16-bit) and word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LK	OBLK	Reserved	SCR	OBRLD	RPERRIE	ERRIE	ENDIE	Reserved					FSTPG	OBSTART	START
rs	rs		rs	rc_w1	rw	rw	rw						rs	rs	rs

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved							PN[5:0]						MER	PER	PG
							rw						rw	rw	rw

Bits	Fields	Descriptions

31	LK	FMC_CTL lock bit This bit is cleared by hardware when right sequent written to FMC_KEY register. This bit can be set by software.
30	OBLK	FMC_OBCTL lock bit If this bit is set, all bits about user option in FMC_OBCTL register and so option page are locked. This bit is cleared by hardware when right sequence written to FMC_OBKEY register. This bit can be set by software.
29	Reserved	Must be kept at reset value
28	SCR	Secure-access area enable bit. This bit is set to lock the secure-access area. It is set by software when exiting the secure-access area, and can only be written once. 0: Disable secure-access area 1: Enable secure-access area This bit can only be set by software and cleared by system reset.
27	OBRLD	Option byte reload bit This bit is set by software. 0: No effect 1: Force option byte reload **NOTE:** It cannot be written if OBLK is set
26	RPERRIE	Read protection error interrupt enable bit This bit is set or cleared by software only when LK is set to 0 0: Disable read protection error interrupt 1: Enable read protection error interrupt
25	ERRIE	OPRERR error interrupt enable bit This bit is set or cleared by software. 0: Disable OPRERR error interrupt 1: Enable OPRERR error interrupt
24	ENDIE	End of operation interrupt enable bit This bit is set or cleared by software. 0: No interrupt generated by hardware 1: End of operation interrupt enable
23:19	Reserved	Must be kept at reset value
18	FSTPG	Main flash fast program command bit This bit is set or clear by software 0: Disabled fast programming 1: Enabled fast programming
17	OBSTART	Send option byte change command to FMC bit. This bit is set by software to send option byte change command to FMC only when

		OBLK is set to 0.This bit is cleared by hardware when the BUSY bit is cleared
16	START	Send erase command to FMC bit This bit is set by software to send erase command to FMC. This bit is cleared by hardware when the BUSY bit is cleared.
15:9	Reserved	Must be kept at reset value
8:3	PN[5:0]	Page number selection These bits select the page to erase: 000000: page 0 000001: page 1 ... 111111: page 63
2	MER	Main flash mass erase command bit This bit is set or cleared by software. 0: No effect 1: Main flash mass erase command
1	PER	Main flash page erase command bit This bit is set or cleared by software. 0: No effect 1: Main flash page erase command
0	PG	Main flash page program command bit This bit is set or cleared by software. 0: Disabled flash programming 1: Enabled flash programming

2.4.6. Option byte control register (FMC_OBCTL)

Address offset: 0x20

Reset value: 0xXXXX XXXX. The value of the register is loaded from Flash memory when OBRLD in FMC_CTL register is set or system reset.

This register can be accessed by byte(8-bit), half-word (16-bit) and word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved			NRST_MDSEL[1:0]		nBOOT0	nBOOT1	SWBT0	Reserved	SRAM_ECCEN	HXTAL_REMAP	Reserved	nWWDG_HW	Reserved		nFWDG_HW
			rw		rw	rw	rw		rw	rw		rw			rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	nRST_STDBY	nRST_DPSLP	BORF_TH[1:0]		BORR_TH[1:0]		BORST_EN	SPC[7:0]							
	rw	rw	rw		rw		rw	rw							

Bits	Fields	Descriptions

31:29	Reserved	Must be kept at reset value
28:27	NRST_MDSEL[1:0]	NRST pin mode selection. 00: Reserved 01: A low level on the NRST pin can reset system, internal reset can not drive NRST pin. 10: NRST pin function as normal GPIO, and only internal RESET. 11: NRST pin configure as input/output mode.
26	nBOOT0	BOOT0 option bit 0: BOOT0 is 1 1: BOOT0 is 0
25	nBOOT1	BOOT1 option bit 0: BOOT1 is 1 1: BOOT1 is 0 Together with the BOOT0 pin, this bit selects boot mode
24	SWBT0	Software BOOT0 0: BOOT0 taken from PA14/BOOT0 pin 1: BOOT0 taken from the option bit nBOOT0
23	Reserved	Must be kept at reset value
22	SRAM_ECCEN	SRAM ECC disable 0: SRAM ECC enable 1: SRAM ECC disable
21	HXTAL_REMAP	HXTAL remapping 0: Remapping enable 1: Remapping disable When this bit is set to 0, the HXTAL clock source will be remapped from PF0-OSC_IN / PF1-OSC_OUT pins to PC14-OSCX_IN/ PC15-OSCX_OUT. Thus PC14-OSCX_IN / PC15-OSCX_OUT are shared by both LXTAL and HXTAL and the two clock sources cannot be use simultaneously
20	Reserved	Must be kept at reset value
19	nWWDG_HW	Window watchdog configuration bit 0: Hardware window watchdog 1: Software window watchdog
18:17	Reserved	Must be kept at reset value
16	nFWDG_HW	Free watchdog configuration bit 0: Hardware free watchdog 1: Software free watchdog
15	Reserved	Must be kept at reset value

14	nRST_STDBY	Option byte standby reset value 0: Generates a reset if entering standby mode 1: No reset if entering standby mode
13	nRST_DPSLP	Option byte deepsleep reset value 0: Generates a reset if entering deepsleep mode 1: No reset if entering deepsleep mode
12:11	BORF_TH[1:0]	BOR threshold at falling VDD supply Falling VDD crossings this threshold activates the reset signal. 00: BOR falling level 1 with threshold around 2.0 V 01: BOR falling level 2 with threshold around 2.2 V 10: BOR falling level 3 with threshold around 2.5 V 11: BOR falling level 4 with threshold around 2.8 V **Note:** All configuration combinations where BORR_TH is less than BORF_TH are considered illegal configurations. In the illegal case, both BORR_TH and BORF_TH are forced at the default value of 0b11.
10:9	BORR_TH[1:0]	BOR threshold at rising VDD supply Falling VDD crossings this threshold activates the reset signal. 00: BOR rising level 1 with threshold around 2.1 V 01: BOR rising level 2 with threshold around 2.3 V 10: BOR rising level 3 with threshold around 2.6 V 11: BOR rising level 4 with threshold around 2.9 V **Note:** All configuration combinations where BORR_TH is less than BORF_TH are considered illegal configurations. In the illegal case, both BORR_TH and BORF_TH are forced at the default value of 0b11.
8	BORST_EN	Brown out reset enable 0: Brown out reset disable, power-on reset defined by POR/PDR levels 1: Brown out reset enable, values of BORR_TH and BORF_TH taken into account
7:0	SPC[7:0]	Security protection level option byte status bits 0xA5: No protection 0xCC: Protection level high, chip read protection enabled Any value except 0xA5 or 0xCC: Protection level low, memory read protection enabled

2.4.7. DCRP start address register 0 (FMC_DCRP_SADDR0)

Address offset: 0x24

Reset value: 0x0000 00XX

This register has to be accessed by word(32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved									DCRP0_SADDR[6:0]						
												rw			

Bits	Fields	Descriptions
31:7	Reserved	Must be kept at reset value
6:0	DCRP0_SADDR[6:0]	DCRP area 0 start address DCRP0_SADDR contains the offset of the first subpage of the DCRP area 0.

2.4.8. DCRP end address register 0(FMC_DCRP_EADDR0)

Address offset: 0x28

Reset value: 0xX000 00XX

This register has to be accessed by word(32-bit)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DCRP_EREN	Reserved														
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved									DCRP0_EADDR[6:0]						
												rw			

Bits	Fields	Descriptions
31	DCRP_EREN	DCRP area erase enable configuration bit. 0: DCRP is not erased when SPC value in FMC_OBCTL register is decreased from value 1 to value 0. 1: DCRP is erased when SPC value in FMC_OBCTL register is decreased from value 1 to value 0.
30:7	Reserved	Must be kept at reset value
6:0	DCRP0_EADDR[6:0]	DCRP area 0 end address offset DCRP0_EADDR contains the offset of the last subpage of the DCRP area 0.

2.4.9. Write protection area 0 register (FMC_WP0)

Address offset: 0x2C

Reset value: 0x00XX 00XX

This register has to be accessed by word(32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved										WP0_EADDR[5:0]					
													rw		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Reserved	WP0_SADDR[5:0]
	rw

Bits	Fields	Descriptions
31:22	Reserved	Must be kept at reset value
21:16	WP0_EADDR[5:0]	Write protection area 0 end address offset WP0_EADDR contains the last page of Write protect area 0.
15:6	Reserved	Must be kept at reset value
5:0	WP0_SADDR[5:0]	Write protection area 0 end address offset WP0_SADDR contains the first page of Write protect area 0.

2.4.10. Write protection area 1 register (FMC_WP1)

Address offset: 0x30

Reset value: 0x00XX 00XX

This register has to be accessed by word(32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved										WP1_EADDR[5:0]					
										rw					

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved										WP1_SADDR[5:0]					
										rw					

Bits	Fields	Descriptions
31:22	Reserved	Must be kept at reset value
21:16	WP1_EADDR[5:0]	Write protection area 1 end address offset WP1_EADDR contains the last page of Write protect area 1
15:6	Reserved	Must be kept at reset value
5:0	WP1_SADDR[5:0]	Write protection area 1 start address offset WP1_SADDR contains the first page of Write protect area 1

2.4.11. DCRP start address register 1(FMC_DCRP_SADDR1)

Address offset: 0x34

Reset value: 0x0000 00XX

This register has to be accessed by word(32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved									DCRP1_SADDR[6:0]						
									rw						

Bits	Fields	Descriptions
31:7	Reserved	Must be kept at reset value
6:0	DCRP1_SADDR[6:0]	DCRP area 1 start address offset DCRP1_SADDR contains the offset of the first subpage of the DCRP area 1

2.4.12. DCRP end address register 1(FMC_DCRP_EADDR1)

Address offset: 0x38

Reset value: 0x0000 00XX

This register has to be accessed by word(32-bit)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved									DCRP1_EADDR[6:0]						
									rw						

Bits	Fields	Descriptions
31:7	Reserved	Must be kept at reset value
6:0	DCRP1_EADDR[6:0]	DCRP area 1 end address offset DCRP1_EADDR contains the offset of the last subpage of the DCRP area 1.

2.4.13. Secure user area register (FMC_SCR)

Address offset: 0x80

Reset value: 0x000X 00XX

This register has to be accessed by word(32-bit)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															BOOTLK
															rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved									SCR_PAGE_CNT[6:0]						
									rw						

Bits	Fields	Descriptions
31:17	Reserved	Must be kept at reset value
16	BOOTLK	This bit is set to force boot from user Flash area.

		0: Support flash, ram and system boot 1: Only boot from main flash
15:7	Reserved	Must be kept at reset value
6:0	SCR_PAGE_CNT[6:0]	Configure the number of pages in secure user area. Secure user area starts at memory 0x0800 0000.The memory size is SCR_PAGE_CNT * page size. **Note:** This field can be changed when SPC level is no protection only.

2.4.14. Product ID register (FMC_PID)

Address offset: 0x110

Reset value: 0xXXXX XXXX

This register has to be accessed by word(32-bit)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PID[31:16]															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PID[15:0]															
r															

Bits	Field	Descriptions
31:0	PID[31:0]	Product reserved ID code register These bits are read only by software. These bits are unchanged constant after power on. These bits are one time program when the chip produced.

3. Power management unit (PMU)

3.1. Overview

The power consumption is regarded as one of the most important issues for the devices of GD32C2x1. Power management unit (PMU) provides six types of power saving modes, including Run1, Sleep, Sleep1, Deep-sleep, Deep-sleep1 and Standby mode. These modes reduce the power consumption and allow the application to achieve the best tradeoff among the conflicting demands of CPU operating time, speed and power consumption. For GD32C2x1 devices, there are two power domains, including V_{DD} / V_{DDA} domain and 1.2V domain. The power of the V_{DD} / V_{DDA} domain is supplied directly. An embedded LDO in the V_{DD} / V_{DDA} domain is used to supply the 1.2V domain power.

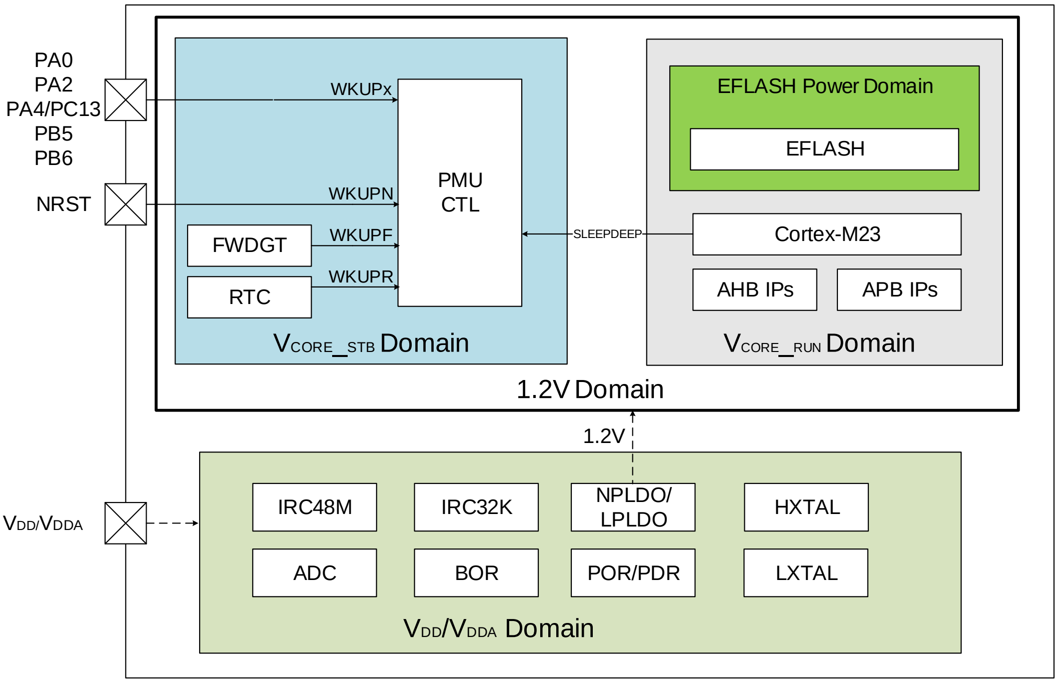
3.2. Characteristics

- Two power domains: V_{DD} / V_{DDA} and 1.2V power domains.
- Six power saving modes: Run1, Sleep, Sleep1, Deep-sleep, Deep-sleep1 and Standby mode.
- Internal Normal Power Voltage regulator (NPLDO) supplies around 1.2V voltage source for 1.2V domain in Run, Sleep and Deep-sleep mode.
- Internal Low Power Voltage regulator (LPLDO) supplies around 1.2V voltage source for 1.2V domain in Run1, Sleep1, Deep-sleep1 and Standby mode.
- EFLASH can be power-off alone in Run / Run1 / Deep-sleep / Deep-sleep1 mode.
- Power supply supervision: POR/PDR monitor, BOR monitor.

3.3. Function overview

Figure 3-1. Power supply overview provides details on the internal configuration of the PMU and the relevant power domains.

Figure 3-1. Power supply overview



POR: Power On Reset PDR: Power Down Reset BOR: Brownout Reset

NPLDO: Normal power Voltage Regulator LPLDO: Low power Voltage Regulator

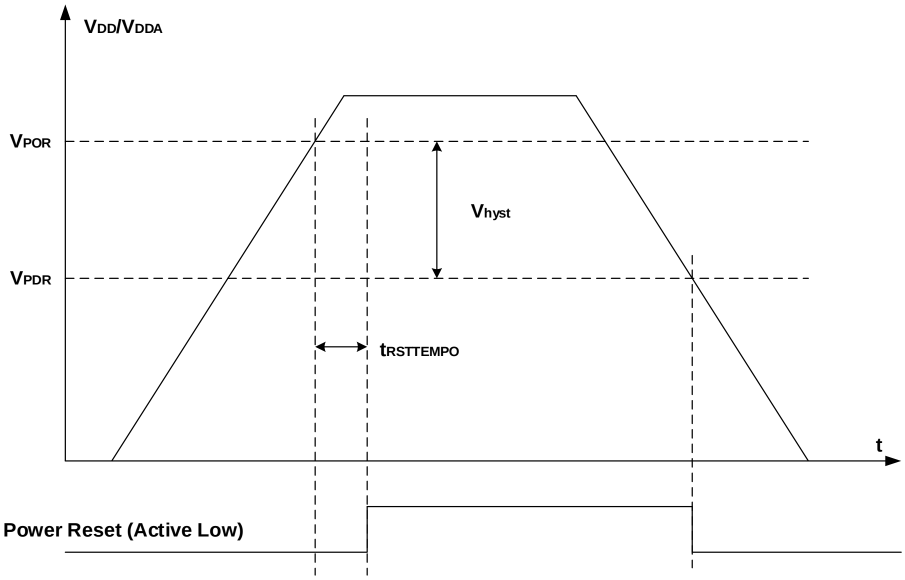
3.3.1. V_{DD} / V_{DDA} power domain

V_{DD} / V_{DDA} domain includes HXTAL (high speed crystal oscillator), LXTAL (low speed crystal oscillator), NPLDO, LPLDO, POR/PDR (power on/down Reset), BOR (Brownout Reset), ADC (A/D converter), IRC48M (internal 48MHz RC oscillator), IRC32K (internal 32KHz RC oscillator) etc.

The LDO, which is implemented to supply power for the 1.2V domain, is always enabled after reset. It can be configured to operate in NPLDO or LPLDO status. NPLDO supplies around 1.2V voltage source for 1.2V domain in Run, Sleep and Deep-sleep mode. LPLDO supplies around 1.2V voltage source for 1.2V domain in Run1, Sleep1, Deep-sleep1 and Standby mode.

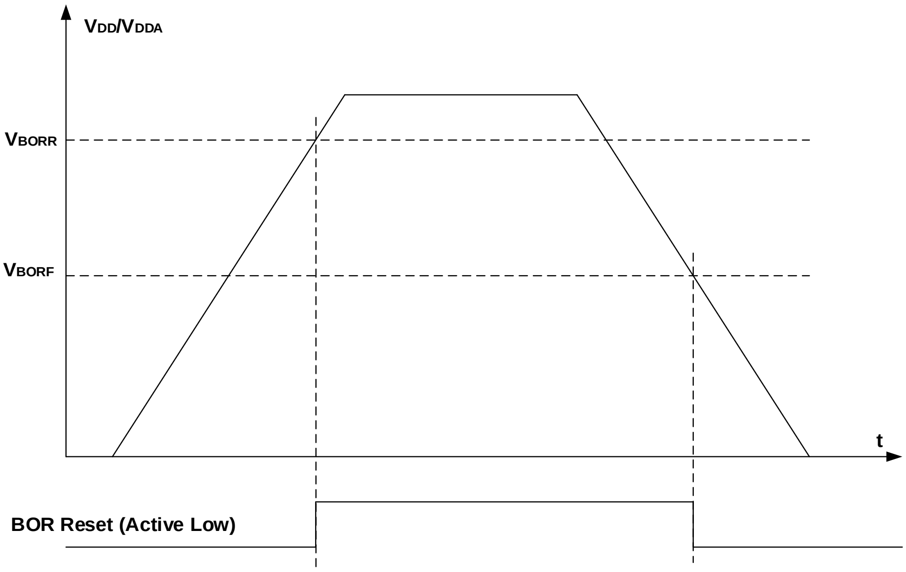
The POR / PDR circuit is implemented to detect V_{DD} / V_{DDA} and generate the power reset signal which resets the whole chip except the V_{CORE_STB} domain when the supply voltage is lower than the specified threshold. *Figure 3-2. Waveform of the POR / PDR* shows the relationship between the supply voltage and the power reset signal. V_{POR} indicates the threshold of power on reset, while V_{PDR} means the threshold of power down reset. The hysteresis voltage is V_{hyst}.

Figure 3-2. Waveform of the POR / PDR



The BOR (Brown-out Reset) circuit is used to detect V_{DD} / V_{DDA}. When the supply voltage is lower than the specified threshold which defined in the BORR_TH and BORF_TH bits in option bytes, BOR will generate the power reset signal which resets the whole chip except the V_{CORE_STB} domain. Notice that the POR/PDR circuit is always implemented. BOR is enabled by setting BORST_EN bit in option bytes. ***Figure 3-3. Waveform of the BOR*** shows the relationship between the supply voltage and the BOR reset signal. V_{BORR} and V_{BORF} indicates the threshold of BOR on reset, which defined in the BORR_TH and BORF_TH bits in option bytes.

Figure 3-3. Waveform of the BOR



3.3.2. 1.2V power domain

The main functions that include Cortex®-M23 logic, AHB / APB peripherals are located in this power domain. Once the 1.2V is powered up, the POR will generate a reset sequence on the 1.2V power domain. If need to enter the expected power saving mode, the associated control bits must be configured. Then, once a WFI (Wait for Interrupt) or WFE (Wait for Event) instruction is executed, the device will enter an expected power saving mode which will be discussed in the following section.

There are two power domains in digital logics of MCU. V_{CORE_RUN} domain shall work when MCU is in Run / Run1 / Sleep / Sleep1 / Deep-sleep / Deep-sleep1 mode and be powered off when MCU in standby mode to save power. V_{CORE_STB} power domain is an always-on power domain when MCU is powered up.

Two LDOs can provides 1.2V voltage for digital logic domain. NPLDO can supply MCU to work in full performance mode. LPLDO can be used to supply MCU which is in low performance mode.

EFLASH power domain

The EFLASH can be power-off alone. The default power status is power on after system reset. In Run/Run1 mode, the EFLASH can be powered off by setting the EFPSLEEP bit in PMU_CTL1 register. When the MCU enters deep-sleep mode, EFLASH can be controlled to switch off by setting the EFDSPSLEEP bit in PMU_CTL1 register.

When NPLDO is off and only LPLDO is on, EFLASH can not work properly. In this case, the code should be stored in SRAM.

3.3.3. Power saving modes

After system reset / power reset or wakeup from standby mode, the MCU enters Run mode. All power domains are active and the NPLDO works in 1.2V mode. Users can achieve lower power consumption through slowing down the system clocks (HCLK and PCLK) or gating the clocks of the unused peripherals. Besides, six power saving modes are provided to achieve even lower power consumption, they are Run1 mode, Sleep mode, Sleep1 mode, Deep-sleep mode, Deep-sleep1 mode and Standby mode.

Run1 mode

When in Run1 mode, the NPLDO is off and LPLDO is on, the system clock source should be IRC32K.

Sleep mode

The Sleep mode is corresponding to the SLEEPING mode of the Cortex®-M23. In Sleep mode, only clock of Cortex®-M23 is off. To enter the Sleep mode, it is only necessary to clear the

SLEEPDEEP bit in the Cortex®-M23 System Control Register, and execute a WFI or WFE instruction. If the Sleep mode is entered by executing a WFI instruction, any interrupt can wake up the system. If it is entered by executing a WFE instruction, any wakeup event can wake up the system (If SEVONPEND is 1, any interrupt can wake up the system, refer to Cortex®-M23 Technical Reference Manual). The mode offers the lowest wakeup time as no time is wasted in interrupt entry or exit.

According to the SLEEPONEXIT bit in the Cortex®-M23 SCR (System Control Register), there are two options to select the Sleep mode entry mechanism.

- Sleep-now: if the SLEEPONEXIT bit is cleared, the MCU enters Sleep mode as soon as WFI or WFE instruction is executed.
- Sleep-on-exit: if the SLEEPONEXIT bit is set, the MCU enters Sleep mode as soon as it exits from the lowest priority ISR.

Sleep1 mode

The Sleep1 mode is corresponding to the SLEEPING mode of the Cortex®-M23 When in Run1 mode. In this mode the NPLDO is off and LPLDO is on, the system clock source is IRC32K.

Deep-sleep mode

The Deep-sleep mode is based on the SLEEPDEEP mode of the Cortex®-M23. In Deep-sleep mode, all clocks in the V_{CORE_RUN} domain are off, and all of IRC48M, HXTAL are disabled. The contents of SRAM and registers are preserved. The NPLDO is on. Before entering the Deep-sleep mode, it is necessary to set the SLEEPDEEP bit in the Cortex®-M23 System Control Register, and set LPMOD bits to “00” in the PMU_CTL0 register. Then, the device enters the Deep-sleep mode after a WFI or WFE instruction is executed. If the Deep-sleep mode is entered by executing a WFI instruction, any interrupt from EXTI lines can wake up the system. If it is entered by executing a WFE instruction, any wakeup event from EXTI lines can wake up the system (If SEVONPEND is 1, any interrupt from EXTI lines can wake up the system, refer to Cortex®-M23 Technical Reference Manual). When exiting the Deep-sleep mode, the IRC48M is selected as the system clock.

Note: In order to enter Deep-sleep mode smoothly, all EXTI line pending status (in the EXTI_PD register) and related peripheral flags must be reset, refer to ***Table 5-3. EXTI source***. If not, the program will skip the entry process of Deep-sleep mode to continue to execute the following procedure.

Deep-sleep1 mode

The Deep-sleep1 mode is based on the SLEEPDEEP mode of the Cortex®-M23. In Deep-sleep1 mode, all clocks in the V_{CORE_RUN} domain are off, and all of IRC48M, HXTAL are disabled. The contents of SRAM and registers are preserved. The NPLDO is off and LPLDO is on. Before entering the Deep-sleep1 mode, it is necessary to set the SLEEPDEEP bit in the Cortex®-M23 System Control Register, and set LPMOD bits to “01” in the PMU_CTL0 register. Then, the device enters the Deep-sleep1 mode after a WFI or WFE instruction is executed. If the Deep-sleep1 mode is entered by executing a WFI instruction, any interrupt from EXTI lines can wake up the system. If it is entered by executing a WFE instruction, any wakeup event from EXTI lines can wake up the system (If SEVONPEND is 1, any interrupt from EXTI lines can wake up the system, refer to Cortex®-M23 Technical Reference Manual). When exiting the Deep-sleep1 mode, the IRC48M is selected as the system clock.

Note: In order to enter Deep-sleep1 mode smoothly, all EXTI line pending status (in the EXTI_PD register) and related peripheral flags must be reset, refer to ***Table 5-3. EXTI source***. If not, the program will skip the entry process of Deep-sleep1 mode to continue to execute the following procedure.

Standby mode

The Standby mode is based on the SLEEPDEEP mode of the Cortex®-M23. In Standby mode, the whole V_{CORE_RUN} domain is power off, the NPLDO is shut down, and all of IRC48M, HXTAL are disabled. Before entering the Standby mode, it is necessary to set the LPMOD bits to “11” in the PMU_CTL0 register, and clear WUF bit in the PMU_CS register, and set the SLEEPDEEP bit in the Cortex®-M23 System Control Register. Then, the device enters the Standby mode after a WFI or WFE instruction is executed, and the STBF status flag in the PMU_CS register indicates that the MCU has been in Standby mode. There are four wakeup sources for the Standby mode, including the external reset from NRST pin, the RTC alarm, the FWDGT reset, LXTAL clock failure detedtion and the rising edge on WKUPx pins. The Standby mode achieves the lowest power consumption, but spends longest time to wake up. Besides, the contents of SRAM and registers in V_{CORE_RUN} power domain are lost in Standby mode. When exiting from the Standby mode, a power-on reset occurs and the Cortex®-M23 will execute instruction code from the 0x00000000 address.

Table 3-1. Power saving mode summary

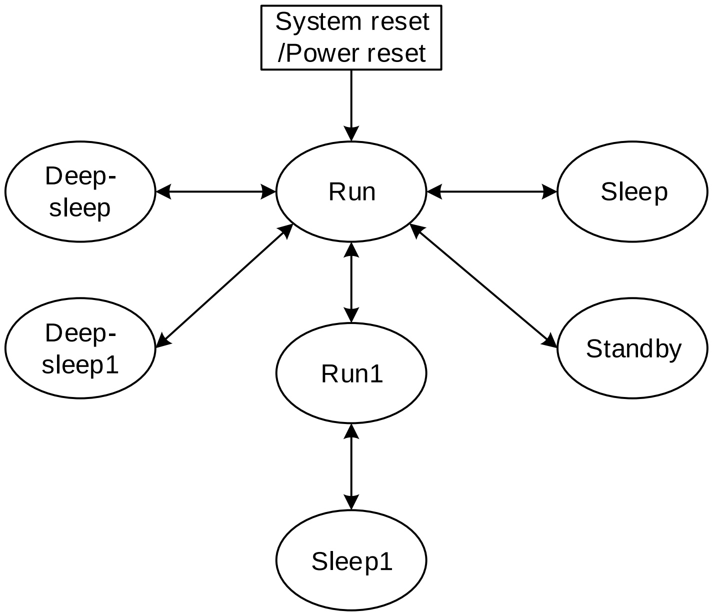
Mode	Description	LDO	Entry	Wakeup	Wakeup status	Wakeup Latency
Run	no effect on all clocks, all power on	NPLDO on LPLDO on	system / power reset or wakeup from standby	-	-	-
Run1	system clock = IRC32K	NPLDO off LPLDO on	Set LPLDOEN	Clear LPLDOEN	-	NPLDO wakeup time + Flash wakeup time
Sleep	Only CPU clock is off	NPLDO on LPLDO on	SLEEPDEEP = 0,WFI or WFE from Run	Any interrupt for WFI Any event (or interrupt when SEVONPEND is 1) for WFE	Run	-
Sleep1	Only CPU clock is off system clock = IRC32K	NPLDO off LPLDO on	SLEEPDEEP = 0,WFI or WFE from Run1	Any interrupt for WFI Any event (or interrupt when SEVONPEND	Run1	-

Mode	Description	LDO	Entry	Wakeup	Wakeup status	Wakeup Latency
				is 1) for WFE		
Deep-sleep	1. All clocks in the V_{CORE_RUN} domain are off 2. Disable IRC48M, HXTAL	NPLDO on LPLDO on	SLEEPDEEP = 1, LPMOD = 00, WFI or WFE	Any interrupt from EXTI lines for WFI Any event(or interrupt when SEVONPEND is 1) from EXTI for WFE	Run	IRC48M wakeup time + Flash wakeup time
Deep-sleep1	1. All clocks in the V_{CORE_RUN} domain are off 2. Disable IRC48M, HXTAL	NPLDO off LPLDO on	SLEEPDEEP = 1, LPMOD = 01, WFI or WFE	Any interrupt from EXTI lines for WFI Any event(or interrupt when SEVONPEND is 1) from EXTI for WFE	Run	IRC48M wakeup time + NPLDO wakeup time + Flash wakeup time
Standby	1. The V_{CORE_RUN} domain is power off 2. Disable IRC48M, HXTAL	NPLDO off LPLDO on	SLEEPDEEP = 1, LPMOD = 11, WFI or WFE	1. NRST pin 2. WKUP pins 3. FWDGT reset 4. RTC 5. LCKMD	Run	IRC48M wakeup time, + NPLDO wakeup time+Flash wakeup time

Note:

- Do not allow to enter Sleep/Deep-sleep/Deep sleep1/Standby directly from Run1 mode. The conversion between different modes is shown in the ***Figure 3-4. Power saving mode conversion diagram***.
- If MCU enters Sleep/Deep-sleep/Deep-sleep1/Standby mode from Run mode, the software should clear LPLDOEN.

Figure 3-4. Power saving mode conversion diagram



- In standby mode, all I/Os are in high-impedance state except NRST pin, PC13 pin when configured for RTC function, PC14 and PC15 pins when used as LXTAL crystal oscillator pins, and WKUPx pin if enabled.

- The status of each module in different modes is shown in ***Table 3-2. The status of module in different modes***.

Table 3-2. The status of module in different modes

Module	Run	Run1	Sleep	Sleep1	DeepSleep		DeepSleep1		Standby	
Wakeup capability	-	-	-	-	-	Wakeup source	-	Wakeup source	-	Wakeup source
CPU	1	1	-	-	-	-	-	-	-	-
Flash	1	-	1	-	4	-	-	-	-	-
SRAM	1	1	1	1	3	-	3	-	-	-
Vcore supply	1	1	1	1	1	1	1	1	-	-
POR/PDR	1	1	1	1	1	1	1	1	1	1
BOR	2	2	2	2	3	3	3	3	3	3
NRST	1	1	1	1	1	1	1	1	1	1
DMA/DMAMUX	2	2	3	3	-	-	-	-	-	-
IRC48M	1	1	3	3	-	-	-	-	-	-
HXTAL	2	2	3	3	-	-	-	-	-	-
IRC32K	2	2	3	3	3	-	3	-	3	-
LXTAL	2	2	3	3	3	-	3	-	3	-
CKM	2	2	3	3	-	-	-	-	-	-
LCKM	2	2	3	3	3	2	3	2	3	2
RTC	2	2	3	3	3	2	3	2	3	2
USART0	2	2	3	3	3	2	3	2	-	-
USART1	2	2	3	3	-	-	-	-	-	-

Module	Run	Run1	Sleep	Sleep1	DeepSleep		DeepSleep1		Standby	
Wakeup capability	-	-	-	-	-	Wakeup source	-	Wakeup source	-	Wakeup source
USART2	2	2	3	3	-	-	-	-	-	-
I2C0	2	2	3	3	3	2	3	2	-	-
I2C1	2	2	3	3	3	2	3	2	-	-
SPI0/I2S	2	2	3	3	-	-	-	-	-	-
SPI1	2	2	3	3	-	-	-	-	-	-
ADC	2	2	3	3	-	-	-	-	-	-
Internal temperature sensor	2	2	3	3	-	-	-	-	-	-
TIMERx	2	2	3	3	-	-	-	-	-	-
FWDGT	2	2	3	3	3	2	3	2	3	2
WWDGT	2	2	3	3	-	-	-	-	-	-
SysTick	2	2	3	3	-	-	-	-	-	-
CRC	2	2	3	3	-	-	-	-	-	-
CMP	2	2	3	3	-	-	-	-	-	-
GPIOs	2	2	3	3	3	2	3	2	-	2(1)
Individual peripheral clock	2	2	4(2)	4(2)	4(2)	-	4(2)	-	-	-

'-'： The module is unavailable or irrelevant.

'1'： The module is enabled after reset or wakeup from deepsleep/deepsleep1/ standby.

'2'： Disabled by default, can be configured by software to enable or not.

'3'： The state is the same as before entering low-power mode.

'4'： Software configurable to automatically disable/power down upon entering low-power mode.

(1). Only WKUPx pin can wake up.

(2). It can be configured by the RCU_AHB1SPDPEN / RCU_AHB2SPDPEN / RCU_APBSPDPE registers.

3.4. Register definition

PMU base address: 0x4000 7000

3.4.1. Control register 0 (PMU_CTL0)

Address offset: 0x00

Reset value: 0x0002 9000 (reset by wakeup from Standby mode).

This register can be accessed by half-word(16-bit) or word(32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved													LPLDOEN	Reserved	
													rw		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved		DSMODVS[1:0]		Reserved			BKPWEN	Reserved				STBRST	WURST	LPMOD[1:0]	
		rw					rw					rc_w1	rc_w1	rw	

Bits	Fields	Descriptions
31:19	Reserved	Must be kept at reset value.
18	LPLDOEN	Low power LDO enable When setting this bit, the LDO will change from normal LDO to low power LDO. 0: Donot use low power LDO 1: Use low power LDO
17:14	Reserved	Must be kept at reset value.
13:12	DSMODVS[1:0]	Deep-sleep mode voltage selection. These bits control the V_{CORE} voltage level in Deep-sleep / Deep-sleep1 mode to obtain the best trade-off between power consumption and performance. 00: 0.9V 01: 1.0V (Default) 10: 1.1V 11: 1.2V **Note:** 0.9V can only be valid when NPLDO is off.
11:9	Reserved	Must be kept at reset value.
8	BKPWEN	RTC Backup register write enable 0: Disable write access to the Backup registers in RTC. 1: Enable write access to the Backup registers in RTC. After reset, any write access to the registers in RTC Backup registers are ignored. This bit has to be set to enable write access to these registers.
7:4	Reserved	Must be kept at reset value.
3	STBRST	Standby Flag Reset

		0: No effect 1: Reset the standby flag This bit is always read as 0.
2	WURST	Wakeup Flag Reset 0: No effect 1: Reset the wakeup flag This bit is always read as 0.
1:0	LPMOD[1:0]	Select the low-power mode to enter when the Cortex®-M23 enters SLEEPDEEP mode. 00: Deep-sleep 01: Deep-sleep1 10: Reserved 11: Standby

3.4.2. Control and status register (PMU_CS)

Address offset: 0x04

Reset value: 0x0000 0000 (not reset by wakeup from Standby mode).

This register can be accessed by half-word(16-bit) or word(32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															NPRDY
															r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	LDOVSRF	WUPEN5	Reserved	WUPEN3	WUPEN2	WUPEN1	WUPEN0	Reserved						STBF	WUF
	r	rw		rw	rw	rw	rw							r	r

Bits	Fields	Descriptions
31:17	Reserved	Must be kept at reset value.
16	NPRDY	NPLDO ready flag 0: NPLDO is not ready. 1: NPLDO is ready.
15	Reserved	Must be kept at reset value.
14	LDOVSRF	LDO voltage select ready flag. 0: LDO voltage select not ready. 1: LDO voltage select ready.
13	WUPEN5	WKUP Pin5 (PB5) Enable 0: Disable WKUP pin5 function 1: Enable WKUP pin5 function If WUPEN5 is set before entering the power saving mode, a rising edge on the

		WKUP pin5 wakes up the system from the power saving mode. As the WKUP pin5 is active high, the WKUP pin5 is internally configured to input pull down mode. And set this bit will trigger a wakup event when the input is aready high.
12	Reserved	Must be kept at reset value.
11	WUPEN3	WKUP Pin3 (PA2) enable 0: Disable WKUP pin3 function. 1: Enable WKUP pin3 function. If WUPEN3 is set before entering the power saving mode, a rising edge on the WKUP pin3 wakes up the system from the power saving mode. As the WKUP pin3 is active high, the WKUP pin3 is internally configured to input pull down mode. And set this bit will trigger a wakup event when the input is aready high.
10	WUPEN2	WKUP Pin2 (PB6) enable 0: Disable WKUP pin2 function. 1: Enable WKUP pin2 function. If WUPEN2 is set before entering the power saving mode, a rising edge on the WKUP pin2 wakes up the system from the power saving mode. As the WKUP pin2 is active high, the WKUP pin2 is internally configured to input pull down mode. And set this bit will trigger a wakup event when the input is aready high.
9	WUPEN1	WKUP Pin1(PC13 / PA4) enable 0: Disable WKUP pin1 function. 1: Enable WKUP pin1 function. If WUPEN1 is set before entering the power saving mode, a rising edge on the WKUP pin1 wakes up the system from the power saving mode. As the WKUP pin1 is active high, the WKUP pin1 is internally configured to input pull down mode. And set this bit will trigger a wakup event when the input is aready high. **Note**: Only PC13 is available in LQFP48 and QFN48 packages.
8	WUPEN0	WKUP Pin0(PA0) enable 0: Disable WKUP pin0 function. 1: Enable WKUP pin0 function. If WUPEN0 is set before entering the power saving mode, a rising edge on the WKUP pin0 wakes up the system from the power saving mode. As the WKUP pin0 is active high, the WKUP pin0 is internally configured to input pull down mode. And set this bit will trigger a wakup event when the input is aready high.
7:2	Reserved	Must be kept at reset value.
1	STBF	Standby Flag 0: The device has not entered the Standby mode. 1: The device has been in the Standby mode. This bit is cleared only by a POR / PDR or by setting the STBRST bit in the PMU_CTL0 register.
0	WUF	Wakeup Flag

		0: No wakeup event has been received. 1: Wakeup event occurred from the WKUP pins or the RTC wakeup event including RTC alarm event, RTC time stamp event, RTC tamper event or RTC auto wakeup event. This bit is cleared only by a POR / PDR or by setting the WURST bit in the PMU_CTL0 register.

3.4.3. Control register 1 (PMU_CTL1)

Address offset: 0x08

Reset value: 0x0000 0000

This register can be accessed by half-word(16-bit) or word(32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved										EFDSPSLEEP	EFPSLEEP	Reserved			
										rw	rw				

Bits	Fields	Descriptions
31:6	Reserved	Must be kept at reset value.
5	EFDSPSLEEP	EFLASH power off contrl when MCU enters to Deep-sleep/Deep-sleep1 mode. This bit is set by software only in Deep-sleep/Deep-sleep1 mode. 0: EFLASH shall power on. 1: EFLASH shall power off.
4	EFPSLEEP	EFLASH power off control when MCU enters to Run/Run1 mode. This bit is set by software only in Run/Run1 mode before. 0: EFLASH shall power on 1: EFLASH shall power off
3:0	Reserved	Must be kept at reset value.

3.4.4. Status register (PMU_STAT)

Address offset: 0x0C

Reset value: 0x00C0 0020 (not reset by wakeup from Standby mode).

This register can be accessed by half-word(16-bit) or word(32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Reserved	EFLASHPS_ACTIVE	EFLASHPS_SLEEP	Reserved
	r	r	

Bits	Fields	Descriptions
31:6	Reserved	Must be kept at reset value.
5	EFLASHPS_ACTIVE	EFLASH is in active state.
4	EFLASHPS_SLEEP	EFLASH is in sleep state.
3:0	Reserved	Must be kept at reset value.

3.4.5. Parameter register (PMU_PAR)

Address offset: 0x10

Reset value: 0x190A 0000

This register can be accessed by half-word(16-bit) or word(32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved			TWK_EFLASH[7:0]								TSW_IRC48MCNT[4:0]				
			rw								rw				
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved															

Bits	Fields	Descriptions
31:29	Reserved	Must be kept at reset value.
28:21	TWK_EFLASH[7:0]	EFLASH wake up from Deep-sleep/Deep-sleep1 mode counter. When wake from Deep-sleep/Deep-sleep1 mode, wait TWK_EFLASH IRC48M clock periods. The default value is 200 (4.17us).
20:16	TSW_IRC48MCNT[4:0]	When enter Deep-sleep/Deep-sleep1 mode, switch to IRC48M clock. Wait the IRC48M COUNTER and then set Deep-sleep signal. The default value 10.
15:0	Reserved	Must be kept at reset value.

4. Reset and clock unit (RCU)

4.1. Reset control unit (RCTL)

4.1.1. Overview

GD32C2x1 reset control includes the control of three kinds of reset: power reset, system reset and backup register(V_{CORE_STB}) reset. The power on reset, known as a cold reset, resets the full system except the backup register during a power up. A system reset resets the processor core and peripheral IP components with the exception of the SW-DP controller and the backup register. A backup register reset resets the backup register. 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 sections.

4.1.2. Function overview

Power Reset

The power reset is generated by either an external reset as power on and power down reset (POR/PDR reset), or by the internal reset generator when exiting standby mode. The power reset sets all registers to their reset values. The power reset which active signal is low will be de-asserted when the internal LDO voltage regulator is ready to provide 1.2V power. The RESET service routine vector is fixed at address 0x0000_0004 in the memory map.

System Reset

A system reset is generated by the following events:

- A power reset (POWER_RSTn).
- A external pin reset (NRST).
- A window watchdog timer reset (WWDGT_RSTn).
- A free watchdog timer reset (FWDGT_RSTn).
- The SYSRESETREQ bit in Cortex®-M23 application interrupt and reset control register is set (SW_RSTn).
- Option byte loader reset (OBL_RSTn).
- Reset generated when entering Standby mode when resetting nRST_STDBY bit in user option bytes (OB_STDBY_RSTn).
- Reset generated when entering Deep-sleep mode when resetting nRST_DPSLP bit in user option bytes (OB_DPSLP_RSTn).

Note: The NRST pin can be configured with the option byte NRST_MDSEL[1:0] in the following three modes:

1. Input/output mode(default mode): the GPIO function of the NRST pin is not available in

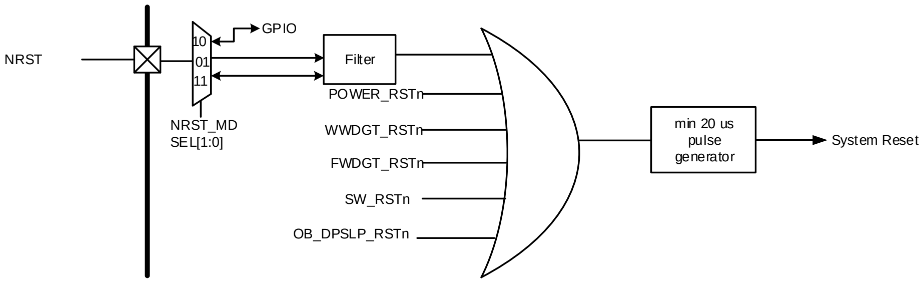
this mode. The reset signal can be transferred from the NRST pin to the device, causing the device to reset, the reset pulse signal can be reflected through the NRST pin, and the minimum reset pulse duration is 20 μs.

2. Input mode: the GPIO function of the NRST pin is not available in this mode, the reset signal can be transferred from the NRST pin to the device, causing the device to reset, but the internal reset of the device is not visible at the NRST pin.
3. GPIO mode: NRST pin can only function as standard GPIO, reset function is not available, reset signal is only inside the device, not reflected in NRST pin.

A system reset resets the processor core and peripheral IP components except for the SW-DP controller and the backup register.

A system reset pulse generator guarantees low level pulse duration of 20 μs for each reset source (external or internal reset).

Figure 4-1. The system reset circuit



Backup register (V_{CORE_STB}) reset

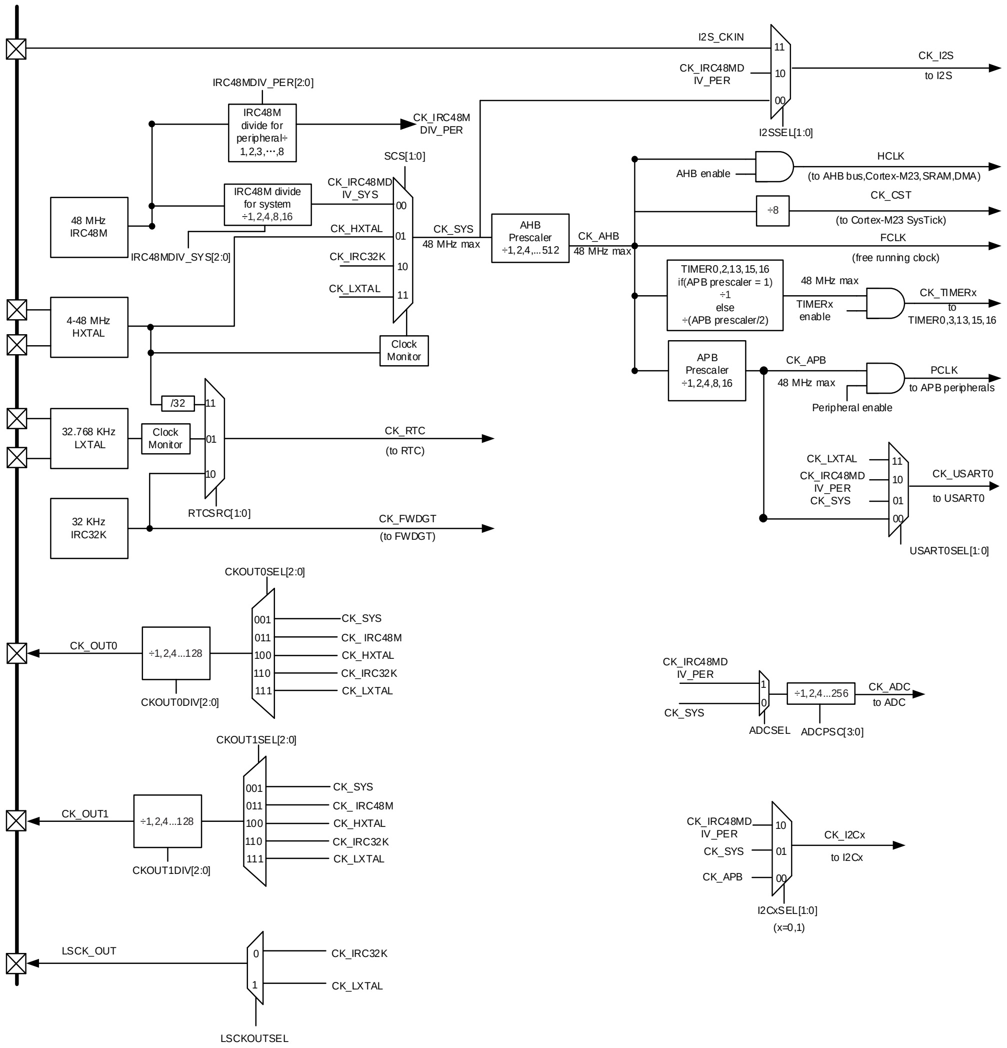
A backup register reset is generated by setting the BKPRST bit in the RCU_CTL1 register or power on reset (V_{DD} power on).

4.2. Clock control unit (CCTL)

4.2.1. Overview

The clock control unit provides a range of frequencies and clock functions. These include an Internal 48 MHz RC oscillator (IRC48M), a High speed crystal oscillator (HXTAL), an Internal 32KHz RC oscillator (IRC32K), a Low speed crystal oscillator (LXTAL), a HXTAL clock monitor, a LXTAL clock monitor, clock prescalers, clock multiplexers and clock gating circuitry.

Figure 4-2. Clock tree



The frequency of AHB and APB domains can be configured by each prescaler. The maximum frequency of the AHB and APB domains is 48MHz / 48MHz. The cortex system timer (SysTick) external clock is clocked with the AHB clock (HCLK) divided by 8. The SysTick can work either with this clock or with the AHB clock (HCLK), configurable in the SysTick control and status register.

The ADC is clocked by the clock of CK_SYS divided by 1, 2, 4, … , 256 or CK_IRC48MDIV_PER divided by 1, 2, 4, … , 256, selected by ADCSEL bit and ADCPSC bits in configuration register 1 (RCU_CFG1)

The USART0 is clocked by IRC48MDIV_PER clock or LXTAL clock or system clock or APB clock, which selected by USART0SEL bits in configuration register 1 (RCU_CFG1).

The I2Cx(x = 0, 1) is clocked by IRC48MDIV_PER clock or system clock or APB clock, which selected by I2CxSEL(x = 0, 1) bits in configuration register 1 (RCU_CFG1).

The RTC is clocked by LXTAL clock or IRC32K clock or HXTAL clock divided by 32 which select by RTCSRC bits in control register 1(RCU_CTL1).

The FWDGT is clocked by IRC32K clock, which is forced on when FWDGT started.

The I2S is clocked by I2S_CKIN clock or CK_IRC48MDIV_PER or CK_SYS, which selected by I2SSEL bits in configuration register 1 (RCU_CFG1).

The TIMERs are clocked by the clock divided from CK_APB. The frequency of TIMERs clock is equal to CK_APB(APB prescaler is 1), twice the CK_APB (APBprescaler is not 1).

4.2.2. Characteristics

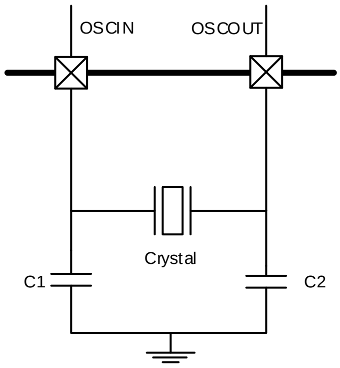
- 4 to 48 MHz High speed crystal oscillator (HXTAL).
- Internal 48 MHz RC oscillator (IRC48M).
- 32.768 KHz Low speed crystal oscillator (LXTAL).
- Internal 32 KHz RC oscillator (IRC32K).
- HXTAL and LXTAL clock monitor.

4.2.3. Function overview

High speed crystal oscillator (HXTAL)

The high speed crystal oscillator (HXTAL), which has a frequency from 4 to 48 MHz, produces a highly accurate clock source for use as the system clock. A crystal with a specific frequency must be connected and located close to the two HXTAL pins. The external resistor and capacitor components connected to the crystal are necessary for proper oscillation.

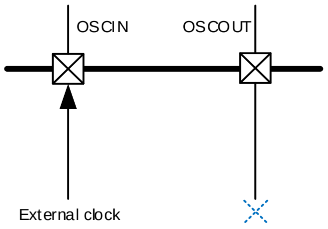
Figure 4-3. HXTAL clock source



The HXTAL crystal oscillator can be switched on or off using the HXTALEN bit in the control register, RCU_CTL0. The HXTALSTB flag in control register, RCU_CTL0 indicates if the high-speed external crystal oscillator is stable. When the HXTAL is powered up, it will not be released for use until this HXTALSTB bit is set by the hardware. This specific delay period is known as the oscillator “Start-up time”. As the HXTAL becomes stable, an interrupt will be generated if the related interrupt enable bit HXTALSTBIE in the interrupt register RCU_INT is set. At this point the HXTAL clock can be used directly as the system clock source.

Select external clock bypass mode by setting the HXTALBPS and HXTALEN bits in the control register RCU_CTL0. During bypass mode, the signal is connected to OSCIN, as shown in ***Figure 4-4. HXTAL clock source in bypass mode***. The CK_HXTAL is equal to the external clock which drives the OSCIN pin.

Figure 4-4. HXTAL clock source in bypass mode



Internal 48M RC oscillators (IRC48M)

The internal 48M RC oscillator, IRC48M, has a fixed frequency of 48 MHz. The IRC48M RC oscillator can be switched on or off using the IRC48MEN bit in the RCU_CTL register. The IRC48MSTB flag in the RCU_CTL register is used to indicate if the internal 48M RC oscillator is stable. An interrupt can be generated if the related interrupt enable bit, IRC48MSTBIE, is set when the IRC48M becomes stable.

Low speed crystal oscillator (LXTAL)

The low speed crystal or ceramic resonator oscillator, which has a frequency of 32,768 Hz, produces a low power but highly accurate clock source for the real time clock circuit. The LXTAL oscillator can be switched on or off using the LXTALEN bit in the control register 1(RCU_CTL1). The LXTALSTB flag in the control register 1(RCU_CTL1) will indicate if the LXTAL clock is stable. An interrupt can be generated if the related interrupt enable bit, LXTALSTBIE, in the Interrupt register RCU_INT is set when the LXTAL becomes stable.

Select external clock bypass mode by setting the LXTALBPS and LXTALEN bits in the control register1 (RCU_CTL1). The CK_LXTAL is equal to the external clock which drives the OSC32IN pin.

Internal 32 KHz RC oscillator (IRC32K)

The Internal 32 KHz RC Oscillator has a frequency of about 32 KHz and is a low power clock source for the real time clock circuit or the free watchdog timer. The IRC32K offers a low cost clock source as no external components are required. The IRC32K RC oscillator can be switched on or off by using the IRC32KEN bit in the reset source / clock register, RCU_RSTSCK. The IRC32KSTB flag in the reset source / clock register RCU_RSTSCK will indicate if the IRC32K clock is stable. An interrupt can be generated if the related interrupt enable bit IRC32KSTBIE in the Interrupt register RCU_INT is set when the IRC32K becomes stable.

System clock (CK_SYS) selection

After the system reset, the default CK_SYS source will be the IRC48M/4 and can be switched to HXTAL, IRC48MDIV_SYS or LXTAL by changing the system clock switch bits, SCS, in the configuration register 0, RCU_CFG0. When the SCS value is changed, the CK_SYS will continue to operate using the original clock source until the target clock source is stable. When a clock source is used directly by the CK_SYS, it is not possible to stop it.

HXTAL clock monitor (CKM)

The HXTAL clock monitor function is enabled by the HXTAL clock monitor enable bit, CKMEN, in the control register, RCU_CTL0. This function should be enabled after the HXTAL start-up delay and disabled when the HXTAL is stopped. Once the HXTAL failure is detected, the HXTAL will be automatically disabled. The HXTAL Clock Stuck Flag, CKMIF, in the interrupt register, RCU_INT, will be set and the HXTAL failure event will be generated. This failure interrupt is connected to the Non-Maskable Interrupt, NMI, of the Cortex-M23. If the HXTAL is selected as the clock source of CK_SYS, the HXTAL failure will force the CK_SYS source to IRC48MDIV_SYS.

LXTAL clock monitor (LCKM)

A clock monitor on LXTAL can be activated by software writing the LCKMEN bit in the control register 1(RCU_CTL1). LCKMEN can not be enabled before LXTAL and IRC32K are enabled and ready.

The clock monitor on LXTAL is working in all modes except V_{CORE_STB}. If a failure is detected on the external 32 KHz oscillator, an interrupt can be sent to CPU. This failure interrupt is connected to the Non-Maskable Interrupt, NMI, of the Cortex-M23. If the LXTAL is selected as the clock source of CK_SYS, the LXTAL failure will force the CK_SYS source to IRC32K.

The software must then disable the LCKMEN bit, stop the defective 32 KHz oscillator, and change the RTC clock source, or take any required action to secure the application.

A 4-bits plus one counter will work at IRC32K domain when LCKMEN enable. If the LXTAL clock has stuck at 0 / 1 error or slow down about 20KHz, the counter will overflow. The LXTAL clock failure will been found.

Clock output capability

The clock output capability is ranging from 32 KHz to 48 MHz. There are several clock signals can be selected via the CK_OUT clock source selection bits, CKOUTxSEL (x = 0,1), in the configuration register 0 (RCU_CFG0). The corresponding GPIO pin should be configured in the properly alternate function I / O (AFIO) mode to output the selected clock signal.

Table 4-1. Clock source select

Clock source selection bits	Clock source
000	No Clock

Clock source selection bits	Clock source
001	CK_SYS
010	Reserved
011	CK_IRC48M
100	CK_HXTAL
101	Reserved
110	CK_IRC32K
111	CK_LXTAL

The CK_OUT frequency can be reduced by a configurable binary divider, controlled by the CKOUTxDIV[2:0] bits(x = 0,1), in the configuration register 0(RCU_CFG0).

The CK_LXTAL and CK_IRC32K clock signals also can be output on LSCK_OUT pin, even in deep-sleep / deep-sleep 1 mode and standby mode, which seleted by LSCKOUTSEL in the control register 1 (RCU_CTL1).

Table 4-2. Low-speed clock output source select

Clock source selection bits	Clock source
0	CK_IRC32K
1	CK_LXTAL

Deep-sleep mode clock control

When the MCU is in deep-sleep / deep-sleep 1 mode, the USART0 can wake up the MCU, when their clock is provided by LXTAL clock and LXTAL clock is enable.

If the USART0 clock is selected IRC48MDIV_PER clock in deep-sleep / deep-sleep 1 mode, they have capable of open IRC48M clock or close IRC48M clock, which used to the USART0 to wake up the Deep-sleep mode.

If the USART0 clock is selected LXTAL clock in deep-sleep / deep-sleep 1 mode, they have capable of open LXTAL clock or close LXTAL clock (if LXTAL is opened by softer, USART0 can’t close the LXTAL).

If the I2C0 / I2C1 clock is selected IRC48MDIV_PER clock in deep-sleep / deep-sleep 1 mode, they have capable of open IRC48M clock or close IRC48M clock, which used to the I2C0 / I2C1 to wake up the deep-sleep / deep-sleep 1 mode.

FMC and PMU also have capable of open IRC48M clock or close IRC48M clock, if they work in deep-sleep / deep-sleep 1 mode.

To save power in deep-sleep / deep-sleep 1 mode. FMC and USART0 function clock can be gated individually, if they don’t work in deep-sleep / deep-sleep 1 mode mode. But I2C0 / I2C1, ADC, PMU function clock can’t be gated by hardware, which can be disable by software.

4.3. Register definition

RCU base address: 0x4002 1000

4.3.1. Control register 0 (RCU_CTL0)

Address offset: 0x00

Reset value: 0x4400 XX43 where X is undefined.

This register can be accessed by byte(8-bit), half-word(16-bit) and word(32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IRC48MDIV_SYS[2:0]			IRC48M_PEREN	IRC48MDIV_PER[2:0]			Reserved					CKMEN	HXTALBPS	HXTALSTB	HXTALEN
rw			rw	rw								rw	rw	r	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IRC48MCALIB[7:0]								IRC48MADJ[5:0]						IRC48MSTB	IRC48MEN
r								rw						r	rw

Bits	Fields	Descriptions
31:29	IRC48MDIV_SYS[2:0]	IRC48M clock divider factor for system clock These bits controlled by software sets the division factor of the IRC48M clock divider to produce CK_IRC48MDIV_SYS clock: 000: CK_IRC48MDIV_SYS = CK_IRC48M / 1 001: CK_IRC48MDIV_SYS = CK_IRC48M / 2 010: CK_IRC48MDIV_SYS = CK_IRC48M / 4 (reset value) 011: CK_IRC48MDIV_SYS= CK_IRC48M / 8 100: CK_IRC48MDIV_SYS = CK_IRC48M / 16 101: CK_IRC48MDIV_SYS = CK_IRC48M / 32 110: CK_IRC48MDIV_SYS = CK_IRC48M / 64 111: CK_IRC48MDIV_SYS = CK_IRC48M / 128
28	IRC48M_PEREN	IRC48M always-enable for peripheral. Set and cleared by software. Setting the bit activates the IRC48M oscillator in Run and deep-sleep modes, regardless of the IRC48MEN bit state. The IRC48M clock can only feed USART0, I2C0 and I2C1 peripherals configured with IRC48M as kernel clock. 0: IRC48M oscillator enable depends on the IRC48MEN bit 1: IRC48M oscillator is active in Run and deep-sleep modes **Note**: Keeping the IRC48M active in deep-sleep mode allows speeding up the serial interface communication as the IRC48M clock is ready immediately upon exiting deep-sleep mode.
27:25	IRC48MDIV_PER[2:0]	IRC48M clock divider factor for peripheral clock These bits controlled by software sets the division factor of the peripheral clock

		divider to produce CK_IRC48MDIV_PER clock. 000: CK_IRC48MDIV_PER = CK_IRC48M / 1 001: CK_IRC48MDIV_PER = CK_IRC48M / 2 010: CK_IRC48MDIV_PER = CK_IRC48M / 3 (reset value) 011: CK_IRC48MDIV_PER = CK_IRC48M / 4 100: CK_IRC48MDIV_PER = CK_IRC48M / 5 101: CK_IRC48MDIV_PER = CK_IRC48M / 6 110: CK_IRC48MDIV_PER = CK_IRC48M / 7 111: CK_IRC48MDIV_PER = CK_IRC48M / 8
24:20	Reserved	Must be kept at reset value.
19	CKMEN	HXTAL clock monitor enable 0: Disable the external 4 ~ 48 MHz crystal oscillator (HXTAL) clock monitor 1: Enable the external 4 ~ 48 MHz crystal oscillator (HXTAL) clock monitor When the hardware detects that the HXTAL clock is stuck at a low or high state, the internal hardware will switch the system clock to be the internal high speed IRC48M RC clock. The way to recover the original system clock is by either an external reset, power on reset or clearing CKMIF by software. **Note**: When the HXTAL clock monitor is enabled, the hardware will automatically enable the IRC48M internal RC oscillator regardless of the control bit, IRC48MEN, state.
18	HXTALBPS	External crystal oscillator (HXTAL) clock bypass mode enable The HXTALBPS bit can be written only if the HXTALEN is 0 0: Disable the HXTAL bypass mode 1: Enable the HXTAL bypass mode in which the HXTAL output clock is equal to the input clock
17	HXTALSTB	External crystal oscillator (HXTAL) clock stabilization flag Set by hardware to indicate if the HXTAL oscillator is stable and ready for use. 0: HXTAL oscillator is not stable 1: HXTAL oscillator is stable
16	HXTALEN	External high speed oscillator enable Set and reset by software. This bit cannot be reset if the HXTAL clock is used as the system clock. Reset by hardware when entering Deep-sleep or Standby mode. 0: External 4 ~ 48 MHz crystal oscillator disabled 1: External 4 ~ 48 MHz crystal oscillator enabled
15:8	IRC48MCALIB[7:0]	Internal 48M RC oscillator calibration value register These bits are load automatically at power on.
7:2	IRC48MADJ[5:0]	Internal 48M RC oscillator clock trim adjust value These bits are set by software. The trimming value is there bits (IRC48MADJ) added to the IRC48MCALIB[7:0] bits. The trimming value should trim the IRC48M to 48 MHz ± 1%.

1	IRC48MSTB	IRC48M high speed internal oscillator stabilization flag Set by hardware to indicate if the IRC48M oscillator is stable and ready for use. 0: IRC48M oscillator is not stable 1: IRC48M oscillator is stable
0	IRC48MEN	Internal high speed oscillator enable Set and reset by software. This bit cannot be reset if the IRC48M clock is used as the system clock. Set by hardware when leaving Deep-sleep or Standby mode or the HXTAL clock is stuck at a low or high state when HXTALCKM is set. 0: Internal 48 MHz RC oscillator disabled 1: Internal 48 MHz RC oscillator enabled

4.3.2. Configuration register 0 (RCU_CFG0)

Address offset: 0x08

Reset value: 0x0000 0000

This register can be accessed by byte(8-bit), half-word(16-bit) and word(32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved	CKOUT0DIV[2:0]			Reserved	CKOUT0SEL[2:0]			Reserved	CKOUT1DIV[2:0]			Reserved	CKOUT1SEL[2:0]		
	rw				rw				rw				rw		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved		APBPSC[2:0]			Reserved			AHBPSC[3:0]				SCSS[1:0]		SCS[1:0]	
		rw						rw				r		rw	

Bits	Fields	Descriptions
31	Reserved	Must be kept at reset value.
30:28	CKOUT0DIV[2:0]	The CK_OUT0 divider which the CK_OUT0 frequency can be reduced, see bits 26:24 of RCU_CFG0 for CK_OUT0. 000: The CK_OUT0 is divided by 1 001: The CK_OUT0 is divided by 2 010: The CK_OUT0 is divided by 4 011: The CK_OUT0 is divided by 8 100: The CK_OUT0 is divided by 16 101: The CK_OUT0 is divided by 32 110: The CK_OUT0 is divided by 64 111: The CK_OUT0 is divided by 128
27	Reserved	Must be kept at reset value.
26:24	CKOUT0SEL[2:0]	CK_OUT0 clock source selection Set and reset by software. 000: No clock selected 001: System clock selected 010: Reserved

		011: Internal 48MHz RC oscillator clock selected 100: External high speed oscillator clock selected 101: Reserved 110: Internal 32KHz RC oscillator clock selected 111: External low speed oscillator clock selected
23	Reserved	Must be kept at reset value.
22:20	CKOUT1DIV[2:0]	The CK_OUT1 divider which the CK_OUT1 frequency can be reduced, see bits 18:16 of RCU_CFG0 for CK_OUT1. 000: The CK_OUT1 is divided by 1 001: The CK_OUT1 is divided by 2 010: The CK_OUT1 is divided by 4 011: The CK_OUT1 is divided by 8 100: The CK_OUT1 is divided by 16 101: The CK_OUT1 is divided by 32 110: The CK_OUT1 is divided by 64 111: The CK_OUT1 is divided by 128
19	Reserved	Must be kept at reset value.
18:16	CKOUT1SEL[2:0]	CK_OUT1 clock source selection Set and reset by software. 000: No clock selected 001: System clock selected 010: Reserved 011: Internal 48MHz RC oscillator clock selected 100: External high speed oscillator clock selected 101: Reserved 110: Internal 32KHz RC oscillator clock selected 111: External low speed oscillator clock selected
15:14	Reserved	Must be kept at reset value.
13:11	APBPSC[2:0]	APB prescaler selection Set and reset by software to control the APB clock division ratio. 0xx: CK_AHB selected 100: (CK_AHB / 2) selected 101: (CK_AHB / 4) selected 110: (CK_AHB / 8) selected 111: (CK_AHB / 16) selected
10:8	Reserved	Must be kept at reset value.
7:4	AHBPSC[3:0]	AHB prescaler selection Set and reset by software to control the AHB clock division ratio 0xxx: CK_SYS selected 1000: (CK_SYS / 2) selected

		1001: (CK_SYS / 4) selected 1010: (CK_SYS / 8) selected 1011: (CK_SYS / 16) selected 1100: (CK_SYS / 64) selected 1101: (CK_SYS / 128) selected 1110: (CK_SYS / 256) selected 1111: (CK_SYS / 512) selected
3:2	SCSS[1:0]	System clock switch status Set and reset by hardware to indicate the clock source of system clock. 00: select CK_IRC48MDIV_SYS as the CK_SYS source 01: Select CK_HXTAL as the CK_SYS source 10: select CK_IRC32K as the CK_SYS source 11: Select CK_LXTAL as the CK_SYS source
1:0	SCS[1:0]	System clock switch Set by software to select the CK_SYS source. Because the change of CK_SYS has inherent latency, software should read SCSS to confirm whether the switching is complete or not. 00: Select CK_IRC48MDIV_SYS as the CK_SYS source 01: Select CK_HXTAL as the CK_SYS source 10: Select IRC32K as the CK_SYS source 11: Select LXTAL as the CK_SYS source

4.3.3. Interrupt register (RCU_INT)

Address offset: 0x0C

Reset value: 0x0000 0000

This register can be accessed by byte(8-bit), half-word(16-bit) and word(32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved								CKMIC	LCKMIC	Reserved		HXTAL STBIC	IRC48M STBIC	LXTAL STBIC	IRC32K STBIC
								w	w			w	w	w	w

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				HXTAL STBIE	IRC48M STBIE	LXTAL STBIE	IRC32K STBIE	CKMIF	LCKMIF	Reserved		HXTAL STBIF	IRC48M STBIF	LXTAL STBIF	IRC32K STBIF
				rw	rw	rw	rw	r	r			r	r	r	r

Bits	Fields	Descriptions
31:24	Reserved	Must be kept at reset value.
23	CKMIC	HXTAL clock stuck interrupt clear Write 1 by software to reset the CKMIF flag. 0: Not reset CKMIF flag

		1: Reset CKMIF flag
22	LCKMIC	LXTAL clock stuck interrupt clear Write 1 by software to reset the LCKMIF flag. 0: Not reset LCKMIF flag 1: Reset LCKMIF flag
21:20	Reserved	Must be kept at reset value.
19	HXTALSTBIC	HXTAL stabilization interrupt clear Write 1 by software to reset the HXTALSTBIF flag. 0: Not reset HXTALSTBIF flag 1: Reset HXTALSTBIF flag
18	IRC48MSTBIC	IRC48M stabilization interrupt clear Write 1 by software to reset the IRC48MSTBIF flag. 0: Not reset IRC48MSTBIF flag 1: Reset IRC48MSTBIF flag
17	LXTALSTBIC	LXTAL stabilization interrupt clear Write 1 by software to reset the LXTALSTBIF flag. 0: Not reset LXTALSTBIF flag 1: Reset LXTALSTBIF flag
16	IRC32KSTBIC	IRC32K stabilization interrupt clear Write 1 by software to reset the IRC32KSTBIF flag. 0: Not reset IRC32KSTBIF flag 1: Reset IRC32KSTBIF flag
15:12	Reserved	Must be kept at reset value.
11	HXTALSTBIE	HXTAL stabilization interrupt enable Set and reset by software to enable/disable the HXTAL stabilization interrupt 0: Disable the HXTAL stabilization interrupt 1: Enable the HXTAL stabilization interrupt
10	IRC48MSTBIE	IRC48M stabilization interrupt enable Set and reset by software to enable/disable the IRC48M stabilization interrupt 0: Disable the IRC48M stabilization interrupt 1: Enable the IRC48M stabilization interrupt
9	LXTALSTBIE	LXTAL stabilization interrupt enable LXTAL stabilization interrupt enable/disable control 0: Disable the LXTAL stabilization interrupt 1: Enable the LXTAL stabilization interrupt
8	IRC32KSTBIE	IRC32K stabilization interrupt enable IRC32K stabilization interrupt enable/disable control 0: Disable the IRC32K stabilization interrupt

		1: Enable the IRC32K stabilization interrupt
7	CKMIF	HXTAL clock stuck interrupt flag Set by hardware when the HXTAL clock is stuck. Reset by software when setting the CKMIC bit. 0: Clock operating normally 1: HXTAL clock stuck
6	LCKMIF	LXTAL clock stuck interrupt flag Set by hardware when the LXTAL clock is stuck. Reset by software when setting the LCKMIC bit. 0: LXTALclock operating normally 1: LXTAL clock stuck
5:4	Reserved	Must be kept at reset value.
3	HXTALSTBIF	HXTAL stabilization interrupt flag Set by hardware when the external 4 ~ 32 MHz crystal oscillator clock is stable and the HXTALSTBIE bit is set. Reset by software when setting the HXTALSTBIC bit. 0: No HXTAL stabilization interrupt generated 1: HXTAL stabilization interrupt generated
2	IRC48MSTBIF	IRC48M stabilization interrupt flag Set by hardware when the internal 48 MHz RC oscillator clock is stable and the IRC48MSTBIE bit is set. Reset by software when setting the IRC48MSTBIC bit. 0: No IRC48M stabilization interrupt generated 1: IRC48M stabilization interrupt generated
1	LXTALSTBIF	LXTAL stabilization interrupt flag Set by hardware when the external 32,768 Hz crystal oscillator clock is stable and the LXTALSTBIE bit is set. Reset by software when setting the LXTALSTBIC bit. 0: No LXTAL stabilization interrupt generated 1: LXTAL stabilization interrupt generated
0	IRC32KSTBIF	IRC32K stabilization interrupt flag Set by hardware when the internal 32KHz RC oscillator clock is stable and the IRC32KSTBIE bit is set. Reset by software when setting the IRC32KSTBIC bit. 0: No IRC32K stabilization clock ready interrupt generated 1: IRC32K stabilization interrupt generated

4.3.4. AHB1 reset register (RCU_AHB1RST)

Address offset: 0x10

Reset value: 0x0000 0000

This register can be accessed by byte(8-bit), half-word(16-bit) and word(32-bit).

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="8">Reserved</td><td>DMAMUX
RST</td><td>Reserved</td><td>DMARST</td><td colspan="5">Reserved</td></tr>
<tr><td colspan="8"></td><td>rw</td><td></td><td>rw</td><td colspan="5"></td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="3">Reserved</td><td>CRCRST</td><td colspan="12">Reserved</td></tr>
<tr><td colspan="3"></td><td>rw</td><td colspan="12"></td></tr>
</table>

Bits	Fields	Descriptions
31:24	Reserved	Must be kept at reset value.
23	DMAMUXRST	DMAMUX reset This bit is set and reset by software. 0: No reset 1: Reset the DMAMUX
22	Reserved	Must be kept at reset value.
21	DMARST	DMA reset This bit is set and reset by software. 0: No reset 1: Reset the DMA
20:13	Reserved	Must be kept at reset value.
12	CRCRST	CRC Reset This bit is set and reset by software. 0: No reset 1: Reset the CRC
11:0	Reserved	Must be kept at reset value.

4.3.5. AHB2 reset register (RCU_AHB2RST)

Address offset: 0x14

Reset value: 0x0000 0000

This register can be accessed by byte(8-bit), half-word(16-bit) and word(32-bit).

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="9">Reserved</td><td>PFRST</td><td>Reserved</td><td>PDRST</td><td>PCRST</td><td>PBRST</td><td>PARST</td><td>Reserved</td></tr>
<tr><td colspan="9"></td><td>rw</td><td></td><td>rw</td><td>rw</td><td>rw</td><td>rw</td><td></td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="16">Reserved</td></tr>
</table>

Bits	Fields	Descriptions
31:23	Reserved	Must be kept at reset value.
22	PFRST	GPIO port F reset This bit is set and reset by software. 0: No reset GPIO port F 1: Reset GPIO port F
21	Reserved	Must be kept at reset value.
20	PDRST	GPIO port D reset This bit is set and reset by software. 0: No reset GPIO port D 1: Reset GPIO port D
19	PCRST	GPIO port C reset This bit is set and reset by software. 0: No reset GPIO port C 1: Reset GPIO port C
18	PBRST	GPIO port B reset This bit is set and reset by software. 0: No reset GPIO port B 1: Reset GPIO port B
17	PARST	GPIO port A reset This bit is set and reset by software. 0: No reset GPIO port A 1: Reset GPIO port A
16:0	Reserved	Must be kept at reset value.

4.3.6. APB reset register (RCU_APBRST)

Address offset: 0x24

Reset value: 0x0000 0000

This register can be accessed by byte(8-bit), half-word(16-bit) and word(32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved			PMURST	Reserved					I2C1RST	I2C0RST	Reserved	USART2RST	TIMER16RST	TIMER15RST	TIMER13RST
			rw						rw	rw		rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
USART1RST	USART0RST	SPI1RST	SPI0RST	TIMER2RST	TIMER0RST	ADCRST	WWDGTRST	Reserved						CMPRST	SYSCFGRST
rw	rw	rw	rw	rw	rw	rw	rw							rw	rw

Bits	Fields	Descriptions
31:29	Reserved	Must be kept at reset value.
28	PMURST	PMU reset This bit is set and reset by software. 0: No reset 1: Reset the PMU
27:23	Reserved	Must be kept at reset value.
22	I2C1RST	I2C1 reset This bit is set and reset by software. 0: No reset 1: Reset the I2C1
21	I2C0RST	I2C0 reset This bit is set and reset by software. 0: No reset 1: Reset the I2C0
20	Reserved	Must be kept at reset value.
19	USART2RST	USART2 Reset This bit is set and reset by software. 0: No reset 1: Reset the USART2
18	TIMER16RST	TIMER16 reset This bit is set and reset by software. 0: No reset 1: Reset the TIMER16
17	TIMER15RST	TIMER15 reset This bit is set and reset by software. 0: No reset 1: Reset the TIMER15
16	TIMER13RST	TIMER13 reset This bit is set and reset by software. 0: No reset 1: Reset the TIMER13
15	USART1RST	USART1 Reset This bit is set and reset by software. 0: No reset 1: Reset the USART1
14	USART0RST	USART0 Reset This bit is set and reset by software. 0: No reset

		1: Reset the USART0
13	SPI1RST	SPI1 Reset This bit is set and reset by software. 0: No reset 1: Reset the SPI1
12	SPI0RST	SPI0 Reset This bit is set and reset by software. 0: No reset 1: Reset the SPI0
11	TIMER2RST	TIMER2 reset This bit is set and reset by software. 0: No reset 1: Reset the TIMER2
10	TIMER0RST	TIMER0 reset This bit is set and reset by software. 0: No reset 1: Reset the TIMER0
9	ADCRST	ADC reset This bit is set and reset by software. 0: No reset 1: Reset the ADC
8	WWDGTRST	WWDGT reset This bit is set and reset by software. 0: No reset 1: Reset the WWDGT
7:2	Reserved	Must be kept at reset value.
1	CMPRST	Comparator reset This bit is set and reset by software. 0: No reset 1: Reset comparator
0	SYSCFGRST	System configuration reset This bit is set and reset by software. 0: No reset 1: Reset system configuration

4.3.7. AHB1 enable register (RCU_AHB1EN)

Address offset: 0x30

Reset value: 0x0000 0010

This register can be accessed by byte(8-bit), half-word(16-bit) and word(32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved								DMAMUX EN	Reserved	DMAEN	Reserved				
								rw		rw					

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved			CRCEN	Reserved							FMCEN	Reserved			
			rw												

Bits	Fields	Descriptions
31:24	Reserved	Must be kept at reset value.
23	DMAMUXEN	DMAMUX clock enable This bit is set and reset by software. 0: Disabled DMAMUX clock 1: Enabled DMAMUX clock
22	Reserved	Must be kept at reset value.
21	DMAEN	DMA clock enable This bit is set and reset by software. 0: Disabled DMA clock 1: Enabled DMA clock
20:13	Reserved	Must be kept at reset value.
12	CRCEN	CRC clock enable This bit is set and reset by software. 0: Disabled CRC clock 1: Enabled CRC clock
11:5	Reserved	Must be kept at reset value.
4	FMCEN	FMC clock enable This bit is set and reset by software. 0: Disabled FMC clock 1: Enabled FMC clock This bit can only be cleared when the flash memory is in power down mode.
3:0	Reserved	Must be kept at reset value.

4.3.8. AHB2 enable register (RCU_AHB2EN)

Address offset: 0x34

Reset value: 0x0000 0000

This register can be accessed by byte(8-bit), half-word(16-bit) and word(32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

Reserved									PFEN	Reserved	PDEN	PCEN	PBEN	PAEN	Reserved
									rw		rw	rw	rw	rw	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															

Bits	Fields	Descriptions
31:23	Reserved	Must be kept at reset value.
22	PFEN	GPIO port F clock enable This bit is set and reset by software. 0: Disabled GPIO port F clock 1: Enabled GPIO port F clock
21	Reserved	Must be kept at reset value.
20	PDEN	GPIO port D clock enable This bit is set and reset by software. 0: Disabled GPIO port D clock 1: Enabled GPIO port D clock
19	PCEN	GPIO port C clock enable This bit is set and reset by software. 0: Disabled GPIO port C clock 1: Enabled GPIO port C clock
18	PBEN	GPIO port B clock enable This bit is set and reset by software. 0: Disabled GPIO port B clock 1: Enabled GPIO port B clock
17	PAEN	GPIO port A clock enable This bit is set and reset by software. 0: Disabled GPIO port A clock 1: Enabled GPIO port A clock
16:0	Reserved	Must be kept at reset value.

4.3.9. APB enable register (RCU_APBEN)

Address offset: 0x44

Reset value: 0x0000 0000

This register can be accessed by byte(8-bit), half-word(16-bit) and word(32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved			PMUEN	DBGEN	Reserved				I2C1EN	I2C0EN	Reserved	USART2EN	TIMER16EN	TIMER15EN	TIMER13EN

			rw	rw					rw	rw		rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
USART1EN	USART0EN	SPI1EN	SPI0EN	TIMER2EN	TIMER0EN	ADCEN	WWDGTEN	Reserved						CMPEN	SYSCFGEN
rw	rw	rw	rw	rw	rw	rw	rw							rw	rw

Bits	Fields	Descriptions
31:29	Reserved	Must be kept at reset value.
28	PMUEN	PMU clock enable This bit is set and reset by software. 0: Disabled PMU clock 1: Enabled PMU clock
27	DBGEN	DBG clock enable This bit is set and reset by software. 0: Disabled DBG clock 1: Enabled DBG clock
26:23	Reserved	Must be kept at reset value.
22	I2C1EN	I2C1 clock enable This bit is set and reset by software. 0: Disabled I2C1 clock 1: Enabled I2C1 clock
21	I2C0EN	I2C0 clock enable This bit is set and reset by software. 0: Disabled I2C0 clock 1: Enabled I2C0 clock
20	Reserved	Must be kept at reset value.
19	USART2EN	USART2 clock enable This bit is set and reset by software. 0: Disabled USART2 clock 1: Enabled USART2 clock
18	TIMER16EN	TIMER16 timer clock enable This bit is set and reset by software. 0: Disabled TIMER16 timer clock 1: Enabled TIMER16 timer clock
17	TIMER15EN	TIMER15 timer clock enable This bit is set and reset by software. 0: Disabled TIMER15 timer clock 1: Enabled TIMER15 timer clock
16	TIMER13EN	TIMER13 timer clock enable

		This bit is set and reset by software. 0: Disabled TIMER13 timer clock 1: Enabled TIMER13 timer clock
15	USART1EN	USART1 clock enable This bit is set and reset by software. 0: Disabled USART1 clock 1: Enabled USART1 clock
14	USART0EN	USART0 clock enable This bit is set and reset by software. 0: Disabled USART0 clock 1: Enabled USART0 clock
13	SPI1EN	SPI1 clock enable This bit is set and reset by software. 0: Disabled SPI1 clock 1: Enabled SPI1 clock
12	SPI0EN	SPI0 clock enable This bit is set and reset by software. 0: Disabled SPI0 clock 1: Enabled SPI0 clock
11	TIMER2EN	TIMER2 timer clock enable This bit is set and reset by software. 0: Disabled TIMER2 timer clock 1: Enabled TIMER2 timer clock
10	TIMER0EN	TIMER0 timer clock enable This bit is set and reset by software. 0: Disabled TIMER0 timer clock 1: Enabled TIMER0 timer clock
9	ADCEN	ADC interface clock enable This bit is set and reset by software. 0: Disabled ADC interface clock 1: Enabled ADC interface clock
8	WWDGTEN	WWDGT clock enable This bit is set and reset by software. 0: Disabled WWDGT clock 1: Enabled WWDGT clock
7:2	Reserved	Must be kept at reset value.
1	CMPEN	Comparator clock enable This bit is set and reset by software. 0: Disabled system comparator clock

		1: Enabled comparator clock
0	SYSCFGEN	System configuration clock enable This bit is set and reset by software. 0: Disabled system configuration clock 1: Enabled system configuration clock

4.3.10. AHB1 sleep and deep sleep mode enable register (RCU_AHB1SPDPEN)

Address offset: 0x50

Reset value: 0x00A0 1014

This register can be accessed by byte(8-bit), half-word(16-bit) and word(32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved								DMAMUX SPDPEN	Reserved	DMASPD PEN	Reserved				
								rw		rw					

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved			CRCSPD PEN	Reserved							FMCSPD PEN	Reserved	SRAMSP DPEN	Reserved	
			rw								rw		rw		

Bits	Fields	Descriptions
31:24	Reserved	Must be kept at reset value.
23	DMAMUXSPDPEN	DMAMUX clock enable when sleep and deep sleep mode This bit is set and reset by software. 0: Disabled DMAMUX clock when sleep and deep sleep mode 1: Enabled DMAMUX clock when sleep and deep sleep mode
22	Reserved	Must be kept at reset value.
21	DMASPDPEN	DMA clock enable when sleep and deep sleep mode This bit is set and reset by software. 0: Disabled DMA clock when sleep and deep sleep mode 1: Enabled DMA clock when sleep and deep sleep mode
20:13	Reserved	Must be kept at reset value.
12	CRCSPDPEN	CRC clock enable when sleep and deep sleep mode This bit is set and reset by software. 0: Disabled CRC clock when sleep and deep sleep mode 1: Enabled CRC clock when sleep and deep sleep mode
11:5	Reserved	Must be kept at reset value.
4	FMCSPDPEN	FMC clock enable when sleep and deep sleep mode This bit is set and reset by software.

		0: Disabled FMC clock when sleep and deep sleep mode 1: Enabled FMC clock when sleep and deep sleep mode
3	Reserved	Must be kept at reset value.
2	SRAMSPDPEN	SRAM interface clock enable when sleep and deep sleep mode This bit is set and reset by software. 0: Disabled SRAM interface clock when sleep and deep sleep mode 1: Enabled SRAM interface clock when sleep and deep sleep mode
1:0	Reserved	Must be kept at reset value.

4.3.11. AHB2 sleep and deep sleep mode enable register (RCU_AHB2SPDPEN)

Address offset: 0x54

Reset value: 0x005E 0000

This register can be accessed by byte(8-bit), half-word(16-bit) and word(32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved									PFSPDPEN	Reserved	PDSPDPEN	PCSPDPEN	PBSPDPEN	PASPDPEN	Reserved
									rw		rw	rw	rw	rw	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															

Bits	Fields	Descriptions
31:23	Reserved	Must be kept at reset value.
22	PFSPDPEN	GPIO port F clock enable when sleep and deep sleep mode This bit is set and reset by software. 0: Disabled GPIO port F clock when sleep and deep sleep mode 1: Enabled GPIO port F clock when sleep and deep sleep mode
21	Reserved	Must be kept at reset value.
20	PDSPDPEN	GPIO port D clock enable when sleep and deep sleep mode This bit is set and reset by software. 0: Disabled GPIO port D clock when sleep and deep sleep mode 1: Enabled GPIO port D clock when sleep and deep sleep mode
19	PCSPDPEN	GPIO port C clock enable when sleep and deep sleep mode This bit is set and reset by software. 0: Disabled GPIO port C clock when sleep and deep sleep mode 1: Enabled GPIO port C clock when sleep and deep sleep mode
18	PBSPDPEN	GPIO port B clock enable when sleep and deep sleep mode This bit is set and reset by software.

Bits	Fields	Descriptions
		0: Disabled GPIO port B clock when sleep and deep sleep mode 1: Enabled GPIO port B clock when sleep and deep sleep mode
17	PASPDPEN	GPIO port A clock enable when sleep and deep sleep mode This bit is set and reset by software. 0: Disabled GPIO port A clock when sleep and deep sleep mode 1: Enabled GPIO port A clock when sleep and deep sleep mode
16:0	Reserved	Must be kept at reset value.

4.3.12. APB sleep and deep sleep mode enable register (RCU_APBSPDPEN)

Address offset: 0x64

Reset value: 0x106F FF03

This register can be accessed by byte(8-bit), half-word(16-bit) and word(32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved			PMUSPD PEN	Reserved					I2C1SPD PEN	I2C0SPD PEN	Reserved	USART2 SPDPEN	TIMER16 SPDPEN	TIMER15 SPDPEN	TIMER13 SPDPEN
			rw						rw	rw		rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
USART1 SPDPEN	USART0 SPDPEN	SPI1SPD PEN	SPI0SPD PEN	TIMER2S PDPEN	TIMER0S PDPEN	ADCSPD PEN	WWDGT SPDPEN	Reserved						CMPSPD PEN	SYSCFG SPDPEN
rw	rw	rw	rw	rw	rw	rw	rw							rw	rw

Bits	Fields	Descriptions
31:29	Reserved	Must be kept at reset value.
28	PMUSPDPEN	PMU clock enable when sleep and deep sleep mode This bit is set and reset by software. 0: Disabled PMU clock when sleep and deep sleep mode 1: Enabled PMU clock when sleep and deep sleep mode
27:23	Reserved	Must be kept at reset value.
22	I2C1SPDPEN	I2C1 clock enable when sleep mode and deep sleep mode This bit is set and reset by software. 0: Disabled I2C1 clock when sleep mode and deep sleep mode 1: Enabled I2C1 clock when sleep mode and deep sleep mode
21	I2C0SPDPEN	I2C0 clock enable when sleep mode and deep sleep mode This bit is set and reset by software. 0: Disabled I2C0 clock when sleep mode and deep sleep mode 1: Enabled I2C0 clock when sleep mode and deep sleep mode
20	Reserved	Must be kept at reset value.
19	USART2SPDPEN	USART2 clock enable when sleep mode and deep sleep mode

		This bit is set and reset by software. 0: Disabled USART2 clock when sleep mode and deep sleep mode 1: Enabled USART2 clock when sleep mode and deep sleep mode
18	TIMER16SPDPEN	TIMER16 clock enable when sleep and deep sleep mode This bit is set and reset by software. 0: Disabled TIMER16 clock when sleep and deep sleep mode 1: Enabled TIMER16 clock when sleep and deep sleep mode
17	TIMER15SPDPEN	TIMER15 clock enable when sleep and deep sleep mode This bit is set and reset by software. 0: Disabled TIMER15 clock when sleep and deep sleep mode 1: Enabled TIMER15 clock when sleep and deep sleep mode
16	TIMER13SPDPEN	TIMER13 clock enable when sleep and deep sleep mode This bit is set and reset by software. 0: Disabled TIMER13 clock when sleep and deep sleep mode 1: Enabled TIMER13 clock when sleep and deep sleep mode
15	USART1SPDPEN	USART1 clock enable when sleep and deep sleep mode This bit is set and reset by software. 0: Disabled USART1 clock when sleep and deep sleep mode 1: Enabled USART1 clock when sleep and deep sleep mode
14	USART0SPDPEN	USART0 clock enable when sleep and deep sleep mode This bit is set and reset by software. 0: Disabled USART0 clock when sleep and deep sleep mode 1: Enabled USART0 clock when sleep and deep sleep mode
13	SPI1SPDPEN	SPI1 clock enable when sleep and deep sleep mode This bit is set and reset by software. 0: Disabled SPI1 clock when sleep and deep sleep mode 1: Enabled SPI1 clock when sleep and deep sleep mode
12	SPI0SPDPEN	SPI0 clock enable when sleep and deep sleep mode This bit is set and reset by software. 0: Disabled SPI0 clock when sleep and deep sleep mode 1: Enabled SPI0 clock when sleep and deep sleep mode
11	TIMER2SPDPEN	TIMER2 clock enable when sleep and deep sleep mode This bit is set and reset by software. 0: Disabled TIMER2 clock when sleep and deep sleep mode 1: Enabled TIMER2 clock when sleep and deep sleep mode
10	TIMER0SPDPEN	TIMER0 clock enable when sleep and deep sleep mode This bit is set and reset by software. 0: Disabled TIMER0 clock when sleep and deep sleep mode 1: Enabled TIMER0 clock when sleep and deep sleep mode

9	ADCSPDPEN	ADC clock enable when sleep and deep sleep mode This bit is set and reset by software. 0: Disabled ADC clock when sleep and deep sleep mode 1: Enabled ADC clock when sleep and deep sleep mode
8	WWDGTSPDPEN	WWDGT clock enable when sleep and deep sleep mode This bit is set and reset by software. 0: Disabled WWDGT clock when sleep and deep sleep mode 1: Enabled WWDGT clock when sleep and deep sleep mode
7:2	Reserved	Must be kept at reset value.
1	CMPSPDPEN	CMP clock enable when sleep and deep sleep mode This bit is set and reset by software. 0: Disabled CMP clock when sleep and deep sleep mode 1: Enabled CMP clock when sleep and deep sleep mode
0	SYSCFGSPDPEN	System configuration clock enable when sleep and deep sleep mode This bit is set and reset by software. 0: Disabled system configuration clock when sleep and deep sleep mode 1: Enabled system configuration clock when sleep and deep sleep mode

4.3.13. Control register 1 (RCU_CTL1)

Address offset: 0x70

Reset value: 0x0000 0008, reset by backup register reset.

This register can be accessed by byte(8-bit), half-word(16-bit) and word(32-bit).

Note: The LXTALEN, LXTALBPS, RTCSRC and RTCEN bits of the control register1 (RCU_CTL1) are only reset after a backup register reset. These bits can be modified only when the BKPWEN bit in the power control register (PMU_CTL0) has to be set.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved						LSCKOUTSEL	LSCKOUTEN	Reserved							BKPRST
						rw	rw								rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RTCEN	Reserved					RTCSRC[1:0]		LXTALSTBRST	LCKMD	LCKMEN	Reserved	LXTALDRI	LXTALBPS	LXTALSTB	LXTALEN
rw						rw		rw	rw	rw		rw	rw	r	rw

Bits	Fields	Descriptions
31:26	Reserved	Must be kept at reset value.
25	LSCKOUTSEL	Low speed clock output selection 0: IRC32K clock selected 1: LXTAL clock selected

24	LSCKOUTEN	Low speed clock output enable Set and cleared by software 0: Low speed clock output(LSCKOUT) disable 1: Low speed clock output(LSCKOUT) enable
23:17	Reserved	Must be kept at reset value.
16	BKPRST	Backup register reset This bit is set and reset by software. 0: No reset 1: Resets backup register
15	RTCEN	RTC clock enable This bit is set and reset by software. 0: Disabled RTC clock 1: Enabled RTC clock
14:10	Reserved	Must be kept at reset value.
9:8	RTCSRC[1:0]	RTC clock entry selection Set and reset by software to control the RTC clock source. Before switching the RTC source clock, the backup register needs to be reset. 00: No clock selected 01: CK_LXTAL selected as RTC source clock 10: CK_IRC32K selected as RTC source clock 11: (CK_HXTAL / 32) selected as RTC source clock
7	LXTALSTBRST	LXTAL stabilization reset. 0: LXTAL stabilization not reset 1: LXTAL stabilization reset
6	LCKMD	LXTAL clock failure detection Set by hardware to indicate when a failure has been detected by the clock security system on the external 32 KHz oscillator (LXTAL). It can be clean by disable LCKMEN or BKPRST. 0: No failure detected on LXTAL (32 KHz oscillator) 1: Failure detected on LXTAL (32 KHz oscillator)
5	LCKMEN	LXTAL clock monitor enable 0: Disable the LXTAL clock monitor 1: Enable the LXTAL clock monitor Set by software to enable the clock security system on LXTAL (32 KHz oscillator). LCKMEN should be enabled only on the LXTAL is enabled (LXTALEN bit enabled) and ready (LXTALSTB flag set by hardware).
4	Reserved	Must be kept at reset value.
3	LXTALDRI	LXTAL drive capability Set and reset by software. Backup register reset reset this value.

		0: Lower driving capability 1: Higher driving capability (reset value) **Note**: The LXTALDRI is not in bypass mode.
2	LXTALBPS	LXTAL bypass mode enable Set and reset by software. 0: Disable the LXTAL Bypass mode 1: Enable the LXTAL Bypass mode
1	LXTALSTB	External low-speed oscillator stabilization Set by hardware to indicate if the LXTAL output clock is stable and ready for use. 0: LXTAL is not stable 1: LXTAL is stable
0	LXTALEN	LXTAL enable Set and reset by software. 0: Disable LXTAL 1: Enable LXTAL

4.3.14. Reset source /clock register (RCU_RSTSCK)

Address offset: 0x74

Reset value: 0xXX00 0000, reset flags reset by power reset only, other reset by system reset.

This register can be accessed by byte(8-bit), half-word(16-bit) and word(32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LPRSTF	WWDGTRSTF	FWDGTRSTF	SWRSTF	PORRSTF	EPRSTF	Reserved	RSTFC	OBLRSTF	Reserved						
r	r	r	r	r	r	r	rw	r							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved														IRC32KSTB	IRC32KEN
														r	rw

Bits	Fields	Descriptions
31	LPRSTF	Low-power reset flag Set by hardware when Deep-sleep /standby reset generated. Reset by writing 1 to the RSTFC bit. 0: No Low-power management reset generated 1: Low-power management reset generated
30	WWDGTRSTF	Window watchdog timer reset flag Set by hardware when a window watchdog timer reset generated. Reset by writing 1 to the RSTFC bit. 0: No window watchdog reset generated

		1: Window watchdog reset generated
29	FWDGTRSTF	Free Watchdog timer reset flag Set by hardware when a Free Watchdog timer generated. Reset by writing 1 to the RSTFC bit. 0: No Free Watchdog timer reset generated 1: Free Watchdog timer reset generated
28	SWRSTF	Software reset flag Set by hardware when a software reset generated. Reset by writing 1 to the RSTFC bit. 0: No software reset generated 1: Software reset generated
27	PORRSTF	Power reset flag Set by hardware when a Power reset generated. Reset by writing 1 to the RSTFC bit. 0: No power reset generated 1: Power reset generated
26	EPRSTF	External PIN reset flag Set by hardware when an External PIN generated. Reset by writing 1 to the RSTFC bit. 0: No external PIN reset generated 1: External PIN reset generated
25	Reserved	Must be kept at reset value.
24	RSTFC	Reset flag clear This bit is set by software to clear all reset flags. 0: Not clear reset flags 1: Clear reset flags
23	OBLRSTF	Option byte loader reset flag Set by hardware when an option byte loader generated. Reset by writing 1 to the RSTFC bit. 0: No option byte loader reset generated 1: Option byte loader reset generated
22:2	Reserved	Must be kept at reset value.
1	IRC32KSTB	IRC32K stabilization Set by hardware to indicate if the IRC32K output clock is stable and ready for use. 0: IRC32K is not stable 1: IRC32K is stable
0	IRC32KEN	IRC32K enable Set and reset by software. 0: Disable IRC32K

1: Enable IRC32K

4.3.15. Configuration register 1 (RCU_CFG1)

Address offset: 0x8C

Reset value: 0x0000 0000

This register can be accessed by byte(8-bit), half-word(16-bit) and word(32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
I2SSEL[1:0]		Reserved	ADCPSC[3:0]				ADCSEL	Reserved		I2C1SEL[1:0]		I2C0SEL[1:0]		USART0SEL[1:0]	
rw			rw				rw			rw		rw		rw	

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:14	I2SSEL[1:0]	I2S clock source selection These bits are controlled by software to select I2S clock source as follows: 00: CK_SYS 01: Reserved 10: CK_IRC48MDIV_PER 11: I2S_CKIN
13	Reserved	Must be kept at reset value.
12:9	ADCPSC[3:0]	ADC clock prescaler selection These bits are written by software to define the ADC clock prescaler. Set and cleared by software. 0000: Output ADC clock not divided 0001: Output ADC clock divided by 2 0010: Output ADC clock divided by 4 0011: Output ADC clock divided by 6 0100: Output ADC clock divided by 8 0101: Output ADC clock divided by 10 0110: Output ADC clock divided by 12 0111: Output ADC clock divided by 16 1000: Output ADC clock divided by 32 1001: Output ADC clock divided by 64 1010: Output ADC clock divided by 128 1011: Output ADC clock divided by 256 Other: Reserved
8	ADCSEL	CK_ADC clock source selection This bit is set and reset by software.

		0: CK_ADC select CK_SYS 1: CK_ADC select CK_IRC48MDIV_PER
7:6	Reserved	Must be kept at reset value.
5:4	I2C1SEL[1:0]	CK_I2C1 clock source selection 00: CK_I2C1 select CK_APB 01: CK_I2C1 select CK_SYS 10 / 11: CK_I2C1 select CK_IRC48MDIV_PER
3:2	I2C0SEL[1:0]	CK_I2C0 clock source selection 00: CK_I2C0 select CK_APB 01: CK_I2C0 select CK_SYS 10 / 11: CK_I2C0 select CK_IRC48MDIV_PER
1:0	USART0SEL[1:0]	CK_USART0 clock source selection This bit is set and reset by software. 00: CK_USART0 select CK_APB 01: CK_USART0 select CK_SYS 10: CK_USART0 select CK_IRC48MDIV_PER 11: CK_USART0 select CK_LXTAL

5. Interrupt / event controller (EXTI)

5.1. Overview

Cortex®-M23 integrates the Nested Vectored Interrupt Controller (NVIC) for efficient exception and interrupts processing. NVIC facilitates low-latency exception and interrupt handling and controls power management. It's tightly coupled to the processer core. You can read the Technical Reference Manual of Cortex®-M23 for more details about NVIC.

EXTI (interrupt / event controller) contains up to 24 independent edge detectors and generates interrupt requests or events to the processer. The EXTI has three trigger types: rising edge, falling edge and both edges. Each edge detector in the EXTI can be configured and masked independently.

5.2. Characteristics

- Cortex®-M23 system exception.
- Up to 39 maskable peripheral interrupts.
- 2 bits interrupt priority configuration - 4 priority levels.
- Efficient interrupt processing.
- Support exception pre-emption and tail-chaining.
- Wake up system from power saving mode.
- Up to 24 independent edge detectors in EXTI.
- Three trigger types: rising, falling and both edges.
- Software interrupt or event trigger.
- Trigger sources configurable.

5.3. Function overview

The ARM Cortex®-M23 processor and the Nested Vectored Interrupt Controller (NVIC) prioritize and handle all exceptions in Handler Mode. The processor state is automatically stored to the stack on an exception and automatically restored from the stack at the end of the Interrupt Service Routine (ISR).

The vector is fetched in parallel to the state saving, enabling efficient interrupt entry. The processor supports tail-chaining, which enables back-to-back interrupts to be performed without the overhead of state saving and restoration. ***Table 5-1. NVIC exception types in Cortex®-M23*** and ***Table 5-2. Interrupt vector table*** list all exception types.

Table 5-1. NVIC exception types in Cortex®-M23

Exception type	Vector number	Priority (a)	Vector address	Description
-	0	-	0x0000_0000	Reserved
Reset	1	-3	0x0000_0004	Reset
NMI	2	-2	0x0000_0008	Non maskable interrupt
HardFault	3	-1	0x0000_000C	All class of fault
-	4-10	-	0x0000_0010 - 0x0000_002B	Reserved
SVCall	11	Programmable	0x0000_002C	System service call via SWI instruction
-	12-13	-	0x0000_0030 – 0x0000_0034	Reserved
PendSV	14	Programmable	0x0000_0038	Pendable request for system service
SysTick	15	Programmable	0x0000_003C	System tick timer

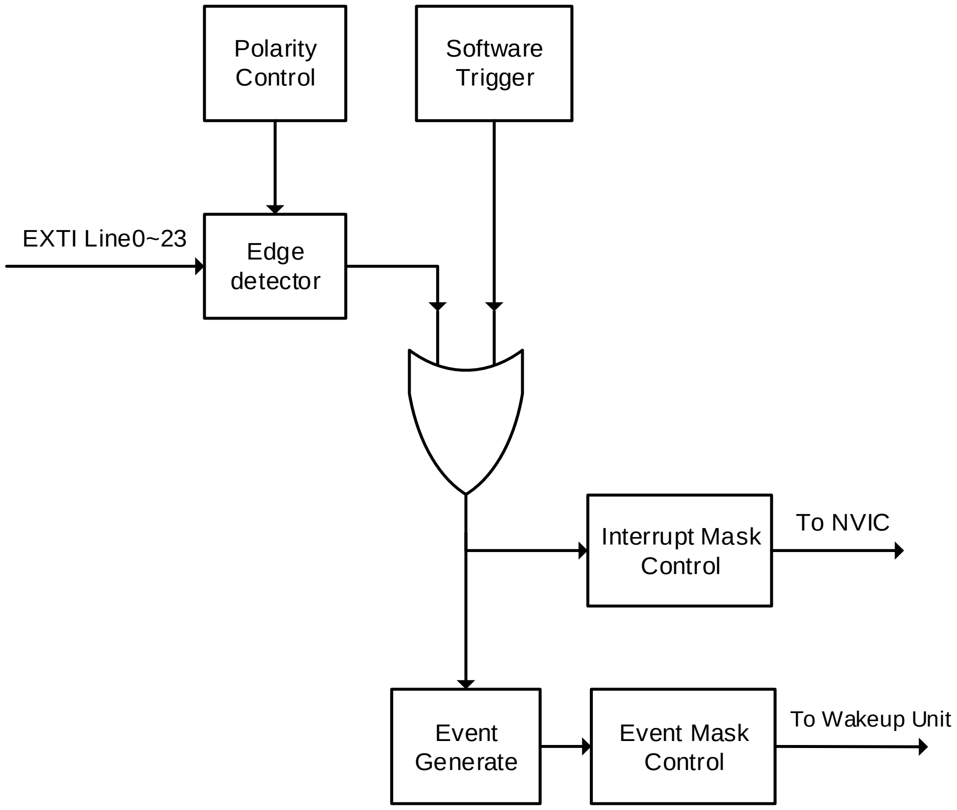
Table 5-2. Interrupt vector table

Interrupt number	Vector number	Peripheral interrupt description	Vector address
IRQ 0	16	WWDGT interrupt	0x0000_0040
IRQ 1	17	RTC Timestamp from EXTI interrupts	0x0000_0044
IRQ 2	18	Reserved	0x0000_0048
IRQ 3	19	FMC global interrupt	0x0000_004C
IRQ 4	20	RCU global interrupt	0x0000_0050
IRQ 5	21	EXTI Line0 interrupt	0x0000_0054
IRQ 6	22	EXTI Line1 interrupt	0x0000_0058
IRQ 7	23	EXTI Line2 interrupt	0x0000_005C
IRQ 8	24	EXTI Line3 interrupt	0x0000_0060
IRQ 9	25	EXTI Line4 interrupt	0x0000_0064
IRQ 10	26	DMA Channel0 global interrupt	0x0000_0068
IRQ 11	27	DMA Channel1 global interrupt	0x0000_006C
IRQ 12	28	DMA Channel2 global interrupt	0x0000_0070
IRQ 13	29	ADC interrupt	0x0000_0074
IRQ 14	30	USART0 global interrupt	0x0000_0078
IRQ 15	31	USART1 global interrupt	0x0000_007C
IRQ 16	32	USART2 global interrupt	0x0000_0080
IRQ 17	33	I2C0 event interrupt	0x0000_0084
IRQ 18	34	I2C0 error interrupt	0x0000_0088
IRQ 19	35	I2C1 event interrupt	0x0000_008C
IRQ 20	36	I2C1 error interrupt	0x0000_0090
IRQ 21	37	SPI0 global interrupt	0x0000_0094
IRQ 22	38	SPI1 global interrupt	0x0000_0098

Interrupt number	Vector number	Peripheral interrupt description	Vector address
IRQ 23	39	RTC alarm from EXTI interrupt	0x0000_009C
IRQ 24	40	EXTI line[9:5] interrupts	0x0000_00A0
IRQ 25	41	TIMER0 trigger and channel commutation interrupts or TIMER0 update interrupt or TIMER0 break interrupt	0x0000_00A4
IRQ 26	42	TIMER0 capture compare interrupt	0x0000_00A8
IRQ 27	43	TIMER2 global interrupt	0x0000_00AC
IRQ 28	44	TIMER13 global interrupt	0x0000_00B0
IRQ 29	45	TIMER15 global interrupt	0x0000_00B4
IRQ 30	46	TIMER16 global interrupt	0x0000_00B8
IRQ 31	47	EXTI line[15:10] interrupts	0x0000_00BC
IRQ 32	48	Reserved	0x0000_00C0
IRQ 33	49	DMA MUX interrupt	0x0000_00C4
IRQ 34	50	CMP0 output from EXTI interrupt	0x0000_00C8
IRQ 35	51	CMP1 output from EXTI interrupt	0x0000_00CC
IRQ 36	52	I2C0 wakeup from EXTI interrupt	0x0000_00D0
IRQ 37	53	I2C1 wakeup from EXTI interrupt	0x0000_00D4
IRQ 38	54	USART0 wakeup from EXTI interrupt	0x0000_00D8

5.4. External interrupt and event block diagram

Figure 5-1. Block diagram of EXTI



5.5. External interrupt and Event function overview

The EXTI contains up to 24 independent edge detectors and generates interrupts request or event to the processer. The EXTI has three trigger types: rising edge, falling edge and both edges. Each edge detector in the EXTI can be configured and masked independently.

The EXTI trigger source includes 16 external lines from GPIO pins and 8 lines from internal modules which refer to ***Table 5-3. EXTI source*** for detail. All GPIO pins can be selected as an EXTI trigger source by configuring SYSCFG_EXTISSx registers in SYSCFG module (please refer to ***System configuration registers*** section for detail).

EXTI can provide not only interrupts but also event signals to the processor. The Cortex®-M23 processor fully implements the Wait For Interrupt (WFI), Wait For Event (WFE) and the Send Event (SEV) instructions. The Wake-up Interrupt Controller (WIC) enables the processor and NVIC to be put into a very low-power sleep mode leaving the WIC to identify and prioritize interrupts and event. EXTI can be used to wake up processor and the whole system when some expected event occurs, such as a special GPIO pin toggling or RTC alarm.

Hardware trigger

Hardware trigger may be used to detect the voltage change of external or internal signals. The software should follow these steps to use this function:

1. Configure EXTI sources in SYSCFG module based on application requirement.
2. Configure EXTI_RTEN and EXTI_FTEN to enable the rising or falling detection on related pins. (Software may set both RTENx and FTENx for a pin at the same time to detect both rising and falling changes on this pin).
3. Enable interrupts or events by setting related EXTI_INTEN or EXTI_EVEN bits.
4. EXTI starts to detect changes on the configured pins. The related PDx bits in EXTI_PD will be set when desired change is detected on these pins and thus, trigger interrupt or event for software. The software should response to the interrupts or events and clear these PDx bits.

Software trigger

Software may also trigger EXTI interrupts or events following these steps:

1. Enable interrupts or events by setting related EXTI_INTEN or EXTI_EVEN bits.
2. Set SWIEVx bits in EXTI_SWIEV register. The related PD bits will be set immediately and thus, trigger interrupts or events. Software should response to these interrupts, and clear related PDx bits.

Table 5-3. EXTI source

EXTI line number	Source
0	PA0 / PB0 / PD0 / PF0
1	PA1 / PB1 / PD1 / PF1
2	PA2 / PB2 / PD2 / PF2
3	PA3 / PB3 / PD3 / PF3
4	PA4 / PB4
5	PA5 / PB5
6	PA6 / PB6 / PC6
7	PA7 / PB7 / PC7
8	PA8 / PB8
9	PA9 / PB9
10	PA10 / PB10
11	PA11 / PB11
12	PA12 / PB12
13	PA13 / PB13 / PC13
14	PA14 / PB14 / PC14
15	PA15 / PB15 / PC15
16	RTC Alarm
17	RTC Timestamp
18	CMP0 output

EXTI line number	Source
19	CMP1 output
20	I2C0 wakeup
21	I2C1 wakeup
22	USART0 wakeup
23	LXTALCS

5.6. Register definition

EXTI base address: 0x4001 0400

5.6.1. Interrupt enable register (EXTI_INTEN)

Address offset: 0x00

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved								INTEN23	INTEN22	INTEN21	INTEN20	INTEN19	INTEN18	INTEN17	INTEN16
								rw	rw	rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INTEN15	INTEN14	INTEN13	INTEN12	INTEN11	INTEN10	INTEN9	INTEN8	INTEN7	INTEN6	INTEN5	INTEN4	INTEN3	INTEN2	INTEN1	INTEN0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31:24	Reserved	Must be kept at reset value.
23:0	INTENx	Interrupt enable bit x(x = 0..23) 0: Interrupt from Linex is disabled 1: Interrupt from Linex is enabled

5.6.2. Event enable register (EXTI_EVEN)

Address offset: 0x04

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved								EVEN23	EVEN22	EVEN21	EVEN20	EVEN19	EVEN18	EVEN17	EVEN16
								rw	rw	rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EVEN15	EVEN14	EVEN13	EVEN12	EVEN11	EVEN10	EVEN9	EVEN8	EVEN7	EVEN6	EVEN5	EVEN4	EVEN3	EVEN2	EVEN1	EVEN0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31:24	Reserved	Must be kept at reset value.
23:0	EVENx	Event enable bit x(x = 0..23) 0: Event from Linex is disabled 1: Event from Linex is enabled

5.6.3. Rising edge trigger enable register (EXTI_RTEN)

Address offset: 0x08

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved								RTEN23	RTEN22	RTEN21	RTEN20	RTEN19	RTEN18	RTEN17	RTEN16
								rw	rw	rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RTEN15	RTEN14	RTEN13	RTEN12	RTEN11	RTEN10	RTEN9	RTEN8	RTEN7	RTEN6	RTEN5	RTEN4	RTEN3	RTEN2	RTEN1	RTEN0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31:24	Reserved	Must be kept at reset value.
23:0	RTENx	Rising edge trigger enable x(x = 0..23) 0: Rising edge of Linex is invalid 1: Rising edge of Linex is valid as an interrupt / event request

5.6.4. Falling edge trigger enable register (EXTI_FTEN)

Address offset: 0x0C

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved								FTEN23	FTEN22	FTEN21	FTEN20	FTEN19	FTEN18	FTEN17	FTEN16
								rw	rw	rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FTEN15	FTEN14	FTEN13	FTEN12	FTEN11	FTEN10	FTEN9	FTEN8	FTEN7	FTEN6	FTEN5	FTEN4	FTEN3	FTEN2	FTEN1	FTEN0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31:24	Reserved	Must be kept at reset value.
23:0	FTENx	Falling edge trigger enable x(x = 0..23) 0: Falling edge of Linex is invalid 1: Falling edge of Linex is valid as an interrupt / event request

5.6.5. Software interrupt event register (EXTI_SWIEV)

Address offset: 0x10

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved								SWIEV23	SWIEV22	SWIEV21	SWIEV20	SWIEV19	SWIEV18	SWIEV17	SWIEV16
								rw	rw	rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SWIEV15	SWIEV14	SWIEV13	SWIEV12	SWIEV11	SWIEV10	SWIEV9	SWIEV8	SWIEV7	SWIEV6	SWIEV5	SWIEV4	SWIEV3	SWIEV2	SWIEV1	SWIEV0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31:24	Reserved	Must be kept at reset value.
23:0	SWIEVx	Interrupt / Event software trigger x(x = 0..23) 0: Deactivate the EXTIx software interrupt / event request 1: Activate the EXTIx software interrupt / event request

5.6.6. Pending register (EXTI_PD)

Address offset: 0x14

Reset value: 0xXXXX XXXX where X is undefined.

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved								PD23	PD22	PD21	PD20	PD19	PD18	PD17	PD16
								rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PD15	PD14	PD13	PD12	PD11	PD10	PD9	PD8	PD7	PD6	PD5	PD4	PD3	PD2	PD1	PD0
rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1

Bits	Fields	Descriptions
31:24	Reserved	Must be kept at reset value.
23:0	PDx	Interrupt pending status x(x = 0..23) 0: EXTI Linex is not triggered 1: EXTI Linex is triggered. This bit is cleared to 0 by writing 1 to it.

6. General-purpose and alternate-function I/Os (GPIO and AFIO)

6.1. Overview

GD32C2x1 devices support up to 45 general purpose I/O pins (GPIO), named PA0 ~ PA15, PB0 ~ PB15, PC6 ~ PC7, PC13 ~ PC15, PD0 ~ PD3, PF0 ~ PF3 for the device to implement logic input/output functions. Each GPIO port has related control and configuration registers to satisfy the requirements of specific applications. The external interrupts on the GPIO pins of the device have related control and configuration registers in the Interrupt/Event Controller Unit (EXTI).

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.

Each of the GPIO pins can be configured by software as output (push-pull or open-drain), input, peripheral alternate function or analog mode. Each GPIO pin can be configured as pull-up, pull-down or no pull-up / pull-down. All GPIOs are high-current capable except for analog mode.

6.2. Characteristics

- Input / output direction control
- Schmitt trigger input function enable control
- Each pin weak pull-up / pull-down function
- Output push-pull/open drain enable control
- Output set / reset control
- External interrupt with programmable trigger edge – using EXTI configuration registers.
- Analog input / output configuration
- Alternate function input / output configuration
- Port configuration lock
- Single cycle toggle output capability

6.3. Function overview

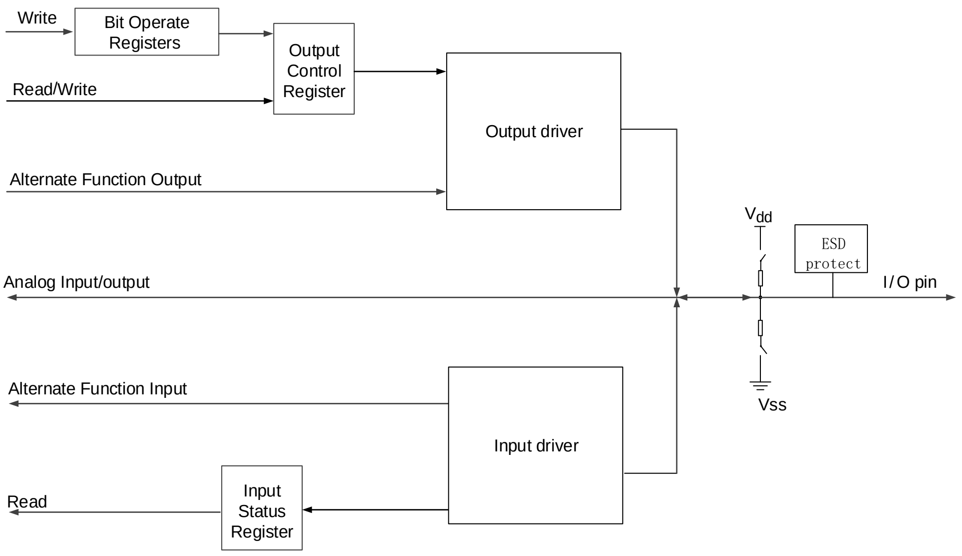
Each of the general-purpose I/O ports can be configured as GPIO inputs, GPIO outputs, AF function or analog mode by GPIO 32-bit configuration registers (GPIOx_CTL). When select AF function, the pad input or output is decided by selected AF function output enable. When the port is output (GPIO output or AFIO output), it can be configured as push-pull or open drain mode by GPIO output mode registers (GPIOx_OMODE). And the port max speed can be configured by GPIO output speed registers (GPIOx_OSPD). Each port can be configured as floating (no pull-up and pull-down), pull-up or pull-down function by GPIO pull-up/pull-down registers (GPIOx_PUD).

Table 6-1. GPIO configuration table

PAD TYPE			CTLy	OMy	PUDy
GPIO INPUT	X	Floating	00	X	00
		pull-up			01
		pull-down			10
GPIO OUTPUT	push-pull	Floating	01	0	00
		pull-up			01
		pull-down			10
	open-drain	Floating		1	00
		pull-up			01
		pull-down			10
AFIO INPUT	X	Floating	10	X	00
		pull-up			01
		pull-down			10
AFIO OUTPUT	push-pull	Floating	10	0	00
		pull-up			01
		pull-down			10
	open-drain	Floating		1	00
		pull-up			01
		pull-down			10
ANALOG	X	X	11	X	XX

Figure 6-1. Basic structure of a general-pupose I/O shows the basic structure of an I/O Port bit.

Figure 6-1. Basic structure of a general-pupose I/O



6.3.1. 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 floating mode that input disabled without Pull-Up(PU) / Pull-Down(PD) resistors. But the Serial-Wired Debug pins are in input PU / PD mode after reset:

PA14: SWCLK in PD mode

PA13: SWDIO in PU mode

The GPIO pins can be configured as inputs or outputs. When the GPIO pins are configured as input pins, all GPIO pins have an internal weak pull-up and weak pull-down which can be chosen. And the data on the external pins can be captured at every AHB clock cycle to the port input status register (GPIOx_ISTAT).

When the GPIO pins are configured as output pins, user can configure the speed of the ports. And chooses the output driver mode: Push-Pull or Open-Drain mode. The value of the port output control register (GPIOx_OCTL) is output on the I/O pin.

There is no need to read-then-write when programming the GPIOx_OCTL at bit level, the user can modify only one or several bits in a single atomic AHB write access by programming ‘1’ to the bit operate register (GPIOx_BOP, or for clearing only GPIOx_BC, or for toggle only GPIOx_TG). The other bits will not be affected.

6.3.2. External interrupt/event lines

All ports have external interrupt capability. To use external interrupt lines, the port must be configured as input mode.

6.3.3. Alternate functions (AF)

When the port is configured as AFIO (set CTLy bits to “0b10”, which is in GPIOx_CTL registers), the port is used as peripheral alternate functions. Each port has sixteen alternate functions can be configured by GPIO alternate functions selected registers (GPIOx_AFSELy (y = 0,1)). The detail alternate function assignments for each port are in the device datasheet.

6.3.4. Additional functions

Some pins have additional functions, which have priority over the configuration in the standard GPIO registers. When for ADC, DAC or additional functions, the port must be configured as analog mode. When for RTC, WKUPx and oscillators additional functions, the port type is set automatically by related RTC, PMU and RCU registers. These ports can be used as normal GPIO when the additional functions disabled.

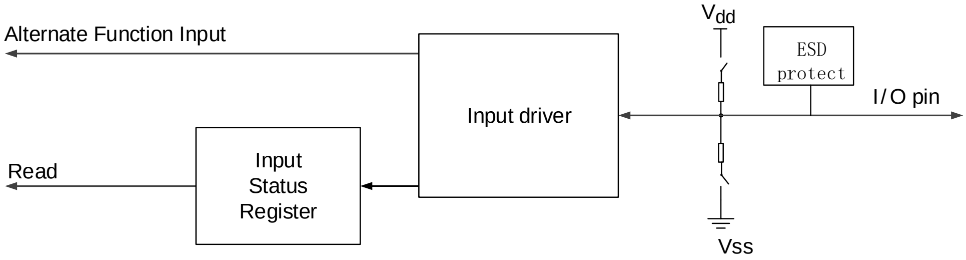
6.3.5. Input configuration

When GPIO pin is configured as Input:

- The schmitt trigger input is enabled.
- The weak pull-up and pull-down resistors could be chosen.
- Every AHB clock cycle the data present on the I/O pin is got to the port input status register.
- The output buffer is disabled.

Figure 6-2. Basic structure of Input configuration shows the input configuration.

Figure 6-2. Basic structure of Input configuration



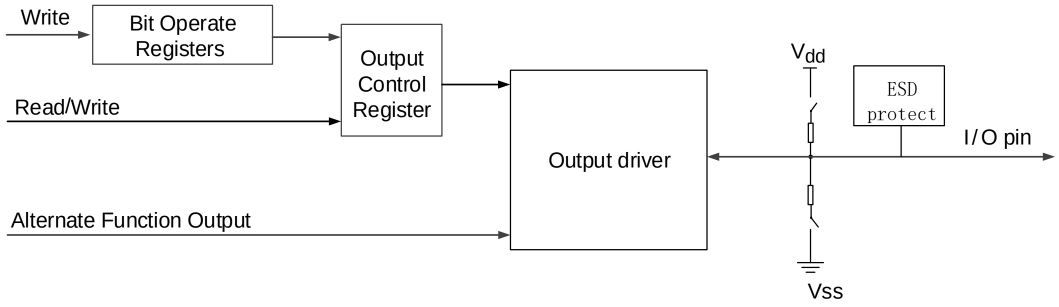
6.3.6. Output configuration

When GPIO pin is configured as output:

- The schmitt trigger input is enabled.
- The weak pull-up and pull-down resistors could be chosen.
- The output buffer is enabled.
- Open Drain Mode: The pad output low level when a “0” in the output control register; while the pad leaves Hi-Z when a “1” in the output control register.
- Push-Pull Mode: The pad output low level when a “0” in the output control register; while the pad output high level when a “1” in the output control register.
- A read access to the port output control register gets the last written value.
- A read access to the port input status register gets the I/O state.

Figure 6-3. Basic structure of Output configuration shows the output configuration.

Figure 6-3. Basic structure of Output configuration



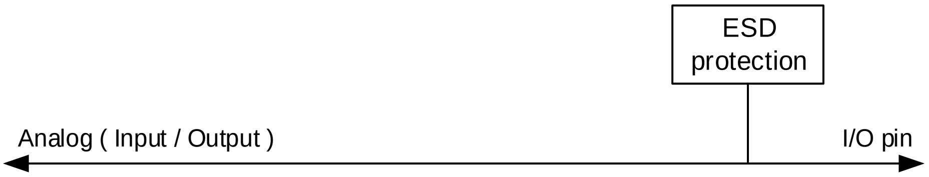
6.3.7. Analog configuration

When GPIO pin is used as analog configuration:

- The weak pull-up and pull-down resistors are disabled.
- The output buffer is disabled.
- The schmitt trigger input is disabled.
- The port input status register of this I/O port bit is “0”.

Figure 6-4. Basic structure of Analog configuration shows the analog configuration.

Figure 6-4. Basic structure of Analog configuration



6.3.8. Alternate function (AF) configuration

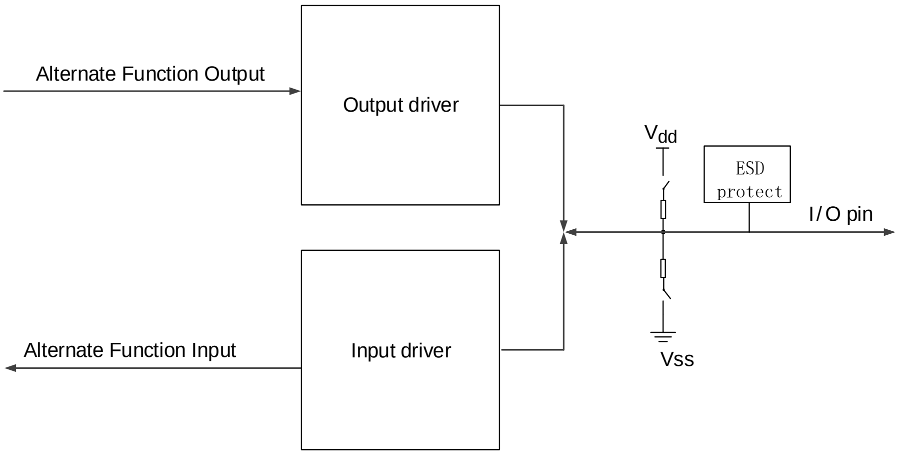
To suit for different device packages, the GPIO supports some alternate functions mapped to some other pins by software.

When be configured as alternate function:

- The output buffer is enabled in open-drain or push-pull configuration.
- The output buffer is driven by the peripheral.
- The schmitt trigger input is enabled.
- The weak pull-up and pull-down resistors could be chosen.
- The I/O pin data is stored into the port input status register every AHB clock.
- A read access to the port input status register gets the I/O state.
- A read access to the port output control register gets the last written value.

Figure 6-5. Basic structure of Alternate function configuration shows the alternate function configuration.

Figure 6-5. Basic structure of Alternate function configuration



6.3.9. GPIO locking function

The locking mechanism allows the IO configuration to be protected.

The protected registers are GPIOx_CTL, GPIOx_OMODE, GPIOx_OSPD, GPIOx_PUD and GPIOx_AFSELy (y=0, 1). It allows the I/O configuration to be frozen by the 32-bit locking register (GPIOx_LOCK). When the special LOCK sequence has occurred on LKK bit in GPIOx_LOCK register and the LKy bit is set in GPIOx_LOCK register, the corresponding port is locked and the corresponding port configuration cannot be modified until the next reset. It recommended to be used in the configuration of driving a power module.

6.3.10. GPIO single cycle toggle function

GPIO could toggle the I/O output level in single AHB cycle by writing 1 to the corresponding bit of GPIOx_TG register. The output signal frequency could up to the half of the AHB clock.

6.3.11. Configure the HXTAL or LXTAL oscillator pins as GPIOs

When HXTALEN or LXTALEN is 0b0, the HXTAL or LXTAL oscillator pins can use as GPIOs. When HXTALEN or LXTALEN is 0b1, the HXTAL or LXTAL oscillator pins are controlled by oscillator, GPIO can not control HXTAL or LXTAL oscillator pins.

When HXTALBPS or LXTALBPS is 0b1, it means the external crystal oscillator (HXTAL or LXTAL) clock is bypassed. The HXTAL or LXTAL oscillator output pin can use as GPIOs, the input pin is used as external clock input.

In the devices which is 48-pin packages, the input and output pins of HXTAL or LXTAL oscillators are independent (refer to HXTAL_REMAP bit of the FMC option bytes). In the devices which pins are less than 48-pin, the input pin OSCX_IN and output pin OSCX_OUT of HXTAL or LXTAL oscillators is common, at the same time only one of them can be used.

6.3.12. Configure the reset pin as GPIO

Set the NRST_MDSEL[1:0] bits of the FMC option bytes can configure PF2 to be used as GPIO. When PF2 is used as GPIO, the reset pin will no longer be able to trigger the device reset.

6.3.13. Configure the boot0 pin (PA14) as GPIO

Set the nBOOT0 bit of the FMC option bytes can configure PA14 to be used as GPIO.

6.4. Register definition

GPIOA base address: 0x4800 0000

GPIOB base address: 0x4800 0400

GPIOC base address: 0x4800 0800

GPIOD base address: 0x4800 0C00

GPIOF base address: 0x4800 1400

6.4.1. Port control register (GPIOx_CTL, x=A..D, F)

Address offset: 0x00

Reset value: 0xE8FF FFFF for port A; 0xFFFF FFFF for others.

This register has to be accessed by word (32-bit)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CTL15[1:0]		CTL14[1:0]		CTL13[1:0]		CTL12[1:0]		CTL11[1:0]		CTL10[1:0]		CTL9[1:0]		CTL8[1:0]	
rw		rw		rw		rw		rw		rw		rw		rw	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CTL7[1:0]		CTL6[1:0]		CTL5[1:0]		CTL4[1:0]		CTL3[1:0]		CTL2[1:0]		CTL1[1:0]		CTL0[1:0]	
rw		rw		rw		rw		rw		rw		rw		rw	

Bits	Fields	Descriptions
31:30	CTL15[1:0]	Pin 15 configuration bits These bits are set and cleared by software. Refer to CTL0[1:0] description
29:28	CTL14[1:0]	Pin 14 configuration bits These bits are set and cleared by software. Refer to CTL0[1:0] description
27:26	CTL13[1:0]	Pin 13 configuration bits These bits are set and cleared by software. Refer to CTL0[1:0] description
25:24	CTL12[1:0]	Pin 12 configuration bits These bits are set and cleared by software. Refer to CTL0[1:0] description
23:22	CTL11[1:0]	Pin 11 configuration bits These bits are set and cleared by software. Refer to CTL0[1:0] description
21:20	CTL10[1:0]	Pin 10 configuration bits These bits are set and cleared by software.

		Refer to CTL0[1:0] description
19:18	CTL9[1:0]	Pin 9 configuration bits These bits are set and cleared by software. Refer to CTL0[1:0] description
17:16	CTL8[1:0]	Pin 8 configuration bits These bits are set and cleared by software. Refer to CTL0[1:0] description
15:14	CTL7[1:0]	Pin 7 configuration bits These bits are set and cleared by software. Refer to CTL0[1:0] description
13:12	CTL6[1:0]	Pin 6 configuration bits These bits are set and cleared by software. Refer to CTL0[1:0] description
11:10	CTL5[1:0]	Pin 5 configuration bits These bits are set and cleared by software. Refer to CTL0[1:0] description
9:8	CTL4[1:0]	Pin 4 configuration bits These bits are set and cleared by software. Refer to CTL0[1:0] description
7:6	CTL3[1:0]	Pin 3 configuration bits These bits are set and cleared by software. Refer to CTL0[1:0] description
5:4	CTL2[1:0]	Pin 2 configuration bits These bits are set and cleared by software. Refer to CTL0[1:0] description
3:2	CTL1[1:0]	Pin 1 configuration bits These bits are set and cleared by software. Refer to CTL0[1:0] description
1:0	CTL0[1:0]	Pin 0 configuration bits These bits are set and cleared by software. 00: Input mode (reset value) 01: GPIO output mode 10: Alternate function mode 11: Analog mode

6.4.2. Port output mode register (GPIOx_OMODE, x=A..D, F)

Address offset: 0x04

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OM15	OM14	OM13	OM12	OM11	OM10	OM9	OM8	OM7	OM6	OM5	OM4	OM3	OM2	OM1	OM0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value
15	OM15	Pin 15 output mode bit These bits are set and cleared by software. Refer to OM0 description
14	OM14	Pin 14 output mode bit These bits are set and cleared by software. Refer to OM0 description
13	OM13	Pin 13 output mode bit These bits are set and cleared by software. Refer to OM0 description
12	OM12	Pin 12 output mode bit These bits are set and cleared by software. Refer to OM0 description
11	OM11	Pin 11 output mode bit These bits are set and cleared by software. Refer to OM0 description
10	OM10	Pin 10 output mode bit These bits are set and cleared by software. Refer to OM0 description
9	OM9	Pin 9 output mode bit These bits are set and cleared by software. Refer to OM0 description
8	OM8	Pin 8 output mode bit These bits are set and cleared by software. Refer to OM0 description
7	OM7	Pin 7 output mode bit These bits are set and cleared by software. Refer to OM0 description
6	OM6	Pin 6 output mode bit These bits are set and cleared by software.

		Refer to OM0 description
5	OM5	Pin 5 output mode bit These bits are set and cleared by software. Refer to OM0 description
4	OM4	Pin 4 output mode bit These bits are set and cleared by software. Refer to OM0 description
3	OM3	Pin 3 output mode bit These bits are set and cleared by software. Refer to OM0 description
2	OM2	Pin 2 output mode bit These bits are set and cleared by software. Refer to OM0 description
1	OM1	Pin 1 output mode bit These bits are set and cleared by software. Refer to OM0 description
0	OM0	Pin 0 output mode bit These bits are set and cleared by software. 0: Output push-pull mode (reset value) 1: Output open-drain mode

6.4.3. Port output speed register (GPIOx_OSPD, x=A..D, F)

Address offset: 0x08

Reset value: 0x0000 2000 for port A; 0x0000 0000 for others.

This register has to be accessed by word (32-bit)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OSPD15	OSPD14	OSPD13	OSPD12]	OSPD11	OSPD10	OSPD9	OSPD8	OSPD7	OSPD6	OSPD5	OSPD4	OSPD3	OSPD2	OSPD1	OSPD0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value
15:0	OSPDy	Pin y output max speed bit This bit bit set and cleared by software. 0: Output speed level 0 (10Mhz) 1: Output speed level 1 (60Mhz)

6.4.4. Port pull-up / down register (GPIOx_PUD, x=A..D, F)

Address offset: 0x0C

Reset value: 0x2400 0000 for port A; 0x0000 0000 for others.

This register has to be accessed by word (32-bit)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PUD15[1:0]		PUD14[1:0]		PUD13[1:0]		PUD12[1:0]		PUD11[1:0]		PUD10[1:0]		PUD9[1:0]		PUD8[1:0]	
rw		rw		rw		rw		rw		rw		rw		rw	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PUD7[1:0]		PUD6[1:0]		PUD5[1:0]		PUD4[1:0]		PUD3[1:0]		PUD2[1:0]		PUD1[1:0]		PUD0[1:0]	
rw		rw		rw		rw		rw		rw		rw		rw	

Bits	Fields	Descriptions
31:30	PUD15[1:0]	Pin 15 pull-up or pull-down bits These bits are set and cleared by software. Refer to PUD0[1:0] description
29:28	PUD14[1:0]	Pin 14 pull-up or pull-down bits These bits are set and cleared by software. Refer to PUD0[1:0] description
27:26	PUD13[1:0]	Pin 13 pull-up or pull-down bits These bits are set and cleared by software. Refer to PUD0[1:0] description
25:24	PUD12[1:0]	Pin 12 pull-up or pull-down bits These bits are set and cleared by software. Refer to PUD0[1:0] description
23:22	PUD11[1:0]	Pin 11 pull-up or pull-down bits These bits are set and cleared by software. Refer to PUD0[1:0] description
21:20	PUD10[1:0]	Pin 10 pull-up or pull-down bits These bits are set and cleared by software. Refer to PUD0[1:0] description
19:18	PUD9[1:0]	Pin 9 pull-up or pull-down bits These bits are set and cleared by software. Refer to PUD0[1:0] description
17:16	PUD8[1:0]	Pin 8 pull-up or pull-down bits These bits are set and cleared by software. Refer to PUD0[1:0] description
15:14	PUD7[1:0]	Pin 7 pull-up or pull-down bits These bits are set and cleared by software.

		Refer to PUD0[1:0] description
13:12	PUD6[1:0]	Pin 6 pull-up or pull-down bits These bits are set and cleared by software. Refer to PUD0[1:0] description
11:10	PUD5[1:0]	Pin 5 pull-up or pull-down bits These bits are set and cleared by software. Refer to PUD0[1:0] description
9:8	PUD4[1:0]	Pin 4 pull-up or pull-down bits These bits are set and cleared by software. Refer to PUD0[1:0] description
7:6	PUD3[1:0]	Pin 3 pull-up or pull-down bits These bits are set and cleared by software. Refer to PUD0[1:0] description
5:4	PUD2[1:0]	Pin 2 pull-up or pull-down bits These bits are set and cleared by software. Refer to PUD0[1:0] description
3:2	PUD1[1:0]	Pin 1 pull-up or pull-down bits These bits are set and cleared by software. Refer to PUD0[1:0] description
1:0	PUD0[1:0]	Pin 0 pull-up or pull-down bits These bits are set and cleared by software. 00: Floating mode, no pull-up and pull-down (reset value) 01: With pull-up mode 10: With pull-down mode 11: Reserved

6.4.5. Port input status register (GPIOx_ISTAT, x=A..D, F)

Address offset: 0x10

Reset value: 0x0000 XXXX

This register has to be accessed by word (32-bit)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ISTAT15	ISTAT14	ISTAT13	ISTAT12	ISTAT11	ISTAT10	ISTAT9	ISTAT8	ISTAT7	ISTAT6	ISTAT5	ISTAT4	ISTAT3	ISTAT2	ISTAT1	ISTAT0
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

Bits	Fields	Descriptions

31:16	Reserved	Must be kept at reset value
15:0	ISTATy	Port input status (y=0..15) These bits are set and cleared by hardware. 0: Input signal low 1: Input signal high

6.4.6. Port output control register (GPIOx_OCTL, x=A..D, F)

Address offset: 0x14

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OCTL15	OCTL14	OCTL13	OCTL12	OCTL11	OCTL10	OCTL9	OCTL8	OCTL7	OCTL6	OCTL5	OCTL4	OCTL3	OCTL2	OCTL1	OCTL0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value
15:0	OCTLy	Port output control (y=0..15) These bits are set and cleared by software. 0: Pin output low 1: Pin output high

6.4.7. Port bit operate register (GPIOx_BOP, x=A..D, F)

Address offset: 0x18

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CR15	CR14	CR13	CR12	CR11	CR10	CR9	CR8	CR7	CR6	CR5	CR4	CR3	CR2	CR1	CR0
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BOP15	BOP14	BOP13	BOP12	BOP11	BOP10	BOP9	BOP8	BOP7	BOP6	BOP5	BOP4	BOP3	BOP2	BOP1	BOP0
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

Bits	Fields	Descriptions
31:16	CRy	Port clear bit y (y=0..15) These bits are set and cleared by software. 0: No action on the corresponding OCTLy bit

		1: Clear the corresponding OCTLy bit
15:0	BOPy	Port set bit y(y=0..15) These bits are set and cleared by software. 0: No action on the corresponding OCTLy bit 1: Set the corresponding OCTLy bit

6.4.8. Port configuration lock register (GPIOx_LOCK, x=A..D, F)

Address offset: 0x1C

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															LKK
															rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LK15	LK14	LK13	LK12	LK11	LK10	LK9	LK8	LK7	LK6	LK5	LK4	LK3	LK2	LK1	LK0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31:17	Reserved	Must be kept at reset value
16	LKK	Lock key It can only be set by using the lock key writing sequence. And is always readable. 0: GPIOx_LOCK register and the port configuration are not locked 1: GPIOx_LOCK register is locked until an MCU reset LOCK key writing sequence: Write 1→Write 0→Write 1→ Read 0→ Read 1 **Note:** The value of LKy(y=0..15) must be held during the LOCK Key writing sequence.
15:0	LKy	Port lock bit y (y=0..15) These bits are set and cleared by software. 0: Port configuration not locked 1: Port configuration locked

6.4.9. Alternate function selected register 0 (GPIOx_AFSEL0, x=A..D, F)

Address offset: 0x20

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SEL7[3:0]				SEL6[3:0]				SEL5[3:0]				SEL4[3:0]			
rw				rw				rw				rw			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SEL3[3:0]				SEL2[3:0]				SEL1[3:0]				SEL0[3:0]			
rw				rw				rw				rw			

Bits	Fields	Descriptions
31:28	SEL7[3:0]	Pin 7 alternate function selected These bits are set and cleared by software. Refer to SEL0[3:0] description
27:24	SEL6[3:0]	Pin 6 alternate function selected These bits are set and cleared by software. Refer to SEL0[3:0] description
23:20	SEL5[3:0]	Pin 5 alternate function selected These bits are set and cleared by software. Refer to SEL0[3:0] description
19:16	SEL4[3:0]	Pin 4 alternate function selected These bits are set and cleared by software. Refer to SEL0[3:0] description
15:12	SEL3[3:0]	Pin 3 alternate function selected These bits are set and cleared by software. Refer to SEL0[3:0] description
11:8	SEL2[3:0]	Pin 2 alternate function selected These bits are set and cleared by software. Refer to SEL0[3:0] description
7:4	SEL1[3:0]	Pin 1 alternate function selected These bits are set and cleared by software. Refer to SEL0[3:0] description
3:0	SEL0[3:0]	Pin 0 alternate function selected These bits are set and cleared by software. 0000: AF0 selected (reset value) 0001: AF1 selected 0010: AF2 selected 0011: AF3 selected 0100: AF4 selected 0101: AF5 selected 0110: AF6 selected 0111: AF7 selected 1000: AF8 selected 1001: AF9 selected ... 1111: AF15 selected

6.4.10. Alternate function selected register 1 (GPIOx_AFSEL1, x=A..D, F)

Address offset: 0x24

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SEL15[3:0]				SEL14[3:0]				SEL13[3:0]				SEL12[3:0]			
rw				rw				rw				rw			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SEL11[3:0]				SEL10[3:0]				SEL9[3:0]				SEL8[3:0]			
rw				rw				rw				rw			

Bits	Fields	Descriptions
31:28	SEL15[3:0]	Pin 15 alternate function selected These bits are set and cleared by software. Refer to SEL8[3:0] description
27:24	SEL14[3:0]	Pin 14 alternate function selected These bits are set and cleared by software. Refer to SEL8[3:0] description
23:20	SEL13[3:0]	Pin 13 alternate function selected These bits are set and cleared by software. Refer to SEL8[3:0] description
19:16	SEL12[3:0]	Pin 12 alternate function selected These bits are set and cleared by software. Refer to SEL8[3:0] description
15:12	SEL11[3:0]	Pin 11 alternate function selected These bits are set and cleared by software. Refer to SEL8[3:0] description
11:8	SEL10[3:0]	Pin 10 alternate function selected These bits are set and cleared by software. Refer to SEL8[3:0] description
7:4	SEL9[3:0]	Pin 9 alternate function selected These bits are set and cleared by software. Refer to SEL8[3:0] description
3:0	SEL8[3:0]	Pin 8 alternate function selected These bits are set and cleared by software. 0000: AF0 selected (reset value) 0001: AF1 selected 0010: AF2 selected 0011: AF3 selected

0100: AF4 selected

0101: AF5 selected

0110: AF6 selected

0111: AF7 selected

1000: AF8 selected

1001: AF9 selected

...

1111: AF15 selected

6.4.11. Bit clear register (GPIOx_BC, x=A..D, F)

Address offset: 0x28

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CR15	CR14	CR13	CR12	CR11	CR10	CR9	CR8	CR7	CR6	CR5	CR4	CR3	CR2	CR1	CR0
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value
15:0	CRy	Port clear bit y (y=0..15) These bits are set and cleared by software. 0: No action on the corresponding OCTLy bit 1: Clear the corresponding OCTLy bit

6.4.12. Port bit toggle register (GPIOx_TG, x=A..D, F)

Address offset: 0x2C

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TG15	TG14	TG13	TG12	TG11	TG10	TG9	TG8	TG7	TG6	TG5	TG4	TG3	TG2	TG1	TG0
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

Bits	Fields	Descriptions

31:16	Reserved	Must be kept at reset value
15:0	TGy	Port toggle bit y (y=0..15) These bits are set and cleared by software. 0: No action on the corresponding OCTLy bit 1: Toggle the corresponding OCTLy bit

7. Cyclic redundancy checks management unit (CRC)

7.1. Overview

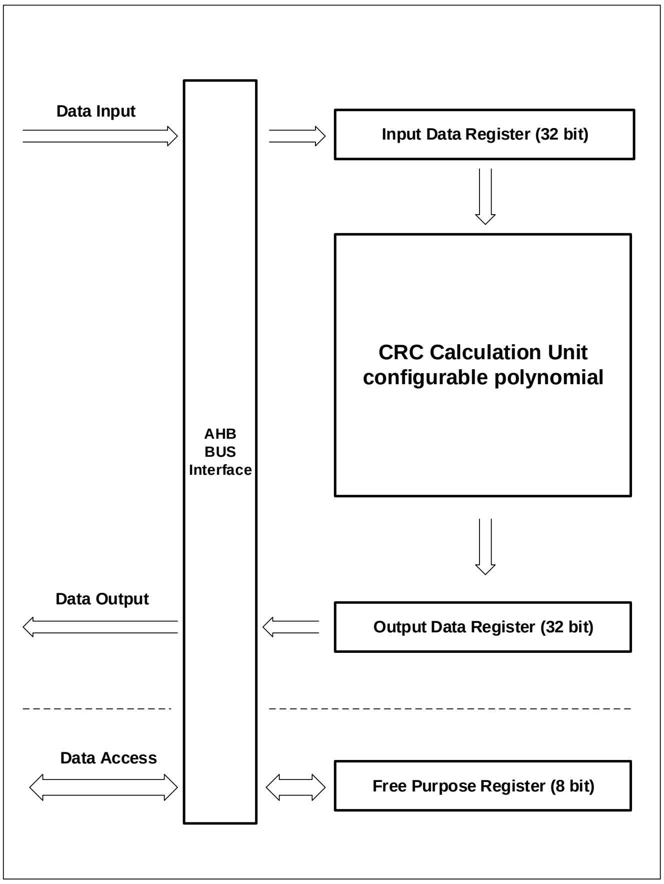
A cyclic redundancy check (CRC) is an error-detecting code commonly used in digital networks and storage devices to detect accidental changes to raw data.

The CRC management unit can be used to calculate 7 / 8 / 16 / 32 bit CRC code within user configurable polynomial.

7.2. Characteristics

- Supports 7 / 8 / 16 / 32 bit data input.
- For 7 (8) / 16 / 32 bit input data length, the calculation cycles are 1 / 2 / 4 AHB clock cycles.
- User configurable polynomial value and size.
- After CRC module·reset, user can configure initial value.
- Free 8-bit register is unrelated to calculation and can be used for any other goals by any other peripheral devices.

Figure 6-6. Block diagram of CRC calculation unit



7.3. Function overview

- CRC calculation unit is used to calculate the 32-bit raw data, and CRC_DATA register will receive the raw data and store the calculation result.

  If the CRC_DATA register has not been cleared by setting the CRC_CTL register, the new input raw data will be calculated based on the result of previous value of CRC_DATA.

  CRC calculation will spend 4 / 2 / 1 AHB clock cycles for 32 / 16 / 8 (7) bit data size. During this period, AHB will not be hanged because of the existence of the 32bit input buffer.

- This module supplies an 8-bit free register CRC_FDATA.

  CRC_FDATA is unrelated to the CRC calculation. Independent read and write operations can be performed at any time.

- Reversible function can reverse the input data and output data.

  For input data, 3 reverse types can be selected.

  Original data is 0x3456CDEF:

  1) byte reverse:
  32-bit data is divided into 4 groups and reverse implement in group inside. Reversed data: 0x2C6AB3F7

  2) half-word reverse:
  32-bit data is divided into 2 groups and reverse implement in group inside. Reversed data: 0x6A2CF7B3

  3) word reverse:
  32-bit data is divided into 1 groups and reverse implement in group inside. Reversed data: 0xF7B36A2C

  For output data, reverse type is word reverse.

  For example: when REV_O=1, calculation result 0x3344CCDD will be converted to 0xBB3322CC.

- User configurable initial calculation data is available.

  When RST bit is set or write operation to CRC_IDATA register, the CRC_DATA register will be automatically initialized to the value in CRC_IDATA.

- User configurable polynomial.

  Depends on PS[1:0] bits, the valid polynomial and output bit width can be selected by user. If the polynomial is less than 32 bit, the high bits of the input data and output data is unavailable. It is strongly recommend resetting the CRC calculation unit after change the PS[1:0] bits or polynomial.

7.4. Register definition

CRC base address: 0x4002 3000

7.4.1. Data register (CRC_DATA)

Address offset: 0x00

Reset value: 0xFFFF FFFF

This register has to be accessed by word (32-bit).

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="16">DATA[31:16]</td></tr>
<tr><td colspan="16">rw</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="16">DATA[15:0]</td></tr>
<tr><td colspan="16">rw</td></tr>
</table>

Bits	Fields	Descriptions
31:0	DATA[31:0]	CRC calculation result bits Software writes and reads. This register is used to calculate new data, and the register can be written the new data directly. Write value cannot be read because the read value is the previous CRC calculation result.

7.4.2. Free data register (CRC_FDATA)

Address offset: 0x04

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="16">Reserved</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="8">Reserved</td><td colspan="8">FDATA[7:0]</td></tr>
<tr><td colspan="8"></td><td colspan="8">rw</td></tr>
</table>

Bits	Fields	Descriptions
31:8	Reserved	Must be kept at reset value.
7:0	FDATA[7:0]	Free data register bits Software writes and reads. These bits are unrelated with CRC calculation. This byte can be used for any goal by any other peripheral. The CRC_CTL register will generate no effect to the byte.

7.4.3. Control register (CRC_CTL)

Address offset: 0x08

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								REV_O	REV_I[1:0]		PS[1:0]		Reserved		RST
								rw	rw		rw				rs

Bits	Fields	Descriptions
31:8	Reserved	Must be kept at reset value.
7	REV_O	Reverse output data value in bit order 0:Not bit reversed for output data 1:Bit reversed for output data
6:5	REV_I[1:0]	Reverse type for input data 0: Dot not use reverse for input data 1: Reverse input data with every 8-bit length 2: Reverse input data with every 16-bit length 3: Reverse input data with whole 32-bit length
4:3	PS[1:0]	Size of polynomial 0: 32 bit 1: 16 bit (POLY [15:0] is used for calculation.) 2: 8 bit (POLY [7:0] is used for calculation.) 3: 7 bit (POLY [6:0] is used for calculation.)
2:1	Reserved	Must be kept at reset value.
0	RST	Software writes and reads. Set this bit can reset the CRC_DATA register. When set, the value of the CRC_DATA register is automatically initialized to the value in the CRC_IDATA register and then automatically cleared by hardware. This bit will take no effect to CRC_FDATA.

7.4.4. Initialization data register (CRC_IDATA)

Address offset: 0x10

Reset value: 0xFFFF FFFF

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

<table>
<tr><td colspan="16">IDATA[31:16]</td></tr>
<tr><td colspan="16">rw</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="16">IDATA[15:0]</td></tr>
<tr><td colspan="16">rw</td></tr>
</table>

Bits	Fields	Descriptions
31:0	IDATA[31:0]	Configurable initial CRC data value When RST bit in CRC_CTL asserted, CRC_DATA will be programmed to this value.

7.4.5. Polynomial register (CRC_POLY)

Address offset: 0x14

Reset value: 0x04C1 1DB7

This register has to be accessed by word (32-bit).

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="16">POLY[31:16]</td></tr>
<tr><td colspan="16">rw</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="16">POLY[15:0]</td></tr>
<tr><td colspan="16">rw</td></tr>
</table>

Bits	Fields	Descriptions
31:0	POLY[31:0]	User configurable polynomial value This value is used together with PS[1:0] bits.

8. Debug (DBG)

8.1. Overview

The GD32C2x1 series provide a large variety of debug, trace and test features. They are implemented with a standard configuration of the ARM® CoreSight™ module together with a daisy chained standard TAP controller. Debug and trace functions are integrated into the ARM® Cortex®-M23. The debug system supports serial wire debug (SWD) and trace functions. The debug and trace functions refer to the following documents:

- Cortex®-M23 Technical Reference Manual.
- ARM Debug Interface v5 Architecture Specification.

The DBG hold unit helps debugger to debug power saving mode, TIMER, I2C, RTC, WWDGT, and FWDGT. When corresponding bit is set, it provides a clock in power saving mode or holds the state for TIMER, I2C, RTC, WWDGT and FWDGT.

8.2. SW function overview

Debug capabilities can be accessed by a debug tool via Serial Wire (SW - Debug Port).

8.2.1. Pin assignment

The synchronous serial wire debug (SWD) provide 2-pin SW interface, known as SW data input / output (SWDIO) and SW clock (SWCLK).

The pin assignment is as following:

PA14	: SWCLK
PA13	: SWDIO

If SWD not used, all 2 pins can be released to other GPIO functions. Please refer to ***General-purpose and alternate-function I/Os (GPIO and AFIO).***

8.3. Debug hold function overview

8.3.1. Debug support for power saving mode

When the DSLP_HOLD bit in DBG control register 0 (DBG_CTL0) is set, and entering the deep-sleep mode, the clock of AHB bus and system clock are provided by CK_IRC48M, and the debugger can debug in deep-sleep mode.

When the SLP_HOLD bit in DBG control register 0 (DBG_CTL0) is set, and entering the sleep mode, the clock of AHB bus for CPU is not closed, and the debugger can debug in sleep mode.

When the STB_HOLD bit in DBG control register 0 (DBG_CTL0) is set, and entering the standby mode, the clock of AHB bus and system clock are provided by CK_IRC48M, and the debugger can debug in standby mode. When exiting the standby mode, a system reset generated.

8.3.2. Debug support for TIMER, I2C, RTC, WWDGT and FWDGT

When the core is halted and the corresponding bit in DBG control register 0 or DBG control register 1 (DBG_CTL0 or DBG_CTL1) is set, the following events occur.

For TIMER, the timer counters are stopped and held for debugging.

For I2C, SMBUS timeout is held for debugging.

For RTC, the counter is stopped for debugging.

For WWDGT or FWDGT, the counter clock is stopped for debugging.

8.4. Register definition

DBG base address: 0x4001 5800

8.4.1. ID code register (DBG_ID)

Address offset: 0x00

Read only

This register has to be accessed by word(32-bit)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ID_CODE[31:16]															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID_CODE[15:0]															
r															

Bits	Fields	Descriptions
31:0	ID_CODE[31:0]	DBG ID code register These bits can only be read by software. These bits are unchanged constant.

8.4.2. Control register 0 (DBG_CTL0)

Address offset: 0x04

Reset value: 0x0000 0000; power reset only

This register has to be accessed by word(32-bit)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved										TIMER16_HOLD	TIMER15_HOLD	Reserved			I2C1_HOLD
										rw	rw				rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
I2C0_HOLD	Reserved	TIMER13_HOLD	TIMER2_HOLD	TIMER0_HOLD	Reserved	WWDGT_HOLD	FWDGT_HOLD	Reserved					STB_HOLD	DSLP_HOLD	SLP_HOLD
rw		rw	rw	rw		rw	rw						rw	rw	rw

Bits	Fields	Descriptions
31:22	Reserved	Must be kept at reset value.
21	TIMER16_HOLD	TIMER 16 hold bit This bit is set and reset by software. 0: no effect 1: hold the TIMER 16 counter for debugging when the core is halted.
20	TIMER15_HOLD	TIMER 15 hold bit This bit is set and reset by software. 0: no effect 1: hold the TIMER 15 counter for debugging when the core is halted.

19:17	Reserved	Must be kept at reset value
16	I2C1_HOLD	I2C1 hold bit This bit is set and reset by software. 0: no effect 1: hold the I2C1 status to avoid SMBUS timeout for debugging when the core is halted.
15	I2C0_HOLD	I2C0 hold bit This bit is set and reset by software. 0: no effect 1: hold the I2C0 status to avoid SMBUS timeout for debugging when the core is halted.
14	Reserved	Must be kept at reset value.
13	TIMER13_HOLD	TIMER 13 hold bit This bit is set and reset by software. 0: no effect 1: hold the TIMER 13 counter for debugging when the core is halted.
12	TIMER2_HOLD	TIMER 2 hold bit This bit is set and reset by software. 0: no effect 1: hold the TIMER 2 counter for debugging when the core is halted.
11	TIMER0_HOLD	TIMER 0 hold bit This bit is set and reset by software. 0: no effect 1: hold the TIMER 0 counter for debugging when the core is halted.
10	Reserved	Must be kept at reset value.
9	WWDGT_HOLD	WWDGT hold bit This bit is set and reset by software. 0: no effect 1: hold the WWDGT counter clock for debugging when the core is halted.
8	FWDGT_HOLD	FWDGT hold bit This bit is set and reset by software. 0: no effect 1: hold the FWDGT counter clock for debugging when the core is halted.
7:3	Reserved	Must be kept at reset value.
2	STB_HOLD	Standby mode hold bit This bit is set and reset by software. 0: no effect 1: In the standby mode, the clock of AHB bus and system clock are provided by

		CK_IRC48M, a system reset generated when exiting standby mode.
1	DSLP_HOLD	Deep-sleep mode hold bit This bit is set and reset by software. 0: no effect 1: In the deep-sleep mode, the clock of AHB bus and system clock are provided by CK_IRC48M.
0	SLP_HOLD	Sleep mode hold bit This bit is set and reset by software. 0: no effect 1: In the sleep mode, the clock of AHB is on.

8.4.3. Control register 1 (DBG_CTL1)

Address offset: 0x08

Reset value: 0x0000 0000; power reset only

This register has to be accessed by word(32-bit)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved.															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved					RTC_HOLD	Reserved									
					rw										

Bits	Fields	Descriptions
31:11	Reserved	Must be kept at reset value.
10	RTC_HOLD	RTC hold bit This bit is set and reset by software. 0: no effect 1: hold the RTC counter for debugging when the core is halted.
9:0	Reserved	Must be kept at reset value.

9. Direct memory access controller (DMA)

9.1. Overview

The direct memory access (DMA) controller provides a hardware method of transferring data between peripherals and / or memory without intervention from the CPU, thereby freeing up bandwidth for other system functions. Data can be quickly moved by DMA between peripherals and memory as well as memory and memory without any CPU actions. There are 3 channels in the DMA controller. Each channel is dedicated to manage memory access requests from one or more peripherals. An arbiter is implemented inside to handle the priority among DMA requests.

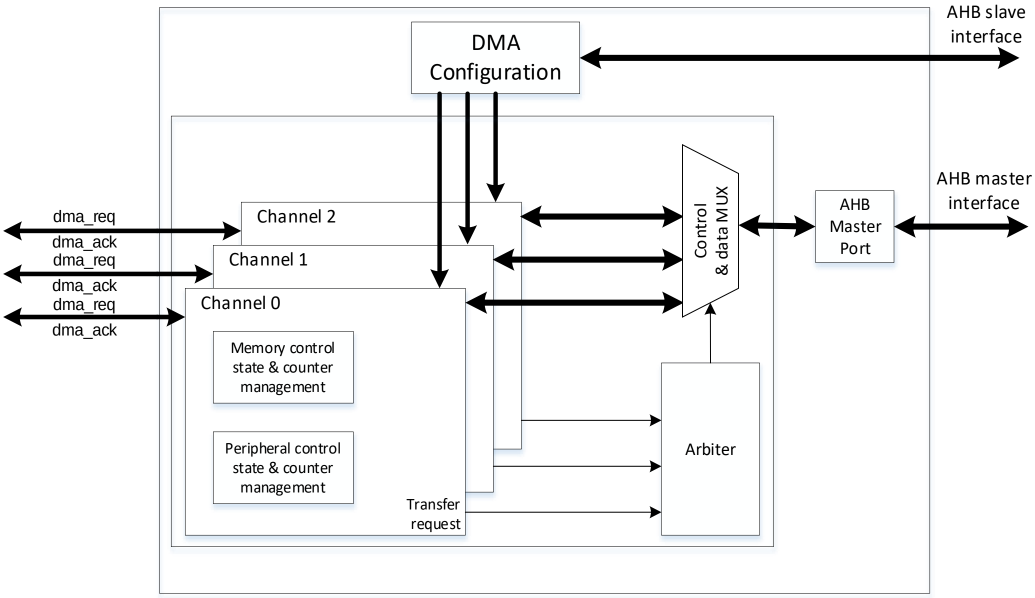
The system bus is shared by the DMA controller and the Cortex®-M23 core. When the DMA and the CPU are targeting the same destination, the DMA access may stop the CPU access to the system bus for some bus cycles. Round-robin scheduling is implemented in the bus matrix to ensure at least half of the system bus bandwidth for the CPU.

9.2. Characteristics

- Programmable length of data to be transferred, max to 65536.
- 3 channels and each channel are configurable.
- AHB and APB peripherals, FLASH, SRAM can be accessed as source and destination.
- Each channel is connected to flexible DMA request.
- Software DMA channel priority (low, medium, high, ultra high) and hardware DMA channel priority (DMA channel 0 has the highest priority and DMA channel 2 has the lowest priority).
- Support independent 8, 16, 32-bit memory and peripheral transfer.
- Support independent fixed and increasing address generation algorithm of memory and peripheral.
- Support circular transfer mode.
- Support peripheral to memory, memory to peripheral, and memory to memory transfers.
- One separate interrupt per channel with three types of event flags.
- Support interrupt enable and clear.

9.3. Block diagram

Figure 9-1. Block diagram of DMA



As shown in ***Figure 9-1. Block diagram of DMA***, a DMA controller consists of four main parts:

- DMA configuration through AHB slave interface.
- Data transmission through two AHB master interfaces for memory access and peripheral access.
- An arbiter inside to manage multiple peripheral requests coming at the same time.
- Channel management to control address / data selection and data counting.

9.4. Function overview

9.4.1. DMA operation

Each DMA transfer consists of two operations, including the loading of data from the source and the storage of the loaded data to the destination. The source and destination addresses are computed by the DMA controller based on the programmed values in the DMA_CHxPADDR, DMA_CHxMADDR, and DMA_CHxCTL registers. The DMA_CHxCNT register controls how many transfers to be transmitted on the channel. The PWIDTH and MWIDTH bits in the DMA_CHxCTL register determine how many bytes to be transmitted in a transfer.

Suppose DMA_CHxCNT is 4, and both PNAGA and MNAGA are set. The DMA transfer operations for each combination of PWIDTH and MWIDTH are shown in the following ***Table 9-1. DMA transfer operation***.

Table 9-1. DMA transfer operation

Transfer size		Transfer operations	
Source	Destination	Source	Destination
32 bits	32 bits	1: Read B3B2B1B0[31:0] @0x0 2: Read B7B6B5B4[31:0] @0x4 3: Read BBBAB9B8[31:0] @0x8 4: Read BFBEBDBC[31:0] @0xC	1: Write B3B2B1B0[31:0] @0x0 2: Write B7B6B5B4[31:0] @0x4 3: Write BBBAB9B8[31:0] @0x8 4: Write BFBEBDBC[31:0] @0xC
32 bits	16 bits	1: Read B3B2B1B0[31:0] @0x0 2: Read B7B6B5B4[31:0] @0x4 3: Read BBBAB9B8[31:0] @0x8 4: Read BFBEBDBC[31:0] @0xC	1: Write B1B0[15:0] @0x0 2: Write B5B4[15:0] @0x2 3: Write B9B8[15:0] @0x4 4: Write BDBC[15:0] @0x6
32 bits	8 bits	1: Read B3B2B1B0[31:0] @0x0 2: Read B7B6B5B4[31:0] @0x4 3: Read BBBAB9B8[31:0] @0x8 4: Read BFBEBDBC[31:0] @0xC	1: Write B0[7:0] @0x0 2: Write B4[7:0] @0x1 3: Write B8[7:0] @0x2 4: Write BC[7:0] @0x3
16 bits	32 bits	1: Read B1B0[15:0] @0x0 2: Read B3B2[15:0] @0x2 3: Read B5B4[15:0] @0x4 4: Read B7B6[15:0] @0x6	1: Write 0000B1B0[31:0] @0x0 2: Write 0000B3B2[31:0] @0x4 3: Write 0000B5B4[31:0] @0x8 4: Write 0000B7B6[31:0] @0xC
16 bits	16 bits	1: Read B1B0[15:0] @0x0 2: Read B3B2[15:0] @0x2 3: Read B5B4[15:0] @0x4 4: Read B7B6[15:0] @0x6	1: Write B1B0[15:0] @0x0 2: Write B3B2[15:0] @0x2 3: Write B5B4[15:0] @0x4 4: Write B7B6[15:0] @0x6
16 bits	8 bits	1: Read B1B0[15:0] @0x0 2: Read B3B2[15:0] @0x2 3: Read B5B4[15:0] @0x4 4: Read B7B6[15:0] @0x6	1: Write B0[7:0] @0x0 2: Write B2[7:0] @0x1 3: Write B4[7:0] @0x2 4: Write B6[7:0] @0x3
8 bits	32 bits	1: Read B0[7:0] @0x0 2: Read B1[7:0] @0x1 3: Read B2[7:0] @0x2 4: Read B3[7:0] @0x3	1: Write 000000B0[31:0] @0x0 2: Write 000000B1[31:0] @0x4 3: Write 000000B2[31:0] @0x8 4: Write 000000B3[31:0] @0xC
8 bits	16 bits	1: Read B0[7:0] @0x0 2: Read B1[7:0] @0x1 3: Read B2[7:0] @0x2 4: Read B3[7:0] @0x3	1, Write 00B0[15:0] @0x0 2, Write 00B1[15:0] @0x2 3, Write 00B2[15:0] @0x4 4, Write 00B3[15:0] @0x6
8 bits	8 bits	1: Read B0[7:0] @0x0 2: Read B1[7:0] @0x1 3: Read B2[7:0] @0x2 4: Read B3[7:0] @0x3	1, Write B0[7:0] @0x0 2, Write B1[7:0] @0x1 3, Write B2[7:0] @0x2 4, Write B3[7:0] @0x3

The CNT bits in the DMA_CHxCNT register control how many data to be transmitted on the channel and must be configured before enable the CHEN bit in the register. During the transmission, the CNT bits indicate the remaining number of data items to be transferred.

The DMA transmission is disabled by clearing the CHEN bit in the DMA_CHxCTL register.

- If the DMA transmission is not completed when the CHEN bit is cleared, two situations may be occurred when restart this DMA channel:
  - If no register configuration operations of the channel occurs before restart the DMA channel, the DMA will continue to complete the rest of the transmission.
  - If any register configuration operations to DMA_CHxCNT, DMA_CHxPADDR or DMA_CHxMADDR of corresponding channel occur, the DMA will restart a new transmission.
- If the DMA transmission has been finished when clearing the CHEN bit, enable the DMA channel without any register configuration operation to DMA_CHxCNT, DMA_CHxPADDR or DMA_CHxMADDR of corresponding channel will not launch any DMA transfer.

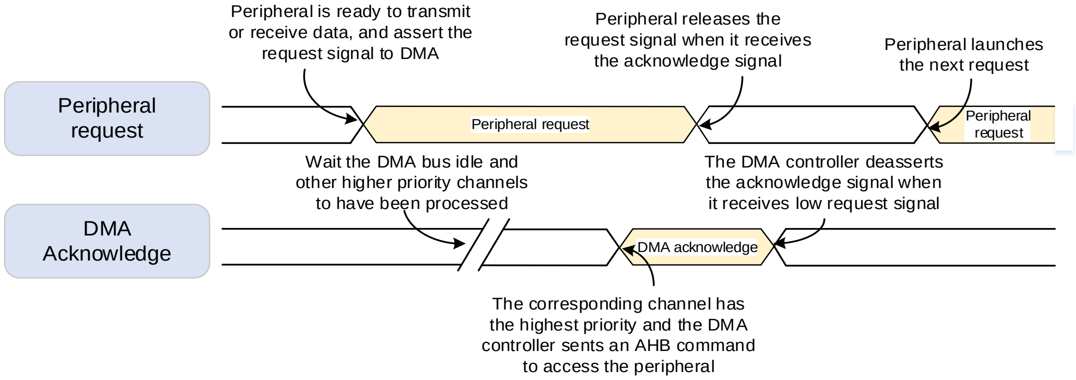
9.4.2. Peripheral handshake

To ensure a well-organized and efficient data transfer, a handshake mechanism is introduced between the DMA and peripherals, including a request signal and a acknowledge signal:

- Request signal asserted by peripheral to DMA controller, indicating that the peripheral is ready to transmit or receive data.
- Acknowledge signal responded by DMA to peripheral, indicating that the DMA controller has initiated an AHB command to access the peripheral.

Figure 9-2. Handshake mechanism shows how the handshake mechanism works between the DMA controller and peripherals.

Figure 9-2. Handshake mechanism


Peripheral is ready to transmit or receive data, and assert the request signal to DMA
Peripheral releases the request signal when it receives the acknowledge signal
Peripheral launches the next request
Peripheral request
Peripheral request
Peripheral request
Wait the DMA bus idle and other higher priority channels to have been processed
The DMA controller deasserts the acknowledge signal when it receives low request signal
DMA Acknowledge
DMA acknowledge
The corresponding channel has the highest priority and the DMA controller sents an AHB command to access the peripheral

9.4.3. Arbitration

When two or more requests are received at the same time, the arbiter determines which request is served based on the priorities of channels. There are two-stage priorities, including the software priority and the hardware priority. The arbiter determines which channel is selected to respond according to the following priority rules:

- Software priority: Four levels, including low, medium, high and ultra-high by configuring the PRIO bits in the DMA_CHxCTL register.

- For channels with equal software priority level, priority is given to the channel with lower channel number.

9.4.4. Address generation

Two kinds of address generation algorithm are implemented independently for memory and peripheral, including the fixed mode and the increased mode. The PNAGA and MNAGA bit in the DMA_CHxCTL register are used to configure the next address generation algorithm of peripheral and memory.

In the fixed mode, the next address is always equal to the base address configured in the base address registers (DMA_CHxPADDR, DMA_CHxMADDR).

In the increasing mode, the next address is equal to the current address plus 1 or 2 or 4, depending on the transfer data width.

9.4.5. Circular mode

Circular mode is implemented to handle continue peripheral requests (for example, ADC scan mode). The circular mode is enabled by setting the CMEN bit in the DMA_CHxCTL register.

In circular mode, the CNT bits are automatically reloaded with the pre-programmed value and the full transfer finish flag is asserted at the end of every DMA transfer. DMA can always responds the peripheral request until the CHEN bit in the DMA_CHxCTL register is cleared.

9.4.6. Memory to memory mode

The memory to memory mode is enabled by setting the M2M bit in the DMA_CHxCTL register. In this mode, the DMA channel can also work without being triggered by a request from a peripheral. The DMA channel starts transferring as soon as it is enabled by setting the CHEN bit in the DMA_CHxCTL register, and completed when the DMA_CHxCNT register reaches zero.

9.4.7. Channel configuration

When starting a new DMA transfer, it is recommended to respect the following steps:

1. Read the CHEN bit and judge whether the channel is enabled or not. If the channel is enabled, clear the CHEN bit by software. When the CHEN bit is read as ‘0’, configuring and starting a new DMA transfer is allowed.
2. Configure the M2M bit and DIR bit in the DMA_CHxCTL register to set the transfer mode.
3. Configure the CMEN bit in the DMA_CHxCTL register to enable / disable the circular mode.
4. Configure the PRIO bits in the DMA_CHxCTL register to set the channel software priority.
5. Configure the memory and peripheral transfer width, memory and peripheral address generation algorithm in the DMA_CHxCTL register.

6. Configure the enable bit for full transfer finish interrupt, half transfer finish interrupt, transfer error interrupt in the DMA_CHxCTL register.
7. Configure the DMA_CHxPADDR register for setting the peripheral base address.
8. Configure the DMA_CHxMADDR register for setting the memory base address.
9. Configure the DMA_CHxCNT register to set the total transfer data number.
10. Configure the CHEN bit with ‘1’ in the DMA_CHxCTL register to enable the channel.

9.4.8. Interrupt

Each DMA channel has a dedicated interrupt. There are three types of interrupt event, including full transfer finish, half transfer finish, and transfer error.

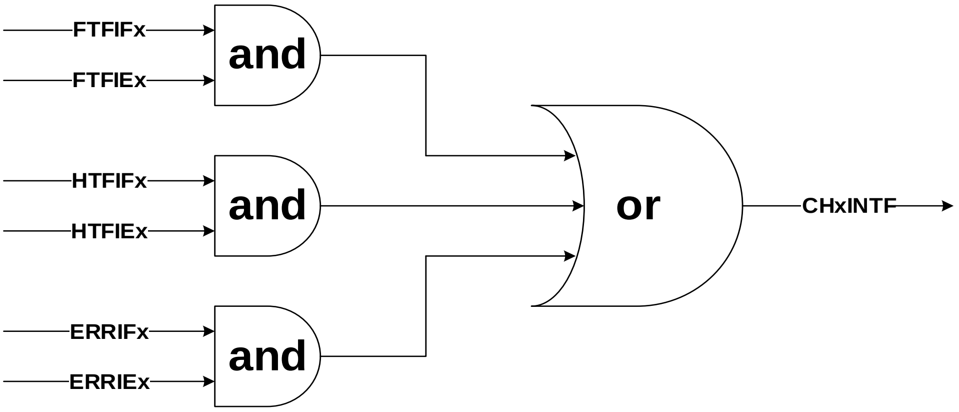
Each interrupt event has a dedicated flag bit in the DMA_INTF register, a dedicated clear bit in the DMA_INTC register, and a dedicated enable bit in the DMA_CHxCTL register. The relationship is described in the following ***Table 9-2. interrupt events***.

Table 9-2. interrupt events

Interrupt event	Flag bit	Clear bit	Enable bit
	DMA_INTF	DMA_INTC	DMA_CHxCTL
Full transfer finish	FTFIF	FTFIFC	FTFIE
Half transfer finish	HTFIF	HTFIFC	HTFIE
Transfer error	ERRIF	ERRIFC	ERRIE

The DMA interrupt logic is shown in the ***Figure 9-3. DMA interrupt logic***, an interrupt can be produced when any type of interrupt event occurs and enabled on the channel.

Figure 9-3. DMA interrupt logic



Note: “x” indicates channel number (x = 0…2).

9.4.9. DMA request mapping

The DMA requests of a channel are coming from the AHB / APB peripherals through the corresponding channel output of DMAMUX request multiplexer, refers to ***Table 10-2. Request multiplexer input mapping***.

9.5. Register definition

DMA base address: 0x4002 0000

9.5.1. Interrupt flag register (DMA_INTF)

Address offset: 0x00

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				ERRIF2	HTFIF2	FTFIF2	GIF2	ERRIF1	HTFIF1	FTFIF1	GIF1	ERRIF0	HTFIF0	FTFIF0	GIF0
				r	r	r	r	r	r	r	r	r	r	r	r

Bits	Fields	Descriptions
31:12	Reserved	Must be kept at reset value.
11/7/3	ERRIFx	Error flag of channel x (x=0…2) Hardware set and software cleared by configuring DMA_INTC register. 0: Transfer error has not occurred on channel x 1: Transfer error has occurred on channel x
10/6/2	HTFIFx	Half transfer finish flag of channel x (x=0…2) Hardware set and software cleared by configuring DMA_INTC register. 0: Half number of transfer has not finished on channel x 1: Half number of transfer has finished on channel x
9/5/1	FTFIFx	Full Transfer finish flag of channel x (x=0…2) Hardware set and software cleared by configuring DMA_INTC register. 0: Transfer has not finished on channel x 1: Transfer has finished on channel x
8/4/0	GIFx	Global interrupt flag of channel x (x=0…2) Hardware set and software cleared by configuring DMA_INTC register. 0: None of ERRIF, HTFIF or FTFIF occurs on channel x 1: At least one of ERRIF, HTFIF or FTFIF occurs on channel x

9.5.2. Interrupt flag clear register (DMA_INTC)

Address offset: 0x04

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				ERRIFC2	HTFIC2	FTFIFC2	GIFC2	ERRIFC1	HTFIFC1	FTFIFC1	GIFC1	ERRIFC0	HTFIFC0	FTFIFC0	GIFC0
				w	w	w	w	w	w	w	w	w	w	w	w

Bits	Fields	Descriptions
31:12	Reserved	Must be kept at reset value.
11/7/3	ERRIFCx	Clear bit for error flag of channel x (x=0…2) 0: No effect 1: Clear error flag
10/6/2	HTFIFCx	Clear bit for half transfer finish flag of channel x (x=0…2) 0: No effect 1: Clear half transfer finish flag
9/5/1	FTFIFCx	Clear bit for full transfer finish flag of channel x (x=0…2) 0: No effect 1: Clear full transfer finish flag
8/4/0	GIFCx	Clear global interrupt flag of channel x (x=0…2) 0: No effect 1: Clear GIFx, ERRIFx, HTFIFx and FTFIFx bits in the DMA_INTF register

9.5.3. Channel x control register (DMA_CHxCTL)

x = 0...2, where x is a channel number

Address offset: 0x08 + 0x14 × x

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	M2M	PRIO[1:0]		MWIDTH[1:0]		PWIDTH[1:0]		MNAGA	PNAGA	CMEN	DIR	ERRIE	HTFIE	FTFIE	CHEN
	rw	rw		rw		rw		rw	rw	rw	rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31:15	Reserved	Must be kept at reset value.
14	M2M	Memory to Memory mode Software set and cleared 0: Disable Memory to Memory mode 1: Enable Memory to Memory mode

		This bit must not be written when CHEN is ‘1’.
13:12	PRIO[1:0]	Priority level Software set and cleared 00: Low 01: Medium 10: High 11: Ultra high These bits can not be written when CHEN is ‘1’.
11:10	MWIDTH[1:0]	Transfer data size of memory Software set and cleared 00: 8-bit 01: 16-bit 10: 32-bit 11: Reserved These bits can not be written when CHEN is ‘1’.
9:8	PWIDTH[1:0]	Transfer data size of peripheral Software set and cleared 00: 8-bit 01: 16-bit 10: 32-bit 11: Reserved These bits can not be written when CHEN is ‘1’.
7	MNAGA	Next address generation algorithm of memory Software set and cleared 0: Fixed address mode 1: Increasing address mode This bit must not be written when CHEN is ‘1’.
6	PNAGA	Next address generation algorithm of peripheral Software set and cleared 0: Fixed address mode 1: Increasing address mode This bit must not be written when CHEN is ‘1’.
5	CMEN	Circular mode enable Software set and cleared 0: Disable circular mode 1: Enable circular mode This bit must not be written when CHEN is ‘1’.
4	DIR	Transfer direction Software set and cleared 0: Read from peripheral and write to memory 1: Read from memory and write to peripheral

		This bit must not be written when CHEN is ‘1’.
3	ERRIE	Enable bit for channel error interrupt Software set and cleared 0: Disable the channel error interrupt 1: Enable the channel error interrupt
2	HTFIE	Enable bit for channel half transfer finish interrupt Software set and cleared 0: Disable channel half transfer finish interrupt 1: Enable channel half transfer finish interrupt
1	FTFIE	Enable bit for channel full transfer finish interrupt Software set and cleared 0: Disable channel full transfer finish interrupt 1: Enable channel full transfer finish interrupt
0	CHEN	Channel enable Software set and cleared 0: Disable channel 1: Enable channel

9.5.4. Channel x counter register (DMA_CHxCNT)

x = 0...2, where x is a channel number

Address offset: 0x0C + 0x14 × x

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNT[15:0]															
rw															

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	CNT[15:0]	Transfer counter These bits can not be written when CHEN in the DMA_CHxCTL register is ‘1’. This register indicates how many transfers remain. Once the channel is enabled, it is read-only, and decreases after each DMA transfer. If the register is zero, no transaction can be issued whether the channel is enabled or not. Once the transmission of the channel is complete, the register can be reloaded automatically by the previously programmed value if the channel is configured in circular mode.

9.5.5. Channel x peripheral base address register (DMA_CHxPADDR)

x = 0...2, where x is a channel number
Address offset: 0x10 + 0x14 × x
Reset value: 0x0000 0000

This register has to be accessed by word (32-bit)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PADDR[31:16]															
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PADDR[15:0]															
rw															

Bits	Fields	Descriptions
31:0	PADDR[31:0]	Peripheral base address These bits can not be written when CHEN in the DMA_CHxCTL register is ‘1’. When PWIDTH is 01 (16-bit), the LSB of these bits is ignored. Access is automatically aligned to a half word address. When PWIDTH is 10 (32-bit), the two LSBs of these bits are ignored. Access is automatically aligned to a word address.

9.5.6. Channel x memory base address register (DMA_CHxMADDR)

x = 0...2, where x is a channel number
Address offset: 0x14 + 0x14 × x
Reset value: 0x0000 0000

This register has to be accessed by word (32-bit)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MADDR[31:16]															
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MADDR[15:0]															
rw															

Bits	Fields	Descriptions
31:0	MADDR[31:0]	Memory base address These bits can not be written when CHEN in the DMA_CHxCTL register is ‘1’. When MWIDTH in the DMA_CHxCTL register is 01 (16-bit), the LSB of these bits is ignored. Access is automatically aligned to a half word address. When MWIDTH in the DMA_CHxCTL register is 10 (32-bit), the two LSBs of these bits are ignored. Access is automatically aligned to a word address.

10. DMA request multiplexer (DMAMUX)

10.1. Overview

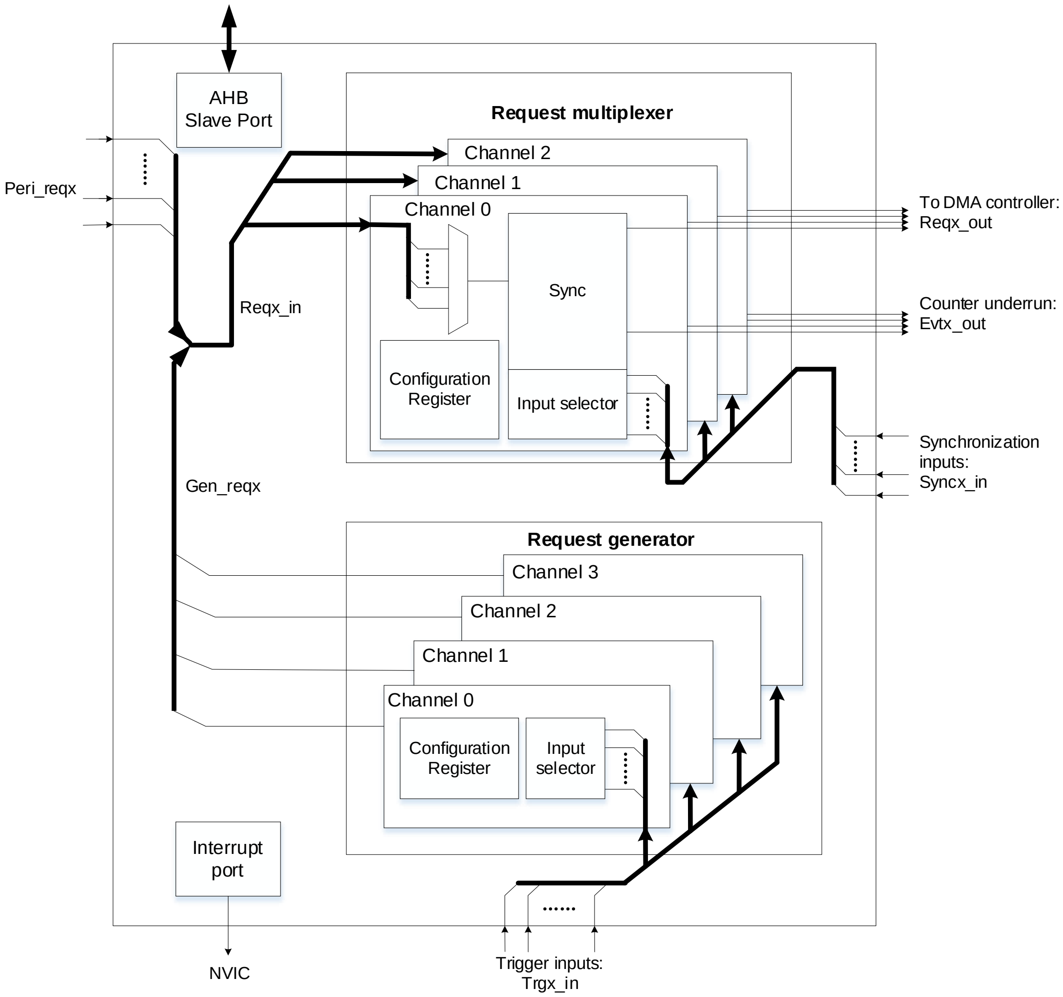
DMAMUX is a transmission scheduler for DMA requests. The DMAMUX request multiplexer is used for routing a DMA request line between the peripherals / generated DMA request (from the DMAMUX request generator) and the DMA controller. Each DMAMUX request multiplexer channel selects a unique DMA request line, unconditionally or synchronously with events from its DMAMUX synchronization inputs. The DMA request is pending until it is served by the DMA controller which generates a DMA acknowledge signal (the DMA request signal is de-asserted).

10.2. Characteristics

- 3 channels for DMAMUX request multiplexer.
- 4 channels for DMAMUX request generator.
- Support 24 trigger inputs.
- Support 24 synchronization inputs.
- Each DMAMUX request generator channel has a DMA request trigger input selector, a DMAMUX request generator counter, and the trigger overrun flag.
- Each DMAMUX request multiplexer channel has input DMA request lines from peripherals ,input DMA request number is 55 channels, a synchronization input selector, one DMA request line output, one channel event output for DMA request chaining, a DMAMUX request multiplexer counter, and the synchronization overrun flag.

10.3. Block diagram

Figure 10-1. Block diagram of DMAMUX



10.4. Signal description

The DMAMUX signals are described as follows:

- Reqx_in: DMAMUX request multiplexer inputs from peripheral requests and request generator channels.
- Peri_reqx: DMAMUX DMA request line inputs from peripherals.
- Gen_reqx: DMAMUX generated DMA request from request generator.
- Reqx_out: DMAMUX requests outputs to DMA controller.
- Trgx_in: DMAMUX DMA request triggers inputs to request generator.
- Syncx_in: DMAMUX synchronization inputs to request multiplexer.
- Evtx_out: DMAMUX request multiplexer counter underrun event outputs.

10.5. Function overview

As shown in ***Figure 10-1. Block diagram of DMAMUX***, DMAMUX includes two sub-blocks:

- DMAMUX request multiplexer.
  DMAMUX request multiplexer inputs (Reqx_in) source from:
  - Peripherals (Peri_reqx).
  - DMAMUX request generator outputs (Gen_reqx).

  DMAMUX request multiplexer outputs (Reqx_out) is connected to channels of DMA controller.
  Synchronization inputs (Syncx_in) source from internal or external signals.
- DMAMUX request generator.
  Trigger inputs (Trgx_in) source from internal or external signals.

10.5.1. DMAMUX request multiplexer

The DMAMUX request multiplexer enables routing a DMA request line between the peripherals / generated DMA request and the DMA controllers of the product. Its component unit is the request multiplexer channels. DMA request lines are connected in parallel to all request multiplexer channels. There is a synchronization unit for each request multiplexer channel. The synchronization inputs are connected in parallel to all synchronization unit of request multiplexer channels. And there is a built-in DMAMUX request multiplexer counter for each request multiplexer channel.

Request multiplexer channel

A DMA request input for the DMAMUX request multiplexer channel x is configured by the MUXID[5:0] / MUXID[6:0] bits in the DMAMUX_RM_CHxCFG register, sourced either from the peripherals or from the DMAMUX request generator, the sources can refer to ***Table 10-2. Request multiplexer input mapping***. A DMAMUX request multiplexer channel is connected and dedicated to one single channel of the DMA controller.

Note: The value 0 of MUXINID[6:0] bits corresponds to no DMA request line is selected. It is not allowed to configure the same DMA request line (same non-null MUXINID[6:0]) to two different request multiplexer channels.

When synchronization mode is disabled

Each time the connected DMAMUX request is served by the DMA controller, the served DMA request is de-asserted, and the built-in DMAMUX request multiplexer counter is decremented. At the request multiplexer counter underrun, the built-in DMAMUX request multiplexer counter is automatically loaded with the value in NBR[4:0] bits of the DMAMUX_RM_CHxCFG register. If the channel event generation is enabled by setting EVGEN bit, the number of DMA requests before an output event generation is NBR[4:0] + 1.

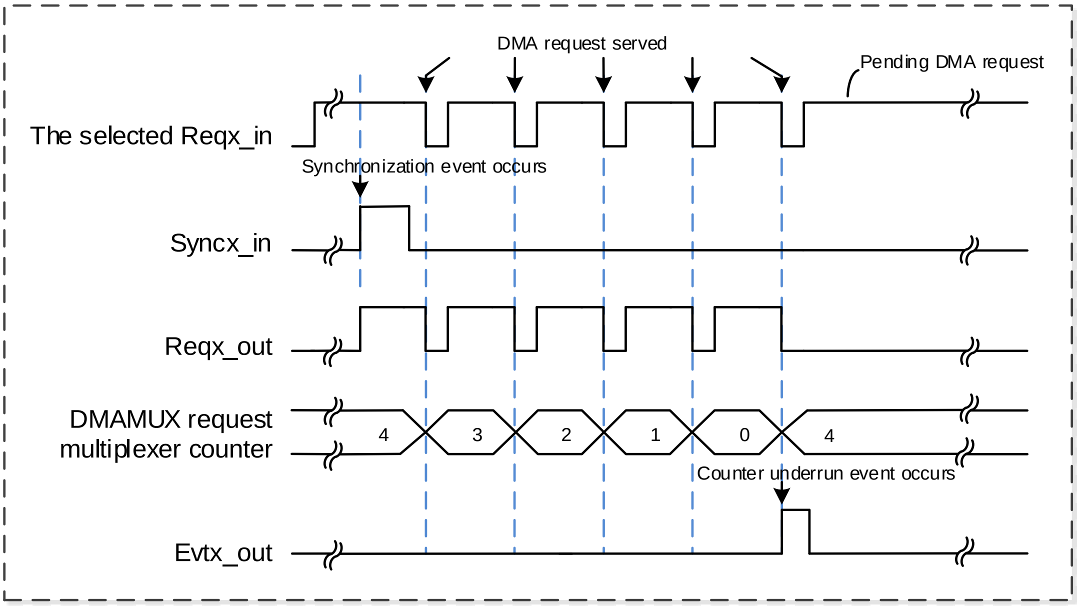
Note: The NBR[4:0] bits value shall only be written by software when both synchronization enable bit SYNCEN and event generation enable EVGEN bit of the corresponding request multiplexer channel x are disabled.

When synchronization mode is enabled

A channel x in synchronization mode, when a rising/falling edge on the selected synchronization input is detected, the pending selected input DMA request line is routed to the multiplexer channel x output. Each time the connected DMAMUX request is served by the DMA controller, the served DMA request is de-asserted, and the built-in DMAMUX request multiplexer counter is decremented. At the request multiplexer counter underrun, the input DMA request line is disconnected from the request multiplexer channel x output, and the built-in DMAMUX request multiplexer counter is automatically loaded with the value in NBR[4:0] bits of the DMAMUX_RM_CHxCFG register. The number of DMA requests transferred to the request multiplexer channel x output following a detected synchronization event is NBR[4:0] + 1.

Figure 10-2. Synchronization mode shows an example when NBR[4:0] = 4, SYNCEN = 1, EVGEN = 1, SYNCP[1:0] = 01.

Figure 10-2. Synchronization mode


DMA request served
Pending DMA request
The selected Reqx_in
Synchronization event occurs
Syncx_in
Reqx_out
DMAMUX request multiplexer counter
4
3
2
1
0
4
Counter underrun event occurs
Evtx_out

DMAMUX request multiplexer channel x can be synchronized by setting the synchronization enable bit SYNCEN in the DMAMUX_RM_CHxCFG register. The synchronization input is selected by SYNCID[4:0] bits in the DMAMUX_RM_CHxCFG register, the sources can refer to *Table 10-4. Synchronization input mapping*. The synchronization input valid edge is configured by the SYNCP[1:0] bits of the DMAMUX_RM_CHxCFG register.

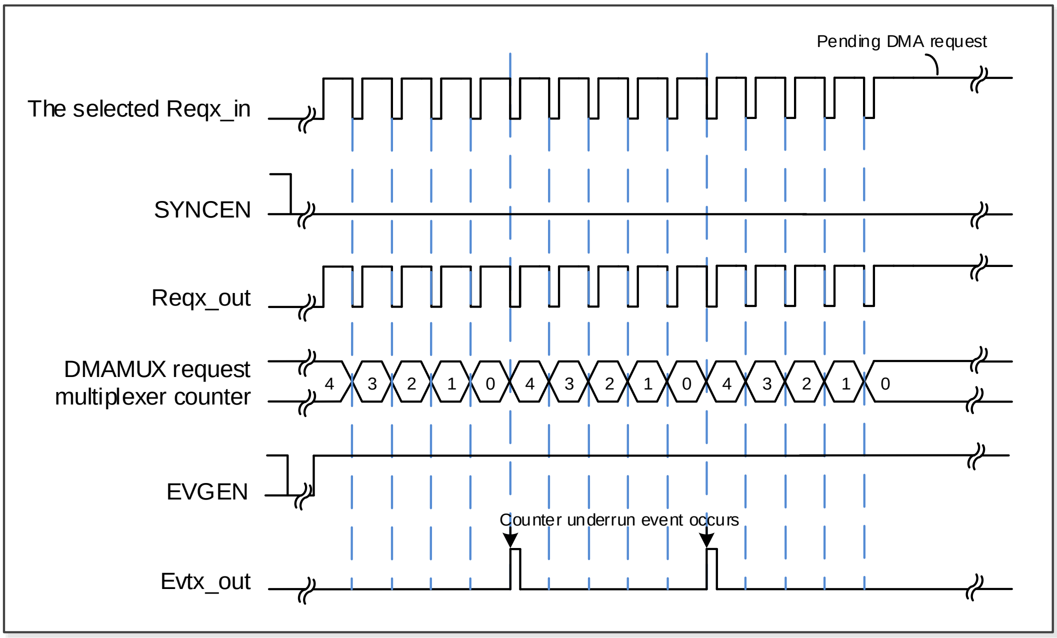
Note: If a synchronization input event occurs when there is no pending selected input DMA request line, the input event is discarded. The following asserted input request lines will not be routed to the DMAMUX multiplexer channel output until a synchronization input event occurs again.

Channel event generation

Each DMA request line multiplexer channel has an event output called Evtx_out, which is the DMA request multiplexer counter underrun event. Signals Evt0_out ~ Evt3_out can be used for DMA request chaining. If event generation bit EVGEN in the DMAMUX_RM_CHxCFG register is enabled on the channel x output, when its DMA request multiplexer counter is automatically reloaded with the value of the programmed NBR[4:0] field, the multiplexer channel generates a channel event, as a pulse of one AHB clock cycle.

Figure 10-3. Event generation shows an example when NBR[4:0] = 4, SYNCEN = 0, EVGEN = 1.

Figure 10-3. Event generation



Note: If EVGEN = 1 and NBR[4:0] = 0, an event is generated after each served DMA request.

Synchronization overrun

If a new synchronization event occurs before the built-in DMAMUX request multiplexer counter underrun, the synchronization overrun flag bit SOIFx is set in the DMAMUX_RM_INTF register.

Note: The synchronization mode of request multiplexer channel x shall be disabled by resetting SYNCEN bit in DMAMUX_RM_CHxCFG register at the completion of the use of the related channel of the DMA controller. Otherwise, when a new synchronization event occurs, there will be a synchronization overrun due to the absence of a DMA acknowledge (that is, no served request) received from the DMA controller.

10.5.2. DMAMUX request generator

The DMAMUX request generator produces DMA requests upon trigger input event. Its component unit is the request generator channels. DMA request trigger inputs are connected in parallel to all request generator channels. And there is a built-in DMAMUX request generator counter for each request generator channel.

The active edge of trigger input events is selected through the RGTP[1:0] bits in DMAMUX_RG_CHxCFG register. The DMA request trigger input for the DMAMUX request generator channel x is selected through the TID[4:0] bits in DMAMUX_RG_CHxCFG register, the sources can refer to ***Table 10-3. Trigger input mapping***. DMAMUX request generator channel x can be enabled by setting RGEN to 1 in DMAMUX_RG_CHxCFG register.

Request generator channel

Upon the trigger input event, the corresponding request generator channel starts generating DMA requests on its output, and the output goes to the input of the DMAMUX request multiplexer. Each time the DMAMUX generated request is served by the connected DMA controller, the served request will be de-asserted, and the built-in DMAMUX request generator counter of the request generator channel is decremented. At the request generator counter underrun, the request generator channel stops generating DMA requests. The built-in DMAMUX request generator counter will be automatically reloaded to its programmed value upon the next trigger input event, the built-in counter is programmed by the NBRG[4:0] bits of the DMAMUX_RG_CHxCFG register.

Note: The number of generated DMA requests after the trigger input event is NBRG[4:0] + 1. The NBRG[4:0] value shall only be written by software when the RGEN bit of the corresponding generator channel x is disabled.

Trigger overrun

If a request generator channel x was enabled by RGEN bit, when a new DMA request trigger event for the request generator channel x occurs before the DMAMUX request generator counter underrun, then the request trigger overrun event flag bit TOIFx is set by hardware in the DMAMUX_RG_INTF register.

Note: The request generator channel x shall be disabled by resetting RGEN bit in DMAMUX_RG_CHxCFG register at the completion of the usage of the related channel of the DMA controller. Otherwise, when a new detected trigger event occurs, there will be a trigger overrun due to the absence of an acknowledge (that is, no served request) received from the DMA.

10.5.3. Channel configurations

The following sequence should be followed to configure a DMAMUX channel y and the related DMA channel x:

1. Set and configure the DMA channel x completely, except enabling the channel x.
2. Set and configure the related DMAMUX channel y completely.
3. Configure the CHEN bit with '1' in the DMA_CHxCTL register to enable the DMA channel x.

10.5.4. Interrupt

There are two types of interrupt event, including synchronization overrun event on each DMAMUX request multiplexer channel, and trigger overrun event on each DMAMUX request generator channel.

Each interrupt event has a dedicated flag bit, a dedicated clear bit, and a dedicated enable bit. The relationship is described in the following ***Table 10-1. Interrupt events***.

Table 10-1. Interrupt events

Interrupt event	Flag bit	Clear bit	Enable bit
Synchronization overrun event on DMAMUX request multiplexer channel x	SOIFx in DMAMUX_RM_INTF register	SOIFCx in DMAMUX_RM_INTC register	SOIE in DMAMUX_RM_CHxCFG register
Trigger overrun event on DMAMUX request generator channel y	TOIFy in DMAMUX_RG_INTF register	TOIFCy in DMAMUX_RG_INTC register	TOIE in DMAMUX_RG_CHxCFG register

Trigger overrun interrupt

When the DMAMUX request trigger overrun flag TOIFx is set, and the trigger overrun interrupt is enabled by setting TOIE bit, a trigger overrun interrupt will be generated. The overrun flag TOIFx is reset by writing 1 to the corresponding clear bit of overrun flag TOIFCx in the DMAMUX_RG_INTC register.

Synchronization overrun interrupt

When the synchronization overrun flag SOIFx is set, and the synchronization overrun interrupt is enabled by setting SOIE bit, a synchronization overrun interrupt will be generated. The overrun flag SOIFx is reset by writing 1 to the corresponding clear bit of synchronization overrun flag bit SOIFCx in the DMAMUX_RM_INTC register.

10.5.5. DMAMUX mapping

Request multiplexer input mapping

A DMA request is sourced either from the peripherals or from the DMAMUX request generator, the sources can refer to ***Table 10-2. Request multiplexer input mapping for*** configured by the MUXINID[6:0] bits in the DMAMUX_RM_CHxCFG register for the DMAMUX request multiplexer channel x.

Table 10-2. Request multiplexer input mapping for GD32C2x1

Request multiplexer channel input identification MUXINID[5:0]	Source
1	Gen_reqx0
2	Gen_reqx1
3	Gen_reqx2
4	Gen_reqx3
5	ADC
6	Reserved
7	Reserved
8	Reserved
9	Reserved
10	I2C0_RX
11	I2C0_TX
12	I2C1_RX
13	I2C1_TX
14	Reserved
15	Reserved
16	SPI0_RX
17	SPI0_TX
18	SPI1_RX
19	SPI1_TX
20	TIMER0_CH0
21	TIMER0_CH1
22	TIMER0_CH2
23	TIMER0_CH3
24	TIMER0_TRIG
25	TIMER0_UP
26	TIMER0_COM
27	Reserved
28	Reserved
29	Reserved
30	Reserved
31	Reserved
32	TIMER2_CH0
33	TIMER2_CH1
34	TIMER2_CH2
35	TIMER2_CH3
36	TIMER2_TRIG
37	TIMER2_UP
38	Reserved
39	Reserved

Request multiplexer channel input identification MUXINID[5:0]	Source
40	Reserved
41	Reserved
42	Reserved
43	Reserved
44	TIMER15_CH0
45	Reserved
46	TIMER15_UP
47	TIMER16_CH0
48	Reserved
49	TIMER16_UP
50	USART0_RX
51	USART0_TX
52	USART1_RX
53	USART1_TX
54	USART2_RX
55	USART2_TX

Trigger input mapping

The DMA request trigger input for the DMAMUX request generator channel x is selected through the TID[4:0] bits in DMAMUX_RG_CHxCFG register, the sources can refer to ***Table 10-3. Trigger input mapping***.

Table 10-3. Trigger input mapping

Trigger input identification TID[4:0]	Source
0	EXTI_0
1	EXTI_1
2	EXTI_2
3	EXTI_3
4	EXTI_4
5	EXTI_5
6	EXTI_6
7	EXTI_7
8	EXTI_8
9	EXTI_9
10	EXTI_10
11	EXTI_11
12	EXTI_12
13	EXTI_13
14	EXTI_14

Trigger input identification TID[4:0]	Source
15	EXTI_15
16	Evtx_out0
17	Evtx_out1
18	Evtx_out2
19	Reserved
20	Reserved
21	TIMER13_O
22	Reserved
23	Reserved

Synchronization input mapping

The synchronization input is selected by SYNCID[4:0] bits in the DMAMUX_RM_CHxCFG register, the sources can refer to ***Table 10-4. Synchronization input mapping***.

Table 10-4. Synchronization input mapping

Synchronization input identification SYNCID[4:0]	Source
0	EXTI_0
1	EXTI_1
2	EXTI_2
3	EXTI_3
4	EXTI_4
5	EXTI_5
6	EXTI_6
7	EXTI_7
8	EXTI_8
9	EXTI_9
10	EXTI_10
11	EXTI_11
12	EXTI_12
13	EXTI_13
14	EXTI_14
15	EXTI_15
16	Evtx_out0
17	Evtx_out1
18	Evtx_out2
19	Reserved
20	Reserved
21	TIMER13_O
22	Reserved
23	Reserved

10.6. Register definition

DMAMUX base address: 0x4002 0800

10.6.1. Request multiplexer channel x configuration register (DMAMUX_RM_CHxCFG)

x = 0...2, where x is a channel number

Address offset: 0x00 + 0x04 × x

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved			SYNCID[4:0]					NBR[4:0]					SYNCP[1:0]		SYNCEN
			rw					rw					rw		rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved						EVGEN	SOIE	Reserved	MUXINID[6:0]						
						rw	rw		rw						

Bits	Fields	Descriptions
31:29	Reserved	Must be kept at reset value.
28:24	SYNCID[4:0]	Synchronization input identification Selects the synchronization input source.
23:19	NBR[4:0]	Number of DMA requests to forward The number of DMA requests to forward to the DMA controller after a synchronization event / before an output event is generated equals to NBR[4:0] + 1. These bits shall only be written when both SYNCEN and EVGEN bits are disabled.
18:17	SYNCP[1:0]	Synchronization input polarity 00: No event detection 01: Rising edge 10: Falling edge 11: Rising and falling edges
16	SYNCEN	Synchronization enable 0: Disable synchronization 1: Enable synchronization
15:10	Reserved	Must be kept at reset value.
9	EVGEN	Event generation enable 0: Disable event generation

		1: Enable event generation
8	SOIE	Synchronization overrun interrupt enable 0: Disable interrupt 1: Enable interrupt
7	Reserved	Must be kept at reset value.
6:0	MUXINID[6:0]	Multiplexer input identification Selects the input DMA request in multiplexer input sources.

10.6.2. Request multiplexer channel interrupt flag register (DMAMUX_RM_INTF)

Address offset: 0x80

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved													SOIF2	SOIF1	SOIF0
													r	r	r

Bits	Fields	Descriptions
31:3	Reserved	Must be kept at reset value.
2	SOIF2	Synchronization overrun event flag of request multiplexer channel 2 Reference the description of SOIF0
1	SOIF1	Synchronization overrun event flag of request multiplexer channel 1 Reference the description of SOIF0
0	SOIF0	Synchronization overrun event flag of request multiplexer channel 0 The flag is set when a synchronization event occurs on a DMA request line multiplexer channel x, while the DMA request counter value is lower than NBR[4:0]. The flag is cleared by writing 1 to the corresponding SOIFC0 bit in DMAMUX_RM_INTC register.

10.6.3. Request multiplexer channel interrupt flag clear register (DMAMUX_RM_INTC)

Address offset: 0x084

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

Reserved

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved													SOIFC2	SOIFC1	SOIFC0
													w	w	w

Bits	Fields	Descriptions
31:3	Reserved	Must be kept at reset value.
2	SOIFC2	Clear bit for synchronization overrun event flag of request multiplexer channel 2 Reference the description of SOIFC0
1	SOIFC1	Clear bit for synchronization overrun event flag of request multiplexer channel 1 Reference the description of SOIFC0
0	SOIFC0	Clear bit for synchronization overrun event flag of request multiplexer channel 0 Writing 1 clears the corresponding overrun flag SOIF0 in the DMAMUX_RM_INTF register.

10.6.4. Request generator channel x configuration register (DMAMUX_RG_CHxCFG)

x = 0...3, where x is a channel number

Address offset: 0x100 + 0x04 × x
Reset value: 0x0000 0000

This register has to be accessed by word (32-bit)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved								NBRG[4:0]					RGTP[1:0]		RGEN
								rw					rw		rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved							TOIE	Reserved			TID[4:0]				
							rw				rw				

Bits	Fields	Descriptions
31:24	Reserved	Must be kept at reset value.
23:19	NBRG[4:0]	Number of DMA requests to be generated The number of DMA requests to be generated after a trigger event equals to NBRG[4:0] + 1. **Note:** These bits shall only be written when RGEN bit is disabled.
18:17	RGTP[1:0]	DMA request generator trigger polarity 00: No event trigger detection 01: Rising edge

		10: Falling edge 11: Rising and falling edges
16	RGEN	DMA request generator channel x enable 0: Disable DMA request generator channel x 1: Enable DMA request generator channel x
15:9	Reserved	Must be kept at reset value.
8	TOIE	Trigger overrun interrupt enable 0: Disable interrupt 1: Enable interrupt
7:5	Reserved	Must be kept at reset value.
4:0	TID[4:0]	Trigger input identification Selects the DMA request trigger input source.

10.6.5. Request generator interrupt flag register (DMAMUX_RG_INTF)

Address offset: 0x140

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved												TOIF3	TOIF2	TOIF1	TOIF0
													r	r	r

Bits	Fields	Descriptions
31:4	Reserved	Must be kept at reset value.
3	TOIF3	Trigger overrun event flag of request generator channel 3 Reference the description of TOIF0
2	TOIF2	Trigger overrun event flag of request generator channel 2 Reference the description of TOIF0
1	TOIF1	Trigger overrun event flag of request generator channel 1 Reference the description of TOIF0
0	TOIF0	Trigger overrun event flag of request generator channel 0 The flag is set when a new trigger event occurs on DMA request generator channel x, before the request counter underrun (the internal request counter programmed via the NBRG[4:0] bits of the DMAMUX_RG_CHxCFG register). The flag is cleared by writing 1 to the corresponding TOIFC0 bit in the DMAMUX_RG_INTC register.

10.6.6. Rquest generator interrupt flag clear register (DMAMUX_RG_INTC)

Address offset: 0x144

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved												TOIFC3	TOIFC2	TOIFC1	TOIFC0
												w	w	w	w

Bits	Fields	Descriptions
31:4	Reserved	Must be kept at reset value.
3	TOIFC3	Clear bit for trigger overrun event flag of request generator channel 3 Reference the description of TOIFC0
2	TOIFC2	Clear bit for trigger overrun event flag of request generator channel 2 Reference the description of TOIFC0
1	TOIFC1	Clear bit for trigger overrun event flag of request generator channel 1 Reference the description of TOIFC0
0	TOIFC0	Clear bit for trigger overrun event flag of request generator channel 0 Writing 1 in each bit clears the corresponding overrun flag TOIF0 in the DMAMUX_RG_INTF register.

11. Analog-to-digital converter (ADC)

11.1. Overview

A 12-bit successive approximation analog-to-digital converter module(ADC) is integrated on the MCU chip, which can sample analog signals from 13 external channels and 3 internal channels. The ADC sampling channels all support a variety of operation modes. After sampling and conversion, the conversion results can be stored in the corresponding data registers according to the least significant bit (LSB) alignment or the most significant bit (MSB) alignment. An on-chip hardware oversample scheme improves performances and reduces the computational burden of MCU.

For motors, power supplies and other applications that have a higher demand for ADC, you can contact our sales staff for more ADC details.

11.2. Characteristics

- High performance.
  - ADC sampling resolution: 12-bit, 10-bit, 8-bit or 6-bit.
  - Programmable sampling time.
  - Data storage mode: the most significant bit and the least significant bit.
  - DMA support.
- Multi clock domain architecture.
  - System clock, IRC48MDIV_PER clock.
- Analog input channels.
  - Up to 13 external analog inputs.
  - 1 internal channel for internal temperature sensor (V_{SENSE}).
  - 1 internal channel for internal reference voltage (V_{REFINT}).
  - 1 internal channel for positive reference voltage (V_{REFP})
- Start-of-conversion can be initiated.
  - By software.
  - By hardware triggers.
- Operation modes.
  - Converts a single channel or scans a sequence of channels.
  - Single operation mode converts selected inputs once per trigger.
  - Continuous operation mode converts selected inputs continuously.
  - Discontinuous operation mode.
- Conversion result threshold monitor function: analog watchdog.
- Interrupt generation.
  - At the end of routine sequence conversion.
  - Analog watchdog event.
- Oversampling.
  - 16-bit data register.

  - Oversampling ratio adjustable from 2x to 256x.
  - Programmable data shift up to 8-bit.
- Channel input range: V_{REFN} ≤V_{IN} ≤V_{REFP}.

11.3. Pins and internal signals

Figure 11-1. ADC module block diagram shows the ADC block diagram. *Table 11-1. ADC internal input /* **output signals** gives the ADC internal signals and *Table 11-2. ADC input pins definition* gives the ADC pins description.

Table 11-1. ADC internal input / output signals

Internal signal name	Description
V_{SENSE}	Internal temperature sensor output voltage
V_{REFINT}	Internal voltage reference output voltage
ADC_WDx_OUT	Analog watchdog x output signal and connected to the TIMER module (x=0,1,2)

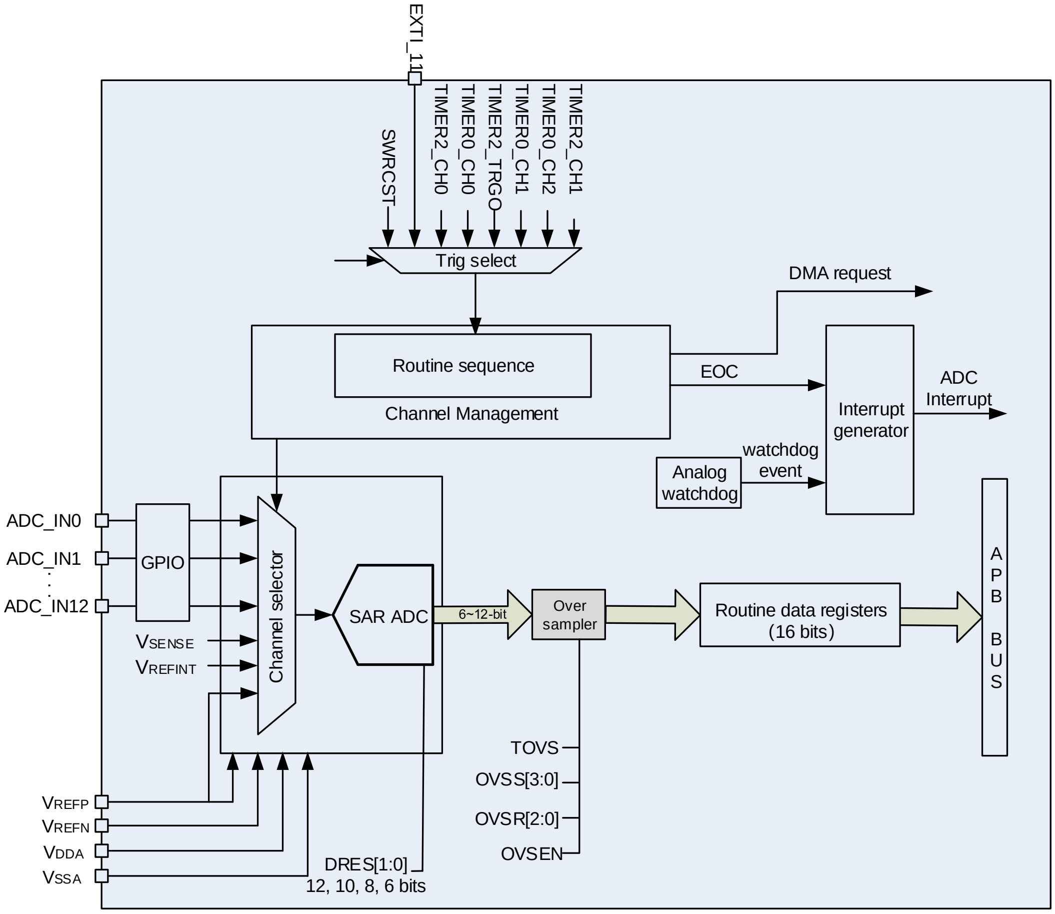
Table 11-2. ADC input pins definition

Name	Description
V_{DDA}	Analog power supply equals to V_{DD}
V_{SSA}	Ground for analog power supply equals to V_{SS}
V_{REFP}	The positive reference voltage for the ADC
V_{REFN}	The negative reference voltage for the ADC
ADCx_IN[12:0]	Up to 13 external channels

Note: V_{DDA} and V_{SSA} have to be connected to V_{DD} and V_{SS} respectively.

11.4. Function overview

Figure 11-1. ADC module block diagram



11.4.1. Multi clock domain architecture

In addition to the system clock CK_SYS, the ADC sub-module clock can also be feed by the IRC48MDIV_PER clock. When using the IRC48MDIV_PER or HXTAL clock as the ADC clock, application can reduce the system clock frequency for low power operation while still keeping optimum ADC performance. The maximum ADC frequency is 24MHz.

Refer to RCU Section *4.2.1* for more information on generating this clock source.

11.4.2. ADC enable

The ADCON bit on the ADC_CTL1 register is the enable switch of the ADC module. The ADC module will keep in reset state if this bit is 0. For power saving, when this bit is reset, the analog sub-module will enter power down mode. After ADC is enabled, you need delay $t_{ST(ADC)}$ time for sampling (the value of $t_{ST(ADC)}$ please refer to the device datasheet).

11.4.3. Routine sequence

The channel management circuit can organize the sampling conversion channels into a sequence: routine sequence.

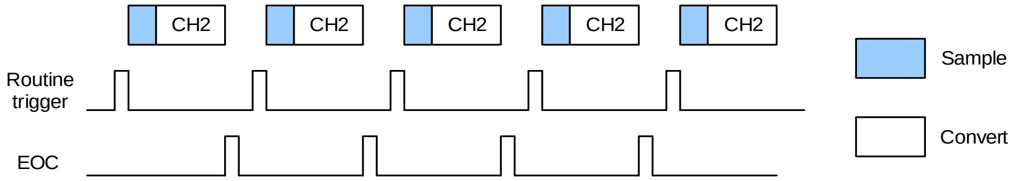
The routine sequence supports up to 16 channels. Each channel is called routine channel. The RL[3:0] bits in the ADC_RSQ0 register specify the total conversion sequence length. The ADC_RSQ0~ADC_RSQ2 registers specify the selected channels of the routine sequence.

11.4.4. Operation modes

Single operation mode

In the single operation mode, the ADC performs conversion on the channel specified in the RSQ0[3:0] bits of ADC_RSQ2. When the ADCON has been set high, the ADC samples and converts a single channel, once the corresponding software trigger or external trigger is active.

Figure 11-2. Single operation mode



After conversion of a single routine channel, the conversion data will be stored in the ADC_RDATA register, the EOC will be set. An interrupt will be generated if the EOCIE bit is set.

Software procedure for single operation mode of a routine channel:

1. Make sure the DISRC, SM bits in the ADC_CTL0 register and CTN bit in the ADC_CTL1 register are reset.
2. Configure the RSQ0[3:0] bits in ADC_RSQ2 register with the analog channel number.
3. Configure the ADC_SAMPTx register.
4. Configure the ETERC and ETSRC bits in the ADC_CTL1 register if in need.
5. Set the SWRCST bit, or generate an external trigger for the routine sequence.
6. Wait the EOC flag to be set.
7. Read the converted data from the ADC_RDATA register.
8. Clear the EOC flag by writing 0 to it.

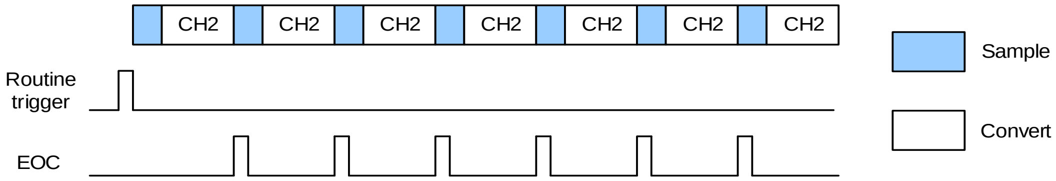
Note: After EOC is set, a delay of one CK_ADC is required before reading the ADC conversion result.

Continuous operation mode

The continuous operation mode will be enabled when the CTN bit in the ADC_CTL1 register is set. In this mode, the ADC performs conversion on the channel specified in the RSQ0[3:0].

When the ADCON has been set high, the ADC samples and converts specified channel, once the corresponding software trigger or external trigger is active. The conversion data will be stored in the ADC_RDATA register.

Figure 11-3. Continuous operation mode



Software procedure for continuous operation mode on a routine channel:

1. Set the CTN bit in the ADC_CTL1 register.
2. Configure the RSQ0[3:0] bits in ADC_RSQ2 register with the analog channel number.
3. Configure the ADC_SAMPTx register.
4. Configure the ETERC and ETSRC bits in the ADC_CTL1 register if in need.
5. Set the SWRCST bit, or generate an external trigger for the routine sequence.
6. Wait the EOC flag to be set.
7. Read the converted data in the ADC_RDATA register.
8. Clear the EOC flag by writing 0 to it.
9. Repeat steps 6~8 as soon as the conversion is in need.

Note: After EOC is set, a delay of one CK_ADC is required before reading the ADC conversion result.

To get rid of checking EOC bit, DMA can be used to transfer the converted data:

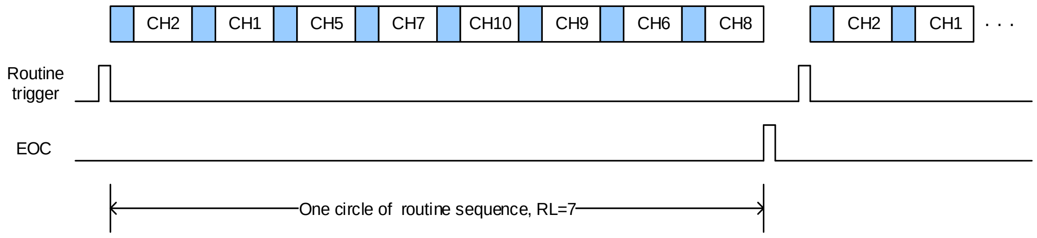
1. Set the CTN and DMA bits in the ADC_CTL1 register.
2. Configure the RSQ0[3:0] bits in ADC_RSQ2 register with the analog channel number.
3. Configure the ADC_SAMPTx register.
4. Configure the ETERC and ETSRC bits in the ADC_CTL1 register if in need.
5. Prepare the ***Direct memory access controller (DMA)*** module to transfer data from the ADC_RDATA.
6. Set the SWRCST bit, or generate an external trigger for the routine sequence.

Scan operation mode

The scan operation mode will be enabled when the SM bit in the ADC_CTL0 register is set. In this mode, the ADC performs conversion on the channels with a specific routine sequence specified in the ADC_RSQ0~ADC_RSQ2 registers. When the ADCON has been set high, the ADC samples and converts specified channels one by one in the routine sequence till the end of the routine sequence, once the corresponding software trigger or external trigger is active. The conversion data will be stored in the ADC_RDATA register. After conversion of the routine sequence, the EOC will be set. An interrupt will be generated if the EOCIE bit is set. The DMA bit in ADC_CTL1 register must be set when the routine sequence works in scan mode.

After conversion of a routine sequence, the conversion can be restarted automatically if the CTN bit in the ADC_CTL1 register is set.

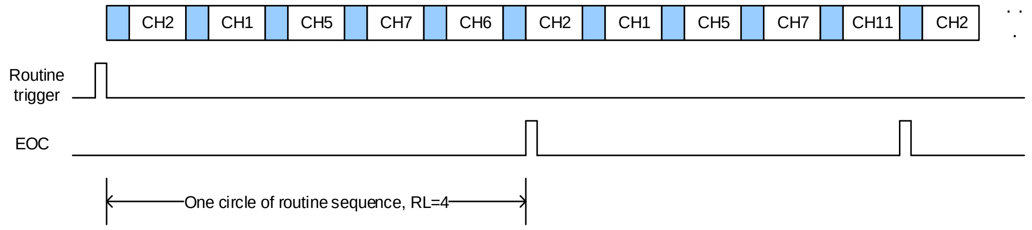
Figure 11-4. Scan operation mode, continuous disable



Software procedure for scan operation mode on a routine sequence:

1. Set the SM bit in the ADC_CTL0 register and the DMA bit in the ADC_CTL1 register.
2. Configure the ADC_RSQx and ADC_SAMPTx registers.
3. Configure the ETERC and ETSRC bits in the ADC_CTL1 register if in need.
4. Prepare the ***Direct memory access controller (DMA)*** module to transfer data from the ADC_RDATA.
5. Set the SWRCST bit, or generate an external trigger for the routine sequence.
6. Wait the EOC flag to be set.
7. Clear the EOC flag by writing 0 to it.

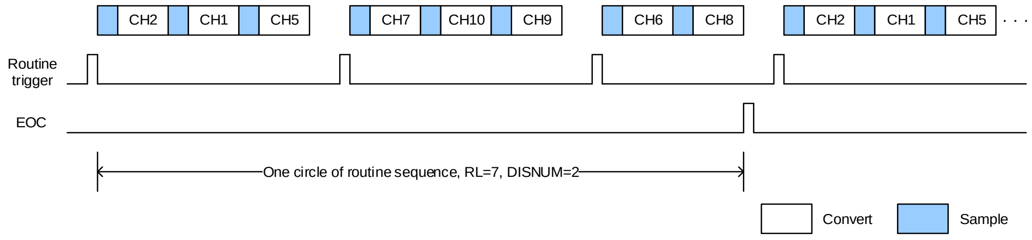
Figure 11-5. Scan operation mode, continuous enable



Discontinuous operation mode

For routine sequence, the discontinuous operation mode will be enabled when the DISRC bit in the ADC_CTL0 register is set. In this mode, the ADC performs a short sequence of n conversions (n does not exceed 8) which is a part of the sequence of conversions selected in the ADC_RSQ0~ADC_RSQ2 registers. The value of n is configured by the DISNUM[2:0] bits in the ADC_CTL0 register. When the corresponding software trigger or external trigger is active, the ADC samples and converts the next n channels configured in the ADC_RSQ0~ADC_RSQ2 registers until all the channels of routine sequence are done. The EOC will be set after every circle of the routine sequence. An interrupt will be generated if the EOCIE bit is set.

Figure 11-6. Discontinuous operation mode



Software procedure for discontinuous operation mode on a routine sequence:

1. Set the DISRC bit in the ADC_CTL0 register and the DMA bit in the ADC_CTL1 register.
2. Configure the DISNUM [2:0] bits in the ADC_CTL0 register.
3. Configure the ADC_RSQx and ADC_SAMPTx registers.
4. Configure the ETERC and ETSRC bits in the ADC_CTL1 register if in need.
5. Prepare the ***Direct memory access controller (DMA)*** module to transfer data from the ADC_RDATA.
6. Set the SWRCST bit, or generate an external trigger for the routine sequence.
7. Repeat step6 if in need.
8. Wait the EOC flag to be set.
9. Clear the EOC flag by writing 0 to it.

11.4.5. Conversion result threshold monitor

Analog watchdog 0

The analog watchdog 0 is enabled when the RWD0EN bit in the ADC_CTL0 register is set for routine sequence. This function is used to monitor whether the conversion result exceeds the set thresholds. The WD0E bit in ADC_STAT register will be set if the conversion result exceeds the thresholds. An interrupt will be generated if the WD0EIE bit is set. The ADC_WD0HT and ADC_WD0LT registers are used to specify the high and low threshold. The comparison is done before the alignment, so the threshold value is independent of the alignment, which is specified by the DAL bit in the ADC_CTL1 register. One or more channels, which are selected by the RWD0EN, WD0SC and WD0CHSEL [3:0] bits in ADC_CTL0 register, can be monitored by the analog watchdog 0.

Analog watchdog 1/2

The analog watchdog 1/2 are more flexible, and can configure the watchdog function of single or several channels.

The analog watchdog 1 function can be enabled by configuring the corresponding bits in the AWD1CS bits in the ADC_WD1SR register. Similarly, the watchdog 2 function can be configured. The high / low threshold of the analog watchdog 1/2 can be configured in the ADC_WD1HT / ADC_WD1LT and ADC_WD2HT / ADC_WD2LT registers.

ADC_WDx_OUT signal output

Each analog watchdog generates a corresponding ADC_WDx_OUT (x=0,1,2) signal, which is connected to the TIMER module. Within the TIMER module, either the ADC_WDx_OUT signal or other signals can be selected as the ETI input source (refer to the ***Timer (TIMERx)*** module).

When the conversion result of the monitored channel exceeds the threshold, ADC_WDx_OUT signal is set to 1. Even if the WDxE flag bit is cleared by software,it will not affect the ADC_WDx_OUT signal status. When the conversion result remains within the threshold range, the ADC_WDx_OUT signal is reset to 0.

11.4.6. Data storage mode

The alignment of data stored after conversion can be specified by DAL bit in the ADC_CTL1 register.

Figure 11-7. Data storage mode of 12-bit resolution

Routine channel data

0	0	0	0	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0

DAL=0

D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	0	0	0	0

DAL=1

Figure 11-8. Data storage mode of 10-bit resolution

Routine channel data

0	0	0	0	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	0	0

DAL=0

D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	0	0	0	0	0	0

DAL=1

Figure 11-9. Data storage mode of 8-bit resolution

Routine channel data

0	0	0	0	D7	D6	D5	D4	D3	D2	D1	D0	0	0	0	0

DAL=0

D7	D6	D5	D4	D3	D2	D1	D0	0	0	0	0	0	0	0	0

DAL=1

Figure 11-10. Data storage mode of 6-bit resolution

Routine channel data

0	0	0	0	0	0	0	0	0	0	D5	D4	D3	D2	D1	D0

DAL=0

0	0	0	0	0	0	0	0	D5	D4	D3	D2	D1	D0	0	0

DAL=1

11.4.7. Sample time configuration

The number of CK_ADC cycles which is used to sample the input voltage can be specified by the SPTn [2:0] bits in the ADC_SAMPT0 and ADC_SAMPT1 registers. Different sample time can be specified for each channel. For 12-bit resolution, the total sampling and conversion time is “sampling time + 12.5” CK_ADC cycles.

For example:

CK_ADC = 24MHz and sample time is 2.5 cycles, the total time is “2.5+12.5” CK_ADC cycles, that means 0.625us.

11.4.8. External trigger configuration

The conversion of routine sequence can be triggered by rising edge of external trigger inputs. The external trigger source of routine sequence is controlled by the ETSRC[2:0] bits in the ADC_CTL1 register.

Table 11-3. External trigger source for routine sequence

ETSRC[2:0]	Trigger Source	Trigger Type
000	TIMER2_CH1	Hardware trigger
001	TIMER0_CH2	
010	TIMER0_CH1	
011	TIMER2_TRGO	
100	TIMER0_CH0	
101	TIMER2_CH0	
110	EXTI_11	
111	SWRCST	Software trigger

11.4.9. DMA request

The DMA request, which is enabled by the DMA bit in ADC_CTL1 register, is used to transfer data of routine sequence for conversion of more than one channel. The ADC generates a DMA request at the end of conversion of a routine channel. When this request is received, the DMA will transfer the converted data from the ADC_RDATA register to the destination which is specified by the user.

11.4.10. ADC internal channels

ADC analog inputs channel 13, channel 14 and channel 15 are internally connected to the temperature sensor, V_{REFINT} and V_{REFP} analog inputs.

When the TSVEN bit in ADC_CTL1 register is set, the temperature sensor channel is enabled. The temperature sensor can be used to measure the ambient temperature of the device. The sensor output voltage can be converted into a digital value by ADC. The sampling time for the temperature sensor is recommended to be set to at least t_{s_temp} (please refer to the datasheet). When this sensor is not in use, it can enter in power down mode by resetting the TSVEN bit.

The output voltage of the temperature sensor changes linearly with temperature. Because there is an offset, which is up to 45°C and varies from chip to chip due to the chip production process variation, the internal temperature sensor is more appropriate to detect temperature variations instead of absolute temperature. When it is used to detect accurate temperature, an external temperature sensor part should be used to calibrate the offset error.

To use the temperature sensor:

1. Configure the conversion sequence and the sampling time (t_{s_temp}) for the channel.
2. Enable the temperature sensor by setting the TSVEN bit in the ADC control register 1 (ADC_CTL1).
3. Start the ADC conversion by setting the ADCON bit or by the triggers.
4. Read the internal temperature sensor output voltage($V_{temperature}$), and get the temperature with the following equation:

$$\text{Temperature (°C)}= \frac{V_{25}\text{-}V_{temperature}}{\text{Avg\_Slope}}+25 \qquad (11\text{-}1)$$

$V_{temperature}$: The output voltage of temperature sensor.
V_{25}: Internal temperature sensor output voltage at 25°C, the typical value please refer to the datasheet.
Avg_Slope: Average slope for curve between temperature vs. internal temperature sensor output voltage, the typical value refer to the datasheet.

When the INREFEN bit in ADC_CTL1 register is set, the V_{REFINT} channel is enabled. The internal reference voltage (V_{REFINT}) provides a stable (bandgap) voltage output for the ADC and comparators.

The V_{REFP} channel is connect to the positive reference voltage pin VREFP. In particular, VREFP pin are not available and the V_{REFP} channel is internally connected to VDDA in some packages.

11.4.11. Programmable resolution (DRES)

The resolution is configured by programming the DRES[1:0] bits in the ADC_CTL0 register. For applications that do not require high data accuracy, lower resolution allows faster conversion time. The DRES[1:0] bits must only be changed when the ADCON bit is reset. The result of the conversion is always 12 bits wide and any unused LSB bits are read as zeroes. Lower resolution reduces the conversion time needed for the successive approximation steps as shown in *Table 11-4. tCONV timings depending on resolution*.

Table 11-4. t_{CONV} timings depending on resolution

DRES[1:0] bits	t_{CONV}(ADC clock cycles)	t_{CONV}(ns) at f_{ADC}=24MHz	t_{SMPL}(min) (ADC clock cycles)	t_{ADC}(ADC clock cycles)	t_{ADC}(ns) at f_{ADC}=24MHz
12	12.5	521ns	2.5	15	625ns
10	10.5	438ns	2.5	13	542ns
8	8.5	354ns	2.5	11	458ns
6	6.5	271ns	2.5	9	375ns

11.4.12. On-chip hardware oversampling

The on-chip hardware oversampling circuit performs data preprocessing to offload the CPU. It can handle multiple conversions and average them into a single data with increased data width, up to 16-bit.

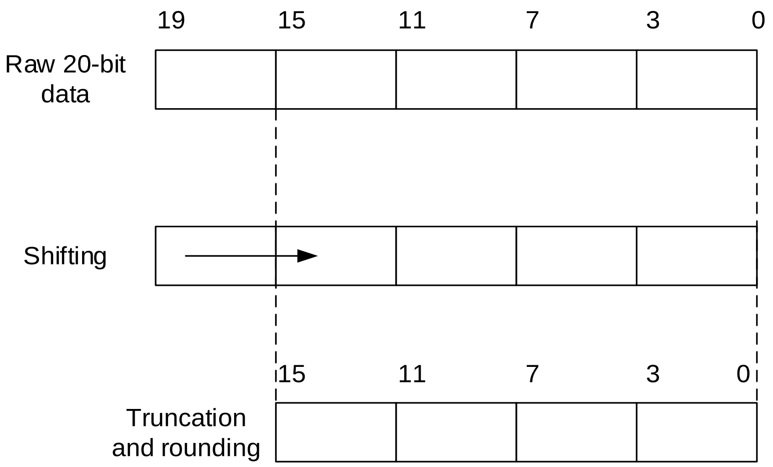
It provides a result with the following form, where N and M can be adjusted, and $D_{out}(n)$ is the n-th output digital signal of the ADC:

$$\text{Result}=\frac{1}{M}*\sum_{n=0}^{N-1}D_{out}(n) \qquad (11\text{-}2)$$

The on-chip hardware oversampling circuit performs the following functions: summing and bit right shifting.The oversampling ratio N is defined by the OVSR[2:0] bits in the ADC_OVSAMPCTL register. It can range from 2x to 256x. The division coefficient M means bit right shifting up to 8-bit. It is configured through the OVSS[3:0] bits in the ADC_OVSAMPCTL register.

Summation units can produce up to 20 bits (256 x 12-bit), which is first shifted right. The upper bits of the result are then truncated, keeping only the 16 least significant bits rounded to the nearest value using the least significant bits left apart by the shifting, before being finally transferred into the data register.

Figure 11-11. 20-bit to 16-bit result truncation



Note: If the intermediate result after the shifting exceeds 16 bits, the upper bits of the result are simply truncated.

Figure 11-12. A numerical example with 5-bit shifting and rounding shows a numerical example of the processing, from a raw 20-bit accumulated data to the final 16-bit result.

Figure 11-12. A numerical example with 5-bit shifting and rounding

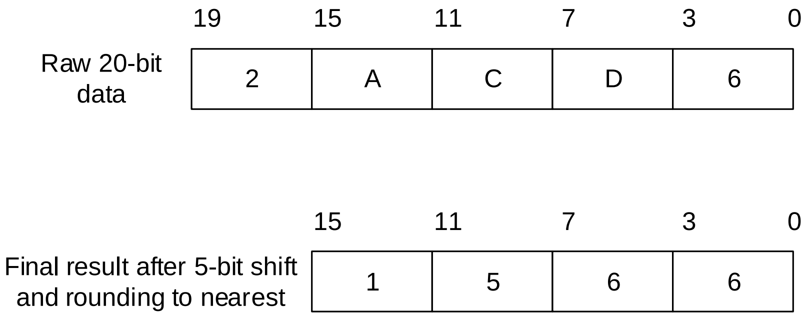


Table 11-5. Maximum output results for N and M combimations (grayed values indicates truncation) below gives the data format for the various N and M combinations, and the raw conversion data equals 0xFFF.

Table 11-5. Maximum output results for N and M combimations (grayed values indicates truncation)

Oversampling ratio	Max Raw data	No-shift OVSS= 0000	1-bit shift OVSS= 0001	2-bit shift OVSS= 0010	3-bit shift OVSS= 0011	4-bit shift OVSS= 0100	5-bit shift OVSS= 0101	6-bit shift OVSS= 0110	7-bit shift OVSS= 0111	8-bit shift OVSS= 1000
2x	0x1FFE	0x1FFE	0x0FFF	0x07FF	0x03FF	0x01FF	0x00FF	0x007F	0x003F	0x001F
4x	0x3FFC	0x3FFC	0x1FFE	0x0FFF	0x07FF	0x03FF	0x01FF	0x00FF	0x007F	0x003F
8x	0x7FF8	0x7FF8	0x3FFC	0x1FFE	0x0FFF	0x07FF	0x03FF	0x01FF	0x00FF	0x007F

Oversampling ratio	Max Raw data	No-shift OVSS= 0000	1-bit shift OVSS= 0001	2-bit shift OVSS= 0010	3-bit shift OVSS= 0011	4-bit shift OVSS= 0100	5-bit shift OVSS= 0101	6-bit shift OVSS= 0110	7-bit shift OVSS= 0111	8-bit shift OVSS= 1000
16x	0xFFF0	0xFFF0	0x7FF8	0x3FFC	0x1FFE	0x0FFF	0x07FF	0x03FF	0x01FF	0x00FF
32x	0x1FFE0	0xFFE0	0xFFF0	0x7FF8	0x3FFC	0x1FFE	0x0FFF	0x07FF	0x03FF	0x01FF
64x	0x3FFC0	0xFFC0	0xFFE0	0xFFF0	0x7FF8	0x3FFC	0x1FFE	0x0FFF	0x07FF	0x03FF
128x	0x7FF80	0xFF80	0xFFC0	0xFFE0	0xFFF0	0x7FF8	0x3FFC	0x1FFE	0x0FFF	0x07FF
256x	0xFFF00	0xFF00	0xFF80	0xFFC0	0xFFE0	0xFFF0	0x7FF8	0x3FFC	0x1FFE	0x0FFF

When compared to standard conversion mode, the conversion timings of oversampling mode do not change, and the sampling time is maintained the same as that of standard conversion mode during the whole oversampling sequence. New data are provided every N conversion, and the equivalent delay is equal to:

$$N \times t_{ADC} = N \times (t_{SMPL} + \mathbf{t}_{CONV}) \qquad (11\text{-}3)$$

11.5. ADC interrupts

The interrupt can be produced on one of the events:

- End of conversion for routine sequence.
- The analog watchdog event.

Separate interrupt enable bits are available for flexibility.

11.6. Register definition

ADC base address: 0x4001 2400

11.6.1. Status register (ADC_STAT)

Address offset: 0x00

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
WD2E	WD1E	Reserved													
rc_w0	rc_w0														

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved											STRC	Reserved		EOC	WD0E
											rc_w0			rc_w0	rc_w0

Bits	Fields	Descriptions
31	WD2E	Analog watchdog 2 event flag 0: No analog watchdog 2 event 1: Analog watchdog 2 event occurred Set by hardware when the converted voltage crosses the values programmed in the ADC_WD2HT and ADC_WD2LT register. Cleared by software writing 0 to it.
30	WD1E	Analog watchdog 1 event flag 0: No analog watchdog 1 event 1: Analog watchdog 1 event occurred Set by hardware when the converted voltage crosses the values programmed in the ADC_WD1HT and ADC_WD1LT register. Cleared by software writing 0 to it.
29:5	Reserved	Must be kept at reset value.
4	STRC	Start flag of routine sequence 0: Routine sequence conversion is not started 1: Routine sequence conversion is started Set by hardware when routine sequence conversion starts. Cleared by software writing 0 to it.
3:2	Reserved	Must be kept at reset value.
1	EOC	End flag of sequence conversion 0: No end of sequence conversion 1: End of sequence conversion Set by hardware at the end of a sequence conversion. Cleared by software writing 0 to it or by reading the ADC_RDATA register.
0	WD0E	Analog watchdog 0 event flag

0: No analog watchdog 0 event

1: Analog watchdog 0 event occurred

Set by hardware when the converted voltage crosses the values programmed in the ADC_WD0LT and ADC_WD0HT registers.

Cleared by software writing 0 to it.

11.6.2. Control register 0 (ADC_CTL0)

Address offset: 0x04

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td>WD2EIE</td><td>WD1EIE</td><td colspan="4">Reserved</td><td colspan="2">DRES [1:0]</td><td>RWD0EN</td><td colspan="7">Reserved</td></tr>
<tr><td>rw</td><td>rw</td><td colspan="4"></td><td colspan="2">rw</td><td>rw</td><td colspan="7"></td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="3">DISNUM [2:0]</td><td>Reserved</td><td>DISRC</td><td>Reserved</td><td>WD0SC</td><td>SM</td><td>Reserved</td><td>WD0EIE</td><td>EOCIE</td><td>Reserved</td><td colspan="4">WD0CHSEL[3:0]</td></tr>
<tr><td colspan="3">rw</td><td></td><td>rw</td><td></td><td>rw</td><td>rw</td><td></td><td>rw</td><td>rw</td><td></td><td colspan="4">rw</td></tr>
</table>

Bits	Fields	Descriptions
31	WD2EIE	Interrupt enable for WD2E 0: WD2E interrupt disable 1: WD2E interrupt enable
30	WD1EIE	Interrupt enable for WD1E 0: WD1E interrupt disable 1: WD1E interrupt enable
29:26	Reserved	Must be kept at reset value.
25:24	DRES[1:0]	ADC resolution 00: 12-bit 01: 10-bit 10: 8-bit 11: 6-bit
23	RWD0EN	Routine channel analog watchdog 0 enable 0: Routine channel analog watchdog 0 disable 1: Routine channel analog watchdog 0 enable
22:16	Reserved	Must be kept at reset value.
15:13	DISNUM[2:0]	Number of conversions in discontinuous mode The number of channels to be converted after a trigger will be DISNUM[2:0]+1 in routine sequence.
12	Reserved	Must be kept at reset value.

11	DISRC	Discontinuous mode on routine sequence 0: Discontinuous operation mode on routine sequence disable 1: Discontinuous operation mode on routine sequence enable
10	Reserved	Must be kept at reset value.
9	WD0SC	When in scan mode, analog watchdog 0 is effective on a single channel. 0: All channels have analog watchdog 0 function 1: A single channel has analog watchdog 0 function
8	SM	Scan mode 0: Scan operation mode disable 1: Scan operation mode enable
7	Reserved	Must be kept at reset value.
6	WD0EIE	Interrupt enable for WD0E 0: WD0E interrupt disable 1: WD0E interrupt enable
5	EOCIE	Interrupt enable for EOC 0: EOC interrupt disable 1: EOC interrupt enable
4	Reserved	Must be kept at reset value.
3:0	WD0CHSEL[3:0]	Analog watchdog 0 channel select 0000: ADC channel 0 0001: ADC channel 1 0010: ADC channel 2 0011: ADC channel 3 0100: ADC channel 4 0101: ADC channel 5 0110: ADC channel 6 0111: ADC channel 7 1000: ADC channel 8 1001: ADC channel 9 1010: ADC channel 10 1011: ADC channel 11 1100: ADC channel 12 1101: ADC channel 13 1110: ADC channel 14 1111: ADC channel 15 **Note:** 1. ADC analog inputs channel 13, channel 14 and channel 15 are internally connected to the temperature sensor, VREFINT and VREFP analog inputs.

11.6.3. Control register 1 (ADC_CTL1)

Address offset: 0x08

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved							INREFEN	TSVEN	SWRCST	Reserved	ETERC	ETSRC [2:0]			Reserved
							rw	rw	rw		rw	rw			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				DAL	Reserved		DMA	Reserved						CTN	ADCON
				rw			rw							rw	rw

Bits	Fields	Descriptions
31:25	Reserved	Must be kept at reset value.
24	INREFEN	Internal reference voltage channel (V_{REFINT}) enable 0: Internal reference voltage channel disable 1: Internal reference voltage channel enable V_{REFINT} analog input is internally connected to the ADC analog inputs channel 14.
23	TSVEN	Temperature sensor output voltage channel enable 0: Temperature sensor output voltage channel disable 1: Temperature sensor output voltage channel enable Temperature sensor output voltage is internally connected to the ADC analog inputs channel 13.
22	SWRCST	Software start of routine sequence Setting 1 on this bit starts the conversion of routine sequence if ETSRC is 111. It is set by software and cleared by software or by hardware immediately after the conversion starts.
21	Reserved	Must be kept at reset value.
20	ETERC	External trigger enable for routine sequence 0: External trigger for routine sequence disable 1: External trigger for routine sequence enable
19:17	ETSRC[2:0]	External trigger selection for routine sequence 000: TIMER2 CH1 001: TIMER0 CH2 010: TIMER0 CH1 011: TIMER2 TRGO 100: TIMER0 CH0 101: TIMER2 CH0 110: EXTI line 11 111: SWRCST

16:12	Reserved	Must be kept at reset value.
11	DAL	Data alignment 0: LSB alignment 1: MSB alignment
10:9	Reserved	Must be kept at reset value.
8	DMA	DMA request enable 0: DMA request disable 1: DMA request enable
7:2	Reserved	Must be kept at reset value.
1	CTN	Continuous mode 0: Continuous operation mode disable 1: Continuous operation mode enable
0	ADCON	ADC ON The ADC will be wake up when this bit is changed from low to high and take a stabilization time ($t_{ST(ADC)}$). When this bit is high and “1” is written to it with other bits of this register unchanged, the conversion will start. For power saving, when this bit is reset, the analog submodule will enter power down mode. 0: ADC disable and power down 1: ADC enable

11.6.4. Sample time register 0 (ADC_SAMPT0)

Address offset: 0x0C

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved														SPT15[2:0]	
														rw	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SPT15[0]	SPT14[2:0]			SPT13[2:0]			SPT12[2:0]			SPT11[2:0]			SPT10[2:0]		
rw		rw			rw			rw			rw			rw	

Bits	Fields	Descriptions
31:18	Reserved	Must be kept at reset value.
17:15	SPT15[2:0]	Refer to SPT10[2:0] description.
14:12	SPT14[2:0]	Refer to SPT10[2:0] description.

11:9	SPT13[2:0]	Refer to SPT10[2:0] description.
8:6	SPT12[2:0]	Refer to SPT10[2:0] description.
5:3	SPT11[2:0]	Refer to SPT10[2:0] description.
2:0	SPT10[2:0]	Channel sample time 000: Channel sampling time is 2.5 cycles 001: Channel sampling time is 3.5 cycles 010: Channel sampling time is 7.5 cycles 011: Channel sampling time is 12.5 cycles 100: Channel sampling time is 19.5 cycles 101: Channel sampling time is 39.5 cycles 110: Channel sampling time is 79.5 cycles 111: Channel sampling time is 160.5 cycles

11.6.5. Sample time register 1 (ADC_SAMPT1)

Address offset: 0x10

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved		SPT9[2:0]			SPT8[2:0]			SPT7[2:0]			SPT6[2:0]			SPT5[2:1]	
		rw			rw			rw			rw			rw	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SPT5[0]	SPT4[2:0]			SPT3[2:0]			SPT2[2:0]			SPT1[2:0]			SPT0[2:0]		
rw	rw			rw			rw			rw			rw		

Bits	Fields	Descriptions
31:30	Reserved	Must be kept at reset value.
29:27	SPT9[2:0]	Refer to SPT0[2:0] description
26:24	SPT8[2:0]	Refer to SPT0[2:0] description
23:21	SPT7[2:0]	Refer to SPT0[2:0] description
20:18	SPT6[2:0]	Refer to SPT0[2:0] description
17:15	SPT5[2:0]	Refer to SPT0[2:0] description
14:12	SPT4[2:0]	Refer to SPT0[2:0] description
11:9	SPT3[2:0]	Refer to SPT0[2:0] description
8:6	SPT2[2:0]	Refer to SPT0[2:0] description
5:3	SPT1[2:0]	Refer to SPT0[2:0] description
2:0	SPT0[2:0]	Channel sample time

000: Channel sampling time is 2.5 cycles

001: Channel sampling time is 3.5 cycles

010: Channel sampling time is 7.5 cycles

011: Channel sampling time is 12.5 cycles

100: Channel sampling time is 19.5 cycles

101: Channel sampling time is 39.5 cycles

110: Channel sampling time is 79.5 cycles

111: Channel sampling time is 160.5 cycles

11.6.6. Watchdog 0 high threshold register (ADC_WD0HT)

Address offset: 0x24

Reset value: 0x0000 0FFF

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				WD0HT[11:0]											
				rw											

Bits	Fields	Descriptions
31:12	Reserved	Must be kept at reset value.
11:0	WD0HT[11:0]	High threshold for analog watchdog 0 These bits define the high threshold for the analog watchdog 0.

11.6.7. Watchdog 0 low threshold register (ADC_WD0LT)

Address offset: 0x28

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				WD0LT[11:0]											
				rw											

Bits	Fields	Descriptions
31:12	Reserved	Must be kept at reset value.
11:0	WD0LT[11:0]	Low threshold for analog watchdog 0

These bits define the low threshold for the analog watchdog 0.

11.6.8. Routine sequence register 0 (ADC_RSQ0)

Address offset: 0x2C

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved								RL[3:0]				Reserved	RSQ15[3:1]		
								rw					rw		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSQ15[0]	Reserved	RSQ14[3:0]				Reserved	RSQ13[3:0]				Reserved	RSQ12[3:0]			
rw		rw					rw					rw			

Bits	Fields	Descriptions
31:24	Reserved	Must be kept at reset value.
23:20	RL[3:0]	Routine sequence length The total number of conversion in routine sequence equals to RL[3:0]+1.
19	Reserved	Must be kept at reset value.
18:15	RSQ15[3:0]	Refer to RSQ0[3:0] description
14	Reserved	Must be kept at reset value.
13:10	RSQ14[3:0]	Refer to RSQ0[3:0] description
9	Reserved	Must be kept at reset value.
8:5	RSQ13[3:0]	Refer to RSQ0[3:0] description
4	Reserved	Must be kept at reset value.
3:0	RSQ12[3:0]	Refer to RSQ0[3:0] description

11.6.9. Routine sequence register 1 (ADC_RSQ1)

Address offset: 0x30

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved			RSQ11[3:0]				Reserved	RSQ10[3:0]				Reserved	RSQ9[3:1]		
			rw					rw					rw		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSQ9[0]	Reserved	RSQ8[3:0]				Reserved	RSQ7[3:0]				Reserved	RSQ6[3:0]			
rw		rw					rw					rw			

Bits	Fields	Descriptions
31:29	Reserved	Must be kept at reset value.
28:25	RSQ11[3:0]	Refer to RSQ0[3:0] description
24	Reserved	Must be kept at reset value.
23:20	RSQ10[3:0]	Refer to RSQ0[3:0] description
19	Reserved	Must be kept at reset value.
18:15	RSQ9[3:0]	Refer to RSQ0[3:0] description
14	Reserved	Must be kept at reset value.
13:10	RSQ8[3:0]	Refer to RSQ0[3:0] description
9	Reserved	Must be kept at reset value.
8:5	RSQ7[3:0]	Refer to RSQ0[3:0] description
4	Reserved	Must be kept at reset value.
3:0	RSQ6[3:0]	Refer to RSQ0[3:0] description

11.6.10. Routine sequence register 2 (ADC_RSQ2)

Address offset: 0x34

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="3">Reserved</td><td colspan="4">RSQ5[3:0]</td><td>Reserved</td><td colspan="4">RSQ4[3:0]</td><td>Reserved</td><td colspan="3">RSQ3[3:1]</td></tr>
<tr><td colspan="3"></td><td colspan="4">rw</td><td></td><td colspan="4">rw</td><td></td><td colspan="3">rw</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>RSQ3[0]</td><td>Reserved</td><td colspan="4">RSQ2[3:0]</td><td>Reserved</td><td colspan="4">RSQ1[3:0]</td><td>Reserved</td><td colspan="4">RSQ0[3:0]</td></tr>
<tr><td>rw</td><td></td><td colspan="4">rw</td><td></td><td colspan="4">rw</td><td></td><td colspan="4">rw</td></tr>
</table>

Bits	Fields	Descriptions
31:30	Reserved	Must be kept at reset value.
29:25	RSQ5[3:0]	Refer to RSQ0[3:0] description
24	Reserved	Must be kept at reset value.
23:20	RSQ4[3:0]	Refer to RSQ0[3:0] description
19	Reserved	Must be kept at reset value.
18:15	RSQ3[3:0]	Refer to RSQ0[3:0] description

14	Reserved	Must be kept at reset value.
13:10	RSQ2[3:0]	Refer to RSQ0[3:0] description
9	Reserved	Must be kept at reset value.
8:5	RSQ1[3:0]	Refer to RSQ0[3:0] description
4	Reserved	Must be kept at reset value.
3:0	RSQ0[3:0]	The channel number is written to these bits to select a channel as the nth conversion in the routine sequence.(The channel number is 0…15)

11.6.11. Routine data register (ADC_RDATA)

Address offset: 0x4C

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RDATA [15:0]															
r															

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	RDATA[15:0]	Routine channel data These bits contain routine channel conversion value, which is read only.

11.6.12. Watchdog 1 channel selection register (ADC_WD1SR)

Address offset: 0x50

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AWD1CS[15:0]															
rw															

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.

15:0	AWD1CS[15:0]	Analog watchdog 1 channel selection These bits are set and cleared by software. They enable and select the input channels to be guarded by the analog watchdog 1. AWD1CS[n] = 0: ADC analog input channel n is not monitored by analog watchdog 1. AWD1CS[n] = 1: ADC analog input channel n is monitored by analog watchdog 1. When AWD1CS[15:0] = 000..0, the analog watchdog 1 is disabled **Note:** 1) The channels selected by AWD1CS must be also selected into the ADC_RSQn registers. 2) ADC analog inputs channel13, channel14 and channel15 are internally connected to the temperature sensor, V_{REFINT} and V_{REFP} analog inputs.

11.6.13. Watchdog 2 channel selection register (ADC_WD2SR)

Address offset: 0x54

Reset value: 0x00000000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AWD2CS[15:0]															
rw															

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	AWD2CS[15:0]	Analog watchdog 2 channel selection These bits are set and cleared by software. They enable and select the input channels to be guarded by the analog watchdog 2. AWD2CS[n] = 0: ADC analog input channel n is not monitored by analog watchdog 2. AWD2CS[n] = 1: ADC analog input channel n is monitored by analog watchdog 2. When AWD2CS[15:0] = 000..0, the analog watchdog 2 is disabled **Note:** 1) The channels selected by AWD2CS must be also selected into the ADC_RSQn registers. 2) ADC analog inputs channel13, channel14 and channel15 are internally connected to the temperature sensor, V_{REFINT} and V_{REFP} analog inputs.

11.6.14. Watchdog 1 high threshold register1 (ADC_WD1HT)

Address offset: 0x58

Reset value: 0x0000 0FFF

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				WD1HT[11:0]											
				rw											

Bits	Fields	Descriptions
31:12	Reserved	Must be kept at reset value.
11:0	WD1HT[11:0]	High threshold for analog watchdog 1 These bits define the high threshold for the analog watchdog 1.

11.6.15. Watchdog 1 low threshold register1 (ADC_WD1LT)

Address offset: 0x5C

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				WD1LT[11:0]											
				rw											

Bits	Fields	Descriptions
31:12	Reserved	Must be kept at reset value.
11:0	WD1LT[11:0]	Low threshold for analog watchdog 1 These bits define the low threshold for the analog watchdog 1.

11.6.16. Watchdog 2 high threshold register2 (ADC_WD2HT)

Address offset: 0x60

Reset value: 0x0000 0FFF

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				WD2HT[11:0]											

rw

Bits	Fields	Descriptions
31:12	Reserved	Must be kept at reset value.
11:0	WD2HT[11:0]	High threshold for analog watchdog 2 These bits define the high threshold for the analog watchdog 2.

11.6.17. Watchdog 2 low threshold register2 (ADC_WD2LT)

Address offset: 0x64

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				WD2LT[11:0]											
				rw											

Bits	Fields	Descriptions
31:12	Reserved	Must be kept at reset value.
11:0	WD2LT[11:0]	Low threshold for analog watchdog 2 These bits define the low threshold for the analog watchdog 2.

11.6.18. Oversample control register (ADC_OVSAMPCTL)

Address offset: 0x80

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved						TOVS	OVSS[3:0]				OVSR[2:0]			Reserved	OVSEN
						rw	rw				rw				rw

Bits	Fields	Descriptions
31:10	Reserved	Must be kept at reset value.
9	TOVS	Triggered oversampling This bit is set and cleared by software. 0: All oversampled conversions for a channel are done consecutively after a trigger

		1: Each conversion needs a trigger for a oversampled channel and the number of triggers is determined by the oversampling ratio(OVSR[2:0]). **Note:** The software allows this bit to be written only when ADCON = 0 (this ensures that no conversion is in progress).
8:5	OVSS [3:0]	Oversampling shift These bits are set and cleared by software. 0000: No shift 0001: Shift 1-bit 0010: Shift 2-bit 0011: Shift 3-bit 0100: Shift 4-bit 0101: Shift 5-bit 0110: Shift 6-bit 0111: Shift 7-bit 1000: Shift 8-bit Other values are reserved. **Note:** The software allows this bit to be written only when ADCON = 0 (this ensures that no conversion is in progress).
4:2	OVSR[2:0]	Oversampling ratio This bit filed defines the number of oversampling ratio. 000: 2x 001: 4x 010: 8x 011: 16x 100: 32x 101: 64x 110: 128x 111: 256x **Note:** The software allows this bit to be written only when ADCON = 0 (this ensures that no conversion is in progress).
1	Reserved	Must be kept at reset value.
0	OVSEN	Oversampler enable This bit is set and cleared by software. 0: Oversampler disabled 1: Oversampler enabled **Note:** The software allows this bit to be written only when ADCON = 0 (this ensures that no conversion is in progress).

12. Watchdog timer (WDGT)

The watchdog timer (WDGT) is a hardware timing circuitry that can be used to detect system failures due to software malfunctions. There are two watchdog timer peripherals in the chip: free watchdog timer (FWDGT) and window watchdog timer (WWDGT). They offer a combination of a high safety level, flexibility of use and timing accuracy. Both watchdog timers are offered to resolve malfunctions of software.

The watchdog timer will generate a reset when the internal counter reaches a given value. The watchdog timer counter can be stopped while the processor is in the debug mode.

12.1. Free watchdog timer (FWDGT)

12.1.1. Overview

The Free watchdog timer (FWDGT) has free clock source (IRC32K). Thereupon the FWDGT can operate even if the main clock fails. It’s suitable for the situation that requires an independent environment and lower timing accuracy.

The Free watchdog timer causes a reset when the internal down counter reaches 0 or the counter is refreshed when the value of the counter is greater than the window register value. The register write protection function in free watchdog can be enabled to prevent it from changing the configuration unexpectedly.

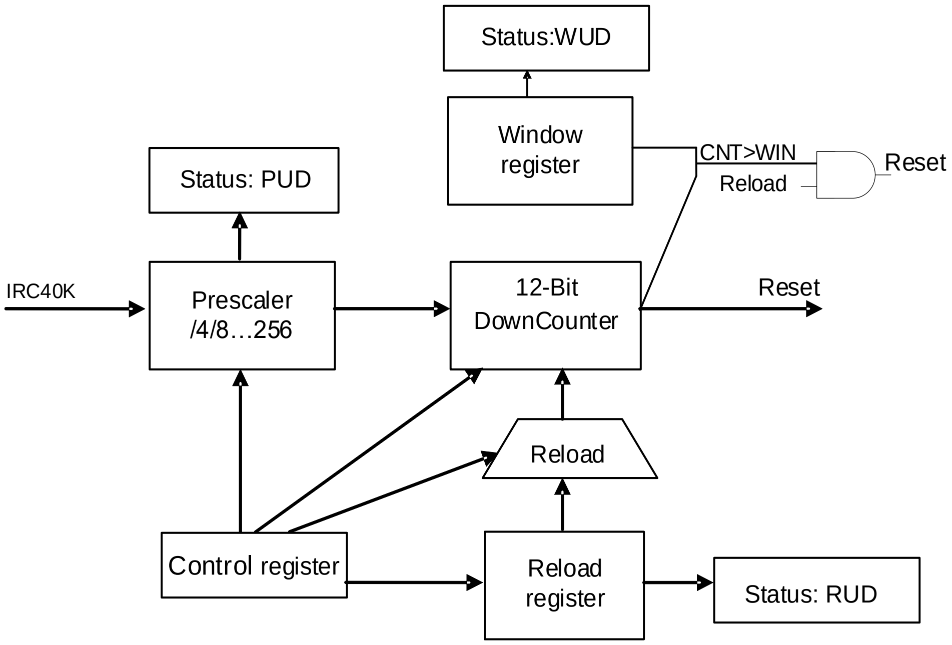
12.1.2. Characteristics

- Free-running 12-bit down counter.
- Generate reset in two conditions when FWDGT is enabled:
  - Reset when the counter reached 0.
  - The counter is refreshed when the value of the counter is greater than the window register value.
- Free clock source, FWDGT can operate even if the main clock fails such as in standby and Deep-sleep modes.
- Hardware free watchdog bit, automatically start the FWDGT or not when power on.
- FWDGT debug mode, the FWDGT can stop or continue to work in debug mode.

12.1.3. Function overview

The free watchdog consists of an 8-stage prescaler and a 12-bit down counter. ***Figure 12-1. Free watchdog block diagram*** shows the functional block of the free watchdog module.

Figure 12-1. Free watchdog block diagram



The free watchdog is enabled by writing the value (0xCCCC) to the control register (FWDGT_CTL), then counter starts counting down. When the counter reaches the value (0x000), there will be a reset.

The counter can be reloaded by writing the value (0xAAAA) to the FWDGT_CTL register at any time. The reload value comes from the FWDGT_RLD register. The software can prevent the watchdog reset by reloading the counter before the counter reaches the value (0x000).

By setting the appropriate window in the FWDGT_WND register, the FWDGT can also work as a window watchdog timer. A reset will occur if the reload operation is performed while the counter is greater than the value stored in the window register (FWDGT_WND). The default value of the FWDGT_WND is 0x0000 0FFF, so if it is not updated, the window option is disabled. A reload operation is performed in order to reset the downcounter to the FWDGT_RLD value and the prescaler counter to generate the next reload, as soon as the window value is changed.

The free watchdog can automatically start at power on when the hardware free watchdog bit in the device option bits is set. To avoid reset, the software should reload the counter before the counter reaches 0x000.

The FWDGT_PSC register, the FWDGT_RLD register and the FWDGT_WND register are write protected. Before writing these registers, the software should write the value (0x5555) to the FWDGT_CTL register. These registers will be protected again by writing any other value to the FWDGT_CTL register. When an update operation of the prescaler register (FWDGT_PSC), window register (FWDGT_WND) or the reload value register (FWDGT_RLD) is ongoing, the status bits in the FWDGT_STAT register are set.

If the FWDGT_HOLD bit in DBG module is cleared, the FWDGT continues to work even the Cortex®-M23 core halted (Debug mode). The FWDGT stops in Debug mode if the

FWDGT_HOLD bit is set.

Table 12-1. Min/max FWDGT timeout period at 32KHz (IRC32K)

Prescaler divider	PSC[2:0] bits	Min timeout (ms) RL[11:0]= 0x000	Max timeout (ms) RL[11:0]= 0xFFF
1/4	000	0.03125	511.90625
1/8	001	0.03125	1023.78125
1/16	010	0.03125	2047.53125
1/32	011	0.03125	4095.03125
1/64	100	0.03125	8190.03125
1/128	101	0.03125	16380.03125
1/256	110 or 111	0.03125	32760.03125

The FWDGT timeout can be more accurately by calibrating the IRC32K.

Note: When after the execution of watchdog reload operation, if the MCU needs enter the deepsleep / standby mode immediately, more than 3 IRC32K clock intervals must be inserted in the middle of reload and deepsleep / standby mode commands by software setting.

12.1.4. Register definition

FWDGT base address: 0x4000 3000

Control register (FWDGT_CTL)

Address offset: 0x00

Reset value: 0x0000 0000

This register can be accessed by half-word(16-bit) or word(32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CMD[15:0]															
w															

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	CMD[15:0]	Write only. Several different functions are realized by writing these bits with different values. 0x5555: Disable the FWDGT_PSC, FWDGT_RLD and FWDGT_WND write protection. 0xCCCC: Start the free watchdog timer counter. When the counter reduces to 0, the free watchdog generates a reset. 0xAAAA: Reload the counter.

Prescaler register (FWDGT_PSC)

Address offset: 0x04

Reset value: 0x0000 0000

This register can be accessed by half-word(16-bit) or word(32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved													PSC[2:0]		
													rw		

Bits	Fields	Descriptions
31:3	Reserved	Must be kept at reset value.
2:0	PSC[2:0]	Free watchdog timer prescaler selection. Write 0x5555 in the FWDGT_CTL register before writing these bits. During a write operation to this register, the PUD bit in the

FWDGT_STAT register is set and the value read from this register is invalid.

000: 1 /4

001: 1 / 8

010: 1 / 16

011: 1 / 32

100: 1 / 64

101: 1 / 128

110: 1 / 256

111: 1 / 256

If several prescaler values are used by the application, it is mandatory to wait until PUD bit has been reset before changing the prescaler value. If the prescaler value has been updated, it is not necessary to wait until PUD has been reset before continuing code execution (Before entering low-power mode, it is necessary to wait until PUD is reset).

Reload register (FWDGT_RLD)

Address offset: 0x08

Reset value: 0x0000 0FFF

This register can be accessed by half-word(16-bit) or word(32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				RLD [11:0]											
				rw											

Bits	Fields	Descriptions
31:12	Reserved	Must be kept at reset value.
11:0	RLD[11:0]	Free watchdog timer counter reload value. Write 0xAAAA in the FWDGT_CTL register will reload the FWDGT conter with the RLD value. These bits are write-protected. Write 0x5555 to the FWDGT_CTL register before writing these bits. During a write operation to this register, the RUD bit in the FWDGT_STAT register is set and the value read from this register is invalid. If several reload values are used by the application, it is mandatory to wait until RUD bit has been reset before changing the reload value. If the reload value has been updated, it is not necessary to wait until RUD has been reset before continuing code execution (Before entering low-power mode, it is necessary to wait until RUD is reset).

Status register (FWDGT_STAT)

Address offset: 0x0C

Reset value: 0x0000 0000

This register can be accessed by half-word(16-bit) or word(32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved													WUD	RUD	PUD
													r	r	r

Bits	Fields	Descriptions
31:3	Reserved	Must be kept at reset value.
2	WUD	Watchdog counter window value update. When a write operation to FWDGT_WND register ongoing, this bit is set and the value read from FWDGT_WND register is invalid.
1	RUD	Free watchdog timer counter reload value update. During a write operation to FWDGT_RLD register, this bit is set and the value read from FWDGT_RLD register is invalid.
0	PUD	Free watchdog timer prescaler value update. During a write operation to FWDGT_PSC register, this bit is set and the value read from FWDGT_PSC register is invalid.

Window register (FWDGT_WND)

Address offset: 0x10

Reset value: 0x0000 0FFF

This register can be accessed by half-word(16-bit) or word(32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				WND[11:0]											
				rw											

Bits	Fields	Descriptions
31:12	Reserved	Must be kept at reset value.
11:0	WND[11:0]	Watchdog counter window value. These bits are used to contain the high limit of the window value to be compared to the downcounter. A reset will occur if the reload operation is performed while the counter is greater than the value stored in this register. The WUD bit in the FWDGT_STAT register must be reset in order to be able to change the reload value. These bits are write protected. Write 0x5555 in the FWDGT_CTL register before

writing these bits.

If several window values are used by the application, it is mandatory to wait until WUD bit has been reset before changing the window value. However, after updating the window value it is not necessary to wait until WUD is reset before continuing code execution except in case of low-power mode entry(Before entering low-power mode, it is necessary to wait until WUD is reset).

12.2. Window watchdog timer (WWDGT)

12.2.1. Overview

The window watchdog timer (WWDGT) is used to detect system failures due to software malfunctions. After the window watchdog timer starts, the value of down counter reduces progressively. The watchdog timer causes a reset when the counter reached 0x3F (the CNT[6] bit has been cleared). The watchdog timer also causes a reset when the counter is refreshed before the counter reached the window register value. So the software should refresh the counter in a limited window. The window watchdog timer generates an early wakeup status flag when the counter reaches 0x40.Interrupt occurs if it is enable.

The window watchdog timer clock is prescaled from the APB clock. The window watchdog timer is suitable for the situation that requires an accurate timing.

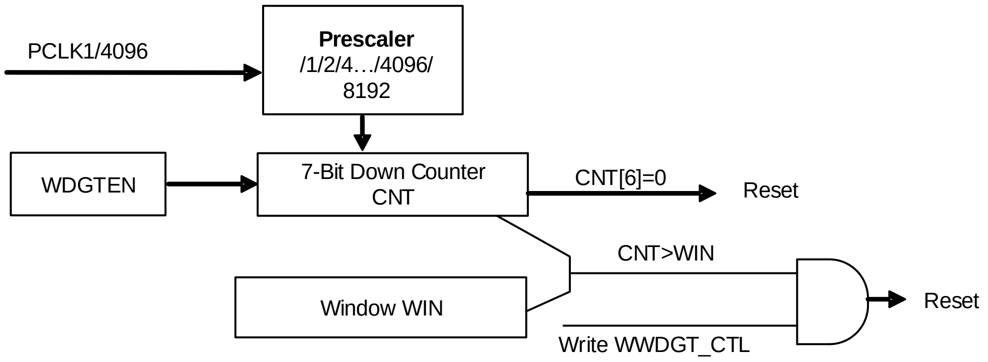
12.2.2. Characteristics

- Programmable free-running 7-bit down counter.
- Generate reset in two conditions when WWDGT is enabled:
  - Reset when the counter reached 0x3F.
  - The counter is refreshed when the value of the counter is greater than the window register value.
- Early wakeup interrupt (EWI): the watchdog is started and the interrupt is enabled, the interrupt occurs when the counter reaches 0x40.
- WWDGT debug mode, the WWDGT can stop or continue to work in debug mode.

12.2.3. Function overview

If the window watchdog timer is enabled (set the WDGTEN bit in the WWDGT_CTL), the watchdog timer cause a reset when the counter reaches 0x3F (the CNT[6] bit has been cleared), or the counter is refreshed before the counter reaches the window register value.

Figure 12-2. Window watchdog timer block diagram



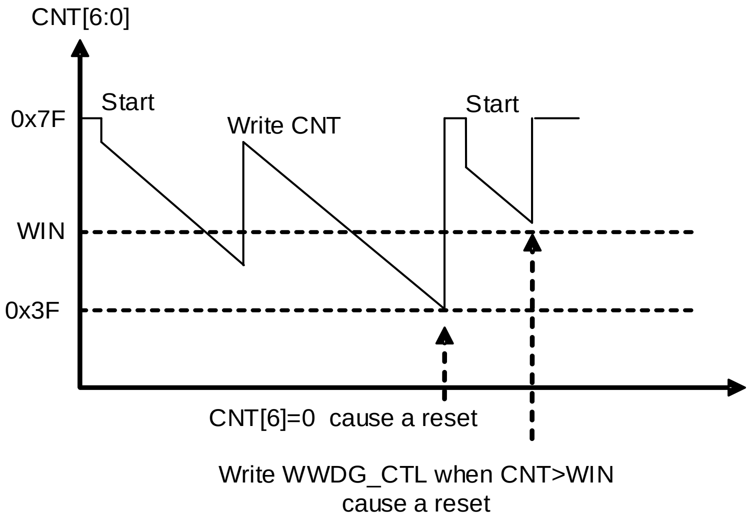
When the software window watchdog is selected, the watchdog timer is always disabled after power on reset. It is enabled by setting the WDGTEN bit in the WWDGT_CTL register, then it cannot be disabled again, except by a reset. When the hardware window watchdog is selected, the watchdog timer is always enabled after a reset, and it cannot be disabled. When window watchdog timer is enabled, the counter counts down all the time, the configured value of the counter should be greater than 0x3F(it implies that the CNT[6] bit should be set). The CNT[5:0] determine the maximum time interval between two reloading. The count down speed depends on the APB clock and the prescaler (PSC[3:0] bits in the WWDGT_CFG register).

The WIN[6:0] bits in the configuration register (WWDGT_CFG) specifies the window value. The software can prevent the reset event by reloading the down counter. The counter value is less than the window value and greater than 0x3F, otherwise the watchdog causes a reset.

The early wakeup interrupt (EWI) is enabled by setting the EWIE bit in the WWDGT_CFG register, and the interrupt will be generated when the counter reaches 0x40 or the counter is refreshed before it reaches the window value. The software can do something such as communication or data logging in the interrupt service routine (ISR) in order to analyse the reason of software malfunctions or save the important data before resetting the device. Moreover the software can reload the counter in ISR to manage a software system check and so on. In this case, the WWDGT will never generate a WWDGT reset but can be used for other things.

The EWI interrupt is cleared by writing '0' to the EWIF bit in the WWDGT_STAT register.

Figure 12-3. Window watchdog timing diagram



Calculate the WWDGT timeout by using the formula below.

$$t_{WWDGT}=t_{PCLK1} \times 4096 \times 2^{PSC} \times (CNT[5:0]+1) \quad (ms) \qquad (12\text{-}1)$$

where:

t_{WWDGT}: WWDGT timeout

t_{PCLK1}: APB clock period measured in ms

The *Table 12-2. Min-max timeout value at 48 MHz (fPCLK1)* shows the minimum and maximum values of the t_{WWDGT}.

Table 12-2. Min-max timeout value at 48 MHz (f_{PCLK1})

Prescaler divider	PSC[3:0]	Min timeout value CNT[6:0] =0x40	Max timeout value CNT[6:0]=0x7F
1 / 1	0000	85.33µs	5.461ms
1 / 2	0001	170.67µs	10.923ms
1 / 4	0010	341.33µs	21.845ms
1 / 8	0011	682.67µs	43.691ms
1 / 16	0100	1.365ms	87.382ms
1 / 32	0101	2.731ms	174.764ms
1 / 64	0110	5.461ms	349.528ms
1 / 128	0111	10.922ms	699.056ms
1 / 256	1000	21.854ms	1398.112ms
1 / 512	1001	43.691ms	2796.224ms
1 / 1024	1010	87.381ms	5592.448ms
1 / 2048	1011	174.763ms	11184.896ms
1 / 4096	1100	349.525ms	22369.792ms
1 / 8192	1101	699.051ms	44739.584ms
1 / 1	1110	85.33µs	5.461ms
1 / 1	1111	85.33µs	5.461ms

If the WWDGT_HOLD bit in DBG module is cleared, the WWDGT continues to work even the Cortex®-M23 core halted (Debug mode). While the WWDGT_HOLD bit is set, the WWDGT stops in Debug mode.

12.2.4. Register definition

WWDGT base address: 0x4000 2C00

Control register (WWDGT_CTL)

Address offset: 0x00

Reset value: 0x0000 007F

This register can be accessed by half-word(16-bit) or word(32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								WDGTEN	CNT[6:0]						
								rs	rw						

Bits	Fields	Descriptions
31:8	Reserved	Must be kept at reset value.
7	WDGTEN	Start the Window watchdog timer. Cleared by a hardware reset. Writing 0 has no effect. 0: Window watchdog timer disabled. 1: Window watchdog timer enabled.
6:0	CNT[6:0]	The value of the watchdog timer counter. A reset occur when the value of this counter decreases from 0x40 to 0x3F. When the value of this counter is greater than the window value, writing this counter also causes a reset.

Configuration register (WWDGT_CFG)

Address offset: 0x04

Reset value: 0x0000 007F

This register can be accessed by half-word(16-bit) or word(32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved														PSC[3:2]	
														rw	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved						EWIE	PSC[1:0]		WIN[6:0]						
						rs	rw		rw						

Bits	Fields	Descriptions
31:18	Reserved	Must be kept at reset value.
17:16	PSC[3:2]	Prescaler. This bits with PSC[1:0] determines the time base of the watchdog counter.

		0000: (PCLK1 / 4096) / 1 0001: (PCLK1 / 4096) / 2 0010: (PCLK1 / 4096) / 4 0011: (PCLK1 / 4096) / 8 0100: (PCLK1 / 4096) / 16 0101: (PCLK1 / 4096) / 32 0110: (PCLK1 / 4096) / 64 0111: (PCLK1 / 4096) / 128 1000: (PCLK1 / 4096) / 256 1001: (PCLK1 / 4096) / 512 1010: (PCLK1 / 4096) / 1024 1011: (PCLK1 / 4096) / 2048 1100: (PCLK1 / 4096) / 4096 1101: (PCLK1 / 4096) / 8192 1110: (PCLK1 / 4096) / 1 1111: (PCLK1 / 4096) / 1
15:10	Reserved	Must be kept at reset value.
9	EWIE	Early wakeup interrupt enable. If the bit is set, an interrupt occurs when the counter reaches 0x40. It can be cleared by a hardware reset or software reset by setting the WWDGTRST bit of the RCU module. A write operation of 0 has no effect.
8:7	PSC[1:0]	Prescaler. This bits with bit[17:16] determines the time base of the watchdog counter.
6:0	WIN[6:0]	The Window value. A reset occur if the watchdog counter (CNT bits in WWDGT_CTL) is written when the value of the watchdog counter is greater than the Window value.

Status register (WWDGT_STAT)

Address offset: 0x08

Reset value: 0x0000 0000

This register can be accessed by half-word(16-bit) or word(32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															EWIF
															rc_w0

Bits	Fields	Descriptions
31:1	Reserved	Must be kept at reset value.
0	EWIF	Early wakeup interrupt flag. When the counter reaches 0x40, this bit is set by

hardware even the interrupt is not enabled (EWIE in WWDGT_CFG is cleared). This bit is cleared by writing 0. There is no effect when writing 1.

13. Real time clock (RTC)

13.1. Overview

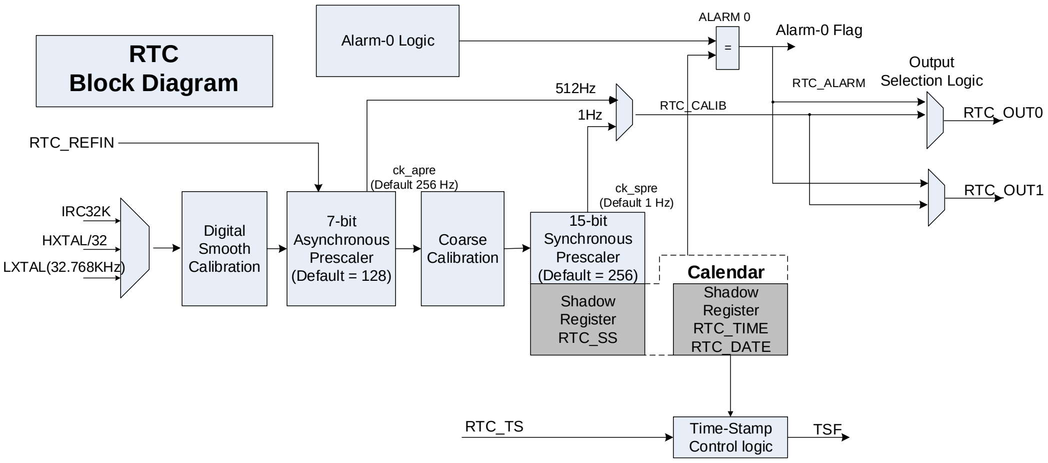
The RTC provides a time which includes hour/minute/second/sub-second and a calendar includes year/month/day/week day. The time and calendar are expressed in BCD code except sub-second. Sub-second is expressed in binary code. Hour adjust for daylight saving time. Working in power saving mode is software configurable. Support improving the calendar accuracy using extern accurate low frequency clock.

13.2. Characteristics

- Daylight saving compensation supported by software
- External high-accurate low frequency(50Hz or 60Hz) clock used to achieve higher calendar accuracy performed by reference clock detection option function
- Atomic clock adjust(max adjust accuracy is 0.95PPM) for calendar calibration performed by digital calibration function
- Sub-second adjustment by shift function
- Time-stamp function for saving event time
- Programmable calendar and one field maskable alarms
- Maskable interrupt source:
  - Alarm 0
  - Time-stamp detection
- Four 16-bit (16 bytes total) universal backup registers which can keep data under power saving mode

13.3. Function overview

Figure 13-1. Block diagram of RTC



The RTC unit includes:

- Alarm event/interrupt
- Optional RTC output function:
  - 512Hz (default prescale) :RTC_OUT0(PC13:package is 48 pin or PA4: package is 20/28/32 pin) or RTC_OUT1(PA4)
  - 1Hz(default prescale): RTC_OUT0(PC13:package is 48 pin or PA4: package is 20/28/32 pin) or RTC_OUT1(PA4)
  - Alarm event(polarity is configurable): RTC_OUT0(PC13:package is 48 pin or PA4: package is 20/28/32 pin) or RTC_OUT1(PA4)
- Optional RTC input function:
  - time stamp event detection: RTC_TS(PC13:package is 48 pin or PA4: package is 20/28/32 pin)
  - reference clock input: RTC_REFIN(PB15 or PB7)

13.3.1. Clock source and prescalers

RTC unit has three independent clock sources: LXTAL, IRC32K and HXTAL with divided by 32(configured in RCU_CTL1 register).

In the RTC unit, there are two prescalers used for implementing the calendar and other functions. One prescaler is a 7-bit asynchronous prescaler and the other is a 15-bit synchronous prescaler. Asynchronous prescaler is mainly used for reducing power consumption. The asynchronous prescaler is recommended to set as high as possible if both prescalers are used.

The frequency formula of two prescalers is shown as below:

$$f_{ck\_apre} = \frac{f_{rtcclk}}{FACTOR\_A + 1} \tag{13-1}$$

$$f_{ck\_spre} = \frac{f_{ck\_apre}}{FACTOR\_S + 1} = \frac{f_{rtcclk}}{(FACTOR\_A + 1)*(FACTOR\_S + 1)} \tag{13-2}$$

The ck_apre clock is used to driven the RTC_SS down counter which stands for the time left to next second in binary format and when it reaches 0 it will automatically reload FACTOR_S value. The ck_spre clock is used to driven the calendar registers. Each clock will make second plus one.

13.3.2. Shadow registers introduction

BPSHAD control bit decides the location when APB bus accesses the RTC calendar register RTC_DATE, RTC_TIME and RTC_SS. By default, the BPSHAD is cleared, and APB bus accesses the shadow calendar registers. Shadow calendar registers is updated with the value of real calendar registers every two RTC clock and at the same time RSYNF bit will be set once. This update mechanism is not performed in Deep-Sleep mode and Standby mode. When exiting these modes, software must clear RSYNF bit and wait it is asserted (the max wait time is 2 RTC clock) before reading calendar register under BPSHAD=0 situation.

Note: When reading calendar registers (RTC_SS, RTC_TIME, RTC_DATE) under BPSHAD=0, the frequency of the APB clock (f_{apb}) must be at least 7 times the frequency of the RTC clock (f_{rtcclk}).

System reset will reset the shadow calendar registers.

13.3.3. Configurable and field maskable alarm

RTC alarm function is divided into some fields and each has a maskable bit.

RTC alarm function can be enabled or disabled by ALRMxEN(x=0) bit in RTC_CTL0. If all the alarm fields value match the corresponding calendar value when ALRMxEN=1(x=0), the Alarm flag will be set.

Note: FACTOR_S in the RTC_PSC register must be larger than 3 if MSKS bit reset in RTC_ALRMxTD(x=0).

If a field is masked, the field is considered as matched in logic. If all the fields have been masked, the Alarm Flag will assert 3 RTC clock later after ALRMxEN(x=0) is set.

13.3.4. RTC initialization and configuration

RTC register write protection

BKPWEN bit in the PMU_CTL register is cleared in default, so writing to RTC registers needs setting BKPWEN bit ahead of time.

After power-on reset, most of RTC registers are write protected. Unlocking this protection is the first step before writing to them.

Following below steps will unlock the write protection:

1. Write '0xCA' into the RTC_WPK register
2. Write '0x53' into the RTC_WPK register

Writing a wrong value to RTC_WPK will make write protection valid again. The state of write protection is not affected by system reset. Following registers are writing protected but others are not:

RTC_TIME, RTC_DATE, RTC_CTL, RTC_STAT, RTC_PSC, RTC_ALRM0TD, RTC_SHIFTCTL, RTC_HRFC, RTC_ALRM0SS

Calendar initialization and configuration

The prescaler and calendar value can be programmed by the following steps:

1. Enter initialization mode (by setting INITM=1) and polling INITF bit until INITF=1.
2. Program both the asynchronous and synchronous prescaler factors in RTC_PSC register.
3. Write the initial calendar values into the shadow calendar registers (RTC_TIME and RTC_DATE), and use the CS bit in the RTC_CTL register to configure the time format (12 or 24 hours).
4. Exit the initialization mode (by setting INITM=0).

About 4 RTC clock cycles later, real calendar registers will load from shadow registers and calendar counter restarts.

Note: Reading calendar register (BPSHAD=0) after initialization, software should confirm the RSYNF bit to 1.

YCM flag indicates whether the calendar has been initialized by checking the year field of calendar.

Daylight saving Time

RTC unit supports daylight saving time adjustment through S1H, A1H and DSM bit.

S1H and A1H can subtract or add 1 hour to the calendar when the calendar is running.S1H and A1H operation can be autologically set and DSM bit can be used to recording this adjust operation. After setting the S1H/A1H, subtract/add 1 hour will perform when next second comes.

Alarm function operation process

To avoid unexpected alarm assertion and metastable state, alarm function has an operation flow:

1. Disable Alarm (by resetting ALRMxEN(x=0) in RTC_CTL)
2. Set the Alarm registers needed(RTC_ALRMxTD/RTC_ALRMxSS(x=0))
3. Enable Alarm function (by setting ALRMxEN(x=0) in the RTC_CTL)

13.3.5. Calendar reading

Reading calendar registers under BPSHAD=0

When BPSHAD=0, calendar value is read from shadow registers. For the existence of synchronization mechanism, a basic request has to meet: the APB bus clock frequency must be equal to or greater than 7 times the RTC clock frequency.APB bus clock frequency lower than RTC clock frequency is not allowed in any case whatever happens. For example: if system clock uses LXTAL,RTC clock can not select HXTAL clock divided by 32, because HXTAL clock divided by 32 fatester than LXTAL.

When APB bus clock frequency is not equal to or greater than 7 times the RTC clock frequency, the calendar reading flow should be obeyed:

1. reading calendar time register and date register twice
2. if the two values are equal, the value can be seen as the correct value
3. if the two values are not equal, a third reading should performed
4. the third value can be seen as the correct value

RSYNF is asserted once every 2 RTC clock and at this time point, the shadow registers will be updated to current time and date.

To ensure consistency of the 3 values (RTC_SS, RTC_TIME, and RTC_DATE), below consistency mechanism is used in hardware:

1. reading RTC_SS will lock the updating of RTC_TIME and RTC_DATE
2. reading RTC_TIME will lock the updating of RTC_DATE
3. reading RTC_DATE will unlock updating of RTC_TIME and RTC_DATE

If the software wants to read calendar in a short time interval(smaller than 2 RTCCLK periods), RSYNF must be cleared by software after the first calendar read, and then the software must wait until RSYNF is set again before next reading.

In below situations, software should wait RSYNF bit asserted before reading calendar registers (RTC_SS, RTC_TIME, and RTC_DATE):

1. after a system reset
2. after an initialization
3. after shift function

Especially that software must clear RSYNF bit and wait it asserted before reading calendar register after wakeup from power saving mode.

Reading calendar registers under BPSHAD=1

When BPSHAD=1, RSYNF is cleared and maintains as 0 by hardware so reading calendar registers does not care about RSYNF bit. Current calendar value is read from real-time calendar counter directly. The benefit of this configuration is that software can get the real current time without any delay after wakeup from power saving mode (Deep-sleep /Standby Mode).

Because of no RSYNF bit periodic assertion, the results of the different calendar registers (RTC_SS/RTC_TIME/RTC_DATE) might not be coherent with each other when clock ck_apre edge occurs between two reading calendar registers.

In addition, if current calendar register is changing and at the same time the APB bus reading calendar register is also performing, the value of the calendar register read out might be not correct.

To ensure the correctness and consistency of the calendar value, software must perform reading operation as this: read all calendar registers continuously, if the last two values are the same, the data is coherent and correct.

13.3.6. Resetting the RTC

There are two reset sources used in RTC unit: system reset and backup registers reset.

System reset will affect calendar shadow registers and some bits of the RTC_STAT. When system reset is valid, the bits or registers mentioned before are reset to the default value.

Backup registers reset will affect the following registers and system reset will not affect them:

- RTC current real-time calendar registers
- RTC Control register (RTC_CTL)

- RTC Prescaler register (RTC_PSC)
- RTC High resolution frequency compensation register (RTC_HRFC)
- RTC Shift control register (RTC_SHIFTCTL)
- RTC Time stamp registers (RTC_SSTS/RTC_TTS/RTC_DTS)
- RTC Alarm registers (RTC_ALRMxSS/RTC_ALRMxTD(x=0))

The RTC unit will go on running when system reset occurs or enter power saving mode, but if backup registers reset occurs, RTC will stop counting and all registers will reset.

13.3.7. RTC shift function

When there is a remote clock with higher degree of precision and RTC 1Hz clock (ck_spre) has an offset (in a fraction of a second) with the remote clock, RTC unit provides a function named shift function to remove this offset and thus make second precision higher.

RTC_SS register indicates the fraction of a second in binary format and is down counting when RTC is running. Therefore by adding the SFS[14:0] value to the synchronous prescaler counter SSC[15:0] or by adding the SFS[14:0] value to the synchronous prescaler counter SSC[15:0] and at the same time set A1S bit can delay or advance the time when next second arrives.

The maximal RTC_SS value depends on the FACTOR_S value in RTC_PSC. The higher FACTOR_S, the higher adjust precision.

Because of the 1Hz clock (ck_spre) is generated by FACTOR_A and FACTOR_S, the higher FACTOR_S means the lower FACTOR_A, then more power consuming.

Note: Before using shift function, the software must check the MSB of SSC in RTC_SS (SSC[15]) and confirm it is 0.

After writing RTC_SHIFTCTL register, the SOPF bit in RTC_STAT will be set at once. When shift operation is complete, SOPF bit is cleared by hardware. System reset does not affect SOPF bit.

Shift operation only works correctly when REFEN=0.

Software must not write to RTC_SHIFTCTL if REFEN=1.

13.3.8. RTC reference clock detection

RTC reference clock detection is another way to increase the precision of RTC second. To enable this function, you should have an external clock source (50Hz or 60 Hz) which is more precise than LXTAL clock source.

After enabling this function (REFEN=1), each 1Hz clock (ck_spre) edge is compared to the nearest RTC_REFIN clock edge. In most cases, the two clock edges are aligned every time. But when two clock edges are misaligned for the reason of LXTAL poor precision, the RTC reference clock detection function will shift the 1Hz clock edge a little to make next 1Hz clock edge aligned to reference clock edge.

When REFEN=1, a time window is applied at every second update time different detection state will use different window period.

7 ck_apre window is used when detecting the first reference clock edge and 3 ck_apre window is used for the edge aligned operation.

Whatever window used, the asynchronous prescaler counter will be forced to reload when the reference clock is detected in the window. When the two clock (ck_spre and reference clock) edges are aligned, this reload operation has no effect for 1Hz clock. But when the two clock edge are not aligned, this reload operation will shift ck_spre clock edge a bit to make the ck_spre(1Hz) clock edge aligned to the reference clock edge.

When reference detection function is running while the external reference clock is removed (no reference clock edge found in 3 ck_apre window), the calendar updating still can be performed by LXTAL clock only. If the reference clock is recovered later, detection function will use 7 ck_apre window to identify the reference clock and use 3 ck_apre window to adjust the 1Hz clock (ck_spre) edge.

Note: Software must configure the FACTOR_A=0x7F and FACTOR_S=0xFF before enabling reference detection function (REFEN=1)

Reference detection function does not work in Standby Mode.

13.3.9. RTC smooth digital calibration

RTC smooth calibration function is a way to calibrate the RTC frequency based on RTC clock in a configurable period time.

This calibration is equally executed in a period time and the cycle number of the RTC clock in the period time will be added or subtracted. The resolution of the calibration is about 0.954PPM with the range from -487.1PPM to +488.5PPM.

The calibration period time can be configured to the $2^{20}/2^{19}/2^{18}$ RTC clock cycles which stands for 32/16/8 seconds if RTC input frequency is 32.768 KHz.

The High resolution frequency compensation register (RTC_HRFC) specifies the number of RTCCLK clock cycles to be calibrated during the period time:

So using CMSK can mask clock cycles from 0 to 511 and thus the RTC frequency can be reduced by up to 487.1PPM.

To increase the RTC frequency the FREQI bit can be set. If FREQI bit is set, there will be 512 additional cycles to be added during period time which means every $2^{11}/2^{10}/2^{9}$ (32/16/8 seconds) RTC clock insert one cycle.

So using FREQI can increase the RTC frequency by 488.5PPM.

The combined using of CMSK and FREQI can adjust the RTC cycles from -511 to +512 cycles in the period time which means the calibration range is -487.1PPM to +488.5PPM with a resolution of about 0.954PPM.

When calibration function is running, the output frequency of calibration is calculated by the following formula:

$$f_{\mathrm{cal}} = f_{\mathrm{rtcclk}} \times (1 + \frac{\mathrm{FREQI} \times 512 - \mathrm{CMSK}}{2^{\mathrm{N}} + \mathrm{CMSK} - \mathrm{FREQI} \times 512}) \qquad (13\text{-}3)$$

Note: N=20/19/18 for 32/16/8 seconds window period

Calibration when FACTOR_A < 3

When asynchronous prescaler value (FACTOR_A) is set to less than 3, software should not set FREQI bit to 1 when using calibration function. FREQI setting will be ignored when FACTOR_A<3.

When the FACTOR_A is less than 3, the FACTOR_S value should be set to a value less than the nominal value. Assuming that RTC clock frequency is nominal 32.768 KHz, the corresponding FACTOR_S should be set as following rule:

FACTOR_A = 1: 4 less than nominal FACTOR_S (16379 with 32.768 KHz)

FACTOR_A = 0: 8 less than nominal FACTOR_S (32759 with 32.768 KHz)

When the FACTOR_A is less than 3, CMSK is 0x100，the formula of calibration frequency is as follows:

$$f_{\mathrm{cal}} = f_{\mathrm{rtcclk}} \times (1 + \frac{256 - \mathrm{CMSK}}{2^{\mathrm{N}} + \mathrm{CMSK} - 256}) \qquad (13\text{-}4)$$

Note: N=20/19/18 for 32/16/8 seconds window period

Verifying the RTC calibration

Calibration 1Hz output is provided to assist software to measure and verify the RTC precision.

Up to 2 RTC clock cycles measurement error may occur when measuring the RTC frequency over a limited measurement period. To eliminate this measurement error the measurement period should be the same as the calibration period.

- When the calibration period is 32 seconds(this is default configuration)

Using exactly 32s period to measure the accuracy of the calibration 1Hz output can guarantee the measure is within 0.477PPM (0.5 RTCCLK cycles over 32s)

- When the calibration period is 16 seconds(by setting CWND16 bit)

In this configuration, CMSK[0] is fixed to 0 by hardware. Using exactly 16s period to measure the accuracy of the calibration 1Hz output can guarantee the measure is within 0.954PPM (0.5 RTCCLK cycles over 16s)

- When the calibration period is 8 seconds(by setting CWND8 bit)

In this configuration, CMSK[1:0] is fixed to 0 by hardware. Using exactly 8s period to measure the accuracy of the calibration 1Hz output can guarantee the measure is within 1.907PPM (0.5 RTCCLK cycles over 8s)

Re-calibration on-the-fly

When the INITF bit is 0, software can update the value of RTC_HRFC using following steps:

1. Wait the SCPF=0
2. Write the new value into RTC_HRFC register
3. After 3 ck_apre clocks, the new calibration settings take effect

13.3.10. Time-stamp function

Time-stamp function is performed on RTC_TS pin and is enabled by control bit TSEN.

When a time-stamp event occurs on RTC_TS pin (TSEN = 1), the calendar value will be saved in time-stamp registers (RTC_DTS/RTC_TTS/RTC_SSTS) and the time-stamp flag (TSF) is set to 1 by hardware.

Time-stamp registers only record the calendar at the first time time-stamp event occurs which means that time-stamp registers will not change when TSF=1.

Note: When the time-stamp event occurs, TSF is set 2 ck_apre cycles delay because of synchronization mechanism.

13.3.11. Calibration clock output

Calibration clock can be output on the RTC_OUT0/1 if COEN bit is set to 1.

When the COS bit is set to 0(this is default) and asynchronous prescaler is set to 0x7F(FACTOR_A), the frequency of RTC_CALIB is f_{rtcclk}/64.When the RTCCLK is 32.768KHz, RTC_CALIB output is corresponding to 512Hz.It's recommend to using rising edge of RTC_CALIB output for there may be a light jitter on falling edge.

When the COS bit is set to 1, the RTC_CALIB frequency is:

$$f_{rtc\_calib} = \frac{f_{rtcclk}}{(FACTOR\_A+1)\times(FACTOR\_S+1)} \quad (13\text{-}5)$$

When the RTCCLK is 32.768 KHz, RTC_CALIB output is corresponding to 1Hz if prescaler are default values.

13.3.12. Alarm output

When OS control bits are not reset, RTC_ALARM alternate function output is enabled. This function will directly output the content of alarm flag bit in RTC_STAT.

The OPOL bit in RTC_CTL can configure the polarity of the alarm flag output which means that the RTC_ALARM output is the opposite of the corresponding flag bit or not.

13.3.13. RTC pin configuration

RTC_OUT0/1, RTC_TS use the same pin (PC13 or PA4). Function of (PC13 or PA4) is controlled by the RTC and regardless of PC13 GPIO configuration.

The priority of the (PC13 or PA4) output shown in ***Table 13-1 RTC pin (PC13 or PA4) configuration***

Table 13-1 RTC pin (PC13 or PA4) configuration[1]

Pin configuration and function		OS[1:0] (output selection)	COEN (calibration output)	OUT1EN	ALARMOUTTYPE(RTC_ALARM output type)	DISPU	TSEN (time stamp enabled)
Alarm out Output OD	No pull	01 or 10 or 11	Don’t care	0	0	0	Don’t care
			1	1			
	Internal Pull-up	01 or 10 or 11	Don’t care	0	0	1	Don’t care
			1	1			
Alarm out output PP		01 or 10 or 11	Don’t care	0	1	0	Don’t care
			1	1			
Calibration out output PP		00	1	0	Don’t care	Don’t care	Don’t care
TIMESTAMP input floating		00	0	Don’t care	Don’t care	Don’t care	1
		00	1	1			
		Don’t care	0				
Wake up pin or Standard GPIO		00	0	Don’t care	Don’t care	Don’t care	0
		00	1	1			
		Don’t care	0				

[1]. OD: open drain; PP: push-pull.

The (PC13 or PA4) can be used for the following purposes:

- RTC_ALARM output: this output can be RTC Alarm 0 depending on the OS [1:0] bits in the RTC_CTL register
- RTC_CALIB output: this feature is enabled by setting the COEN [23] in the RTC_CTL register
- RTC_TS: time stamp event detection

ALRMOUTTYPE in RTC_TYPE is used to select whether the RTC_ALRM is output in push-pull or open-drain mode.

It is possible to output RTC_OUT1 on PA4 pin thanks to OUT1EN bit in RTC_CTL [31].

Table 13-2 RTC_OUT configuration

OS [1:0] bits ALARM ouput enable	COEN bit (CALIB output enable)	OUT1EN bit	RTC_OUT0	RTC_OUT1

00	0	0	-	-
00	1		CALIB	-
01 or 10 or 11	Don’t care		ALRM	-
00	0	1	-	-
00	1		-	CALIB
01 or 10 or 11	0		-	ALRM
01 or 10 or 11	1		ALRM	CALIB

Table 13-3 RTC power saving mode management

Mode	Active in Mode	Exit Mode
Sleep	Yes	RTC Interrupts
Deep-sleep	Yes: if clock source is LXTAL or IRC32K	RTC Alarm / Timestamp Event
Standby	Yes: if clock source is LXTAL or IRC32K	RTC Alarm / Timestamp Event

13.3.14. RTC interrupts

All RTC interrupts are connected to the EXTI controller.

Below steps should be followed if you want to use the RTC alarm/ timestamp interrupt:

1. Configure and enable the corresponding interrupt line to RTC alarm /timestamp event of EXTI and set the rising edge for triggering
2. Configure and enable the RTC alarm/timestamp interrupt
3. Configure and enable the RTC alarm/timestamp function

Table 13-4 RTC interrupts control

Interrupt	Event flag	Control Bit	Exit Sleep	Exit Deep-sleep And Standby
Alarm 0	ALRM0F	ALRM0IE	Y	Y(1)
Timestamp	TSF	TSIE	Y	Y(1)

(1) Only active when RTC clock source is LXTAL or IRC32K.

13.4. Register definition

RTC base address: 0x4000 2800

13.4.1. Time register (RTC_TIME)

Address offset: 0x00

System reset value: 0x0000 0000 when BPSHAD = 0.

Not affected when BPSHAD = 1.

This register is write protected and can only be written in initialization state

This register has to be accessed by word (32-bit)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved									PM	HRT[1:0]		HRU[3:0]			
									rw	rw		rw			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	MNT[2:0]			MNU[3:0]				Reserved	SCT[2:0]			SCU[3:0]			
	rw			rw					rw			rw			

Bits	Fields	Descriptions
31:23	Reserved	Must be kept at reset value.
22	PM	AM/PM mark 0: AM or 24-hour format 1: PM
21:20	HRT[1:0]	Hour tens in BCD code
19:16	HRU[3:0]	Hour units in BCD code
15	Reserved	Must be kept at reset value.
14:12	MNT[2:0]	Minute tens in BCD code
11:8	MNU[3:0]	Minute units in BCD code
7	Reserved	Must be kept at reset value.
6:4	SCT[2:0]	Second tens in BCD code
3:0	SCU[3:0]	Second units in BCD code

13.4.2. Date register (RTC_DATE)

Address offset: 0x04

System reset value: 0x0000 2101 when BPSHAD = 0.

Not affected when BPSHAD = 1.

This register is write protected and can only be written in initialization state

This register has to be accessed by word (32-bit)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved								YRT[3:0]				YRU[3:0]			
								rw				rw			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DOW[2:0]			MONT	MONU[3:0]				Reserved		DAYT[1:0]		DAYU[3:0]			
rw			rw	rw						rw		rw			

Bits	Fields	Descriptions
31:24	Reserved	Must be kept at reset value.
23:20	YRT	Year tens in BCD code
19:16	YRU[3:0]	Year units in BCD code
15:13	DOW[2:0]	Days of the week 0x0: Reserved 0x1: Monday … 0x7: Sunday
12	MONT	Month tens in BCD code
11:8	MONU[3:0]	Month units in BCD code
7:6	Reserved	Must be kept at reset value.
5:4	DAYT[1:0]	Day tens in BCD code
3:0	DAYU[3:0]	Day units in BCD code

13.4.3. Control register (RTC_CTL)

Address offset: 0x08

System reset: not affected

Backup registers reset value: 0x0000 0000

This register is writing protected

This register has to be accessed by word (32-bit)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
OUT1EN	Reserved							COEN	OS[1:0]		OPOL	COS	DSM	S1H	A1H
rw								rw	rw		rw	rw	rw	w	w

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TSIE	Reserved		ALRM0IE	TSEN	Reserved		ALRM0EN	Reserved	CS	BPSHAD	REFEN	TSEG	Reserved		
rw			rw	rw			rw		rw	rw	rw	rw			

Bits	Fields	Descriptions
31	OUT1EN	RTC_OUT1 pin enable 0:RTC_OUT1 is disable 1: RTC_OUT1 is enable

30:24	Reserved	Must be kept at reset value.
23	COEN	Calibration output enable 0: Disable calibration output 1: Enable calibration output
22:21	OS[1:0]	Output selection This bit is used for selecting flag source to output 0x0: Disable output RTC_ALARM 0x1: Enable alarm0 flag output 0x2: Reseved 0x3: Reseved
20	OPOL	Output polarity This bit is used to invert output RTC_ALARM 0: Disable invert output RTC_ALARM 1: Enable invert output RTC_ALARM
19	COS	Calibration output selection Valid only when COEN=1 and prescalers are at default values 0: Calibration output is 512 Hz 1: Calibration output is 1Hz
18	DSM	Daylight saving mark This bit is flexible used by software. Often can be used to recording the daylight saving hour adjustment.
17	S1H	Subtract 1 hour(winter time change) One hour will be subtracted from current time if it is not 0 0: No effect 1: 1 hour will be subtracted at next second change time.
16	A1H	Add 1 hour(summer time change) One hour will be added from current time 0: No effect 1: 1 hour will be added at next second change time
15	TSIE	Time-stamp interrupt enable 0: Disable time-stamp interrupt 1: Enable time-stamp interrupt
14:13	Reserved	Must be kept at reset value.
12	ALRM0IE	RTC alarm-0 interrupt enable 0: Disable alarm interrupt 1: Enable alarm interrupt
11	TSEN	Time-stamp function enable 0: Disable time-stamp function

		1: Enable time-stamp function
10:9	Reserved	Must be kept at reset value.
8	ALRM0EN	Alarm-0 function enable 0: Disable alarm function 1: Enable alarm function
7	Reserved	Must be kept at reset value.
6	CS	Clock System 0: 24-hour format 1: 12-hour format Note: Can only be written in initialization state
5	BPSHAD	Shadow registers bypass control 0: Reading calendar from shadow registers 1: Reading calendar from current real-time calendar Note: If frequency of APB clock is less than seven times the frequency of RTCCLK, this bit must set to 1.
4	REFEN	Reference clock detection function enable 0: Disable reference clock detection function 1: Enable reference clock detection function Note: Can only be written in initialization state and FACTOR_S must be 0x00FF
3	TSEG	Valid event edge of time-stamp 0: rising edge is valid event edge for time-stamp event 1: falling edge is valid event edge for time-stamp event
2:0	Reserved	Must be kept at reset value.

13.4.4. Status register (RTC_STAT)

Address offset: 0x0C

System reset: Only INITM, INITF and RSYNF bits are set to 0. Others are not affected

Backup registers reset value: 0x0000 0007

This register has to be accessed by word (32-bit)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															SCPF
															r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved			TSOVRF	TSF	Reserved		ALRM0F	INITM	INITF	RSYNF	YCM	SOPF	Reserved		ALRM0WF
			rc_w0	rc_w0			rc_w0	rw	r	rc_w0	r	r			r

Bits	Fields	Descriptions
31:17	Reserved	Must be kept at reset value.
16	SCPF	Smooth calibration pending flag

		Set to 1 by hardware when software writes to RTC_HRFC without entering initialization mode and set to 0 by hardware when smooth calibration configuration is taken into account.
15:13	Reserved	Must be kept at reset value.
12	TSOVRF	Time-stamp overflow flag This bit is set by hardware when a time-stamp event is detected if TSF bit is set before. Cleared by software writing 0.
11	TSF	Time-stamp flag Set by hardware when time-stamp event is detected. Cleared by software writing 0.
10:9	Reserved	Must be kept at reset value.
8	ALRM0F	Alarm-0 occurs flag Set to 1 by hardware when current time/date matches the time/date of alarm 0 setting value. Cleared by software writing 0.
7	INITM	Enter initialization mode 0: Free running mode 1: Enter initialization mode for setting calendar time/date and prescaler. Counter will stop under this mode.
6	INITF	Initialization state flag Set to 1 by hardware and calendar register and prescaler can be programmed in this state. 0: Calendar registers and prescaler register cannot be changed 1: Calendar registers and prescaler register can be changed
5	RSYNF	Register synchronization flag Set to 1 by hardware every 2 RTCCLK which will copy current calendar time/date into shadow register. Initialization mode (INITM), shift operation pending flag (SOPF) or bypass mode (BPSHAD) will clear this bit. This bit is also can be cleared by software writing 0. 0:Shadow register are not yet synchronized 1:Shadow register are synchronized
4	YCM	Year configuration mark Set by hardware if the year field of calendar date register is not the default value 0. 0: Calendar has not been initialized 1: Calendar has been initialized
3	SOPF	Shift function operation pending flag 0: No shift operation is pending 1: Shift function operation is pending

2:1	Reserved	Must be kept at reset value.
0	ALRM0WF	Alarm 0 configuration can be write flag Set by hardware if alarm register can be wrote after ALRM0EN bit has reset. 0: Alarm registers programming is not allowed. 1: Alarm registers programming is allowed.

13.4.5. Prescaler register (RTC_PSC)

Address offset: 0x10

System reset: not effected

Backup registers reset value: 0x007F 00FF

This register is write protected and can only be written in initialization state

This register has to be accessed by word (32-bit)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved									FACTOR_A[6:0]						
									rw						

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	FACTOR_S[14:0]														
	rw														

Bits	Fields	Descriptions
31:23	Reserved	Must be kept at reset value.
22:16	FACTOR_A[6:0]	Asynchronous prescaler factor ck_apre frequency = RTCCLK frequency/(FACTOR_A+1)
15	Reserved	Must be kept at reset value.
14:0	FACTOR_S[14:0]	Synchronous prescaler factor ck_spre frequency = ck_apre frequency/(FACTOR_S+1)

13.4.6. Alarm 0 time and date register (RTC_ALRM0TD)

Address offset: 0x1C

System reset: not effect

Backup registers reset value: 0x0000 0000

This register is write protected and can only be written in initialization state or ALRM0WF is set to 1.

This register has to be accessed by word (32-bit)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MSKD	DOWS	DAYT[1:0]		DAYU[3:0]				MSKH	PM	HRT[1:0]		HRU[3:0]			
rw	rw	rw		rw				rw	rw	rw		rw			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MSKM	MNT[2:0]			MNU[3:0]				MSKS	SCT[2:0]			SCU[3:0]			
rw	rw			rw				rw	rw			rw			

Bits	Fields	Descriptions
31	MSKD	Alarm date mask bit 0: Not mask date/day field 1: Mask date/day field
30	DOWS	Day of the week selected 0: DAYU[3:0] indicates the date units 1: DAYU[3:0] indicates the week day and DAYT[1:0] has no means.
29:28	DAYT[1:0]	Date tens in BCD code
27:24	DAYU[3:0]	Date units or week day in BCD code
23	MSKH	Alarm hour mask bit 0: Not mask hour field 1: Mask hour field
22	PM	AM/PM flag 0: AM or 24-hour format 1: PM
21:20	HRT[1:0]	Hour tens in BCD code
19:16	HRU[3:0]	Hour units in BCD code
15	MSKM	Alarm minutes mask bit 0: Not mask minutes field 1: Mask minutes field
14:12	MNT[2:0]	Minutes tens in BCD code
11:8	MNU[3:0]	Minutes units in BCD code
7	MSKS	Alarm second mask bit 0: Not mask second field 1: Mask second field
6:4	SCT[2:0]	Second tens in BCD code
3:0	SCU[3:0]	Second units in BCD code

13.4.7. Write protection key register (RTC_WPK)

Address offset: 0x24

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								WPK[7:0]							
								w							

Bits	Fields	Descriptions
31:8	Reserved	Must be kept at reset value.
7:0	WPK[7:0]	Key for write protection

13.4.8. Sub second register (RTC_SS)

Address offset: 0x28

System reset value: 0x0000 0000 when BPSHAD = 0.

Not affected when BPSHAD = 1.

This register has to be accessed by word (32-bit)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSC[15:0]															
r															

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	SSC[15:0]	Sub second value This value is the counter value of synchronous prescaler. Second fraction value is calculated by the below formula: Second fraction = (FACTOR_S - SSC) / (FACTOR_S + 1)

13.4.9. Shift function control register (RTC_SHIFTCTL)

Address offset: 0x2C

System reset: not effect

Backup registers reset value: 0x0000 0000

This register is writing protected and can only be wrote when SOPF=0

This register has to be accessed by word (32-bit)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
A1S	Reserved														
w															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	SFS[14:0]														
	w														

Bits	Fields	Descriptions
31	A1S	One second add

		0: Not add 1 second 1: Add 1 second to the clock/calendar. This bit is jointly used with SFS field to add a fraction of a second to the clock.
30:15	Reserved	Must be kept at reset value.
14:0	SFS[14:0]	Subtract a fraction of a second The value of this bit will add to the counter of synchronous prescaler. When only using SFS, the clock will delay because the synchronous prescaler is a down counter: Delay (seconds) = SFS / (FACTOR_S + 1) When jointly using A1S and SFS, the clock will advance: Advance (seconds) = (1 - (SFS / (FACTOR_S + 1)))

Note: Writing to this register will cause RSYNF bit to be cleared.

13.4.10. Time of time stamp register (RTC_TTS)

Address offset: 0x30

Backup registers reset value: 0x0000 0000

System reset: no effect

This register will record the calendar time when TSF is set to 1.

Reset TSF bit will also clear this register.

This register has to be accessed by word (32-bit)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved									PM	HRT[1:0]		HRU[3:0]			
									r	r		r			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	MNT[2:0]			MNU[3:0]				Reserved	SCT[2:0]			SCU[3:0]			
	r			r					r			r			

Bits	Fields	Descriptions
31:23	Reserved	Must be kept at reset value.
22	PM	AM/PM mark 0:AM or 24-hour format 1:PM
21:20	HRT[1:0]	Hour tens in BCD code
19:16	HRU[3:0]	Hour units in BCD code
15	Reserved	Must be kept at reset value.
14:12	MNT[2:0]	Minute tens in BCD code
11:8	MNU[3:0]	Minute units in BCD code

7	Reserved	Must be kept at reset value.
6:4	SCT[2:0]	Second tens in BCD code
3:0	SCU[3:0]	Second units in BCD code

13.4.11. Date of time stamp register (RTC_DTS)

Address offset: 0x34
Backup registers reset value: 0x0000 0000
System reset: no effect
This register will record the calendar date when TSF is set to 1.
Reset TSF bit will also clear this register.

This register has to be accessed by word (32-bit)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DOW[2:0]			MONT	MONU[3:0]				Reserved		DAYT[1:0		DAYU[3:0]			
r			r	r						r					

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:13	DOW[2:0]	Days of the week
12	MONT	Month tens in BCD code
11:8	MONU[3:0]	Month units in BCD code
7:6	Reserved	Must be kept at reset value.
5:4	DAYT[1:0]	Day tens in BCD code
3:0	DAYU[3:0]	Day units in BCD code

13.4.12. Sub second of time stamp register (RTC_SSTS)

Address offset: 0x38
Backup registers reset: 0x0000 0000
System reset: no effect
This register will record the calendar date when TSF is set to 1.
Reset TSF bit will also clear this register.

This register has to be accessed by word (32-bit)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

SSC[15:0]
r

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	SSC[15:0]	Sub second value This value is the counter value of synchronous prescaler when TSF is set to 1.

13.4.13. High resolution frequency compensation register (RTC_HRFC)

Address offset: 0x3C

Backup registers reset: 0x0000 0000

System Reset: no effect

This register is write protected.

This register has to be accessed by word (32-bit)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FREQI	CWND8	CWND16	Reserved				CMSK[8:0]								
rw	rw	rw					rw								

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15	FREQI	Increase RTC frequency by 488.5PPM 0: No effect 1: One RTCCLK pulse is inserted every 2^{11} pulses. This bit should be used in conjunction with CMSK bit. If the input clock frequency is 32.768KHz, the number of RTCCLK pulses added during 32s calibration window is (512 * FREQI) - CMSK
14	CWND8	Frequency compensation window 8 second selected 0: No effect 1: Calibration window is 8 second Note: When CWND8=1, CMSK[1:0] are stuck at “00”.
13	CWND16	Frequency compensation window 16 second selected 0: No effect 1: Calibration window is 16 second Note: When CWND16=1, CMSK[0] are stuck at “0”.
12:9	Reserved	Must be kept at reset value.
8:0	CMSK[8:0]	Calibration mask number The number of mask pulse out of 2^{20} RTCCLK pulse. This feature will decrease the frequency of calendar with a resolution of 0.9537

PPM.

13.4.14. Type register (RTC_TYPE)

Address offset: 0x40

Backup registers reset: 0x0000 0000

System reset: no effect

This register has to be accessed by word (32-bit)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved													ALRMOUTTYPE	Reserved	
													rw		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DISPU	Reserved														
rw															

Bits	Fields	Descriptions
31:19	Reserved	Must be kept at reset value.
18	ALRMOUTTYPE	RTC_ALARM Output Type 0: Open-drain output type 1: Push-pull output type
17:16	Reserved	Must be kept at reset value.
15	DISPU	RTC_ALRM output pull up disable bit 0: No pull-up is applied on RTC_ALRM output 1: A pull-up is applied on RTC_ALRM output
14:0	Reserved	Must be kept at reset value.

13.4.15. Alarm 0 sub second register (RTC_ALRM0SS)

Address offset: 0x44

Backup registers reset: 0x0000 0000

System reset: no effect

This register is write protected and can only be wrote when ALRM0EN=0 or INITM=1

This register has to be accessed by word (32-bit)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved				MSKSSC[3:0]				Reserved							
				rw											

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	SSC[14:0]														
	rw														

Bits	Fields	Descriptions
31:28	Reserved	Must be kept at reset value.

27:24	MSKSSC[3:0]	Mask control bit of SSC 0x0: Mask alarm sub second setting. The alarm asserts at every second time point if all the rest alarm fields are matched. 0x1: SSC[0] is to be compared and all others are ignored 0x2: SSC[1:0] is to be compared and all others are ignored 0x3: SSC[2:0] is to be compared and all others are ignored 0x4: SSC[3:0] is to be compared and all others are ignored 0x5: SSC[4:0] is to be compared and all others are ignored 0x6: SSC[5:0] is to be compared and all others are ignored 0x7: SSC[6:0] is to be compared and all others are ignored 0x8: SSC[7:0] is to be compared and all others are ignored 0x9: SSC[8:0] is to be compared and all others are ignored 0xA: SSC[9:0] is to be compared and all others are ignored 0xB: SSC[10:0] is to be compared and all others are ignored 0xC: SSC[11:0] is to be compared and all others are ignored 0xD: SSC[12:0] is to be compared and all others are ignored 0xE: SSC[13:0] is to be compared and all others are ignored 0xF: SSC[14:0] is to be compared and all others are ignored Note: The bit 15 of synchronous counter (SSC[15] in RTC_SS) is never compared.
23:15	Reserved	Must be kept at reset value.
14:0	SSC[14:0]	Alarm sub second value This value is the alarm sub second value which is to be compared with synchronous prescaler counter SSC. Bit number is controlled by MSKSSC bits.

13.4.16. Backup registers (RTC_BKPx) (x=0..3)

Address offset: 0x50~0x5C

Backup registers reset: 0x0000 0000

System reset: no effect

This register has to be accessed by word (32-bit)

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="16">Reserved</td></tr>
<tr><td colspan="16">rw</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="16">DATA[15:0]</td></tr>
<tr><td colspan="16">rw</td></tr>
</table>

Bits	Fields	Descriptions
15:0	DATA [15:0]	Data These registers can be wrote or read by software. The content remains valid even in power saving mode.

14. Timer (TIMERx)

Table 14-1. Timers (TIMERx) are devided into four sorts

TIMER	**TIMER0**	**TIMER2**	**TIMER13**	**TIMER15/16**
TYPE	Advanced	General-L0	General-L2	General-L4
Prescaler	16-bit	16-bit	16-bit	16-bit
Counter	16-bit	16-bit	16-bit	16-bit
Count mode	UP,DOWN, Center-aligned	UP,DOWN, Center-aligned	UP ONLY	UP ONLY
Repetition	●	×	×	●
CH Capture/ Compare	4	4	1	1
Complementary & Dead-time	3	×	×	●
Break	●	×	×	●
Single Pulse	●	●	×	●
Quadrature Decoder	●	●	×	×
Master-slave management	●	●	×	×
Inter connection	●(1)	●(2)	×	×
DMA	●	●	×	●
Debug Mode	●	●	●	●

(1)	TIMER0	**IT**I0: Reserved	**ITI1:** Reserved	**ITI2:** TIMER2_TRGO	**ITI3:** TIMER16_CH1
(2)	TIMER2	**ITI0:** TIMER0_TRGO	**ITI1:** Reserved	**ITI2:** Reserved	**ITI3:** TIMER13_CH1
(3)	TIMER13	**ITI0:** Reserved	**ITI1:** Reserved	**ITI2:** Reserved	**ITI3:** Reserved
(4)	TIMER15	**ITI0:** Reserved	**ITI1:** Reserved	**ITI2:** Reserved	**ITI3:** Reserved
	TIMER16	**ITI0:** Reserved	**ITI1:** Reserved	**ITI2:** Reserved	**ITI3:** Reserved

14.1. Advanced timer (TIMERx,x=0)

14.1.1. Overview

The advanced timer module (TIMER0) is a four-channel timer that supports both input capture and output compare. They can generate PWM signals to control motor or be used for power management applications. The advanced timer has a 16-bit counter that can be used as an unsigned counter.

In addition, the advanced timers can be programmed and be used for counting, their external events can be used to drive other timers.

Timer also includes a dead-time insertion module which is suitable for motor control applications.

Timers are completely independent with each other, but they may be synchronized to provide a larger timer with their counter value increasing in unison.

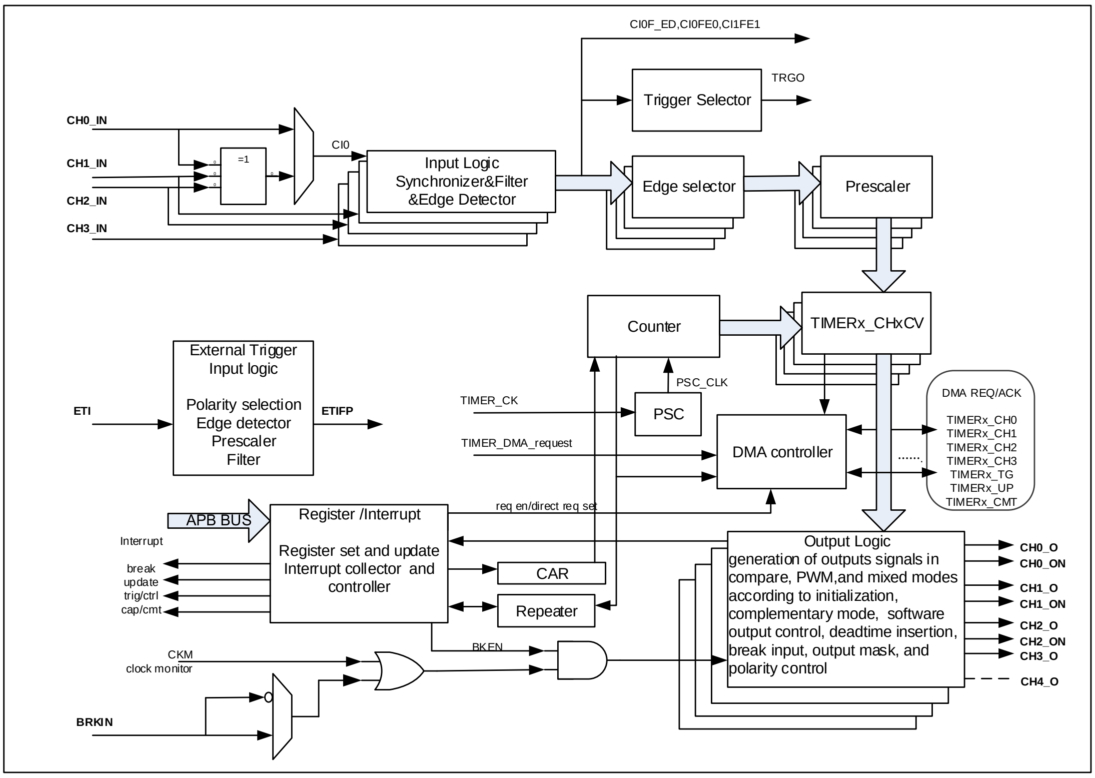
14.1.2. Characteristics

- Total channel num: 4.
- Counter width: 16 bits.
- Clock source of timer is selectable: internal clock, internal trigger, external input, external trigger.
- Multiple counter modes: up counting, down counting and center-aligned counting.
- Quadrature decoder: used for motion tracking and determination of both rotation direction and position.
- Hall sensor function: used for 3-phase motor control.
- Programmable prescaler: 16 bits. The factor can be changed ongoing.
- Each channel is user-configurable: input capture mode, output compare mode, programmable PWM mode and single pulse mode.
- Programmable dead time insertion.
- Auto reload function.
- Programmable counter repetition function.
- Break input.
- Interrupt output or DMA request: update event, trigger event, compare/capture event and break input.
- Daisy chaining of timer module allows a single timer to start multiple timers.
- Timer synchronization allows the selected timers to start counting on the same clock cycle.
- Timer master-slave management.

14.1.3. Block diagram

Figure 14-1. Advanced timer block diagram provides details of the internal configuration of the advanced timer.

Figure 14-1. Advanced timer block diagram



Note: CH4_O is an internal signal with no corresponding pin.

14.1.4. Function overview

Clock source configuration

The clock source of the advanced timer can be either the CK_TIMER or an alternate clock source controlled by TSCFGy[2:0] in SYSCFG_TIMER0CFG(y=0,1…7).

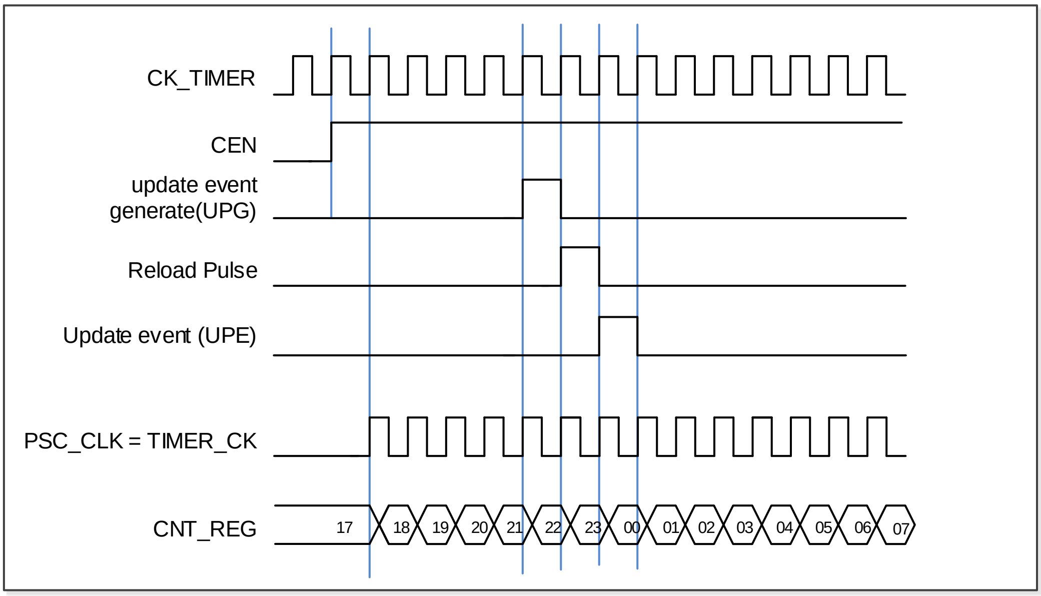
- TSCFGy[2:0] = 3’b000 in SYSCFG_TIMER0CFG(y=0,1…7). Internal clock CK_TIMER is selected as timer clock source which is from module RCU.

The default clock source is the CK_TIMER for driving the counter prescaler when TSCFGy[2:0] = 3’b000 in SYSCFG_TIMER0CFG(y=0,1…7). When the CEN is set, the CK_TIMER will be divided by PSC value to generate PSC_CLK.

In this mode, the TIMER_CK which drives counter’s prescaler to count is equal to CK_TIMER which is from RCU module.

- if TSCFGy[2:0] !=3’b000 in SYSCFG_TIMER0CFG(y=0,1,2,6), the prescaler is clocked by other clock sources selected in the TSCFG6[2:0] register, more details will be introduced later. When the TSCFGy[2:0] (y=3,4,5) are setting to an available value, the internal clock TIMER_CK is the counter prescaler driving clock source.

Figure 14-2. Normal mode, internal clock divided by 1



- TSCFG6[2:0] are setting to an available value (external clock mode 0). External input pin is selected as timer clock source.

The TIMER_CK, which drives counter's prescaler to count, can be triggered by the event of rising or falling edge on the external pin TIMERx_CH0/TIMERx_CH1. This mode can be selected by setting TSCFG6[2:0] to 0x4, 0x5, 0x6.

And, the counter prescaler can also be driven by rising edge on the internal trigger input pin ITI0/2/3. This mode can be selected by setting TSCFG6[2:0] to 0x1,0x2,0x3

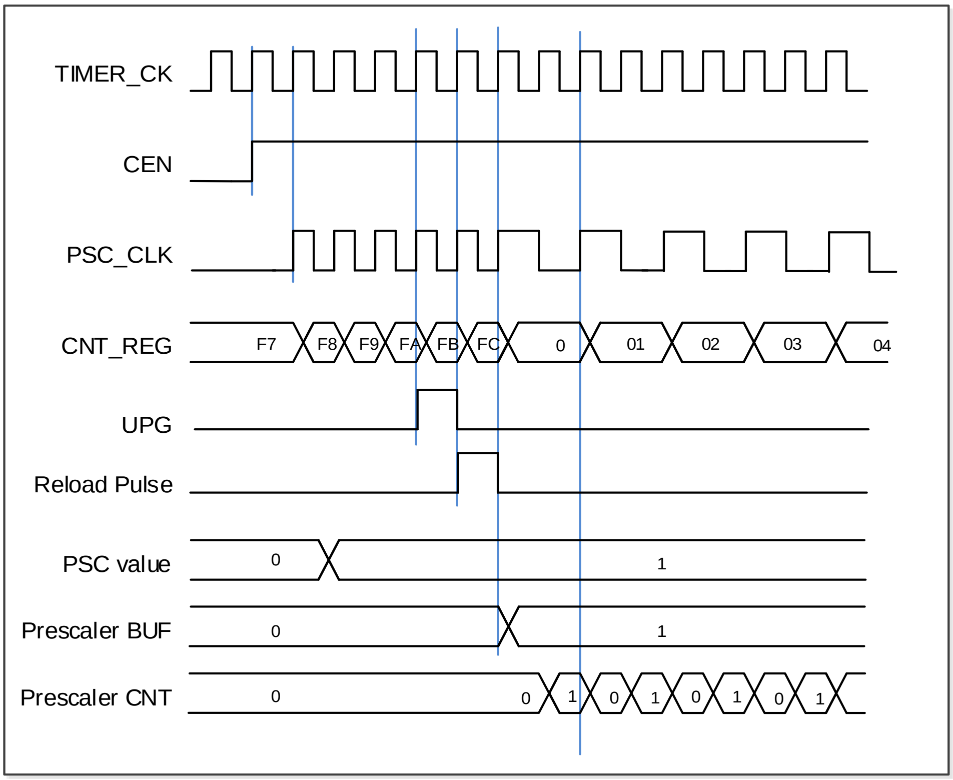
- SMC1= 1'b1 (external clock mode 1). External input ETI is selected as timer clock source.

The TIMER_CK, which drives counter's prescaler to count, can be triggered by the event of rising or falling edge on the external pin ETI. This mode can be selected by setting the SMC1 bit in the TIMERx_SMCFG register to 1. The other way to select the ETI signal as the clock source is setting TSCFG6[2:0] to 0x7. Note that the ETI signal is derived from the ETI pin sampled by a digital filter. When the ETI signal is selected as the clock source, the trigger controller including the edge detection circuitry will generate a clock pulse on each ETI signal rising edge to clock the counter prescaler.

Clock Prescaler

The prescaler can divide the timer clock (TIMER_CK) to a counter clock (PSC_CLK) by any factor ranging from 1 to 65536. It is controlled by prescaler register (TIMERx_PSC) which can be changed ongoing, but it is adopted at the next update event.

Figure 14-3. Counter timing diagram with prescaler division change from 1 to 2



Counter up counting

In this mode, the counter counts up continuously from 0 to the counter reload value, which is defined in the TIMERx_CAR register, in a count-up direction. Once the counter reaches the counter reload value, the counter restarts from 0. If the repetition counter is set, the update event will be generated after (TIMERx_CREP+1) times of overflow. Otherwise the update event is generated each time when counter overflows. The counting direction bit DIR in the TIMERx_CTL0 register should be set to 0 for the up-counting mode.

Whenever, if the update event software trigger is enabled by setting the UPG bit in the TIMERx_SWEVG register, the counter value will be initialized to 0 and an update event will be generated.

If the UPDIS bit in TIMERx_CTL0 register is set, the update event is disabled.

When an update event occurs, all the registers (repetition counter register, auto reload register, prescaler register) are updated.

Figure 14-4. Timing chart of up counting mode, PSC=0/1 and *Figure 14-5. Timing chart of up counting mode, change TIMERx_CAR ongoing* show some examples of the counter behavior for different clock prescaler factors when TIMERx_CAR=0x63.

Figure 14-4. Timing chart of up counting mode, PSC=0/1

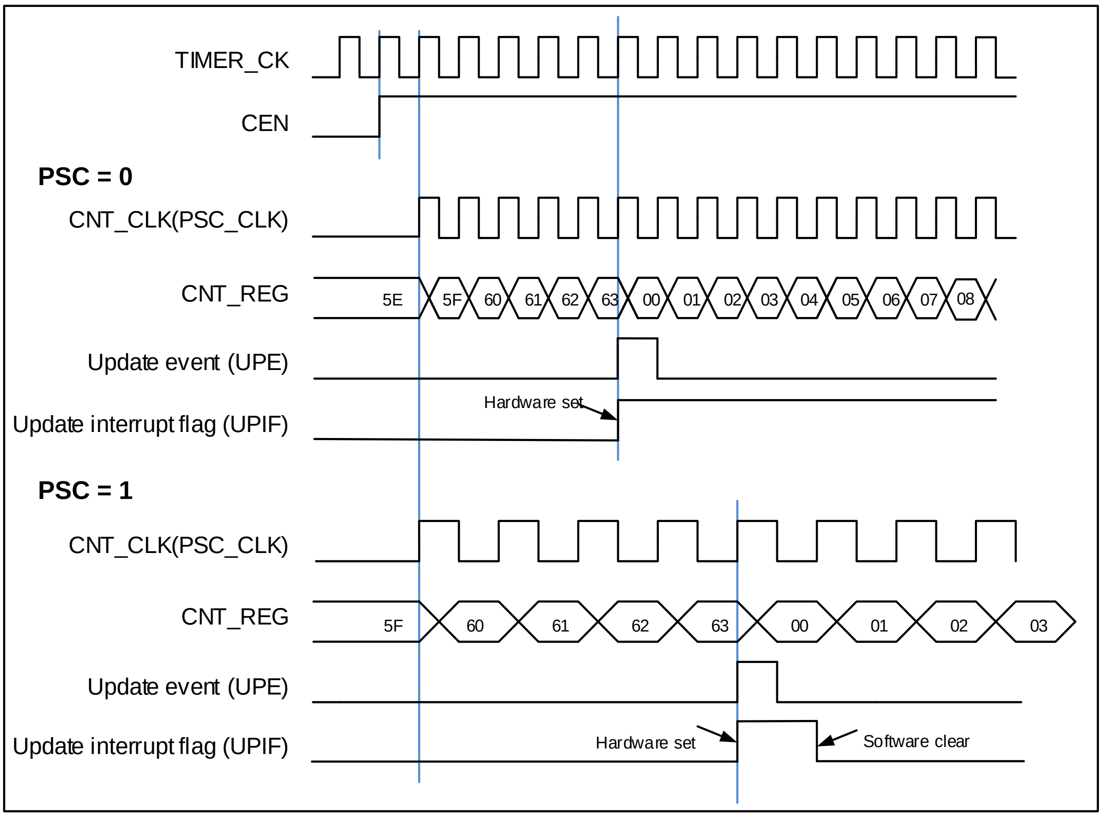
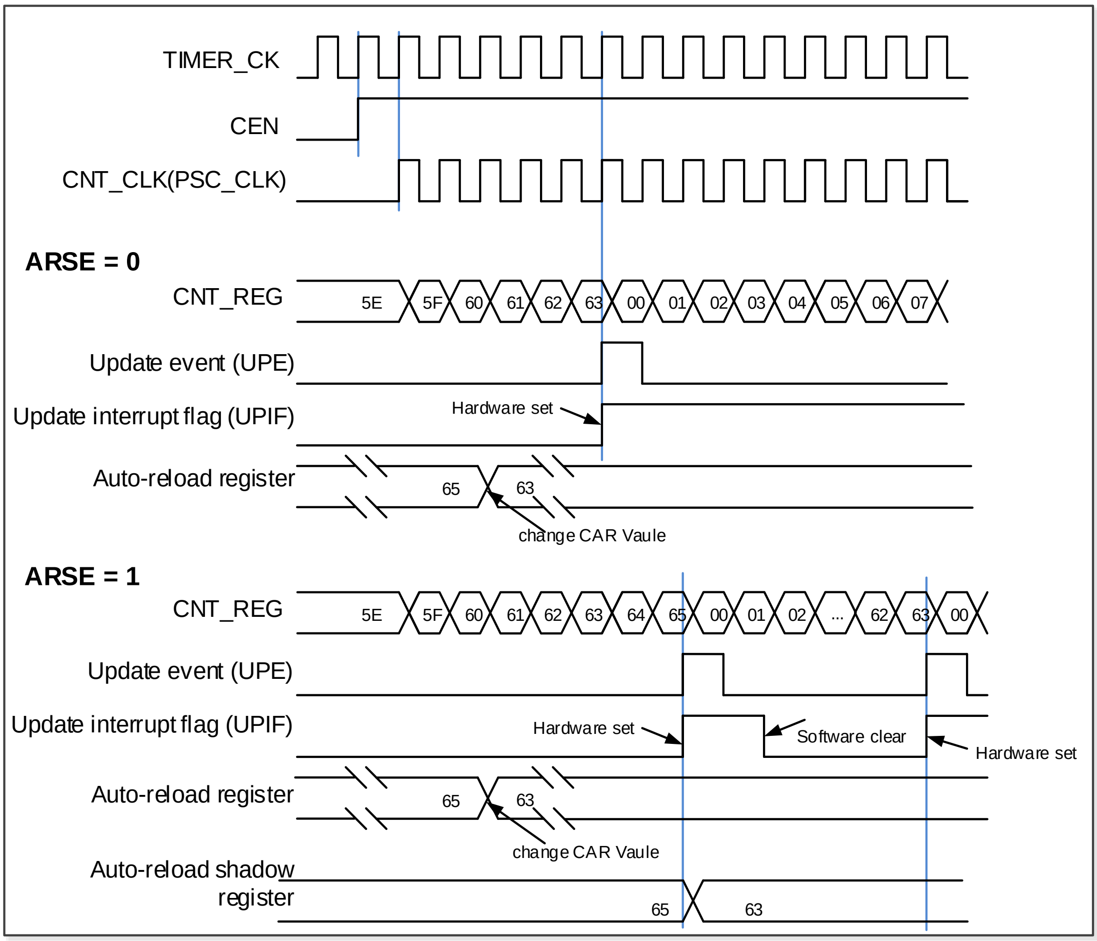


Figure 14-5. Timing chart of up counting mode, change TIMERx_CAR ongoing



Counter down counting

In this mode, the counter counts down continuously from the counter reload value, which is defined in the TIMERx_CAR register, in a count-down direction. Once the counter reaches 0, the counter restarts to count again from the counter reload value. If the repetition counter is set, the update event will be generated after (TIMERx_CREP+1) times of underflow. Otherwise, the update event is generated each time when counter underflows. The counting direction bit DIR in the TIMERx_CTL0 register should be set to 1 for the down counting mode.

When the update event is set by the UPG bit in the TIMERx_SWEVG register, the counter value will be initialized to the counter reload value and an update event will be generated.

If the UPDIS bit in TIMERx_CTL0 register is set, the update event is disabled.

When an update event occurs, all the registers (repetition counter register, auto reload register, prescaler register) are updated.

Figure 14-6. Timing chart of down counting mode, PSC=0/1 and *Figure 14-7. Timing chart of down counting mode, change TIMERx_CAR ongoing* show some examples of the counter behavior in different clock frequencies when TIMERx_CAR = 0x63.

Figure 14-6. Timing chart of down counting mode, PSC=0/1

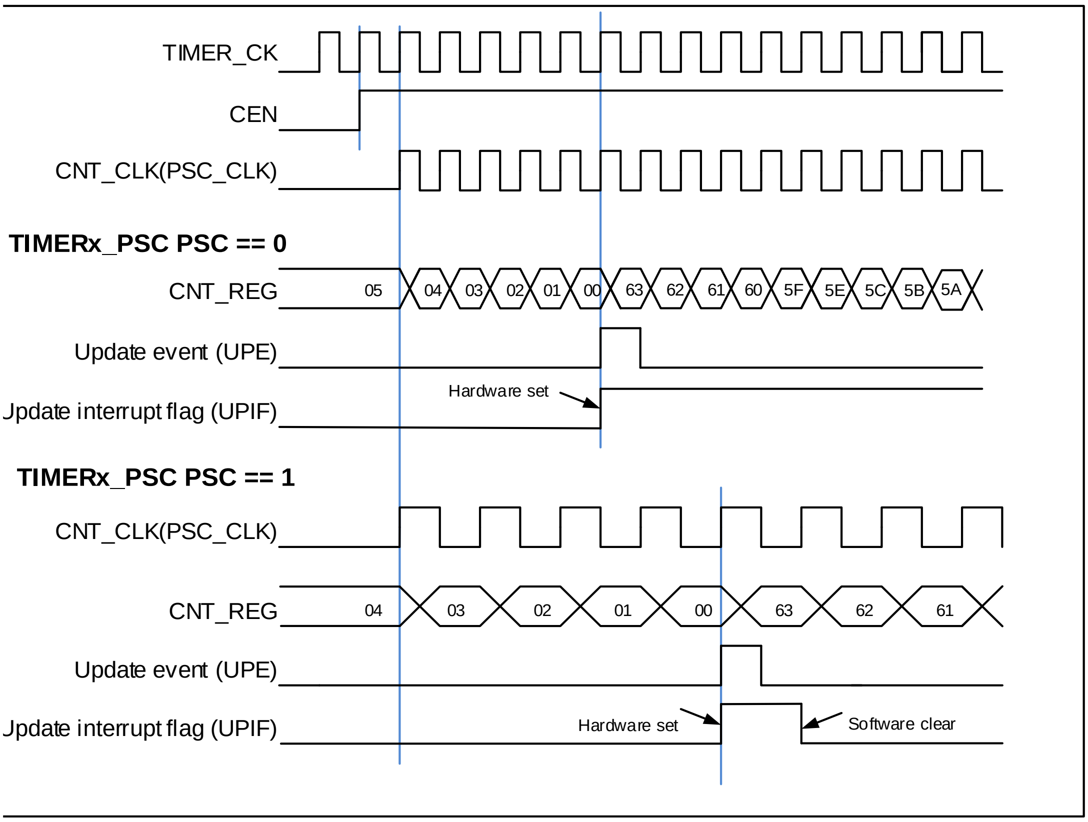
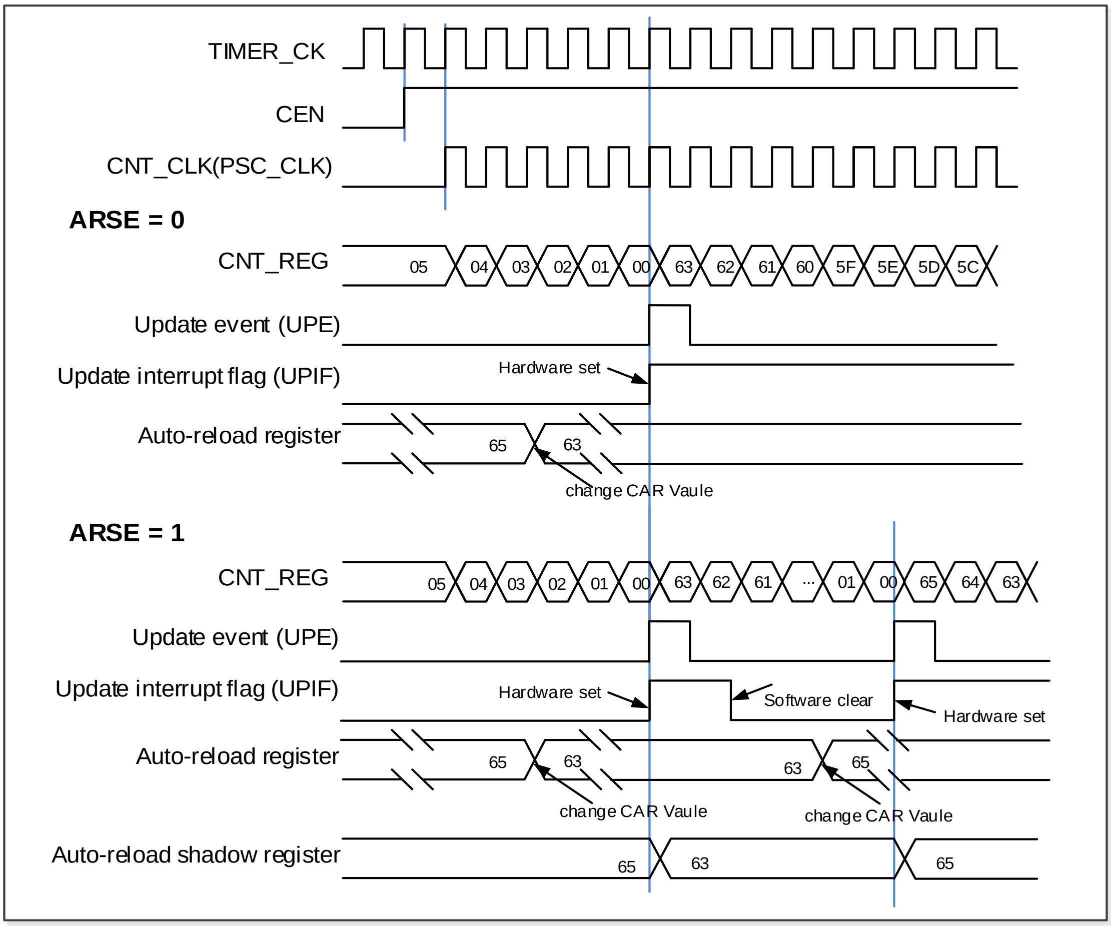


Figure 14-7. Timing chart of down counting mode, change TIMERx_CAR ongoing



Counter 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 (TIMERx_CREP-1) in the count-up direction and generates an underflow event when the counter counts to 1 in the count-down direction. The counting direction bit DIR in the TIMERx_CTL0 register is read-only and indicates the counting direction when in the center-aligned counting mode.

Setting the UPG bit in the TIMERx_SWEVG register will initialize the counter value to 0 and generate an update event irrespective of whether the counter is counting up or down in the center-aligned counting mode.

The UPIF bit in the TIMERx_INTF register will be set to 1 either when an underflow event or an overflow event occurs. While the CHxIF bit is associated with the value of CAM in TIMERx_CTL0. The details refer to *Figure 14-8. Timing chart of center-aligned counting* .

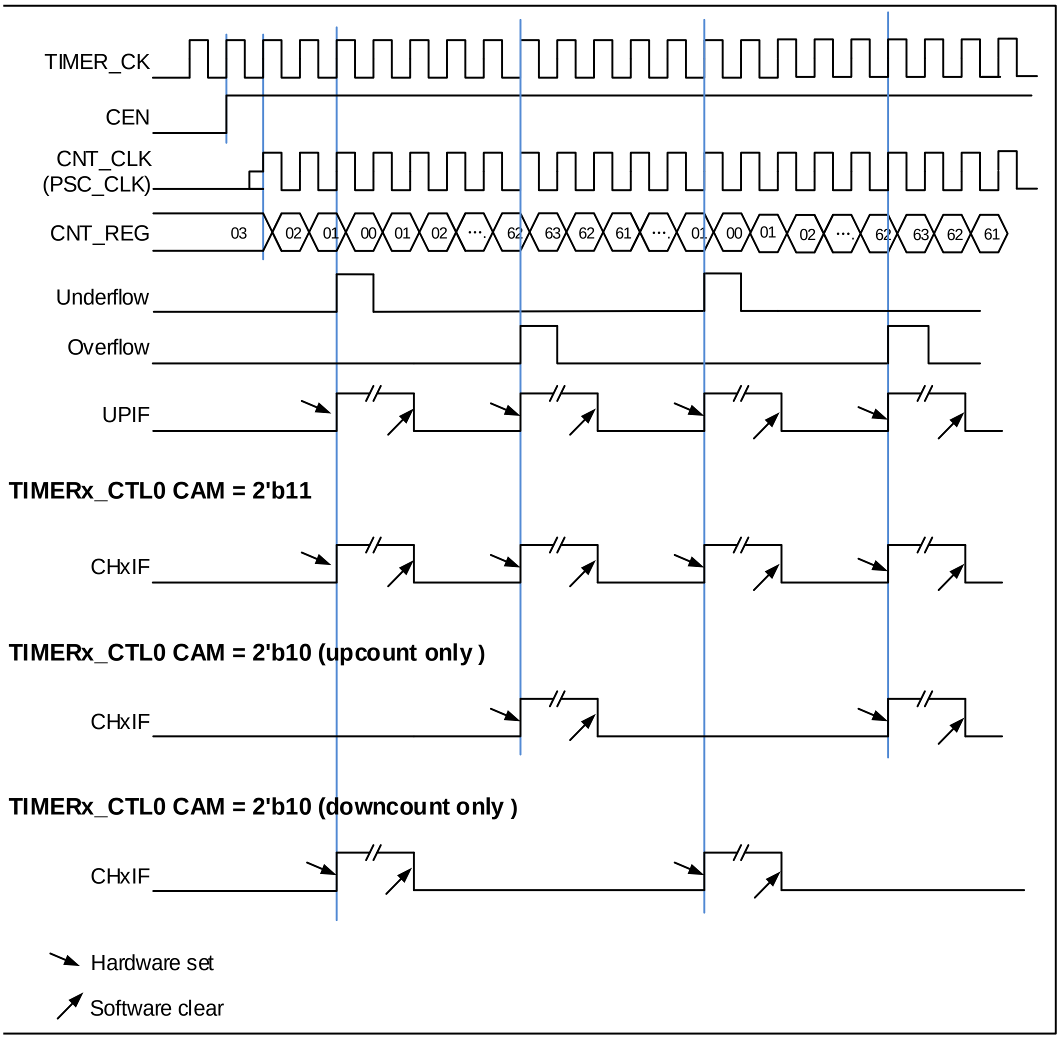
If the UPDIS bit in the TIMERx_CTL0 register is set, the update event is disabled.

When an update event occurs, all the registers (repetition counter register, auto-reload register, prescaler register) are updated.

Figure 14-8. Timing chart of center-aligned counting

shows some examples of the counter behavior when TIMERx_CAR=0x63. TIMERx_PSC=0x0.

Figure 14-8. Timing chart of center-aligned counting mode



Update event (from overflow/underflow) rate configuration

The rate of update events generation (from overflow and underflow events) is used to generate the update event or update the timer registers only after a given number (N+1) cycles of the counter, where N is the value of CREP bit in TIMERx_CREP register. The repetition counter is decremented at each counter overflow in up counting mode, at each counter underflow in down counting mode or at each counter overflow and at each counter underflow in center-aligned mode.

Setting the UPG bit in the TIMERx_SWEVG register will reload the content of CREP in TIMERx_CREP register and generate an update event.

For odd values of CREP in center-aligned mode, the update event occurs either on the overflow or on the underflow depending on when the CREP register was written and when the counter was started. The update event is generated at overflow when the CREP was written before starting the counter and generated at underflow when the CREP was written after starting the counter.

Figure 14-9. Repetition counter timing chart of center-aligned counting mode

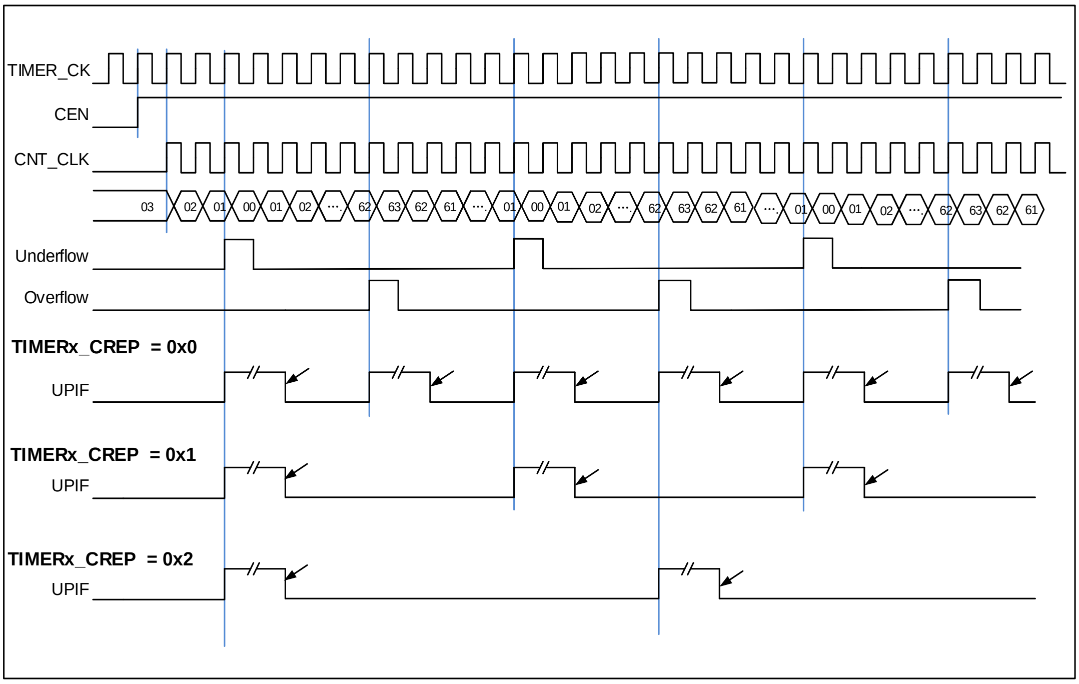


Figure 14-10. Repetition counter timing chart of up counting mode

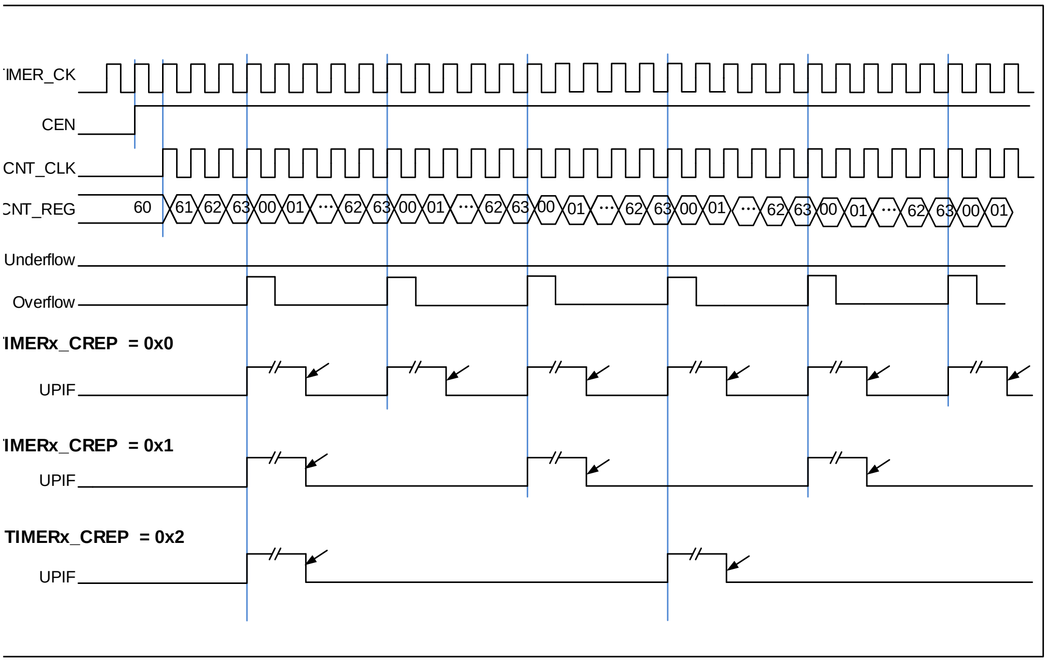
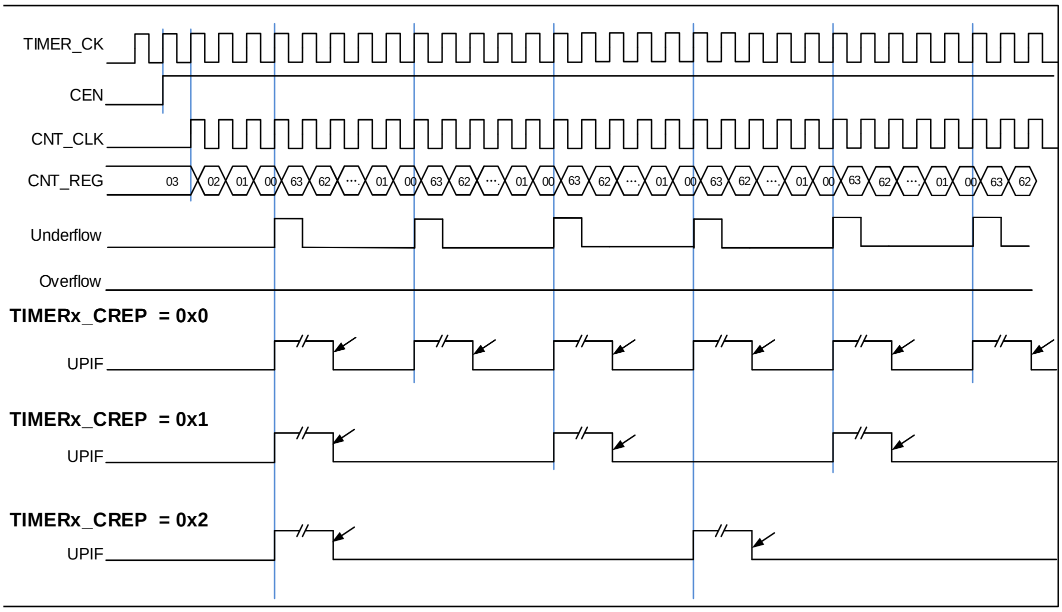


Figure 14-11. Repetition counter timing chart of down counting mode



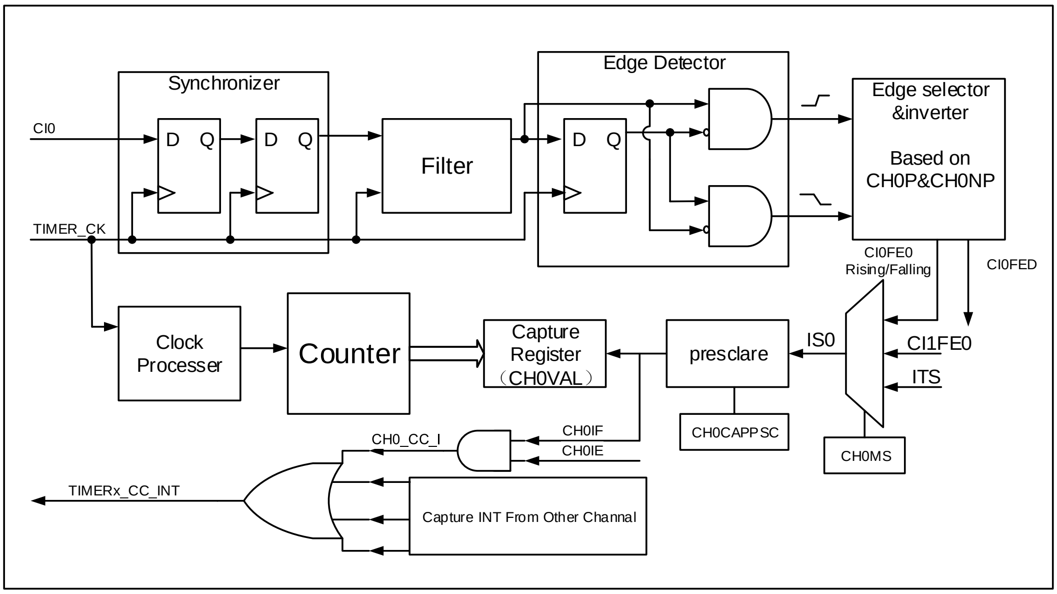
Input capture and output compare channels

The advanced timer has four independent channels which can be used as capture inputs or compare outputs. Each channel is built around a channel capture compare register including an input stage, a channel controller and an output stage.

Input capture mode

Input capture mode allows the channel to perform measurements such as pulse timing, frequency, period, duty cycle and so on. The input stage consists of a digital filter, a channel polarity selection, edge detection and a channel prescaler. When a selected edge occurs on the channel input, the current value of the counter is captured into the TIMERx_CHxCV register, at the same time the CHxIF bit is set and the channel interrupt is generated if it is enabled when CHxIE=1.

Figure 14-12. Input capture logic



The input signals of channelx (CIx) can be the TIMERx_CHx signal or the XOR signal of the TIMERx_CH0, TIMERx_CH1 and TIMERx_CH2 signals. First, the input signal of channel (CIx) is synchronized to TIMER_CK signal, and then sampled by a digital filter to generate a filtered input signal. Then through the edge detector, the rising or falling edge is detected by configuring CHxP bit. The input capture signal can also be selected from the input signal of other channel or the internal trigger signal by configuring CHxMS bits. The IC prescaler makes several input events generate one effective capture event. On the capture event, TIMERx_CHxCV will store the value of counter.

So, the process can be divided into several steps as below:

Step1: Filter configuration (CHxCAPFLT in TIMERx_CHCTL0).

Based on the input signal and quality of requested signal, configure compatible CHxCAPFLT.

Step2: Edge selection (CHxP/CHxNP in TIMERx_CHCTL2).

Rising edge or falling edge, choose one by configuring CHxP/CHxNP bits.

Step3: Capture source selection (CHxMS in TIMERx_CHCTL0).

As soon as selecting one input capture source by CHxMS, the channel must be set to input mode (CHxMS! =0x0) and TIMERx_CHxCV cannot be written any more.

Step4: Interrupt enable (CHxIE and CHxDEN in TIMERx_DMAINTEN).

Enable the related interrupt to get the interrupt and DMA request.

Step5: Capture enable (CHxEN in TIMERx_CHCTL2).

Result: When the wanted input signal is captured, TIMERx_CHxCV will be set by counter's value and CHxIF is asserted. If the CHxIF is 1, the CHxOF will also be asserted. The interrupt and DMA request will be asserted or not based on the configuration of CHxIE and CHxDEN in TIMERx_DMAINTEN.

Direct generation: A DMA request or interrupt is generated by setting CHxG directly.

The input capture mode can be also used for pulse width measurement from signals on the TIMERx_CHx pins. For example, PWM signal connects to CI0 input. Select CI0 as channel 0 capture signals by setting CH0MS to 2’b01 in the channel control register (TIMERx_CHCTL0) and set capture on rising edge. Select CI0 as channel 1 capture signal by setting CH1MS to 2’b10 in the channel control register (TIMERx_CHCTL0) and set capture on falling edge. The counter is set to restart mode and is restarted on channel 0 rising edge. Then the TIMERX_CH0CV can measure the PWM period and the TIMERx_CH1CV can measure the PWM duty cycle.

Output compare mode

Figure 14-13. Output compare logic (with complementary output, x=0,1,2)

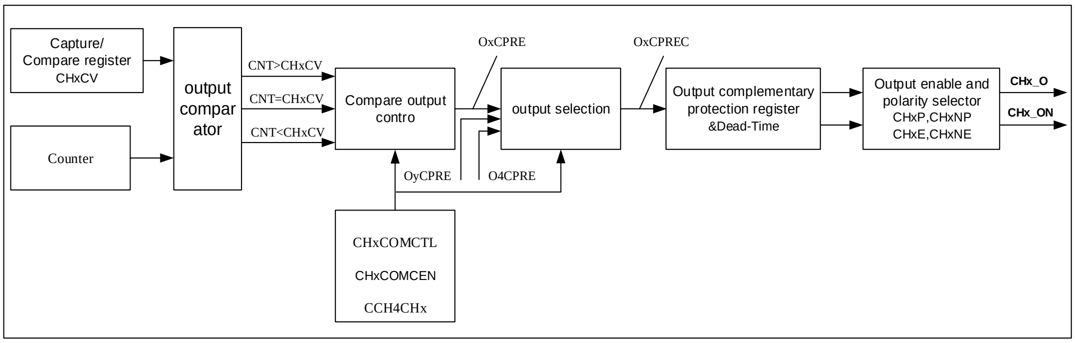


Figure 14-14. Output compare logic (CH3_O)

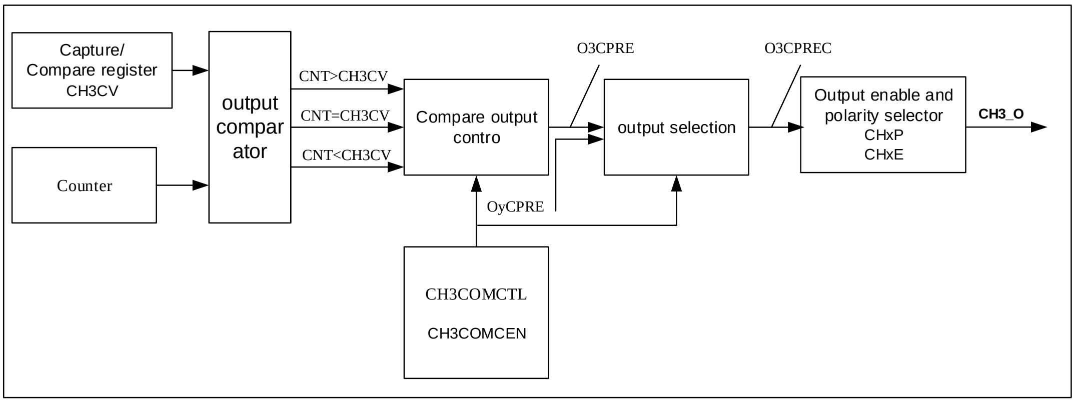
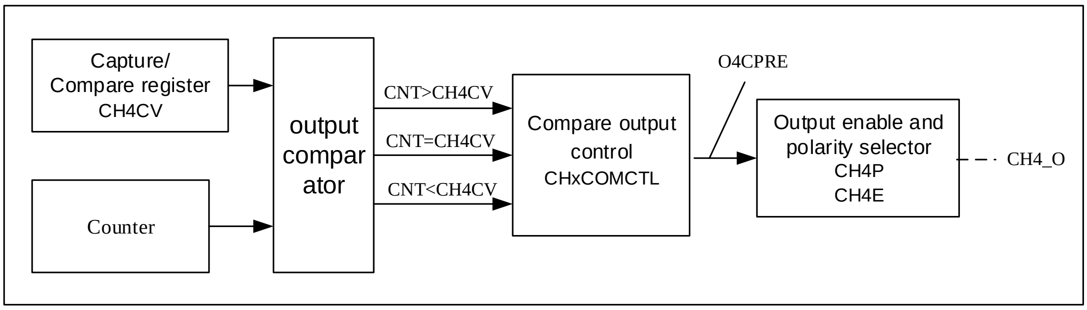


Figure 14-15. Output compare logic (CH4_O)



Note: CH4_0 is an internal signal with no corresponding pin.

Figure 14-13. Output compare logic (with complementary output, x=0,1,2) and

Figure 14-14. Output compare logic (CH3_O) show the logic circuit of output compare mode. The relationship between the channel output signal CHx_O/CHx_ON and the OxCPRE signal (more details refer to ***Channel output prepare signal***) is described as blew: The active level of O0CPRE is high, the output level of CH0_O/CH0_ON depends on OxCPRE signal, CHxP/CHxNP bit and CH0E/CH0NE bit (please refer to the TIMERx_CHCTL2 register for more details). OxCPREC = OxCPRE when CHxCMOCTL[3:0] < 4'b1100, and combined PWM is output when CHxCMOCTL[3:0] = 4'b1100 or 4'b1101, and OxCPREC is the logic AND or OR of OxCPRE and OyCPRE(y = x+1 or x-1). Asymmetric PWM is output when CHxCMOCTL[3:0] = 4'b1110 or 4'b1111, the output of OxCPREC is OxCPRE or OyCPRE(y = x+1 or x-1)when counting up and the output of OxCPREC is OyCPRE or OyCPRE(y = x+1 or x-1). For examples,

1) Configure CHxP=0 (the active level of CHx_O is high, the same as OxCPRE), CHxE=1 (the output of CHx_O is enabled),
   If the output of OxCPRE is active(high) level, the output of CHx_O is active(high) level;
   If the output of OxCPRE is inactive(low) level, the output of CHx_O is active(low) level.

2) Configure CHxNP=0 (the active level of CHx_ON is low, contrary to OxCPRE), CHxNE=1 (the output of CHx_ON is enabled),
   If the output of OxCPRE is active(high) level, the output of CHx_O is active(low) level;
   If the output of OxCPRE is inactive(low) level, the output of CHx_O is active(high) level.

When CH0_O and CH0_ON are output at the same time, the specific outputs of CH0_O and CH0_ON are related to the relevant bits (ROS, IOS, POE and DTCFG bits) in the TIMERx_CCHP register. Please refer to ***Channel output complementary PWM*** for more details.

In output compare mode, the TIMERx can generate timed pulses with programmable position, polarity, duration and frequency. When the counter matches the value in the TIMERx_CHxCV register of an output compare channel, the channel (n) output can be set, cleared, or toggled based on CHxCOMCTL. When the counter reaches the value in the TIMERx_CHxCV register, the CHxIF bit will be set and the channel (n) interrupt is generated if CHxIE = 1. And the DMA request will be asserted, if CxCDE=1.

So, the process can be divided into several steps as below:

Step1: Clock Configuration. Such as clock source, clock prescaler and so on.

Step2: Compare mode configuration.

- Set the shadow enable mode by CHxCOMSEN.
- Set the output mode (set/clear/toggle) by CHxCOMCTL.
- Select the active polarity by CHxP/CHxNP.
- Enable the output by CHxEN.

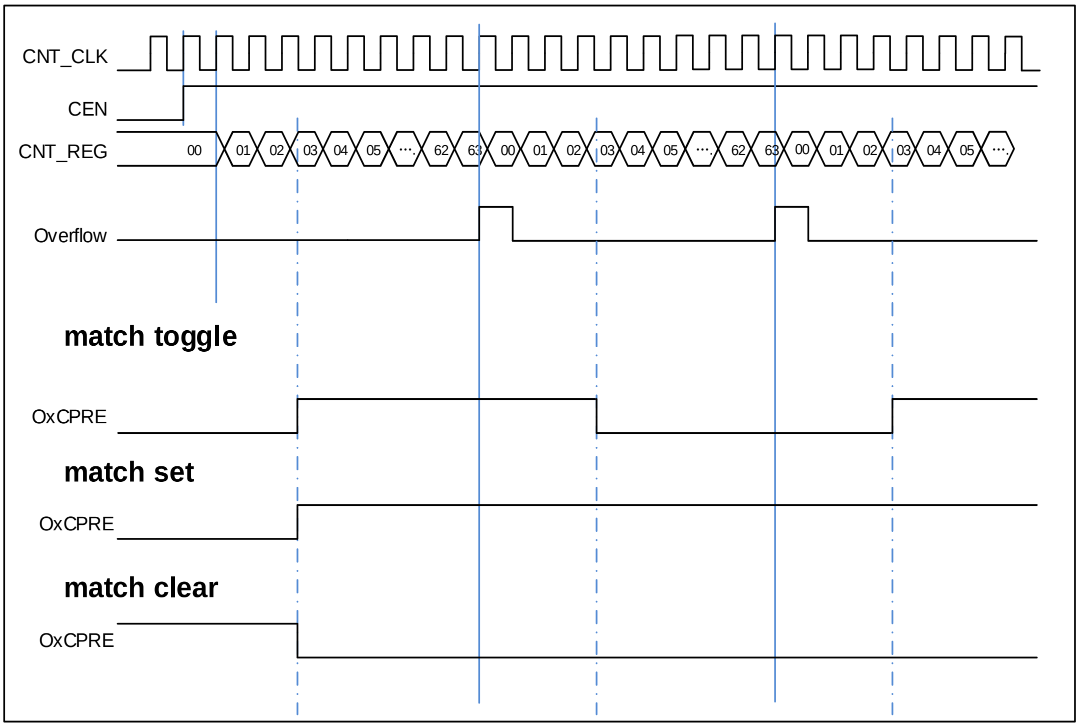
Step3: Interrupt/DMA request enable configuration by CHxIE/CxCDE.

Step4: Compare output timing configuration by TIMERx_CAR and TIMERx_CHxCV.
The TIMERx_CHxCV can be changed onging to meet the expected waveform.

Step5: Start the counter by configuring CEN to 1.

Figure 14-16. Output-compare in three modes shows the three compare modes: toggle/set/clear. CAR=0x63, CHxVAL=0x3.

Figure 14-16. Output-compare in three modes



Output PWM function

In the PWM output mode (by setting the CHxCOMCTL bit to 3'b110 (PWM mode 0) or to 3'b 111(PWM mode 1)), the channel can generate PWM waveform according to the TIMERx_CAR registers and TIMERx_CHxCV registers.

Based on the counter mode, PWM can also be divided into EAPWM (Edge-aligned PWM) and CAPWM (Center-aligned PWM).

The EAPWM's period is determined by TIMERx_CAR and the duty cycle is determined by TIMERx_CHxCV. *Figure 14-17. Timing chart of EAPWM* shows the EAPWM output and interrupts waveform.

The CAPWM's period is determined by 2*TIMERx_CAR, and the duty cycle is determined by 2*TIMERx_CHxCV. *Figure 14-18. Timing chart of CAPWM* shows the CAPWM output and interrupts waveform.

In up counting mode, if the value of TIMERx_CHxCV is greater than the value of TIMERx_CAR, the output will be always inactive in PWM mode 0 (CHxCOMCTL=3'b110). And if the value of TIMERx_CHxCV is greater than the value of TIMERx_CAR, the output will be always active in PWM mode 1 (CHxCOMCTL=3'b111).

Figure 14-17. Timing chart of EAPWM

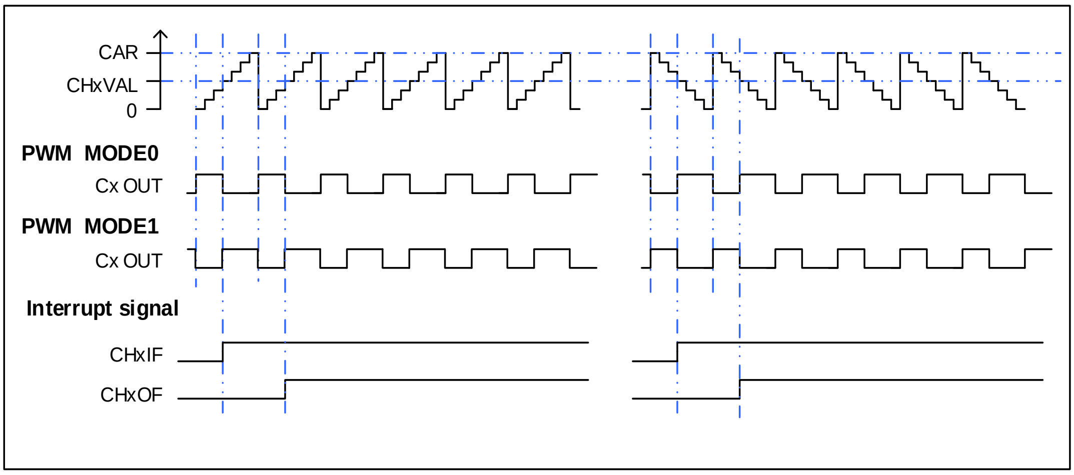
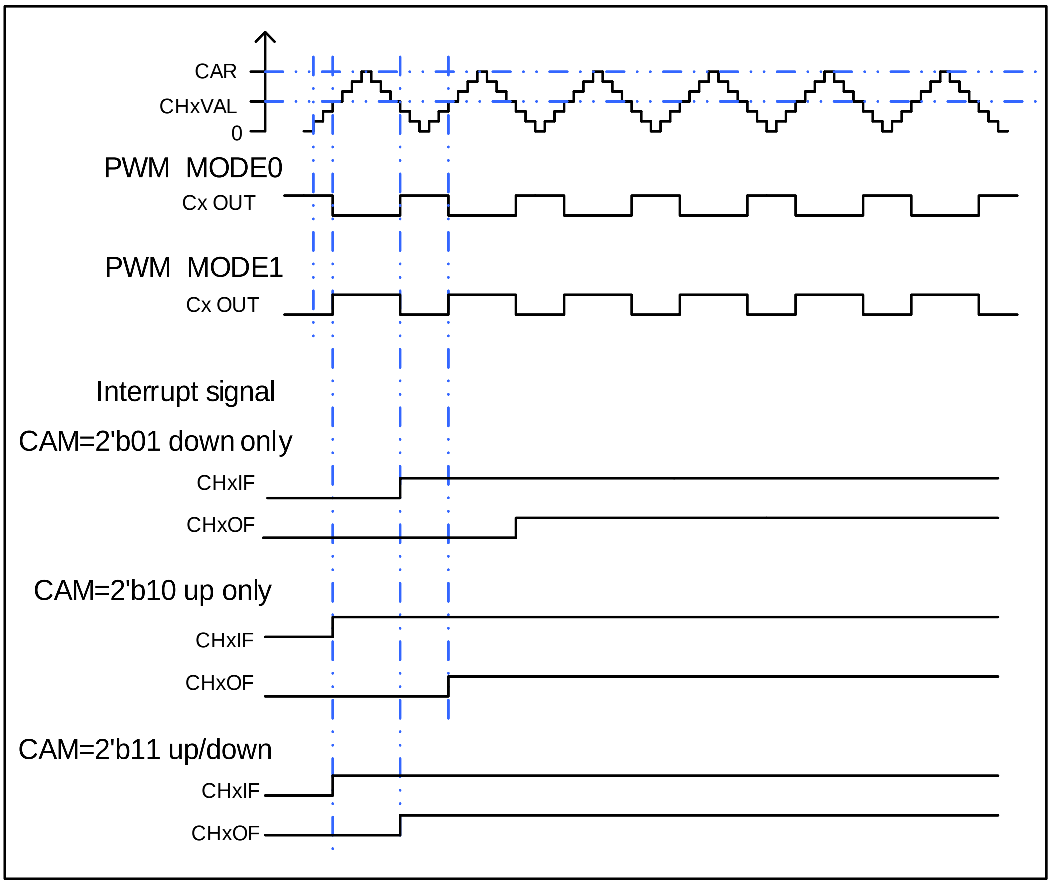


Figure 14-18. Timing chart of CAPWM



Asymmetric PWM mode

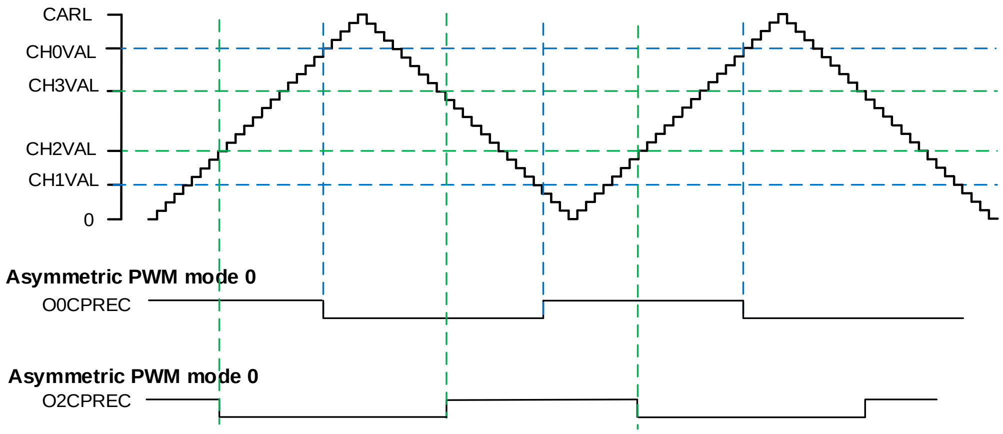
The asymmetric PWM mode 0 / 1 are (by setting the CHxCOMCTL[3:0] bit-field to 4'b1110 or 4'b1111) used in center-aligned PWM to generate a programmable phase shift. The CPWM's frequency is determined by TIMERx_CAR register, and the duty cycle and phase shift are determined by a pair of TIMER_CHxCV / TIMER_CH(x+1)CV (or TIMER_CH(x-1)CV) registers with adjacent offset addresses.

The TIMER_CHxCV register determines the waveform when counting up, and the TIMER_CH(x+1)CV (or TIMER_CH(x-1)CV) register determines the waveform when counting down. The details are as follows:

- The CPWM of O0CPREC / O1CPREC is determined by TIMER_CH0CV and TIMER_CH1CV registers.
- The CPWM of O2CPREC / O3CPREC is determined by TIMER_CH2CV and TIMER_CH3CV registers.

When using asymmetric PWM modes, CH0 / CH1 (or CH2 / CH3) can independently output different waveforms, which are configured by the CHxCOMCTL[3:0] bit-field (the two channels can be configured with different values).

Figure 14-19. O0CPREC and O2CPREC with asymmetric PWM mode



Combined PWM mode

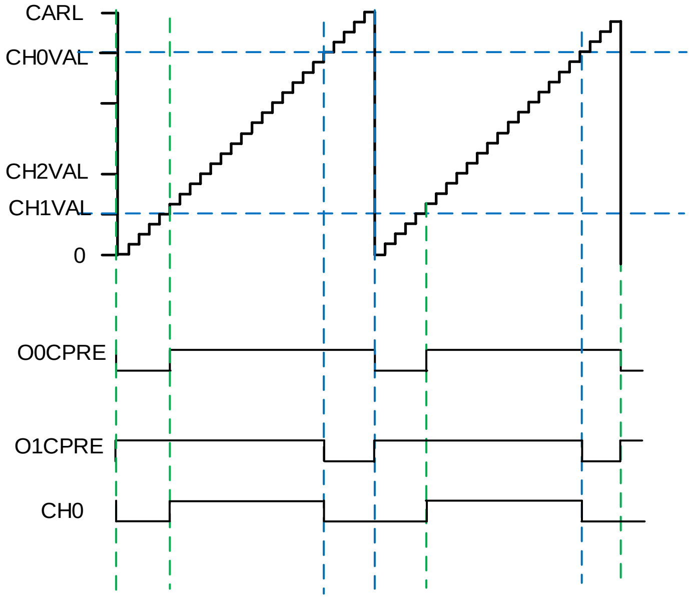
The combined PWM mode 0 / 1 are (by setting the CHxCOMCTL[3:0] bit-field to 4’b1100 or 4’b1101) used in center-aligned PWM to generate a programmable phase shift. The CPWM’s frequency is determined by TIMERx_CAR register, and the duty cycle and phase shift are determined by a pair of TIMER_CHxCV / TIMER_CH(x+1)CV (or TIMER_CH(x-1)CV) registers with adjacent offset addresses.

- The CPWM of O0CPREC / O1CPREC is determined by TIMER_CH0CV and TIMER_CH1CV registers.
- The CPWM of O2CPREC / O3CPREC is determined by TIMER_CH2CV and TIMER_CH3CV registers.

When O0CPREC selects the combined PWM mode 0（CH0COMCTL[3:0] == 4'b1100）, and O0CPREC output signal which is the logic OR of O0CPRE and O1CPRE. When O0CPREC selects the combined PWM mode 1（CH0COMCTL[3:0] == 4'b1101）, and O0CPREC output signal which is the logic AND of O0CPRE and O1CPRE.

When using combined PWM modes, CH0 / CH1 (or CH2 / CH3) can independently output different waveforms, which are configured by the CHxCOMCTL[3:0] bit-field (the two channels can be configured with different values).

Figure 14-20. O0CPRE is combined PWM mode 1 and O1CPRE is PWM mode 0

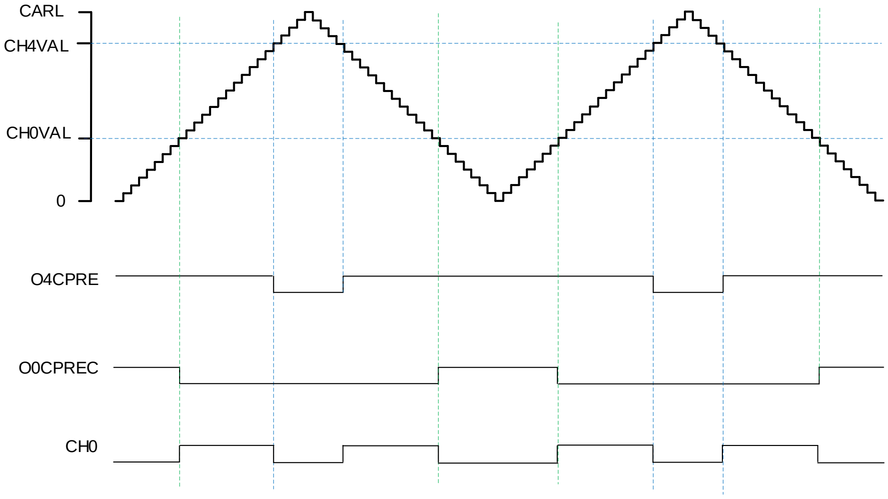


Combined 3-phase PWM mode

The combined 3-phase PWM mode allows the generation of 1-3 PWM waveforms combined with O4CPRE. As the combined signal, O4CPRE performs a logical AND operation with O0CPREC, O1CPREC, and O2CPREC to output the corresponding waveform.

- O0CPREC is controlled by TIMER_CH0CV and TIMER_CH4CV when CCH4CH0 is set.
- O1CPREC is controlled by TIMER_CH1CV and TIMER_CH4CV when CCH4CH1 is set.
- O2CPREC is controlled by TIMER_CH2CV and TIMER_CH4CV when CCH4CH2 is set

Figure 14-21. O0CPREC is combined PWM 3-phase mode and CCH4CH0 is set



Retriggerable single pulse mode

This mode enables the counter to generate a pulse of programmable length in response to a trigger.

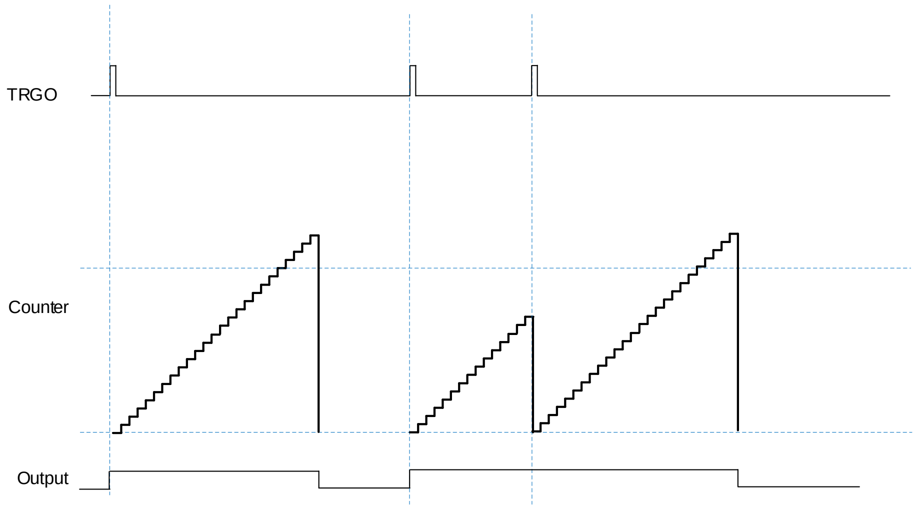
- The pulse begins immediately upon receiving a trigger (no delay).
- If a new trigger is received before the current pulse ends, the pulse duration is extended.

The timer must operate in Slave mode, and set TSCFG7[2:0] in SYSCFG_TIMERxCFG0(x = 0), setting the CHxCOMCTL[3:0] bit-field to 4'b1100 or 4'b1101 to chose Retriggerable single pulse mode 0/1.

In up-counting mode, set TIMERx_CHyCV to 0; the pulse length is defined by the TIMERx_CAR register. in down counting mode, ensure TIMERx_CHyCV is great than or equal to TIMERx_CAR.

Note : This mode is incompatible with center-aligned PWM modes. Set CAM[1:0] = 2'b00 in TIMERx_CTL0.

Figure 14-22. Retriggerable single pulse mode



Channel output prepare signal

As is shown in ***Figure 14-13. Output compare logic (with complementary output, x=0,1,2)***, when TIMERx is configured in compare match output mode,a middle signal which is OxCPRE signal (Channel x output prepare signal) will be generated before the channel outputs signal. The OxCPRE signal type is defined by configuring the CHxCOMCTL bit. The OxCPRE signal has several types of output function. These include keeping the original level by configuring the CHxCOMCTL field to 0x00, setting to high by configuring the CHxCOMCTL field to 0x01, setting to low by configuring the CHxCOMCTL field to 0x02 or toggling signal by configuring the CHxCOMCTL field to 0x03 when the counter value matches the content of the TIMERx_CHxCV register.

The PWM mode 0/PWM mode 1 output is another output type of OxCPRE which is setup by configuring the CHxCOMCTL field to 0x06/0x07. In these modes, the OxCPRE signal level is changed according to the counting direction and the relationship between the counter value and the TIMERx_CHxCV content. Refer to the definition of relative bit for more details.

Another special function of the OxCPRE signal is a forced output which can be achieved by configuring the CHxCOMCTL field to 0x04/0x05. The output can be forced to an inactive/active level irrespective of the comparison condition between the values of the counter and the TIMERx_CHxCV.

Configure the CHxCOMCEN bit to 1 in the TIMERx_CHCTL0 register, the OxCPRE signal can be forced to 0 when the ETIFP signal derived from the external ETI pin is set to a high level. The OxCPRE signal will not return to its active level until the next update event occurs.

Channel output complementary PWM

Function of complementary is for a pair of channels, CHx_O and CHx_ON, the two output signals cannot be active at the same time. The TIMERx has 5 channels, but only the first three channels have this function. The complementary signals CHx_O and CHx_ON are controlled by a group of parameters: the CHxEN and CHxNEN bits in the TIMERx_CHCTL2 register, the POEN, ROS and IOS bits in the TIMERx_CCHP register, ISOx and ISOxN bits in the TIMERx_CTL1 register. The output polarity is determined by CHxP and CHxNP bits in the TIMERx_CHCTL2 register.

Table 14-2. Complementary outputs controlled by parameters

Complementary Parameters					Output Status	
POEN	**ROS**	**IOS**	**CHxEN**	**CHxNEN**	**CHx_O**	**CHx_ON**
0	0/1	0	0	0	CHx_O / CHx_ON = LOW CHx_O / CHx_ON output disable[1].	
				1	CHx_O/ CHx_ON output “off-state” [2]: the CHx_O/ CHx_ON output inactive level firstly: CHx_O = CHxP, CHx_ON = CHxNP; If the clock for deadtime generator is present, after a deadtime: CHx_O = ISOx, CHx_ON = ISOxN. [3]	
			1	0		
				1		
		1	x	x	CHx_O/ CHx_ON output “off-state”: the CHx_O/ CHx_ON output inactive level firstly: CHx_O = CHxP, CHx_ON = CHxNP; If the clock for deadtime generator is present, after a deadtime: CHx_O = ISOx, CHx_ON = ISOxN.	
1	0	0/1	0	0	CHx_O/CHx_ON = LOW CHx_O/CHx_ON output disable.	
				1	CHx_O = LOW CHx_O output disable.	CHx_ON =OxCPRE⊕ [4]CHxNP CHx_ON output enable.
			1	0	CHx_O=OxCPRE⊕CHxP CHx_O output enable.	CHx_ON = LOW CHx_ON output disable.
				1	CHx_O=OxCPRE⊕CHxP CHx_O output enable.	CHx_ON =(!OxCPRE)[5]⊕ CHxNP. CHx_ON output enable.
	1		0	0	CHx_O = CHxP CHx_O output “off-state”.	CHx_ON = CHxNP CHx_ON output “off-state”.
				1	CHx_O = CHxP CHx_O output “off-state”	CHx_ON =OxCPRE⊕CHxNP CHx_ON output enable
			1	0	CHx_O=OxCPRE⊕CHxP CHx_O output enable	CHx_ON = CHxNP CHx_ON output “off-state”.
				1	CHx_O=OxCPRE⊕CHxP CHx_O output enable	CHx_ON =(!OxCPRE)⊕ CHxNP CHx_ON output enable.

Note:

(1) output disable: the CHx_O / CHx_ON are disconnected to corresponding pins, the pin is floating with GPIO pull up/down setting which will be Hi-Z if no pull.

(2) “off-state”: CHx_O / CHx_ON output with inactive state (e.g., CHx_O = 0⊕CHxP = CHxP).

(3) See Break mode section for more details.

(4) ⊕: Xor calculate.

(5) (!OxCPRE)：the complementary output of the OxCPRE signal.

Insertion dead time for complementary PWM

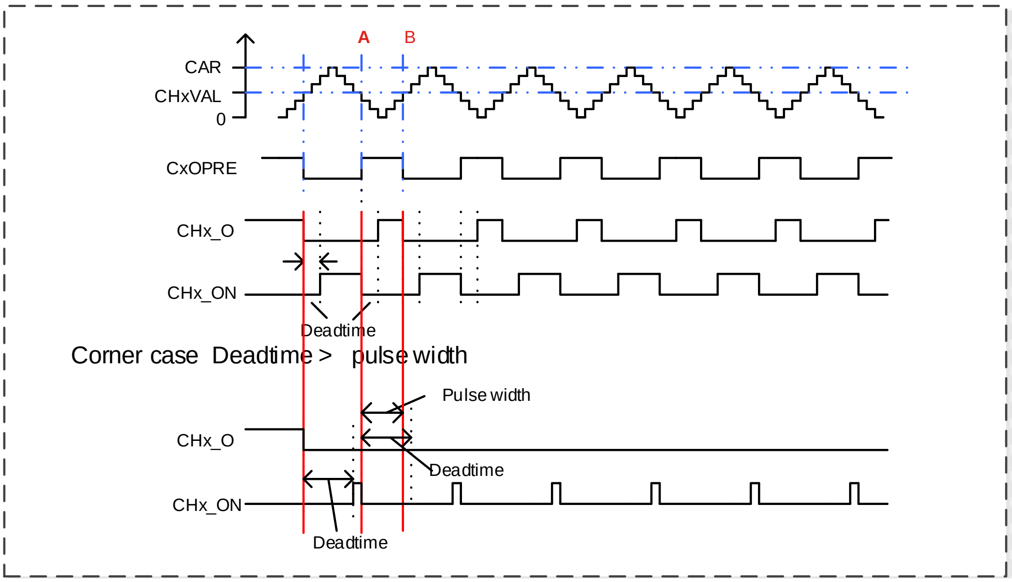
The dead time insertion is enabled when both CHxEN and CHxNEN are configured to 1'b1, it is also necessary to configure POEN to 1. The field named DTCFG defines the dead time delay that can be used for all channels except channel 3. Refer to the TIMERx_CCHP register for details about the delay time.

The dead time delay insertion ensures that two complementary signals are not active at the same time.

When the channelx match event (TIMERx counter = CHxVAL) occurs, OxCPRE will be toggled in PWM mode 0. At point A in ***Figure 14-23. Complementary output with dead time insertion***, CHx_O signal remains at the low level until the end of the dead time delay, while CHx_ON signal will be cleared at once. Similarly, at point B when the channelx match event (TIMERx counter = CHxVAL) occurs again, OxCPRE is cleared, and CHx_O signal will be cleared at once, while CHx_ON signal remains at the low level until the end of the dead time delay.

Sometimes, we can see corner cases about the dead time insertion. For example: the dead time delay is greater than or equal to the duty cycle of the CHx_O signal, then the CHx_O signal is always inactive (As shown in *Figure 14-23. Complementary output with dead time insertion*).

Figure 14-23. Complementary output with dead time insertion



Break function

The advanced timers have two kinds of break function: BREAK0 and BREAK1. The break functions can be enabled by setting the BRK0EN and BRK1EN bits in the TIMERx_CCHP register. The break input polarities are configured by the BRK0P and BRK1P bits in TIMERx_CCHP register, the inputs are active on level.

In this function, CHx_O and CHx_ON are controlled by the POEN, IOS and ROS bits in the

TIMERx_CCHP register, ISOx and ISOxN bits in the TIMERx_CTL1 register. The break event is the result of logic ORed of all sources. The break functions can handle three types of event sources:

- External sources: coming from BRKINx (x=0..1) inputs;
- System sources: the Cortex® -M23+ LOCKUP output, the SRAM parity error signal, a clock failure event generated by the CSS detector
- Break events can also be generated by software using BRK0G or BRK1G bits in the TIMERx_SWEVG register.

The signal for BREAK1 originates from an external source linked to one of the BRKIN pins, as configured in the GPIO alternate function registers. This signal is subject to polarity selection and can optionally undergo digital filtering. Additionally, break events can be triggered via software by utilizing the BRK0G and BRK1G bits within the TIMx_SWEVG register. Software-based break generation using BRK0G and BRK1G remains effective regardless of the states of the BRK0EN and BRK1EN enable bits.

Figure 14-24. BREAK0 function logic diagram

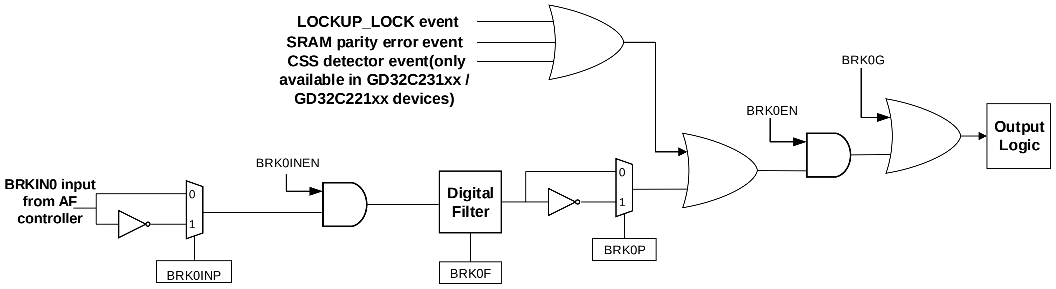
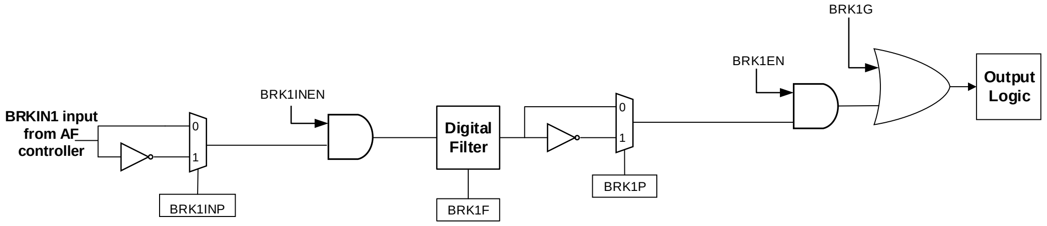


Figure 14-25. BREAK1 function logic diagram

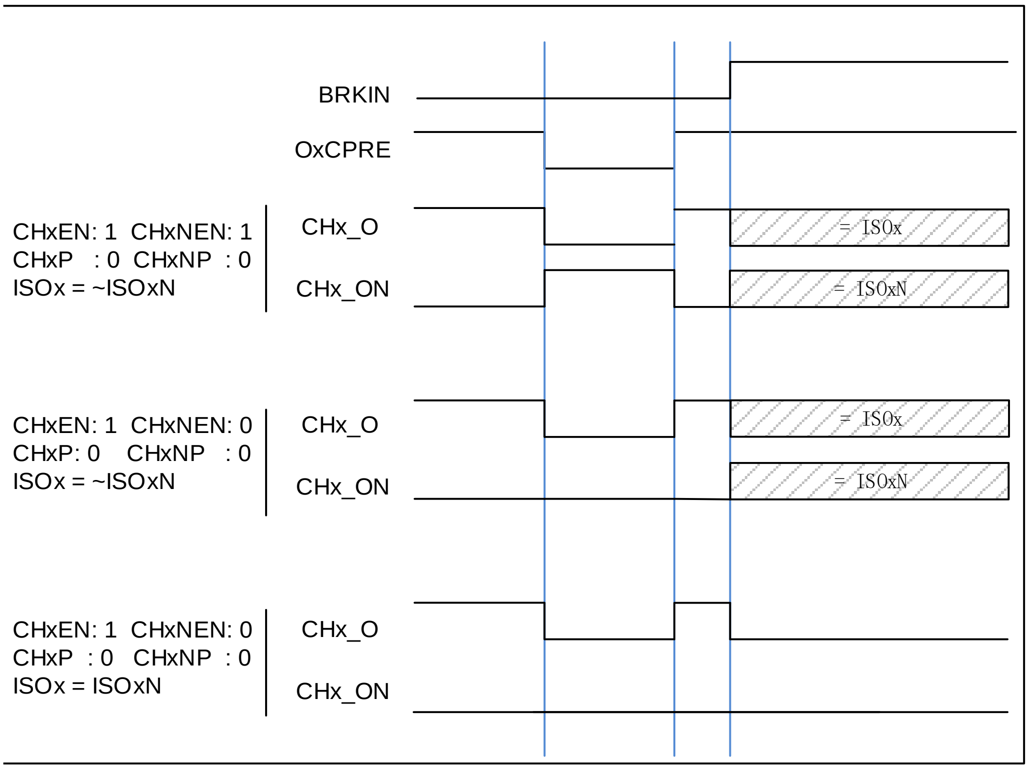


BREAK0 can be used to handle the faults of system sources and external sources. When a BREAK0 event occurs, the outputs is force at a inactive level, or at a predefined level (either active or inactive) after a deadtime duration. BREAK1 only can be used to handle the faults of on-chip peripheral events and external sources. When a BREAK1 event occurs, the outputs is force at a inactive level, the BREAK0 has a higher priority than BREAK1 input, BREAK1 must only be used with ROS = IOS = 1.

When a break occurs, the POEN bit is cleared asynchronously, the output CHx_O and CHx_ON are driven with the level programmed in the ISOx bit and ISOxN in the TIMERx_CTL1 register as soon as POEN is 0. If IOS is 0 then the timer releases the enable output else the enable output remains high. The complementary outputs are first put in reset state, and then the dead-time generator is reactivated in order to drive the outputs with the level programmed in the ISOx and ISOxN bits after a dead-time.

When a break occurs, the BRK0IF/BRK1IF bit in the TIMERx_INTF register is set. If BRK0IE/BRK1IE is 1, an interrupt generated

Figure 14-26. Output behavior of the channel in response to a break (the break high active)



Quadrature decoder

The quadrature decoder function uses two quadrature inputs CI0 and CI1 derived from the TIMERx_CH0 and TIMERx_CH1 pins respectively to interact with each other to generate the counter value. Setting TSCFGy[2:0] != 3'b000 (y=0,1,2) to select that the counting direction of timer is determined only by the CI0, only by the CI1, or by the CI0 and the CI1. The DIR bit is modified during the voltage level change of each direction selection source. The mechanism of changing the counter direction is shown in ***Table 14-3. Counting direction versus quadrature decoder signals.*** The quadrature decoder can be regarded as an external clock with a direction selection. This means that the counter counts continuously from 0 to the counter-reload value. Therefore, users must configure the TIMERx_CAR register before the counter starts to count.

Table 14-3. Counting direction versus quadrature decoder signals

Counting mode	Level	CI0FE0		CI1FE1	
		Rising	Falling	Rising	Falling
CI0 only counting	CI1FE1=High	Down	Up	-	-
	CI1FE1=Low	Up	Down	-	-
CI1 only counting	CI0FE0=High	-	-	Up	Down
	CI0FE0=Low	-	-	Down	Up

Counting mode	Level	CI0FE0		CI1FE1	
		Rising	Falling	Rising	Falling
CI0 and CI1 counting	CI1FE1=High	Down	Up	X	X
	CI1FE1=Low	Up	Down	X	X
	CI0FE0=High	X	X	Up	Down
	CI0FE0=Low	X	X	Down	Up

Note: "-" means "no counting"; "X" means impossible.

Figure 14-27. Example of counter operation in quadrature decoder interface mode

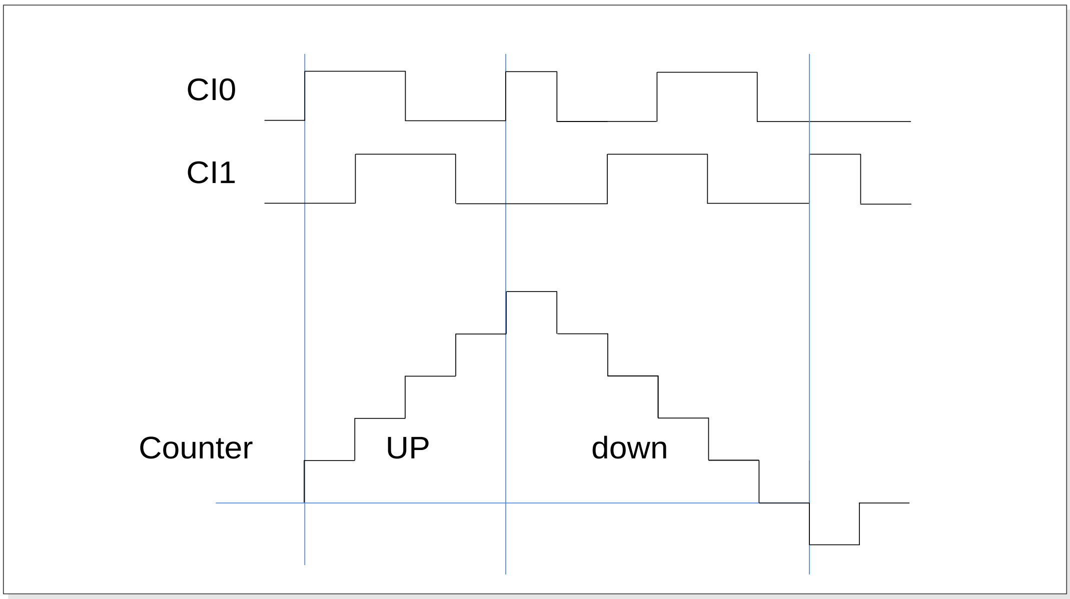
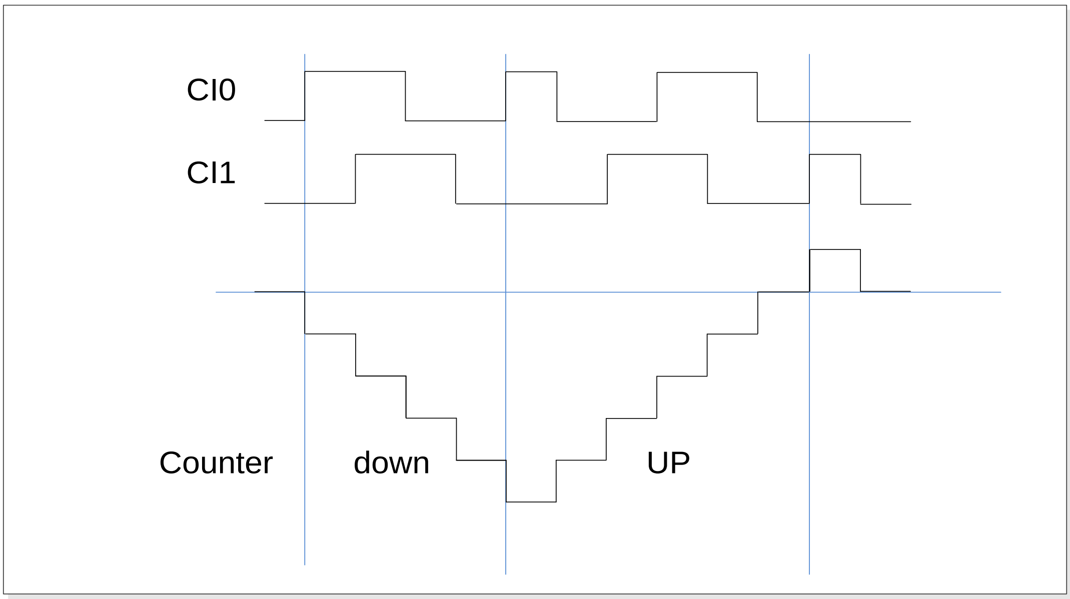


Figure 14-28. Example of quadrature decoder interface mode with CI0FE0 polarity inverted



Hall sensor function

Hall sensor is generally used to control BLDC Motor; advanced timer can support this function.

Figure 14-29. Hall sensor is used to BLDC motor show how to connect. And we can see we need two timers. First TIMER_in(Advanced/GeneralL0 TIMER) should accept three Rotor Position signals from Motor.

Each of the 3 sensors provides a pulse that applied to an input capture pin, can then be analyzed and both speed and position can be deduced.

By using TSCFGy[2:0] in SYSCFG_TIMERxCFG register, TIMER_in and TIMER_out can be connected. TIMER_out will generate PWM signal to control BLDC motor's speed based on the ITRx. Then, the feedback circuit is finished, also you change configuration to fit your request.

About the TIMER_in, it need have input XOR function, so you can choose from Advanced/GeneralL0 TIMER.

And TIMER_out need have functions of complementary and Dead-time, so only advanced timer can be chosen. Else, based on the timers' internal connection relationship, pair's timers can be selected.

TIMER_in (TIMER2) -> TIMER_out (TIMER0 ITI2)

And so on.

After getting appropriate timers combination, and wire connection, we need to configure timers. Some key settings include:

- Enable XOR by setting TI0S, then, each of input signal change will make the CI0 toggle. CH0VAL will record the value of counter at that moment.
- Enable ITIx connected to commutation function directly by setting CCUC and CCSE.
- Configuration PWM parameter based on your request.

Figure 14-29. Hall sensor is used to BLDC motor

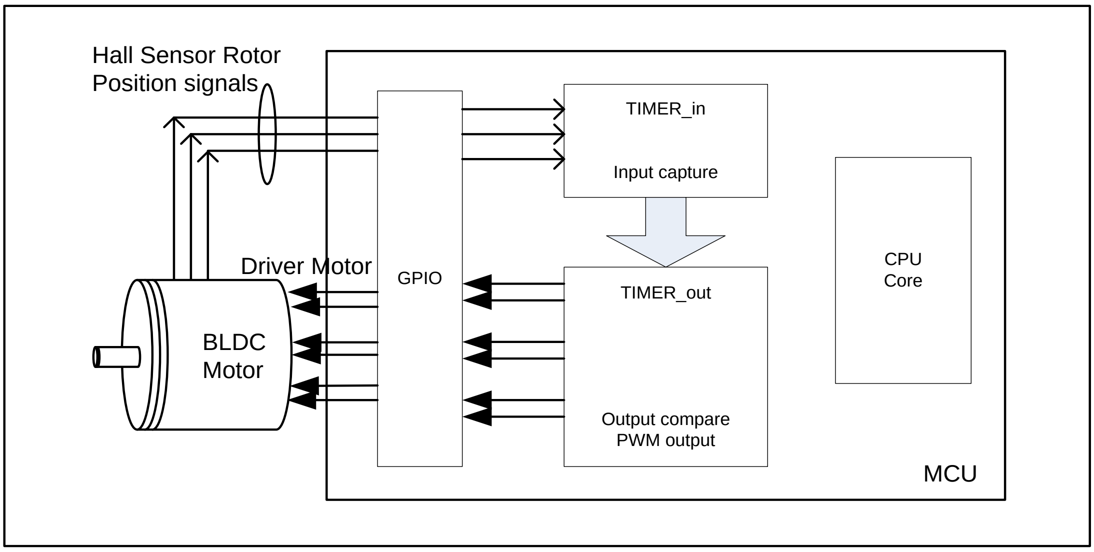
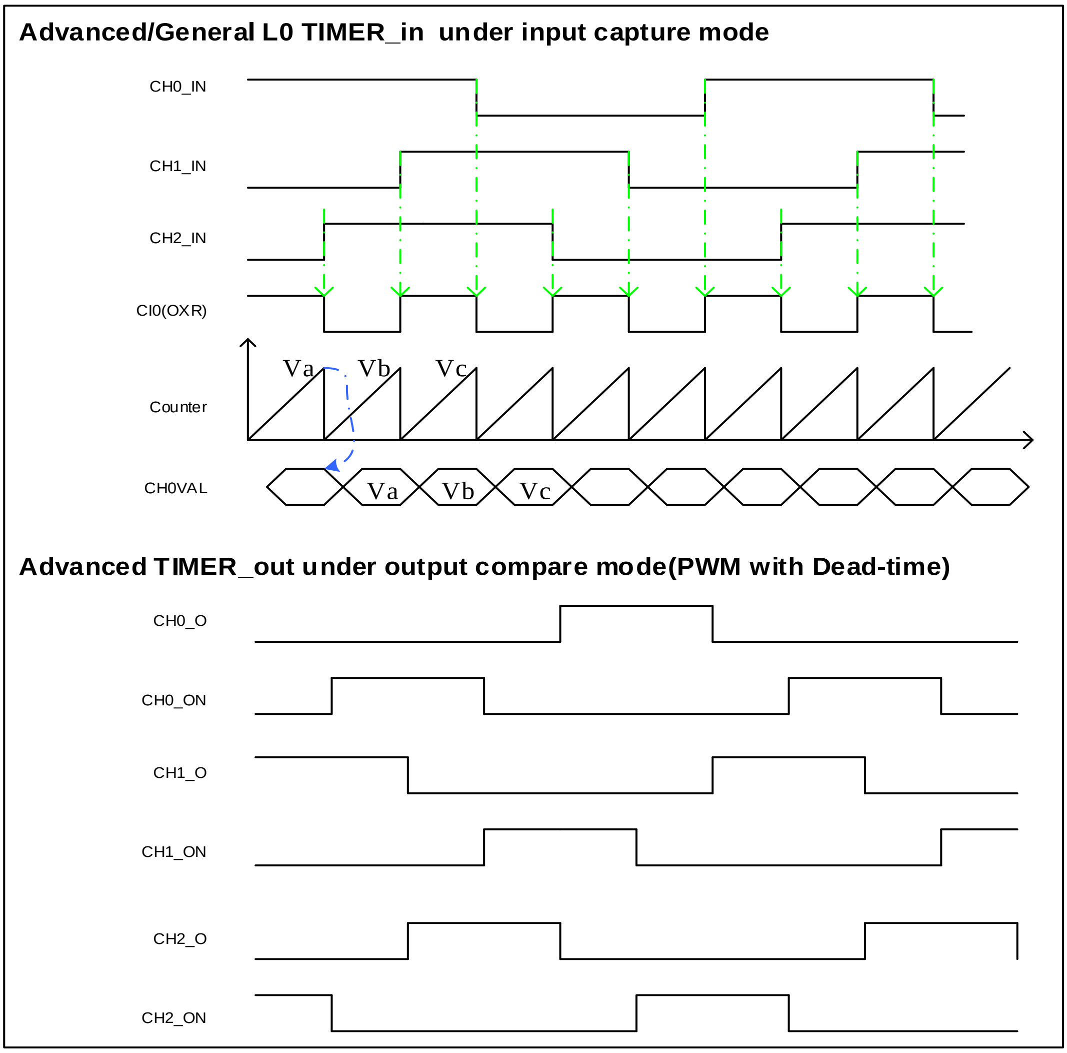


Figure 14-30. Hall sensor timing between two timers


Advanced/General L0 TIMER_in under input capture mode
CH0_IN
CH1_IN
CH2_IN
CI0(OXR)
Counter
Va
Vb
Vc
CH0VAL
Va
Vb
Vc
Advanced TIMER_out under output compare mode(PWM with Dead-time)
CH0_O
CH0_ON
CH1_O
CH1_ON
CH2_O
CH2_ON

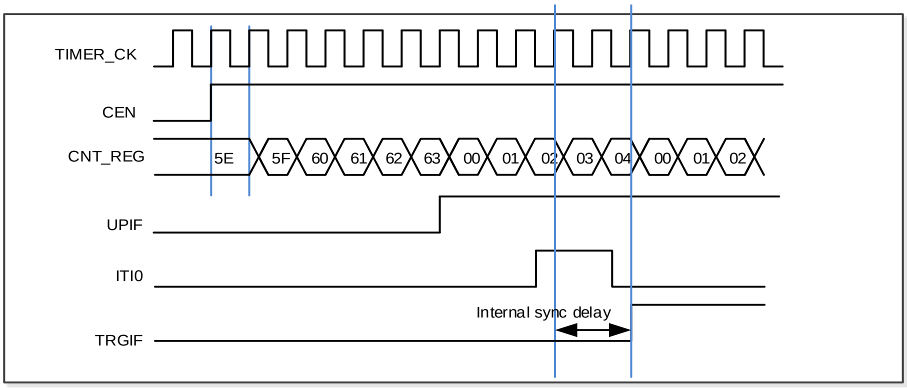
Master-slave management

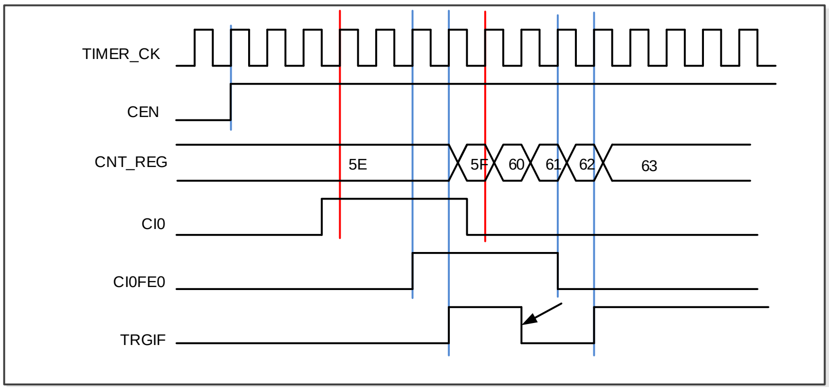
The TIMERx can be synchronized with a trigger in several modes including restart mode, pause mode and event mode which is selected by the TSCFGy[2:0] in SYSCFG_TIMER0CFG(x=3,4,5).

Table 14-4. Examples of slave mode

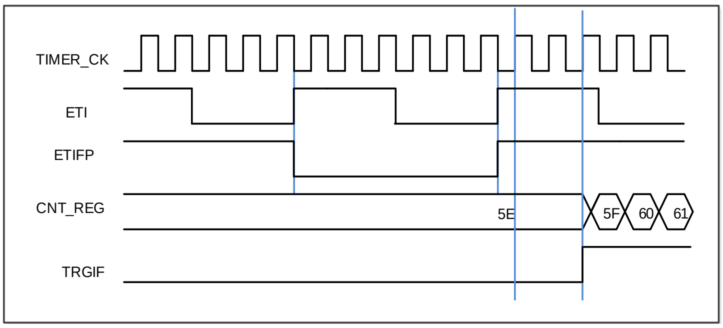
	Mode Selection	Source Selection	Polarity Selection	Filter and Prescaler

LIST	TSCFGy[2:0]
	y=3 (restart mode)
	y=4 (pause mode)
	y=5 (event mode)
		TSCFGy[2:0]
		001: ITI0
		010: ITI2
		011: ITI3
		100: CI0F_ED
		101: CI0FE0
		110: CI1FE1
		111: ETIFP
			If CI0FE0 or CI1FE1 is selected as the trigger source, configure the CHxP and CHxNP for the polarity selection and inversion.
			If ETIFP is selected as the trigger source, configure the ETP for polarity selection and inversion.
				For the ITIx, no filter and prescaler can be used.
				For the CIx, filter can be used by configuring CHxCAPFLT, no prescaler can be used.
				For the ETIFP, filter can be used by configuring ETFC and prescaler can be used by configuring ETPSC.

Exam1	**Restart mode** The counter will be cleared and restart when a rising edge of trigger input comes.	TSCFG3[2:0] = 3'b 001.ITI0 is selected.	For ITI0, no polarity selector can be used.	For the ITI0, no filter and prescaler can be used.
	Figure 14-31. Restart mode  TIMER_CK CEN CNT_REG 5E 5F 60 61 62 63 00 01 02 03 04 00 01 02 UPIF ITI0 Internal sync delay TRGIF			
Exam2	**Pause mode** The counter will be paused when the trigger input is low, and it will start when the trigger input is high.	TSCFG4[2:0] = 3'b 101 CI0FE0 is selected.	TI0S=0 (Non-xor) [CH0NP=0, CH0P=0] CI0FE0 does not invert. The capture event will occur on the rising edge only.	Filter is bypassed in this example.

	Mode Selection	Source Selection	Polarity Selection	Filter and Prescaler
	Figure 14-32. Pause mode  TIMER_CK CEN CNT_REG 5E 5F 60 61 62 63 CI0 CI0FE0 TRGIF			

Exam3	**Event mode**
	The counter will start to count when a rising edge of trigger input comes.
		TSCFG5[2:0] = 3'b 111
		ETIFP is selected.
			ETP = 0, the polarity of ETI does not change.	ETPSC = 1, ETI is divided by 2.
				ETFC = 0, ETI does not filter.

	Figure 14-33. Event mode  TIMER_CK ETI ETIFP CNT_REG 5E 5F 60 61 TRGIF			

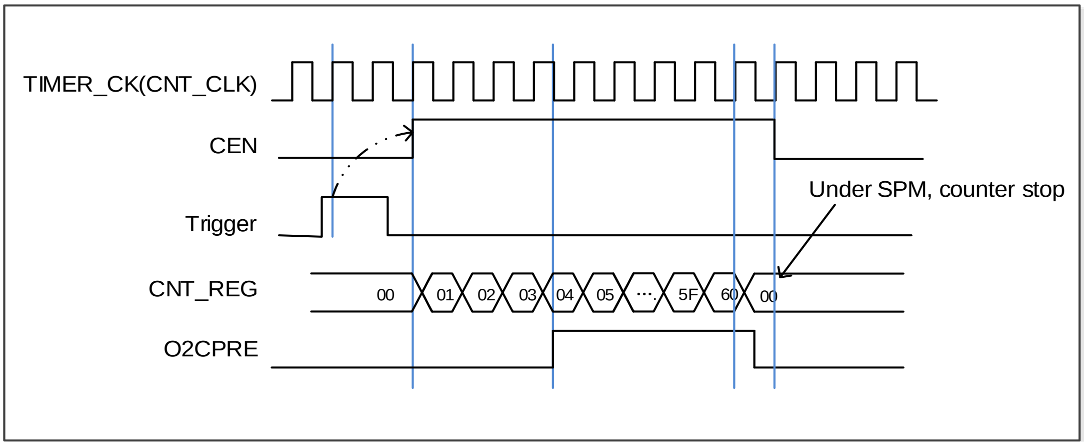
Single pulse mode

Single pulse mode is enabled by setting SPM in TIMERx_CTL0. If SPM is set, the counter will be cleared and stopped when the next update event occurs. In order to get a pulse waveform, the TIMERx is configured to PWM mode or compare mode by CHxCOMCTL.

Once the timer is set to the single pulse mode, it is not necessary to configure the timer enable bit CEN in the TIMERx_CTL0 register to 1 to enable the counter. Setting the CEN bit to 1 or a trigger signal edge can generate a pulse and then keep the CEN bit at a high state until the update event occurs or the CEN bit is written to 0 by software. If the CEN bit is cleared to 0 by software, the counter will be stopped and its value will be held.

In the single pulse mode, the active edge of trigger which sets the CEN bit to 1 will enable the counter. However, there exists several clock delays to perform the comparison result between the counter value and the TIMERx_CHxCV value. In order to reduce the delay to a minimum value, the user can set the CHxCOMFEN bit in TIMERx_CHCTL0/1 register. After a trigger rising occurs in the single pulse mode, the OxCPRE signal will immediately be forced to the state which the OxCPRE signal will change to, as the compare match event occurs without taking the comparison result into account. The CHxCOMFEN bit is available only when the output channel is configured to the PWM mode 0 or PWM mode 1 and the trigger source is derived from the trigger signal.

Figure 14-34. Single pulse mode TIMERx_CHxCV=0x04, TIMERx_CAR=0x60



Timers 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.

- TIMER2 as prescaler for TIMER0

1. Configure TIMER2 in master mode and select its Update Event (UPE) as trigger output (MMC=010 in the TIMER2_CTL1 register). Then TIMER2 drives a periodic signal on each counter overflow.
2. Configure the TIMER2 period (TIMER2_CAR registers).
3. Configure TIMER0 in external clock mode 1 and select the TIMER0 input trigger source from TIMER2 (TSCFG6[2:0] = 010 in the_SYSCFG_TIMERxCFG register).
4. Start TIMER0 by writing ‘1 in the CEN bit (TIMER0_CTL0 register).Start TIMER2 by writing ‘1 in the CEN bit (TIMER2_CTL0 register).

- Start TIMER0 with TIMER2’s Enable/Update signal

First, we enable TIMER0 with the enable out of TIMER2. Refer to ***Figure 14-35. Triggering TIMER0 with Enable of TIMER2*** TIMER0 starts counting from its current value on the divided internal clock after trigger by TIMER2 enable output.

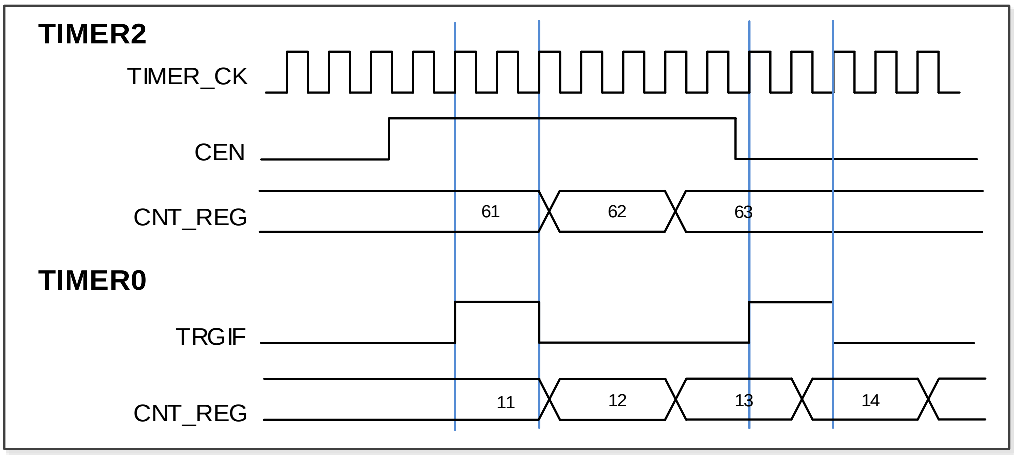
When TIMER0 receives the trigger signal its CEN bit is automatically set and the counter counts until we disable TIMER0. Both counter clock frequencies are divided by 3 by the prescaler compared to TIMER_CK ($f_{CNT\_CLK} = f_{TIMER\_CK}/3$). Do as follow:

1. Configure TIMER2 master mode to send its enable signal as trigger output, and configure

TIMER0 to select the input trigger from TIMER2 (TSCFG5[2:0] = 010 in the SYSCFG_TIMERxCFG register).

2. Start TIMER2 by writing 1 in the CEN bit (TIMER2_CTL0 register).

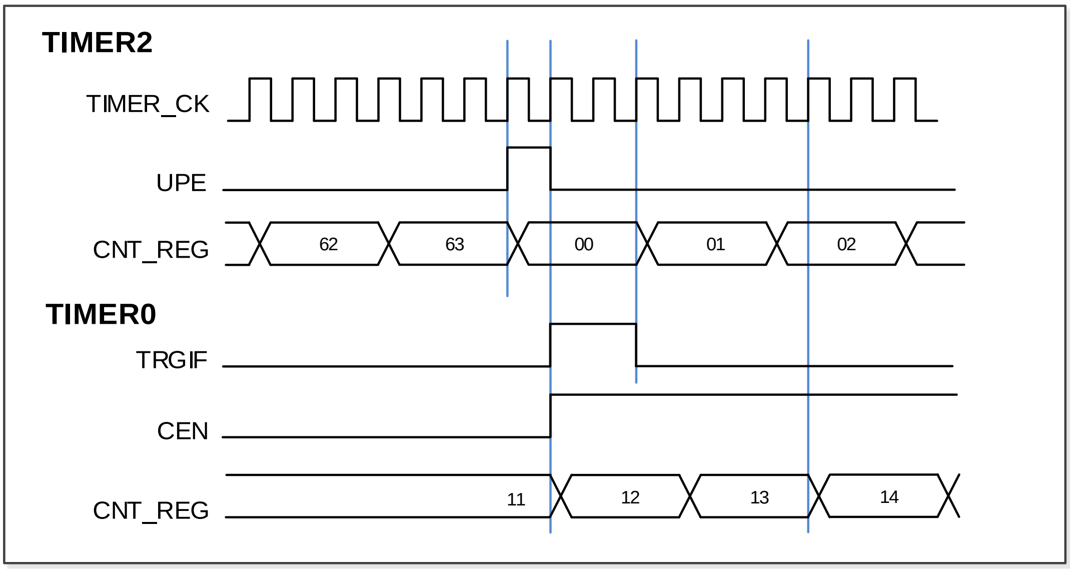
Figure 14-35. Triggering TIMER0 with Enable of TIMER2



In this example, we also can use update Event as trigger source instead of enable signal. Refer to *Figure 14-36. Triggering TIMER0 with update signal of TIMER2*. Do as follow:

1. Configure TIMER2 in master mode and send its Update Event (UPE) as trigger output (MMC=3'b010 in the TIMER2_CTL1 register).
2. Configure the TIMER2 period (TIMER2_CAR registers).
3. Configure TIMER0 to get the input trigger from TIMER2, configure TIMER0 in event mode. (TSCFG5[2:0] = 010 in the_ SYSCFG_TIMERxCFG register).
4. Start TIMER2 by writing '1 in the CEN bit (TIMER2_CTL0 register).

Figure 14-36. Triggering TIMER0 with update signal of TIMER2

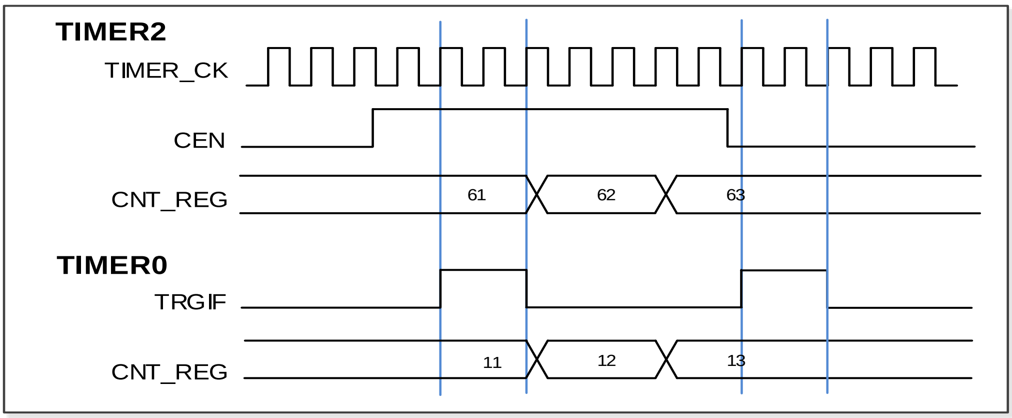


- Enable TIMER0 count with TIMER2's enable/O0CPRE signal

In this example, we control the enable of TIMER0 with the enable output of TIMER2 .Refer to *Figure 14-37. Pause TIMER0 with enable of TIMER2* TIMER0 counts on the divided internal clock only when TIMER2 is enable. Both counter clock frequencies are divided by 3 by the prescaler compared to TIMER_CK (f_{CNT_CLK} = f_{TIMER_CK} /3). Do as follow:

1. Configure TIMER2 input master mode and Output enable signal as trigger output (MMC=001 in the TIMER2_CTL1 register).
2. Configure TIMER0 to get the input trigger from TIMER2, configure TIMER0 in pause mode (TSCFG5[2:0] = 010 in the_ SYSCFG_TIMERxCFG register).
3. Enable TIMER0 by writing '1 in the CEN bit (TIMER0_CTL0 register)
4. Start TIMER2 by writing '1 in the CEN bit (TIMER2_CTL0 register).
5. Stop TIMER2 by writing '0 in the CEN bit (TIMER2_CTL0 register).

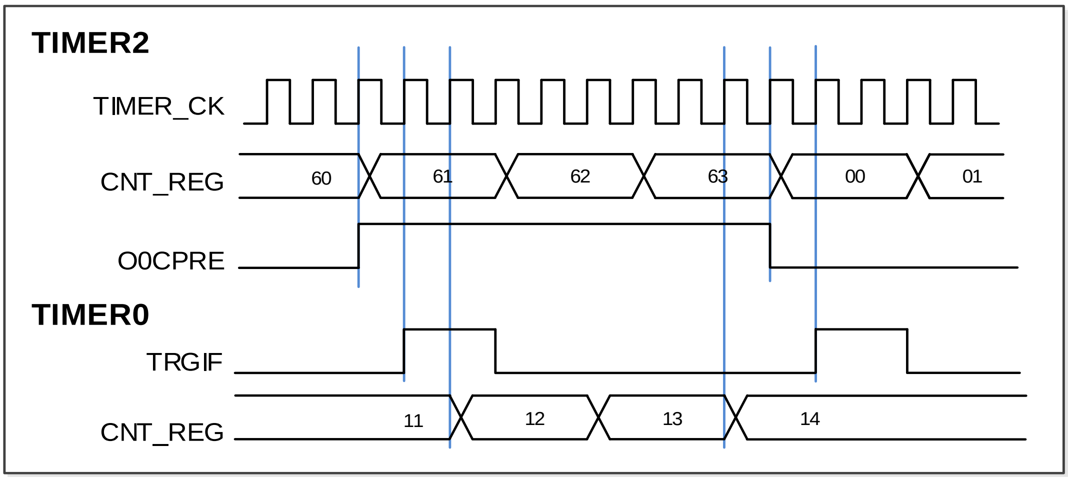
Figure 14-37. Pause TIMER0 with enable of TIMER2



In this example, we also can use O0CPRE as trigger source instead of enable signal output. Do as follow:

1. Configure TIMER2 in master mode and Output Compare 0 Reference (O0CPRE) signal as trigger output (MMS=100 in the TIMER2_CTL1 register).
2. Configure the TIMER2 O0CPRE waveform (TIMER2_ CHCTL0 register).
3. Configure TIMER0 to get the input trigger from TIMER2, configure TIMER0 in pause mode (TSCFG5[2:0] = 010 in the_ SYSCFG_TIMERxCFG register).
4. Enable TIMER0 by writing '1 in the CEN bit (TIMER0_CTL0 register).
5. Start TIMER2 by writing '1 in the CEN bit (TIMER2_CTL0 register).

Figure 14-38. Pause TIMER0 with O0CPREof TIMER2



Timer DMA mode

DMA mode is the function that configures timer's register by DMA module. The relative registers are TIMERx_DMACFG and TIMERx_DMATB. Corresponding DMA request bit should be asserted to enable DMA request for internal interrupt event. TIMERx will send a request to DMA when the interrupt event occurs. DMA is configured to M2P (memory to peripheral) mode and the address of TIMERx_DMATB is configured to PADDR (peripheral base address), then DMA will access the TIMERx_DMATB. In fact, TIMERx_DMATB register is only a buffer, timer will map the TIMERx_DMATB to an internal register, appointed by the field of DMATA in TIMERx_DMACFG. If the field of DMATC in TIMERx_DMACFG is 0 (1 transfer), the timer sends only one DMA request. While if TIMERx_DMATC is not 0, such as 3 (4 transfers), then timer will send 3 more requests to DMA, and DMA will access timer's registers DMATA+0x4, DMATA+0x8 and DMATA+0xC at the next 3 accesses to TIMERx_DMATB. In a word, one-time DMA internal interrupt event asserts, (DMATC+1) times request will be sent by TIMERx.

If one more DMA request event occurs, TIMERx will repeat the process above.

Timer debug mode

When the Cortex™-M23 is halted, and the TIMERx_HOLD configuration bit in DBG_CTL0 register is set to 1, the TIMERx counter stops.

14.1.5. TIMERx registers(x=0)

TIMER0 base address: 0x4001 2C00

Control register 0 (TIMERx_CTL0)

Address offset: 0x00

Reset value: 0x0000 0000

This register has to be accessed by word(32-bit)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved						CKDIV[1:0]		ARSE	CAM[1:0]		DIR	SPM	UPS	UPDIS	CEN
						rw		rw	rw		rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31:10	Reserved	Must be kept at reset value
9:8	CKDIV[1:0]	Clock division The CKDIV bits can be configured by software to specify division factor between the CK_TIMER and the dead-time and digital filter sample clock (DTS). 00: $f_{DTS}=f_{CK\_TIMER}$ 01: $f_{DTS}= f_{CK\_TIMER} /2$ 10: $f_{DTS}= f_{CK\_TIMER} /4$ 11: Reserved
7	ARSE	Auto-reload shadow enable 0: The shadow register for TIMERx_CAR register is disabled 1: The shadow register for TIMERx_CAR register is enabled
6:5	CAM[1:0]	Counter aligns mode selection 00: No center-aligned mode (edge-aligned mode). The direction of the counter is specified by the DIR bit. 01: Center-aligned and counting down assert mode. The counter counts under center-aligned and channel is configured in output mode (CHxMS=00 in TIMERx_CHCTL0 register). Only when the counter is counting down, compare interrupt flag of channels can be set. 10: Center-aligned and counting up assert mode. The counter counts under center-aligned and channel is configured in output mode (CHxMS=00 in TIMERx_CHCTL0 register). Only when the counter is counting up, compare interrupt flag of channels can be set. 11: Center-aligned and counting up/down assert mode. The counter counts under center-aligned and channel is configured in output mode (CHxMS=00 in

		TIMERx_CHCTL0 register). Both when the counter is counting up and counting down, compare interrupt flag of channels can be set. After the counter is enabled, cannot be switched from 0x00 to non 0x00.
4	DIR	Direction 0: Count up 1: Count down This bit is read only when the timer is configured in center-aligned mode or quadrature decoder mode.
3	SPM	Single pulse mode. 0: Counter continues after update event. 1: The counter counts until the next update event occurs.
2	UPS	Update source This bit is used to select the update event sources by software. 0: Any of the following events generate an update interrupt or DMA request: The UPG bit is set The counter generates an overflow or underflow event The restart mode controller generates an update event. 1: Only counter overflow/underflow generates an update interrupt or DMA request.
1	UPDIS	Update disable. This bit is used to enable or disable the update event generation. 0: update event enable. The update event is generate and the buffered registers are loaded with their preloaded values when one of the following events occurs: The UPG bit is set The counter generates an overflow or underflow event The restart mode controller generates an update event. 1: update event disable. The buffered registers keep their value, while the counter and the prescaler are reinitialized if the UG bit is set or if the restart mode controller generates a hardware reset event.
0	CEN	Counter enable 0: Counter disable 1: Counter enable The CEN bit must be set by software when timer works in external clock, pause mode and quadrature decoder mode.

Control register 1 (TIMERx_CTL1)

Address offset: 0x04

Reset value: 0x0000 0000

This register has to be accessed by word(32-bit)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															ISO4

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	ISO3	ISO2N	ISO2	ISO1N	ISO1	ISO0N	ISO0	TI0S	MMC[2:0]			DMAS	CCUC	Reserved	CCSE
	rw	rw	rw	rw	rw	rw	rw	rw	rw			rw	rw		rw

Bits	Fields	Descriptions
31:17	Reserved	Must be kept at reset value
16	ISO4	Idle state of channel 4 output Refer to ISO0 bit
15	Reserved	Must be kept at reset value
14	ISO3	Idle state of channel 3 output Refer to ISO0 bit
13	ISO2N	Idle state of channel 2 complementary output Refer to ISO0N bit
12	ISO2	Idle state of channel 2 output Refer to ISO0 bit
11	ISO1N	Idle state of channel 1 complementary output Refer to ISO0N bit
10	ISO1	Idle state of channel 1 output Refer to ISO0 bit
9	ISO0N	Idle state of channel 0 complementary output 0: When POEN bit is reset, CH0_ON is set low. 1: When POEN bit is reset, CH0_ON is set high This bit can be modified only when PROT [1:0] bits in TIMERx_CCHP register is 00.
8	ISO0	Idle state of channel 0 output 0: When POEN bit is reset, CH0_O is set low. 1: When POEN bit is reset, CH0_O is set high The CH0_O output changes after a dead-time if CH0_ON is implemented. This bit can be modified only when PROT [1:0] bits in TIMERx_CCHP register is 00.
7	TI0S	Channel 0 trigger input selection 0: The TIMERx_CH0 pin input is selected as channel 0 trigger input. 1: The result of combinational XOR of TIMERx_CH0, CH1 and CH2 pins is selected as channel 0 trigger input.
6:4	MMC[2:0]	Master mode control These bits control the selection of TRGO signal, which is sent in master mode to slave timers for synchronization function. 000: Reset. When the UPG bit in the TIMERx_SWEVG register is set or a reset is generated by the slave mode controller, a TRGO pulse occurs. And in the latter case, the signal on TRGO is delayed compared to the actual reset.

		001: Enable. This mode is useful to start several timers at the same time or to control a window in which a slave timer is enabled. In this mode the master mode controller selects the counter enable signal as TRGO. The counter enable signal is set when CEN control bit is set or the trigger input in pause mode is high. There is a delay between the trigger input in pause mode and the TRGO output, except if the master-slave mode is selected. 010: Update. In this mode the master mode controller selects the update event as TRGO. 011: Capture/compare pulse. In this mode the master mode controller generates a TRGO pulse when a capture or a compare match occurred in channal0. 100: Compare. In this mode the master mode controller selects the O0CPRE signal is used as TRGO 101: Compare. In this mode the master mode controller selects the O1CPRE signal is used as TRGO 110: Compare. In this mode the master mode controller selects the O2CPRE signal is used as TRGO 111: Compare. In this mode the master mode controller selects the O3CPRE signal is used as TRGO
3	DMAS	DMA request source selection 0: DMA request of channel x is sent when capture/compare event occurs. 1: DMA request of channel x is sent when update event occurs.
2	CCUC	Commutation control shadow register update control When the commutation control shadow enable (for CHxEN, CHxNEN and CHxCOMCTL bits) are set (CCSE=1), these shadow registers update are controlled as below: 0: The shadow registers update by when CMTG bit is set. 1: The shadow registers update by when CMTG bit is set or a rising edge of TRGI occurs. When a channel does not have a complementary output, this bit has no effect.
1	Reserved	Must be kept at reset value.
0	CCSE	Commutation control shadow enable 0: The shadow registers for CHxEN, CHxNEN and CHxCOMCTL bits are disabled. 1: The shadow registers for CHxEN, CHxNEN and CHxCOMCTL bits are enabled. After these bits have been written, they are updated based when commutation event coming. When a channel does not have a complementary output, this bit has no effect.

Slave mode configuration register (TIMERx_SMCFG)

Address offset: 0x08
Reset value: 0x0000 0000

This register has to be accessed by word(32-bit)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ETP	SMC1	ETPSC[1:0]		ETFC[3:0]				MSM	Reserved						
rw	rw	rw		rw				rw							

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value
15	ETP	External trigger polarity This bit specifies the polarity of ETI signal 0: ETI is active at high level or rising edge. 1: ETI is active at low level or falling edge.
14	SMC1	Part of SMC for enable External clock mode1 In external clock mode 1, the counter is clocked by any active edge on the ETIF signal. 0: External clock mode 1 disabled 1: External clock mode 1 enabled. Setting the SMC1 bit has the same effect as selecting external clock mode 0 with TSCFG6[2:0]= 3’b111. It is possible to simultaneously use external clock mode 1 with the reset mode, pause mode or event mode. But the TSCFGy[2:0](y=3,4,5) bits must not be 3’b111 in this case. The external clock input will be ETIF if external clock mode 1 and external clock mode 1 are enabled at the same time. **Note:** External clock mode 0 enable is in SYSCFG_TIMERxCFG register.
13:12	ETPSC[1:0]	External trigger prescaler The frequency of external trigger signal ETI must not be at higher than 1/4 of TIMERx_CK frequency. When the external trigger signal is a fast clocks, the prescaler can be enabled to reduce ETI frequency. 00: Prescaler disable 01: ETI frequency will be divided by 2 10: ETI frequency will be divided by 4 11: ETI frequency will be divided by 8
11:8	ETFC[3:0]	External trigger filter control The external trigger can be filtered by digital filter and this bit-field configure the filtering capability. Basic principle of digital filter: continuously sample the external trigger signal according to f_{SAMP} and record the number of times of the same level of the signal. After reaching the filtering capacity configured by this bit-field, it is considered to be an effective level. The filtering capability configuration is as follows:

EXTFC[3:0]	Times	f_{SAMP}
4’b0000	Filter disabled.	
4’b0001	2	f_{CK_TIMER}
4’b0010	4	
4’b0011	8	
4’b0100	6	$f_{DTS\_CK}/2$
4’b0101	8	
4’b0110	6	$f_{DTS\_CK}/4$
4’b0111	8	
4’b1000	6	$f_{DTS\_CK}/8$
4’b1001	8	
4’b1010	5	$f_{DTS\_CK}/16$
4’b1011	6	
4’b1100	8	
4’b1101	5	$f_{DTS\_CK}/32$
4’b1110	6	
4’b1111	8	

7	MSM	Master-slave mode This bit can be used to synchronize selected timers to begin counting at the same time. The TRGI is used as the start event, and through TRGO, timers are connected together. 0: Master-slave mode disable 1: Master-slave mode enable
6:0	Reserved	Must be kept at reset value

DMA and interrupt enable register (TIMERx_DMAINTEN)

Address offset: 0x0C
Reset value: 0x0000 0000

This register has to be accessed by word(32-bit)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	TRGDEN	CMTDEN	CH3DEN	CH2DEN	CH1DEN	CH0DEN	UPDEN	BRKIE	TRGIE	CMTIE	CH3IE	CH2IE	CH1IE	CH0IE	UPIE
	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31:15	Reserved	Must be kept at reset value.
14	TRGDEN	Trigger DMA request enable 0: disabled

		1: enabled
13	CMTDEN	Commutation DMA request enable 0: disabled 1: enabled
12	CH3DEN	Channel 3 capture/compare DMA request enable 0: disabled 1: enabled
11	CH2DEN	Channel 2 capture/compare DMA request enable 0: disabled 1: enabled
10	CH1DEN	Channel 1 capture/compare DMA request enable 0: disabled 1: enabled
9	CH0DEN	Channel 0 capture/compare DMA request enable 0: disabled 1: enabled
8	UPDEN	Update DMA request enable 0: disabled 1: enabled
7	BRKIE	Break interrupt enable 0: disabled 1: enabled
6	TRGIE	Trigger interrupt enable 0: disabled 1: enabled
5	CMTIE	commutation interrupt enable 0: disabled 1: enabled
4	CH3IE	Channel 3 capture/compare interrupt enable 0: disabled 1: enabled
3	CH2IE	Channel 2 capture/compare interrupt enable 0: disabled 1: enabled
2	CH1IE	Channel 1 capture/compare interrupt enable 0: disabled 1: enabled
1	CH0IE	Channel 0 capture/compare interrupt enable

		0: disabled 1: enabled
0	UPIE	Update interrupt enable 0: disabled 1: enabled

Interrupt flag register (TIMERx_INTF)

Address offset: 0x10

Reset value: 0x0000 0000

This register has to be accessed by word(32-bit)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															CH4IF
															rc_w0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved		SYSBIF	CH3OF	CH2OF	CH1OF	CH0OF	BRK1IF	BRK0IF	TRGIF	CMTIF	CH3IF	CH2IF	CH1IF	CH0IF	UPIF
		rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0.	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0

Bits	Fields	Descriptions
31:17	Reserved	Must be kept at reset value.
16	CH4IF	Channel 4 ‘s capture/compare interrupt flag Refer to CH0IF description
15:14	Reserved	Must be kept at reset value.
13	SYSBIF	System source break interrupt flag This flag is set by hardware when the system sources are active, and cleared by software if the system sources are inactive. 0: No system source break interrupt occurred 1: System source break interrupt occurred Note: When this bit is set, this bit must be cleared by software before the channel outputs are restored.
12	CH3OF	Channel 3 over capture flag Refer to CH0OF description
11	CH2OF	Channel 2 over capture flag Refer to CH0OF description
10	CH1OF	Channel 1 over capture flag Refer to CH0OF description
9	CH0OF	Channel 0 over capture flag

		When channel 0 is configured in input mode, this flag is set by hardware when a capture event occurs while CH0IF flag has already been set. This flag is cleared by software. 0: No over capture interrupt occurred 1: Over capture interrupt occurred
8	BRK1IF	BREAK1 interrupt flag This flag is set by hardware as soon as the BREAK1 input is active, and cleared by software if the BREAK1 input is not at active level. 0: No active level on BREAK1 inputs has been detected. 1: An active level on BREAK1 inputs has been detected. An interrupt is generated if BRKIE=1 in the TIMERx_DMAINTEN register.
7	BRK0IF	BREAK0 interrupt flag This flag is set by hardware when the BREAK0 input is active, and cleared by software if the BREAK0 input is not at active level. 0: No active level on break input has been detected. 1: An active level on break input has been detected.
6	TRGIF	Trigger interrupt flag 0: No trigger event occurred. 1: Trigger interrupt occurred.
5	CMTIF	Channel commutation interrupt flag This flag is set by hardware when channel’s commutation event occurs, and cleared by software 0: No channel commutation interrupt occurred 1: Channel commutation interrupt occurred
4	CH3IF	Channel 3 ‘s capture/compare interrupt flag Refer to CH0IF description
3	CH2IF	Channel 2 ‘s capture/compare interrupt flag Refer to CH0IF description
2	CH1IF	Channel 1 ‘s capture/compare interrupt flag Refer to CH0IF description
1	CH0IF	Channel 0 ‘s capture/compare interrupt flag This flag is set by hardware and cleared by software. When channel 0 is in input mode, this flag is set when a capture event occurs. When channel 0 is in output mode, this flag is set when a compare event occurs. 0: No Channel 0 interrupt occurred 1: Channel 0 interrupt occurred
0	UPIF	Update interrupt flag This bit is set by hardware on an update event and cleared by software. 0: No update interrupt occurred

1: Update interrupt occurred

Software event generation register (TIMERx_SWEVG)

Address offset: 0x14

Reset value: 0x0000 0000

This register has to be accessed by word(32-bit)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved							BRK1G	BRK0G	TRGG	CMTG	CH3G	CH2G	CH1G	CH0G	UPG
							w	w	w	w	w	w	w	w	w

Bits	Fields	Descriptions
31:9	Reserved	Must be kept at reset value.
8	BRK1G	BREAK1 event generation This bit is set by software to generate an event and cleared by hardware automatically. When this bit is set, the POEN(MOE) bit will be cleared and BRK1IF flag will be set. 0: No generate a BREAK1 event 1: Generate a BREAK1 event
7	BRK0G	Break0 event generation This bit is set by software and cleared by hardware automatically. When this bit is set, the POEN bit is cleared and BRKIF flag is set, related interrupt or DMA transfer can occur if enabled. 0: No generate a break event 1: Generate a break event
6	TRGG	Trigger event generation This bit is set by software and cleared by hardware automatically. When this bit is set, the TRGIF flag in TIMERx_INTF register is set, related interrupt or DMA transfer can occur if enabled. 0: No generate a trigger event 1: Generate a trigger event
5	CMTG	Channel commutation event generation This bit is set by software and cleared by hardware automatically. When this bit is set, channel's capture/compare control registers (CHxEN, CHxNEN and CHxCOMCTL bits) are updated based on the value of CCSE (in the TIMERx_CTL1). 0: No affect

		1: Generate channel's c/c control update event
4	CH3G	Channel 3's capture or compare event generation Refer to CH0G description
3	CH2G	Channel 2's capture or compare event generation Refer to CH0G description
2	CH1G	Channel 1's capture or compare event generation Refer to CH0G description
1	CH0G	Channel 0's capture or compare event generation This bit is set by software in order to generate a capture or compare event in channel 0, it is automatically cleared by hardware. When this bit is set, the CH0IF flag is set, the corresponding interrupt or DMA request is sent if enabled. In addition, if channel 1 is configured in input mode, the current value of the counter is captured in TIMERx_CH0CV register, and the CH0OF flag is set if the CH0IF flag was already high. 0: No generate a channel 1 capture or compare event 1: Generate a channel 1 capture or compare event
0	UPG	Update event generation This bit can be set by software, and cleared by hardware automatically. When this bit is set, the counter is cleared if the center-aligned or up counting mode is selected, else (down counting) it takes the auto-reload value. The prescaler counter is cleared at the same time. 0: No generate an update event 1: Generate an update event

Channel control register 0 (TIMERx_CHCTL0)

Address offset: 0x18

Reset value: 0x0000 0000

This register has to be accessed by word(32-bit)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved							CH1COMCTL[3]	Reserved							CH0COMCTL[3]

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH1COMCEN	CH1COMCTL[2:0]			CH1COMSEN	CH1COMFEN	CH1MS[1:0]		CH0COMCEN	CH0COMCTL[2:0]			CH0COMSEN	CH0COMFEN	CH0MS[1:0]	
CH1CAPFLT[3:0]				CH1CAPPSC[1:0]				CH0CAPFLT[3:0]				CH0CAPPSC[1:0]			
rw				rw		rw		rw				rw		rw	

Output compare mode:

Bits	Fields	Descriptions

31:25	Reserved	Must be kept at reset value
24	CH1COMCTL[3]	Refer to CH1COMCTL[2:0]
23:17	Reserved	Must be kept at reset value
16	CH0COMCTL[3]	Refer to CH0COMCTL[2:0]
15	CH1COMCEN	Channel 1 output compare clear enable Refer to CH0COMCEN description
14:12	CH1COMCTL[2:0]	Channel 1 compare output control, combine with CH1COMCTL[3]. Refer to CH0COMCTL description
11	CH1COMSEN	Channel 1 output compare shadow enable Refer to CH0COMSEN description
10	CH1COMFEN	Channel 1 output compare fast enable Refer to CH0COMSEN description
9:8	CH1MS[1:0]	Channel 1 mode selection This bit-field specifies the direction of the channel and the input signal selection. This bit-field is writable only when the channel is not active. (CH1EN bit in TIMERx_CHCTL2 register is reset). 00: Channel 1 is configured as output 01: Channel 1 is configured as input, IS1 is connected to CI1FE1 10: Channel 1 is configured as input, IS1 is connected to CI0FE1 11: Channel 1 is configured as input, IS1 is connected to ITS. This mode is working only if an internal trigger input is selected, through TSCFGx[2:0](x = 3,4,5,6,7) bit-field in SYSCFG_TIMER0CFG (x=0) register.
7	CH0COMCEN	Channel 0 output compare clear enable. When this bit is set, the O0CPRE signal is cleared when High level is detected on ETIF input. 0: Channel 0 output compare clear disable 1: Channel 0 output compare clear enable
6:4	CH0COMCTL[2:0]	Channel 0 compare output control This bit-field controls the behavior of the output reference signal O0CPRE which drives CH0_O and CH0_ON. O0CPRE is active high, while CH0_O and CH0_ON active level depends on CH0P and CH0NP bits. 0000: Frozen. The O0CPRE signal keeps stable, independent of the comparison between the register TIMERx_CH0CV and the counter TIMERx_CNT. 0001: Set the channel output. O0CPRE signal is forced high when the counter matches the output compare register TIMERx_CH0CV. 0010: Clear the channel output. O0CPRE signal is forced low when the counter matches the output compare register TIMERx_CH0CV. 0011: Toggle on match. O0CPRE toggles when the counter matches the output compare register TIMERx_CH0CV.

		0100: Force low. O0CPRE is forced low level. 0101: Force high. O0CPRE is forced high level. 0110: PWM mode0. When counting up, O0CPRE is active as long as the counter is smaller than TIMERx_CH0CV else inactive. When counting down, O0CPRE is inactive as long as the counter is larger than TIMERx_CH0CV else active. 0111: PWM mode1. When counting up, O0CPRE is inactive as long as the counter is smaller than TIMERx_CH0CV else active. When counting down, O0CPRE is active as long as the counter is larger than TIMERx_CH0CV else inactive. When configured in PWM mode, the O0CPRE level changes only when the output compare mode switches from “frozen” mode to “PWM” mode or when the result of the comparison changes. 1000: Retriggerable single pulse mode 0. The behavior of O0CPRE is performed as in PWM mode 0. When counting up, the O0CPRE is active. When an trigger event occurs, the O0CPRE is inactive. The O0CPRE is active again at the next update event; When counting down, the O0CPRE is inactive, When an trigger event occurs, the O0CPRE is active. The O0CPRE is inactive again at the next update event. 1001: Retriggerable single pulse mode 1. The behavior of O0CPRE is performed as in PWM mode 1. When counting up, the O0CPRE is inactive, When an trigger event occurs,the O0CPRE is active. The O0CPRE is inactive again at the next update event; When counting down, the O0CPRE is active. When an trigger event occurs, the O0CPRE is inactive.The O0CPRE is active again at the next update event. 1010:Reserved. 1011: Reserved. 1100: Combined PWM mode 0 – O0CPRE has the same behavior as in PWM mode 0. O0CPREC is the logical OR between O0CPRE and O1CPRE 1101: Combined PWM mode 1– O0CPRE has the same behavior as in PWM mode 1. O0CPREC is the logical AND between O0CPRE and O1CPRE. 1110: Asymmetric PWM mode 0 –O0CPRE has the same behavior as in PWM mode 0. O0CPREC outputs O0CPRE when the counter is counting up, O1CPRE when it is counting down. 1111: Asymmetric PWM mode 1 –O0CPRE has the same behavior as in PWM mode 1. O0CPREC outputs O0CPRE when the counter is counting up, O1CPRE when it is counting down. This bit cannot be modified when PROT [1:0] bit-filed in TIMERx_CCHP register is 11 and CH0MS bit-filed is 00(COMPARE MODE).
3	CH0COMSEN	Channel 0 compare output shadow enable When this bit is set, the shadow register of TIMERx_CH0CV register, which updates at each update event, will be enabled. 0: Channel 0 output compare shadow disable 1: Channel 0 output compare shadow enable The PWM mode can be used without validating the shadow register only in single pulse mode (SPM bit in TIMERx_CTL0 register is set).

		This bit cannot be modified when PROT [1:0] bit-filed in TIMERx_CCHP register is 11 and CH0MS bit-filed is 00.
2	CH0COMFEN	Channel 0 output compare fast enable When this bit is set, the effect of an event on the trigger in input on the capture/compare output will be accelerated if the channel is configured in PWM0 or PWM1 mode. The output channel will treat an active edge on the trigger input as a compare match, and CH0_O is set to the compare level independently from the result of the comparison. 0: Channel 0 output quickly compare disable. The minimum delay from an edge on the trigger input to activate CH0_O output is 5 clock cycles. 1: Channel 0 output quickly compare enable. The minimum delay from an edge on the trigger input to activate CH0_O output is 3 clock cycles.
1:0	CH0MS[1:0]	Channel 0 I/O mode selection This bit-field specifies the work mode of the channel and the input signal selection. This bit-field is writable only when the channel is not active. (CH0EN bit in TIMERx_CHCTL2 register is reset).). 00: Channel 0 is configured as output 01: Channel 0 is configured as input, IS0 is connected to CI0FE0 10: Channel 0 is configured as input, IS0 is connected to CI1FE0 11: Channel 0 is configured as input, IS0 is connected to ITS, This mode is working only if an internal trigger input is selected, through TSCFGx[2:0](x = 3,4,5,6,7) bit-field in SYSCFG_TIMER0CFG (x=0) register.

Input capture mode:

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value
15:12	CH1CAPFLT[3:0]	Channel 1 input capture filter control Refer to CH0CAPFLT description
11:10	CH1CAPPSC[1:0]	Channel 1 input capture prescaler Refer to CH0CAPPSC description
9:8	CH1MS[1:0]	Channel 1 mode selection Same as Output compare mode
7:4	CH0CAPFLT[3:0]	Channel 0 input capture filter control An event counter is used in the digital filter, in which a transition on the output occurs after N input events. This bit-field specifies the frequency used to sample CI0 input signal and the length of the digital filter applied to CI0. 0000: Filter disabled, $f_{SAMP}=f_{DTS}$, N=1 0001: $f_{SAMP}=f_{TIMER\_CK}$, N=2 0010: $f_{SAMP}= f_{TIMER\_CK}$, N=4 0011: $f_{SAMP}= f_{TIMER\_CK}$, N=8 0100: $f_{SAMP}=f_{DTS}/2$, N=6

		0101: $f_{SAMP}=f_{DTS}/2$, N=8 0110: $f_{SAMP}=f_{DTS}/4$, N=6 0111: $f_{SAMP}=f_{DTS}/4$, N=8 1000: $f_{SAMP}=f_{DTS}/8$, N=6 1001: $f_{SAMP}=f_{DTS}/8$, N=8 1010: $f_{SAMP}=f_{DTS}/16$, N=5 1011: $f_{SAMP}=f_{DTS}/16$, N=6 1100: $f_{SAMP}=f_{DTS}/16$, N=8 1101: $f_{SAMP}=f_{DTS}/32$, N=5 1110: $f_{SAMP}=f_{DTS}/32$, N=6 1111: $f_{SAMP}=f_{DTS}/32$, N=8
3:2	CH0CAPPSC[1:0]	Channel 0 input capture prescaler This bit-field specifies the factor of the prescaler on channel 0 input. The prescaler is reset when CH0EN bit in TIMERx_CHCTL2 register is clear. 00: Prescaler disable, capture is done on each channel input edge 01: Capture is done every 2 channel input edges 10: Capture is done every 4channel input edges 11: Capture is done every 8 channel input edges
1:0	CH0MS[1:0]	Channel 0 mode selection Same as Output compare mode

Channel control register 1 (TIMERx_CHCTL1)

Address offset: 0x1C

Reset value: 0x0000 0000

This register has to be accessed by word(32-bit)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved							CH3COMCTL[3]	Reserved							CH2COMCTL[3]

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH3COMCEN	CH3COMCTL[2:0]			CH3COMSEN	CH3COMFEN	CH3MS[1:0]		CH2COMCEN	CH2COMCTL[2:0]			CH2COMSEN	CH2COMFEN	CH2MS[1:0]	
CH3CAPFLT[3:0]				CH3CAPPSC[1:0]				CH2CAPFLT[3:0]				CH2CAPPSC[1:0]			
rw				rw		rw		rw				rw		rw	

Output compare mode:

Bits	Fields	Descriptions
31:25	Reserved	Must be kept at reset value
24	CH3COMCTL[3]	Channel 3 compare output control Refer to CH0COMCTL description

22:17	Reserved	Must be kept at reset value
16	CH2COMCTL[3]	Channel 2 compare output control Refer to CH0COMCTL description
15	CH3COMCEN	Channel 3 output compare clear enable Refer to CH0COMCEN description
14:12	CH3COMCTL[2:0]	Channel 3 compare output control Refer to CH0COMCTL description
11	CH3COMSEN	Channel 3 output compare shadow enable Refer to CH0COMSEN description
10	CH3COMFEN	Channel 3 output compare fast enable Refer to CH0COMSEN description
9:8	CH3MS[1:0]	Channel 3 mode selection This bit-field specifies the direction of the channel and the input signal selection. This bit-field is writable only when the channel is not active. (CH3EN bit in TIMERx_CHCTL2 register is reset). 00: Channel 3 is configured as output 01: Channel 3 is configured as input, IS3 is connected to CI3FE3 10: Channel 3 is configured as input, IS3 is connected to CI2FE3 11: Channel 3 is configured as input, IS3 is connected to ITS, This mode is working only if an internal trigger input is selected, through TSCFGx[2:0](x = 3,4,5,6,7) bit-field in SYSCFG_TIMER0CFG (x=0) register.
7	CH2COMCEN	Channel 2 output compare clear enable. When this bit is set, the O2CPRE signal is cleared when High level is detected on ETIF input. 0: Channel 2 output compare clear disable 1: Channel 2 output compare clear enable
6:4	CH2COMCTL[2:0]	Channel 2 compare output control This bit-field controls the behavior of the output reference signal O2CPRE which drives CH2_O and CH2_ON. O2CPRE is active high, while CH2_O and CH2_ON active level depends on CH2P and CH2NP bits. 0000: Frozen. The O2CPRE signal keeps stable, independent of the comparison between the output compare register TIMERx_CH2CV and the counter TIMERx_CNT. 0001: Set high on match. O2CPRE signal is forced high when the counter matches the output compare register TIMERx_CH2CV. 0010: Set low on match. O2CPRE signal is forced low when the counter matches the output compare register TIMERx_CH2CV. 0011: Toggle on match. O2CPRE toggles when the counter matches the output compare register TIMERx_CH2CV. 0100: Force low. O2CPRE is forced low level.

		0101: Force high. O2CPRE is forced high level. 0110: PWM mode0. When counting up, O2CPRE is active as long as the counter is smaller than TIMERx_CH2CV else inactive. When counting down, O2CPRE is inactive as long as the counter is larger than TIMERx_CH2CV else active. 0111: PWM mode1. When counting up, O2CPRE is inactive as long as the counter is smaller than TIMERx_CH2CV else active. When counting down, O2CPRE is active as long as the counter is larger than TIMERx_CH2CV else inactive. When configured in PWM mode, the O2CPRE level changes only when the output compare mode switches from “frozen” mode to “PWM” mode or when the result of the comparison changes. 1000: Retriggerable single pulse mode 0. The behavior of O2CPRE is performed as in PWM mode 0. When counting up, the O2CPRE is active. When an trigger event occurs, the O2CPRE is inactive. The O2CPRE is active again at the next update event; When counting down, the O0CPRE is inactive, When an trigger event occurs, the O0CPRE is active. The O0CPRE is inactive again at the next update event. 1001: Retriggerable single pulse mode 1. The behavior of O0CPRE is performed as in PWM mode 1. When counting up, the O0CPRE is inactive, When an trigger event occurs,the O0CPRE is active. The O0CPRE is inactive again at the next update event; When counting down, the O0CPRE is active. When an trigger event occurs, the O0CPRE is inactive.The O0CPRE is active again at the next update event. 1010:Reserved. 1011: Reserved. 1100: Combined PWM mode 0 – O2CPRE has the same behavior as in PWM mode 0. O2CPREC is the logical OR between O2CPRE and O3CPRE 1101: Combined PWM mode 1– O0CPRE has the same behavior as in PWM mode 1. O2CPREC is the logical AND between O2CPRE and O3CPRE. 1110: Asymmetric PWM mode 0 –O2CPRE has the same behavior as in PWM mode 0. O2CPREC outputs O2CPRE when the counter is counting up, O3CPRE when it is counting down. 1111: Asymmetric PWM mode 1 –O0CPRE has the same behavior as in PWM mode 1. O2CPREC outputs O2CPRE when the counter is counting up, O3CPRE when it is counting down. This bit cannot be modified when PROT [1:0] bit-filed in TIMERx_CCHP register is 11 and CH2MS bit-filed is 00(COMPARE MODE).
3	CH2COMSEN	Channel 2 compare output shadow enable When this bit is set, the shadow register of TIMERx_CH2CV register, which updates at each update event will be enabled. 0: Channel 2 output compare shadow disable 1: Channel 2 output compare shadow enable The PWM mode can be used without validating the shadow register only in single pulse mode (SPM bit in TIMERx_CTL0 register is set). This bit cannot be modified when PROT [1:0] bit-filed in TIMERx_CCHP register is

		11 and CH0MS bit-filed is 00.
2	CH2COMFEN	Channel 2 output compare fast enable When this bit is set, the effect of an event on the trigger in input on the capture/compare output will be accelerated if the channel is configured in PWM1 or PWM2 mode. The output channel will treat an active edge on the trigger input as a compare match, and CH2_O is set to the compare level independently from the result of the comparison. 0: Channel 2 output quickly compare disable. The minimum delay from an edge on the trigger input to activate CH2_O output is 5 clock cycles. 1: Channel 2 output quickly compare enable. The minimum delay from an edge on the trigger input to activate CH2_O output is 3 clock cycles.
1:0	CH2MS[1:0]	Channel 2 I/O mode selection This bit-field specifies the work mode of the channel and the input signal selection. This bit-field is writable only when the channel is not active. (CH2EN bit in TIMERx_CHCTL2 register is reset).). 00: Channel 2 is configured as output 01: Channel 2 is configured as input, IS2 is connected to CI2FE2 10: Channel 2 is configured as input, IS2 is connected to CI3FE2 11: Channel 2 is configured as input, IS2 is connected to ITS. This mode is working only if an internal trigger input is selected, through TSCFGx[2:0](x = 3,4,5,6,7) bit-field in SYSCFG_TIMER0CFG (x=0) register.

Input capture mode:

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value
15:12	CH3CAPFLT[3:0]	Channel 3 input capture filter control Refer to CH0CAPFLT description
11:10	CH3CAPPSC[1:0]	Channel 3 input capture prescaler Refer to CH0CAPPSC description
9:8	CH3MS[1:0]	Channel 3 mode selection Same as Output compare mode
7:4	CH2CAPFLT[3:0]	Channel 2 input capture filter control An event counter is used in the digital filter, in which a transition on the output occurs after N input events. This bit-field specifies the frequency used to sample CI2 input signal and the length of the digital filter applied to CI2. 0000: Filter disable, $f_{SAMP}=f_{DTS}$, N=1 0001: $f_{SAMP}=f_{TIMER\_CK}$, N=2 0010: $f_{SAMP}= f_{TIMER\_CK}$, N=4 0011: $f_{SAMP}= f_{TIMER\_CK}$, N=8 0100: $f_{SAMP}=f_{DTS}/2$, N=6 0101: $f_{SAMP}=f_{DTS}/2$, N=8

		0110: $f_{SAMP}=f_{DTS}/4$, N=6 0111: $f_{SAMP}=f_{DTS}/4$, N=8 1000: $f_{SAMP}=f_{DTS}/8$, N=6 1001: $f_{SAMP}=f_{DTS}/8$, N=8 1010: $f_{SAMP}=f_{DTS}/16$, N=5 1011: $f_{SAMP}=f_{DTS}/16$, N=6 1100: $f_{SAMP}=f_{DTS}/16$, N=8 1101: $f_{SAMP}=f_{DTS}/32$, N=5 1110: $f_{SAMP}=f_{DTS}/32$, N=6 1111: $f_{SAMP}=f_{DTS}/32$, N=8
3:2	CH2CAPPSC[1:0]	Channel 2 input capture prescaler This bit-field specifies the factor of the prescaler on channel 2 input. The prescaler is reset when CH2EN bit in TIMERx_CHCTL2 register is clear. 00: Prescaler disable, capture is done on each channel input edge 01: Capture is done every 2 channel input edges 10: Capture is done every 4 channel input edges 11: Capture is done every 8 channel input edges
1:0	CH2MS[1:0]	Channel 2 mode selection Same as Output compare mode

Channel control register 2 (TIMERx_CHCTL2)

Address offset: 0x20

Reset value: 0x0000 0000

This register has to be accessed by word(32-bit)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved														CH4P	CH4EN

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH3NP	Reserved	CH3P	CH3EN	CH2NP	CH2NEN	CH2P	CH2EN	CH1NP	CH1NEN	CH1P	CH1EN	CH0NP	CH0NEN	CH0P	CH0EN
rw		rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31:18	Reserved	Must be kept at reset value
17	CH4P	Channel 4 capture/compare function polarity Refer to CH0P description
16	CH4EN	Channel 4 capture/compare function enable Refer to CH0EN description
15	CH3NP	Channel3 complementary output polarity Refer to CH0NP description

14	Reserved	Must be kept at reset value
13	CH3P	Channel 3 capture/compare function polarity Refer to CH0P description
12	CH3EN	Channel 3 capture/compare function enable Refer to CH0EN description
11	CH2NP	Channel 2 complementary output polarity Refer to CH0NP description
10	CH2NEN	Channel 2 complementary output enable Refer to CH0NEN description
9	CH2P	Channel 2 capture/compare function polarity Refer to CH0P description
8	CH2EN	Channel 2 capture/compare function enable Refer to CH0EN description
7	CH1NP	Channel 1 complementary output polarity Refer to CH0NP description
6	CH1NEN	Channel 1 complementary output enable Refer to CH0NEN description
5	CH1P	Channel 1 capture/compare function polarity Refer to CH0P description
4	CH1EN	Channel 1 capture/compare function enable Refer to CH0EN description
3	CH0NP	Channel 0 complementary output polarity When channel 0 is configured in output mode, this bit specifies the complementary output signal polarity. 0: Channel 0 active high 1: Channel 0 active low When channel 0 is configured in input mode, in conjunction with CH0P, this bit is used to define the polarity of CI0. This bit cannot be modified when PROT [1:0] bit-filed in TIMERx_CCHP register is 11 or 10.
2	CH0NEN	Channel 0 complementary output enable When channel 0 is configured in output mode, setting this bit enables the complementary output in channel0. 0: Channel 0 complementary output disabled 1: Channel 0 complementary output enabled
1	CH0P	Channel 0 capture/compare function polarity When channel 0 is configured in output mode, this bit specifies the output signal polarity.

		0: Channel 0 active high 1: Channel 0 active low When channel 0 is configured in input mode, this bit specifies the CI0 signal polarity. [CH0NP, CH0P] will select the active trigger or capture polarity for CI0FE0 or CI1FE0. [CH0NP==0, CH0P==0]: CIxFE0's rising edge is the active signal for capture or trigger operation in slave mode. And CIxFE0 will not be inverted. [CH0NP==0, CH0P==1]: CIxFE0's falling edge is the active signal for capture or trigger operation in slave mode. And CIxFE0 will be inverted. [CH0NP==1, CH0P==0]: Reserved. [CH0NP==1, CH0P==1]: CIxFE0's falling and rising edge are both the active signal for capture or trigger operation in slave mode. And CIxFE0 will be not inverted. This bit cannot be modified when PROT [1:0] bit-filed in TIMERx_CCHP register is 11 or 10.
0	CH0EN	Channel 0 capture/compare function enable When channel 0 is configured in output mode, setting this bit enables CH0_O signal in active state. When channel 0 is configured in input mode, setting this bit enables the capture event in channel0. 0: Channel 0 disabled 1: Channel 0 enabled

Counter register (TIMERx_CNT)

Address offset: 0x24

Reset value: 0x0000 0000

This register has to be accessed by word(32-bit)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNT[15:0]															
rw															

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value
15:0	CNT[15:0]	This bit-filed indicates the current counter value. Writing to this bit-filed can change the value of the counter.

Prescaler register (TIMERx_PSC)

Address offset: 0x28

Reset value: 0x0000 0000

This register has to be accessed by word(32-bit)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PSC[15:0]															
rw															

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value
15:0	PSC[15:0]	Prescaler value of the counter clock The PSC clock is divided by (PSC+1) to generate the counter clock. The value of this bit-filed will be loaded to the corresponding shadow register at every update event.

Counter auto reload register (TIMERx_CAR)

Address offset: 0x2C

Reset value: 0x0000 FFFF

This register has to be accessed by word(32-bit)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CARL[15:0]															
rw															

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value
15:0	CARL[15:0]	Counter auto reload value This bit-filed specifies the auto reload value of the counter.

Counter repetition register (TIMERx_CREP)

Address offset: 0x30

Reset value: 0x0000 0000

This register has to be accessed by word(32-bit)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CREP[15:0]															
rw															

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	CREP15:0]	Counter repetition value This bit-filed specifies the update event generation rate. Each time the repetition counter counting down to zero, an update event is generated. The update rate of the shadow registers is also affected by this bit-filed when these shadow registers are enabled.

Channel 0 capture/compare value register (TIMERx_CH0CV)

Address offset: 0x34

Reset value: 0x0000 0000

This register has to be accessed by word(32-bit)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH0VAL[15:0]															
rw															

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value
15:0	CH0VAL[15:0]	Capture or compare value of channel0 When channel 0 is configured in input mode, this bit-filed indicates the counter value corresponding to the last capture event. And this bit-filed is read-only. When channel 0 is configured in output mode, this bit-filed contains value to be compared to the counter. When the corresponding shadow register is enabled, the shadow register updates every update event.

Channel 1 capture/compare value register (TIMERx_CH1CV)

Address offset: 0x38

Reset value: 0x0000 0000

This register has to be accessed by word(32-bit)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH1VAL[15:0]															
rw															

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value
15:0	CH1VAL[15:0]	Capture or compare value of channel1 When channel 1 is configured in input mode, this bit-filed indicates the counter value corresponding to the last capture event. And this bit-filed is read-only. When channel 1 is configured in output mode, this bit-filed contains value to be compared to the counter. When the corresponding shadow register is enabled, the shadow register updates every update event.

Channel 2 capture/compare value register (TIMERx_CH2CV)

Address offset: 0x3C
Reset value: 0x0000 0000

This register has to be accessed by word(32-bit)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH2VAL[15:0]															
rw															

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value
15:0	CH2VAL[15:0]	Capture or compare value of channel 2 When channel 2 is configured in input mode, this bit-filed indicates the counter value corresponding to the last capture event. And this bit-filed is read-only. When channel 2 is configured in output mode, this bit-filed contains value to be compared to the counter. When the corresponding shadow register is enabled, the shadow register updates every update event.

Channel 3 capture/compare value register (TIMERx_CH3CV)

Address offset: 0x40
Reset value: 0x0000 0000

This register has to be accessed by word(32-bit)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH3VAL[15:0]															
rw															

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value
15:0	CH3VAL[15:0]	Capture or compare value of channel 3 When channel3 is configured in input mode, this bit-filed indicates the counter value corresponding to the last capture event. And this bit-filed is read-only. When channel 3 is configured in output mode, this bit-filed contains value to be compared to the counter. When the corresponding shadow register is enabled, the shadow register updates every update event.

Complementary channel protection register (TIMERx_CCHP)

Address offset: 0x44

Reset value: 0x0000 0000

This register has to be accessed by word(32-bit)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved						BRK1P	BRK1EN	BRK1F[3:0]				BRK0F[3:0]			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
POEN	OAEN	BRK0P	BRK0EN	ROS	IOS	PROT[1:0]		DTCFG[7:0]							
rw	rw	rw	rw	rw	rw	rw		rw							

Bits	Fields	Descriptions
31:26	Reserved	Must be kept at reset value
25	BRK1P	BREAK1 input signal polarity This bit specifies the polarity of the BREAK1 input signal. 0: BREAK1 input active low 1: BREAK1 input active high This bit can be modified only when PROT(LOCK)[1:0] bit-field in TIMERx_CCHP(TIMx_BDTR) register is 00. Note: Every write operation to this bit needs a delay of 1 APB clock to active.
24	BRK1EN	BREAK1 input signal enable This bit can be set to enable the BREAK1 input signal. 0: BREAK1 input disabled 1: BREAK1 input enabled This bit can be modified only when PROT(LOCK)[1:0] bit-field in TIMERx_CCHP(TIMx_BDTR) register is 00. Note:

		1) Every write operation to this bit needs a delay of 1 APB clock to active. This bit is only used with ROS(OSSR)=1 and IOS(OSSI)=1.
23:20	BRK1F[3:0]	BREAK1 input signal filter An event counter is used in the digital filter, in which a transition on the output occurs after N input events. This bit-field specifies the frequency used to sample BREAK1 input signal and the length of the digital filter applied to BREAK1. 0000: Filter disabled. BREAK1 act asynchronously, N=1 0001: fSAMP = fCK_TIMER, N=2 0010: fSAMP = fCK_TIMER, N=4 0011: fSAMP = fCK_TIMER, N=8 0100: fSAMP = fDTS/2, N=6 0101: fSAMP = fDTS/2, N=8 0110: fSAMP = fDTS/4, N=6 0111: fSAMP = fDTS/4, N=8 1000: fSAMP = fDTS/8, N=6 1001: fSAMP = fDTS/8, N=8 1010: fSAMP = fDTS/16, N=5 1011: fSAMP = fDTS/16, N=6 1100: fSAMP = fDTS/16, N=8 1101: fSAMP = fDTS/32, N=5 1110: fSAMP = fDTS/32, N=6 1111: fSAMP = fDTS/32, N=8 This bit can be modified only when PROT(LOCK)[1:0] bit-field in TIMERx_CCHP(TIMx_BDTR) register is 00.
19:16	BRK0F[3:0]	BREAK0 input signal filter An event counter is used in the digital filter, in which a transition on the output occurs after N input events. This bit-field specifies the frequency used to sample BREAK0 input signal and the length of the digital filter applied to BREAK0. 0000: Filter disabled. BREAK0 act asynchronously, N=1 0001: fSAMP = fCK_TIMER, N=2 0010: fSAMP = fCK_TIMER, N=4 0011: fSAMP = fCK_TIMER, N=8 0100: fSAMP = fDTS/2, N=6 0101: fSAMP = fDTS/2, N=8 0110: fSAMP = fDTS/4, N=6 0111: fSAMP = fDTS/4, N=8 1000: fSAMP = fDTS/8, N=6 1001: fSAMP = fDTS/8, N=8 1010: fSAMP = fDTS/16, N=5 1011: fSAMP = fDTS/16, N=6 1100: fSAMP = fDTS/16, N=8 1101: fSAMP = fDTS/32, N=5 1110: fSAMP = fDTS/32, N=6 1111: fSAMP = fDTS/32, N=8

		This bit can be modified only when PROT(LOCK)[1:0] bit-field in TIMERx_CCHP(TIMx_BDTR) register is 00.
15	POEN	Primary output enable The bit can be set to 1 by: - Write 1 to this bit - If OAEN is set to 1, this bit is set to 1 at the next update event. The bit can be cleared to 0 by: - Write 0 to this bit - Valid fault input. When one of channels is configured in output mode, setting this bit enables the channel outputs (CHx_O and CHx_ON) if the corresponding enable bits (CHxEN, CHxNEN in TIMERx_CHCTL2 register) have been set. 0: Channel outputs are disabled or forced to idle state. 1: Channel outputs are enabled.
14	OAEN	Output automatic enable This bit specifies whether the POEN bit can be set automatically by hardware. 0: The POEN bit can only be set by software. 1: POEN can be set at the next update event, if the break input is not active. This bit can be modified only when PROT [1:0] bit-filed in TIMERx_CCHP register is 00.
13	BRK0P	BREAK0 input signal polarity This bit specifies the polarity of the BREAK1 input signal. 0: BREAK0 input active low 1: BREAK0 input active high This bit can be modified only when PROT(LOCK)[1:0] bit-field in TIMERx_CCHP(TIMx_BDTR) register is 00. Note: Every write operation to this bit needs a delay of 1 APB clock to active.
12	BRK0EN	BREAK0 input signal enable This bit can be set to enable the BREAK0 input signal. 0: BREAK0 input disabled 1: BREAK0 input enabled This bit can be modified only when PROT(LOCK)[1:0] bit-field in TIMERx_CCHP(TIMx_BDTR) register is 00. Note: 2) Every write operation to this bit needs a delay of 1 APB clock to active. This bit is only used with ROS(OSSR)=1 and IOS(OSSI)=1.
11	ROS	Run mode “off-state” enable When POEN bit is set (Run mode), this bit can be set to enable the “off-state” for the channels which has been configured in output mode. 0: “off-state” disabled. If the CHxEN or CHxNEN bit is reset, the corresponding channel is output disabled. 1: “off-state” enabled. If the CHxEN or CHxNEN bit is reset, the corresponding

		channel is “off-state”. This bit cannot be modified when PROT [1:0] bit-filed in TIMERx_CCHP register is 10 or 11.
10	IOS	Idle mode “off-state” enable When POEN bit is reset (Idle mode), this bit can be set to enable the “off-state” for the channels which has been configured in output mode. 0: “off-state” disabled. If the CHxEN/CHxNEN bits are both reset, the channels are output disabled. 1: “off-state” enabled. No matter the CHxEN/CHxNEN bits, the channels are “off-state”. This bit cannot be modified when PROT [1:0] bit-filed in TIMERx_CCHP register is 10 or 11.
9:8	PROT[1:0]	Complementary register protect control This bit-filed specifies the write protection property of registers. 00: protect disable. No write protection. 01: PROT mode 0.The ISOx/ISOxN bits in TIMERx_CTL1 register and the BRKEN/BRKP/OAEN/DTCFG bits in TIMERx_CCHP register are writing protected. 10: PROT mode 1. In addition of the registers in PROT mode 0, the CHxP/CHxNP bits in TIMERx_CHCTL2 register (if related channel is configured in output mode) and the ROS/IOS bits in TIMERx_CCHP register are writing protected. 11: PROT mode 2. In addition of the registers in PROT mode 1, the CHxCOMCTL/ CHxCOMSEN bits in TIMERx_CHCTL0/1 registers (if the related channel is configured in output) are writing protected. This bit-field can be written only once after the reset. Once the TIMERx_CCHP register has been written, this bit-field will be writing protected.
7:0	DTCFG[7:0]	Dead time configure This bit-field controls the value of the dead-time, which is inserted before the output transitions. The relationship between DTCFG value and the duration of dead-time is as follow: DTCFG [7:5] =3’b0xx: DTvalue =DTCFG [7:0]x tDT, tDT=tDTS. DTCFG [7:5] =3’b 10x: DTvalue = (64+DTCFG [5:0])xtDT, tDT =tDTS*2. DTCFG [7:5] =3’b 110: DTvalue = (32+DTCFG [4:0])xtDT, tDT=tDTS*8. DTCFG [7:5] =3’b 111: DTvalue = (32+DTCFG [4:0])xtDT, tDT =tDTS*16. This bit can be modified only when PROT [1:0] bit-filed in TIMERx_CCHP register is 00.

DMA configuration register (TIMERx_DMACFG)

Address offset: 0x48

Reset value: 0x0000 0000

This register has to be accessed by word(32-bit)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved			DMATC[4:0]					Reserved			DMATA [4:0]				
			rw								rw				

Bits	Fields	Descriptions
31:13	Reserved	Must be kept at reset value.
12:8	DMATC [4:0]	DMA transfer count This filed is defined the number of DMA will access(R/W) the register of TIMERx_DMATB
7:5	Reserved	Must be kept at reset value.
4:0	DMATA [4:0]	DMA transfer access start address This filed define the first address for the DMA access the TIMERx_DMATB. When access is done through the TIMERx_DMA address first time, this bit-field specifies the address you just access. And then the second access to the TIMERx_DMATB, you will access the address of start address + 0x4. 5'b0_0000: TIMERx_CTL0 5'b0_0001: TIMERx_CTL1 … In a word: Start Address = TIMERx_CTL0 + DMATA*4

DMA transfer buffer register (TIMERx_DMATB)

Address offset: 0x4C

Reset value: 0x0000 0000

This register has to be accessed by word(32-bit)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DMATB[15:0]															
rw															

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value
15:0	DMATB[15:0]	DMA transfer buffer When a read or write operation is assigned to this register, the register located at the address range (Start Addr + Transfer Timer* 4) will be accessed. The transfer Timer is calculated by hardware, and ranges from 0 to DMATC.

Channel control register 1 (TIMERx_CHCTL3)

Address offset: 0x54

Reset value: 0x0000 0000

This register has to be accessed by word(32-bit)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															CH4COM CTL[3]

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								CH4COM CEN	CH4COMCTL[2:0]			CH4COM SEN	CH4COM FEN	Reserved	
	rw			rw			rw			rw		rw			

Output compare mode:

Bits	Fields	Descriptions
31:17	Reserved	Must be kept at reset value
16	CH4COMCTL[3]	Channel 4 compare output control Refer to CH0COMCTL description
15:8	Reserved	Must be kept at reset value
7	CH4COMCEN	Channel 4 output compare clear enable. When this bit is set, the O4CPRE signal is cleared when High level is detected on ETIF input. 0: Channel 4 output compare clear disable 1: Channel 4 output compare clear enable
6:4	CH4COMCTL[2:0]	Channel 4 compare output control This bit-field controls the behavior of the output reference signal O4CPRE which drives CH4_O and CH4_ON. O4CPRE is active high, while CH4_O active level depends on CH4P bits. 0000: Frozen. The O4CPRE signal keeps stable, independent of the comparison between the output compare register TIMERx_CH4CV and the counter TIMERx_CNT. 0001: Set high on match. O4CPRE signal is forced high when the counter matches the output compare register TIMERx_CH4CV. 0010: Set low on match. O4CPRE signal is forced low when the counter matches the output compare register TIMERx_CH4CV. 0011: Toggle on match. O4CPRE toggles when the counter matches the output compare register TIMERx_CH4CV. 0100: Force low. O4CPRE is forced low level. 0101: Force high. O4CPRE is forced high level. 0110: PWM mode0. When counting up, O4CPRE is active as long as the counter is

		smaller than TIMERx_CH4CV else inactive. When counting down, O4CPRE is inactive as long as the counter is larger than TIMERx_CH4CV else active. 0111: PWM mode1. When counting up, O4CPRE is inactive as long as the counter is smaller than TIMERx_CH4CV else active. When counting down, O4CPRE is active as long as the counter is larger than TIMERx_CH4CV else inactive. When configured in PWM mode, the O4CPRE level changes only when the output compare mode switches from “frozen” mode to “PWM” mode or when the result of the comparison changes. 1000: Retriggerable single pulse mode 0. The behavior of O4CPRE is performed as in PWM mode 0. When counting up, the O4CPRE is active. When an trigger event occurs, the O4CPRE is inactive. The O4CPRE is active again at the next update event; When counting down, the O4CPRE is inactive, When an trigger event occurs, the O4CPRE is active. The O4CPRE is inactive again at the next update event. 1001: Retriggerable single pulse mode 1. The behavior of O4CPRE is performed as in PWM mode 1. When counting up, the O4CPRE is inactive, When an trigger event occurs,the O4CPRE is active. The O4CPRE is inactive again at the next update event; When counting down, the O4CPRE is active. When an trigger event occurs, the O4CPRE is inactive.The O4CPRE is active again at the next update event. Others:Reserved. This bit cannot be modified when PROT [1:0] bit-filed in TIMERx_CCHP register is 11 .
3	CH4COMSEN	Channel 4 compare output shadow enable When this bit is set, the shadow register of TIMERx_CH4CV register, which updates at each update event will be enabled. 0: Channel 4 output compare shadow disable 1: Channel 4 output compare shadow enable The PWM mode can be used without validating the shadow register only in single pulse mode (SPM bit in TIMERx_CTL0 register is set). This bit cannot be modified when PROT [1:0] bit-filed in TIMERx_CCHP register is 11 and CH0MS bit-filed is 00.
2	CH4COMFEN	Channel 4 output compare fast enable When this bit is set, the effect of an event on the trigger in input on the capture/compare output will be accelerated if the channel is configured in PWM0 or PWM1 mode. The output channel will treat an active edge on the trigger input as a compare match, and CH4_O is set to the compare level independently from the result of the comparison. 0: Channel 4 output quickly compare disable. The minimum delay from an edge on the trigger input to activate CH4_O output is 5 clock cycles. 1: Channel 4 output quickly compare enable. The minimum delay from an edge on the trigger input to activate CH4_O output is 3 clock cycles.

1:0	Reserved	Must be kept at reset value

Channel 4 capture/compare value register (TIMERx_CH4CV)

Address offset: 0x58

Reset value: 0x0000 0000

This register has to be accessed by word(32-bit)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CCH4CH2	CCH4CH1	CCH4CH0	Reserved												

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH4VAL[15:0]															
rw															

Bits	Fields	Descriptions
31	CCH4CH2	Combine Channel 4 and Channel 2 0: O4CPRE and O2CPREF independent output 1: O2CPREF is is the logical AND of O2CPREF and O4CPRE This bit can either have immediate effect or be preloaded and taken into account after an update event (if compare shadow is enable in TIMERx_CHCTL1).
30	CCH4CH1	Combine Channel 4 and Channel 1 0: O4CPRE and O1CPREF independent output 1: O1CPREF is is the logical AND of O1CPREF and O4CPRE This bit can either have immediate effect or be preloaded and taken into account after an update event (if compare shadow is enable in TIMERx_CHCTL1).
29	CCH4CH0	Combine Channel 4 and Channel 0 0: O4CPRE and O0CPREF independent output 1: O0CPREF is is the logical AND of O0CPREF and O4CPRE This bit can either have immediate effect or be preloaded and taken into account after an update event (if compare shadow is enable in TIMERx_CHCTL1).
28:16	Reserved	Must be kept at reset value
15:0	CH4VAL[15:0]	Capture or compare value of channel 4 When channel4 is configured in input mode, this bit-filed indicates the counter value corresponding to the last capture event. And this bit-filed is read-only. When channel 4 is configured in output mode, this bit-filed contains value to be compared to the counter. When the corresponding shadow register is enabled, the shadow register updates every update event.

Alternate function control register 0 (TIMER0_AFCTL0)

Address offset: 0x60

Reset value: 0x0000 0001

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved														ETISEL[3:2]	
														rw	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ETISEL[1:0]		Reserved				BRK0INP	Reserved								BRK0INEN
rw						rw									rw

Bits	Fields	Descriptions
31:18	Reserved	Must be kept at reset value.
17:14	ETISEL	ETI source selection This bits select the ETI input source. 0000: ETI legacy mode 0011: ADC_WD0_OUT 0100: ADC_WD1_OUT 0101: ADC_WD2_OUT Others: Reserved This bit cannot be modified when PROT [1:0] bit-filed in TIMERx_CCHP register is 01.
13:10	Reserved	Must be kept at reset value.
9	BRK0INP	BREAK0 alternate function input polarity This bit is used to configure the BRKIN0 input polarity, and the specitic polarity is determined by this bit and the BRK0P bit. 0: BRKIN0 input signal will not be inverted (BRK0P =0, the input signal is active low; BRK0P =1, the input signal is active high) 1: BRKIN0 input signal will be inverted (BRK0P=0, the input signal is active high; BRK0P =1, the input signal is active low) This bit can be modified only when PROT[1:0] bit-field in TIMERx_CCHP register is 00.
8:1	Reserved	Must be kept at reset value.
0	BRK0INEN	BREAK0 alternate function input enable 0: BRKIN0 alternate function input disabled 1: BRKIN0 alternate function input enabled This bit can be modified only when PROT[1:0] bit-field in TIMERx_CCHP register is 00.

Alternate function control register 1 (TIMER0_AFCTL1)

Address offset: 0x64

Reset value: 0x0000 0001

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved						BRK1INP	Reserved								BRK1INE N
						rw									rw

Bits	Fields	Descriptions
31:10	Reserved	Must be kept at reset value.
9	BRK1INP	BRKIN1 alternate function input polarity This bit is used to configure the BRKIN1 input polarity, and the specitic polarity is determined by this bit and the BRK0P bit. 0: BRKIN1 input signal will not be inverted (BRK0P =0, the input signal is active low; BRK0P =1, the input signal is active high) 1: BRKIN1 input signal will be inverted (BRK0P=0, the input signal is active high; BRK0P =1, the input signal is active low) This bit can be modified only when PROT[1:0] bit-field in TIMERx_CCHP register is 00.
8:1	Reserved	Must be kept at reset value.
0	BRK1INEN	BRKIN1 alternate function input enable 0: BRKIN1 alternate function input disabled 1: BRKIN1 alternate function input enabled This bit can be modified only when PROT[1:0] bit-field in TIMERx_CCHP register is 00.

input selection register (TIMERx_INSEL)

Address offset: 0x68

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				CI1_SEL[3:0]				Reserved				CI0_SEL[3:0]			

Bits	Fields	Descriptions
31:12	Reserved	Must be kept at reset value.
11:8	CI1_SEL[3:0]	TIMER0_CH1 input selection 0000: TIMER0_CH1 input capture. 0001: CMP1 output. Others: Reserved
7:4	Reserved	Must be kept at reset value.
3:0	CI0_SEL[3:0]	TIMER0_CH0 input selection 0000: TIMER0_CH0 input capture. 0001: CMP0 output. Others: Reserved

Configuration register (TIMERx_CFG)

Address offset: 0xFC

Reset value: 0x0000 0000

This register has to be accessed by word(32-bit)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved														CHVSEL	OUTSEL
														rw	rw

Bits	Fields	Descriptions
31:2	Reserved	Must be kept at reset value
1	CHVSEL	Write CHxVAL register selection This bit-field set and reset by software. 1: If write the CHxVAL register, the write value is same as the CHxVAL value, the write access ignored 0: No effect
0	OUTSEL	The output value selection This bit-field set and reset by software 1: If POEN and IOS is 0, the output disabled 0: No effect

14.2. General level0 timer (TIMERx, x= 2)

14.2.1. Overview

The general level0 timer module (TIMER 2) is a four-channel timer that supports input capture and output compare. They can generate PWM signals to control motor or be used for power management applications. The general level0 timer has a 16-bit counter that can be used as an unsigned counter.

In addition, the general level0 timers can be programmed and be used for counting, their external events can be used to drive other timers.

Timers are completely independent with each other, but they may be synchronized to provide a larger timer with their counter value increasing in unison.

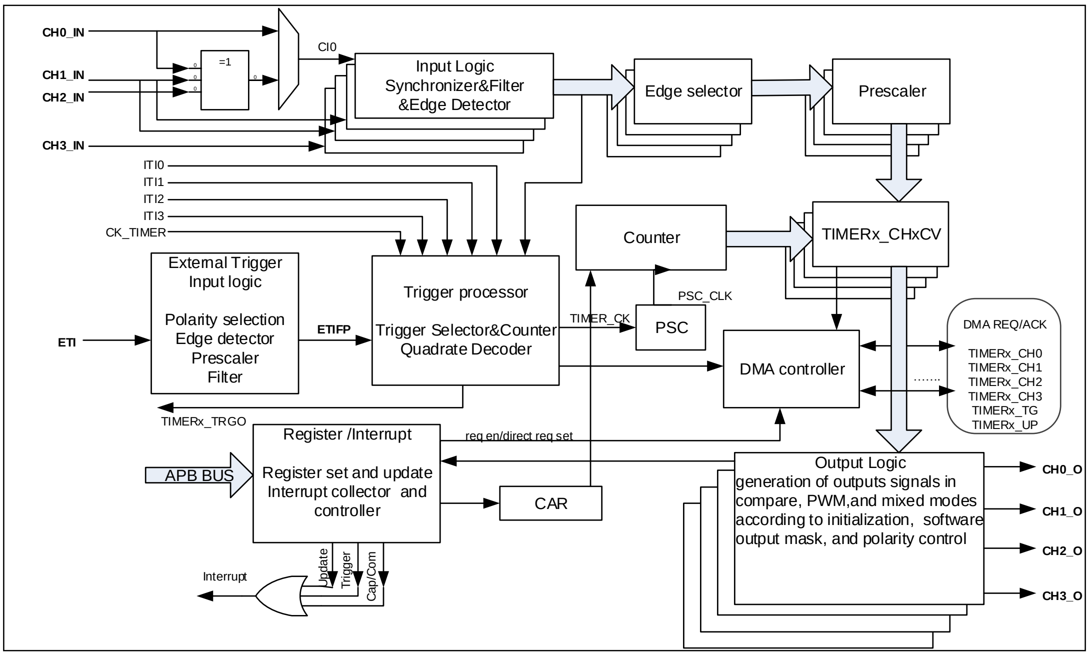
14.2.2. Characteristics

- Total channel num: 4.
- Counter width: 16 bits
- Clock source of timer is selectable: internal clock, internal trigger, external input, external trigger.
- Multiple counter modes: up counting, down counting and center-aligned counting.
- Quadrature decoder: used for motion tracking and determination of both rotation direction and position.
- Hall sensor function: used for 3-phase motor control.
- Programmable prescaler: 16 bits. The factor can be changed ongoing.
- Each channel is user-configurable: input capture mode, output compare mode, programmable PWM mode and single pulse mode.
- Auto reload function.
- Interrupt output or DMA request: update event, trigger event and compare/capture event.
- Daisy chaining of timer module allows a single timer to start multiple timers.
- Timer synchronization allows the selected timers to start counting on the same clock cycle.
- Timer master-slave management.

14.2.3. Block diagram

Figure 14-39. General Level 0 timer block diagram provides details on the internal configuration of the general level0 timer.

Figure 14-39. General Level 0 timer block diagram



14.2.4. Function overview

Clock source configuration

The general level0 TIMER has the capability of being clocked by either the CK_TIMER or an alternate clock source controlled by SMC (TIMERx_SMCFG bit[2:0]).

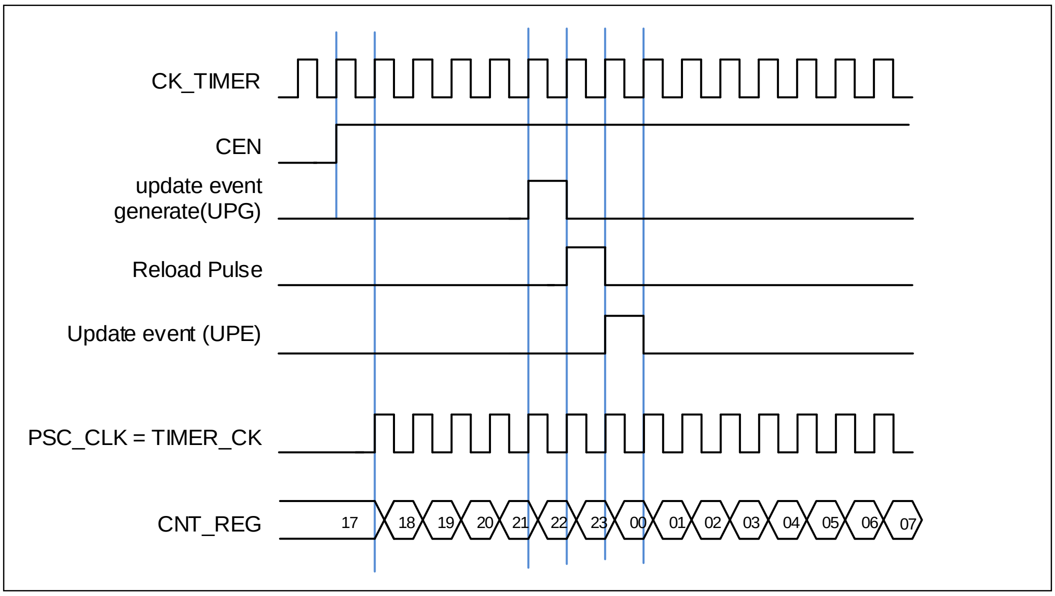
- SMC[2:0] = 3’b000. Internal clock CK_TIMER is selected as timer clock source which is from module RCU.

The default clock source is the CK_TIMER for driving the counter prescaler when the SMC[2:0] = 3’b000. When the CEN is set, the CK_TIMER will be divided by PSC value to generate PSC_CLK.

In this mode, the TIMER_CK which drives counter’s prescaler to count is equal to CK_TIMER which is from RCU module.

If the SMC[2:0] in the TIMERx_SMCFG register are setting to an available value including 0x1, 0x2, 0x3 and 0x7, the prescaler is clocked by other clock sources selected by the TRGS[2:0] in the TIMERx_SMCFG register, more details will be introduced later. When the SMC[2:0] bits are set to 0x4, 0x5 or 0x6, the internal clock CK_TIMER is the counter prescaler driving clock source.

Figure 14-40. Timing chart of internal clock divided by 1



- SMC[2:0] = 3'b111 (external clock mode 0). External input pin is selected as timer clock source.

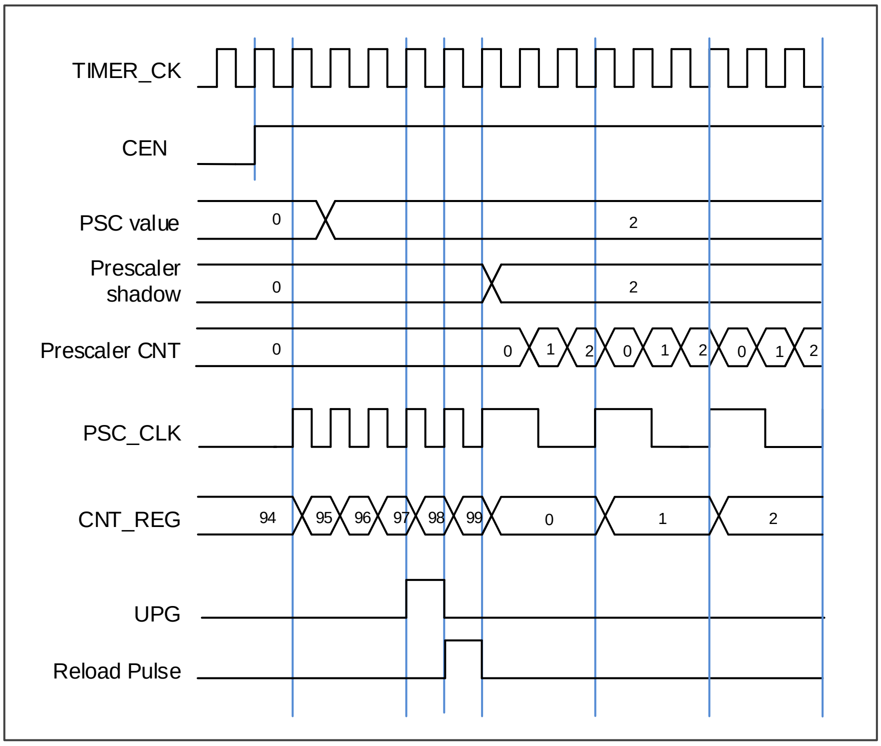
The TIMER_CK, which drives counter's prescaler to count, can be triggered by the event of rising or falling edge on the external pin TIMERx_CH0/TIMERx_CH1. This mode can be selected by setting SMC[2:0] to 0x7 and the TRGS[2:0] to 0x4, 0x5 or 0x6.

And, the counter prescaler can also be driven by rising edge on the internal trigger input pin ITI0/1/2/3. This mode can be selected by setting SMC[2:0] to 0x7 and the TRGS[2:0] to 0x0, 0x1, 0x2 or 0x3.

Clock prescaler

The counter clock (PSC_CK) is obtained by the TIMER_CK through the prescaler, and the prescale factor can be configured from 1 to 65536 through the prescaler register (TIMERx_PSC). The new written prescaler value will not take effect until the next update event.

Figure 14-41. Timing chart of PSC value change from 0 to 2



Counter up counting

In this mode, the counter counts up continuously from 0 to the counter-reload value, which is defined in the TIMERx_CAR register, in a count-up direction. Once the counter reaches the counter reload value, the counter will start counting up from 0 again. The update event is generated at each counter overflow. The counting direction bit DIR in the TIMERx_CTL1 register should be set to 0 for the up counting mode.

Whenever, if the update event software trigger is enabled by setting the UPG bit in the TIMERx_SWEVG register, the counter value will be initialized to 0 and an update event will be generated.

If the UPDIS bit in TIMERx_CTL0 register is set, the update event is disabled.

When an update event occurs, all the shadow registers (counter auto reload register, prescaler register) are updated.

Figure 14-42. Timing chart of up counting mode, PSC=0/2

_and ***Figure 14-43. Timing chart of up counting, change TIMERx_CAR ongoing***

show some examples of the counter behavior for different clock prescaler factor when TIMERx_CAR=0x99.

Figure 14-42. Timing chart of up counting mode, PSC=0/2

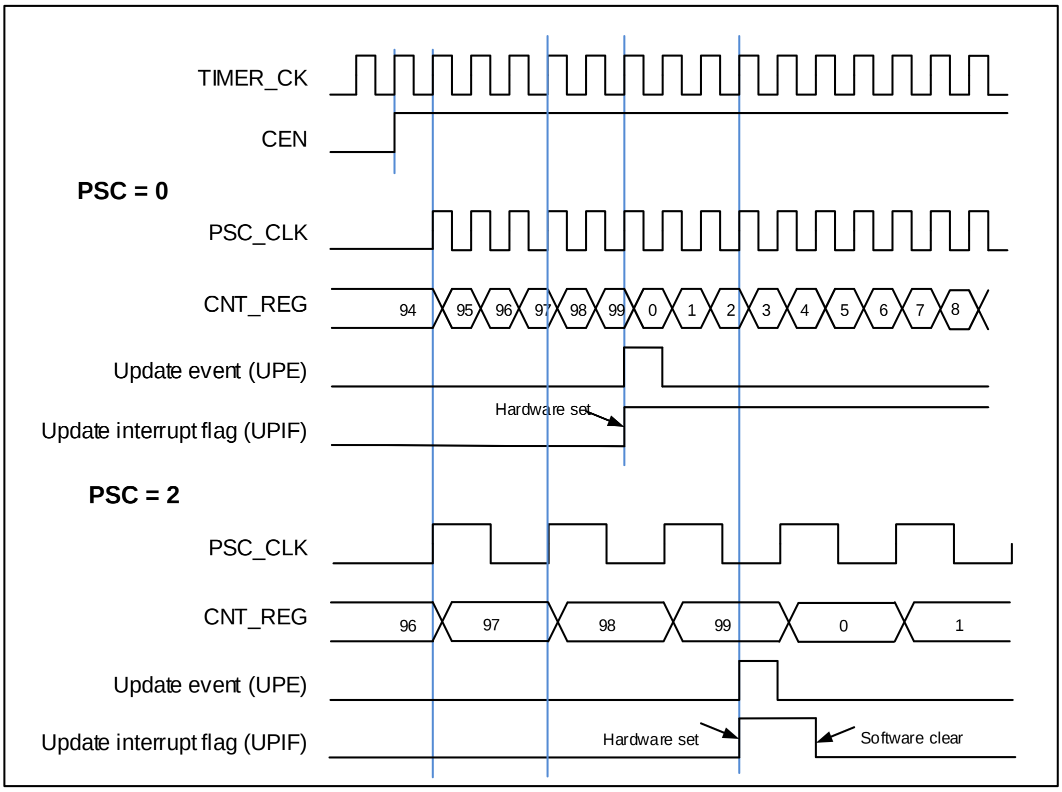
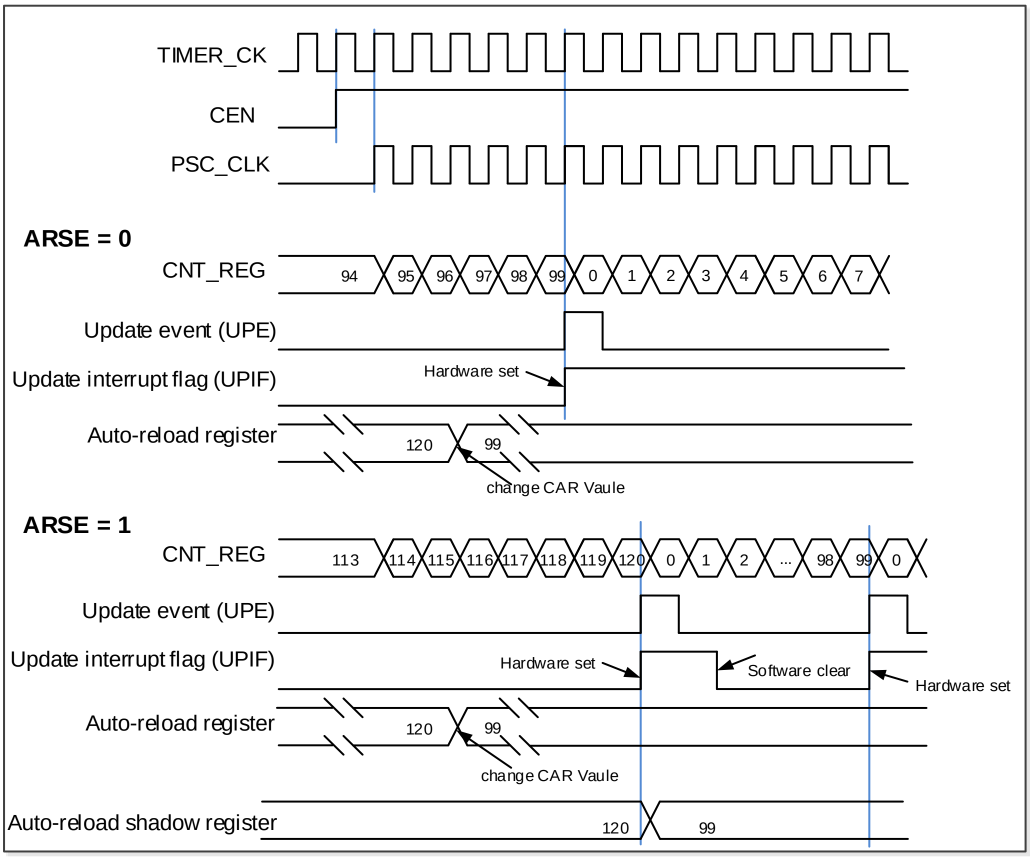


Figure 14-43. Timing chart of up counting, change TIMERx_CAR ongoing



Counter down counting

In this mode, the counter counts down continuously from the counter reload value, which is defined in the TIMERx_CAR register, in a count-down direction. Once the counter reaches 0, the counter will start counting down from the counter-reload value again. The counting direction bit DIR in the TIMERx_CTL0 register should be set to 1 for the down counting mode.

When the update event is set by the UPG bit in the TIMERx_SWEVG register, the counter value will be initialized to the counter reload value and an update event will be generated.

If the UPDIS bit in TIMERx_CTL0 register is set, the update event is disabled.

When an update event occurs, all the shadow registers (counter auto reload register, prescaler register) are updated.

Figure 14-44. Timing chart of down counting mode, PSC=0/2

_and *Figure 14-45. Timing chart of down counting mode, change TIMERx_CAR ongoing*

show some examples of the counter behavior for different clock frequencies when TIMERx_CAR = 0x99.

Figure 14-44. Timing chart of down counting mode, PSC=0/2

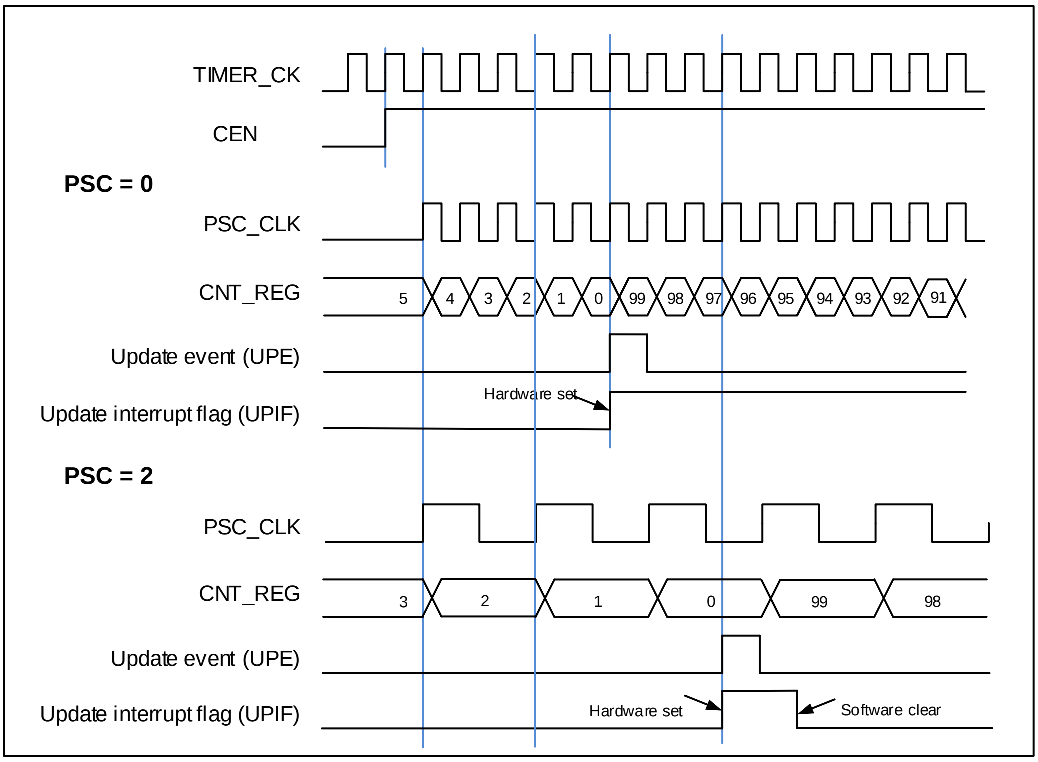
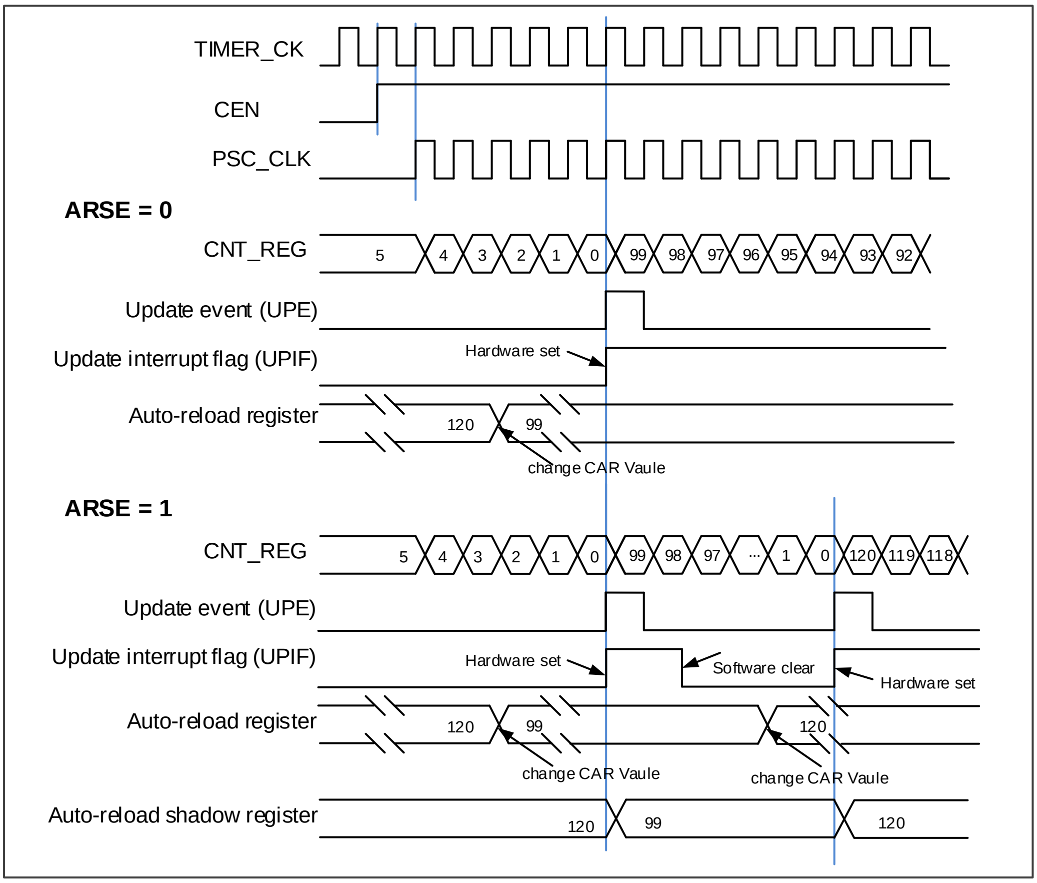


Figure 14-45. Timing chart of down counting mode, change TIMERx_CAR ongoing



Counter center-aligned counting

In this 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 subtract 1 in the up-counting direction and generates an underflow event when the counter counts to 1 in the down-counting mode. The counting direction bit DIR in the TIMERx_CTL0 register is read-only and indicates the counting direction when in the center-aligned mode.

Setting the UPG bit in the TIMERx_SWEVG register will initialize the counter value to 0 and generate an update event irrespective of whether the counter is counting up or down in the center-aligned counting mode.

The UPIF bit in the TIMERx_INTF register will be set to 1 either when an underflow event or an overflow event occurs. While the CHxIF bit is associated with the value of CAM in TIMERx_CTL0. The details refer to ***Figure 14-46. Timing chart of center-aligned counting mode***

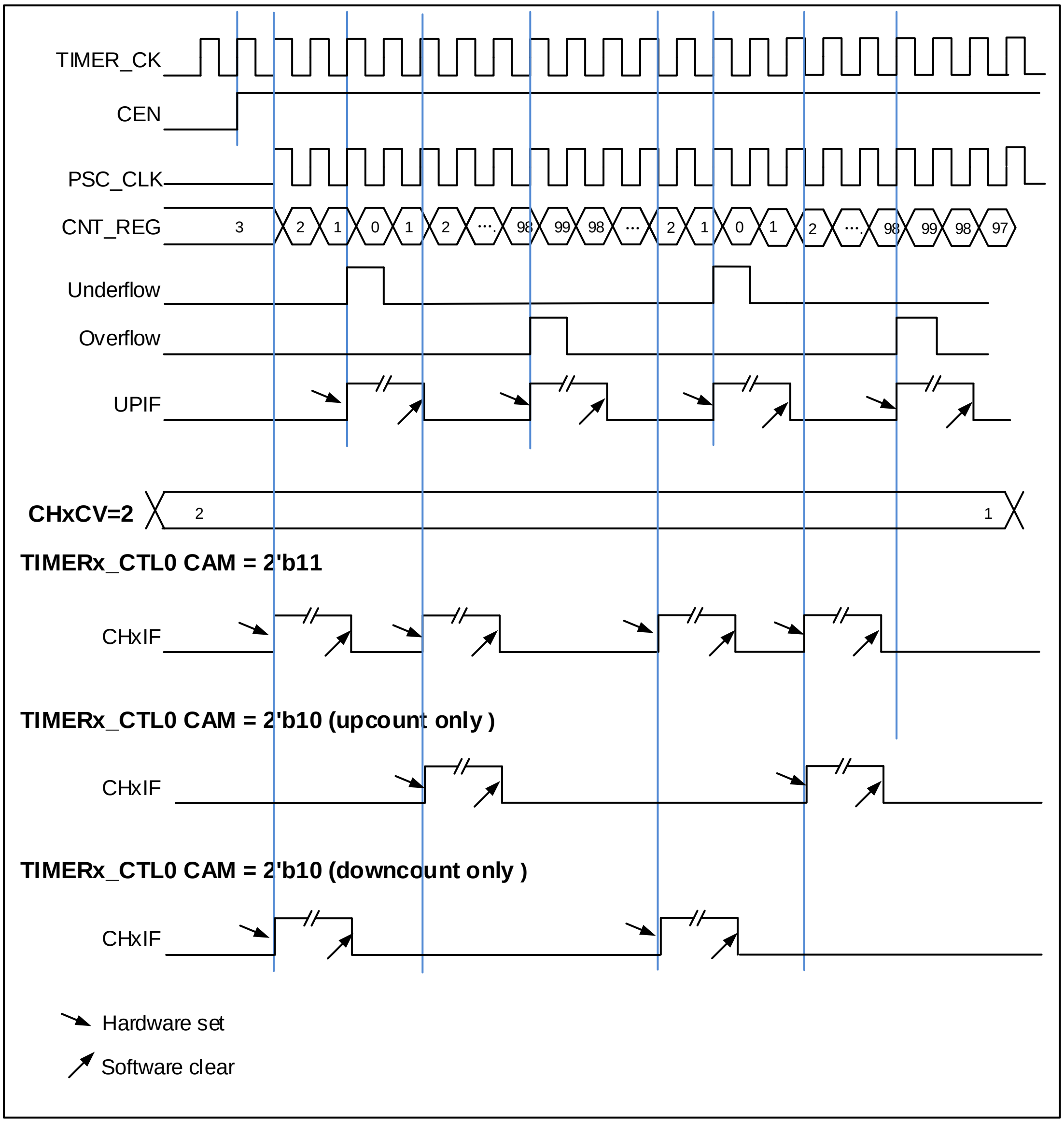
.

If the UPDIS bit in the TIMERx_CTL0 register is set, the update event is disabled.

When an update event occurs, all the shadow registers (counter auto-reload register, prescaler register) are updated.

Figure 14-46. Timing chart of center-aligned counting mode shows the example of the counter behavior when TIMERx_CAR=0x99, TIMERx_PSC=0x0

Figure 14-46. Timing chart of center-aligned counting mode



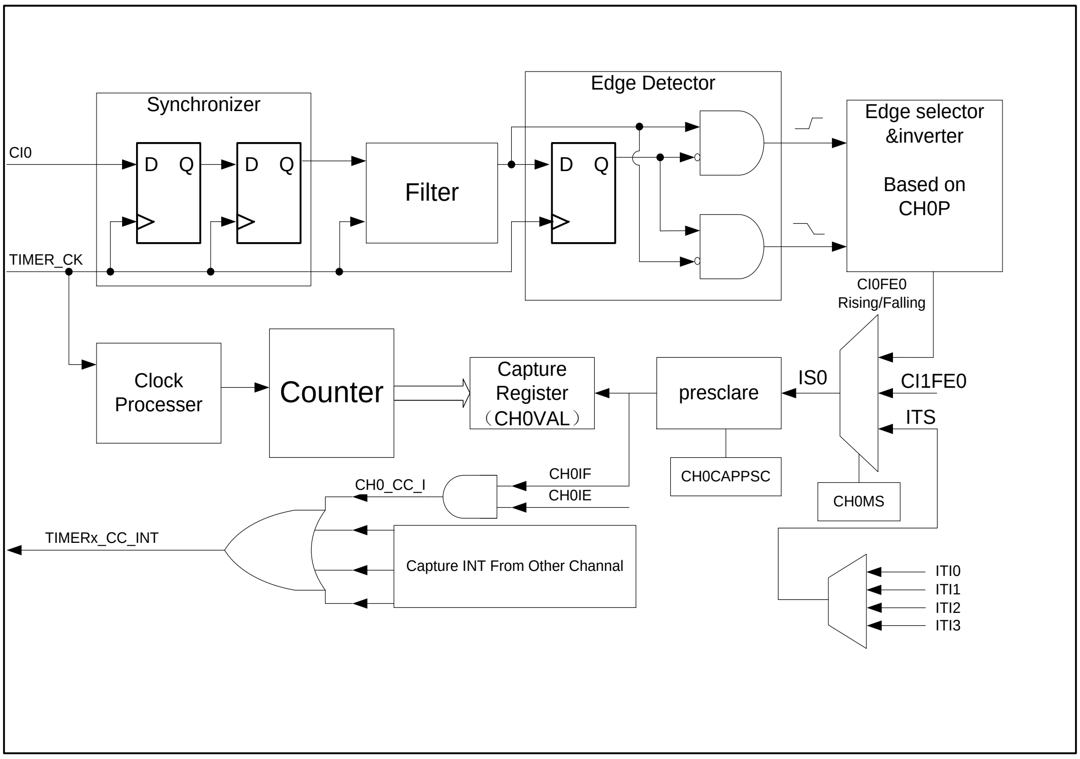
Input capture and output compare channels

The general level0 Timer has four independent channels which can be used as capture inputs or compare match outputs. Each channel is built around a channel capture compare register including an input stage, channel controller and an output stage.

Channel input capture function

Input capture mode allows the channel to perform measurements such as pulse timing, frequency, period, duty cycle and so on. The input stage consists of a digital filter, a channel polarity selection, edge detection and a channel prescaler. When a selected edge occurs on the channel input, the current value of the counter is captured into the TIMERx_CHxCV register, at the same time the CHxIF bit is set and the channel interrupt is generated if it is enabled when CHxIE=1.

Figure 14-47. Channel input capture principle



The input signals of channelx (CIx) can be the TIMERx_CHx signal or the XOR signal of the TIMERx_CH0, TIMERx_CH1 and TIMERx_CH2 signals. First, the input signal of channel (CIx) is synchronized to TIMER_CK signal, and then sampled by a digital filter to generate a filtered input signal. Then through the edge detector, the rising or falling edge is detected by configuring CHxP bit. The input capture signal can also be selected from the input signal of other channel or the internal trigger signal by configuring CHxMS bits. The IC prescaler makes several input events generate one effective capture event. On the capture event, TIMERx_CHxCV will store the value of counter.

So, the process can be divided into several steps as below:

Step1: Filter configuration (CHxCAPFLT in TIMERx_CHCTL0).

Based on the input signal and quality of requested signal, configure compatible CHxCAPFLT.

Step2: Edge selection (CHxP/CHxNP in TIMERx_CHCTL2).

Rising edge or falling edge, choose one by configuring CHxP/CHxNP bits.

Step3: Capture source selection (CHxMS in TIMERx_CHCTL0)

As soon as selecting one input capture source by CHxMS, the channel must be set to input mode (CHxMS! =0x0) and TIMERx_CHxCV cannot be written any more.

Step4: Interrupt enable (CHxIE and CHxDEN in TIMERx_DMAINTEN)

Enable the related interrupt to get the interrupt and DMA request.

Step5: Capture enable (CHxEN in TIMERx_CHCTL2)

Result: When the wanted input signal is captured, TIMERx_CHxCV will be set by counter's value and CHxIF is asserted. If the CHxIF is 1, the CHxOF will also be asserted. The interrupt and DMA request will be asserted or not based on the configuration of CHxIE and CHxDEN in TIMERx_DMAINTEN.

Direct generation: A DMA request or interrupt is generated by setting CHxG directly.

The channel input capture function can be also used for pulse width measurement from signals on the TIMERx_CHx pins. For example, PWM signal connects to CI0 input. Select CI0 as channel 0 capture signals by setting CH0MS to 2'b01 in the channel control register (TIMERx_CHCTL0) and set capture on rising edge. Select CI0 as channel 1 capture signal by setting CH1MS to 2'b10 in the channel control register (TIMERx_CHCTL0) and set capture on falling edge. The counter is set to restart mode and is restarted on channel 0 rising edge. Then the TIMERX_CH0CV can measure the PWM period and the TIMERx_CH1CV can measure the PWM duty cycle.

Channel output compare function

Figure 14-48. Output compare logic (with complementary output, x=0,1,2,3)

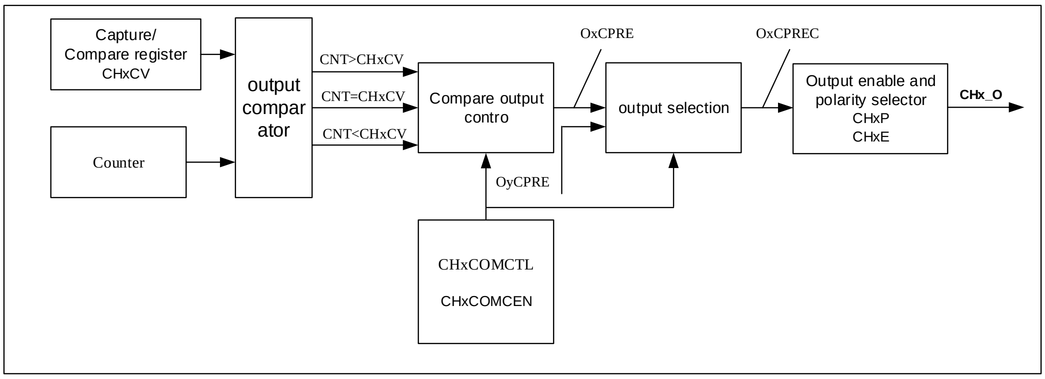


Figure 14-48. Output compare logic (with complementary output, x=0,1,2,3) shows the principle circuit of channels output compare function. The relationship between the channel output signal CHx_O and the OxCPRE signal (more details refer to *Channel output prepare signal*) is described as blew: The active level of O0CPRE is high, the output level of CH0_O depends on OxCPRE signal, CHxP bit and CH0P bit (please refer to the TIMERx_CHCTL2 register for more details). OxCPREC = OxCPRE when CHxCMOCTL[3:0] < 4'b1100, and combined PWM is output when CHxCMOCTL[3:0] = 4'b1100 or 4'b1101, and OxCPREC is the logic AND or OR of OxCPRE and OyCPRE(y = x+1 or x-1). Asymmetric PWM is output when CHxCMOCTL[3:0] = 4'b1110 or 4'b1111, the output of OxCPREC is OxCPRE or OyCPRE(y = x+1 or x-1)when counting up and the output of OxCPREC is OyCPRE or OyCPRE(y = x+1 or x-1).For example, configure CHxP=0 (the active level of CHx_O is high, the same as OxCPRE), CHxE=1 (the output of CHx_O is enabled):

If the output of OxCPRE is active(high) level, the output of CHx_O is active(high) level.

If the output of OxCPRE is inactive(low) level, the output of CHx_O is active(low) level.

In channel output compare function, the TIMERx can generate timed pulses with programmable position, polarity, duration and frequency. When the counter matches the value in the TIMERx_CHxCV register of an output compare channel, the channel (n) output can be set, cleared, or toggled based on CHxCOMCTL. When the counter reaches the value in the TIMERx_CHxCV register, the CHxIF bit will be set and the channel (n) interrupt is generated if CHxIE = 1. And the DMA request will be asserted, if CxCDE=1.

So, the process can be divided into several steps as below:

Step1: Clock configuration. Such as clock source, clock prescaler and so on.

Step2: Compare mode configuration.

- Set the shadow enable mode by CHxCOMSEN.
- Set the output mode (set/clear/toggle) by CHxCOMCTL.
- Select the active polarity by CHxP.
- Enable the output by CHxEN.

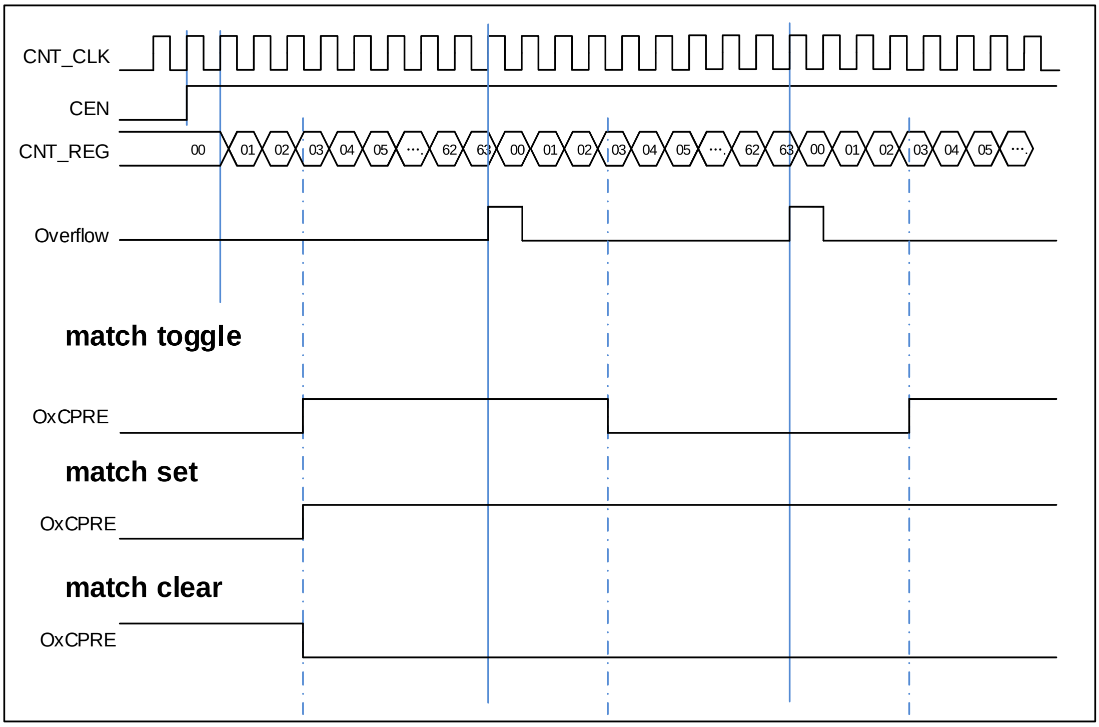
Step3: Interrupt/DMA-request enables configuration by CHxIE/CxCDE.

Step4: Compare output timing configuration by TIMERx_CAR and TIMERx_CHxCV.
The TIMERx_CHxCV can be changed onging to meet the expected waveform.

Step5: Start the counter by configuring CEN to 1.

The timing chart below shows the three compare modes toggle/set/clear. CAR=0x63, CHxVAL=0x3

Figure 14-49. Output-compare under three modes



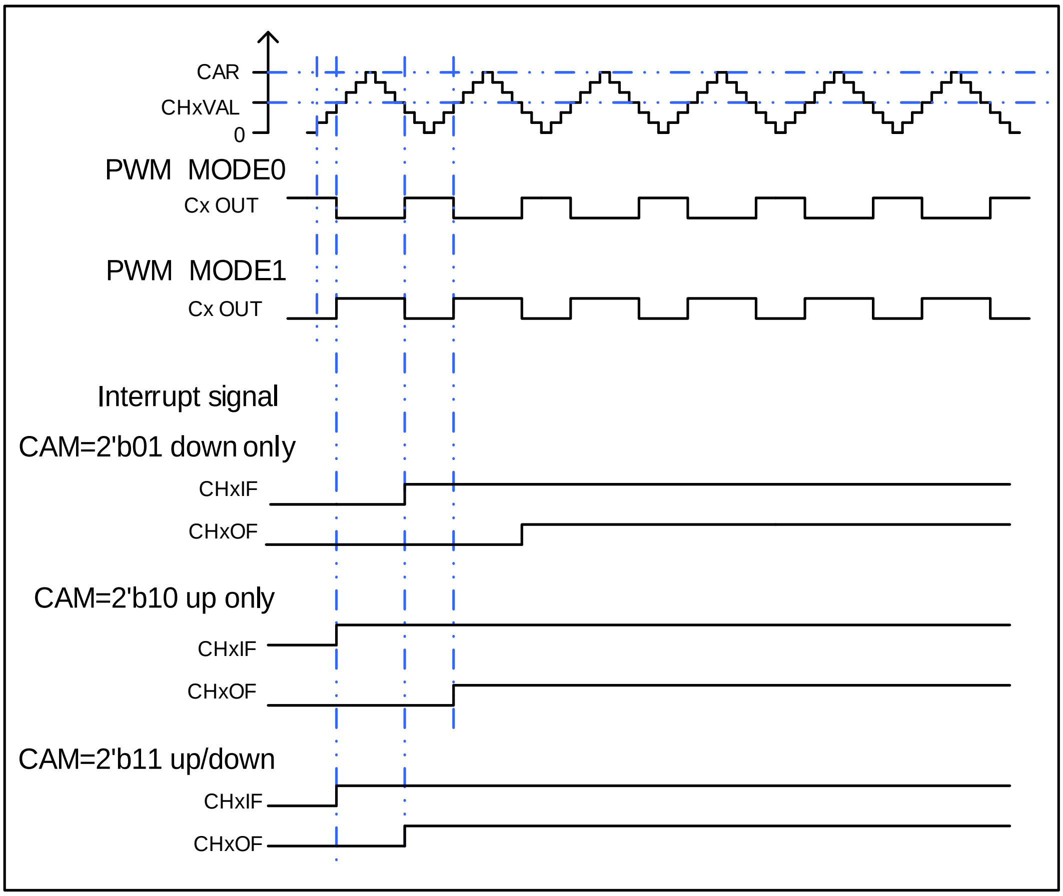
Output PWM function

In the output PWM function (by setting the CHxCOMCTL bit to 3'b110 (PWM mode 0) or to 3'b 111(PWM mode 1)), the channel can generate PWM waveform according to the TIMERx_CAR registers and TIMERx_CHxCV registers.

Based on the counter mode, PWM can also be divided into EAPWM (Edge-aligned PWM) and CAPWM (Center-aligned PWM).

The EAPWM's period is determined by TIMERx_CAR and the duty cycle is determined by TIMERx_CHxCV.*Figure 14-50. Timing chart of EAPWM* shows the EAPWM output and interrupts waveform.

The CAPWM period is determined by 2*TIMERx_CAR, and duty cycle is determined by 2*TIMERx_CHxCV. *Figure 14-51. Timing chart of CAPWM*



shows the CAPWM output and interrupts waveform.

In up counting mode, if the value of TIMERx_CHxCV is greater than the value of TIMERx_CAR, the output will be always inactive in PWM mode 0 (CHxCOMCTL=3'b110). And if the value of TIMERx_CHxCV is greater than the value of TIMERx_CAR, the output will be always active in PWM mode 1 (CHxCOMCTL=3'b111).

Figure 14-50. Timing chart of EAPWM

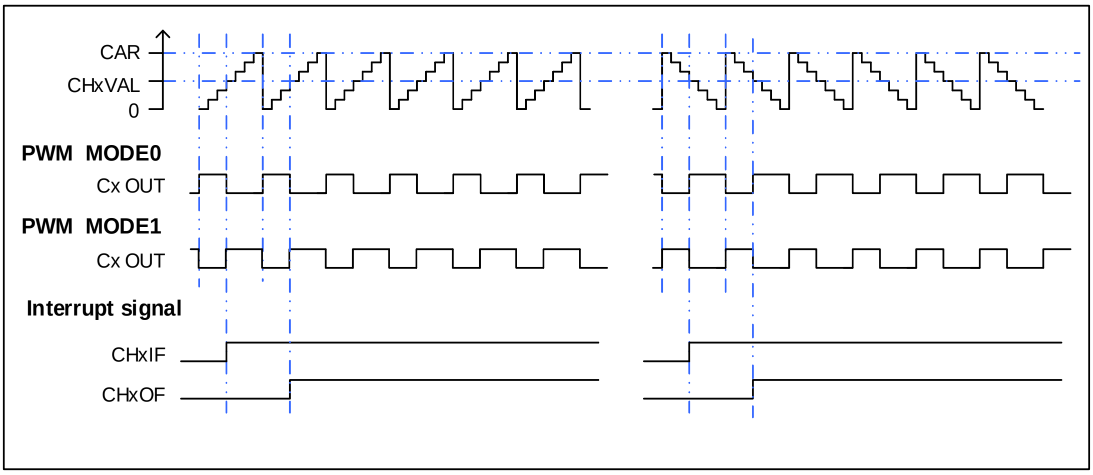
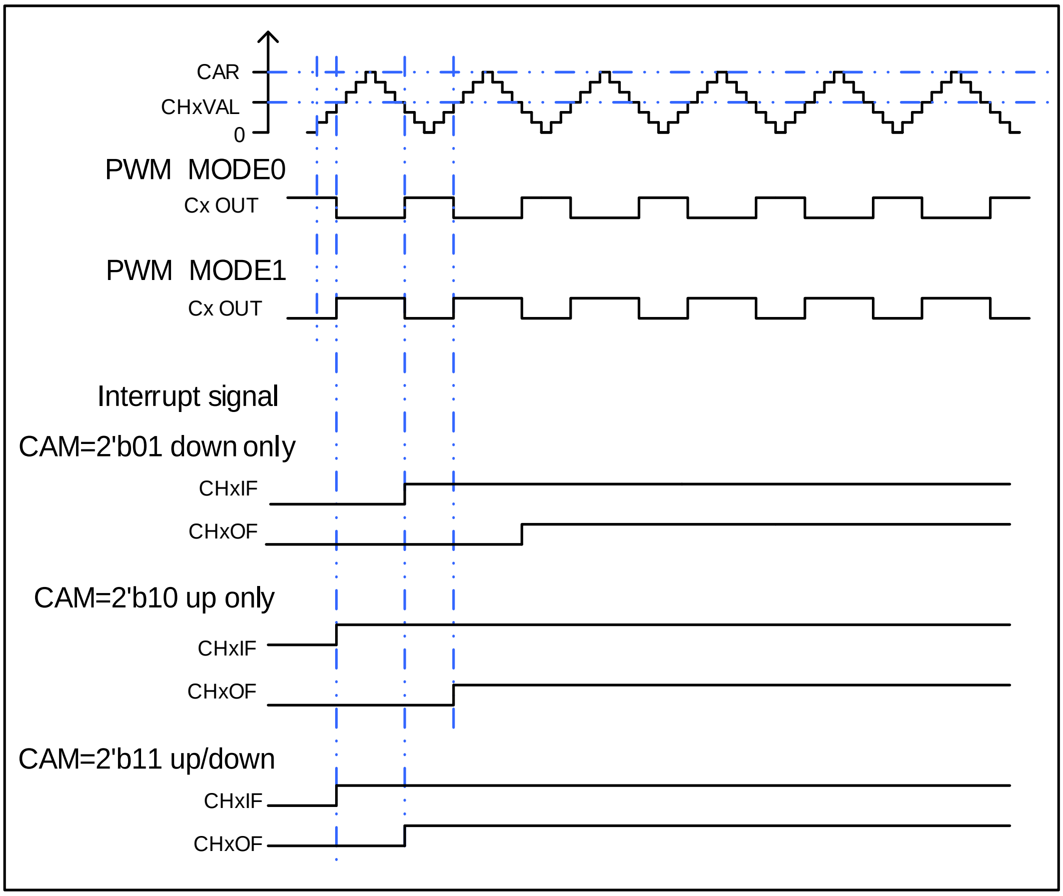


Figure 14-51. Timing chart of CAPWM



Asymmetric PWM mode

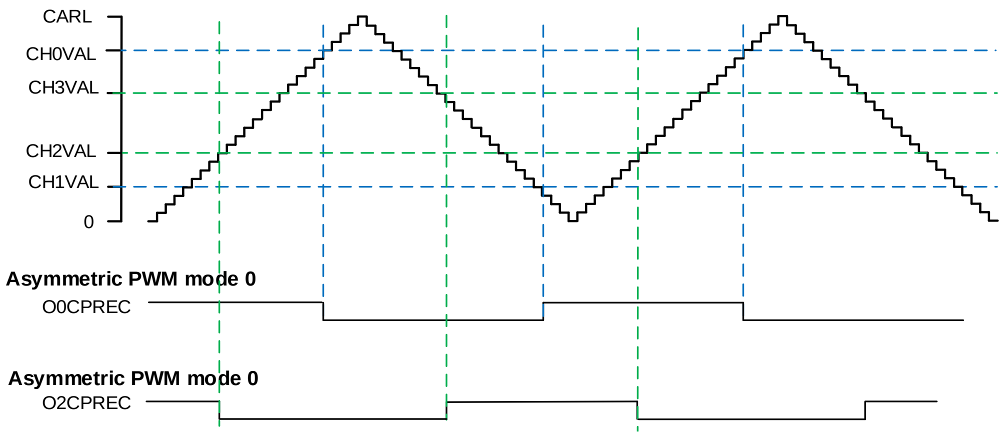
The asymmetric PWM mode 0 / 1 are (by setting the CHxCOMCTL[3:0] bit-field to 4'b1110 or 4'b1111) used in center-aligned PWM to generate a programmable phase shift. The CPWM's frequency is determined by TIMERx_CAR register, and the duty cycle and phase shift are determined by a pair of TIMER_CHxCV / TIMER_CH(x+1)CV (or TIMER_CH(x-1)CV) registers with adjacent offset addresses.

The TIMER_CHxCV register determines the waveform when counting up, and the TIMER_CH(x+1)CV (or TIMER_CH(x-1)CV) register determines the waveform when counting down. The details are as follows:

- The CPWM of O0CPREC / O1CPREC is determined by TIMER_CH0CV and TIMER_CH1CV registers.
- The CPWM of O2CPREC / O3CPREC is determined by TIMER_CH2CV and TIMER_CH3CV registers.

When using asymmetric PWM modes, CH0 / CH1 (or CH2 / CH3) can independently output different waveforms, which are configured by the CHxCOMCTL[3:0] bit-field (the two channels can be configured with different values).

Figure 14-52. O0CPREC and O2CPREC with asymmetric PWM mode



Combined PWM mode

The combined PWM mode 0 / 1 are (by setting the CHxCOMCTL[3:0] bit-field to 4’b1100 or 4’b1101) used in center-aligned PWM to generate a programmable phase shift. The CPWM’s frequency is determined by TIMERx_CAR register, and the duty cycle and phase shift are determined by a pair of TIMER_CHxCV / TIMER_CH(x+1)CV (or TIMER_CH(x-1)CV) registers with adjacent offset addresses.

- The CPWM of O0CPREC / O1CPREC is determined by TIMER_CH0CV and TIMER_CH1CV registers.
- The CPWM of O2CPREC / O3CPREC is determined by TIMER_CH2CV and TIMER_CH3CV registers.

When O0CPREC selects the combined PWM mode 0（CH0COMCTL[3:0] == 4'b1100）, and O0CPREC output signal which is the logic OR of O0CPRE and O1CPRE. When O0CPREC selects the combined PWM mode 1（CH0COMCTL[3:0] == 4'b1101）, and O0CPREC output signal which is the logic AND of O0CPRE and O1CPRE.

Retriggerable single pulse mode

This mode enables the counter to generate a pulse of programmable length in response to a trigger.

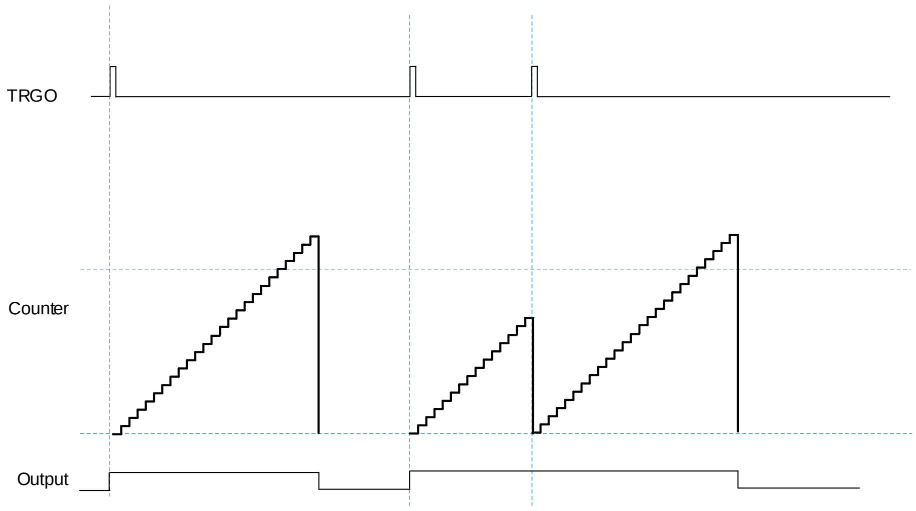
- The pulse begins immediately upon receiving a trigger (no delay).
- If a new trigger is received before the current pulse ends, the pulse duration is extended.

The timer must operate in slave mode, and set TSCFG7[2:0] in SYSCFG_TIMERxCFG0(x = 2), setting the CHxCOMCTL[3:0] bit-field to 4'b1100 or 4'b1101 to chose Retriggerable single pulse mode 0/1.

In up-counting mode, set TIMERx_CHyCV to 0; the pulse length is defined by the TIMERx_CAR register. in down counting mode, ensure TIMERx_CHyCV is great than or equal to TIMERx_CAR.

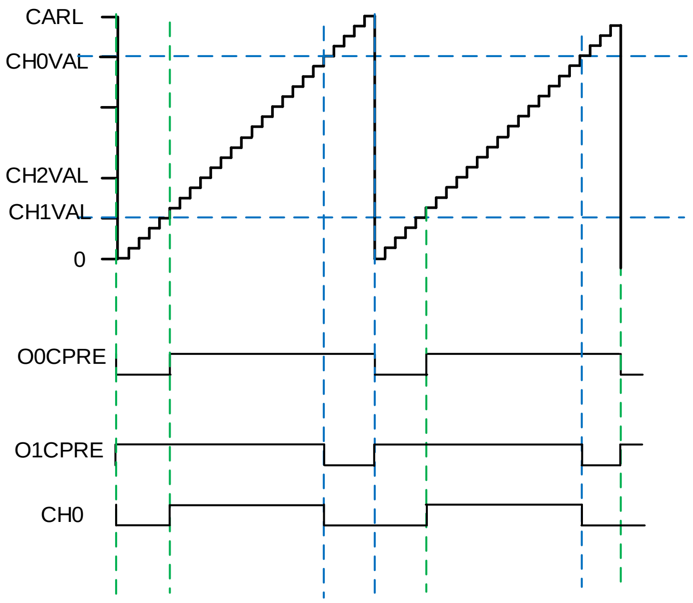
Note : This mode is incompatible with center-aligned PWM modes. Set CAM[1:0] = 2'b00 in TIMERx_CTL0.

Figure 14-53. Retriggerable single pulse mode



When using combined PWM modes, CH0 / CH1 (or CH2 / CH3) can independently output different waveforms, which are configured by the CHxCOMCTL[3:0] bit-field (the two channels can be configured with different values).

Figure 14-54. O0CPRE is combined PWM mode 1 and O1CPRE is PWM mode 0



Channel output prepare signal

As is shown in *Figure 14-48. Output compare logic (with complementary output, x=0,1,2,3)*, when TIMERx is configured in compare match output mode,a middle signal which is OxCPRE signal (Channel x output prepare signal) will be generated before the channel outputs signal. The OxCPRE signal type is defined by configuring the CHxCOMCTL bit. The OxCPRE signal has several types of output function. These include keeping the original level by configuring the CHxCOMCTL field to 0x00, setting to high by configuring the CHxCOMCTL field to 0x01, setting to low by configuring the CHxCOMCTL field to 0x02 or toggling signal by configuring the CHxCOMCTL field to 0x03 when the counter value matches the content of the TIMERx_CHxCV register.

The PWM mode 0/PWM mode 1 output is another output type of OxCPRE which is setup by configuring the CHxCOMCTL field to 0x06/0x07. In these modes, the OxCPRE signal level is changed according to the counting direction and the relationship between the counter value and the TIMERx_CHxCV content. Refer to the definition of relative bit for more details.

Another special function of the OxCPRE signal is a forced output which can be achieved by configuring the CHxCOMCTL field to 0x04/0x05. The output can be forced to an inactive/active level irrespective of the comparison condition between the values of the counter and the TIMERx_CHxCV.

Configure the CHxCOMCEN bit to 1 in the TIMERx_CHCTL0 register, the OxCPRE signal can be forced to 0 when the ETIFP signal derived from the external ETI pin is set to a high level. The OxCPRE signal will not return to its active level until the next update event occurs.

Quadrature decoder

The quadrature decoder function uses two quadrature inputs CI0 and CI1 derived from the TIMERx_CH0 and TIMERx_CH1 pins respectively to interact with each other to generate the counter value. Setting TSCFGy[2:0] != 3'b000 (y=0,1,2) to select that the counting direction of timer is determined only by the CI0, only by the CI1, or by the CI0 and the CI1. The DIR bit is modified during the voltage level change of each direction selection source. The mechanism of changing the counter direction is shown in *Table 14-3. Counting direction versus quadrature decoder signals.* The quadrature decoder can be regarded as an external clock with a direction selection. This means that the counter counts continuously from 0 to the counter-reload value. Therefore, users must configure the TIMERx_CAR register before the counter starts to count.

Table 14-5. Counting direction versus quadrature decoder signals

Counting mode	Level	CI0FE0		CI1FE1	
		Rising	Falling	Rising	Falling
CI0 only counting	CI1FE1=High	Down	Up	-	-
	CI1FE1=Low	Up	Down	-	-
CI1 only counting	CI0FE0=High	-	-	Up	Down
	CI0FE0=Low	-	-	Down	Up

Counting mode	Level	CI0FE0		CI1FE1	
		Rising	Falling	Rising	Falling
CI0 and CI1 counting	CI1FE1=High	Down	Up	X	X
	CI1FE1=Low	Up	Down	X	X
	CI0FE0=High	X	X	Up	Down
	CI0FE0=Low	X	X	Down	Up

Note: "-" means "no counting"; "X" means impossible.

Figure 14-55. Example of counter operation in quadrature decoder interface mode

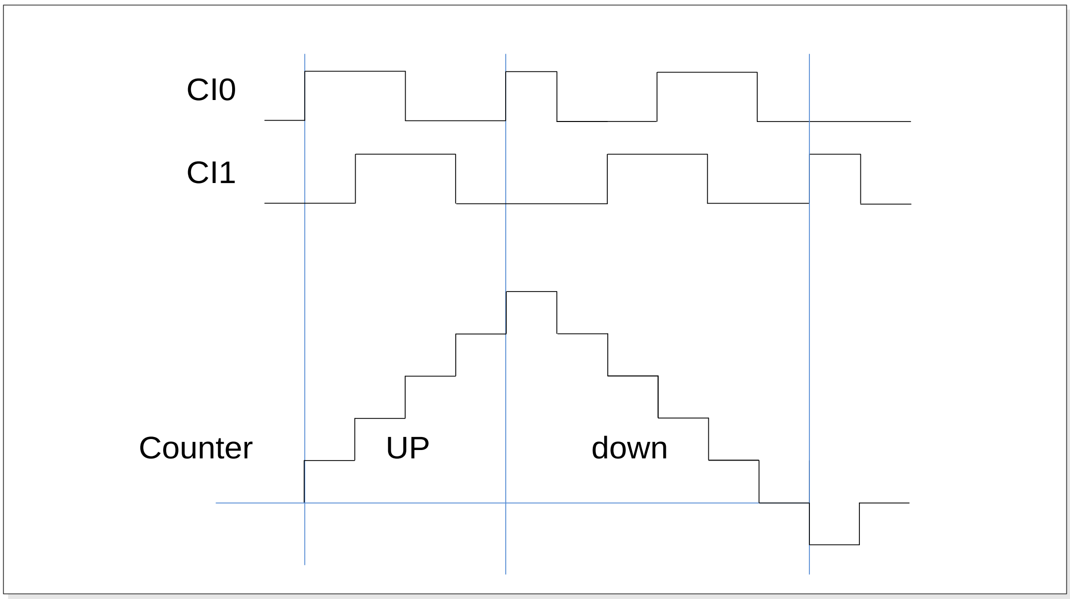
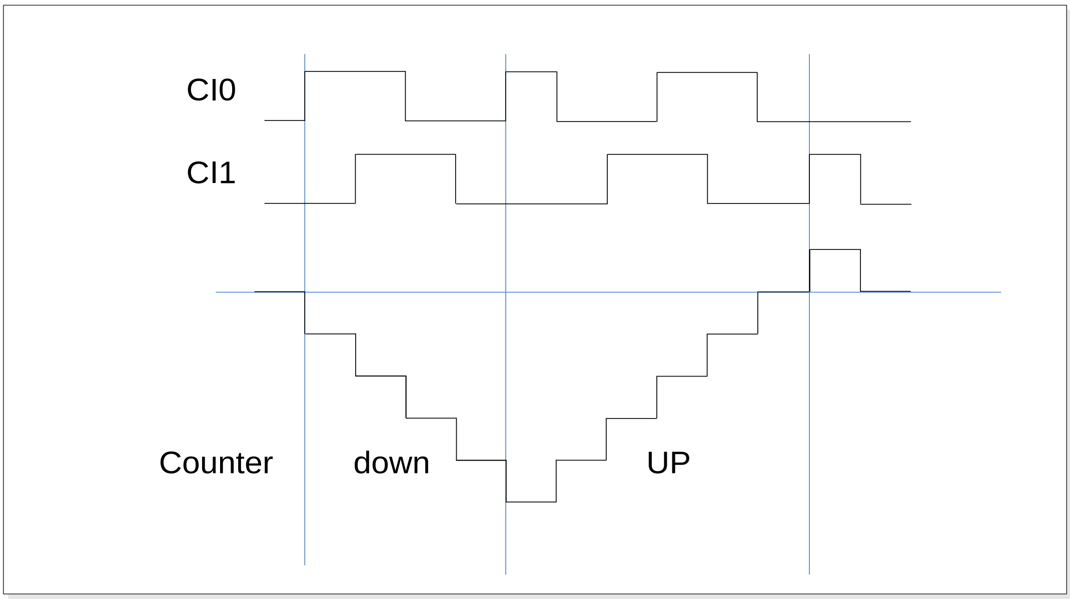


Figure 14-56. Example of quadrature decoder interface mode with CI0FE0 polarity inverted



Hall sensor function

Hall sensor is generally used to control BLDC Motor; the general level0 timer can support this function.

Each of the 3 HALL sensors provides a pulse that applied to an input capture pin, can then be analyzed and both speed and position can be deduced.

Enable XOR by setting TI0S, then, each of input signal change will make the CI0 toggle. CH0VAL will record the value of counter at that moment.

Master-slave management

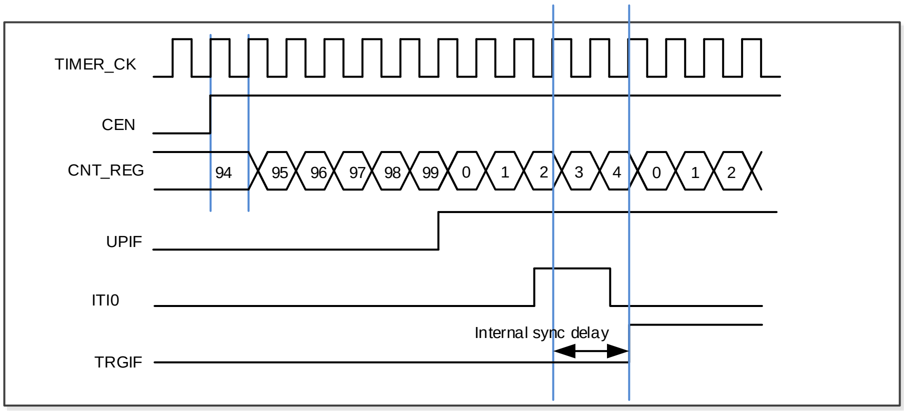
The TIMERx can be synchronized with a trigger in several modes including the restart mode, the pause mode and the event mode which is selected by the SMC [2:0] in the TIMERx_SMCFG register. The trigger input of these modes can be selected by the TRGS [2:0] in the TIMERx_SMCFG register.

Table 14-6. Slave mode example table

	Mode Selection	Source Selection	Polarity Selection	Filter and Prescaler
LIST	SMC[2:0] 3'b100 (restart mode) 3'b101 (pause mode) 3'b110 (event mode)	TRGS[2:0] 000: ITI0 001: ITI1 010: ITI2 011: ITI3 100: CI0F_ED 101: CI0FE0 110: CI1FE1 111: ETIFP	If you choose the CI0FE0 or CI1FE1, configure the CHxP and CHxNP for the polarity selection and inversion. If you choose the ETIF, configure the ETP for polarity selection and inversion.	For the ITIx no filter and prescaler can be used. For the CIx, configure Filter by CHxCAPFLT, no prescaler can be used. For the ETIF, configure Filter by ETFC and Prescaler by ETPSC.
Exam1	Restart mode The counter can be clear and restart when a rising trigger input.	TRGS[2:0]=3’b000 ITI0 is the selection.	For ITI0, no polarity selector can be used.	For the ITI0, no filter and prescaler can be used.

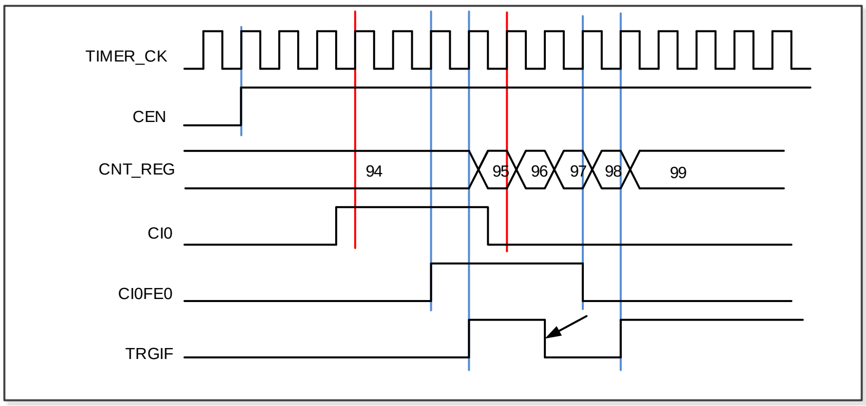
<table>
<tr><th></th><th>Mode Selection</th><th>Source Selection</th><th>Polarity Selection</th><th>Filter and Prescaler</th></tr>
<tr><td></td><td colspan="4">

Figure 14-57. Restart mode

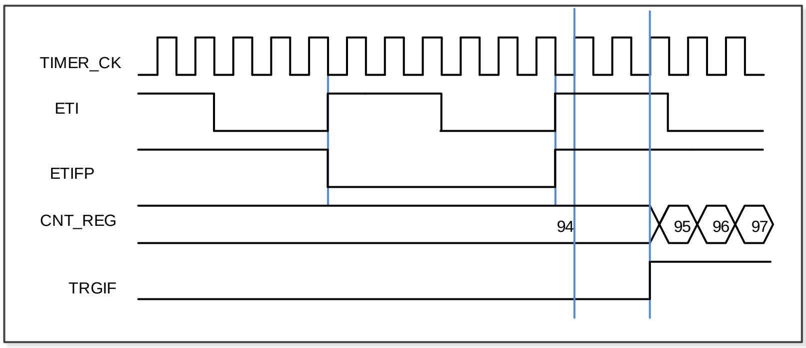
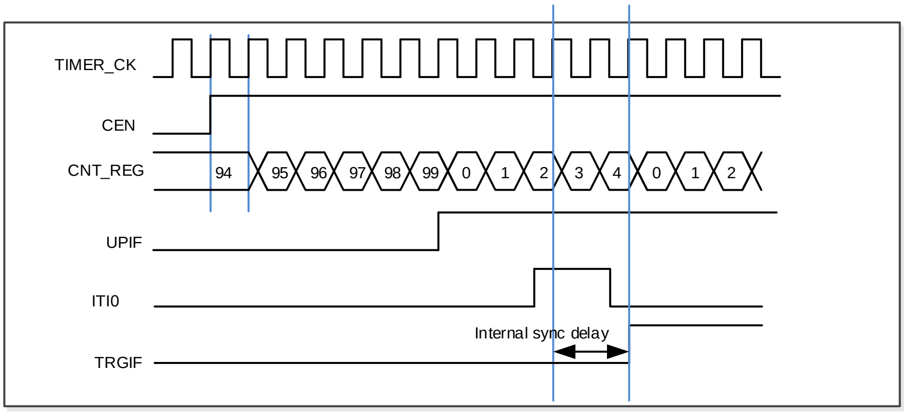


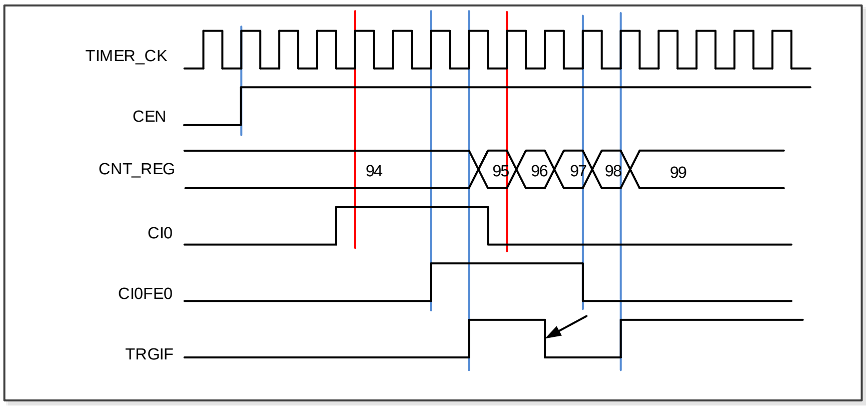
</td></tr>
<tr><td rowspan="2">Exam2</td><td>Pause mode
The counter can be paused when the trigger input is low.</td><td>TRGS[2:0]=3’b101
CI0FE0 is the selection.</td><td>TI0S=0(Non-xor)
[CH0NP==0, CH0P==0] no inverted. Capture will be sensitive to the rising edge only.</td><td>Filter is bypass in this example.</td></tr>
<tr><td colspan="4">

Figure 14-58. Pause mode

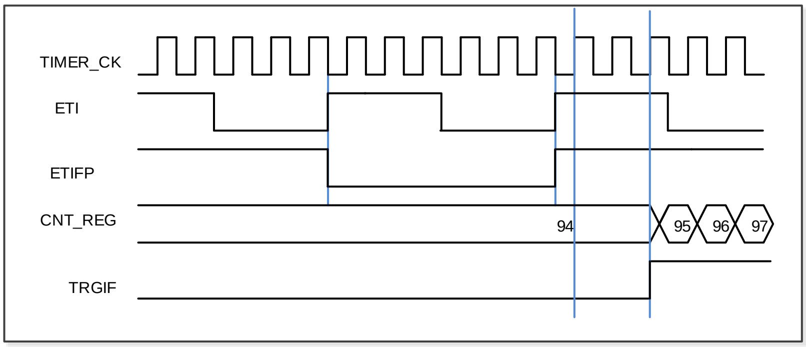


</td></tr>
<tr><td>Exam3</td><td>Event mode
The counter will start to count when a rising trigger input.</td><td>TRGS[2:0]=3’b111
ETIF is the selection.</td><td>ETP = 0 no polarity change.</td><td>ETPSC = 1, divided by 2.
ETFC = 0, no filter</td></tr>
</table>

	Mode Selection	Source Selection	Polarity Selection	Filter and Prescaler
	Figure 14-59. Event mode  TIMER_CK ETI ETIFP CNT_REG TRGIF 94 95 96 97			
	Mode Selection	**Source Selection**	**Polarity Selection**	**Filter and Prescaler**
LIST	SMC[2:0] 3'b100 (restart mode) 3'b101 (pause mode) 3'b110 (event mode)	TRGS[2:0] 000: ITI0 001: ITI1 010: ITI2 011: ITI3 100: CI0F_ED 101: CI0FE0 110: CI1FE1 111: ETIFP	If you choose the CI0FE0 or CI1FE1, configure the CHxP and CHxNP for the polarity selection and inversion. If you choose the ETIF, configure the ETP for polarity selection and inversion.	For the ITIx no filter and prescaler can be used. For the CIx, configure Filter by CHxCAPFLT, no prescaler can be used. For the ETIF, configure Filter by ETFC and Prescaler by ETPSC.
Exam1	Restart mode The counter can be clear and restart when a rising trigger input.	TRGS[2:0]=3'b000 ITI0 is the selection.	For ITI0, no polarity selector can be used.	For the ITI0, no filter and prescaler can be used.
	Figure 14-60. Restart mode  TIMER_CK CEN CNT_REG 94 95 96 97 98 99 0 1 2 3 4 0 1 2 UPIF ITI0 Internal sync delay TRGIF			
Exam2	Pause mode The counter can be paused when	TRGS[2:0]=3'b101 CI0FE0 is the	TI0S=0(Non-xor) [CH0NP==0, CH0P==0] no inverted. Capture will	Filter is bypass in this example.

	Mode Selection	Source Selection	Polarity Selection	Filter and Prescaler
	the trigger input is low.	selection.	be sensitive to the rising edge only.	
	Figure 14-61. Pause mode  TIMER_CK CEN CNT_REG 94 95 96 97 98 99 CI0 CI0FE0 TRGIF			

Exam3	Event mode
	The counter will start to count when a rising trigger input.
		TRGS[2:0]=3'b111
		ETIF is the selection.
			ETP = 0 no polarity change.	ETPSC = 1, divided by 2.
				ETFC = 0, no filter

	Figure 14-62. Event mode  TIMER_CK ETI ETIFP CNT_REG 94 95 96 97 TRGIF			

Single pulse mode

Single pulse mode is opposite to the repetitive mode, which can be enabled by setting SPM in TIMERx_CTL0. When you set SPM, the counter will be clear and stop when the next update event. In order to get pulse waveform, you can set the TIMERx to PWM mode or compare by CHxCOMCTL.

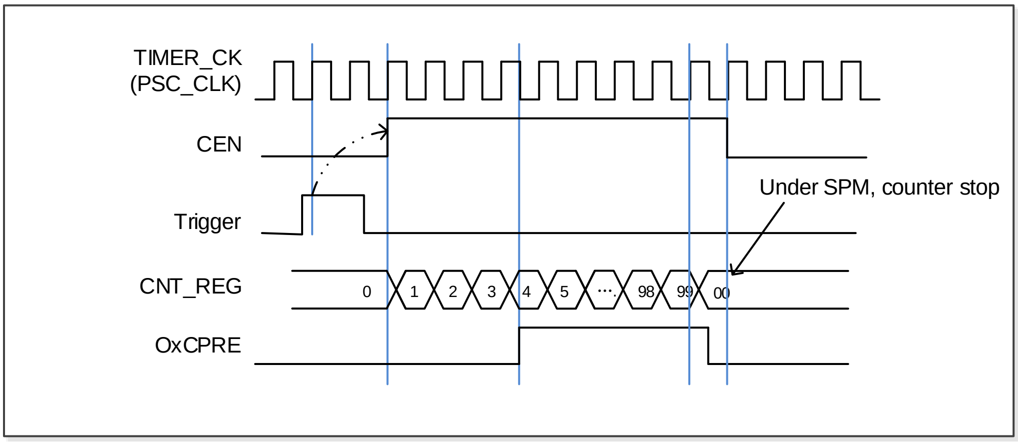
Once the timer is set to operate in the single pulse mode, it is not necessary to set the timer enable bit CEN in the TIMERx_CTL0 register to 1 to enable the counter. The trigger to generate a pulse can be sourced from the trigger signals edge or by setting the CEN bit to 1 using software. Setting the CEN bit to 1 or a trigger from the trigger signals edge can generate a pulse and then keep the CEN bit at a high state until the update event occurs or the CEN bit is written to 0 by software. If the CEN bit is cleared to 0 using software, the counter will be stopped and its value held.

In the single pulse mode, the trigger active edge which sets the CEN bit to 1 will enable the counter. However, there exist several clock delays to perform the comparison result between the counter value and the TIMERx_CHxCV value. In order to reduce the delay to a minimum value, the user can set the CHxCOMFEN bit in each TIMERx_CHCTL0/1 register. After a trigger rising occurs in the single pulse mode, the OxCPRE signal will immediately be forced to the state which the OxCPRE signal will change to, as the compare match event occurs without taking the comparison result into account. The CHxCOMFEN bit is available only when the output channel is configured to operate in the PWM0 or PWM1 output mode and the trigger source is derived from the trigger signal.

Figure 14-63. Single pulse mode TIMERx_CHxCV = 4 TIMERx_CAR=99

shows an example.

Figure 14-63. Single pulse mode TIMERx_CHxCV = 4 TIMERx_CAR=99



Timers interconnection

Refer to ***Timers interconnection.***

Timer DMA mode

DMA mode is the function that configures timer’s register by DMA module. The relative registers are TIMERx_DMACFG and TIMERx_DMATB. Corresponding DMA request bit should be asserted to enable DMA request for internal interrupt event. TIMERx will send a request to DMA when the interrupt event occurs. DMA is configured to M2P (memory to peripheral) mode and the address of TIMERx_DMATB is configured to PADDR (peripheral base address), then DMA will access the TIMERx_DMATB. In fact, TIMERx_DMATB register is only a buffer, timer will map the TIMERx_DMATB to an internal register, appointed by the field of DMATA in TIMERx_DMACFG. If the field of DMATC in TIMERx_DMACFG is 0 (1 transfer), the timer sends only one DMA request. While if TIMERx_DMATC is not 0, such as 3 (4 transfers), then timer will send 3 more requests to DMA, and DMA will access timer’s registers DMATA+0x4, DMATA+0x8 and DMATA+0xC at the next 3 accesses to TIMERx_DMATB. In a word, one-time DMA internal interrupt event asserts, (DMATC+1) times request will be sent by TIMERx.

If one more DMA request event occurs, TIMERx will repeat the process above.

Timer debug mode

When the Cortex®-M23 is halted, and the TIMERx_HOLD configuration bit in DBG_CTL0 register set to 1, the TIMERx counter stops.

14.2.5. TIMERx registers(x=2)

TIMER2 base address: 0x4000 0400

Control register 0 (TIMERx_CTL0)

Address offset: 0x00

Reset value: 0x0000 0000

This register has to be accessed by word(32-bit)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved						CKDIV[1:0]		ARSE	CAM[1:0]		DIR	SPM	UPS	UPDIS	CEN
						rw		rw	rw		rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31:10	Reserved	Must be kept at reset value
9:8	CKDIV[1:0]	Clock division The CKDIV bits can be configured by software to specify division factor between the CK_TIMER and the dead-time and digital filter sample clock (DTS). 00: $f_{DTS}=f_{CK\_TIMER}$ 01: $f_{DTS}= f_{CK\_TIMER}$ /2 10: $f_{DTS}= f_{CK\_TIMER}$ /4 11: Reserved
7	ARSE	Auto-reload shadow enable 0: The shadow register for TIMERx_CAR register is disabled 1: The shadow register for TIMERx_CAR register is enabled
6:5	CAM[1:0]	Counter aligns mode selection 00: No center-aligned mode (edge-aligned mode). The direction of the counter is specified by the DIR bit. 01: Center-aligned and counting down assert mode. The counter counts under center-aligned and channel is configured in output mode (CHxMS=00 in TIMERx_CHCTL0 register). Only when counting down, CHxF bit can be set. 10: Center-aligned and counting up assert mode. The counter counts under center-aligned and channel is configured in output mode (CHxMS=00 in TIMERx_CHCTL0 register). Only when counting up, CHxF bit can be set. 11: Center-aligned and counting up/down assert mode. The counter counts under center-aligned and channel is configured in output mode (CHxMS=00 in TIMERx_CHCTL0 register). Both when counting up and counting down, CHxF bit can be set.

		After the counter is enabled, cannot be switched from 0x00 to non 0x00.
4	DIR	Direction 0: Count up 1: Count down If the timer work in center-aligned mode or quadrature decoder mode, this bit is read only.
3	SPM	Single pulse mode. 0: Single pulse mode disable. The counter continues after update event. 1: Single pulse mode enable. The counter counts until the next update event occurs.
2	UPS	Update source This bit is used to select the update event sources by software. 0: These events generate update interrupts or DMA requests: The UPG bit is set The counter generates an overflow or underflow event The restart mode generates an update event. 1: This event generates update interrupts or DMA requests: The counter generates an overflow or underflow event
1	UPDIS	Update disable. This bit is used to enable or disable the update event generation. 0: Update event enable. When an update event occurs, the corresponding shadow registers are loaded with their preloaded values. These events generate update event: The UPG bit is set The counter generates an overflow or underflow event The restart mode generates an update event. 1: Update event disable. **Note:** When this bit is set to 1, setting UPG bit or the restart mode does not generate an update event, but the counter and prescaler are initialized.
0	CEN	Counter enable 0: Counter disable 1: Counter enable The CEN bit must be set by software when timer works in external clock, pause mode and quadrature decoder mode.

Control register 1 (TIMERx_CTL1)

Address offset: 0x04

Reset value: 0x0000 0000

This register has to be accessed by word(32-bit)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								TI0S	MMC[2:0]			DMAS	Reserved		
								rw	rw			rw			

Bits	Fields	Descriptions
31:8	Reserved	Must be kept at reset value
7	TI0S	Channel 0 trigger input selection 0: The TIMERx_CH0 pin input is selected as channel 0 trigger input. 1: The result of combinational XOR of TIMERx_CH0, CH1 and CH2 pins is selected as channel 0 trigger input.
6:4	MMC[2:0]	Master mode control These bits control the selection of TRGO signal, which is sent in master mode to slave timers for synchronization function. 000: When a counter reset event occurs, a TRGO trigger signal is output. The counter resert source: Master timer generate a reset the UPG bit in the TIMERx_SWEVG register is set 001: Enable. When a conter start event occurs, a TRGO trigger signal is output. The counter start source : CEN control bit is set The trigger input in pause mode is high 010: When an update event occurs, a TRGO trigger signal is output. The update source depends on UPDIS bit and UPS bit. 011: When a capture or compare pulse event occurs in channel0, a TRGO trigger signal is output. 100: When a compare event occurs, a TRGO trigger signal is output. The compare source is from O0CPRE. 101: When a compare event occurs, a TRGO trigger signal is output. The compare source is from O1CPRE. 110: When a compare event occurs, a TRGO trigger signal is output. The compare source is from O2CPRE. 111: When a compare event occurs, a TRGO trigger signal is output. The compare source is from O3CPRE.
3	DMAS	DMA request source selection 0: When capture or compare event occurs, the DMA request of channel x is sent 1: When update event occurs, the DMA request of channel x is sent.
2:0	Reserved	Must be kept at reset value.

Slave mode configuration register (TIMERx_SMCFG)

Address offset: 0x08

Reset value: 0x0000 0000

This register has to be accessed by word(32-bit)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ETP	SMC1	ETPSC[1:0]		ETFC[3:0]				MSM	Reserved						
rw	rw	rw		rw				rw							

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15	ETP	External trigger polarity This bit specifies the polarity of ETI signal 0: ETI is active at rising edge or high level . 1: ETI is active at falling edge or low level .
14	SMC1	Part of SMC for enable External clock mode1 In external clock mode 1, the counter is clocked by any active edge on the ETIF signal. 0: External clock mode 1 disabled 1: External clock mode 1 enabled. When the slave mode is configured as restart mode, pause mode or event mode, the timer can still work in the external clock 1 mode by setting this bit. But the TRGS bits must not be 3'b111 in this case. The clock source of the timer will be ETIFP if external clock mode 0 and external clock mode 1 are configured at the same time. **Note:** External clock mode 0 enable is in this register's SMC[2:0] bit-filed.
13:12	ETPSC[1:0]	The prescaler of external trigger The frequency of external trigger signal ETIFP must not be at higher than 1/4 of TIMER_CK frequency. When the external trigger signal is a fast clock, the prescaler can be enabled to reduce ETIFP frequency. 00: Prescaler disable. 01: The prescaler is 2. 10: The prescaler is 4. 11: The prescaler is 8.
11:8	ETFC[3:0]	External trigger filter control The external trigger can be filtered by digital filter and this bit-field configure the filtering capability. Basic principle of digital filter: continuously sample the external trigger signal according to f_{SAMP} and record the number of times of the same level of the signal. After reaching the filtering capacity configured by this bit-field, it is considered to be an effective level. The filtering capability configuration is as follows:

EXTFC[3:0]	Times	f_{SAMP}
4’b0000	Filter disabled.	
4’b0001	2	f_{TIMER_CK}
4’b0010	4	
4’b0011	8	
4’b0100	6	$f_{DTS\_CK}/2$
4’b0101	8	
4’b0110	6	$f_{DTS\_CK}/4$
4’b0111	8	
4’b1000	6	$f_{DTS\_CK}/8$
4’b1001	8	
4’b1010	5	$f_{DTS\_CK}/16$
4’b1011	6	
4’b1100	8	
4’b1101	5	$f_{DTS\_CK}/32$
4’b1110	6	
4’b1111	8	

7	MSM	Master-slave mode This bit can be used to synchronize selected timers to begin counting at the same time. The TRGI is used as the start event, and through TRGO, timers are connected together. 0: Master-slave mode disable 1: Master-slave mode enable
6:0	Reserved	Must be kept at reset value.

DMA and interrupt enable register (TIMERx_DMAINTEN)

Address offset: 0x0C

Reset value: 0x0000 0000

This register has to be accessed by word(32-bit)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	TRGDEN	Reserved	CH3DEN	CH2DEN	CH1DEN	CH0DEN	UPDEN	Reserved	TRGIE	Reserved	CH3IE	CH2IE	CH1IE	CH0IE	UPIE
	rw		rw	rw	rw	rw	rw		rw		rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31:15	Reserved	Must be kept at reset value.
14	TRGDEN	Trigger DMA request enable 0: disabled

		1: enabled
13	Reserved	Must be kept at reset value.
12	CH3DEN	Channel 3 capture/compare DMA request enable 0: disabled 1: enabled
11	CH2DEN	Channel 2 capture/compare DMA request enable 0: disabled 1: enabled
10	CH1DEN	Channel 1 capture/compare DMA request enable 0: disabled 1: enabled
9	CH0DEN	Channel 0 capture/compare DMA request enable 0: disabled 1: enabled
8	UPDEN	Update DMA request enable 0: disabled 1: enabled
7	Reserved	Must be kept at reset value.
6	TRGIE	Trigger interrupt enable 0: disabled 1: enabled
5	Reserved	Must be kept at reset value.
4	CH3IE	Channel 3 capture/compare interrupt enable 0: disabled 1: enabled
3	CH2IE	Channel 2 capture/compare interrupt enable 0: disabled 1: enabled
2	CH1IE	Channel 1 capture/compare interrupt enable 0: disabled 1: enabled
1	CH0IE	Channel 0 capture/compare interrupt enable 0: disabled 1: enabled
0	UPIE	Update interrupt enable 0: disabled 1: enabled

Interrupt flag register (TIMERx_INTF)

Address offset: 0x10

Reset value: 0x0000 0000

This register has to be accessed by word(32-bit)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved			CH3OF	CH2OF	CH1OF	CH0OF	Reserved		TRGIF	Reserved	CH3IF	CH2IF	CH1IF	CH0IF	UPIF
			rc_w0	rc_w0	rc_w0	rc_w0			rc_w0		rc_w0	rc_w0	rc_w0	rc_w0	rc_w0

Bits	Fields	Descriptions
31:13	Reserved	Must be kept at reset value.
12	CH3OF	Channel 3 over capture flag Refer to CH0OF description
11	CH2OF	Channel 2 over capture flag Refer to CH0OF description
10	CH1OF	Channel 1 over capture flag Refer to CH0OF description
9	CH0OF	Channel 0 over capture flag When channel 0 is configured in input mode, this flag is set by hardware when a capture event occurs while CH0IF flag has already been set. This flag is cleared by software. 0: No over capture interrupt occurred 1: Over capture interrupt occurred
8:7	Reserved	Must be kept at reset value.
6	TRGIF	Trigger interrupt flag This flag is set on trigger event and cleared by software. When in pause mode, both edges on trigger input generates a trigger event, otherwise, only an active edge on trigger input can generates a trigger event. 0: No trigger event occurred. 1: Trigger interrupt occurred.
5	Reserved	Must be kept at reset value.
4	CH3IF	Channel 3 ‘s capture/compare interrupt enable Refer to CH0IF description
3	CH2IF	Channel 2 ‘s capture/compare interrupt enable Refer to CH0IF description
2	CH1IF	Channel 1 ‘s capture/compare interrupt flag

		Refer to CH0IF description
1	CH0IF	Channel 0 ‘s capture/compare interrupt flag This flag is set by hardware and cleared by software. When channel 0 is in input mode, this flag is set when a capture event occurs. When channel 0 is in output mode, this flag is set when a compare event occurs. 0: No Channel 1 interrupt occurred 1: Channel 1 interrupt occurred
0	UPIF	Update interrupt flag This bit is set by hardware on an update event and cleared by software. 0: No update interrupt occurred 1: Update interrupt occurred

Software event generation register (TIMERx_SWEVG)

Address offset: 0x14

Reset value: 0x0000 0000

This register has to be accessed by word(32-bit)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved									TRGG	Reserved	CH3G	CH2G	CH1G	CH0G	UPG
									w		w	w	w	w	w

Bits	Fields	Descriptions
31:7	Reserved	Must be kept at reset value.
6	TRGG	Trigger event generation This bit is set by software and cleared by hardware automatically. When this bit is set, the TRGIF flag in TIMERx_STAT register is set, related interrupt or DMA transfer can occur if enabled. 0: No generate a trigger event 1: Generate a trigger event
5	Reserved	Must be kept at reset value.
4	CH3G	Channel 3’s capture or compare event generation Refer to CH0G description
3	CH2G	Channel 2’s capture or compare event generation Refer to CH0G description
2	CH1G	Channel 1’s capture or compare event generation Refer to CH0G description

1	CH0G	Channel 0's capture or compare event generation This bit is set by software in order to generate a capture or compare event in channel 0, it is automatically cleared by hardware. When this bit is set, the CH1IF flag is set, the corresponding interrupt or DMA request is sent if enabled. In addition, if channel 1 is configured in input mode, the current value of the counter is captured in TIMERx_CH0CV register, and the CH0OF flag is set if the CH0IF flag was already high. 0: No generate a channel 1 capture or compare event 1: Generate a channel 1 capture or compare event
0	UPG	Update event generation This bit can be set by software, and cleared by hardware automatically. When this bit is set, the counter is cleared if the center-aligned or up counting mode is selected, else (down counting) it takes the auto-reload value. The prescaler counter is cleared at the same time. 0: No generate an update event 1: Generate an update event

Channel control register 0 (TIMERx_CHCTL0)

Address offset: 0x18

Reset value: 0x0000 0000

This register has to be accessed by word(32-bit)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved							CH1COMCTL[3]	Reserved							CH0COMCTL[3]

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH1COMCEN	CH1COMCTL[2:0]			CH1COMSEN	CH1COMFEN	CH1MS[1:0]		CH0COMCEN	CH0COMCTL[2:0]			CH0COMSEN	CH0COMFEN	CH0MS[1:0]	
CH1CAPFLT[3:0]				CH1CAPPSC[1:0]				CH0CAPFLT[3:0]				CH0CAPPSC[1:0]			
Rw				rw		rw		rw				rw		rw	

Output compare mode:

Bits	Fields	Descriptions
31:25	Reserved	Must be kept at reset value
24	CH1COMCTL[3]	Refer to CH1COMCTL[2:0].
23:17	Reserved	Must be kept at reset value
16	CH0COMCTL[3]	Refer to CH0COMCTL[2:0].
15	CH1COMCEN	Channel 1 output compare clear enable Refer to CH0COMCEN description
14:12	CH1COMCTL[2:0]	Channel 1 compare output control

		Refer to CH0COMCTL description
11	CH1COMSEN	Channel 1 output compare shadow enable Refer to CH0COMSEN description
10	CH1COMFEN	Channel 1 output compare fast enable Refer to CH0COMFEN description
9:8	CH1MS[1:0]	Channel 1 mode selection This bit-field specifies the direction of the channel and the input signal selection. This bit-field is writable only when the channel is not active. (CH1EN bit in TIMERx_CHCTL2 register is reset). 00: Channel 1 is programmed as output mode 01: Channel 1 is programmed as input mode, IS1 is connected to CI1FE1 10: Channel 1 is programmed as input mode, IS1 is connected to CI0FE1 11: Channel 1 is programmed as input mode, IS1 is connected to ITS. This mode is working only if an internal trigger input is selected, through TSCFGx[2:0](x = 3,4,5,6,7) bit-field in SYSCFG_TIMER2CFG (x=0) register.
7	CH0COMCEN	Channel 0 output compare clear enable. When this bit is set, if the ETIFP signal is detected as high level, the O0CPRE signal will be cleared. 0: Channel 0 output compare clear disable 1: Channel 0 output compare clear enable
6:4	CH0COMCTL[2:0]	Channel 0 compare output control This bit-field controls the behavior of the output reference signal O0CPRE which drives CH0_O and CH0_ON. O0CPRE is active high, while CH0_O and CH0_ON active level depends on CH0P and CH0NP bits. 0000: Frozen. The O0CPRE signal keeps stable, independent of the comparison between the register TIMERx_CH0CV and the counter TIMERx_CNT. 0001: Set the channel output. O0CPRE signal is forced high when the counter matches the output compare register TIMERx_CH0CV. 0010: Clear the channel output. O0CPRE signal is forced low when the counter matches the output compare register TIMERx_CH0CV. 0011: Toggle on match. O0CPRE toggles when the counter matches the output compare register TIMERx_CH0CV. 0100: Force low. O0CPRE is forced low level. 0101: Force high. O0CPRE is forced high level. 0110: PWM mode0. When counting up, O0CPRE is active as long as the counter is smaller than TIMERx_CH0CV else inactive. When counting down, O0CPRE is inactive as long as the counter is larger than TIMERx_CH0CV else active. 0111: PWM mode1. When counting up, O0CPRE is inactive as long as the counter is smaller than TIMERx_CH0CV else active. When counting down, O0CPRE is active as long as the counter is larger than TIMERx_CH0CV else inactive. When configured in PWM mode, the O0CPRE level changes only when the output compare mode switches from “frozen” mode to “PWM” mode or when the result of

		the comparison changes. 1000: Retriggerable single pulse mode 0. The behavior of O0CPRE is performed as in PWM mode 0. When counting up, the O0CPRE is active. When an trigger event occurs, the O0CPRE is inactive. The O0CPRE is active again at the next update event; When counting down, the O0CPRE is inactive, When an trigger event occurs, the O0CPRE is active. The O0CPRE is inactive again at the next update event. 1001: Retriggerable single pulse mode 1. The behavior of O0CPRE is performed as in PWM mode 1. When counting up, the O0CPRE is inactive, When an trigger event occurs,the O0CPRE is active. The O0CPRE is inactive again at the next update event; When counting down, the O0CPRE is active. When an trigger event occurs, the O0CPRE is inactive.The O0CPRE is active again at the next update event. 1010:Reserved. 1011: Reserved. 1100: Combined PWM mode 0 – O0CPRE has the same behavior as in PWM mode 0. O0CPREC is the logical OR between O0CPRE and O1CPRE 1101: Combined PWM mode 1– O0CPRE has the same behavior as in PWM mode 1. O0CPREC is the logical AND between O0CPRE and O1CPRE. 1110: Asymmetric PWM mode 0 –O0CPRE has the same behavior as in PWM mode 0. O0CPREC outputs O0CPRE when the counter is counting up, O1CPRE when it is counting down. 1111: Asymmetric PWM mode 1 –O0CPRE has the same behavior as in PWM mode 1. O0CPREC outputs O0CPRE when the counter is counting up, O1CPRE when it is counting down. This bit cannot be modified when PROT [1:0] bit-filed in TIMERx_CCHP register is 11 and CH0MS bit-filed is 00(COMPARE MODE).
3	CH0COMSEN	Channel 0 compare output shadow enable When this bit is set, the shadow register of TIMERx_CH0CV register, which updates at each update event, will be enabled. 0: Channel 0 output compare shadow disable 1: Channel 0 output compare shadow enable The PWM mode can be used without verifying the shadow register only in single pulse mode (when SPM=1)
2	CH0COMFEN	Channel 0 output compare fast enable When this bit is set, the effect of an event on the trigger in input on the capture/compare output will be accelerated if the channel is configured in PWM0 or PWM1 mode. The output channel will treat an active edge on the trigger input as a compare match, and CH0_O is set to the compare level independently from the result of the comparison. 0: Channel 0 output quickly compare disable. 1: Channel 0 output quickly compare enable.
1:0	CH0MS[1:0]	Channel 0 I/O mode selection

This bit-field specifies the work mode of the channel and the input signal selection. This bit-field is writable only when the channel is not active. (CH0EN bit in TIMERx_CHCTL2 register is reset).).

00: Channel 0 is programmed as output mode

01: Channel 0 is programmed as input mode, IS0 is connected to CI0FE0

10: Channel 0 is programmed as input mode, IS0 is connected to CI1FE0

11: Channel 0 is programmed as input mode, IS0 is connected to ITS**.** This mode is working only if an internal trigger input is selected, through TSCFGx[2:0](x = 3,4,5,6,7) bit-field in SYSCFG_TIMER2CFG (x=0) register.

Input capture mode:

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value
15:12	CH1CAPFLT[3:0]	Channel 1 input capture filter control Refer to CH0CAPFLT description
11:10	CH1CAPPSC[1:0]	Channel 1 input capture prescaler Refer to CH0CAPPSC description
9:8	CH1MS[1:0]	Channel 1 mode selection Same as Output compare mode
7:4	CH0CAPFLT[3:0]	Channel 0 input capture filter control The CI0 input signal can be filtered by digital filter and this bit-field configure the filtering capability. Basic principle of digital filter: continuously sample the CI0 input signal according to f_{SAMP} and record the number of times of the same level of the signal. After reaching the filtering capacity configured by this bit, it is considered to be an effective level. The filtering capability configuration is as follows:

CH0CAPFLT [3:0]	Times	f_{SAMP}
4’b0000	Filter disabled.	
4’b0001	2	f_{CK_TIMER}
4’b0010	4	
4’b0011	8	
4’b0100	6	$f_{DTS}/2$
4’b0101	8	
4’b0110	6	$f_{DTS}/4$
4’b0111	8	
4’b1000	6	$f_{DTS}/8$
4’b1001	8	
4’b1010	5	$f_{DTS}/16$
4’b1011	6	
4’b1100	8	
4’b1101	5	$f_{DTS}/32$

4’b1110	6	
4’b1111	8	

3:2 CH0CAPPSC[1:0] Channel 0 input capture prescaler

This bit-field specifies the factor of the prescaler on channel 0 input. The prescaler is reset when CH0EN bit in TIMERx_CHCTL2 register is clear.

00: Prescaler disable, input capture occurs on every channel input edge

01: The input capture occurs on every 2 channel input edges

10: The input capture occurs on every 4 channel input edges

11: The input capture occurs on every 8 channel input edges

1:0 CH0MS[1:0] Channel 0 mode selection

Same as Output compare mode

Channel control register 1 (TIMERx_CHCTL1)

Address offset: 0x1C

Reset value: 0x0000 0000

This register has to be accessed by word(32-bit)

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="7">Reserved</td><td>CH1COM
CTL[3]</td><td colspan="7">Reserved</td><td>CH0COM
CTL[3]</td></tr>
</table>

<table>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>CH3COM
CEN</td><td colspan="3">CH3COMCTL[2:0]</td><td>CH3COM
SEN</td><td>CH3COM
FEN</td><td colspan="2" rowspan="2">CH3MS[1:0]</td><td>CH2COM
CEN</td><td colspan="3">CH2COMCTL[2:0]</td><td>CH2COM
SEN</td><td>CH2COM
FEN</td><td colspan="2" rowspan="2">CH2MS[1:0]</td></tr>
<tr><td colspan="4">CH3CAPFLT[3:0]</td><td colspan="2">CH3CAPPSC[1:0]</td><td colspan="4">CH2CAPFLT[3:0]</td><td colspan="2">CH2CAPPSC[1:0]</td></tr>
<tr><td colspan="4">Rw</td><td colspan="2">rw</td><td colspan="2">rw</td><td colspan="4">rw</td><td colspan="2">rw</td><td colspan="2">rw</td></tr>
</table>

Output compare mode:

Bits	Fields	Descriptions
31:25	Reserved	Must be kept at reset value
24	CH1COMCTL[3]	Refer to CH1COMCTL[2:0]
23:17	Reserved	Must be kept at reset value
16	CH0COMCTL[3]	Refer to CH0COMCTL[2:0]
15	CH3COMCEN	Channel 3 output compare clear enable Refer to CH0COMCEN description
14:12	CH3COMCTL[2:0]	Channel 3 compare output control Refer to CH0COMCTL description
11	CH3COMSEN	Channel 3 output compare shadow enable Refer to CH0COMSEN description

10	CH3COMFEN	Channel 3 output compare fast enable Refer to CH0COMFEN description
9:8	CH3MS[1:0]	Channel 3 mode selection This bit-field specifies the direction of the channel and the input signal selection. This bit-field is writable only when the channel is not active. (CH3EN bit in TIMERx_CHCTL2 register is reset). 00: Channel 3 is programmed as output mode 01: Channel 3 is programmed as input mode, IS3 is connected to CI3FE3 10: Channel 3 is programmed as input mode, IS3 is connected to CI2FE3 11: Channel 3 is programmed as input mode, IS3 is connected to ITS. This mode is working only if an internal trigger input is selected, through TSCFGx[2:0](x = 3,4,5,6,7) bit-field in SYSCFG_TIMER2CFG (x=0) register.
7	CH2COMCEN	Channel 2 output compare clear enable. When this bit is set, if the ETIFP signal is detected as high level, the O2CPRE signal will be cleared. 0: Channel 2 output compare clear disable 1: Channel 2 output compare clear enable
6:4	CH2COMCTL[2:0]	Channel 2 compare output control This bit-field controls the behavior of the output reference signal O2CPRE which drives CH2_O and CH2_ON. O2CPRE is active high, while CH2_O and CH2_ON active level depends on CH2P and CH2NP bits. 0000: Frozen. The O2CPRE signal keeps stable, independent of the comparison between the output compare register TIMERx_CH2CV and the counter TIMERx_CNT. 0001: Set high on match. O2CPRE signal is forced high when the counter matches the output compare register TIMERx_CH2CV. 0010: Set low on match. O2CPRE signal is forced low when the counter matches the output compare register TIMERx_CH2CV. 0011: Toggle on match. O2CPRE toggles when the counter matches the output compare register TIMERx_CH2CV. 0100: Force low. O2CPRE is forced low level. 0101: Force high. O2CPRE is forced high level. 0110: PWM mode0. When counting up, O2CPRE is active as long as the counter is smaller than TIMERx_CH2CV else inactive. When counting down, O2CPRE is inactive as long as the counter is larger than TIMERx_CH2CV else active. 0111: PWM mode1. When counting up, O2CPRE is inactive as long as the counter is smaller than TIMERx_CH2CV else active. When counting down, O2CPRE is active as long as the counter is larger than TIMERx_CH2CV else inactive. When configured in PWM mode, the O2CPRE level changes only when the output compare mode switches from “frozen” mode to “PWM” mode or when the result of the comparison changes. 1000: Retriggerable single pulse mode 0. The behavior of O2CPRE is performed as in PWM mode 0. When counting up, the O2CPRE is active. When an trigger

		event occurs, the O2CPRE is inactive. The O2CPRE is active again at the next update event; When counting down, the O0CPRE is inactive, When an trigger event occurs, the O0CPRE is active. The O0CPRE is inactive again at the next update event. 1001: Retriggerable single pulse mode 1. The behavior of O0CPRE is performed as in PWM mode 1. When counting up, the O0CPRE is inactive, When an trigger event occurs,the O0CPRE is active. The O0CPRE is inactive again at the next update event; When counting down, the O0CPRE is active. When an trigger event occurs, the O0CPRE is inactive.The O0CPRE is active again at the next update event. 1010:Reserved. 1011: Reserved. 1100: Combined PWM mode 0 – O2CPRE has the same behavior as in PWM mode 0. O2CPREC is the logical OR between O2CPRE and O3CPRE. 1101: Combined PWM mode 1– O0CPRE has the same behavior as in PWM mode 1. O2CPREC is the logical AND between O2CPRE and O3CPRE. 1110: Asymmetric PWM mode 0 –O2CPRE has the same behavior as in PWM mode 0. O2CPREC outputs O2CPRE when the counter is counting up, O3CPRE when it is counting down. 1111: Asymmetric PWM mode 1 –O0CPRE has the same behavior as in PWM mode 1. O2CPREC outputs O2CPRE when the counter is counting up, O3CPRE when it is counting down. This bit cannot be modified when PROT [1:0] bit-filed in TIMERx_CCHP register is 11 and CH2MS bit-filed is 00(COMPARE MODE).
3	CH2COMSEN	Channel 2 compare output shadow enable When this bit is set, the shadow register of TIMERx_CH2CV register, which updates at each update event will be enabled. 0: Channel 2 output compare shadow disable 1: Channel 2 output compare shadow enable The PWM mode can be used without verifying the shadow register only in single pulse mode (when SPM=1)
2	CH2COMFEN	Channel 2 output compare fast enable When this bit is set, the effect of an event on the trigger in input on the capture/compare output will be accelerated if the channel is configured in PWM1 or PWM2 mode. The output channel will treat an active edge on the trigger input as a compare match, and CH2_O is set to the compare level independently from the result of the comparison. 0: Channel 2 output quickly compare disable. 1: Channel 2 output quickly compare enable.
1:0	CH2MS[1:0]	Channel 2 I/O mode selection This bit-field specifies the work mode of the channel and the input signal selection. This bit-field is writable only when the channel is not active. (CH2EN bit in TIMERx_CHCTL2 register is reset).).

00: Channel 2 is programmed as output mode

01: Channel 2 is programmed as input mode, IS2 is connected to CI2FE2

10: Channel 2 is programmed as input mode, IS2 is connected to CI3FE2

11: Channel 2 is programmed as input mode, IS2 is connected to ITS. This mode is working only if an internal trigger input is selected, through TSCFGx[2:0](x = 3,4,5,6,7) bit-field in SYSCFG_TIMER2CFG (x=0) register.

Input capture mode:

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value
15:12	CH3CAPFLT[3:0]	Channel 3 input capture filter control Refer to CH0CAPFLT description
11:10	CH3CAPPSC[1:0]	Channel 3 input capture prescaler Refer to CH0CAPPSC description
9:8	CH3MS[1:0]	Channel 3 mode selection Same as Output compare mode
7:4	CH2CAPFLT[3:0]	Channel 2 input capture filter control The CI2 input signal can be filtered by digital filter and this bit-field configure the filtering capability. Basic principle of digital filter: continuously sample the CI2 input signal according to f_{SAMP} and record the number of times of the same level of the signal. After reaching the filtering capacity configured by this bit, it is considered to be an effective level. The filtering capability configuration is as follows:

CH2CAPFLT [3:0]	Times	f_{SAMP}
4’b0000	Filter disabled.	
4’b0001	2	f_{CK_TIMER}
4’b0010	4	
4’b0011	8	
4’b0100	6	$f_{DTS}/2$
4’b0101	8	
4’b0110	6	$f_{DTS}/4$
4’b0111	8	
4’b1000	6	$f_{DTS}/8$
4’b1001	8	
4’b1010	5	$f_{DTS}/16$
4’b1011	6	
4’b1100	8	
4’b1101	5	$f_{DTS}/32$
4’b1110	6	
4’b1111	8	

3:2	CH2CAPPSC[1:0]	Channel 2 input capture prescaler This bit-field specifies the factor of the prescaler on channel 2 input. The prescaler is reset when CH2EN bit in TIMERx_CHCTL2 register is clear. 00: Prescaler disable, input capture occurs on every channel input edge 01: The input capture occurs on every 2 channel input edges 10: The input capture occurs on every 4 channel input edges 11: The input capture occurs on every 8 channel input edges
1:0	CH2MS[1:0]	Channel 2 mode selection Same as output compare mode

Channel control register 2 (TIMERx_CHCTL2)

Address offset: 0x20

Reset value: 0x0000 0000

This register has to be accessed by word(32-bit)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH3NP	Reserved	CH3P	CH3EN	CH2NP	Reserved	CH2P	CH2EN	CH1NP	Reserved	CH1P	CH1EN	CH0NP	Reserved	CH0P	CH0EN
rw		rw	rw	rw		rw	rw	rw		rw	rw	rw		rw	rw

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value
15	CH3NP	Channel 3 complementary output polarity Refer to CH0NP description
14	Reserved	Must be kept at reset value
13	CH3P	Channel 3 capture/compare function polarity Refer to CH0P description
12	CH3EN	Channel 3 capture/compare function enable Refer to CH0EN description
11	CH2NP	Channel 2 complementary output polarity Refer to CH0NP description
10	Reserved	Must be kept at reset value
9	CH2P	Channel 2 capture/compare function polarity Refer to CH0P description
8	CH2EN	Channel 2 capture/compare function enable Refer to CH0EN description

7	CH1NP	Channel 1 complementary output polarity Refer to CH0NP description
6	Reserved	Must be kept at reset value
5	CH1P	Channel 1 capture/compare function polarity Refer to CH0P description
4	CH1EN	Channel 1 capture/compare function enable Refer to CH0EN description
3	CH0NP	Channel 0 complementary output polarity When channel 0 is configured in output mode, this bit should be keep reset value. When channel 0 is configured in input mode, together with CH0P, this bit is used to define the polarity of CI0. This bit cannot be modified when PROT [1:0] bit-filed in TIMERx_CCHP register is 11 or 10.
2	Reserved	Must be kept at reset value
1	CH0P	Channel 0 capture/compare function polarity When channel 0 is configured in output mode, this bit specifies the output signal polarity. 0: Channel 0 high level is active level 1: Channel 0 low level is active level When channel 0 is configured in input mode, this bit specifies the CI0 signal polarity. [CH0NP, CH0P] will select the active trigger or capture polarity for CI0FE0 or CI1FE0. [CH0NP==0, CH0P==0]: CIxFE0’s rising edge is the active signal for capture or trigger operation in slave mode. And CIxFE0 will not be inverted. [CH0NP==0, CH0P==1]: CIxFE0’s falling edge is the active signal for capture or trigger operation in slave mode. And CIxFE0 will be inverted. [CH0NP==1, CH0P==0]: Reserved. [CH0NP==1, CH0P==1]: CIxFE0’s falling and rising edge are both the active signal for capture or trigger operation in slave mode. And CIxFE0 will be not inverted.
0	CH0EN	Channel 0 capture/compare function enable When channel 0 is configured in output mode, setting this bit enables CH0_O signal in active state. When channel 0 is configured in input mode, setting this bit enables the capture event in channel0. 0: Channel 0 disabled 1: Channel 0 enabled

Counter register (TIMERx_CNT)

Address offset: 0x24
Reset value: 0x0000 0000

This register has to be accessed by word(32-bit)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNT[15:0]															
rw															

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value
15:0	CNT[15:0]	This bit-filed indicates the current counter value. Writing to this bit-filed can change the value of the counter.

Prescaler register (TIMERx_PSC)

Address offset: 0x28

Reset value: 0x0000 0000

This register has to be accessed by word(32-bit)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PSC[15:0]															
rw															

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value
15:0	PSC[15:0]	Prescaler value of the counter clock The TIMER_CK clock is divided by (PSC+1) to generate the counter clock. The value of this bit-filed will be loaded to the corresponding shadow register at every update event.

Counter auto reload register (TIMERx_CAR)

Address offset: 0x2C

Reset value: 0x0000 0000

This register has to be accessed by word(32-bit)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CARL[15:0]															
rw															

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value
15:0	CARL[15:0]	Counter auto reload value This bit-filed specifies the auto reload value of the counter.

Channel 0 capture/compare value register (TIMERx_CH0CV)

Address offset: 0x34

Reset value: 0x0000 0000

This register has to be accessed by word(32-bit)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH0VAL[15:0]															
rw															

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value
15:0	CH0VAL[15:0]	Capture or compare value of channel0 When channel 0 is configured in input mode, this bit-filed indicates the counter value corresponding to the last capture event. And this bit-filed is read-only. When channel 0 is configured in output mode, this bit-filed contains value to be compared to the counter. When the corresponding shadow register is enabled, the shadow register updates every update event.

Channel 1 capture/compare value register (TIMERx_CH1CV)

Address offset: 0x38

Reset value: 0x0000 0000

This register has to be accessed by word(32-bit)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH1VAL[15:0]															
rw															

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value

15:0	CH1VAL[15:0]	Capture or compare value of channel1 When channel 1 is configured in input mode, this bit-filed indicates the counter value corresponding to the last capture event. And this bit-filed is read-only. When channel 1 is configured in output mode, this bit-filed contains value to be compared to the counter. When the corresponding shadow register is enabled, the shadow register updates every update event.

Channel 2 capture/compare value register (TIMERx_CH2CV)

Address offset: 0x3C

Reset value: 0x0000 0000

This register has to be accessed by word(32-bit)

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="16">Reserved</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="16">CH2VAL[15:0]</td></tr>
<tr><td colspan="16">rw</td></tr>
</table>

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value
15:0	CH2VAL[15:0]	Capture or compare value of channel 2 When channel 2 is configured in input mode, this bit-filed indicates the counter value corresponding to the last capture event. And this bit-filed is read-only. When channel 2 is configured in output mode, this bit-filed contains value to be compared to the counter. When the corresponding shadow register is enabled, the shadow register updates every update event.

Channel 3 capture/compare value register (TIMERx_CH3CV)

Address offset: 0x40

Reset value: 0x0000 0000

This register has to be accessed by word(32-bit)

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="16">Reserved</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="16">CH3VAL[15:0]</td></tr>
<tr><td colspan="16">rw</td></tr>
</table>

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value

15:0	CH3VAL[15:0]	Capture or compare value of channel 3 When channel3 is configured in input mode, this bit-filed indicates the counter value corresponding to the last capture event. And this bit-filed is read-only. When channel 3 is configured in output mode, this bit-filed contains value to be compared to the counter. When the corresponding shadow register is enabled, the shadow register updates every update event.

DMA configuration register (TIMERx_DMACFG)

Address offset: 0x48
Reset value: 0x0000 0000

This register has to be accessed by word(32-bit)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved			DMATC[4:0]					Reserved			DMATA [4:0]				
			rw								rw				

Bits	Fields	Descriptions
31:13	Reserved	Must be kept at reset value.
12:8	DMATC [4:0]	DMA transfer count This filed defines the number(n) of the register that DMA will access(R/W), n = (DMATC [4:0] +1). DMATC [4:0] is from 5’b0_0000 to 5’b1_0001.
7:5	Reserved	Must be kept at reset value.
4:0	DMATA [4:0]	DMA transfer access start address This filed define the first address for the DMA access the TIMERx_DMATB. When access is done through the TIMERx_DMA address first time, this bit-field specifies the address you just access. And then the second access to the TIMERx_DMATB, you will access the address of start address + 0x4.

DMA transfer buffer register (TIMERx_DMATB)

Address offset: 0x4C
Reset value: 0x0000 0000

This register has to be accessed by word(32-bit)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DMATB[15:0]															

rw

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value
15:0	DMATB[15:0]	DMA transfer buffer When a read or write operation is assigned to this register, the register located at the address range (Start Addr + Transfer Timer* 4) will be accessed. The transfer Timer is calculated by hardware, and ranges from 0 to DMATC.

Configuration register (TIMERx_CFG)

Address offset: 0xFC

Reset value: 0x0000 0000

This register has to be accessed by word(32-bit)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved														CHVSEL	Reserved
														rw	

Bits	Fields	Descriptions
31:2	Reserved	Must be kept at reset value
1	CHVSEL	Write CHxVAL register selection This bit-field set and reset by software. 1: If write the CHxVAL register, the write value is same as the CHxVAL value, the write access ignored 0: No effect
0	Reserved	Must be kept at reset value

14.3. General level2 timer (TIMERx, x=13)

14.3.1. Overview

The general level2 timer module (TIMER 13) is a one-channel timer that supports input capture, output compare. They can generate PWM signals to control motor or be used for power management applications. The general level2 time reference is a 16-bit counter that can be used as an unsigned counter.

In addition, the general level2 timers can be programmed and be used to count or time external events that drive other timers.

14.3.2. Characteristics

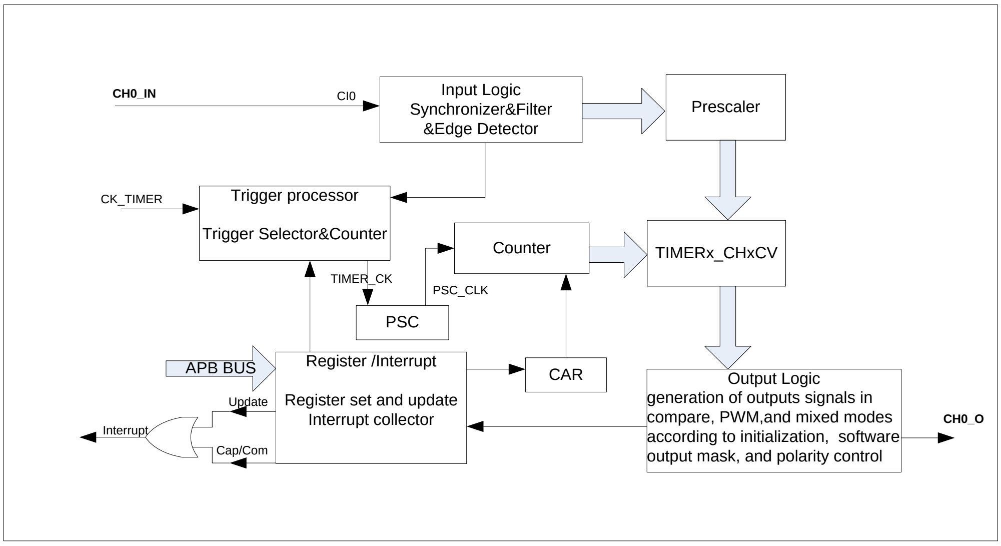
- Total channel num: 1.
- Counter width: 16-bit.
- Source of count clock: internal clock.
- Counter mode: count up only.
- Programmable prescaler: 16-bit. Factor can be changed on the go.
- Each channel is user-configurable:
  Input capture mode, output compare mode, programmable and PWM mode.
- Auto-reload function.
- Interrupt output: update, compare/capture event.

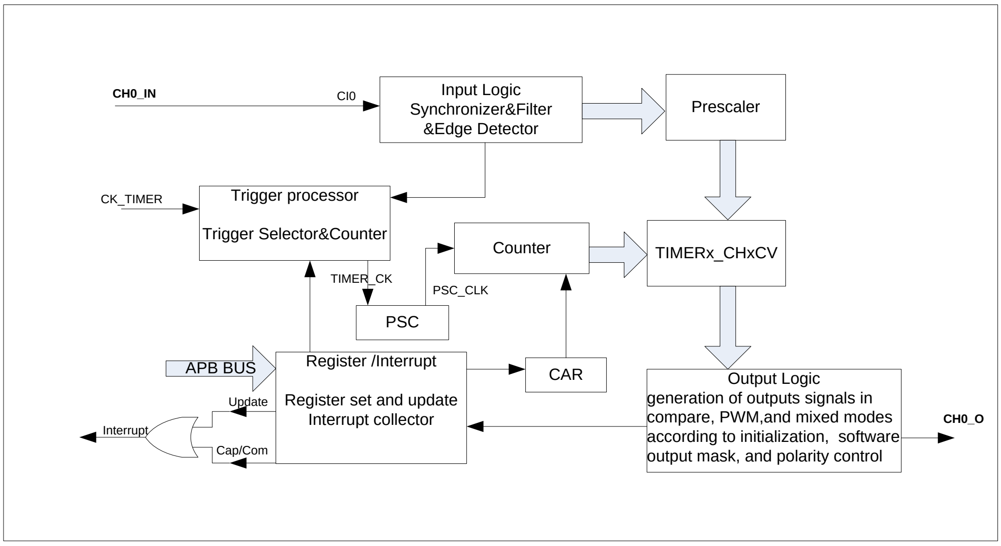
14.3.3. Block diagram

Figure 14-64. General level2 timer block diagram

provides details on the internal configuration of the general level2 timer.

Figure 14-64. General level2 timer block diagram





14.3.4. Function overview

Clock source configuration

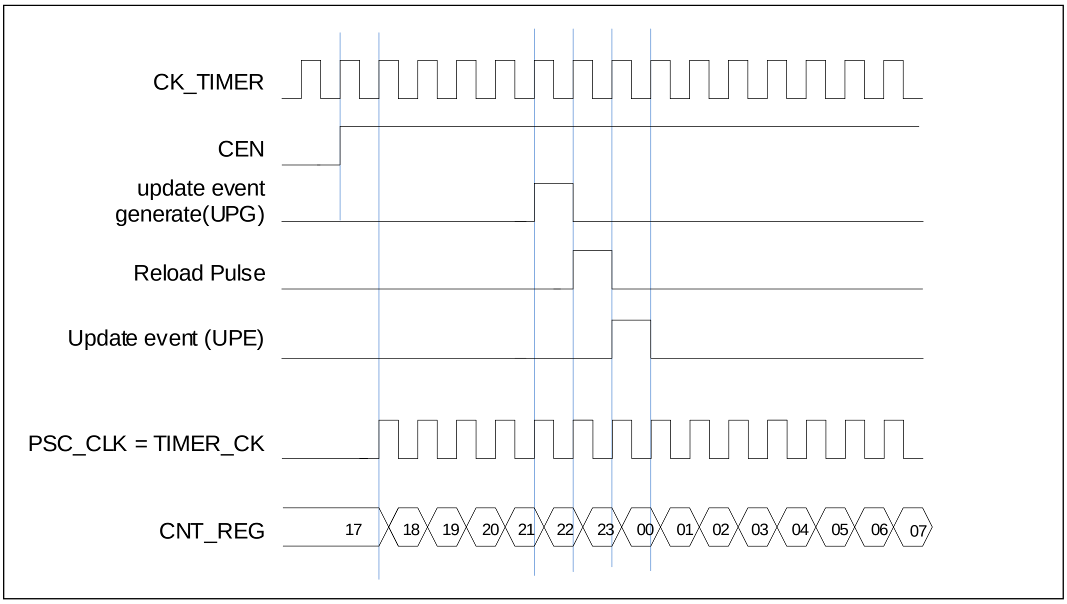
The general level2 TIMER can only being clocked by the CK_TIMER.

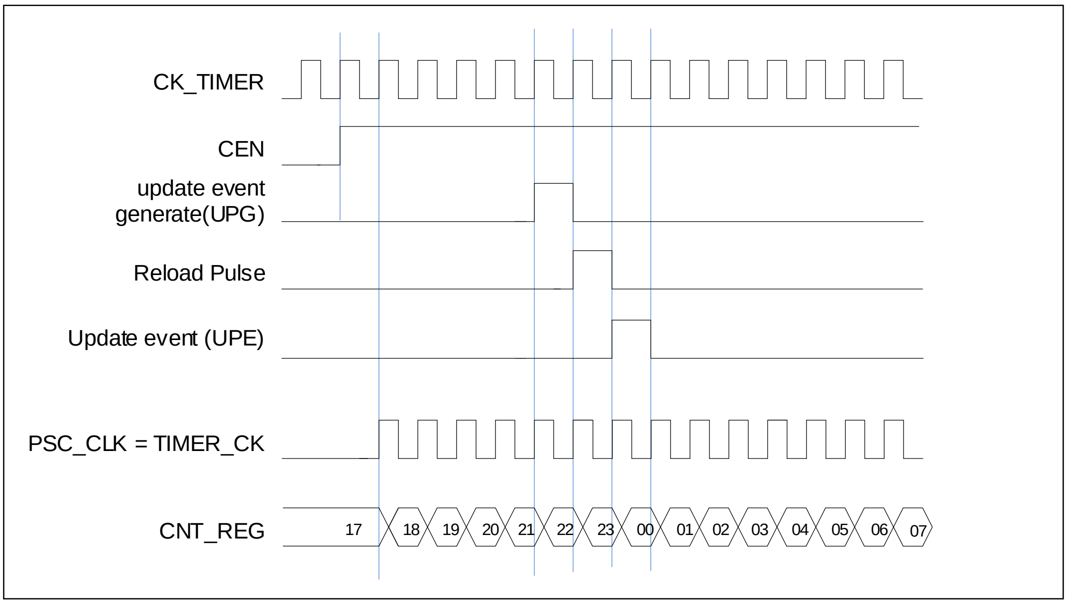
- Internal timer clock CK_TIMER which is from module RCU.

The general level2 TIMER has only one clock source which is the internal CK_TIMER, used to drive the counter prescaler. When the CEN is set, the CK_TIMER will be divided by PSC value to generate PSC_CLK.

The TIMER_CK, driven counter’s prescaler to count, is equal to CK_TIMER which is from RCU.

Figure 14-65. Timing chart of internal clock divided by 1

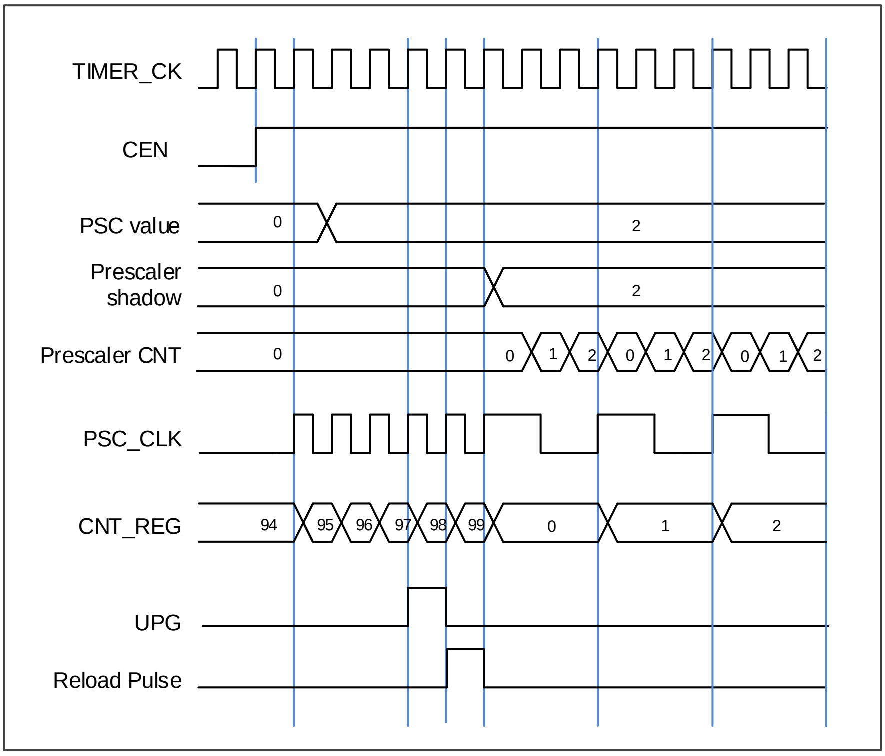


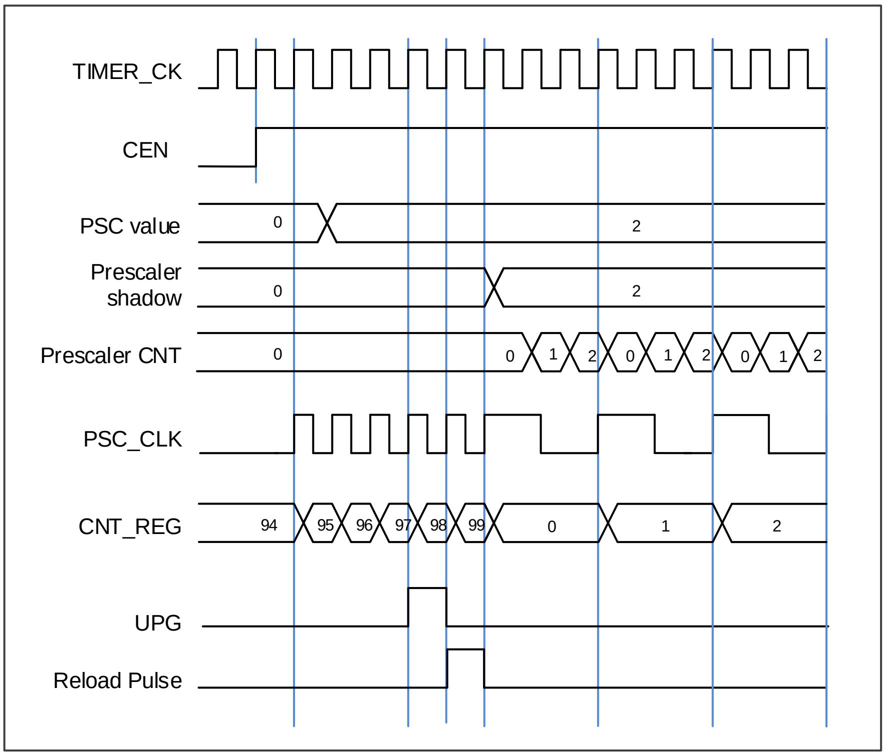


Clock prescaler

The counter clock (PSC_CK) is obtained by the TIMER_CK through the prescaler, and the prescale factor can be configured from 1 to 65536 through the prescaler register (TIMERx_PSC). The new written prescaler value will not take effect until the next update event.

Figure 14-66. Timing chart of PSC value change from 0 to 2





Counter up counting

In this mode, the counter counts up continuously from 0 to the counter-reload value, which is defined in the TIMERx_CAR register, in a count-up direction. Once the counter reaches the counter reload value, the counter will start counting up from 0 again. The update event is generated at each counter overflow. The counting direction bit DIR in the TIMERx_CTL1 register should be set to 0 for the up counting mode.

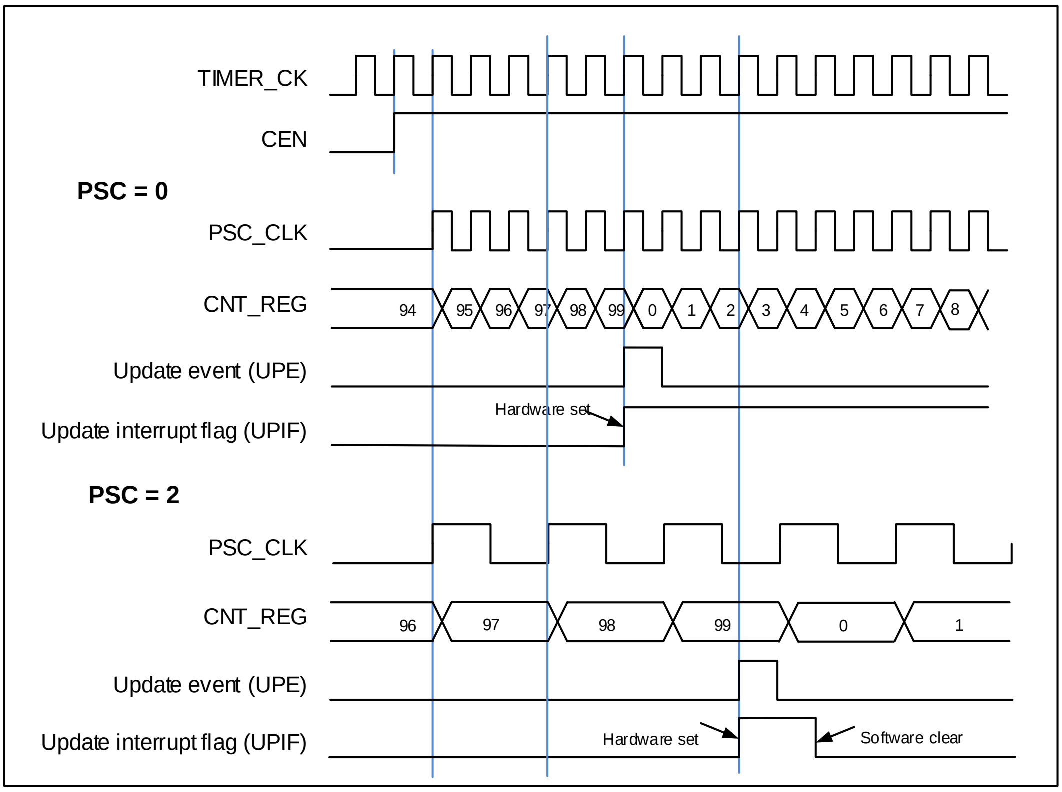
When the update event is set by the UPG bit in the TIMERx_SWEVG register, the counter value will be initialized to 0 and generates an update event.

If the UPDIS bit in TIMERx_CTL0 register is set, the update event is disabled.

When an update event occurs, all the shadow registers (counter auto reload register, prescaler register) are updated.

The following figures show some examples of the counter behavior for different clock prescaler factor when TIMERx_CAR=0x99.

Figure 14-67. Timing chart of up counting mode, PSC=0/2



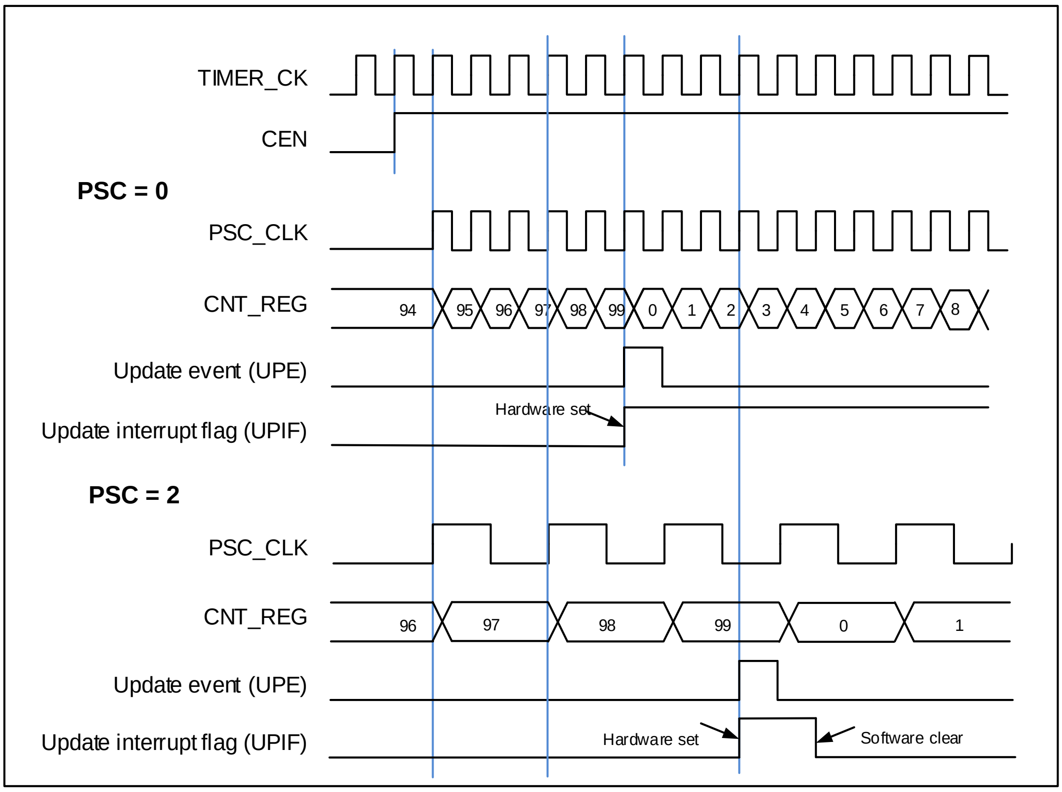
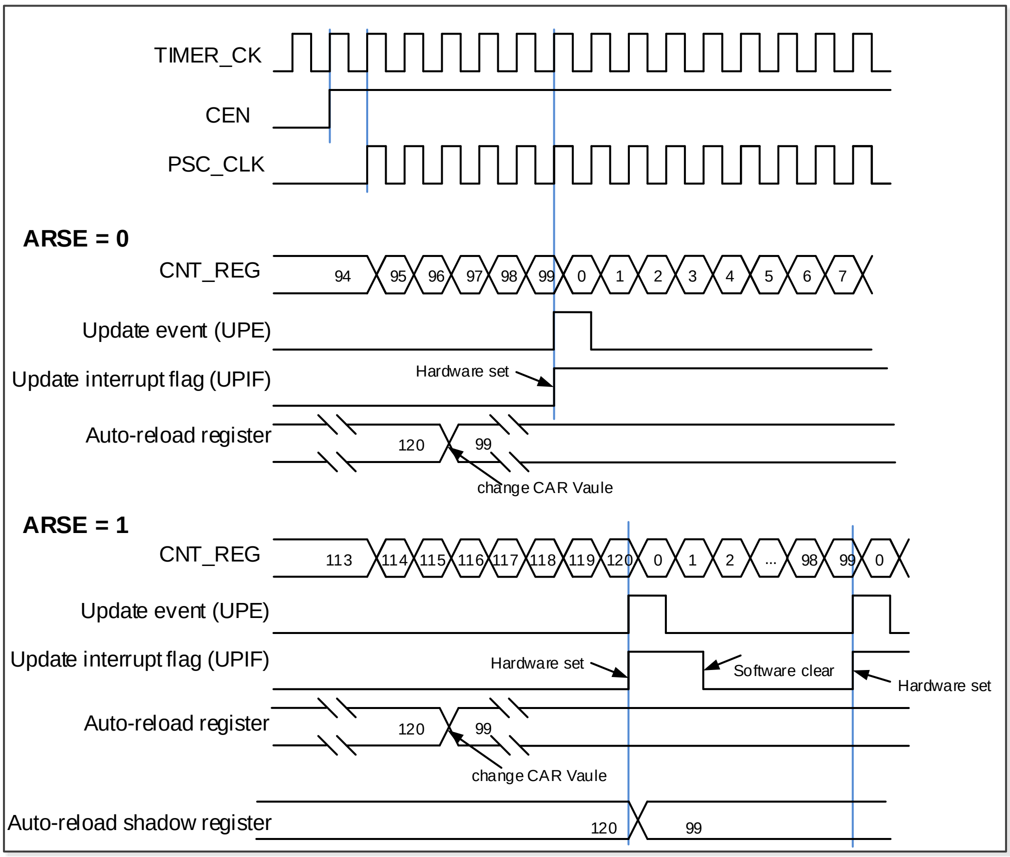
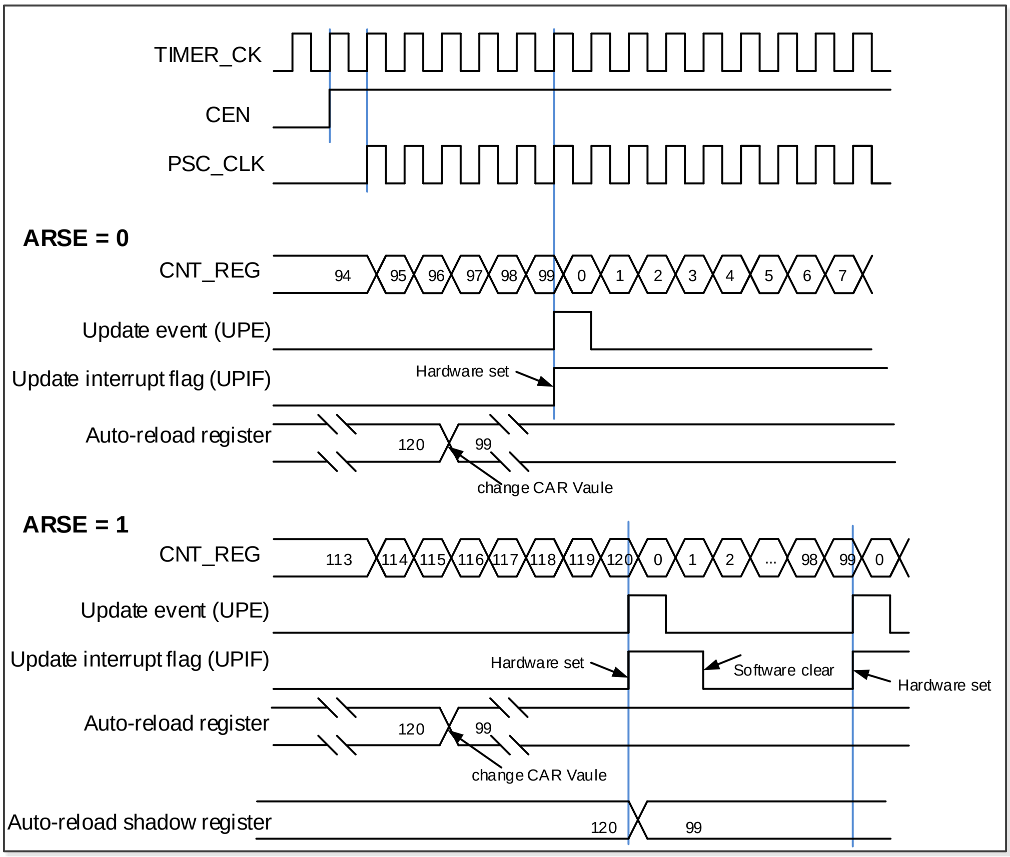


Figure 14-68. Timing chart of up counting mode, change TIMERx_CAR on the go





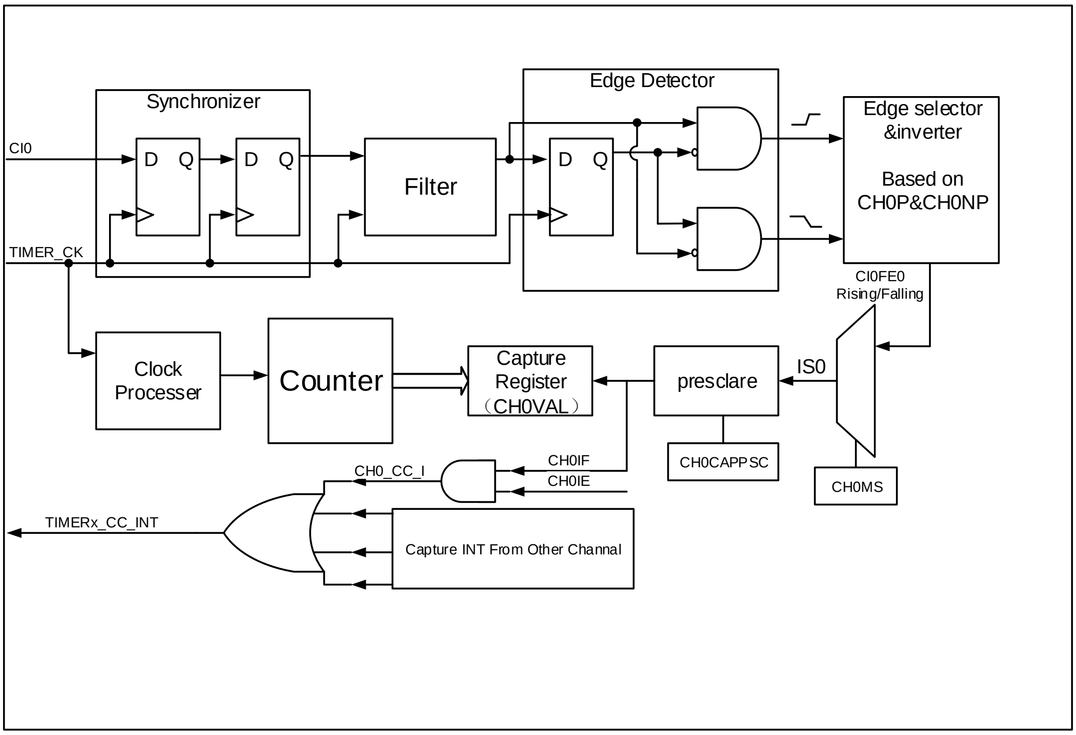
Input capture and output compare channels

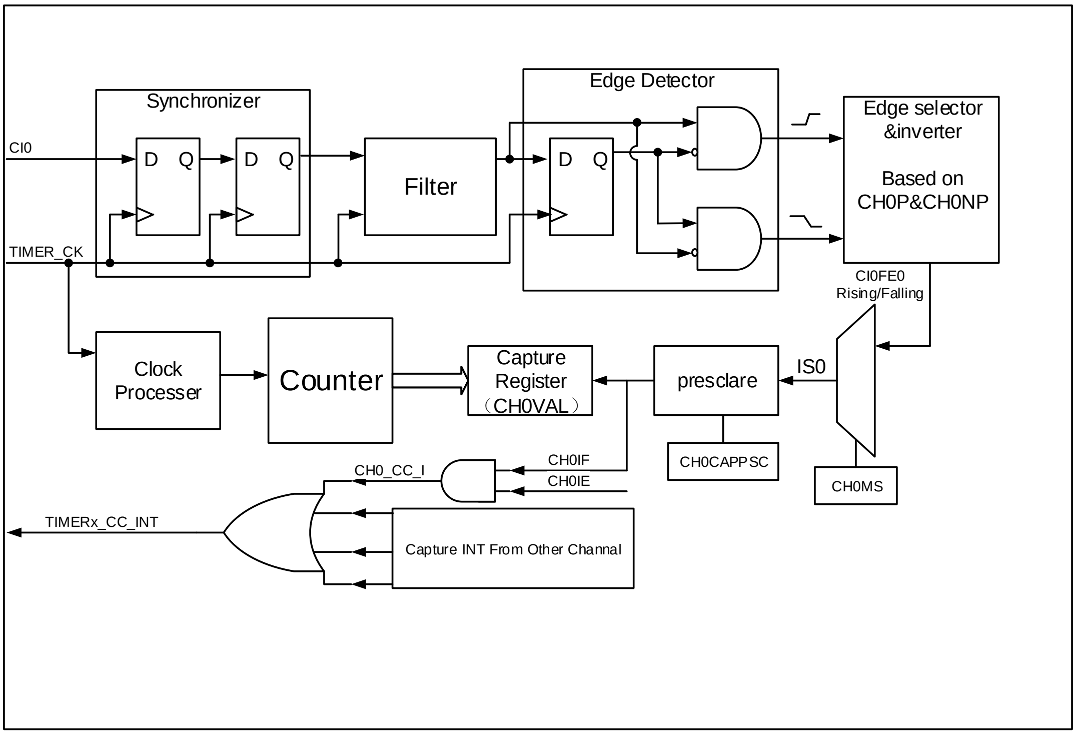
The general level2 timer has one independent channel which can be used as capture inputs or compare match outputs. Each channel is built around a channel capture compare register including an input stage, channel controller and an output stage.

- **Channel input capture function**

Channel input capture function allows the channel to perform measurements such as pulse timing, frequency, period, duty cycle and so on. The input stage consists of a digital filter, a channel polarity selection, edge detection and a channel prescaler. When a selected edge occurs on the channel input, the current value of the counter is captured into the TIMERx_CHxCV register, at the same time the CHxIF bit is set and the channel interrupt is generated if enabled by CHxIE = 1.

Figure 14-69. Channel input capture principle





First, the channel input signal (CIx) is synchronized to TIMER_CK domain, and then sampled by a digital filter to generate a filtered input signal. Then through the edge detector, the rising and fall edge are detected. You can select one of them by CHxP. One more selector is for the other channel and trig, controlled by CHxMS. The IC_prescaler make several the input event generate one effective capture event. On the capture event, CHxVAL will restore the value of Counter.

So, the process can be divided to several steps as below:

Step1: Filter configuration. (CHxCAPFLT in TIMERx_CHCTL0)

Based on the input signal and requested signal quality, configure compatible CHxCAPFLT.

Step2: Edge selection. (CHxP/CHxNP in TIMERx_CHCTL2)

Rising or falling edge, choose one by CHxP/CHxNP.

Step3: Capture source selection. (CHxMS in TIMERx_CHCTL0)

As soon as you select one input capture source by CHxMS, you have set the channel to input mode (CHxMS! =0x0) and TIMERx_CHxCV cannot be written any more.

Step4: Interrupt enable. (CHxIE in TIMERx_DMAINTEN)

Enable the related interrupt; you can get the interrupt.

Step5: Capture enables. (CHxEN in TIMERx_CHCTL2).

Result: When you wanted input signal is got, TIMERx_CHxCV will be set by Counter’s value. And CHxIF is asserted. If the CHxIF is high, the CHxOF will be asserted also. The interrupt will be asserted based on the configuration of CHxIE in TIMERx_DMAINTEN.

Direct generation: If you want to generate a DMA request or Interrupt, you can set CHxG by software directly.

- **Channel output compare function**

In Output Compare mode, the TIMERx can generate timed pulses with programmable position, polarity, duration, and frequency. When the counter matches the value in the CHxVAL register of an output compare channel, the channel (n) output can be set, cleared, or toggled based on CHxCOMCTL. when the counter reaches the value in the CHxVAL register, the CHxIF bit is set and the channel (n) interrupt is generated if CHxIE = 1.d

So, the process can be divided to several steps as below:
Step1: Clock configuration. Such as clock source, clock prescaler and so on.

Step2: Compare mode configuration.
* Set the shadow enable mode by CHxCOMSEN
* Set the output mode (Set/Clear/Toggle) by CHxCOMCTL.
* Select the active high polarity by CHxP/CHxNP
* Enable the output by CHxEN

Step3: Interrupt/DMA-request enables configuration by CHxIE

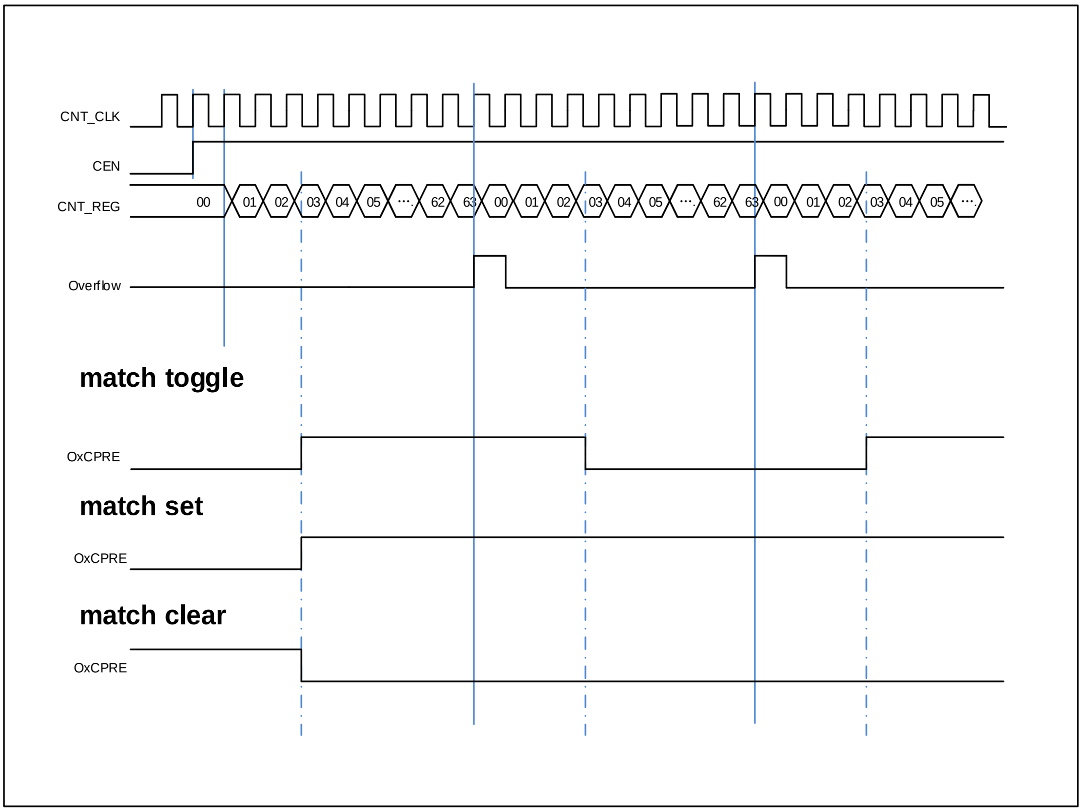
Step4: Compare output timing configuration by TIMERx_CAR and TIMERx_CHxCV.
About the CHxVAL, you can change it on the go to meet the waveform you expected.

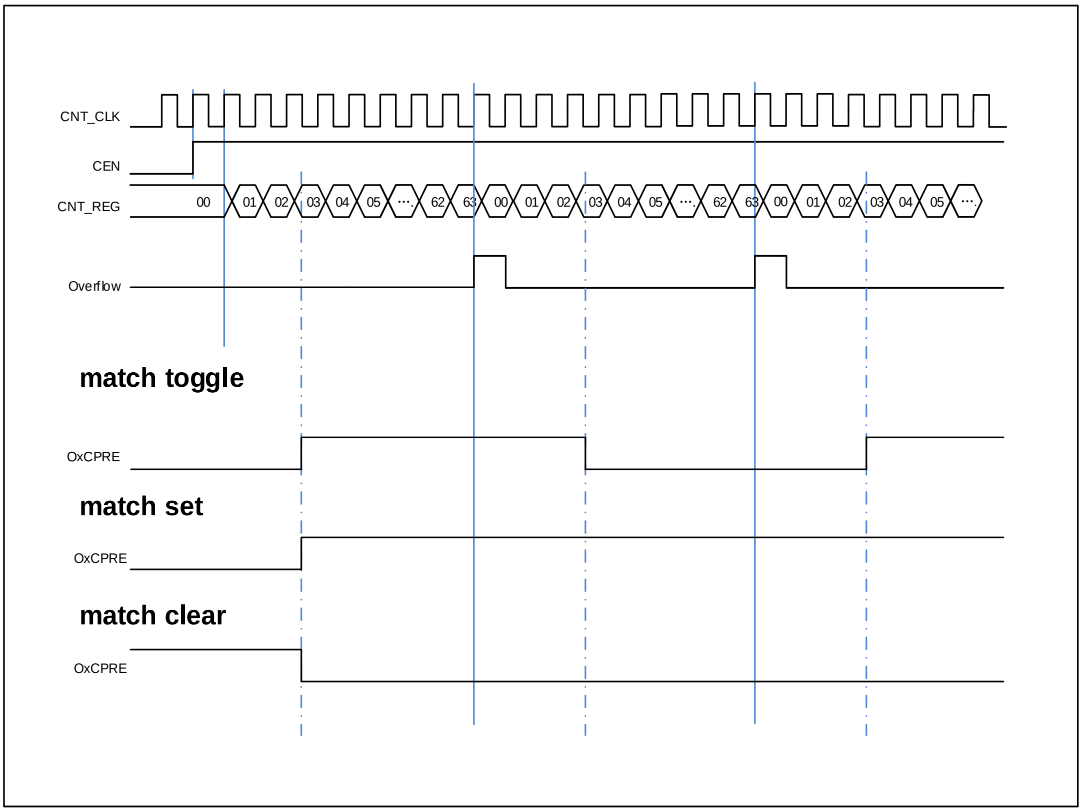
Step5: Start the counter by CEN.

The timechart ***Figure 14-70. Output-compare under three modes***

_show the three compare modes toggle/set/clear. CAR=0x63, CHxVAL=0x3

Figure 14-70. Output-compare under three modes





Output PWM function

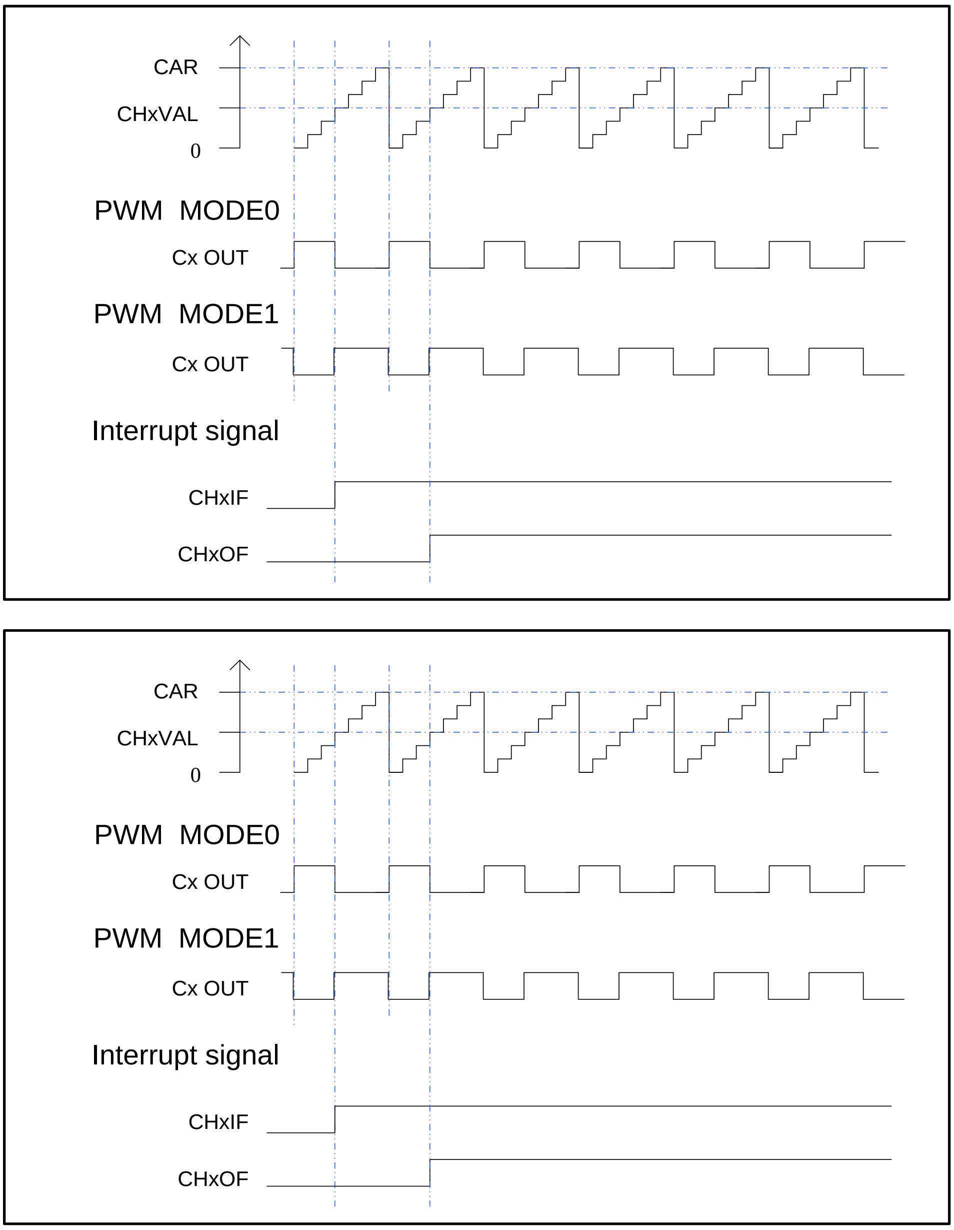
In the output PWM function (by setting the CHxCOMCTL bits to 3’b110 (PWM mode0) or to

3’b 111(PWM mode1), the channel can generate PWM waveform according to the TIMERx_CAR registers and TIMERx_CHxCV registers.

The period is determined by TIMERx_CAR and duty cycle is determined by TIMERx_CHxCV. *Figure 14-71. PWM mode timechart* shows the PWM output mode and interrupts waveform.

If TIMERx_CHxCV is greater than TIMERx_CAR, the output will be always active under PWM mode0 (CHxCOMCTL==3’b110).
And if TIMERx_CHxCV is equal to zero, the output will be always inactive under PWM mode0 (CHxCOMCTL==3’b110).

Figure 14-71. PWM mode timechart



Channel output prepare signal

When the TIMERx is used in the compare match output mode, the OxCPRE signal (Channel x Output prepare signal) is defined by setting the CHxCOMCTL filed. The OxCPRE signal has several types of output function. These include, keeping the original level by setting the CHxCOMCTL field to 0x00, set to 1 by setting the CHxCOMCTL field to 0x01, set to 0 by setting the CHxCOMCTL field to 0x02 or signal toggle by setting the CHxCOMCTL field to 0x03 when the counter value matches the content of the TIMERx_CHxCV register.

The PWM mode 0 and PWM mode 1 outputs are also another kind of OxCPRE output which is setup by setting the CHxCOMCTL field to 0x06/0x07. In these modes, the OxCPRE signal level is changed according to the counting direction and the relationship between the counter value and the TIMERx_CHxCV content. With regard to a more detail description refer to the relative bit definition.

Another special function of the OxCPRE signal is a forced output which can be achieved by setting the CHxCOMCTL field to 0x04/0x05. Here the output can be forced to an inactive/active level irrespective of the comparison condition between the counter and the TIMERx_CHxCV values.

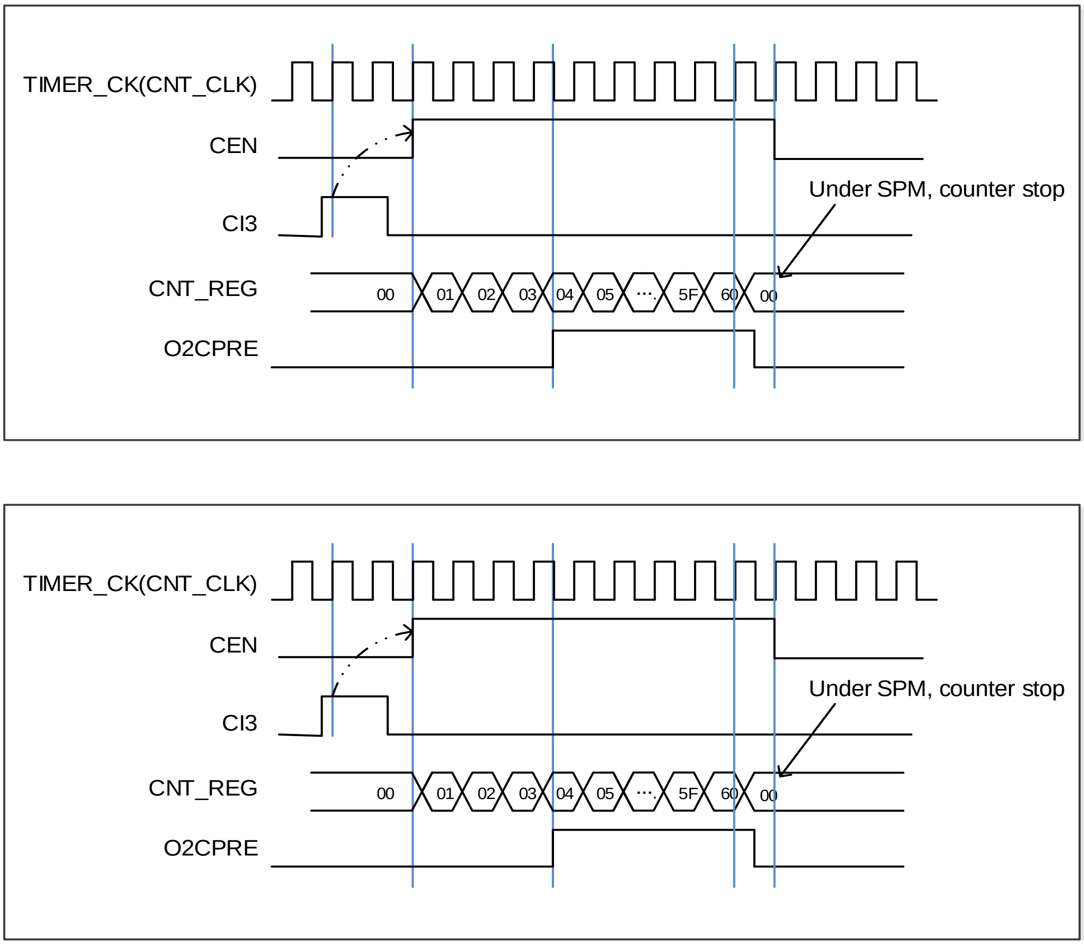
Single pulse mode

Single pulse mode is enabled by setting SPM in TIMERx_CTL0. If SPM is set, the counter will be cleared and stopped when the next update event occurs. In order to get a pulse waveform, the TIMERx is configured to PWM mode or compare mode by CHxCOMCTL.

Once the timer is set to the single pulse mode, it is not necessary to configure the timer enable bit CEN in the TIMERx_CTL0 register to 1 to enable the counter. Setting the CEN bit to 1 or a trigger signal edge can generate a pulse and then keep the CEN bit at a high state until the update event occurs or the CEN bit is written to 0 by software. If the CEN bit is cleared to 0 by software, the counter will be stopped and its value will be held.

In the single pulse mode, the active edge of trigger which sets the CEN bit to 1 will enable the counter. However, there exists several clock delays to perform the comparison result between the counter value and the TIMERx_CHxCV value. In order to reduce the delay to a minimum value, the user can set the CHxCOMFEN bit in TIMERx_CHCTL0/1 register. After a trigger rising occurs in the single pulse mode, the OxCPRE signal will immediately be forced to the state which the OxCPRE signal will change to, as the compare match event occurs without taking the comparison result into account. The CHxCOMFEN bit is available only when the output channel is configured to the PWM mode 0 or PWM mode 1 and the trigger source is derived from the trigger signal.

Figure 14-72. Single pulse mode TIMERx_CHxCV=0x04, TIMERx_CAR=0x60



Timer debug mode

When the Cortex®-M23 halted, and the TIMERx_HOLD configuration bit in DBG_CTL0 register set to 1, the TIMERx counter stops.

14.3.5. Register definition

TIMER13 base address: 0x4000 2000

Control register 0 (TIMERx_CTL0)

Address offset: 0x00

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved						CKDIV[1:0]		ARSE	Reserved				UPS	UPDIS	CEN
						rw		rw					rw	rw	rw

Bits	Fields	Descriptions
31:10	Reserved	Must be kept at reset value.
9:8	CKDIV[1:0]	Clock division The CKDIV bits can be configured by software to specify division factor between the CK_TIMER and the dead-time and digital filter sample clock (DTS). 00: f_{DTS}=f_{CK_TIMER} 01: f_{DTS}= f_{CK_TIMER} /2 10: f_{DTS}= f_{CK_TIMER} /4 11: Reserved
7	ARSE	Auto-reload shadow enable 0: The shadow register for TIMERx_CAR register is disabled 1: The shadow register for TIMERx_CAR register is enabled
6:4	Reserved	Must be kept at reset value.
3	SPM	Single pulse mode. 0: Single pulse mode disable. The counter continues after update event. 1: Single pulse mode enable. The counter counts until the next update event occurs.
2	UPS	Update source This bit is used to select the update event sources by software. 0: These events generate update interrupts or DMA requests: The UPG bit is set The counter generates an overflow or underflow event The restart mode generates an update event. 1: This event generates update interrupts or DMA requests: The counter generates an overflow or underflow event
1	UPDIS	Update disable.

		This bit is used to enable or disable the update event generation. 0: Update event enable. When an update event occurs, the corresponding shadow registers are loaded with their preloaded values. These events generate update event: The UPG bit is set The counter generates an overflow or underflow event The restart mode generates an update event. 1: Update event disable. **Note:** When this bit is set to 1, setting UPG bit or restart mode does not generate an update event, but the counter and prescaler are initialized.
0	CEN	Counter enable 0: Counter disable 1: Counter enable The CEN bit must be set by software when timer works in external clock, pause mode and quadrature decoder mode.

Interrupt enable register (TIMERx_DMAINTEN)

Address offset: 0x0C

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="16">Reserved</td></tr>
</table>

<table>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="14">Reserved</td><td>CH0IE</td><td>UPIE</td></tr>
<tr><td colspan="14"></td><td>rw</td><td>rw</td></tr>
</table>

Bits	Fields	Descriptions
31:2	Reserved	Must be kept at reset value.
1	CH0IE	Channel 0 capture/compare interrupt enable 0: disabled 1: enabled
0	UPIE	Update interrupt enable 0: disabled 1: enabled

Interrupt flag register (TIMERx_INTF)

Address offset: 0x10

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved						CH0OF	Reserved.							CH0IF	UPIF
						rc_w0								rc_w0	rc_w0

Bits	Fields	Descriptions
31:10	Reserved	Must be kept at reset value.
9	CH0OF	Channel 0 over capture flag When channel 0 is configured in input mode, this flag is set by hardware when a capture event occurs while CH0IF flag has already been set. This flag is cleared by software. 0: No over capture interrupt occurred 1: Over capture interrupt occurred
8:2	Reserved	Must be kept at reset value.
1	CH0IF	Channel 0 ‘s capture/compare interrupt flag This flag is set by hardware and cleared by software. When channel 0 is in input mode, this flag is set when a capture event occurs. When channel 0 is in output mode, this flag is set when a compare event occurs. 0: No Channel 1 interrupt occurred 1: Channel 1 interrupt occurred
0	UPIF	Update interrupt flag This bit is set by hardware on an update event and cleared by software. 0: No update interrupt occurred 1: Update interrupt occurred

Software event generation register (TIMERx_SWEVG)

Address offset: 0x14
Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved														CH0G	UPG
														w	w

Bits	Fields	Descriptions
31:2	Reserved	Must be kept at reset value.

1	CH0G	Channel 0's capture or compare event generation This bit is set by software in order to generate a capture or compare event in channel 0, it is automatically cleared by hardware. When this bit is set, the CH1IF flag is set, the corresponding interrupt or DMA request is sent if enabled. In addition, if channel 1 is configured in input mode, the current value of the counter is captured in TIMERx_CH0CV register, and the CH0OF flag is set if the CH0IF flag was already high. 0: No generate a channel 1 capture or compare event 1: Generate a channel 1 capture or compare event
0	UPG	This bit can be set by software, and cleared by hardware automatically. When this bit is set, the counter is cleared. The prescaler counter is cleared at the same time. 0: No generate an update event 1: Generate an update event

Channel control register 0 (TIMERx_CHCTL0)

Address offset: 0x18

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved.								Reserved	CH0COMCTL[2:0]			CH0COMSEN	CH0COMFEN	CH0MS[1:0]	
								CH0CAPFLT[3:0]				CH0CAPPSC[1:0]			
								rw				rw		rw	

Output compare mode:

Bits	Fields	Descriptions
31:7	Reserved	Must be kept at reset value.
6:4	CH0COMCTL[2:0]	Channel 0 compare output control This bit-field specifies the compare output mode of the the output prepare signal O0CPRE. In addition, the high level of O0CPRE is the active level, and CH0_O and CH0_ON channels polarity depends on CH0P and CH0NP bits. 000: Timing mode. The O0CPRE signal keeps stable, independent of the comparison between the register TIMERx_CH0CV and the counter TIMERx_CNT. 001: Set the channel output. O0CPRE signal is forced high when the counter is equals to the output compare register TIMERx_CH0CV. 010: Clear the channel output. O0CPRE signal is forced low when the counter is equals to the output compare register TIMERx_CH0CV. 011: Toggle on match. O0CPRE toggles when the counter is equals to the output compare register TIMERx_CH0CV. 100: Force low. O0CPRE is forced to low level.

		101: Force high. O0CPRE is forced to high level. 110: PWM mode0. When counting up, O0CPRE is high when the counter is smaller than TIMERx_CH0CV, and low otherwise. When counting down, O0CPRE is low when the counter is larger than TIMERx_CH0CV, and high otherwise. 111: PWM mode1. When counting up, O0CPRE is low when the counter is smaller than TIMERx_CH0CV, and high otherwise. When counting down, O0CPRE is high when the counter is larger than TIMERx_CH0CV, and low otherwise. If configured in PWM mode, the O0CPRE level changes only when the output compare mode is adjusted from “Timing” mode to “PWM” mode or the comparison result changes.
3	CH0COMSEN	Channel 0 compare output shadow enable When this bit is set, the shadow register of TIMERx_CH0CV register, which updates at each update event, will be enabled. 0: Channel 0 output compare shadow disable 1: Channel 0 output compare shadow enable The PWM mode can be used without verifying the shadow register only in single pulse mode (when SPM=1)
2	CH0COMFEN	Channel 0 output compare fast enable When this bit is set, the effect of an event on the trigger in input on the capture/compare output will be accelerated if the channel is configured in PWM0 or PWM1 mode. The output channel will treat an active edge on the trigger input as a compare match, and CH0_O is set to the compare level independently from the result of the comparison. 0: Channel 0 output quickly compare disable. 1: Channel 0 output quickly compare enable.
1:0	CH0MS[1:0]	Channel 0 I/O mode selection This bit-field specifies the work mode of the channel and the input signal selection. This bit-field is writable only when the channel is not active. (CH0EN bit in TIMERx_CHCTL2 register is reset).). 00: Channel 0 is programmed as output mode 01: Channel 0 is programmed as input mode, IS0 is connected to CI0FE0 10: Reserved 11: Reserved

Input capture mode:

Bits	Fields	Descriptions
31:8	Reserved	Must be kept at reset value.
7:4	CH0CAPFLT[3:0]	Channel 0 input capture filter control The CI0 input signal can be filtered by digital filter and this bit-field configure the filtering capability. Basic principle of digital filter: continuously sample the CI0 input signal according to f_{SAMP} and record the number of times of the same level of the signal. After reaching

the filtering capacity configured by this bit, it is considered to be an effective level.

The filtering capability configuration is as follows:

CH0CAPFLT [3:0]	Times	f_{SAMP}
4’b0000	Filter disabled.	
4’b0001	2	f_{CK_TIMER}
4’b0010	4	
4’b0011	8	
4’b0100	6	$f_{DTS}/2$
4’b0101	8	
4’b0110	6	$f_{DTS}/4$
4’b0111	8	
4’b1000	6	$f_{DTS}/8$
4’b1001	8	
4’b1010	5	$f_{DTS}/16$
4’b1011	6	
4’b1100	8	
4’b1101	5	$f_{DTS}/32$
4’b1110	6	
4’b1111	8	

3:2 CH0CAPPSC[1:0] Channel 0 input capture prescaler

This bit-field specifies the factor of the prescaler on channel 0 input. The prescaler is reset when CH0EN bit in TIMERx_CHCTL2 register is clear.

00: Prescaler disable, input capture occurs on every channel input edge

01: The input capture occurs on every 2 channel input edges

10: The input capture occurs on every 4 channel input edges

11: The input capture occurs on every 8 channel input edges

1:0 CH0MS[1:0] Channel 0 mode selection

Same as output compare mode

Channel control register 2 (TIMERx_CHCTL2)

Address offset: 0x20

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved												CH0NP	Reserved	CH0P	CH0EN
												rw		rw	rw

Bits	Fields	Descriptions

31:4	Reserved	Must be kept at reset value.
3	CH0NP	Channel 0 complementary output polarity When channel 0 is configured in output mode, this bit specifies the complementary output signal polarity. 0: Channel 0 active high 1: Channel 0 active low When channel 0 is configured in input mode, together with CH0P, this bit is used to define the polarity of CI0. This bit cannot be modified when PROT [1:0] bit-filed in TIMERx_CCHP register is 11 or 10.
2	Reserved	Must be kept at reset value.
1	CH0P	Channel 0 capture/compare polarity When channel 0 is configured in output mode, this bit specifies the output signal polarity. 0: Channel 0 high level is active level 1: Channel 0 low level is active level When channel 0 is configured in input mode, this bit specifies the CI0 signal polarity. [CH0NP, CH0P] will select the active trigger or capture polarity for CI0FE0 or CI1FE0. [CH0NP==0, CH0P==0]: CIxFE0's rising edge is the active signal for capture or trigger operation in slave mode. And CIxFE0 will not be inverted. [CH0NP==0, CH0P==1]: CIxFE0's falling edge is the active signal for capture or trigger operation in slave mode. And CIxFE0 will be inverted. [CH0NP==1, CH0P==0]: Reserved. [CH0NP==1, CH0P==1]: CIxFE0's falling and rising edge are both the active signal for capture or trigger operation in slave mode. And CIxFE0 will be not inverted. This bit cannot be modified when PROT [1:0] bit-filed in TIMERx_CCHP register is 11 or 10.
0	CH0EN	Channel 0 capture/compare function enable When channel 0 is configured in input mode, setting this bit enables CH0_O signal in active state. When channel 0 is configured in output mode, setting this bit enables the capture event in channel0. 0: Channel 0 disabled 1: Channel 0 enabled

Counter register (TIMERx_CNT)

Address offset: 0x24

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNT[15:0]															
rw															

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	CNT[15:0]	This bit-filed indicates the current counter value. Writing to this bit-filed can change the value of the counter.

Prescaler register (TIMERx_PSC)

Address offset: 0x28

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PSC[15:0]															
rw															

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	PSC[15:0]	Prescaler value of the counter clock The TIMER_CK clock is divided by (PSC+1) to generate the counter clock. The value of this bit-filed will be loaded to the corresponding shadow register at every update event.

Counter auto reload register (TIMERx_CAR)

Address offset: 0x2C

Reset value: 0x0000 FFFF

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CARL[15:0]															
rw															

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.

15:0	CARL[15:0]	Counter auto reload value This bit-filed specifies the auto reload value of the counter. **Note:** When the timer is configured in input capture mode, this register must be configured a non-zero value (such as 0xFFFF) which is larger than user expected value.

Channel 0 capture/compare value register (TIMERx_CH0CV)

Address offset: 0x34

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH0VAL[15:0]															
rw															

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	CH0VAL[15:0]	Capture or compare value of channel0 When channel 0 is configured in input mode, this bit-filed indicates the counter value corresponding to the last capture event. And this bit-filed is read-only. When channel 0 is configured in output mode, this bit-filed contains value to be compared to the counter. When the corresponding shadow register is enabled, the shadow register updates every update event.

input selection register (TIMERx_INSEL)

Address offset: 0x68

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
												CI0_SEL[3:0]			
												rw			

Bits	Fields	Descriptions
31:4	Reserved	Must be kept at reset value.
3:0	CI0_SEL[3:0]	Channel 0 input selection 0000: Channel 0 input is connected to GPIO(TIMER13_CH0)

0001: Channel 0 input is connected to the RTCCLK

0010: Channel 0 input is connected to HXTAL/32 clock

0011: Channel 0 input is connected to CKOUTSEL0

0100: Channel 0 input is connected to CKOUTSEL1

Other:Reserved

Configuration register (TIMERx_CFG)

Address offset: 0xFC

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved														CHVSEL	Reserved
														rw	rw

Bits	Fields	Descriptions
31:2	Reserved	Must be kept at reset value.
1	CHVSEL	Write CHxVAL register selection This bit-field set and reset by software. 1: If write the CHxVAL register, the write value is same as the CHxVAL value, the write access ignored 0: No effect
0	Reserved	Must be kept at reset value.

14.4. General level4 timer (TIMERx, x=15,16)

14.4.1. Overview

The general level4 timer module (TIMER15, TIMER16) is a one-channel timer that supports both input capture and output compare. They can generate PWM signals to control motor or be used for power management applications. The general level4 timer has a 16-bit counter that can be used as an unsigned counter.

In addition, the general level4 timers can be programmed and be used for counting, their external events can be used to drive other timers.

Timer also includes a dead-time Insertion module which issuitable for motor control applications.

14.4.2. Characteristics

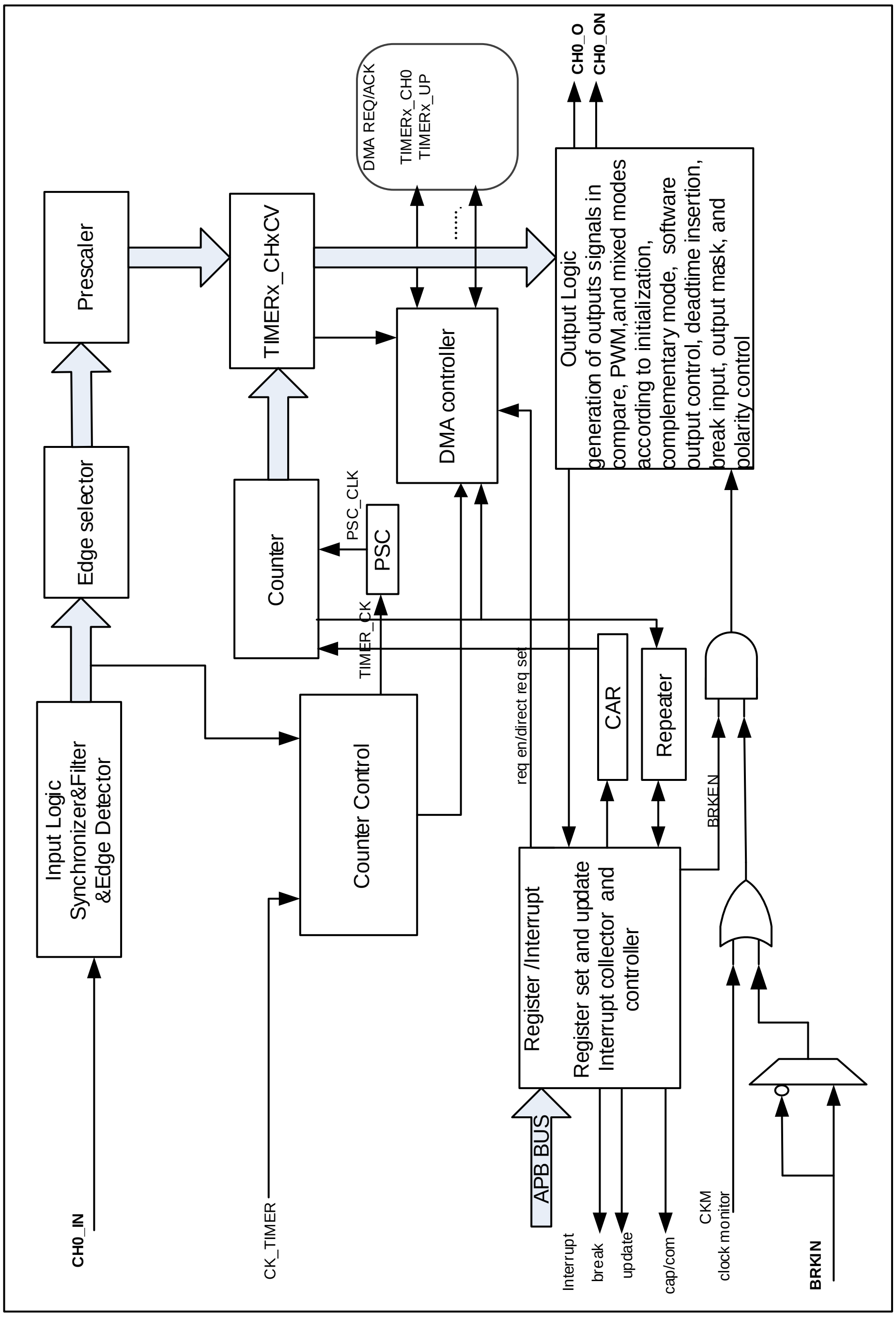
- Total channel num: 1.
- Counter width: 16-bit.
- Source of counter clock: internal clock.
- Counter modes: count up only.
- Programmable prescaler: 16-bit. The factor can be changed on the go.
- Each channel is user-configurable:
  input capture mode, output compare mode, programmable PWM mode, single pulse mode
- Programmable dead time insertion.
- Auto reload function.
- Programmable counter repetition function.
- Break input.
- Interrupt output or DMA request: update, compare/capture event, commutation event and break input.

14.4.3. Block diagram

Figure 14-73. General level4 timer block diagram

provides details of the internal configuration of the general level4 timer.

Figure 14-73. General level4 timer block diagram



14.4.4. Function overview

Clock source configuration

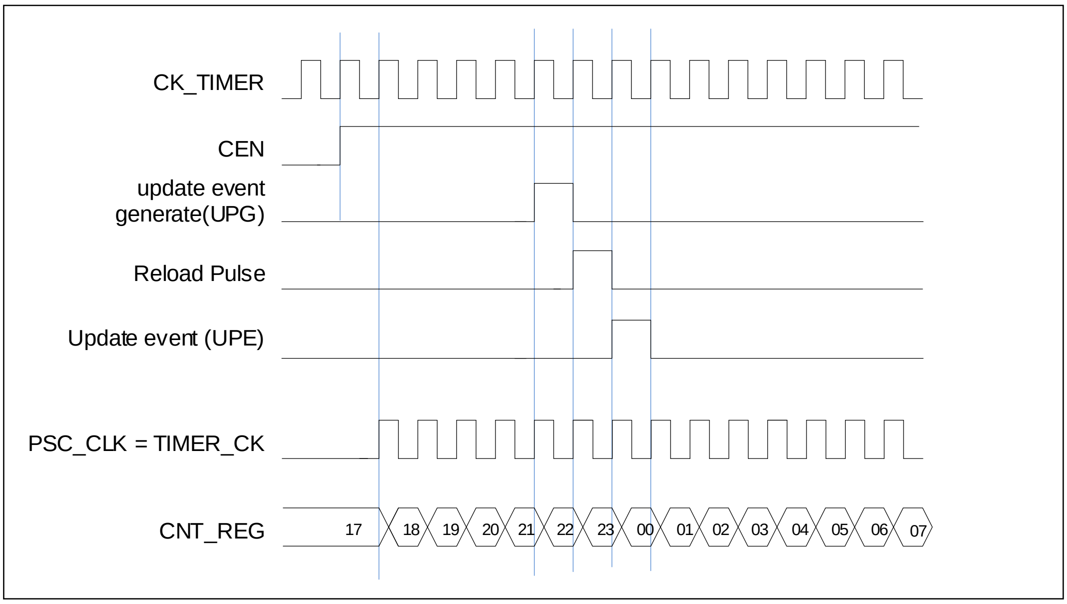
The general level4 TIMER can only being clocked by the CK_TIMER.

- Internal timer clock CK_TIMER which is from module RCU

The general level4 TIMER has only one clock source which is the internal CK_TIMER, used to drive the counter prescaler. When the CEN is set, the CK_TIMER will be divided by PSC value to generate PSC_CLK.

The TIMER_CK, driven counter's prescaler to count, is equal to CK_TIMER which is from RCU

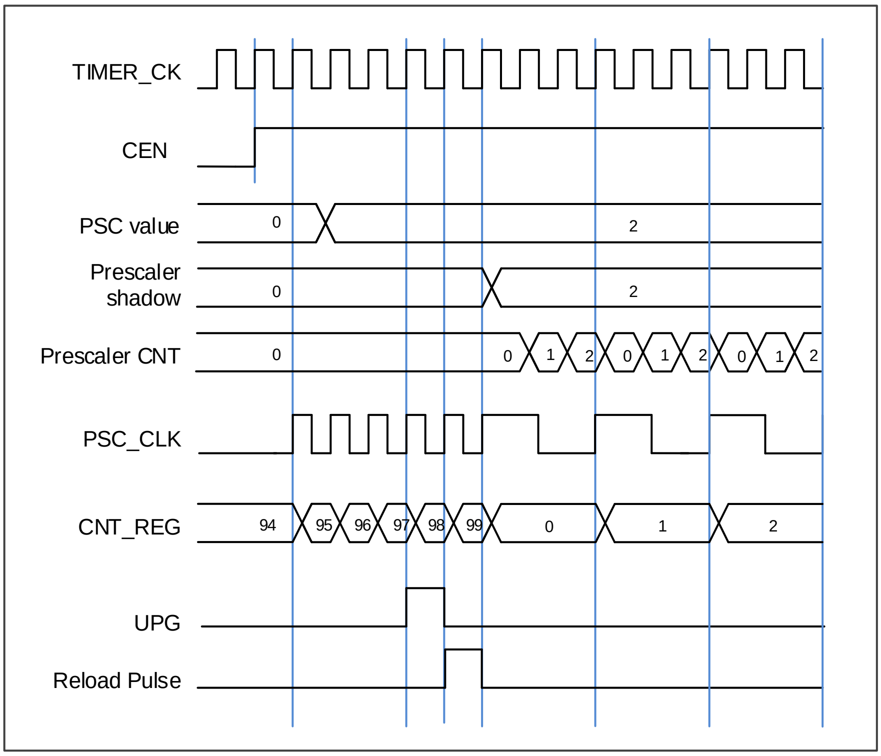
Figure 14-74. Timing chart of internal clock divided by 1



Clock prescaler

The counter clock (PSC_CK) is obtained by the TIMER_CK through the prescaler, and the prescale factor can be configured from 1 to 65536 through the prescaler register (TIMERx_PSC). The new written prescaler value will not take effect until the next update event.

Figure 14-75. Timing chart of PSC value change from 0 to 2



Counter up counting

In this mode, the counter counts up continuously from 0 to the counter-reload value, which is defined in the TIMERx_CAR register, in a count-up direction. Once the counter reaches the counter reload value, the counter will start counting up from 0 again and an overflow event will be generated. In addition, the update events will be generated after (TIMERx_CREP+1) times of overflow events. The counting direction bit DIR in the TIMERx_CTL0 register should be set to 0 for the up counting mode.

Whenever, if the update event software trigger is enabled by setting the UPG bit in the TIMERx_SWEVG register, the counter value will be initialized to 0 and generates an update event.

If set the UPDIS bit in TIMERx_CTL0 register, the update event is disabled.

When an update event occurs, all the shadow registers (repetition counter, counter auto reload register, prescaler register) are updated.

Figure 14-76. Timing chart of up counting mode, PSC=0/2

_show some examples of the counter behavior for different clock prescaler factor when TIMERx_CAR=0x99.

Figure 14-76. Timing chart of up counting mode, PSC=0/2

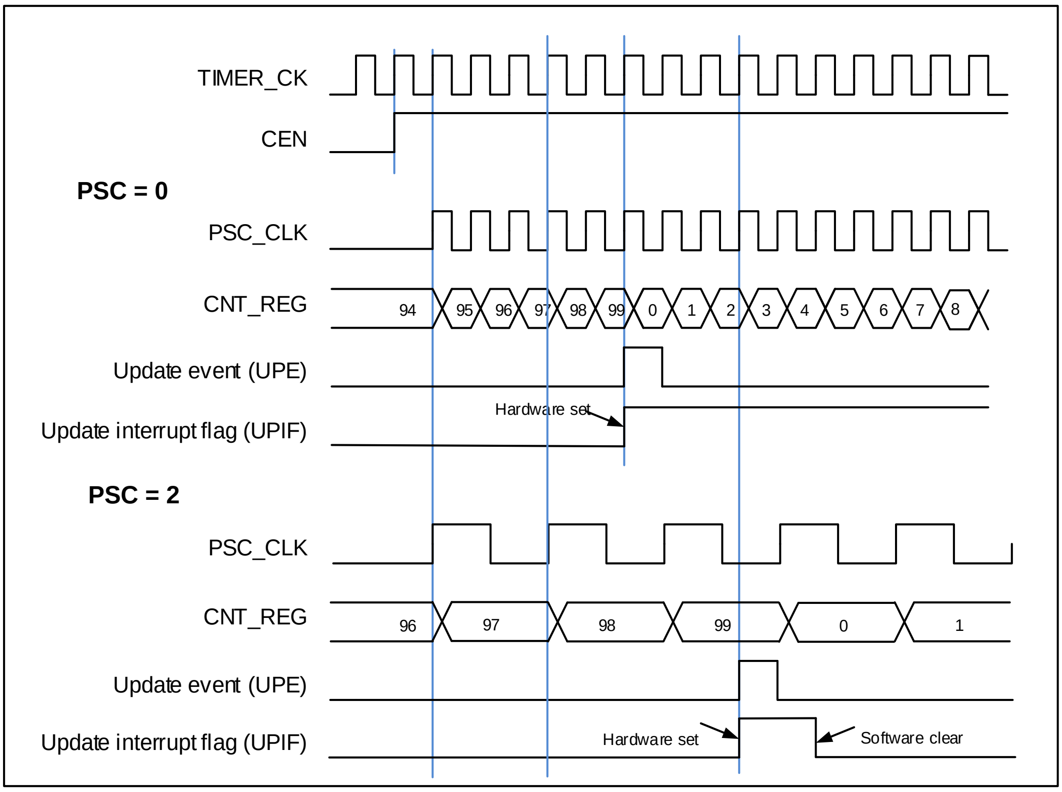
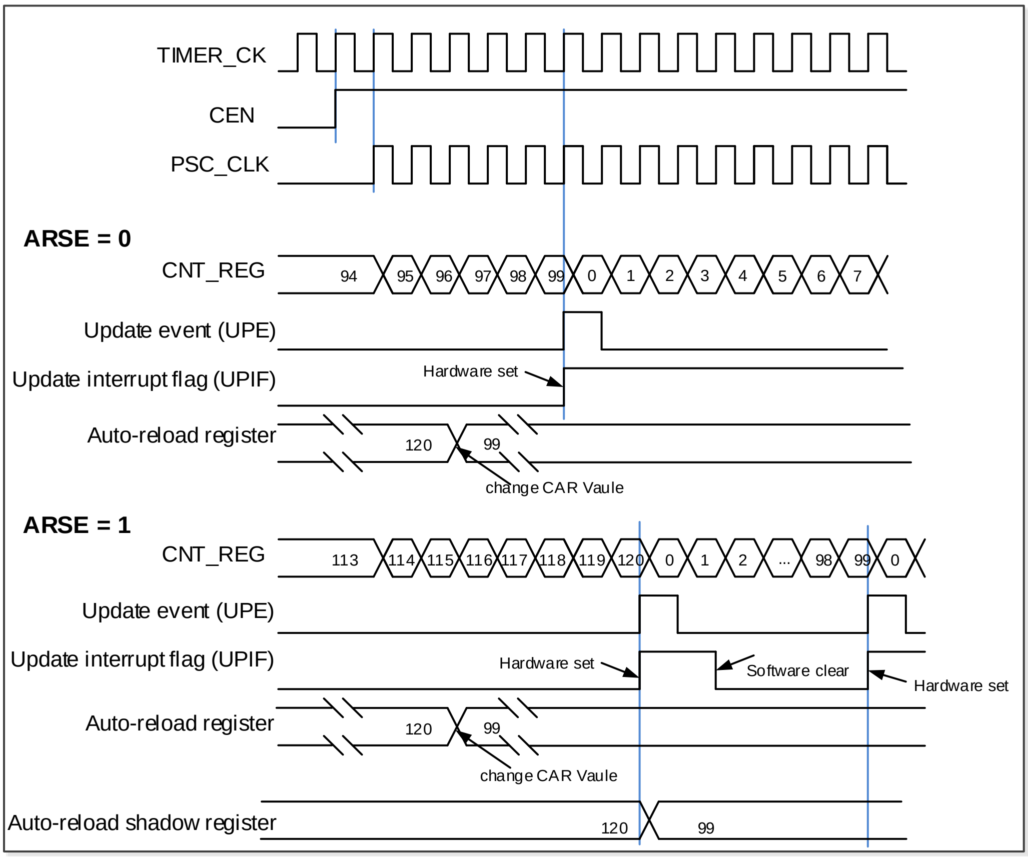


Figure 14-77. Timing chart of up counting mode, change TIMERx_CAR on the go

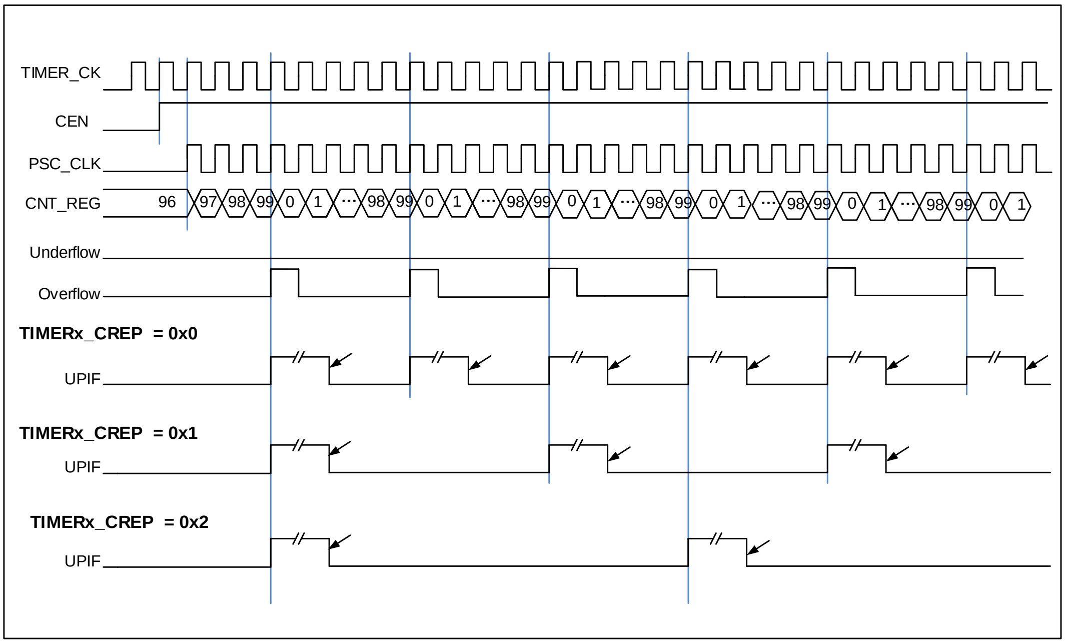


Update event (from overflow/underflow) rate configuration

The rate of update events generation (from overflow and underflow events) can be configured by the TIMERx_CREP register. Counter repetition is used to generator update event or updates the timer registers only after a given number (N+1) of cycles of the counter, where N is CREP in TIMERx_CREP register. The repetition counter is decremented at each counter overflow in up counting mode.

Setting the UPG bit in the TIMERx_SWEVG register will reload the content of CREP in TIMERx_CREP register and generator an update event.

Figure 14-78. Repetition counter timing chart of up counting mode



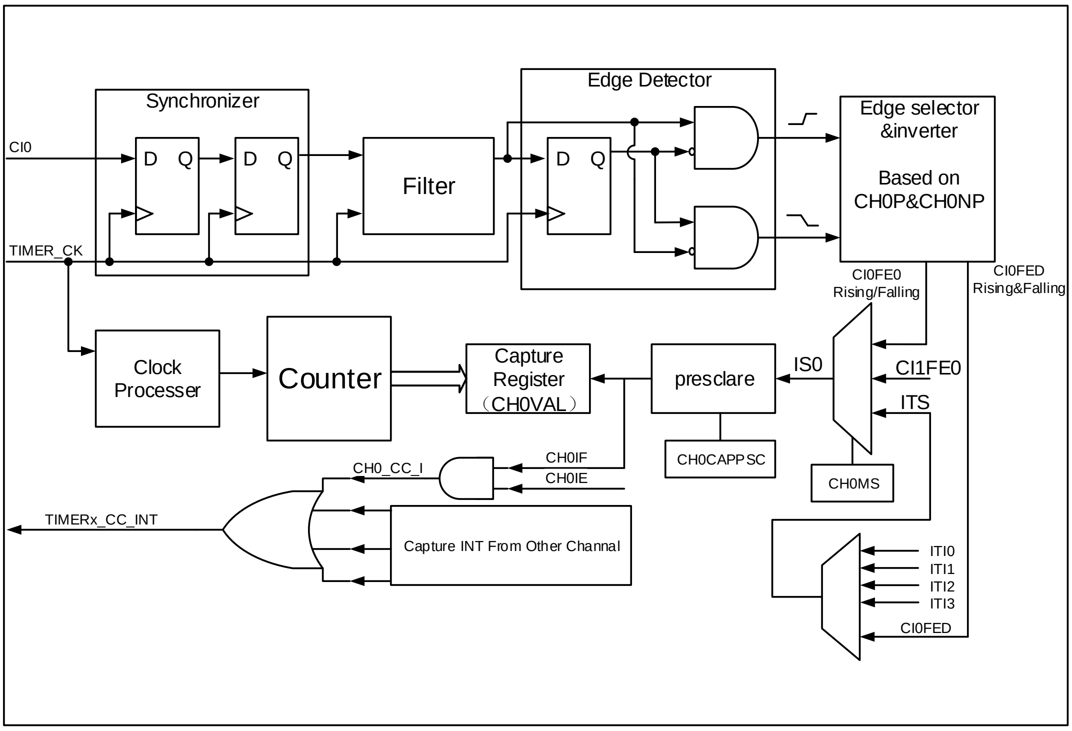
Input capture and output compare channels

The general level4 timer has one independent channels which can be used as capture inputs or compare match outputs. Each channel is built around a channel capture compare register including an input stage, channel controller and an output stage.

- **Channel input capture function**

Capture mode allows the channel to perform measurements such as pulse timing, frequency, period, duty cycle and so on. The input stage consists of a digital filter, a channel polarity selection, edge detection and a channel prescaler. When a selected edge occurs on the channel input, the current value of the counter is captured into the TIMERx_CHxCV register, at the same time the CHxIF bit is set and the channel interrupt is generated if enabled by CHxIE = 1.

Figure 14-79. Channel input capture principle



Channels' input signals (CIx) is the TIMERx_CHx signal. First, the channel input signal (CIx) is synchronized to TIMER_CK domain, and then sampled by a digital filter to generate a filtered input signal. Then through the edge detector, the rising and falling edge are detected. You can select one of them by CHxP. One more selector is for the other channel and trig, controlled by CHxMS. The IC_prescaler make several the input event generate one effective capture event. On the capture event, CHxVAL will restore the value of Counter.

So, the process can be divided to several steps as below:

Step1: Filter configuration. (CHxCAPFLT in TIMERx_CHCTL0)
Based on the input signal and requested signal quality, configure compatible CHxCAPFLT.

Step2: Edge selection. (CHxP/CHxNP in TIMERx_CHCTL2)
Rising or falling edge, choose one by CHxP/CHxNP.

Step3: Capture source selection. (CHxMS in TIMERx_CHCTL0)
As soon as you select one input capture source by CHxMS, you have set the channel to input mode (CHxMS! = 0x0) and TIMERx_CHxCV cannot be written any more.

Step4: Interrupt enable. (CHxIE and CHxDEN in TIMERx_DMAINTEN)
Enable the related interrupt; you can get the interrupt and DMA request.

Step5: Capture enables. (CHxEN in TIMERx_CHCTL2).

Result: when you wanted input signal is got, TIMERx_CHxCV will be set by counter's value. And CHxIF is asserted. If the CHxIF is high, the CHxOF will be asserted also. The interrupt and DMA request will be asserted based on the configuration of CHxIE and CHxDEN in

TIMERx_DMAINTEN.

Direct generation: if you want to generate a DMA request or Interrupt, you can set CHxG by software directly.

- **Channel output compare function**

In channel output compare function, the TIMERx can generate timed pulses with programmable position, polarity, duration and frequency. When the counter matches the value in the CHxVAL register of an output compare channel, the channel (n) output can be set, cleared, or toggled based on CHxCOMCTL. When the counter reaches the value in the CHxVAL register, the CHxIF bit is set and the channel (n) interrupt is generated if CHxIE = 1. And the DMA request will be assert, if CHxDEN =1.

So, the process can be divided to several steps as below:
Step1: Clock Configuration. Such as clock source, clock prescaler and so on.

Step2: Compare mode configuration.
* Set the shadow enable mode by CHxCOMSEN.
* Set the output mode (Set/Clear/Toggle) by CHxCOMCTL.
* Select the active high polarity by CHxP/CHxNP.
* Enable the output by CHxEN.

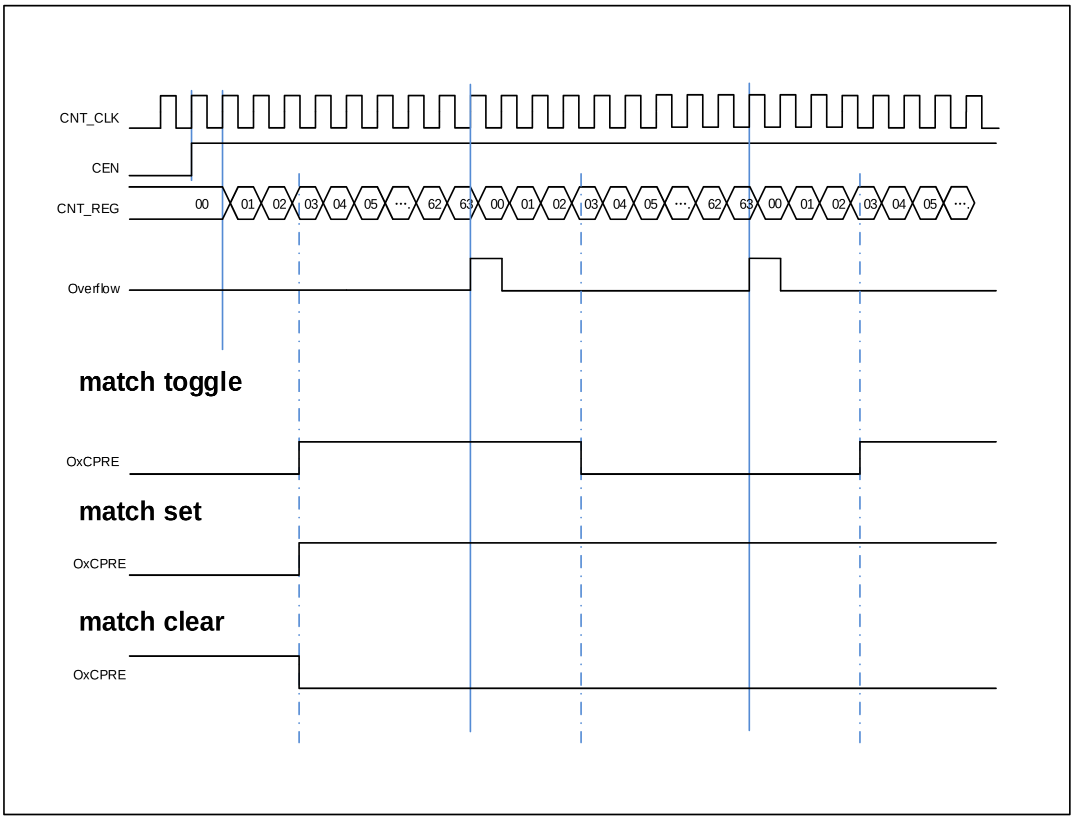
Step3: Interrupt/DMA-request enables configuration by CHxIE/CHxDEN.

Step4: Compare output timing configuration by TIMERx_CAR and TIMERx_CHxCV
About the CHxVAL; you can change it on the go to meet the waveform you expected.

Step5: Start the counter by CEN.

The timechart below show the three compare modes toggle/set/clear. CAR=0x63, CHxVAL=0x3.

Figure 14-80. Output-compare under three modes



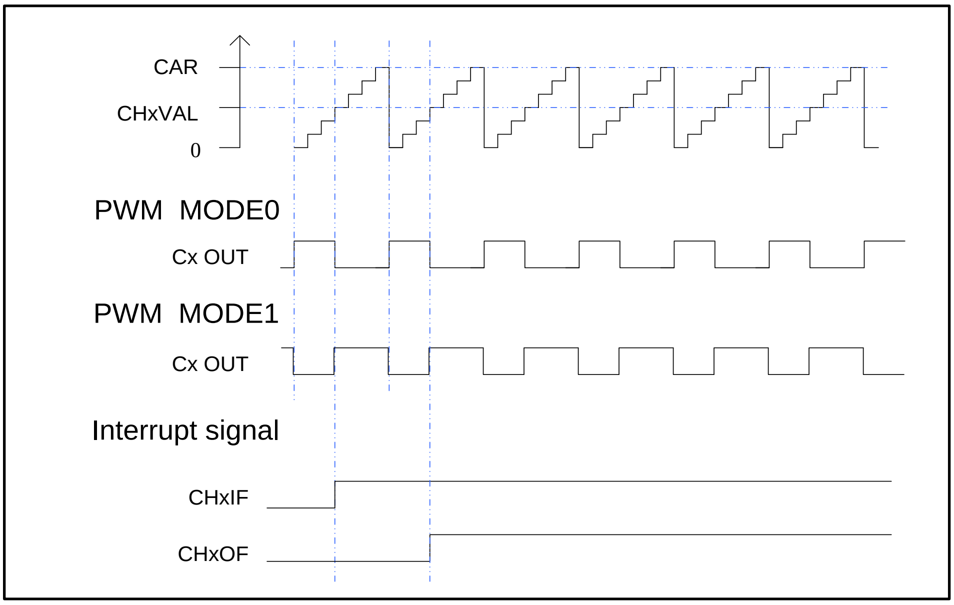
Output PWM function

In the output PWM function (by setting the CHxCOMCTL bits to 3'b110 (PWM mode0) or to 3'b 111(PWM mode1), the channel can generate PWM waveform according to the TIMERx_CAR registers and TIMERx_CHxCV registers.

The period is determined by TIMERx_CAR and duty cycle is determined by TIMERx_CHxCV. ***Figure 14-81. PWM mode timechart*** shows the PWM output mode and interrupts waveform.

If TIMERx_CHxCV is greater than TIMERx_CAR, the output will be always active under PWM mode0 (CHxCOMCTL==3'b110).
And if TIMERx_CHxCV is equal to zero, the output will be always inactive under PWM mode0 (CHxCOMCTL==3'b110).

Figure 14-81. PWM mode timechart



Channel output prepare signal

When the TIMERx is used in the compare match output mode, the OxCPRE signal (Channel x Output prepare signal) is defined by setting the CHxCOMCTL filed. The OxCPRE signal has several types of output function. These include, keeping the original level by setting the CHxCOMCTL field to 0x00, set to 1 by setting the CHxCOMCTL field to 0x01, set to 0 by setting the CHxCOMCTL field to 0x02 or signal toggle by setting the CHxCOMCTL field to 0x03 when the counter value matches the content of the TIMERx_CHxCV register.

The PWM mode 0 and PWM mode 1 outputs are also another kind of OxCPRE output which is setup by setting the CHxCOMCTL field to 0x06/0x07. In these modes, the OxCPRE signal level is changed according to the counting direction and the relationship between the counter value and the TIMERx_CHxCV content. With regard to a more detail description refer to the relative bit definition.

Another special function of the OxCPRE signal is a forced output which can be achieved by setting the CHxCOMCTL field to 0x04/0x05. Here the output can be forced to an inactive/active level irrespective of the comparison condition between the counter and the TIMERx_CHxCV values.

Channel output complementary PWM

Function of complementary is for a pair of CHx_O and CHx_ON. Those two output signals cannot be active at the same time. The TIMERx has only 1 channel have this function. The complementary signals CHx_O and CHx_ON are controlled by a group of parameters: the CHxEN and CHxNEN bits in the TIMERx_CHCTL2 register and the POEN, ROS, IOS, ISOx and ISOxN bits in the TIMERx_CCHP and TIMERx_CTL1 registers. The outputs polarity is determined by CHxP and CHxNP bits in the TIMERx_CHCTL2 register.

Table 14-7. Complementary outputs controlled by parameters

Complementary Parameters					Output Status	
POEN	ROS	IOS	CHxEN	CHxNEN	CHx_O	CHx_ON
0	0/1	0	0	0	CHx_O / CHx_ON = LOW CHx_O / CHx_ON output disable[1].	
				1	CHx_O/ CHx_ON output “off-state” [2]: the CHx_O/ CHx_ON output inactive level firstly: CHx_O = CHxP, CHx_ON = CHxNP; If the clock for deadtime generator is present, after a deadtime: CHx_O = ISOx, CHx_ON = ISOxN. [3]	
			1	0		
				1		
		1	x	x	CHx_O/ CHx_ON output “off-state”: the CHx_O/ CHx_ON output inactive level firstly: CHx_O = CHxP, CHx_ON = CHxNP; If the clock for deadtime generator is present, after a deadtime: CHx_O = ISOx, CHx_ON = ISOxN.	
1	0	0/1	0	0	CHx_O/CHx_ON = LOW CHx_O/CHx_ON output disable.	
				1	CHx_O = LOW CHx_O output disable.	CHx_ON =OxCPRE⊕ [4]CHxNP CHx_ON output enable.
			1	0	CHx_O=OxCPRE⊕CHxP CHx_O output enable.	CHx_ON = LOW CHx_ON output disable.
				1	CHx_O=OxCPRE⊕CHxP CHx_O output enable.	CHx_ON =(!OxCPRE)[5]⊕ CHxNP. CHx_ON output enable.
	1		0	0	CHx_O = CHxP CHx_O output “off-state”.	CHx_ON = CHxNP CHx_ON output “off-state”.
				1	CHx_O = CHxP CHx_O output “off-state”	CHx_ON =OxCPRE⊕CHxNP CHx_ON output enable
			1	0	CHx_O=OxCPRE⊕CHxP CHx_O output enable	CHx_ON = CHxNP CHx_ON output “off-state”.
				1	CHx_O=OxCPRE⊕CHxP CHx_O output enable	CHx_ON =(!OxCPRE)⊕ CHxNP CHx_ON output enable.

Note:

(2) output disable: the CHx_O / CHx_ON are disconnected to corresponding pins, the pin is floating with GPIO pull up/down setting which will be Hi-Z if no pull.

(2) “off-state”: CHx_O / CHx_ON output with inactive state (e.g., CHx_O = 0⊕CHxP = CHxP).

(3) See Break mode section for more details.

(4) ⊕: Xor calculate.

(5) (!OxCPRE)：the complementary output of the OxCPRE signal.

Insertion dead time for complementary PWM

The dead time insertion is enabled when both CHxEN and CHxNEN are configured to 1'b1, it is also necessary to configure POEN to 1. The field named DTCFG defines the dead time delay that can be used for channel 1. The detail about the delay time, refer to the register TIMERx_CCHP.

The dead time delay insertion ensures that no two complementary signals drive the active state at the same time.

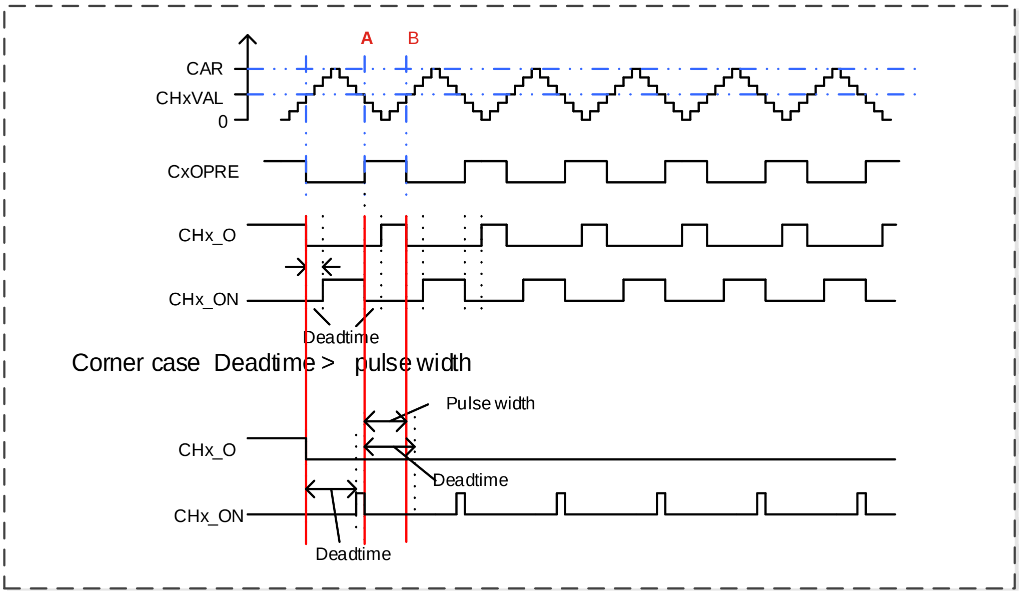
When the channel (x) match (TIMERx counter = CHxVAL) occurs, OxCPRE will be toggled because under PWM0 mode. At point A in the ***Figure 14-82. Complementary output with dead-time insertion***. CHx_O signal remains at the low value until the end of the deadtime delay, while CHx_ON will be cleared at once. Similarly, at point B when counter match (counter = CHxVAL) occurs again, OxCPRE is cleared, CHx_O signal will be cleared at once, while CHx_ON signal remains at the low value until the end of the dead time delay.

Sometimes, we can see corner cases about the dead time insertion. For example:

The dead time delay is greater than or equal to the CHx_O duty cycle, then the CHx_O signal is always the inactive value. (as show in the ***Figure 14-82. Complementary output with dead-time insertion***.)

The dead time delay is greater than or equal to the CHx_ON duty cycle, then the CHx_ON signal is always the inactive value.

Figure 14-82. Complementary output with dead-time insertion.



Break mode

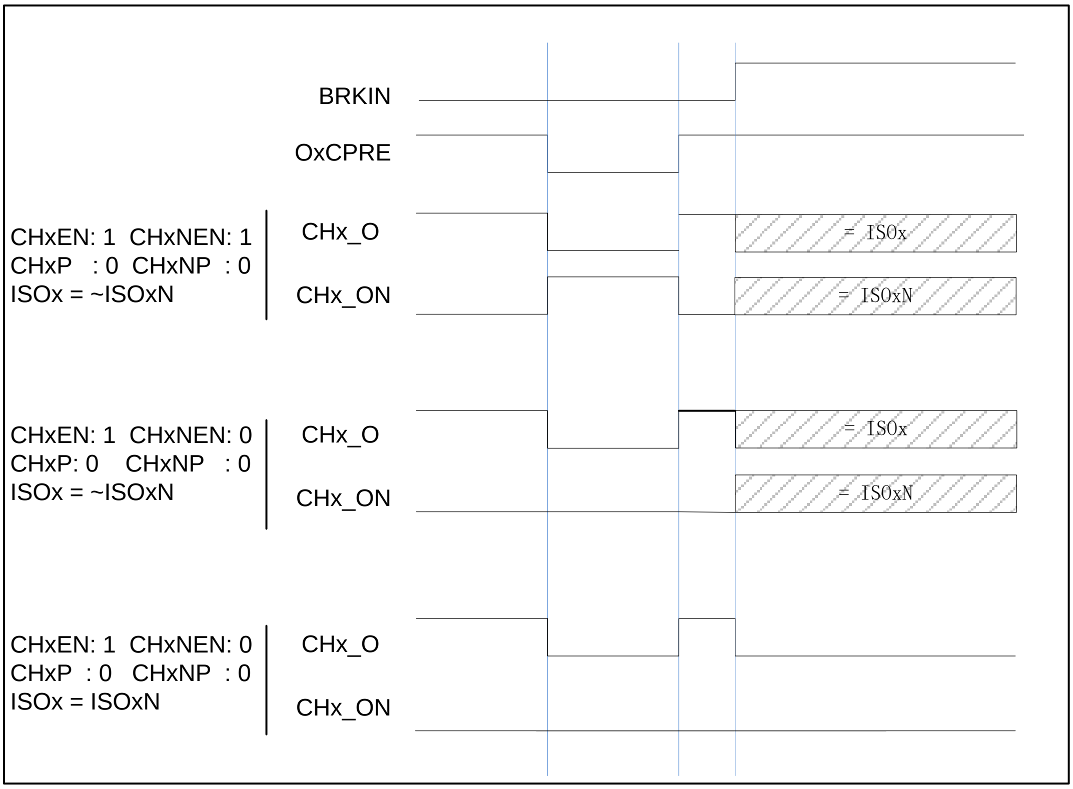
In this mode, the output CHx_O and CHx_ON are controlled by the POEN, IOS and ROS bits in the TIMERx_CCHP register, ISOx and ISOxN bits in the TIMERx_CTL1 register and cannot be set both to active level when break occurs. The break sources are input break pin and

HXTAL stuck event by Clock Monitor (CKM) in RCU. The break function enabled by setting the BRKEN bit in the TIMERx_CCHP register. The break input polarity is setting by the BRKP bit in TIMERx_CCHP.

When a break occurs, the POEN bit is cleared asynchronously, the output CHx_O and CHx_ON are driven with the level programmed in the ISOx bit and ISOxN in the TIMERx_CTL1 register as soon as POEN is 0. If IOS is 0 then the timer releases the enable output else the enable output remains high. The complementary outputs are first put in reset state, and then the dead-time generator is reactivated in order to drive the outputs with the level programmed in the ISOx and ISOxN bits after a dead-time.

When a break occurs, the BRKIF bit in the TIMERx_INTF register is set. If BRKIE is 1, an interrupt generated.

Figure 14-83. Output behavior in response to a break (The break high active)



Single pulse mode

Single pulse mode is opposite to the repetitive mode, which can be enabled by setting SPM in TIMERx_CTL0. When you set SPM, the counter will be clear and stop when the next update event. In order to get pulse waveform, you can set the TIMERx to PWM mode or compare by CHxCOMCTL.

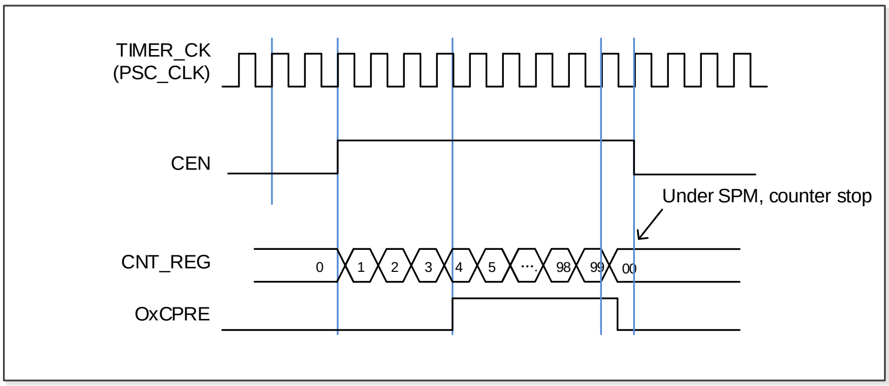
Once the timer is set to operate in the single pulse mode, it is necessary to set the timer enable bit CEN in the TIMERx_CTL0 register to 1 to enable the counter. Setting the CEN bit to 1 can generate a pulse and then keep the CEN bit at a high state until the update event occurs or the CEN bit is written to 0 by software. If the CEN bit is cleared to 0 using software, the counter will be stopped and its value held.

In the single pulse mode, the trigger active edge which sets the CEN bit to 1 will enable the counter. However, there exist several clock delays to perform the comparison result between the counter value and the TIMERx_CHxCV value. In order to reduce the delay to a minimum value, the user can set the CHxCOMFEN bit in each TIMERx_CHCTL0 register. After a trigger rising occurs in the single pulse mode, the OxCPRE signal will immediately be forced to the state which the OxCPRE signal will change to, as the compare match event occurs without taking the comparison result into account. The CHxCOMFEN bit is available only when the output channel is configured to operate in the PWM0 or PWM1 output mode and the trigger source is derived from the trigger signal.

Figure 14-84. Single pulse mode TIMERx_CHxCV = 0x04 TIMERx_CAR=0x60

_shows an example.

Figure 14-84. Single pulse mode TIMERx_CHxCV = 0x04 TIMERx_CAR=0x60



Timer DMA mode

Timer’s DMA mode is the function that configures timer’s register by DMA module. The relative registers are TIMERx_DMACFG and TIMERx_DMATB. Of course, you have to enable a DMA request which will be asserted by some internal event. When the interrupt event was asserted, TIMERx will send a request to DMA, which is configured to M2P mode and PADDR is TIMERx_DMATB, then DMA will access the TIMERx_DMATB. In fact, register TIMERx_DMATB is only a buffer; timer will map the TIMERx_DMATB to an internal register, appointed by the field of DMATA in TIMERx_DMACFG. If the field of DMATC in TIMERx_DMACFG is 0(1 transfer), then the timer’s DMA request is finished. While if TIMERx_DMATC is not 0, such as 3(4 transfers), then timer will send 3 more requests to DMA, and DMA will access timer’s registers DMATA+0x4, DMATA+0x8, DMATA+0xc at the next 3 accesses to TIMERx_DMATB. In a word, one-time DMA internal interrupt event assert, DMATC+1 times request will be send by TIMERx.

If one more time DMA request event coming, TIMERx will repeat the process as above.

Timer debug mode

When the Cortex®-M23 halted, and the TIMERx_HOLD configuration bit in DBG_CTL1 register set to 1, the TIMERx counter stops.

14.4.5. Register definition

TIMER15 base address: 0x4001 4400

TIMER16 base address: 0x4001 4800

Control register 0 (TIMERx_CTL0)

Address offset: 0x00

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved						CKDIV[1:0]		ARSE	Reserved			SPM	UPS	UPDIS	CEN
						rw		rw				rw	rw	rw	rw

Bits	Fields	Descriptions
31:10	Reserved	Must be kept at reset value.
9:8	CKDIV[1:0]	Clock division The CKDIV bits can be configured by software to specify division factor between the CK_TIMER and the dead-time and digital filter sample clock (DTS). 00: $f_{DTS}=f_{CK\_TIMER}$ 01: $f_{DTS}= f_{CK\_TIMER}$ /2 10: $f_{DTS}= f_{CK\_TIMER}$ /4 11: Reserved
7	ARSE	Auto-reload shadow enable 0: The shadow register for TIMERx_CAR register is disabled 1: The shadow register for TIMERx_CAR register is enabled
6:4	Reserved	Must be kept at reset value.
3	SPM	Single pulse mode. 0: Single pulse mode disable. The counter continues after update event. 1: Single pulse mode enable. The counter counts until the next update event occurs.
2	UPS	Update source This bit is used to select the update event sources by software. 0: These events generate update interrupts or DMA requests: The UPG bit is set The counter generates an overflow or underflow event The restart mode generates an update event. 1: This event generates update interrupts or DMA requests:

		The counter generates an overflow or underflow event
1	UPDIS	Update disable. This bit is used to enable or disable the update event generation. 0: Update event enable. When an update event occurs, the corresponding shadow registers are loaded with their preloaded values. These events generate update event: The UPG bit is set The counter generates an overflow or underflow event The restart mode generates an update event. 1: Update event disable. **Note:** When this bit is set to 1, setting UPG bit or the restart mode does not generate an update event, but the counter and prescaler are initialized.
0	CEN	Counter enable 0: Counter disable 1: Counter enable The CEN bit must be set by software when timer works in external clock, pause mode and quadrature decoder mode.

Control register 1 (TIMERx_CTL1)

Address offset: 0x04

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved						ISO0N	ISO0	Reserved				DMAS	CCUC	Reserved	CCSE
						rw	rw					rw	rw		rw

Bits	Fields	Descriptions
31:10	Reserved	Must be kept at reset value.
9	ISO0N	Idle state of channel 0 complementary output 0: When POEN bit is reset, CH0_ON is set low. 1: When POEN bit is reset, CH0_ON is set high This bit can be modified only when PROT [1:0] bits in TIMERx_CCHP register is 00.
8	ISO0	Idle state of channel 0 output 0: When POEN bit is reset, CH0_O is set low. 1: When POEN bit is reset, CH0_O is set high The CH0_O output changes after a dead-time if CH0_ON is implemented. This bit

		can be modified only when PROT [1:0] bits in TIMERx_CCHP register is 00.
7:4	Reserved	Must be kept at reset value.
3	DMAS	DMA request source selection 0: When capture or compare event occurs, the DMA request of channel x is sent 1: When update event occurs, the DMA request of channel x is sent.
2	CCUC	Commutation control shadow register update control When the commutation control shadow enable (for CHxEN, CHxNEN and CHxCOMCTL bits) are set (CCSE=1), these shadow registers update are controlled as below: 0: The shadow registers update by when CMTG bit is set. 1: The shadow registers update by when CMTG bit is set or a rising edge of TRGI occurs. When a channel does not have a complementary output, this bit has no effect.
1	Reserved	Must be kept at reset value.
0	CCSE	Commutation control shadow enable 0: The shadow registers for CHxEN, CHxNEN and CHxCOMCTL bits are disabled. 1: The shadow registers for CHxEN, CHxNEN and CHxCOMCTL bits are enabled. After these bits have been written, they are updated based when commutation event coming. When a channel does not have a complementary output, this bit has no effect.

DMA and interrupt enable register (TIMERx_DMAINTEN)

Address offset: 0x0C

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved						CH0DEN	UPDEN	BRK0IE	Reserved	CMTIE	Reserved			CH0IE	UPIE
						rw	rw	rw		rw				rw	rw

Bits	Fields	Descriptions
31:10	Reserved	Must be kept at reset value.
9	CH0DEN	Channel 0 capture/compare DMA request enable 0: disabled 1: enabled
8	UPDEN	Update DMA request enable 0: disabled

		1: enabled
7	BRKIE	Break interrupt 0 enable 0: disabled 1: enabled
6	Reserved	Must be kept at reset value.
5	CMTIE	Commutation interrupt enable 0: disabled 1: enabled
4:2	Reserved	Must be kept at reset value.
1	CH0IE	Channel 0 capture/compare interrupt enable 0: disabled 1: enabled
0	UPIE	Update interrupt enable 0: disabled 1: enabled

Interrupt flag register (TIMERx_INTF)

Address offset: 0x10

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved		SYSBIF	Reserved			CH0OF	Reserved	BRK0IF	Reserved	CMTIF	Reserved			CH0IF	UPIF
						rc_w0		rc_w0		rc_w0				rc_w0	rc_w0

Bits	Fields	Descriptions
31:14	Reserved	Must be kept at reset value.
13	SYSBIF	System source break interrupt flag This flag is set by hardware when the system sources are active, and cleared by software if the system sources are inactive. 0: No system source break interrupt occurred 1: System source break interrupt occurred **Note**: When this bit is set, this bit must be cleared by software before the channel outputs are restored.
12:10	Reserved	Must be kept at reset value.
9	CH0OF	Channel 0 over capture flag

		When channel 0 is configured in input mode, this flag is set by hardware when a capture event occurs while CH0IF flag has already been set. This flag is cleared by software. 0: No over capture interrupt occurred 1: Over capture interrupt occurred
8	Reserved	Must be kept at reset value.
7	BRK0IF	Break 0 interrupt flag When the break input is inactive, the bit is set by hardware. When the break input is inactive, the bit can be cleared by software. 0: No active level break has been detected. 1: An active level has been detected.
6	Reserved	Must be kept at reset value.
5	CMTIF	Channel commutation interrupt flag This flag is set by hardware when channel's commutation event occurs, and cleared by software 0: No channel commutation interrupt occurred 1: Channel commutation interrupt occurred
4:2	Reserved	Must be kept at reset value.
1	CH0IF	Channel 0 's capture/compare interrupt flag This flag is set by hardware and cleared by software. When channel 0 is in input mode, this flag is set when a capture event occurs. When channel 0 is in output mode, this flag is set when a compare event occurs. 0: No Channel 0 interrupt occurred 1: Channel 0 interrupt occurred
0	UPIF	Update interrupt flag This bit is set by hardware on an update event and cleared by software. 0: No update interrupt occurred 1: Update interrupt occurred

Software event generation register (TIMERx_SWEVG)

Address offset: 0x14

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								BRK0G	Reserved	CMTG	Reserved			CH0G	UPG
								w		w				w	w

Bits	Fields	Descriptions
31:8	Reserved	Must be kept at reset value.
7	BRKG	Break 0 event generation This bit is set by software and cleared by hardware automatically. When this bit is set, the POEN bit is cleared and BRKIF flag is set, related interrupt or DMA transfer can occur if enabled. 0: No generate a break event 1: Generate a break event
6	Reserved	Must be kept at reset value.
5	CMTG	Channel commutation event generation This bit is set by software and cleared by hardware automatically. When this bit is set, channel’s capture/compare control registers (CHxEN, CHxNEN and CHxCOMCTL bits) are updated based on the value of CCSE (in the TIMERx_CTL1). 0: No affect 1: Generate channel’s c/c control update event
4:2	Reserved	Must be kept at reset value.
1	CH0G	Channel 0’s capture or compare event generation This bit is set by software in order to generate a capture or compare event in channel 0, it is automatically cleared by hardware. When this bit is set, the CH0IF flag is set, the corresponding interrupt or DMA request is sent if enabled. In addition, if channel 1 is configured in input mode, the current value of the counter is captured in TIMERx_CH0CV register, and the CH0OF flag is set if the CH0IF flag was already high. 0: No generate a channel 1 capture or compare event 1: Generate a channel 1 capture or compare event
0	UPG	Update event generation This bit can be set by software, and cleared by hardware automatically. When this bit is set, the counter is cleared if the center-aligned or up counting mode is selected, else (down counting) it takes the auto-reload value. The prescaler counter is cleared at the same time. 0: No generate an update event 1: Generate an update event

Channel control register 0 (TIMERx_CHCTL0)

Address offset: 0x18

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

<table>
<tr><td colspan="16">Reserved</td></tr>
</table>

<table>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="8" rowspan="2">Reserved</td><td>Reserved</td><td colspan="3">CH0COMCTL[2:0]</td><td>CH0COM
SEN</td><td>CH0COM
FEN</td><td colspan="2" rowspan="2">CH0MS[1:0]</td></tr>
<tr><td colspan="4">CH0CAPFLT[3:0]</td><td colspan="2">CH0CAPPSC[1:0]</td></tr>
<tr><td colspan="8"></td><td colspan="4">rw</td><td colspan="2">rw</td><td colspan="2">rw</td></tr>
</table>

Output compare mode:

Bits	Fields	Descriptions
31:7	Reserved	Must be kept at reset value.
6:4	CH0COMCTL[2:0]	Channel 0 compare output control This bit-field specifies the compare output mode of the the output prepare signal O0CPRE. In addition, the high level of O0CPRE is the active level, and CH0_O and CH0_ON channels polarity depends on CH0P and CH0NP bits. 000: Timing mode. The O0CPRE signal keeps stable, independent of the comparison between the register TIMERx_CH0CV and the counter TIMERx_CNT. 001: Set the channel output. O0CPRE signal is forced high when the counter is equals to the output compare register TIMERx_CH0CV. 010: Clear the channel output. O0CPRE signal is forced low when the counter is equals to the output compare register TIMERx_CH0CV. 011: Toggle on match. O0CPRE toggles when the counter is equals to the output compare register TIMERx_CH0CV. 100: Force low. O0CPRE is forced to low level. 101: Force high. O0CPRE is forced to high level. 110: PWM mode0. When counting up, O0CPRE is high when the counter is smaller than TIMERx_CH0CV, and low otherwise. When counting down, O0CPRE is low when the counter is larger than TIMERx_CH0CV, and high otherwise. 111: PWM mode1. When counting up, O0CPRE is low when the counter is smaller than TIMERx_CH0CV, and high otherwise. When counting down, O0CPRE is high when the counter is larger than TIMERx_CH0CV, and low otherwise. If configured in PWM mode, the O0CPRE level changes only when the output compare mode is adjusted from “Timing” mode to “PWM” mode or the comparison result changes. This bit cannot be modified when PROT [1:0] bit-filed in TIMERx_CCHP register is 11 and CH0MS bit-filed is 00(COMPARE MODE).
3	CH0COMSEN	Channel 0 compare output shadow enable When this bit is set, the shadow register of TIMERx_CH0CV register, which updates at each update event, will be enabled. 0: Channel 0 output compare shadow disable 1: Channel 0 output compare shadow enable The PWM mode can be used without verifying the shadow register only in single pulse mode (when SPM=1) This bit cannot be modified when PROT [1:0] bit-filed in TIMERx_CCHP register is

		11 and CH0MS bit-filed is 00.
2	CH0COMFEN	Channel 0 output compare fast enable When this bit is set, the effect of an event on the trigger in input on the capture/compare output will be accelerated if the channel is configured in PWM0 or PWM1 mode. The output channel will treat an active edge on the trigger input as a compare match, and CH0_O is set to the compare level independently from the result of the comparison. 0: Channel 0 output quickly compare disable. 1: Channel 0 output quickly compare enable.
1:0	CH0MS[1:0]	Channel 0 I/O mode selection This bit-field specifies the work mode of the channel and the input signal selection. This bit-field is writable only when the channel is not active. (CH0EN bit in TIMERx_CHCTL2 register is reset).). 00: Channel 0 is programmed as output mode 01: Channel 0 is programmed as input mode, IS0 is connected to CI0FE0 10: Reserved 11: Reserved **Note:** When CH0MS[1:0]=11, it is necessary to select an internal trigger input through TRGS bits in TIMERx_SMCFG register.

Input capture mode:

Bits	Fields	Descriptions
31:8	Reserved	Must be kept at reset value.
7:4	CH0CAPFLT[3:0]	Channel 0 input capture filter control The CI0 input signal can be filtered by digital filter and this bit-field configure the filtering capability. Basic principle of digital filter: continuously sample the CI0 input signal according to f_{SAMP} and record the number of times of the same level of the signal. After reaching the filtering capacity configured by this bit, it is considered to be an effective level. The filtering capability configuration is as follows:

CH0CAPFLT [3:0]	Times	f_{SAMP}
4'b0000	Filter disabled.	
4'b0001	2	f_{CK_TIMER}
4'b0010	4	
4'b0011	8	
4'b0100	6	$f_{DTS}/2$
4'b0101	8	
4'b0110	6	$f_{DTS}/4$
4'b0111	8	
4'b1000	6	$f_{DTS}/8$
4'b1001	8	
4'b1010	5	$f_{DTS}/16$

4'b1011	6	
4'b1100	8	
4'b1101	5	f_{DTS}/32
4'b1110	6	
4'b1111	8	

3:2	CH0CAPPSC[1:0]	Channel 0 input capture prescaler This bit-field specifies the factor of the prescaler on channel 0 input. The prescaler is reset when CH0EN bit in TIMERx_CHCTL2 register is clear. 00: Prescaler disable, input capture occurs on every channel input edge 01: The input capture occurs on every 2 channel input edges 10: The input capture occurs on every 4 channel input edges 11: The input capture occurs on every 8 channel input edges
1:0	CH0MS[1:0]	Channel 0 mode selection Same as Output compare mode

Channel control register 2 (TIMERx_CHCTL2)

Address offset: 0x20

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved												CH0NP	CH0NEN	CH0P	CH0EN
												rw	rw	rw	rw

Bits	Fields	Descriptions
31:4	Reserved	Must be kept at reset value.
3	CH0NP	Channel 0 complementary output polarity When channel 0 is configured in output mode, this bit specifies the complementary output signal polarity. 0: Channel 0 complementary output high level is active level 1: Channel 0 complementary output low level is active level When channel 0 is configured in input mode, together with CH0P, this bit is used to define the polarity of CI0. This bit cannot be modified when PROT [1:0] bit-filed in TIMERx_CCHP register is 11 or 10.
2	CH0NEN	Channel 0 complementary output enable When channel 0 is configured in output mode, setting this bit enables the complementary output in channel0. 0: Channel 0 complementary output disabled

		1: Channel 0 complementary output enabled
1	CH0P	Channel 0 capture/compare function polarity When channel 0 is configured in output mode, this bit specifies the output signal polarity. 0: Channel 0 high level is active level 1: Channel 0 low level is active level When channel 0 is configured in input mode, this bit specifies the CI0 signal polarity. [CH0NP, CH0P] will select the active trigger or capture polarity for CI0FE0 or CI1FE0. [CH0NP==0, CH0P==0]: CIxFE0's rising edge is the active signal for capture or trigger operation in slave mode. And CIxFE0 will not be inverted. [CH0NP==0, CH0P==1]: CIxFE0's falling edge is the active signal for capture or trigger operation in slave mode. And CIxFE0 will be inverted. [CH0NP==1, CH0P==0]: Reserved. [CH0NP==1, CH0P==1]: CIxFE0's falling and rising edge are both the active signal for capture or trigger operation in slave mode. And CIxFE0 will be not inverted. This bit cannot be modified when PROT [1:0] bit-filed in TIMERx_CCHP register is 11 or 10.
0	CH0EN	Channel 0 capture/compare function enable When channel 0 is configured in output mode, setting this bit enables CH0_O signal in active state. When channel 0 is configured in input mode, setting this bit enables the capture event in channel0. 0: Channel 0 disabled 1: Channel 0 enabled

Counter register (TIMERx_CNT)

Address offset: 0x24

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

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

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	CNT[15:0]	This bit-filed indicates the current counter value. Writing to this bit-filed can

change the value of the counter.

Prescaler register (TIMERx_PSC)

Address offset: 0x28

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PSC[15:0]															
rw															

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	PSC[15:0]	Prescaler value of the counter clock The TIMER_CK clock is divided by (PSC+1) to generate the counter clock. The value of this bit-filed will be loaded to the corresponding shadow register at every update event.

Counter auto reload register (TIMERx_CAR)

Address offset: 0x2C

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CARL[15:0]															
rw															

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	CARL[15:0]	Counter auto reload value This bit-filed specifies the auto reload value of the counter. **Note:** When the timer is configured in input capture mode, this register must be configured a non-zero value (such as 0xFFFF) which is larger than user expected value.

Counter repetition register (TIMERx_CREP)

Address offset: 0x30

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								CREP[7:0]							
								rw							

Bits	Fields	Descriptions
31:8	Reserved	Must be kept at reset value.
7:0	CREP[7:0]	Counter repetition value This bit-filed specifies the update event generation rate. Each time the repetition counter counting down to zero, an update event is generated. The update rate of the shadow registers is also affected by this bit-filed when these shadow registers are enabled.

Channel 0 capture/compare value register (TIMERx_CH0CV)

Address offset: 0x34

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH0VAL[15:0]															
rw															

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	CH0VAL[15:0]	Capture or compare value of channel0 When channel 0 is configured in input mode, this bit-filed indicates the counter value corresponding to the last capture event. And this bit-filed is read-only. When channel 0 is configured in output mode, this bit-filed contains value to be compared to the counter. When the corresponding shadow register is enabled, the shadow register updates every update event.

Complementary channel protection register (TIMERx_CCHP)

Address offset: 0x44

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved												BRK0F[3:0]			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
POEN	OAEN	BRKP	BRKEN	ROS	IOS	PROT[1:0]		DTCFG[7:0]							
rw	rw	rw	rw	rw	rw	rw		rw							

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value
19:16	BRK0F[3:0]	BREAK0 input signal filter An event counter is used in the digital filter, in which a transition on the output occurs after N input events. This bit-field specifies the frequency used to sample BREAK0 input signal and the length of the digital filter applied to BREAK0. 0000: Filter disabled. BREAK0 act asynchronously, N=1 0001: $f_{SAMP} = f_{CK\_TIMER}$, N=2 0010: $f_{SAMP} = f_{CK\_TIMER}$, N=4 0011: $f_{SAMP} = f_{CK\_TIMER}$, N=8 0100: $f_{SAMP} = f_{DTS}/2$, N=6 0101: $f_{SAMP} = f_{DTS}/2$, N=8 0110: $f_{SAMP} = f_{DTS}/4$, N=6 0111: $f_{SAMP} = f_{DTS}/4$, N=8 1000: $f_{SAMP} = f_{DTS}/8$, N=6 1001: $f_{SAMP} = f_{DTS}/8$, N=8 1010: $f_{SAMP} = f_{DTS}/16$, N=5 1011: $f_{SAMP} = f_{DTS}/16$, N=6 1100: $f_{SAMP} = f_{DTS}/16$, N=8 1101: $f_{SAMP} = f_{DTS}/32$, N=5 1110: $f_{SAMP} = f_{DTS}/32$, N=6 1111: $f_{SAMP} = f_{DTS}/32$, N=8 This bit can be modified only when PROT(LOCK)[1:0] bit-field in TIMERx_CCHP(TIMx_BDTR) register is 00.
15	POEN	Primary output enable The bit can be set to 1 by: - Write 1 to this bit - If OAEN is set to 1, this bit is set to 1 at the next update event.. The bit can be cleared to 0 by: - Write 0 to this bit - Valid fault input. When one of channels is configured in output mode, setting this bit enables the channel outputs (CHx_O and CHx_ON) if the corresponding enable bits (CHxEN, CHxNEN in TIMERx_CHCTL2 register) have been set. 0: Disable channel outputs (CHxO or CHxON).

		1: Enabled channel outputs (CHxO or CHxON). **Note:** This bit is only valid when CHxMS=2’b00.
14	OAEN	Output automatic enable 0: The POEN bit can only be set by software. 1: POEN can be set at the next update event, if the break input is not active. This bit can be modified only when PROT [1:0] bit-filed in TIMERx_CCHP register is 00.
13	BRKP	Break polarity This bit specifies the polarity of the BRKIN input signal. 0: BRKIN input active low 1: BRKIN input active high
12	BRKEN	Break enable This bit can be set to enable the BRKIN and CKM clock failure event inputs. 0: Break inputs disabled 1: Break inputs enabled This bit can be modified only when PROT [1:0] bit-filed in TIMERx_CCHP register is 00.
11	ROS	Run mode “off-state” enable When POEN bit is set (Run mode), this bit can be set to enable the “off-state” for the channels which has been configured in output mode. 0: “off-state” disabled. If the CHxEN or CHxNEN bit is reset, the corresponding channel is output disabled. 1: “off-state” enabled. If the CHxEN or CHxNEN bit is reset, the corresponding channel is “off-state”. This bit cannot be modified when PROT [1:0] bit-filed in TIMERx_CCHP register is 10 or 11.
10	IOS	Idle mode “off-state” enable When POEN bit is reset (Idle mode), this bit can be set to enable the “off-state” for the channels which has been configured in output mode. 0: “off-state” disabled. If the CHxEN/CHxNEN bits are both reset, the channels are output disabled. 1: “off-state” enabled. No matter the CHxEN/CHxNEN bits, the channels are “off-state”. This bit cannot be modified when PROT [1:0] bit-filed in TIMERx_CCHP register is 10 or 11.
9:8	PROT[1:0]	Complementary register protect control This bit-filed specifies the write protection property of registers. 00: protect disable. No write protection. 01: PROT mode 0.The ISOx/ISOxN bits in TIMERx_CTL1 register and the BRKEN/BRKP/OAEN/DTCFG bits in TIMERx_CCHP register are writing protected. 10: PROT mode 1. In addition of the registers in PROT mode 0, the CHxP/CHxNP

bits in TIMERx_CHCTL2 register (if related channel is configured in output mode) and the ROS/IOS bits in TIMERx_CCHP register are writing protected.

11: PROT mode 2. In addition of the registers in PROT mode 1, the CHxCOMCTL/ CHxCOMSEN bits in TIMERx_CHCTL0 registers (if the related channel is configured in output) are writing protected.

This bit-field can be written only once after the reset. Once the TIMERx_CCHP register has been written, this bit-field will be writing protected.

7:0 DTCFG[7:0] Dead time configure

The relationship between DTVAL value and the duration of dead-time is as follow:

DTCFG[7:5]	The duration of dead-time
3’b0xx	DTCFG[7:0] * t_{DTS_CK}
3’b10x	(64+ DTCFG[5:0]) * t_{DTS_CK} *2
3’b110	(32+ DTCFG[4:0]) * t_{DTS_CK} *8
3’b111	(32+ DTCFG[4:0]) * t_{DTS_CK} *16

Note:

1. t_{DTS_CK} is the period of DTS_CK which is configured by CKDIV[1:0] in TIMERx_CTL0.

2. This bit can be modified only when PROT [1:0] bit-filed in TIMERx_CCHP register is 00.

DMA configuration register (TIMERx_DMACFG)

Address offset: 0x48

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved			DMATC[4:0]					Reserved			DMATA [4:0]				
			rw								rw				

Bits **Fields** **Descriptions**

31:14 Reserved Must be kept at reset value.

12:8 DMATC [4:0] DMA transfer count

This filed defines the number(n) of the register that DMA will access(R/W), n = (DMATC [4:0] +1). DMATC [4:0] is from 5’b0_0000 to 5’b1_0001.

7:5 Reserved Must be kept at reset value.

4:0 DMATA [4:0] DMA transfer access start address

This filed define the first address for the DMA access the TIMERx_DMATB. When access is done through the TIMERx_DMA address first time, this bit-field specifies the address you just access. And then the second access to the

TIMERx_DMATB, you will access the address of start address + 0x4.

DMA transfer buffer register (TIMERx_DMATB)

Address offset: 0x4C

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	DMATB[15:0]	DMA transfer buffer When a read or write operation is assigned to this register, the register located at the address range (Start Addr + Transfer Timer* 4) will be accessed. The transfer Timer is calculated by hardware, and ranges from 0 to DMATC.

TIMERx alternate function control register 0 (TIMERx_AFCTL0)

Address offset: 0x60

Reset value: 0x0000 0001

This register has to be accessed by word (32-bit).

Bits	Fields	Descriptions
31:10	Reserved	Must be kept at reset value.
9	BRK0IN0P	BREAK0 BRKIN0 alternate function input polarity This bit is used to configure the BRKIN0 input polarity, and the specitic polarity is determined by this bit and the BRK0P bit. 0: BRKIN0 input signal will not be inverted (BRK0P =0, the input signal is active low; BRK0P =1, the input signal is active high) 1: BRKIN0 input signal will be inverted (BRK0P=0, the input signal is active high; BRK0P =1, the input signal is active low) This bit can be modified only when PROT[1:0] bit-field in TIMERx_CCHP register is 00.
8:1	Reserved	Must be kept at reset value.
0	BRK0IN0EN	BREAK0 BRKIN0 alternate function input enable 0: BRKIN0 alternate function input disabled 1: BRKIN0 alternate function input enabled This bit can be modified only when PROT[1:0] bit-field in TIMERx_CCHP register is 00.

input selection register (TIMERx_INSEL)

Address offset: 0x68

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
												CI0_SEL[3:0]			

Bits	Fields	Descriptions
31:2	Reserved	Must be kept at reset value.
1:0	CI0_SEL[3:0]	Channel 0 input selection 0000: Channel 0 input is connected to TIMERx_CH0 0001: Channel 0 input is connected to IRC32,Reserved for TIMER16 0010: Reserved for TIMER15,Channel 0 input is connected HXTAL/32 0011: Reserved for TIMER15. Channel 0 input is connected CKOUTSEL0 for TIMER16 0100: Channel 0 input is connected to CKOUTSEL1 Other:Reserved

Configuration register (TIMERx_CFG)

Address offset: 0xFC

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved														CHVSEL	OUTSEL
														rw	rw

Bits	Fields	Descriptions
31:2	Reserved	Must be kept at reset value.
1	CHVSEL	Write CHxVAL register selection This bit-field set and reset by software. 1: If write the CHxVAL register, the write value is same as the CHxVAL value, the write access ignored 0: No effect
0	OUTSEL	The output value selection This bit-field set and reset by software 1: If POEN and IOS is 0, the output disabled

0: No effect

15. Universal synchronous/asynchronous receiver / transmitter (USART)

15.1. Overview

The Universal Synchronous / Asynchronous Receiver / Transmitter (USART) provides a flexible serial data exchange interface. Data frames can be transferred in full duplex or half duplex mode, synchronously or asynchronously through this interface. A programmable baud rate generator divides the UCLK (PCLK, CK_SYS, LXTAL or IRC48MDIV_PER) to produces a dedicated wide range baudrate clock for the USART transmitter and receiver.

Besides the standard asynchronous receiver and transmitter mode, the USART implements several other types of serial data exchange modes, such as IrDA (infrared data association) SIR mode, smartcard mode, LIN (local interconnection network) mode, half-duplex mode and synchronous mode. It also supports multiprocessor communication mode, and hardware flow control protocol (CTS / RTS). The data frame can be transferred from LSB or MSB bit. The polarity of the TX / RX pins can be configured independently and flexibly.

All USARTs support DMA function for high-speed data communication.

15.2. Characteristics

- NRZ standard format.
- Asynchronous, full duplex communication.
- Half duplex single wire communications.
- Receive FIFO function.
- Dual clock domain:
  - Asynchronous PCLK and USART clock.
  - Baud rate programming independent from the UCLK reprogramming.
- Programmable baud-rate generator allowing speed up to 6 MBits/s when the clock frequency is 48 MHz and oversampling is by 8.
- Fully programmable serial interface characteristics:
  - A data word (8 or 9 bits) LSB or MSB first.
  - Even, odd or no-parity bit generation/detection.
  - 0.5, 1, 1.5 or 2 stop bit generation.
- Swappable Tx / Rx pin.
- Configurable data polarity.
- Hardware modem operations (CTS / RTS) and RS485 drive enable.
- Configurable multibuffer communication using centralized DMA.
- Separate enable bits for transmitter and receiver.
- Parity control:
  - Transmits parity bit.

  - Checks parity of received data byte.
- LIN break generation and detection.
- IrDA support.
- Synchronous mode and transmitter clock output for synchronous transmission.
- ISO 7816-3 compliant smartcard interface:
  - Character mode (T=0).
  - Block mode (T=1).
  - Direct and inverse convention.
- Multiprocessor communication:
  - Enter into mute mode if address match does not occur.
  - Wake up from mute mode by idle line or address match detection.
- Support for ModBus communication:
  - Timeout feature.
  - CR / LF character recognition.
- Wake up from deep-sleep mode:
  - By standard RBNE interrupt.
  - By WUF interrupt.
- Various status flags:
  - Flags for transfer detection: Receive buffer not empty (RBNE), receive FIFO full (RFF), Transmit buffer empty (TBE), transfer complete (TC).
  - Flags for error detection: overrun error (ORERR), noise error (NERR), frame error (FERR) and parity error (PERR).
  - Flag for hardware flow control: CTS changes (CTSF).
  - Flag for LIN mode: LIN break detected (LBDF).
  - Flag for multiprocessor communication: IDLE frame detected (IDLEF).
  - Flag for ModBus communication: address/character match (AMF) and receiver timeout (RTF).
  - Flags for smartcard block mode: end of block (EBF) and receiver timeout (RTF).
  - Wakeup from deep-sleep mode flag.
  - Interrupt occurs at these events when the corresponding interrupt enable bits are set.

While USART0 is fully implemented, USART1 / USART2 is only partially implemented with the following features not supported.

- Receive FIFO function.
- Smartcard mode.
- IrDA SIR ENDEC block.
- LIN mode.
- Dual clock domain and wakeup from deep-sleep mode.
- Receiver timeout interrupt.
- ModBus communication.

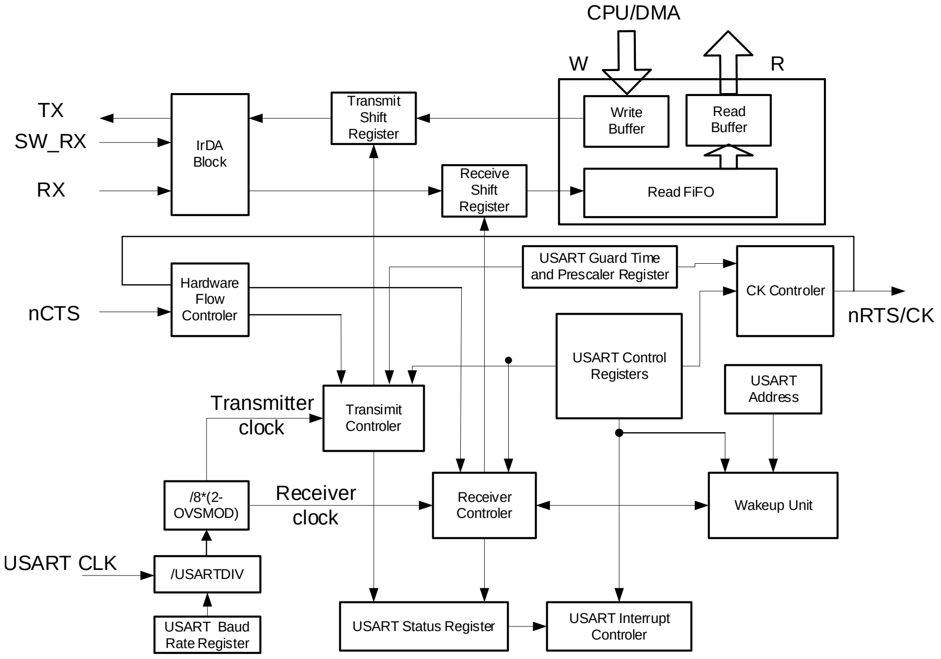
15.3. Function overview

The interface is externally connected to another device by the main pins listed in ***Table 15-1. Description of USART important pins***.

Table 15-1. Description of USART important pins

Pin	Type	Description
RX	Input	Receive Data
TX	Output I/O (single-wire/smartcard mode)	Transmit Data. High level When enabled but nothing to be transmitted
nCTS	Input	Clear to send in Hardware flow control mode
nRTS/CK	Output	Request to send in Hardware flow control mode/ Serial clock for synchronous communication

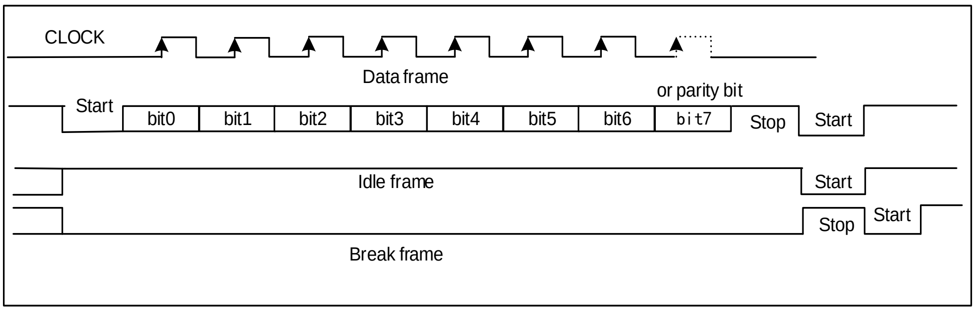
Figure 15-1. USART module block diagram



15.3.1. USART frame format

The USART frame starts with a start bit and ends up with a number of stop bits. The length of the data frame is configured by the WL bit in the USART_CTL0 register. The last data bit can be used as parity check bit by setting the PCEN bit of in USART_CTL0 register. When the WL bit is reset, the parity bit is the 7th bit. When the WL bit is set, the parity bit is the 8th bit. The method of calculating the parity bit is selected by the PM bit in USART_CTL0 register.

Figure 15-2. USART character frame (8 bits data and 1 stop bit)



In transmission and reception, the number of stop bits can be configured by the STB[1:0] bits in the USART_CTL1 register.

Table 15-2. Configuration of stop bits

STB[1:0]	stop bit length (bit)	usage description
00	1	Default value
01	0.5	Smartcard mode for receiving
10	2	Normal USART and single-wire modes
11	1.5	Smartcard mode for transmitting and receiving

In an idle frame, all the frame bits are logic 1. The frame length is equal to the normal USART frame.

The break frame structure is a number of low bits followed by the configured number of stop bits. The transfer speed of a USART frame depends on the frequency of the UCLK, the configuration of the baud rate generator and the oversampling mode.

15.3.2. Baud rate generation

The baud-rate divider is a 16-bit number which consists of a 12-bit integer and a 4-bit fractional part. The number formed by these two values is used by the baud rate generator to determine the bit period. Having a fractional baud-rate divider allows the USART to generate all the standard baud rates.

The baud-rate divider (USARTDIV) has the following relationship with the USART clock:

In case of oversampling by 16, the equation is:

$$USARTDIV=\frac{UCLK}{16\times Baud\ Rate} \qquad (15\text{-}1)$$

In case of oversampling by 8, the equation is:

$$USARTDIV=\frac{UCLK}{8\times Baud\ Rate} \qquad (15\text{-}2)$$

Note: Only USART0 supports dual clock domain. The UCLK is PCLK for USART1 and USART2.

For example, when oversampled by 16:

1. Get USARTDIV by caculating the value of USART_BAUD:

If USART_BAUD=0x21D, then INTDIV=33 (0x21), FRADIV=13 (0xD).
USARTDIV=33+13/16=33.81.

2. Get the value of USART_BAUD by calculating the value of USARTDIV:
   If USARTDIV=30.37, then INTDIV=30 (0x1E).
   16*0.37=5.92, the nearest integer is 6, so FRADIV=6 (0x6).
   USART_BAUD=0x1E6.

   Note: If the roundness of FRADIV is 16 (overflow), the carry must be added to the integer part.

15.3.3. USART transmitter

If the transmit enable bit (TEN) in USART_CTL0 register is set, when the transmit data buffer is not empty, the transmitter shifts out the transmit data frame through the TX pin. The polarity of the TX pin can be configured by the TINV bit in the USART_CTL1 register. Clock pulses can output through the CK pin.

After the TEN bit is set, an idle frame will be sent. The TEN bit should not be cleared while the transmission is ongoing.

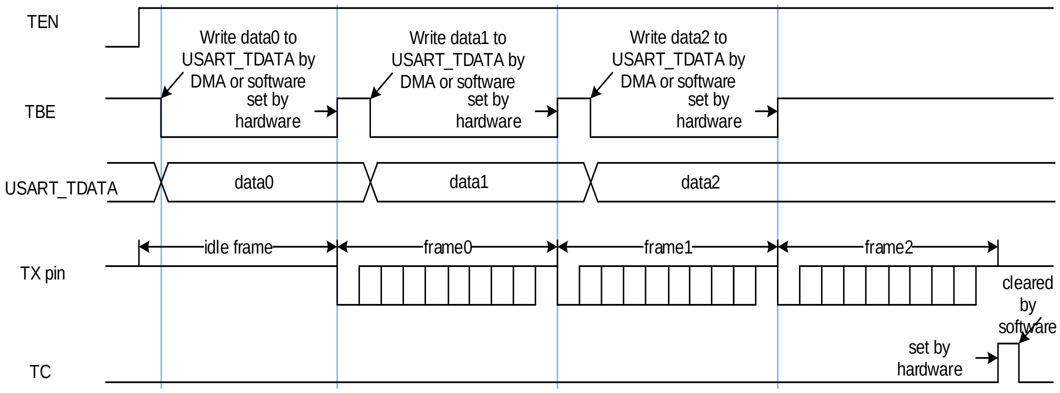
After power on, the TBE bit is high by default. Data can be written to the USART_TDATA when the TBE bit in the USART_STAT register is asserted. The TBE bit is cleared by writing USART_TDATA register and it is set by hardware after the data is put into the transmit shift register. If a data is written to the USART_TDATA register while a transmission is ongoing, it will be firstly stored in the transmit buffer, and transferred to the transmit shift register after the current transmission is done. If a data is written to the USART_TDATA register while no transmission is ongoing, the TBE bit will be cleared and set soon, because the data will be transferred to the transmit shift register immediately.

If a frame is transmitted and the TBE bit is asserted, the TC bit of the USART_STAT register will be set. An interrupt will be generated if the corresponding interrupt enable bit (TCIE) is set in the USART_CTL0 register.

The USART transmit procedure is shown in ***Figure 15-3. USART transmit procedure***. The software operating process is as follows:

1. Write the WL bit in USART_CTL0 to set the data bits length.
2. Set the STB[1:0] bits in USART_CTL1 to configure the number of stop bits.
3. Enable DMA (DENT bit) in USART_CTL2 if multibuffer communication is selected.
4. Set the baud rate in USART_BAUD.
5. Set the UEN bit in USART_CTL0 to enable the USART.
6. Set the TEN bit in USART_CTL0.
7. Wait for the TBE being asserted.
8. Write the data to the USART_TDATA register.
9. Repeat step7-8 for each data, if DMA is not enabled.
10. Wait until TC=1 to finish.

Figure 15-3. USART transmit procedure



It is necessary to wait for the TC bit to be asserted before disabling the USART or entering the power saving mode. This bit can be cleared by set the TCC bit in USART_INTC register.

The break frame is sent when the SBKCMD bit is set, and SBKCMD bit is reset after the transmission.

15.3.4. USART receiver

After power on, the USART receiver can be enabled by the following procedure:

1. Write the WL bit in USART_CTL0 to set the data bits length.
2. Set the STB[1:0] bits in USART_CTL1.
3. Enable DMA (DENR bit) in USART_CTL2 if multibuffer communication is selected.
4. Set the baud rate in USART_BAUD.
5. Set the UEN bit in USART_CTL0 to enable the USART.
6. Set the REN bit in USART_CTL0.

After being enabled, the receiver receives a bit stream after a valid start pulse has been detected. Detection on noisy error, parity error, frame error and overrun error is performed during the reception of a frame.

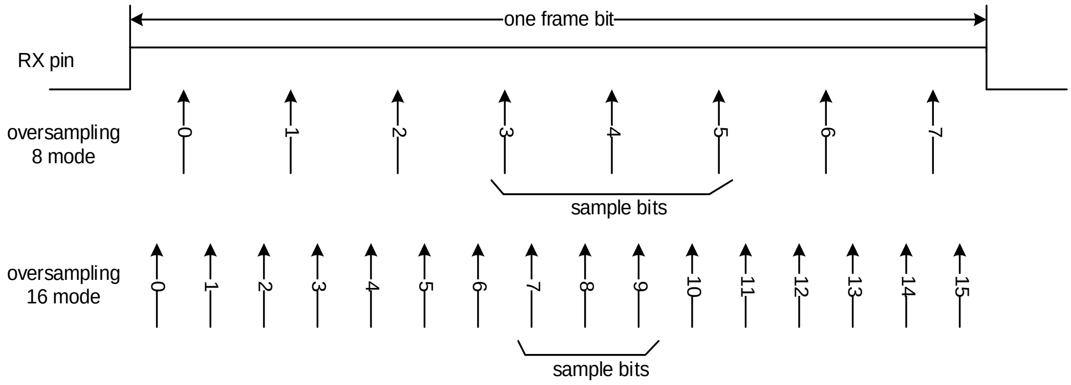
When a frame is received, the RBNE bit in USART_STAT is asserted, an interrupt is generated if the corresponding interrupt enable bit (RBNEIE) is set in the USART_CTL0 register. The status of the reception are stored in the USART_STAT register.

The software can get the received data by reading the USART_RDATA register directly, or through DMA. The RBNE bit is cleared by a read operation on the USART_RDATA register, whatever it is performed by software directly, or through DMA.

The REN bit should not be disabled when reception is ongoing, or the current frame will be lost.

By default, the receiver gets three samples to evaluate the value of a frame bit. If the oversampling 8 mode is enabled, the 3rd, 4th and 5th samples are used, while in the oversampling 16 mode, the 7th, 8th, and 9th samples are used. If two or more samples of a frame bit is 0, the frame bit is confirmed as a 0, else 1. If the value of the three samples of any bit are not the same, whatever it is a start bit, data bit, parity bit or stop bit, a noisy error (NERR) status will be generated for the frame. An interrupt will be generated, If the ERRIE bit in USART_CTL2 register is set. If the OSB bit in USART_CTL2 register is set, the receiver gets only one sample to evaluate a bit value. In this situation, no noisy error will be detected.

Figure 15-4. Oversampling method of a receive frame bit (OSB=0)



If the parity check function is enabled by setting the PCEN bit in the USART_CTL0 register, the receiver calculates the expected parity value while receiving a frame. The received parity bit will be compared with this expected value. If they are not the same, the parity error (PERR) bit in USART_STAT register will be set. An interrupt is generated, if the PERRIE bit in USART_CTL0 register is set.

If the RX pin is evaluated as 0 during a stop bit, the frame error (FERR) bit in USART_STAT register will be set. An interrupt is generated, If the ERRIE bit in USART_CTL2 register is set. According to the configuration of the stop bit, there are the following situations:

- 0.5 stop bit: When 0.5 stop bit, stop bit is not sampled
- 1 stop bit: When 1 stop bit, sampling in the middle of stop bit.
- 1.5 stop bits: When 1.5 stop bits, the 1.5 stop bits are divided into 2 parts: the 0.5 stop bit part is not sampled and sampling in the middle of 1 stop bit.
- 2 stop bits: When 2 stop bits, if a frame error is detected during the first stop bit, the frame error flag is set, the second stop bit is not checked frame error. If no frame error is detected during the first stop bit, then continue to check the second stop bit for frame error.

When a frame is received, if the RBNE bit is not cleared yet, the last frame will not be stored in the receive data buffer. The overrun error (ORERR) bit in USART_STAT register will be set. An interrupt is generated, if the ERRIE bit in USART_CTL2 register is set, or if the RBNEIE is set.

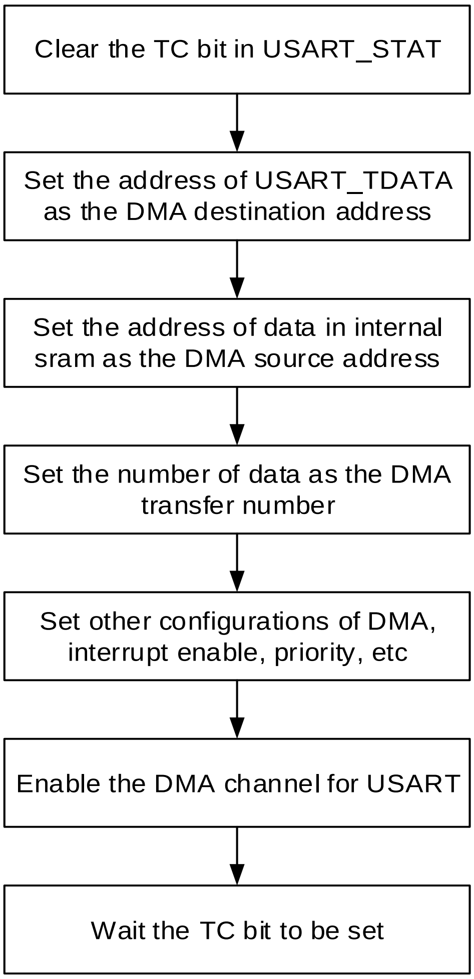
The RBNE, NERR, PERR, FERR and ORERR flags are always set at the same time in a reception. If the receive DMA is not enabled, software can check NERR, PERR, FERR and ORERR flags when serving the RBNE interrupt.

15.3.5. Use DMA for data buffer access

To reduce the burden of the processor, DMA can be used to access the transmitting and receiving data buffer. The DENT bit in USART_CTL2 is used to enable the DMA transmission, and the DENR bit in USART_CTL2 is used to enable the DMA reception.

When DMA is used for USART transmission, DMA transfers data from internal SRAM to the transmit data buffer of the USART. The configuration step are shown in ***Figure 15-5. Configuration step when using DMA for USART transmission***.

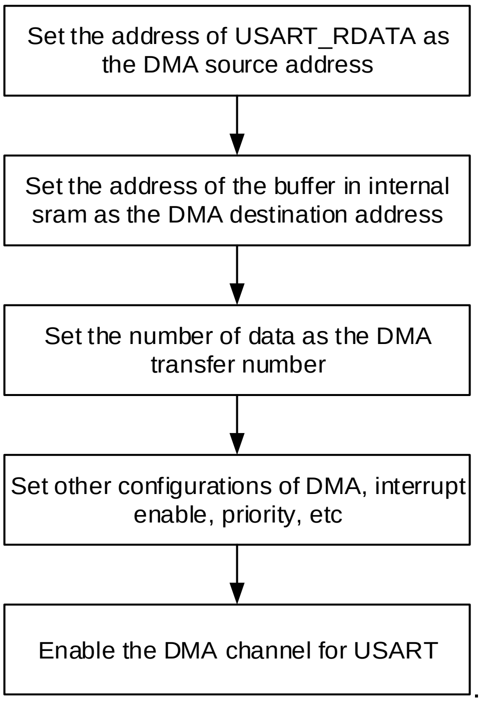
Figure 15-5. Configuration step when using DMA for USART transmission


Clear the TC bit in USART_STAT
Set the address of USART_TDATA as the DMA destination address
Set the address of data in internal sram as the DMA source address
Set the number of data as the DMA transfer number
Set other configurations of DMA, interrupt enable, priority, etc
Enable the DMA channel for USART
Wait the TC bit to be set

After all of the data frames are transmitted, the TC bit in USART_STAT is set. An interrupt occurs if the TCIE bit in USART_CTL0 is set.

When DMA is used for USART reception, DMA transfers data from the receive data buffer of the USART to the internal SRAM. The configuration steps are shown in ***Figure 15-6. Configuration step when using DMA for USART reception***. If the ERRIE bit in USART_CTL2 is set, interrupts can be generated by the Error status bits (FERR, ORERR and NERR) in USART_STAT.

Figure 15-6. Configuration step when using DMA for USART reception

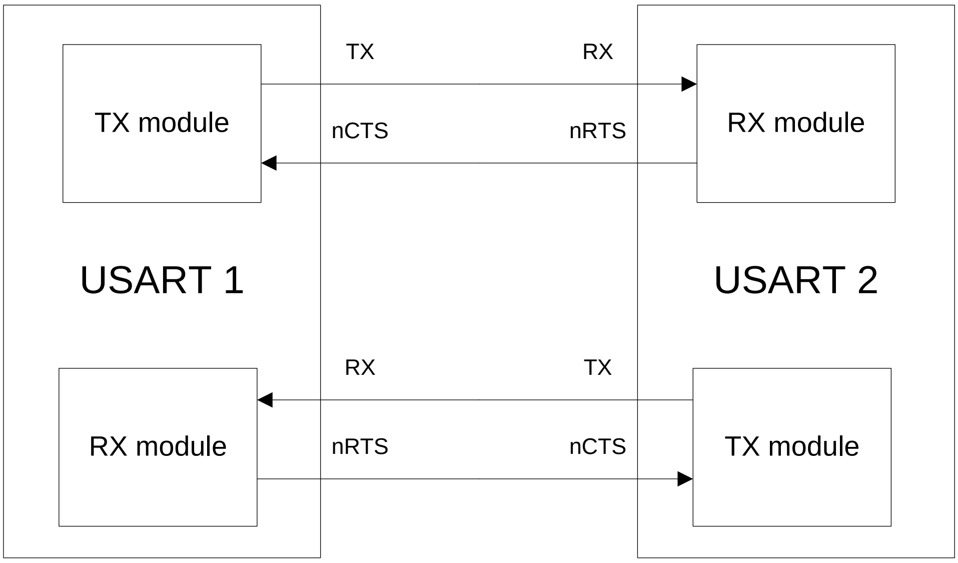

Set the address of USART_RDATA as the DMA source address
Set the address of the buffer in internal sram as the DMA destination address
Set the number of data as the DMA transfer number
Set other configurations of DMA, interrupt enable, priority, etc
Enable the DMA channel for USART

When the number of the data received by USART reaches the DMA transfer number, an end of transfer interrupt can be generated in the DMA module.

15.3.6. Hardware flow control

The hardware flow control function is realized by the nCTS and nRTS pins. The RTS flow control is enabled by writing ‘1’ to the RTSEN bit in USART_CTL2 and the CTS flow control is enabled by writing ‘1’ to the CTSEN bit in USART_CTL2.

Figure 15-7. Hardware flow control between two USARTs



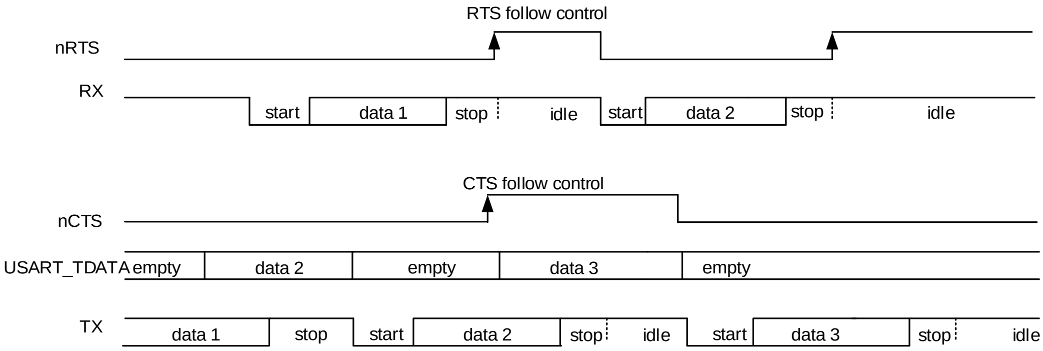
RTS flow control

The USART receiver outputs the nRTS, which reflects the status of the receive buffer. When data frame is received, the nRTS signal goes high to prevent the transmitter from sending next frame. The nRTS signal keeps high when the receive buffer is full.

CTS flow control

The USART transmitter monitors the nCTS input pin to decide whether a data frame can be transmitted. If the TBE bit in USART_STAT is ‘0’ and the nCTS signal is low, the transmitter transmits the data frame. When the nCTS signal goes high during a transmission, the transmitter stops after the current transmission is accomplished.

Figure 15-8. Hardware flow control



RS485 Driver Enable

The driver enable feature, which is enabled by setting bit DEM in the USART_CTL2 control register, allows the user to activate the external transceiver control, through the DE (Driver Enable) signal. The assertion time, which is programmed using the DEA [4:0] bits field in the USART_CTL0 control register, is the time between the activation of the DE signal and the beginning of the START bit. The de-assertion time, which is programmed using the DED [4:0] bits field in the USART_CTL0 control register, is the time between the end of the last stop bit and the de-activation of the DE signal. The polarity of the DE signal can be configured using the DEP bit in the USART_CTL2 control register.

15.3.7. Multi-processor communication

In multiprocessor communication, several USARTs are connected as a network. It will be a big burden for a device to monitor all of the messages on the RX pin. To reduce the burden of a device, software can put an USART module into a mute mode by writing 1 to the MMCMD bit in USART_CMD register.

If a USART is in mute mode, all of the receive status bits cannot be set. The USART can also be wake up by hardware by one of the two methods: idle frame method and address match method.

The idle frame wake up method is selected by default. If the RWU bit is reset, an idle frame is detected on the RX pin, the IDLEF bit in USART_STAT will be set. If the RWU bit is set, an idle frame is detected on the RX pin, the hardware clears the RWU bit and exits the mute mode. When it is woken up by an idle frame, the IDLEF bit in USART_STAT will not be set.

When the WM bit of in USART_CTL0 register is set, the MSB bit of a frame is detected as the address flag. If the address flag is high, the frame is treated as an address frame. If the address flag is low, the frame is treated as a data frame. If the LSB 4 or 7 bits, which are configured by the ADDM bit of the USART_CTL1 register, of an address frame is the same as the ADDR bits in the USART_CTL1 register, the hardware will clear the RWU bit and exits the mute mode. The RBNE bit will be set when the frame that wakes up the USART. The status bits are available in the USART_STAT register. If the LSB 4/7 bits of an address frame defers from the ADDR bits in the USART_CTL1 register, the hardware sets the RWU bit and enters mute mode automatically. In this situation, the RBNE bit is not set.

If the PCEN bit in USART_CTL0 is set, the MSB bit will be checked as the parity bit, and the bit preceding the MSB bit is detected as the address bit. If the ADDM bit is set and the receive frame is a 7bit data, the LSB 6 bits will be compared with ADDR[5:0]. If the ADDM bit is set and the receive frame is a 9bit data, the LSB 8 bits will be compared with ADDR[7:0].

Note: If the MEN bit is set, the WM bit is reset and the RWU bit is reset, an idle frame is detected on the RX pin, the IDLEF bit will be set. If the RWU bit is set, the IDLEF is not set.

15.3.8. LIN mode

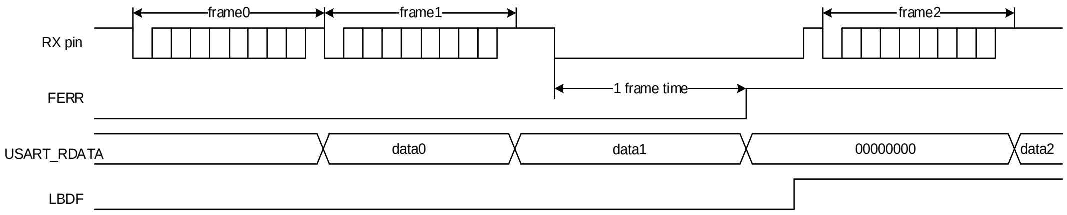
The local interconnection network mode is enabled by setting the LMEN bit in USART_CTL1. The CKEN, STB[1:0] bit in USART_CTL1 and the SCEN, HDEN, IREN bits in USART_CTL2 should be cleared in LIN mode.

When transmitting a normal data frame, the transmission procedure is the same as the normal USART mode. The data bits length can only be 8. And the break frame is 13-bit ’0’, followed by 1 stop bit.

The break detection function is totally independent of the normal USART receiver. So a break frame can be detected during the idle state or during a frame. The expected length of a break frame can be selected by configuring LBLEN in USART_CTL1. When the RX pin is detected at low state for a time that is equal to or longer than the expected break frame length (10 bits when LBLEN=0, or 11 bits when LBLEN=1), the LBDF bit in USART_STAT is set. An interrupt occurs if the LBDIE bit in USART_CTL1 is set.

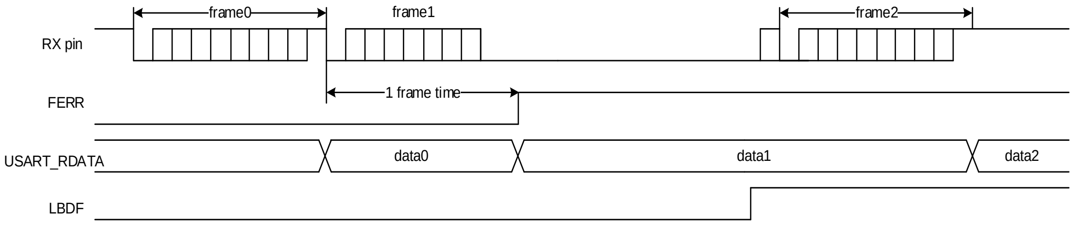
As shown in ***Figure 15-9. Break frame occurs during idle state***, if a break frame occurs during the idle state on the RX pin, the USART receiver will receive an all ‘0’ frame, with an asserted FERR status.

Figure 15-9. Break frame occurs during idle state



As shown in *Figure 15-10. Break frame occurs during a frame*, if a break frame occurs during a frame on the RX pin, the FERR status will be asserted for the current frame.

Figure 15-10. Break frame occurs during a frame



15.3.9. Synchronous mode

The USART can be used for full-duplex synchronous serial communications only in master mode, by setting the CKEN bit in USART_CTL1. The LMEN bit in USART_CTL1 and SCEN, HDEN, IREN bits in USART_CTL2 should be cleared in synchronous mode. The CK pin is the clock output of the synchronous USART transmitter, and can be only activated when the TEN bit is enabled. No clock pulse will be sent through the CK pin during the transmission of the start bit and stop bit. The CLEN bit in USART_CTL1 can be used to determine whether the clock is output or not during the last (address flag) bit transmission. The clock output is also not activated during idle and break frame sending. The CPH bit in USART_CTL1 can be used to determine whether data is captured on the first or the second clock edge. The CPL bit in USART_CTL1 can be used to configure the clock polarity in the USART Synchronous idle state.

The CPL, CPH and CLEN bits in USART_CTL1 determine the waveform on the CK pin. Software can only change them when the USART is disabled (UEN=0).

The clock is synchronized with the data transmitted. The receiver in synchronous mode samples the data on the transmitter clock without any oversampling.

Figure 15-11. Example of USART in synchronous mode

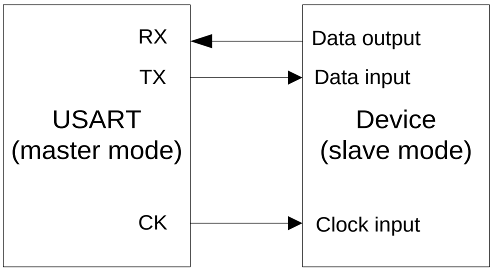
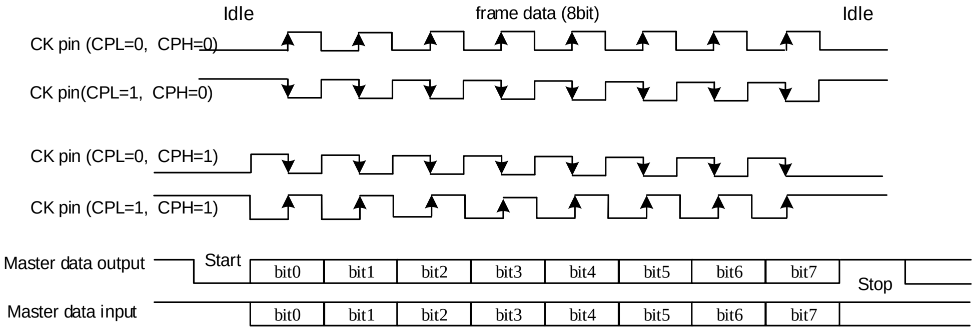


Figure 15-12. 8-bit format USART synchronous waveform (CLEN=1)

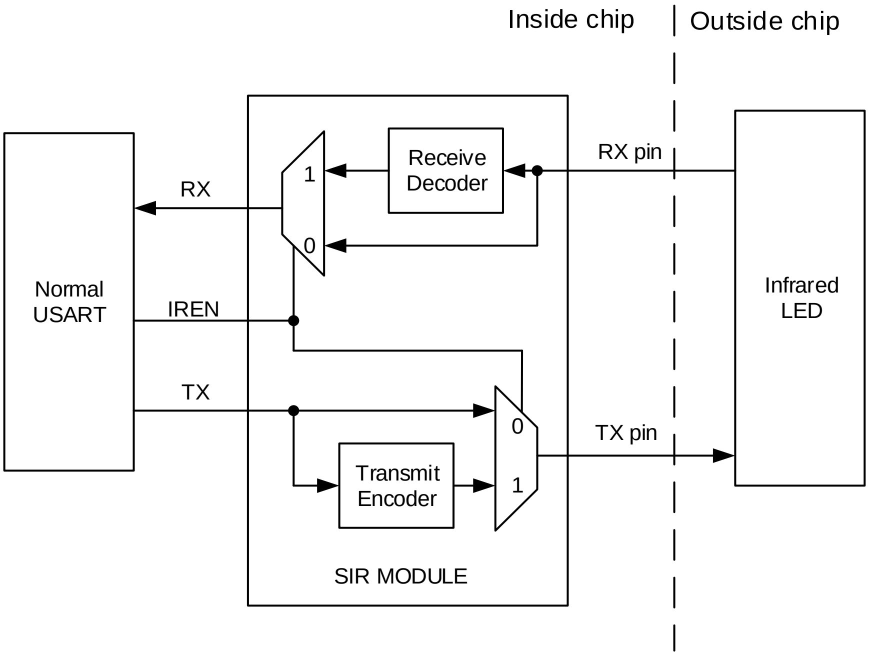


15.3.10. IrDA SIR ENDEC mode

The IrDA mode is enabled by setting the IREN bit in USART_CTL2. The LMEN, STB[1:0], CKEN bits in USART_CTL1 and HDEN, SCEN bits in USART_CTL2 should be cleared in IrDA mode.

In IrDA mode, the USART transmission data frame is modulated in the SIR transmit quadrature decoder and transmitted to the infrared LED through the TX pin. The SIR receive decoder receives the modulated signal from the infrared LED through the RX pin, and puts the demodulated data frame to the USART receiver. The baud rate should not be larger than 115200 for the quadrature decoder.

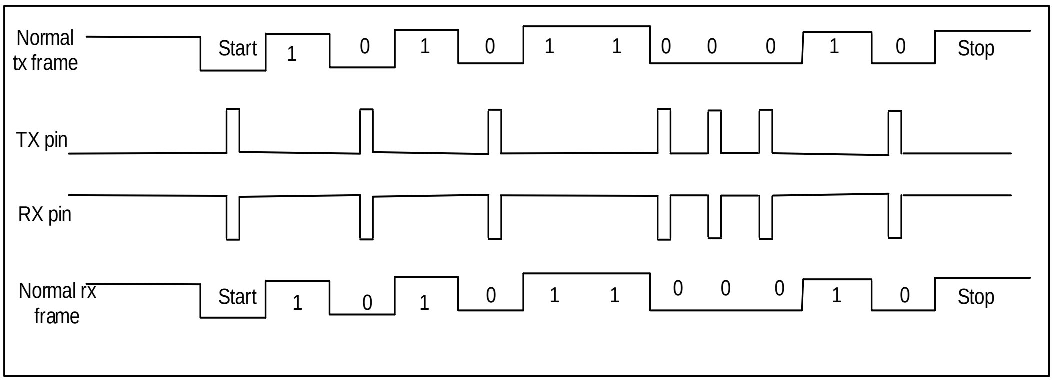
Figure 15-13. IrDA SIR ENDEC module



In IrDA mode, the polarity of the TX and RX pins is different. The TX pin is usually at low state, while the RX pin is usually at high state. The IrDA pins keep stable to represent the logic ‘1’, while an infrared light pulse on the IrDA pins (a Return to Zero signal) represents the logic ‘0’. The pulse width should be 3/16 of a bit period. The IrDA could not detect any pulse if the pulse width is less than 1 PSC clock. While it can detect a pulse by chance if the pulse width is greater than 1 but smaller than 2 times of PSC clock.

Because the IrDA is a half-duplex protocol, the transmission and the reception should not be carried out at the same time in the IrDA SIR ENDEC block.

Figure 15-14. IrDA data modulation



The SIR sub module can work in low power mode by setting the IRLP bit in USART_CTL2. The transmit quadrature decoder is driven by a low speed clock, which is divided from the PCLK. The division ratio is configured by the PSC[7:0] bits in USART_GP register. The pulse width on the TX pin is 3 cycles of this low speed period. The receiver decoder works in the same manner as the normal IrDA mode.

15.3.11. Half-duplex communication mode

The half-duplex communication mode is enabled by setting the HDEN bit in USART_CTL2. The LMEN, CKEN bits in USART_CTL1 and SCEN, IREN bits in USART_CTL2 should be cleared in half-duplex communication mode.

Only one wire is used in half-duplex mode. The TX and RX pins are connected together internally, and the RX pin is no longer used. The TX pin should be configured in open-drain mode, and communication conflicts should be handled by software.

15.3.12. Smartcard (ISO7816-3) mode

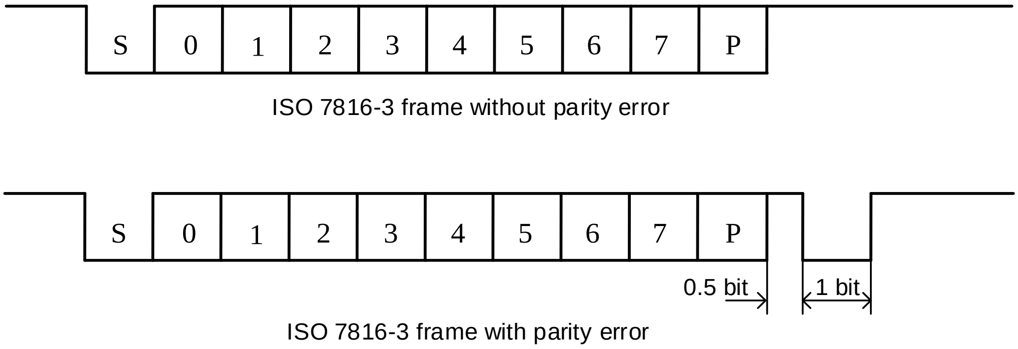
The smartcard mode is an asynchronous mode, which is designed to support the ISO7816-3 protocol. Both the character (T=0) mode and the block (T=1) mode are supported. The smartcard mode is enabled by setting the SCEN bit in USART_CTL2. The LMEN bit in USART_CTL1 and HDEN, IREN bits in USART_CTL2 should be reset in smartcard mode.

A clock is provided to the smartcard if the CKEN bit is set. The clock can be divided for other use.

The frame consists of 1 start bit, 9 data bits (1 parity bit included) and 1.5 stop bits.

The smartcard mode is a half-duplex communication protocol. When connected to a smartcard, the TX pin must be configured as open drain mode, and drives a bidirectional line that is also driven by the smartcard.

Figure 15-15. ISO7816-3 frame format


S
0
1
2
3
4
5
6
7
P
ISO 7816-3 frame without parity error
S
0
1
2
3
4
5
6
7
P
0.5 bit
1 bit
ISO 7816-3 frame with parity error

Character (T=0) mode

Compared to the timing in normal operation, the transmission time from transmit shift register to the TX pin is delayed by half baud clock, and the TC flag assertion time is delayed by a guard time that is configured by the GUAT[7:0] bits in USART_GP. In Smartcard mode, the internal guard time counter starts counting up after the stop bits of the last data frame, and the GUAT[7:0] bits should be configured as the character guard time (CGT) in ISO7816-3 protocol minus 12. The TC status is forced reset while the guard time counter is counting up. When the counter reaches the programmed value TC is asserted high.

During USART transmission, if a parity error event is detected, the smartcard may NACK the current frame by pulling down the TX pin during the last 1 bit time of the stop bits. The USART can automatically resend data according to the protocol for SCRTNUM times. An interframe gap of 2.5 bits time will be inserted before the start of a resented frame. At the end of the last repeated character the TC bit is set immediately without guard time. The USART will stop transmitting and assert the frame error status if it still receives the NACK signal after the programmed number of retries. The USART will not take the NACK signal as the start bit.

During USART reception, if the parity error is detected in the current frame, the TX pin is pulled low during the last 1 bit time of the stop bits. This signal is the NACK signal to smartcard. Then a frame error occurs in smartcard side. The RBNE/receive DMA request is not activated if the received character is erroneous. According to the protocol, the smartcard can resend the data. The USART stops transmitting the NACK and the error is regarded as a parity error if the received character is still erroneous after the maximum number of retries which is specified in the SCRTNUM bit field. The NACK signal is enabled by setting the NKEN bit in USART_CTL2.

The idle frame and break frame are not supported in the Smartcard mode.

Block (T=1) mode

In block (T=1) mode, the NKEN bit in the USART_CTL2 register should be cleared to deactivate the NACK transmission.

When requesting a read from the smartcard, the RT[23:0] bits in USART_RT register should be programmed with the BWT (block wait time) - 11 value and RBNEIE must be set. A timeout interrupt will be generated, if no answer is received from the card before the expiration of this period. If the first character is received before the expiration of the period, it is signaled by the RBNE interrupt. If DMA is used to read from the smartcard in block mode, the DMA must be enabled only after the first character is received.

In order to allow the automatic check of the maximum wait time between two consecutive characters, the USART_RT register must be programmed to the CWT (character wait time) - 11 value, which is expressed in baudtime units, after the reception of the first character (RBNE interrupt). The USART signals to the software through the RT flag and interrupt (when RTIE bit is set), if the smartcard doesn't send a new character in less than the CWT period after the end of the previous character.

The USART uses a block length counter, which is reset when the USART is transmitting (TBE=0), to count the number of received characters. The length of the block, which must be programmed in the BL[7:0] bits in the USART_RT register, is received from the smartcard in the third byte of the block (prologue field). This register field must be programmed to the minimum value (0x0), before the start of the block, when using DMA mode. With this value, an interrupt is generated after the 4th received character. The software must read the third byte as block length from the receive buffer.

In interrupt driven receive mode, the length of the block may be checked by software or by programming the BL value. However, before the start of the block, the maximum value of BL (0xFF) may be programmed. The real value will be programmed after the reception of the third character.

The total block length (including prologue, epilogue and information fields) equals BL+4. The end of the block is signaled to the software through the EBF flag and interrupt (when EBIE bit is set). The RT interrupt may occur in case of an error in the block length.

Direct and inverse convention

The smartcard protocol defines two conventions: direct and inverse.

The direct convention is defined as: LSB first, logical bit value of 1 corresponds to H state of the line and parity is even. In this case, the following control bits must be programmed: MSBF=0, DINV=0 (default values).

The inverse convention is defined as: MSB first, logical bit value 1 corresponds to an L state on the signal line and parity is even. In this case, the following control bits must be programmed: MSBF=1, DINV=1.

15.3.13. ModBus communication

The USART offers basic support for the implementation of ModBus / RTU and ModBus / ASCII protocols by implementing an end of block detection.

In the ModBus / RTU mode, the end of one block is recognized by an idle line for more than 2 characters time. This function is implemented through the programmable timeout function.

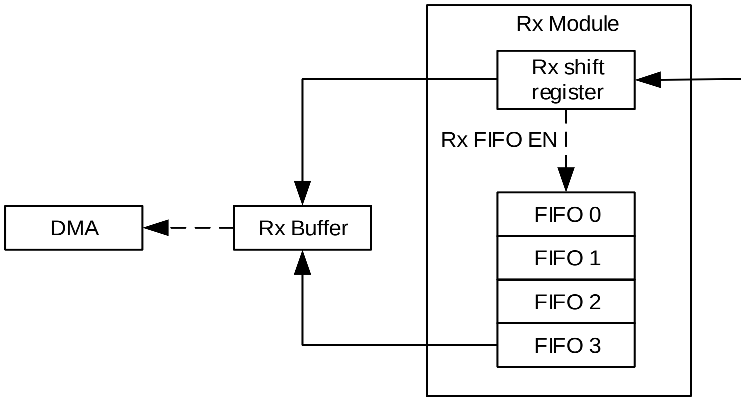
To detect the idle line, the RTEN bit in the USART_CTL1 register and the RTIE in the USART_CTL0 register must be set. The USART_RT register must be set to the value corresponding to a timeout of 2 characters time. After the last stop bit is received, when the receive line is idle for this duration, an interrupt will be generated, informing the software that the current block reception is completed.

In the ModBus / ASCII mode, the end of a block is recognized by a specific (CR / LF) character sequence. The USART manages this mechanism using the character match function by programming the LF ASCII code in the ADDR field and activating the address match interrupt (AMIE=1). When a LF has been received or can check the CR / LF in the DMA buffer, the software will be informed.

15.3.14. Receive FIFO

The receive FIFO can be enabled by setting the RFEN bit of the USART_RFCS register to avoid the overrun error when the CPU can't serve the RBNE interrupt immediately. Up to 5 frames receive data can be stored in the receive FIFO and receive buffer. The RFFINT flag will be set when the receive FIFO is full. An interrupt is generated if the RFFIE bit is set.

Figure 15-16. USART Receive FIFO structure



If the software read receive data buffer in the routing of the RBNE interrupt, the RBNEIE bit should be reset at the beginning of the routing and set after all of the receive data is read out. The PERR / NERR / FERR / EBF flags should be cleared before reading a receive data out.

15.3.15. Wakeup from Deep-sleep mode

The USART is able to wake up the MCU from Deep-sleep mode by the standard RBNE interrupt or the WUM interrupt.

The UESM bit must be set and the USART clock must be set to IRC48MDIV_PER or LXTAL (refer to the reset and clock unit RCU section).

When using the standard RBNE interrupt, the RBNEIE bit must be set before entering Deep-sleep mode.

When using the WUIE interrupt, the source of WUIE interrupt may be selected through the WUM bit fields.

DMA must be disabled before entering Deep-sleep mode. Before entering Deep-sleep mode, software must check that the USART is not performing a transfer, by checking the BSY flag in the USART_STAT register. The REA bit must be checked to ensure the USART is actually enabled.

When the wakeup event is detected, the WUF flag is set by hardware and a wakeup interrupt is generated if the WUIE bit is set, independently of whether the MCU is in stop or active mode.

15.3.16. USART interrupts

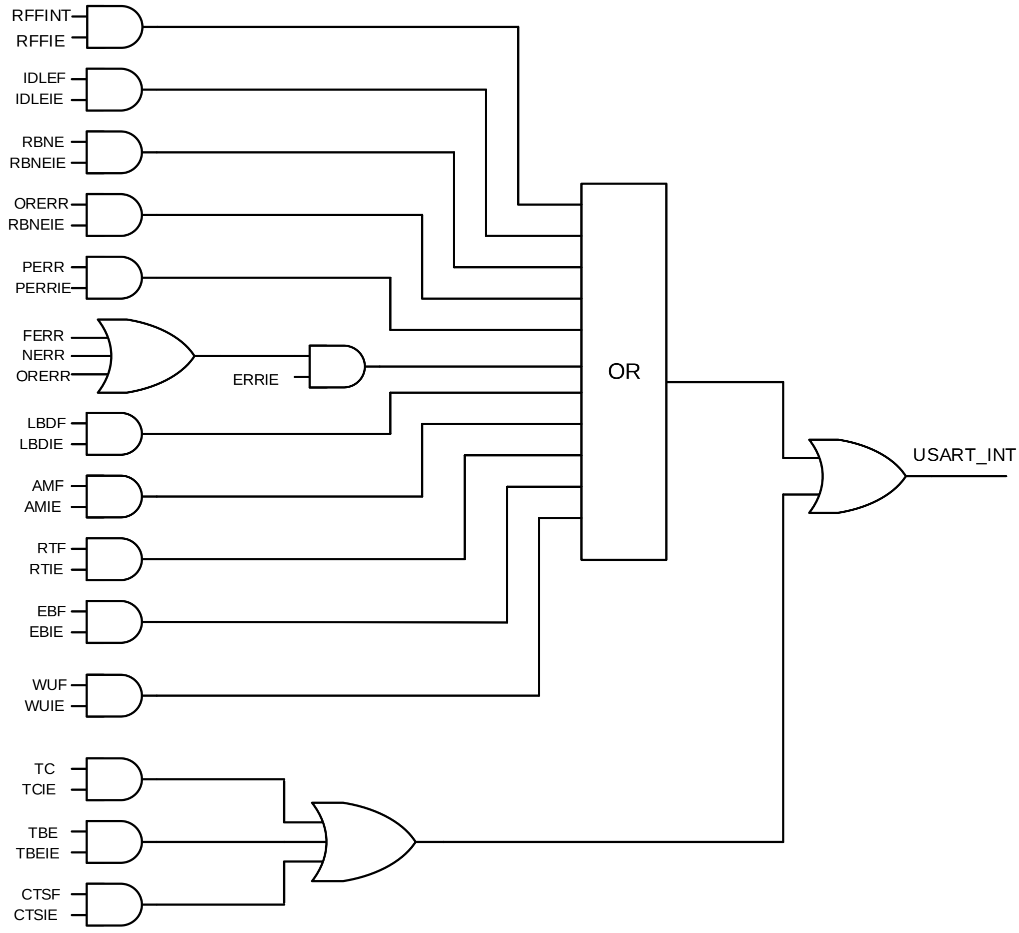
The USART interrupt events and flags are listed in ***Table 15-3. USART interrupt requests***.

Table 15-3. USART interrupt requests

Interrupt event	Event flag	Enable Control bit
Transmit data register empty	TBE	TBEIE
CTS flag	CTSF	CTSIE
Transmission complete	TC	TCIE
Received data ready to be read	RBNE	RBNEIE
Overrun error detected	ORERR	
Receive FIFO full	RFFINT	RFFIE
Idle line detected	IDLEF	IDLEIE
Parity error flag	PERR	PERRIE
Break detected flag in LIN mode	LBDF	LBDIE
Reception Errors (Noise flag, overrun error, framing error)	NERR or ORERR or FERR	ERRIE
Character match	AMF	AMIE
Receiver timeout error	RTF	RTIE
End of Block	EBF	EBIE
Wakeup from Deep-sleep mode	WUF	WUIE

All of the interrupt events are ORed together before being sent to the interrupt controller, so the USART can only generate a single interrupt request to the controller at any given time. Software can service multiple interrupt events in a single interrupt service routine

Figure 15-17. USART interrupt mapping diagram



15.4. Register definition

USART0 base address: 0x4001 3800

USART1 base address: 0x4000 4400

USART2 base address: 0x4000 4800

15.4.1. Control register 0 (USART_CTL0)

Address offset: 0x00

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved				EBIE	RTIE	DEA[4:0]					DED[4:0]				
				rw	rw	rw					rw				

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OVSMOD	AMIE	MEN	WL	WM	PCEN	PM	PERRIE	TBEIE	TCIE	RBNEIE	IDLEIE	TEN	REN	UESM	UEN
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31:28	Reserved	Must be kept at reset value
27	EBIE	End of Block interrupt enable 0: End of Block interrupt is disabled 1: End of Block interrupt is enabled This bit is reserved in USART1 and USART2.
26	RTIE	Receiver timeout interrupt enable 0: Receiver timeout interrupt is disabled 1: Receiver timeout interrupt is enabled This bit is reserved in USART1 and USART2.
25:21	DEA[4:0]	Driver Enable assertion time These bits are used to define the time between the activation of the DE (Driver Enable) signal and the beginning of the start bit. It is expressed in sample time units (1/8 or 1/16 bit time), which are configured by the OVSMOD bit. This bit field cannot be written when the USART is enabled (UEN=1).
20:16	DED[4:0]	Driver Enable de-assertion time These bits are used to define the time between the end of the last stop bit, in a transmitted message, and the de-activation of the DE (Driver Enable) signal. It is expressed in sample time units (1/8 or 1/16 bit time), which are configured by the OVSMOD bit. This bit field cannot be written when the USART is enabled (UEN=1).
15	OVSMOD	Oversample mode 0: Oversampling by 16

		1: Oversampling by 8 This bit must be kept cleared in LIN, IrDA and smartcard modes. This bit field cannot be written when the USART is enabled (UEN=1).
14	AMIE	ADDR match interrupt enable 0: ADDR match interrupt is disabled 1: ADDR match interrupt is enabled
13	MEN	Mute mode enable 0: Mute mode disabled 1: Mute mode enabled
12	WL	Word length 0: 8 Data bits 1: 9 Data bits This bit field cannot be written when the USART is enabled (UEN=1).
11	WM	Wakeup method in mute mode 0: Idle Line 1: Address Mark This bit field cannot be written when the USART is enabled (UEN=1).
10	PCEN	Parity control enable 0: Parity control disabled 1: Parity control enabled This bit field cannot be written when the USART is enabled (UEN=1).
9	PM	Parity mode 0: Even parity 1: Odd parity This bit field cannot be written when the USART is enabled (UEN=1).
8	PERRIE	Parity error interrupt enable 0: Parity error interrupt is disabled 1: An interrupt will occur whenever the PERR bit is set in USART_STAT.
7	TBEIE	Transmitter register empty interrupt enable 0: Interrupt is inhibited 1: An interrupt will occur whenever the TBE bit is set in USART_STAT
6	TCIE	Transmission complete interrupt enable If this bit is set, an interrupt occurs when the TC bit in USART_STAT is set. 0: Transmission complete interrupt is disabled 1: Transmission complete interrupt is enabled
5	RBNEIE	Read data buffer not empty interrupt and overrun error interrupt enable 0: Read data register not empty interrupt and overrun error interrupt disabled 1: An interrupt will occur whenever the ORERR bit is set or the RBNE bit is set in USART_STAT.

4	IDLEIE	IDLE line detected interrupt enable 0: IDLE line detected interrupt disabled 1: An interrupt will occur whenever the IDLEF bit is set in USART_STAT.
3	TEN	Transmitter enable 0: Transmitter is disabled 1: Transmitter is enabled
2	REN	Receiver enable 0: Receiver is disabled 1: Receiver is enabled and begins searching for a start bit
1	UESM	USART enable in Deep-sleep mode 0: USART not able to wake up the MCU from Deep-sleep mode. 1: USART able to wake up the MCU from Deep-sleep mode. Providing that the clock source for the USART must be IRC16M or LXTAL. This bit is reserved in USART1 and USART2.
0	UEN	USART enable 0: USART prescaler and outputs disabled 1: USART prescaler and outputs enabled

15.4.2. Control register 1 (USART_CTL1)

Address offset: 0x04

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ADDR[7:0]								RTEN	Reserved			MSBF	DINV	TINV	RINV
rw								rw				rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STRP	LMEN	STB[1:0]		CKEN	CPL	CPH	CLEN	Reserved	LBDIE	LBLEN	ADDM	Reserved			
rw	rw	rw		rw	rw	rw	rw		rw	rw	rw				

Bits	Fields	Descriptions
31:24	ADDR[7:0]	Address of the USART terminal These bits give the address of the USART terminal. In multiprocessor communication during mute mode or Deep-sleep mode, this is used for wakeup with address mark detection. The received frame, the MSB of which is equal to 1, will be compared to these bits. When the ADDM bit is reset, only the ADDR[3:0] bits are used to compare. In normal reception, these bits are also used for character detection. The whole received character (8-bit) is compared to the ADDR[7:0] value and AMF flag is set on matching. This bit field cannot be written when both reception (REN=1) and USART (UEN=1)

		are enabled.
23	RTEN	Receiver timeout enable 0: Receiver timeout function disabled 1: Receiver timeout function enabled This bit is reserved in USART1 and USART2.
22:20	Reserved	Must be kept at reset value.
19	MSBF	Most significant bit first 0: Data is transmitted / received with the LSB first 1: Data is transmitted / received with the MSB first This bit field cannot be written when the USART is enabled (UEN=1).
18	DINV	Data bit level inversion 0: Data bit signal values are not inverted 1: Data bit signal values are inverted This bit field cannot be written when the USART is enabled (UEN=1).
17	TINV	TX pin level inversion 0: TX pin signal values are not inverted 1: TX pin signal values are inverted This bit field cannot be written when the USART is enabled (UEN=1).
16	RINV	RX pin level inversion 0: RX pin signal values are not inverted 1: RX pin signal values are inverted This bit field cannot be written when the USART is enabled (UEN=1).
15	STRP	Swap TX / RX pins 0: The TX and RX pins functions are not swapped 1: The TX and RX pins functions are swapped This bit field cannot be written when the USART is enabled (UEN=1).
14	LMEN	LIN mode enable 0: LIN mode disabled 1: LIN mode enabled This bit field cannot be written when the USART is enabled (UEN=1). This bit is reserved in USART1 and USART2.
13:12	STB[1:0]	STOP bits length 00: 1 Stop bit 01: 0.5 Stop bits 10: 2 Stop bits 11: 1.5 Stop bits This bit field cannot be written when the USART is enabled (UEN=1).
11	CKEN	CK pin enable 0: CK pin disabled

		1: CK pin enabled This bit field cannot be written when the USART is enabled (UEN=1).
10	CPL	Clock polarity 0: Steady low value on CK pin outside transmission window in synchronous mode 1: Steady high value on CK pin outside transmission window in synchronous mode This bit field cannot be written when the USART is enabled (UEN=1).
9	CPH	Clock phase 0: The first clock transition is the first data capture edge in synchronous mode 1: The second clock transition is the first data capture edge in synchronous mode This bit field cannot be written when the USART is enabled (UEN=1).
8	CLEN	CK length 0: The clock pulse of the last data bit (MSB) is not output to the CK pin in synchronous mode 1: The clock pulse of the last data bit (MSB) is output to the CK pin in synchronous mode This bit field cannot be written when the USART is enabled (UEN=1)
7	Reserved	Must be kept at reset value.
6	LBDIE	LIN break detection interrupt enable 0: LIN break detection interrupt is disabled 1: An interrupt will occur whenever the LBDF bit is set in USART_STAT This bit is reserved in USART1 and USART2.
5	LBLEN	LIN break frame length 0: 10 bit break detection 1: 11 bit break detection This bit field cannot be written when the USART is enabled (UEN=1). This bit is reserved in USART1 and USART2.
4	ADDM	Address detection mode This bit is used to select between 4-bit address detection and full-bit address detection. 0: 4-bit address detection 1: full-bit address detection. In 7-bit, 8-bit and 9-bit data modes, the address detection is done on 6-bit, 7-bit and 8-bit address (ADDR[5:0], ADDR[6:0] and ADDR[7:0]) respectively. This bit field cannot be written when the USART is enabled (UEN=1).
3:0	Reserved	Must be kept at reset value.

15.4.3. Control register 2 (USART_CTL2)

Address offset: 0x08

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved									WUIE	WUM[1:0]		SCRTNUM[2:0]			Reserved
									rw	rw		rw			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DEP	DEM	DDRE	OVRD	OSB	CTSIE	CTSEN	RTSEN	DENT	DENR	SCEN	NKEN	HDEN	IRLP	IREN	ERRIE
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31:23	Reserved	Must be kept at reset value.
22	WUIE	Wakeup from Deep-sleep mode interrupt enable 0: Wakeup from Deep-sleep mode interrupt is disabled 1: Wakeup from Deep-sleep mode interrupt is enabled This bit is reserved in USART1 and USART2.
21:20	WUM[1:0]	Wakeup mode from Deep-sleep mode These bits are used to specify the event which activates the WUF (Wakeup from Deep-sleep mode flag) in the USART_STAT register. 00: WUF active on address match, which is defined by ADDR and ADDM 01: Reserved 10: WUF active on Start bit 11: WUF active on RBNE This bit field cannot be written when the USART is enabled (UEN=1). This bit is reserved in USART1 and USART2.
19:17	SCRTNUM[2:0]	Smartcard auto-retry number In smartcard mode, these bits specify the number of retries in transmission and reception. In transmission mode, a transmission error (FERR bit set) will occur after this number of automatic retransmission retries. In reception mode, reception error (RBNE and PERR bits set) will occur after this number or erroneous reception trials. When these bits are configured as 0x0, there will be no automatic retransmission in transmit mode. This bit field is only can be cleared to 0 when the USART is enabled (UEN=1), to stop retransmission. This bit is reserved in USART1 and USART2.
16	Reserved	Must be kept at reset value.
15	DEP	Driver enable polarity mode 0: DE signal is active high 1: DE signal is active low This bit field cannot be written when the USART is enabled (UEN=1)
14	DEM	Driver enable mode

		This bit is used to activate the external transceiver control, through the DE signal, which is output on the RTS pin. 0: DE function is disabled 1: DE function is enabled This bit field cannot be written when the USART is enabled (UEN=1).
13	DDRE	Disable DMA on reception error 0: DMA is not disabled in case of reception error. The DMA request is not asserted to make sure the erroneous data is not transferred, but the next correct received data will be transferred. The RBNE is kept 0 to prevent overrun, but the corresponding error flag is set. This mode can be used in Smartcard mode. 1: DMA is disabled following a reception error. The DMA request is not asserted until the error flag is cleared. The RBNE flag and corresponding error flag will be set. The software must first disable the DMA request (DMAR = 0) or clear RBNE before clearing the error flag. This bit field cannot be written when the USART is enabled (UEN=1).
12	OVRD	Overrun disable 0: Overrun functionality is enabled. The ORERR error flag will be set when received data is not read before receiving new data, and the new data will be lost. 1: Overrun functionality is disabled. The ORERR error flag will not be set when received data is not read before receiving new data, and the new received data overwrites the previous content of the USART_RDATA register. This bit field cannot be written when the USART is enabled (UEN=1).
11	OSB	One sample bit method 0: Three sample bit method 1: One sample bit method This bit field cannot be written when the USART is enabled (UEN=1).
10	CTSIE	CTS interrupt enable 0: CTS interrupt is disabled 1: An interrupt will occur whenever the CTS bit is set in USART_STAT
9	CTSEN	CTS enable 0: CTS hardware flow control disabled 1: CTS hardware flow control enabled This bit field cannot be written when the USART is enabled (UEN=1).
8	RTSEN	RTS enable 0: RTS hardware flow control disabled 1: RTS hardware flow control enablNed, data can be requested only when there is space in the receive buffer. This bit field cannot be written when the USART is enabled (UEN=1).
7	DENT	DMA enable for transmission 0: DMA mode is disabled for transmission

		1: DMA mode is enabled for transmission
6	DENR	DMA enable for reception 0: DMA mode is disabled for reception 1: DMA mode is enabled for reception
5	SCEN	Smartcard mode enable 0: Smartcard Mode disabled 1: Smartcard Mode enabled This bit field cannot be written when the USART is enabled (UEN=1). This bit is reserved in USART1 and USART2.
4	NKEN	NACK enable in Smartcard mode 0: Disable NACK transmission when parity error 1: Enable NACK transmission when parity error This bit field cannot be written when the USART is enabled (UEN=1). This bit is reserved in USART1 and USART2.
3	HDEN	Half-duplex enable 0: Half duplex mode is disabled 1: Half duplex mode is enabled This bit field cannot be written when the USART is enabled (UEN=1).
2	IRLP	IrDA low-power 0: Normal mode 1: Low-power mode This bit field cannot be written when the USART is enabled (UEN=1).
1	IREN	IrDA mode enable 0: IrDA disabled 1: IrDA enabled This bit field cannot be written when the USART is enabled (UEN=1). This bit is reserved in USART1 and USART2.
0	ERRIE	Error interrupt enable 0: Error interrupt disabled 1: An interrupt will occur whenever the FERR bit or the ORERR bit or the NERR bit is set in USART_STAT in multibuffer communication.

15.4.4. Baud rate generator register (USART_BAUD)

Address offset: 0x0C

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

This register cannot be written when the USART is enabled (UEN=1).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BRR [15:4]												BRR[3:0]			
rw												rw			

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:4	BRR[15:4]	Integer of baud-rate divider DIV_INT[11:0] = BRR[15:4]
3:0	BRR [3:0]	Fraction of baud-rate divider If OVSMOD = 0, USARTDIV [3:0] = BRR [3:0]; If OVSMOD = 1, USARTDIV [3:1] = BRR [2:0], BRR [3] must be reset.

15.4.5. Prescaler and guard time configuration register (USART_GP)

Address offset: 0x10

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

This register cannot be written when the USART is enabled (UEN=1).

This register is reserved in USART1, USART2.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GUAT[7:0]								PSC[7:0]							
rw								rw							

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:8	GUAT[7:0]	Guard time value in smartcard mode This bit field cannot be written when the USART is enabled (UEN=1).
7:0	PSC[7:0]	Prescaler value for dividing the system clock In IrDA Low-power mode, the division factor is the prescaler value. 00000000: Reserved - do not program this value 00000001: divides the source clock by 1 00000010: divides the source clock by 2 ... In IrDA normal mode, 00000001: can be set this value only In smartcard mode, the prescaler value for dividing the system clock is stored in PSC[4:0] bits. And the bits of PSC[7:5] must be kept at reset value. The division

factor is twice as the prescaler value.

00000: Reserved - do not program this value

00001: divides the source clock by 2

00010: divides the source clock by 4

00011: divides the source clock by 6

...

This bit field cannot be written when the USART is enabled (UEN=1).

15.4.6. Receiver timeout register (USART_RT)

Address offset: 0x14

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

This bit is reserved in USART1 and USART2.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BL[7:0]								RT[23:16]							
rw								rw							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RT[15:0]															
rw															

Bits	Fields	Descriptions
31:24	BL[7:0]	Block Length These bits specify the block length in smartcard T=1 Reception. Its value equals the number of information characters + the length of the Epilogue Field (1-LEC/2-CRC) - 1. This value, which must be programmed only once per received block, can be programmed after the start of the block reception (using the data from the LEN character in the Prologue Field). The block length counter is reset when TBE=0 in smartcard mode. In other modes, when REN=0 (receiver disabled) and / or when the EBC bit is written to 1, the Block length counter is reset.
23:0	RT[23:0]	Receiver timeout threshold These bits are used to specify receiver timeout value in terms of number of baud clocks. In standard mode, the RTF flag is set if no new start bit is detected for more than the RT value after the last received character. In smartcard mode, the CWT and BWT are implemented by this value. In this case, the timeout measurement is started from the start bit of the last received character. These bits can be written on the fly. The RTF flag will be set if the new value is lower than or equal to the counter. These bits must only be programmed once per received character.

15.4.7. Command register (USART_CMD)

Address offset: 0x18

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved											TXFCMD	RXFCMD	MMCMD	SBKCMD	Reserved
											w	w	w	w	

Bits	Fields	Descriptions
31:5	Reserved	Must be kept at reset value.
4	TXFCMD	Transmit data flush request Writing 1 to this bit sets the TBE flag, to discard the transmit data. This bit is reserved in USART1 and USART2.
3	RXFCMD	Receive data flush command Writing 1 to this bit clears the RBNE flag to discard the received data without reading it.
2	MMCMD	Mute mode command Writing 1 to this bit makes the USART into mute mode and sets the RWU flag.
1	SBKCMD	Send break command Writing 1 to this bit sets the SBKF flag and makes the USART send a BREAK frame, as soon as the transmit machine is idle.
0	Reserved	Must be kept at reset value.

15.4.8. Status register (USART_STAT)

Address offset: 0x1C

Reset value: 0x0000 00C0

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved									REA	TEA	WUF	RWU	SBF	AMF	BSY
									r	r	r	r	r	r	r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved			EBF	RTF	CTS	CTSF	LBDF	TBE	TC	RBNE	IDLEF	ORERR	NERR	FERR	PERR
			r	r	r	r	r	r	r	r	r	r	r	r	r

Bits	Fields	Descriptions

31:23	Reserved	Must be kept at reset value.
22	REA	Receive enable acknowledge flag This bit, which is set / reset by hardware, reflects the receive enable state of the USART core logic. 0: The USART core receiving logic has not been enabled 1: The USART core receiving logic has been enabled
21	TEA	Transmit enable acknowledge flag This bit, which is set / reset by hardware, reflects the transmit enable state of the USART core logic. 0: The USART core transmitting logic has not been enabled 1: The USART core transmitting logic has been enabled
20	WUF	Wakeup from Deep-sleep mode flag 0: No wakeup from Deep-sleep mode 1: Wakeup from Deep-sleep mode. An interrupt is generated if WUFIE=1 in the USART_CTL2 register and the MCU is in Deep-sleep mode. This bit is set by hardware when a wakeup event, which is defined by the WUM bit field, is detected. Cleared by writing a 1 to the WUC in the USART_INTC register. This bit can also be cleared when UESM is cleared. This bit is reserved in USART1 and USART2.
19	RWU	Receiver wakeup from mute mode This bit is used to indicate if the USART is in mute mode. 0: Receiver in active mode 1: Receiver in mute mode It is cleared / set by hardware when a wakeup / mute sequence (address or IDLEIE) is recognized, which is selected by the WAKE bit in the USART_CTL0 register. This bit can only be set by writing 1 to the MMCMD bit in the USART_CMD register when wakeup on IDLEIE mode is selected.
18	SBF	Send break flag 0: No break character is transmitted 1: Break character will be transmitted This bit indicates that a send break character was requested. Set by software, by writing 1 to the SBKCMD bit in the USART_CMD register. Cleared by hardware during the stop bit of break transmission.
17	AMF	ADDR match flag 0: ADDR does not match the received character 1: ADDR matches the received character, An interrupt is generated if AMIE=1 in the USART_CTL0 register. Set by hardware, when the character defined by ADDR [7:0] is received. Cleared by writing 1 to the AMC in the USART_INTC register.
16	BSY	Busy flag

		0: USART reception path is idle 1: USART reception path is working
15:13	Reserved	Must be kept at reset value
12	EBF	End of block flag 0: End of Block not reached 1: End of Block (number of characters) reached. An interrupt is generated if the EBIE=1 in the USART_CTL1 register. Set by hardware when the number of received bytes (from the start of the block, including the prologue) is equal or greater than BLEN + 4. Cleared by writing 1 to EBC bit in USART_INTC register. This bit is reserved in USART1 and USART2.
11	RTF	Receiver timeout flag 0: Timeout value not reached 1: Timeout value reached without any data reception. An interrupt is generated if RTIE bit in the USART_CTL1 register is set. Set by hardware when the RT value, programmed in the USART_RT register has lapsed without any communication. Cleared by writing 1 to RTC bit in USART_INTC register. The timeout corresponds to the CWT or BWT timings in smartcard mode. This bit is reserved in USART1 and USART2
10	CTS	CTS level This bit equals to the inverted level of the nCTS input pin. 0: nCTS input pin is in high level 1: nCTS input pin is in low level
9	CTSF	CTS change flag 0: No change occurred on the nCTS status line 1: A change occurred on the nCTS status line. An interrupt will occur if the CTSIE bit is set in USART_CTL2 Set by hardware when the nCTS input toggles. Cleared by writing 1 to CTSC bit in USART_INTC register.
8	LBDF	LIN break detected flag 0: LIN Break is not detected 1: LIN Break is detected. An interrupt will occur if the LBDIE bit is set in USART_CTL1. Set by hardware when the LIN break is detected. Cleared by writing 1 to LBDC bit in USART_INTC register. This bit is reserved in USART1 and USART2.
7	TBE	Transmit data register empty 0: Data is not transferred to the shift register 1: Data is transferred to the shift register. An interrupt will occur if the TBEIE bit is set in USART_CTL0.

		Set by hardware when the content of the USART_TDATA register has been transferred into the transmit shift register or writing 1 to TXFCMD bit of the USART_CMD register. Cleared by a write to the USART_TDATA.
6	TC	Transmission completed 0: Transmission is not completed 1: Transmission is complete. An interrupt will occur if the TCIE bit is set in USART_CTL0. Set by hardware if the transmission of a frame containing data is completed and if the TBE bit is set. Cleared by writing 1 to TCC bit in USART_INTC register.
5	RBNE	Read data buffer not empty 0: Data is not received 1: Data is received and ready to be read. An interrupt will occur if the RBNEIE bit is set in USART_CTL0. Set by hardware when the content of the receive shift register has been transferred to the USART_RDATA. Cleared by reading the USART_RDATA or writing 1 to RXFCMD bit of the USART_CMD register.
4	IDLEF	IDLE line detected flag 0: No Idle Line is detected 1: Idle Line is detected. An interrupt will occur if the IDLEIE bit is set in USART_CTL0. Set by hardware when an Idle Line is detected. It will not be set again until the RBNE bit has been set itself. Cleared by writing 1 to IDLEC bit in USART_INTC register.
3	ORERR	Overrun error 0: No Overrun error is detected 1: Overrun error is detected. An interrupt will occur if the RBNEIE bit is set in USART_CTL0. In multibuffer communication, an interrupt will occur if the ERRIE bit is set in USART_CTL2. Set by hardware when the word in the receive shift register is ready to be transferred into the USART_RDATA register while the RBNE bit is set. Cleared by writing 1 to OREC bit in USART_INTC register.
2	NERR	Noise error flag 0: No noise error is detected 1: Noise error is detected. In multibuffer communication, an interrupt will occur if the ERRIE bit is set in USART_CTL2. Set by hardware when noise error is detected on a received frame. Cleared by writing 1 to NEC bit in USART_INTC register.
1	FERR	Frame error flag

Bits	Fields	Descriptions
		0: No framing error is detected 1: Frame error flag or break character is detected. In multibuffer communication, an interrupt will occur if the ERRIE bit is set in USART_CTL2. Set by hardware when a de-synchronization, excessive noise or a break character is detected. This bit will be set when the maximum number of transmit attempts is reached without success (the card NACKs the data frame), when USART transmits in smartcard mode. Cleared by writing 1 to FEC bit in USART_INTC register.
0	PERR	Parity error flag 0: No parity error is detected 1: Parity error flag is detected. An interrupt will occur if the PERRIE bit is set in USART_CTL0. Set by hardware when a parity error occurs in receiver mode. Cleared by writing 1 to PEC bit in USART_INTC register.

15.4.9. Interrupt status clear register (USART_INTC)

Address offset: 0x20

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved											WUC	Reserved		AMC	Reserved
											w			w	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved			EBC	RTC	Reserved	CTSC	LBDC	Reserved	TCC	Reserved	IDLEC	OREC	NEC	FEC	PEC
			w	w		w	w		w		w	w	w	w	w

Bits	Fields	Descriptions
31:21	Reserved	Must be kept at reset value.
20	WUC	Wakeup from Deep-sleep mode clear Writing 1 to this bit clears the WUF bit in the USART_STAT register. This bit is reserved in USART1 and USART2.
19:18	Reserved	Must be kept at reset value.
17	AMC	ADDR match clear Writing 1 to this bit clears the AMF bit in the USART_STAT register.
16:13	Reserved	Must be kept at reset value.
12	EBC	End of block clear Writing 1 to this bit clears the EBF bit in the USART_STAT register. This bit is reserved in USART1 and USART2.
11	RTC	Receiver timeout clear

		Writing 1 to this bit clears the RTF flag in the USART_STAT register. This bit is reserved in USART1 and USART2.
10	Reserved	Must be kept at reset value.
9	CTSC	CTS change clear Writing 1 to this bit clears the CTSF bit in the USART_STAT register.
8	LBDC	LIN break detected clear Writing 1 to this bit clears the LBDF flag in the USART_STAT register. This bit is reserved in USART1 and USART2.
7	Reserved	Must be kept at reset value.
6	TCC	Transmission complete clear Writing 1 to this bit clears the TC bit in the USART_STAT register.
5	Reserved	Must be kept at reset value.
4	IDLEC	Idle line detected clear Writing 1 to this bit clears the IDLEF bit in the USART_STAT register.
3	OREC	Overrun error clear Writing 1 to this bit clears the ORERR bit in the USART_STAT register.
2	NEC	Noise detected clear Writing 1 to this bit clears the NERR bit in the USART_STAT register.
1	FEC	Frame error flag clear Writing 1 to this bit clears the FERR bit in the USART_STAT register
0	PEC	Parity error clear Writing 1 to this bit clears the PERR bit in the USART_STAT register.

15.4.10. Receive data register (USART_RDATA)

Address offset: 0x24

Reset value: Undefined

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved							RDATA[8:0]								
							r								

Bits	Fields	Descriptions
31:9	Reserved	Must be kept at reset value.

Bits	Fields	Descriptions
8:0	RDATA[8:0]	Receive Data value The received data character is contained in these bits. The value read in the MSB (bit 7 or bit 8 depending on the data length) will be the received parity bit, if receiving with the parity is enabled (PCEN bit set to 1 in the USART_CTL0 register).

15.4.11. Transmit data register (USART_TDATA)

Address offset: 0x28

Reset value: Undefined

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved							TDATA[8:0]								
							rw								

Bits	Fields	Descriptions
31:9	Reserved	Must be kept at reset value.
8:0	TDATA[8:0]	Transmit Data value The transmit data character is contained in these bits. The value written in the MSB (bit 7 or bit 8 depending on the data length) will be replaced by the parity, when transmitting with the parity is enabled (PCEN bit set to 1 in the USART_CTL0 register). This register must be written only when TBE bit in USART_STAT register is set.

15.4.12. USART coherence control register (USART_CHC)

Address offset: 0xC0

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved							EPERR	Reserved							HCM
							rc_w0								rw

Bits	Fields	Descriptions
31:9	Reserved	Must be kept at reset value.

8	EPERR	Early parity error flag. This flag will be set as soon as the parity bit has been detected, which is before RBNE flag. This flag is cleared by writing 0. 0: No parity error is detected 1: Parity error is detected.
7:1	Reserved	Must be kept at reset value.
0	HCM	Hardware flow control coherence mode 0: nRTS signal equals to the RBNE in status register 1: nRTS signal is set when the last data bit (parity bit when pce is set) has been sampled.

15.4.13. USART receive FIFO control and status register (USART_RFCS)

Address offset: 0xD0

Reset value: 0x0000 0400

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RFFINT	RFCNT[2:0]			RFF	RFE	RFFIE	RFEN	Reserved							ELNACK
r_w0	r			r	r	rw	rw								rw

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15	RFFINT	Receive FIFO full interrupt flag
14:12	RFCNT[2:0]	Receive FIFO counter number
11	RFF	Receive FIFO full flag 0: Receive FIFO not full 1: Receive FIFO full
10	RFE	Receive FIFO empty flag 0: Receive FIFO not empty 1: Receive FIFO empty
9	RFFIE	Receive FIFO full interrupt enable 0: Receive FIFO full interrupt disable 1: Receive FIFO full interrupt enable
8	RFEN	Receive FIFO enable This bit can be set when UESM = 1. 0: Receive FIFO disable 1: Receive FIFO enable

7:1	Reserved	Must be kept at reset value.
0	ELNACK	Early NACK when smartcard mode is selected. The NACK pulse occurs 1/16 bit time earlier when the parity error is detected. 0:Early NACKdisable when smartcard mode is selected 1:Early NACKenable when smartcard mode is selected This bit is reserved in USART1 and USART2.

16. Inter-integrated circuit interface (I2C)

16.1. Overview

The I2C (inter-integrated circuit) module provides an I2C interface which is an industry standard two-line serial interface for MCU to communicate with external I2C interface. I2C bus uses two serial lines: a serial data line, SDA, and a serial clock line, SCL.

The I2C interface module implements the standard mode, fast mode, and fast mode plus of the I2C protocol. It features CRC calculation and verification functions and supports SMBus (System Management Bus) and PMBus (Power Management Bus). It also supports multi-master I2C bus. The I2C interface provides DMA mode for users to reduce CPU overload.

16.2. Characteristics

- Parallel-bus to I2C-bus protocol converter and interface.
- Both master and slave functions with the same interface.
- Bi-directional data transfer between master and slave.
- Supports 7-bit and 10-bit addressing and general call addressing.
- Multiple 7-bit slave addresses (2 addresses with configurable mask).
- Programmable setup time and hold time.
- Multi-master capability.
- Supports standard mode (up to 100 kHz) and fast mode (up to 400 kHz).
- The I2C0 supports fast mode plus (up to 1 MHz, this mode must be enabled in SYSCFG_CFG0).
- Configurable SCL stretching in slave mode.
- Supports DMA mode.
- I2C0 is compatible with SMBus 3.0 and PMBus 1.3.
- Optional PEC (packet error checking) generation and check.
- Programmable analog and digital noise filters.
- Wakeup from Deep-sleep mode and Deep-sleep 1 mode on I2C address match.
- Independent clock from PCLK.

16.3. Function overview

Figure 16-1. I2C module block diagram below provides details on the internal configuration of the I2C interface.

Figure 16-1. I2C module block diagram

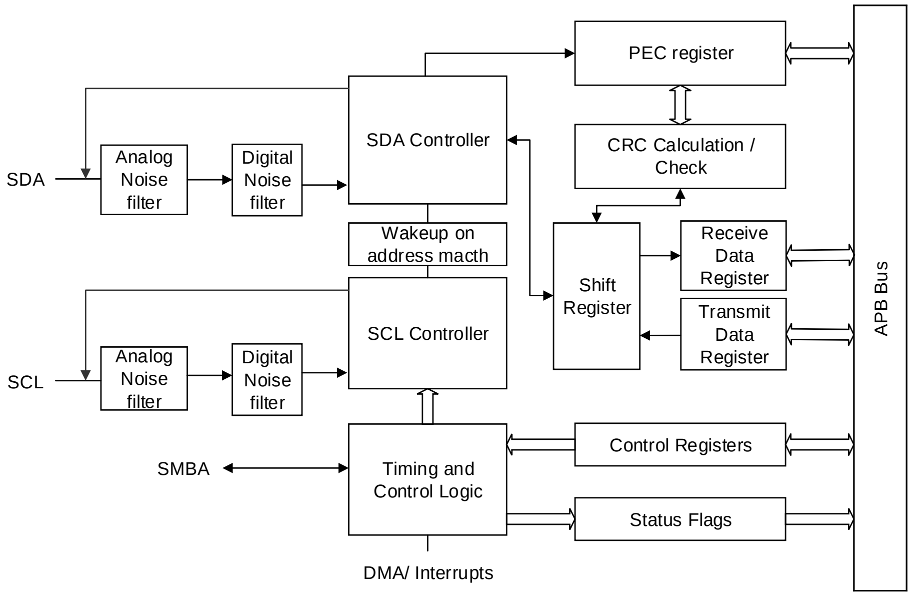


Table 16-1. Definition of I2C-bus terminology (refer to the I2C specification of Philips semiconductors)

Term	Description
Transmitter	the device which sends data to the bus
Receiver	the device which receives data from the bus
Master	the device which initiates a transfer, generates clock signals and terminates a transfer
Slave	the device addressed by a master
Multi-master	more than one master can attempt to control the bus at the same time without corrupting the message
Arbitration	procedure to ensure that, if more than one master tries to control the bus simultaneously, only one is allowed to do so and the winning master's message is not corrupted

16.3.1. Clock requirements

The I2C clock is independent of the PCLK frequency, so that the I2C can be operated independently.

This I2C clock (I2CCLK) can be selected from the following three clock sources:

- CK_APB: APB clock (default value)
- CK_IRC48MDIV_PER: CK_IRC48M / 3 (reset value)
- CK_SYS: system clock

The I2CCLK period t_{I2CCLK} must match the conditions as follows:

- $t_{I2CCLK}<(t_{LOW}-t_{filters})/4$
- $t_{I2CCLK}<t_{HIGH}$

with:

t_{LOW}: SCL low time

t_{HIGH}: SCL high time

$t_{filters}$: When the filters are enabled, represent the delays by the analog filter and digital filter.

Analog filter delay is maximum 260ns. Digital filter delay is DNF[3:0]×t_{I2CCLK}.

The period of PCLK clock t_{PCLK} match the conditions as follows:

- $t_{PCLK}<4/3*t_{SCL}$

with:

t_{SCL}: the period of SCL

Note: When the I2C kernel is provided by PCLK, this clock must match the conditions for t_{I2CCLK}.

16.3.2. I2C communication flow

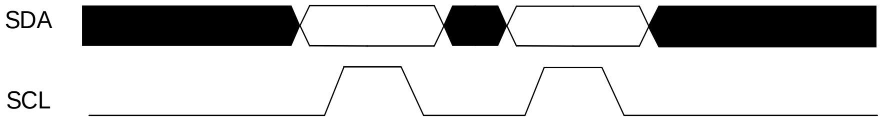
An I2C device is able to transmit or receive data whether it's a master or a slave, thus, there're 4 operation modes for an I2C device:

- Slave transmitter
- Slave receiver
- Master transmitter
- Master receiver

Data validation

The data on the SDA line must be stable during the HIGH period of the clock. The HIGH or LOW state of the data line can only change when the clock signal on the SCL line is LOW (see ***Figure 16-2. Data validation***). One clock pulse is generated for each data bit transferred.

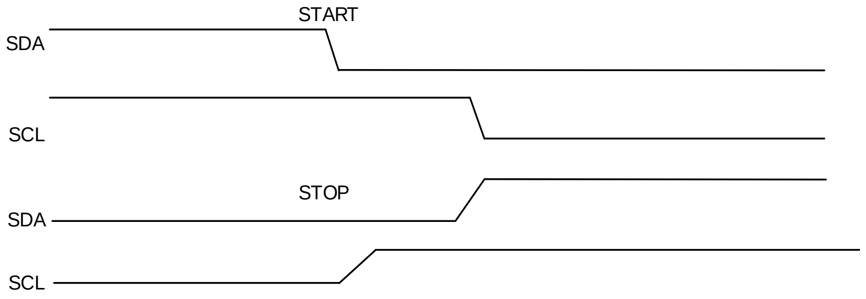
Figure 16-2. Data validation



START and STOP signal

All transactions begin with a START and are terminated by a STOP (see ***Figure 16-3. START and STOP signal***). A HIGH to LOW transition on the SDA line while SCL is HIGH defines a START signal. A LOW to HIGH transition on the SDA line while SCL is HIGH defines a STOP signal.

Figure 16-3. START and STOP signal



Each I2C device is recognized by a unique address (whether it is a microcontroller, LCD driver, memory or keyboard interface) and can operate as either a transmitter or receiver, depending on the function of the device. It operates in slave mode by default. When it generates a START signal, the interface automatically switches from slave to master. If an arbitration loss or a STOP generation occurs, then the interface switches from master to slave, allowing multimaster capability.

An I2C slave will continue to detect addresses after a START signalon I2C bus and compare the detected address with its slave address which is programmable by software. Once the two addresses match, the I2C slave will send an ACK to the I2C bus and responses to the following command on I2C bus: transmitting or receiving the desired data. Additionally, if General Call is enabled by software, the I2C slave always responses to a General Call Address (0x00). The I2C block support both 7-bit and 10-bit address modes.

Data and addresses are transferred as 8-bit bytes, MSB first. The first byte(s) following the START signalcontain the address (one in 7-bit mode, two in 10-bit mode). The address is always transmitted in master mode.

A 9th clock pulse follows the 8 clock cycles of byte transmission, during which the receiver must send an acknowledge bit to the transmitter. Acknowledge can be enabled or disabled by software.

An I2C master always initiates or end a transfer using START or STOP signaland it’s also responsible for SCL clock generation.

In master mode, if AUTOEND = 1, the STOP signalis generated automatically by hardware. If AUTOEND = 0, the STOP signalgenerated by software, or the master can generate a RESTART signalto start a new transfer.

Figure 16-4. I2C communication flow with 10-bit address (Master Transmit)

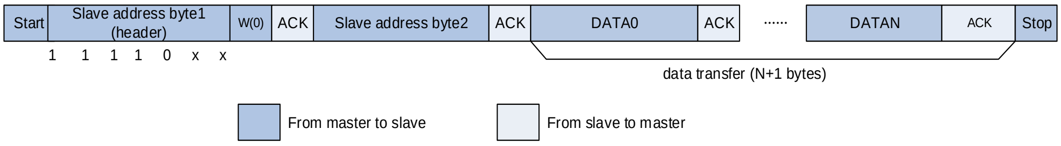


Figure 16-5. I2C communication flow with 7-bit address (Master Transmit)

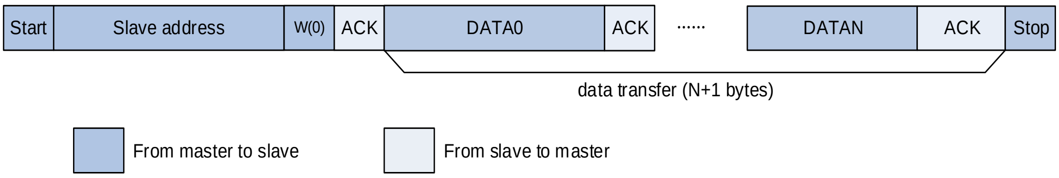
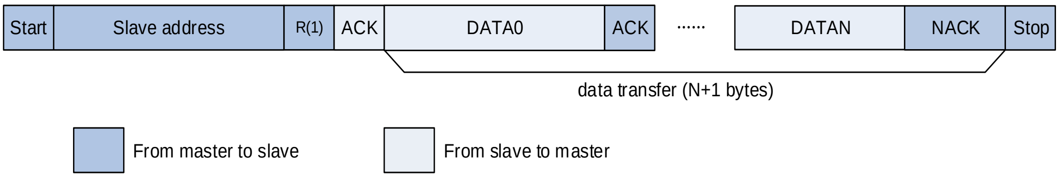


Figure 16-6. I2C communication flow with 7-bit address (Master Receive)



In 10-bit addressing mode, the HEAD10R bit can configured to decide whether the complete address sequence must be executed, or only the header to be sent. When HEAD10R = 0, the complete 10 bit address read sequence must be excuted with START + header of 10-bit address in write direction + slave address byte 2 + RESTART + header of 10-bit address in read direction, as is shown in *Figure 16-7. I2C communication flow with 10-bit address (Master Receive when HEAD10R=0)*.

In 10-bit addressing mode, if the master reception follows a master transmission between the same master and slave, the address read sequence can be RESTART + header of 10-bit address in read direction, as is shown in *Figure 16-8. I2C communication flow with 10-bit address (Master Receive when HEAD10R=1)*.

Figure 16-7. I2C communication flow with 10-bit address (Master Receive when HEAD10R=0)

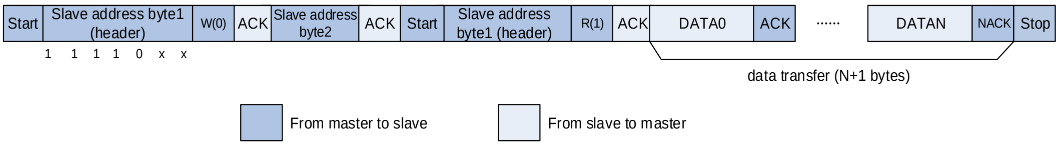
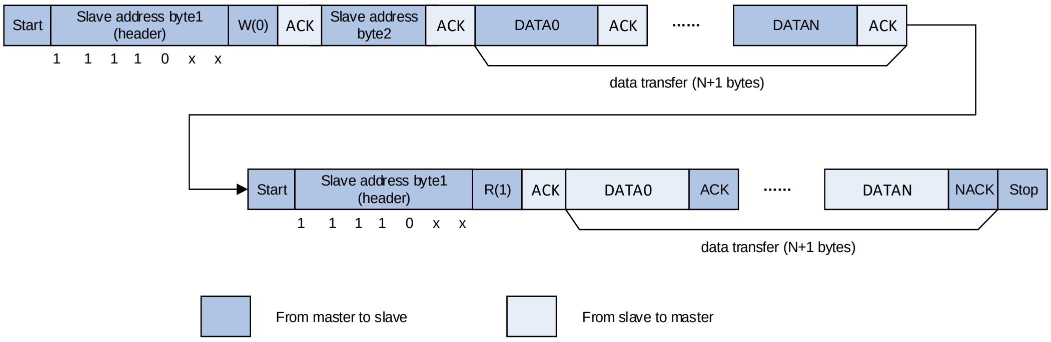


Figure 16-8. I2C communication flow with 10-bit address (Master Receive when HEAD10R=1)



16.3.3. Noise filter

Analog noise filter and digital noise filter are integrated in I2C peripherals, the noise filters can be configured before the I2C peripheral is enabled according to the actual requirements.

The analog noise filter is disabled by setting the ANOFF bit in I2C_CTL0 register and enabled when ANOFF is 0. It can suppress spikes with a pulse width up to 50ns in fast mode and fast mode plus mode.

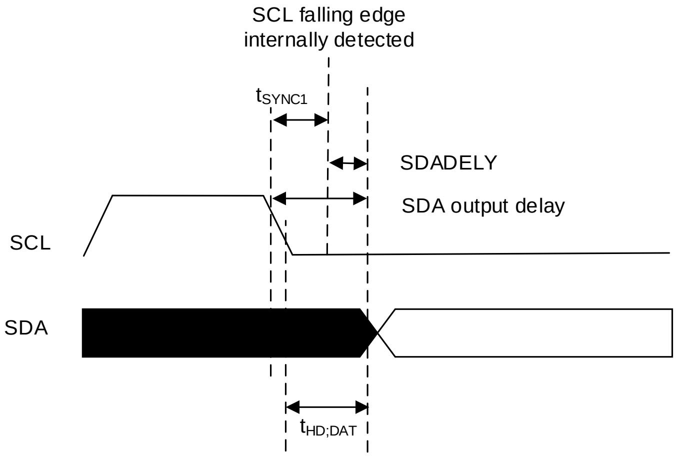
The digital noise filter can be used by configuring the DNF[3:0] bit in I2C_CTL0 register. The level of the SCL or the SDA will be changed if the level is stable for more than DNF[3:0]×t_{I2CCLK}. The length of spikes to be suppressed is configured by DNF[3:0].

16.3.4. I2C timings configuration

The PSC[3:0], SCLDELY[3:0] and SDADELY[3:0] bits in the I2C_TIMING register must be configured in order to guarantee a correct data hold and setup time used in I2C communication.

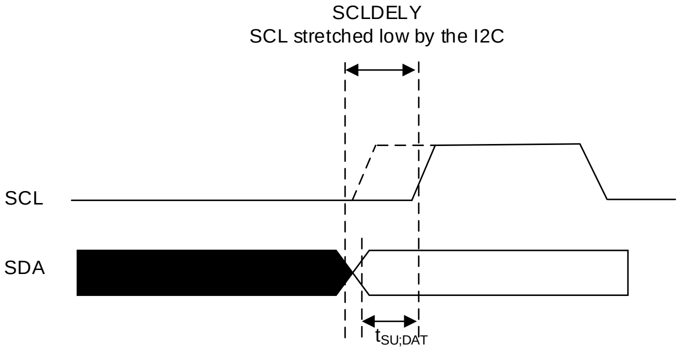
If the data is already available in I2C_TDATA register, the data will be sent on SDA after the SDADELY delay. As is shown in ***Figure 16-9. Data hold time***.

Figure 16-9. Data hold time



The SCLDELY counter starts when the data is sent on SDA output. As is shown in ***Figure 16-10. Data setup time***.

Figure 16-10. Data setup time



When the SCL falling edge is internally detected, a delay is inserted before sending SDA output. This delay is $t_{SDADELY}$=SDADELY*t_{PSC}+t_{I2CCLK} where t_{PSC}=(PSC+1)*t_{I2CCLK}. $t_{SDADELY}$ effects $t_{HD;DAT}$. The total delay of SDA output is t_{SYNC1}+{[SDADELY*(PSC+1)+1]*t_{I2CCLK}}. t_{SYNC1} depends on SCL falling slope, the delay of analog filter, the delay of digital filter and delay of SCL synchronization to I2CCLK clock. The delay of SCL synchronization to I2CCLK clock is 2 to 3 t_{I2CCLK}.

SDADELY must match condition as follows:

- SDADELY≥{t_f(max)+$t_{HD;DAT}$(min)-t_{AF}(min)-[(DNF+3)*t_{I2CCLK}]}/[(PSC+1)*t_{I2CCLK}]
- SDADELY≤{$t_{HD;DAT}$(max)-t_{AF}(max)-[(DNF+4)*t_{I2CCLK}]}/[(PSC+1)*t_{I2CCLK}]

Note: t_{AF} is the delay of analog filter. The $t_{HD;DAT}$ should be less than the maximum of $t_{VD;DAT}$.

When SS = 0, after $t_{SDADELY}$ delay, the slave had to stretch the clock before the data writing to I2C_TDATA register, SCL is low during the data setup time. The setup time is $t_{SCLDELY}$=(SCLDELY+1)*t_{PSC}. $t_{SCLDELY}$ effects $t_{SU;DAT}$.

SCLDELY must match condition as follows:

- SCLDELY≥{[t_r(max)+$t_{SU;DAT}$(min)]/[(PSC+1)*t_{I2CCLK}]}-1

In master mode, the SCL clock high and low levels must be configured by programming the PSC[3:0], SCLH[7:0] and SCLL[7:0] bits in the I2C_TIMING register.

When the SCL falling edge is internally detected, a delay is inserted before releasing the SCL output. This delay is t_{SCLL}=(SCLL+1)*t_{PSC} where t_{PSC}=(PSC+1)*t_{I2CCLK}. t_{SCLL} impacts the SCL low time t_{LOW}.

When the SCL rising edge is internally detected, a delay is inserted before forcing the SCL output to low level. This delay is t_{SCLH}=(SCLH+1)*t_{PSC} where t_{PSC}=(PSC+1)*t_{I2CCLK}. t_{SCLH} impacts the SCL high time t_{HIGH}.

Note: When the I2C is enabled, the timing configuration and SS mode must not be changed.

Table 16-2. Data setup time and data hold time

Symbol	Parameter	Standard mode		Fast mode		Fast mode plus		SMBus		Unit
		Min	Max	Min	Max	Min	Max	Min	Max	
$t_{HD;DAT}$	Data hold time	0	-	0	-	0	-	0.3	-	us
$t_{VD;DAT}$	Data valid time	-	3.45	-	0.9	-	0.45	-	-	
$t_{SU;DAT}$	Data setup time	250	-	100	-	50	-	250	-	ns
t_r	Rising time of SCL and SDA	-	1000	-	300	-	120	-	1000	
t_f	falling time of SCL and SDA	-	300	-	300	-	120	-	300	

16.3.5. I2C reset

A software reset can be performed by clearing the I2CEN bit in the I2C_CTL0 register. When a software reset is generated, the SCL and SDA are released. The communication control bits and status bits come back to the reset value. Software reset have no effect on configuration registers. The impacted register bits are START, STOP, NACKEN in I2C_CTL1 register, I2CBSY, TBE, TI, RBNE, ADDSEND, NACK, TCR, TC, STPDET, BERR, LOSTARB and OUERR in I2C_STAT register. Additionally, when the SMBus is supported, PECTRANS in I2C_CTL1 register, PECERR, TIMEOUT and SMBALT in I2C_STAT are also impacted.

In order to perform the software reset, I2CEN must be kept low during at least 3 APB clock cycles. This is ensured by writing software sequence as follows:

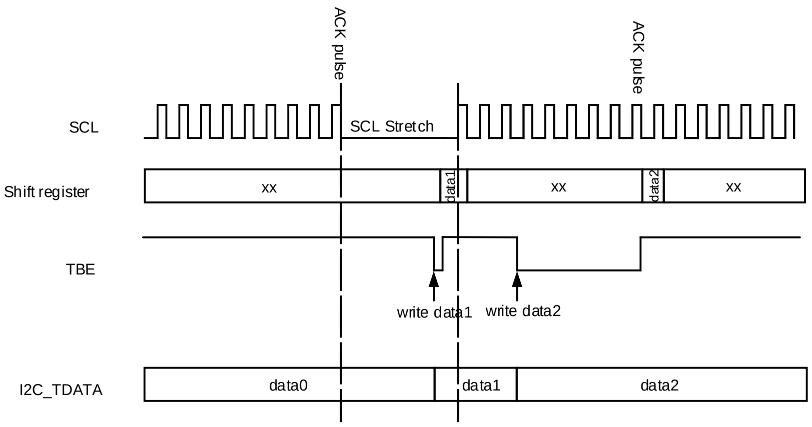
- Write I2CEN = 0
- Check I2CEN = 0
- Write I2CEN = 1

16.3.6. Data transfer

Data Transmission

When transmitting data, if TBE is 0, it indicates that the I2C_TDATA register is not empty, the data in I2C_TDATA register is moved to the shift register after the 9th SCL pulse. Then the data will be transmitted through the SDA line from the shift register. If TBE is 1, it indicates that the I2C_TDATA register is empty, the SCL line is stretched low until I2C_TDATA is not empty. The stretch begins after the 9th SCL pulse.

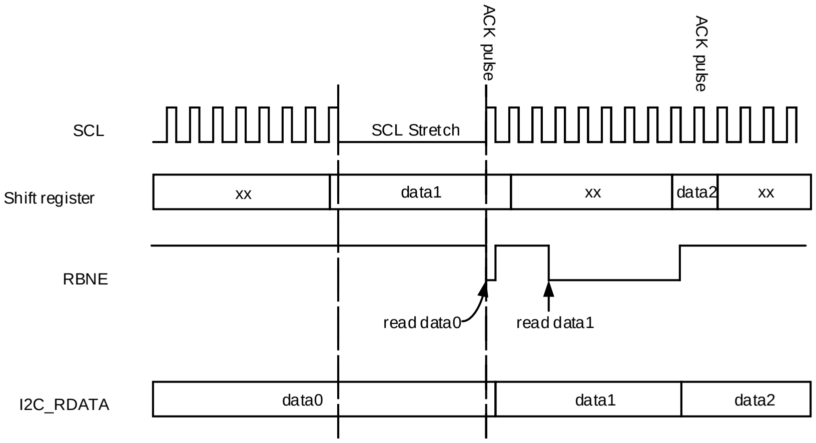
Figure 16-11. Data transmission



Data Reception

When receiving data, the data will be received in the shift register first. If RBNE is 0, the data in the shift register will move into I2C_RDATA register. If RBNE is 1, the SCL line wii be stretched until the previous received data in I2C_RDATA is read. The stretch is inserted before the acknowledge pulse.

Figure 16-12. Data reception



Reload and automatic end mode

In order to manage byte transfer and to shut down the communication in modes as is shown in ***Table 16-3. Communication modes to be shut down***, the I2C embedded a byte counter in the hardware.

Table 16-3. Communication modes to be shut down

Working mode	Action
Master mode	NACK, STOP and RESTART generation
Slave receiver mode	ACK control

Working mode	Action
SMBus mode	PEC generation/checking

The number of bytes to be transferred is configured by BYTENUM[7:0] in I2C_CTL1 register. If BYTENUM is greater than 255, or in slave byte control mode, the reload mode must be enabled by setting the RELOAD bit in I2C_CTL1 register. In reload mode, when BYTENUM counts to 0, the TCR bit will be set, and an interrupt will be generated if TCIE is set. Once the TCR flag is set, SCL is stretched. The TCR bit is cleared by writing a non-zero number in BYTENUM.

Note: The reload mode must be disabled after the last reloading of BYTENUM[7:0].

The reload mode must be disabled when the automatic end mode is enabled. In automatic end mode, the master will send a STOP signal automatically when the BYTENUM[7:0] counts to 0.

When reload mode and automatic end mode are disabled, the I2C communication process needs to be terminated by software. If the number of bytes in BYTENUM[7:0] has been transferred, the STOP bit should be set by software to generate a STOP signal, and then TC flag must be cleared.

16.3.7. I2C slave mode

Initialization

When works in slave mode, at least one slave address should be enabled. Slave address 1 can be programmed in I2C_SADDR0 register and slave address 2 can be programmed in I2C_SADDR1 register. ADDRESSEN in I2C_SADDR0 register and ADDRESS2EN in I2C_SADDR1 register should be set when the corresponding address is used. 7-bit address or 10-bit address can be programmed in ADDRESS[9:0] in I2C_SADDR0 register by configuring the ADDFORMAT bit in 7-bit address or 10-bit address.

The ADDM[6:0] in I2C_CTL2 register defines which bits of ADDRESS[7:1] are compared with an incoming address byte, and which bits are ignored.

The ADDMSK2[2:0] is used to mask ADDRESS2[7:1] in I2C_SADDR1 register. For details, refer to the description of ADDMSK2[2:0] in I2C_SADDR1 register.

When the I2C received address matches one of its enabled addresses, the ADDSEND will be set, and an interrupt is generated if the ADDMIE bit is set. The READDR[6:0] bits in I2C_STAT register will store the received address. And TR bit in I2C_STAT register updates after the ADDSEND is set. The bit will let the slave to know whether to act as a transmitter or receiver.

SCL line stretching

The clock stretching is used in slave mode by default (SS = 0), the SCL line can be stretched low if necessary. The SCL will be stretched in following cases.

- The SCL is stretched when the ADDSEND bit is set, and released when the ADDSEND bit is cleared.

- In slave transmitting mode, after the ADDSEND bit is cleared, the SCL will be stretched before the first data byte writing to the I2C_TDATA register. Or the SCL will be stretched before the new data is written to the I2C_TDATA register after the previous data transmission is completed.
- In slave receiving mode, a new reception is completed but the data in I2C_RDATA register has not been read.
- When SBCTL = 1 and RELOAD = 1, after the transfer of the last byte, TCR is set. Before the TCR is cleared, the SCL will be stretched.
- The I2C stretches SCL low during [(SDADELY+SCLDELY+1)*(PSC+1)+1]*t_{I2CCLK} after detecting the SCL falling edge.

The clock stretching can be disabled by setting the SS bit in I2C_CTL0 register (SS = 1). The SCL will not be stretched in following cases.

- When the ADDSEND is set, the SCL will be not stretched.
- In slave transmitting mode, before the first SCL pulse, the data must be written in the I2C_TDATA register . Or else the OUERR bit in the I2C_STAT register will be set, if the ERRIE bit is set, an interrupt will be generated. When the STPDET bit is set and the first data transmission starts, OUERR bit in the I2C_STAT register will also be set.
- In slave receiving mode, before the 9th SCL pulse (ACK pulse) occurred by the next data byte, the data must be read out from the I2C_RDATA register. Or else the OUERR bit in the I2C_STAT register will be set, if the ERRIE bit is set, and an interrupt will be generated.

Slave byte control mode

In slave receiving mode, the slave byte control mode can be enabled by setting the SBCTL bit in the I2C_CTL0 register to allow byte ACK control. When SS = 1, the slave byte control mode is not allowed.

When using slave byte control mode, the reload mode must be enabled by setting the RELOAD bit in I2C_CTL1 register. In slave byte control mode, BYTENUM[7:0] in I2C_CTL1 register must be configured as 1 in the ADDSEND interrupt service routine and reloaded to 1 after each byte received. The TCR bit in I2C_STAT register will be set when a byte is received, the SCL will be stretched low by slave between the 8th and 9th clock pulses. Then the data can be read from the I2C_RDATA register, and the slave determines to send an ACK or a NACK by configuring the NACKEN bit in the I2C_CTL1 register. When the BYTENUM[7:0] is written a non-zero value, the slave will release the stretch.

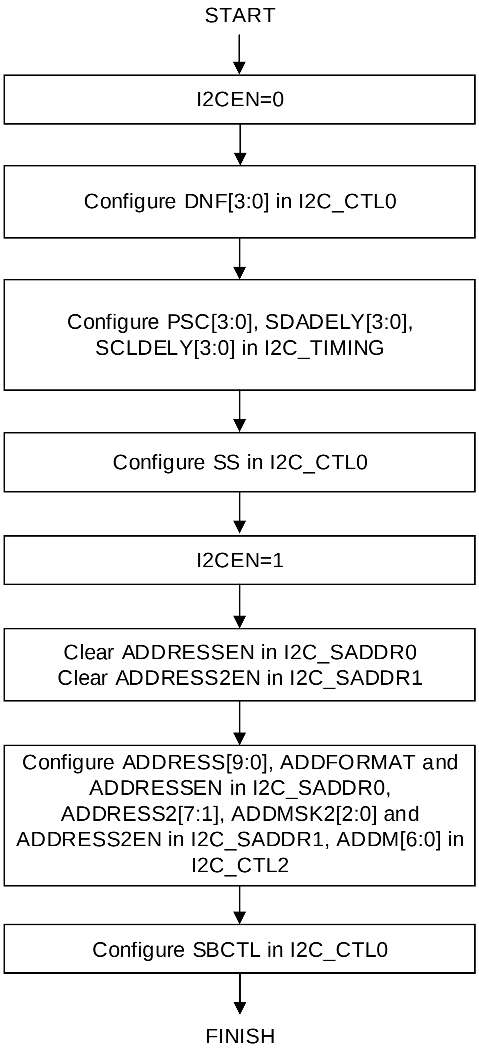
When the BYTENUM[7:0] is greater than 0x1, there is no stretch between the reception of two data bytes.

Note: The SBCTL bit can be configured in following cases:

1. I2CEN = 0.
2. The slave has not been addressed.
3. ADDSEND = 1.

Only when the ADDSEND = 1 or TCR = 1, the RELOAD bit can be modified.

Figure 16-13. I2C initialization in slave mode


START
I2CEN=0
Configure DNF[3:0] in I2C_CTL0
Configure PSC[3:0], SDADELY[3:0], SCLDELY[3:0] in I2C_TIMING
Configure SS in I2C_CTL0
I2CEN=1
Clear ADDRESSEN in I2C_SADDR0
Clear ADDRESS2EN in I2C_SADDR1
Configure ADDRESS[9:0], ADDFORMAT and ADDRESSEN in I2C_SADDR0, ADDRESS2[7:1], ADDMSK2[2:0] and ADDRESS2EN in I2C_SADDR1, ADDM[6:0] in I2C_CTL2
Configure SBCTL in I2C_CTL0
FINISH

Programming model in slave transmitting mode

When the I2C_TDATA register is empty, the TI bit in I2C_STAT register will be set. If the TIE bit in I2C_CTL0 register is set, an interrupt will be generated. The NACK bit in I2C_STAT register will be set when a NACK is received. And an interrupt is generated if the NACKIE bit is set in the I2C_CTL0 register. The TI bit in I2C_STAT register will not be set when a NACK is received.

The STPDET bit in I2C_STAT register will be set when a STOP is received. If the STPDETIE in I2C_CTL0 register is set, an interrupt will be generated.

When SBCTL is 0, if ADDSEND = 1, and the TBE bit in I2C_STAT register is 0, the data in I2C_TDATA register can be chosed to be transmitted or flushed. The data is flushed by setting the TBE bit.

When SBCTL = 1, the slave works in slave byte control mode, the BYTENUM[7:0] must be configured in the ADDSEND interrupt service routine. And the number of TI events is equal to the value of BYTENUM[7:0].

When SS = 1, the SCL will not be stretched when ADDSEND bit in I2C_STAT register is set. In this case, the data in I2C_TDATA register can not be flushed in ADDSEND interrupt service routine. So the first byte to be sent must be programmed in the I2C_TDATA register previously.

- This data can be the one which is written in the last TI event of the last transfer.
- Setting the TBE bit can flush the data if it is not the one to be sent, then a new byte can be written in I2C_TDATA register. The STPDET must be 0 when the data transmission begins. Or else the OUERR bit in I2C_STAT register will be set and an underrun error occurs.
- When interrupt or DMA is used in slave transmitter, if a TI event is needed, in order to generate a TI event both the TI bit and the TBE bit must be set.

Figure 16-14. Programming model for slave transmitting when SS=0

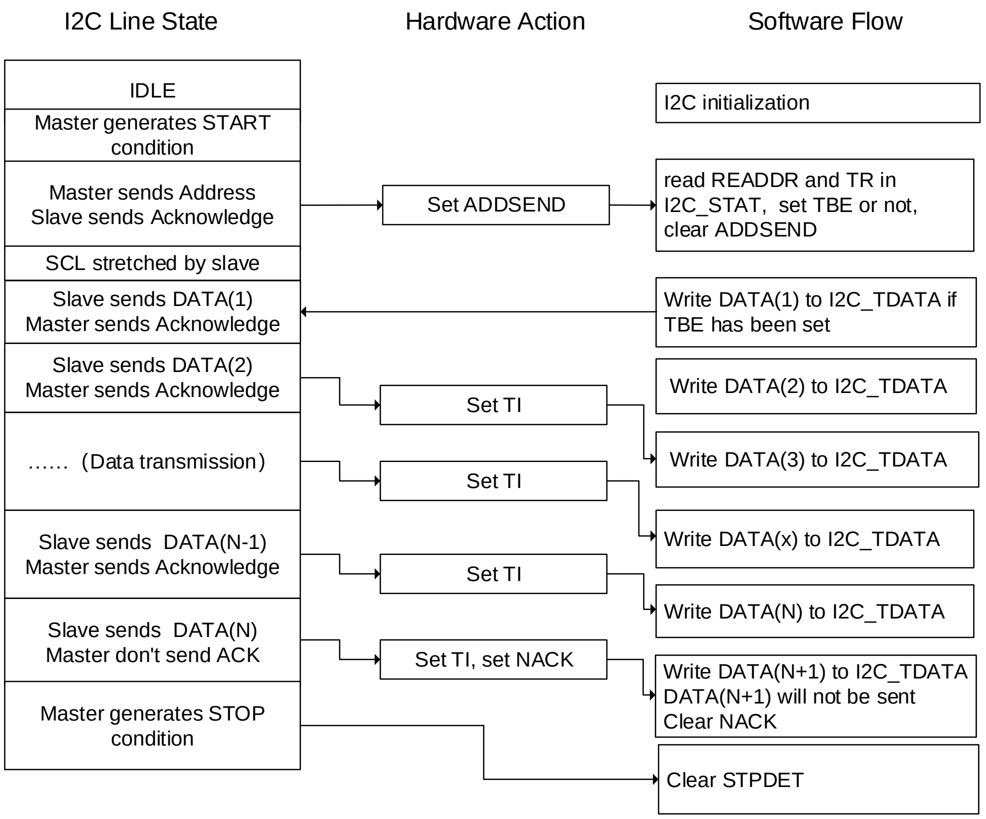
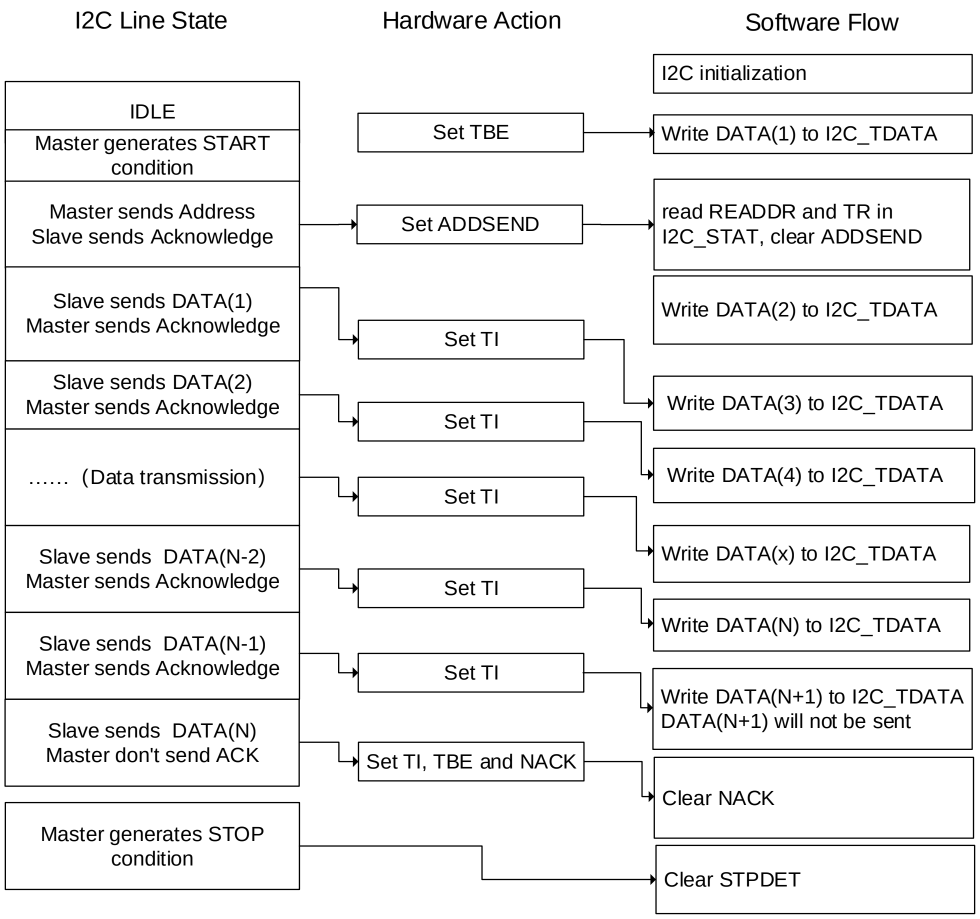

I2C Line State
Hardware Action
Software Flow
IDLE
Master generates START condition
Master sends Address Slave sends Acknowledge
SCL stretched by slave
Slave sends DATA(1) Master sends Acknowledge
Slave sends DATA(2) Master sends Acknowledge
…… (Data transmission)
Slave sends DATA(N-1) Master sends Acknowledge
Slave sends DATA(N) Master don't send ACK
Master generates STOP condition
Set ADDSEND
Set TI
Set TI
Set TI
Set TI, set NACK
I2C initialization
read READDR and TR in I2C_STAT, set TBE or not, clear ADDSEND
Write DATA(1) to I2C_TDATA if TBE has been set
Write DATA(2) to I2C_TDATA
Write DATA(3) to I2C_TDATA
Write DATA(x) to I2C_TDATA
Write DATA(N) to I2C_TDATA
Write DATA(N+1) to I2C_TDATA DATA(N+1) will not be sent Clear NACK
Clear STPDET

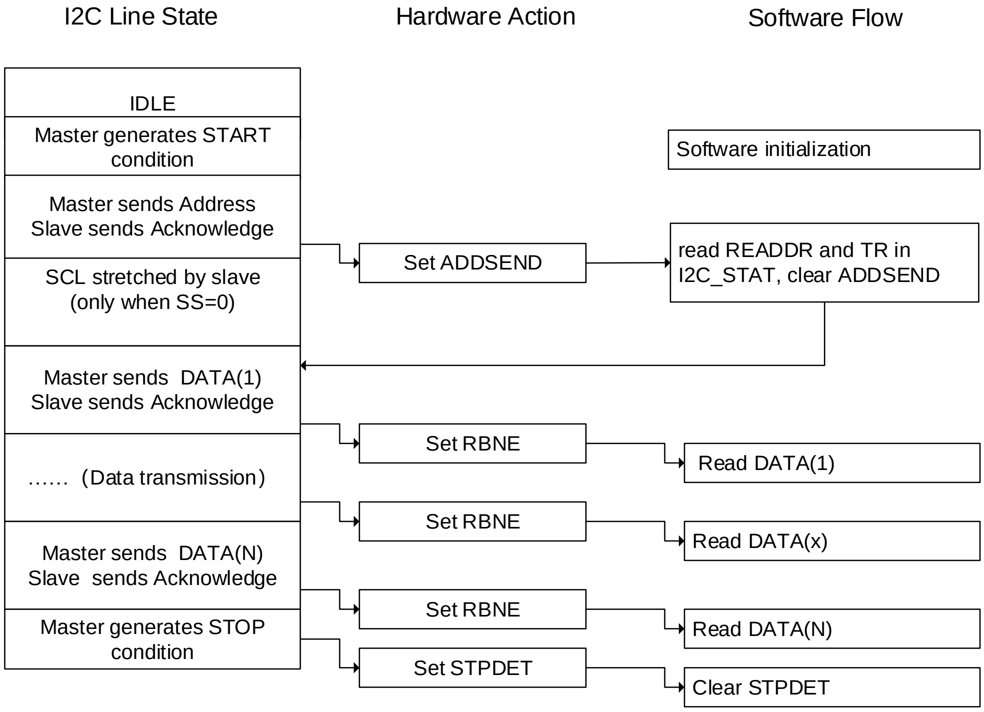
Figure 16-15. Programming model for slave transmitting when SS=1



Programming model in slave receiving mode

When the I2C_RDATA is not empty, the RBNE bit in I2C_STAT register is set, and if the RBNEIE bit in I2C_CTL0 register is set, an interrupt will be generated. When a STOP is received, STPDET will be set in I2C_STAT register. If the STPDETIE bit in I2C_CTL0 register is set, and an interrupt will be generated.

Figure 16-16. Programming model for slave receiving


I2C Line State
Hardware Action
Software Flow
IDLE
Master generates START condition
Master sends Address Slave sends Acknowledge
SCL stretched by slave (only when SS=0)
Master sends DATA(1) Slave sends Acknowledge
...... (Data transmission)
Master sends DATA(N) Slave sends Acknowledge
Master generates STOP condition
Set ADDSEND
Set RBNE
Set RBNE
Set RBNE
Set STPDET
Software initialization
read READDR and TR in I2C_STAT, clear ADDSEND
Read DATA(1)
Read DATA(x)
Read DATA(N)
Clear STPDET

16.3.8. I2C master mode

Initialization

The SCLH[7:0] and SCLL[7:0] in I2C_TIMING register should be configured when I2CEN is 0. In order to support multi-master communication and slave clock stretching, a clock synchronization mechanism is implemented.

The SCLL[7:0] and SCLH[7:0] are used for the low level counting and high level couting respectively. After a t_{SYNC1} delay, when the SCL low level is detected, the SCLL[7:0] starts counting, if the SCLL[7:0] in I2C_TIMING register is reached by SCLL[7:0] counter, the I2C will release the SCL clock. After a t_{SYNC2} delay, when the SCL high level is detected, the SCLH[7:0] starts counting, if the SCLH[7:0] in I2C_TIMING register is reached by SCLH[7:0] counter, the I2C will stretch the SCL clock.

So the master clock period is:

$t_{SCL}=t_{SYNC1}+t_{SYNC2}+\{[(SCLH[7:0]+1)+(SLLL[7:0]+1)]*(PSC+1)*t_{I2CCLK}\}$.

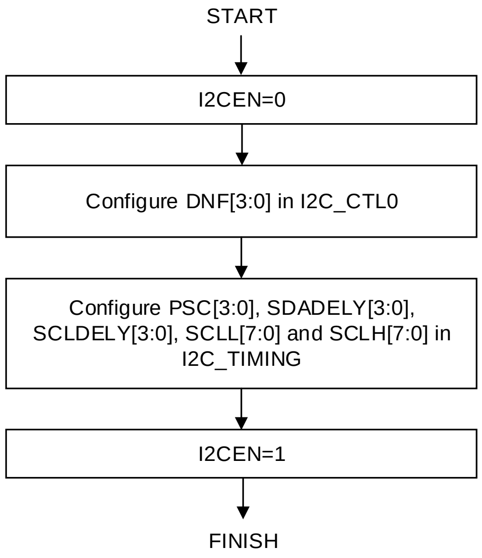
The t_{SYNC1} depends on the SCL falling slope, delay by input analog and digital noise filter and SCL synchronization with I2CCLK clock, which generally 2 to 3 I2CCLK periods. The t_{SYNC2} depends on the SCL rising slope, delay by input analog and digital noise filter and SCL synchronization with I2CCLK clock, which generally 2 to 3 I2CCLK periods. The delay by digital noise filter is DNF[3:0]*t_{I2CCLK}.

When works in master mode, the ADD10EN bit, SADDRESS[9:0] bits, TRDIR bit should be configured in I2C_CTL1 register. When the addressing mode is 10-bit in master receiving mode, the HEAD10R bit must be configured to decide whether the complete address sequence must be executed, or only the header to be sent. The number of bytes to be transferred should be configured in BYTENUM[7:0] in I2C_CTL1 register. If the number of bytes to be transferred is equal to or greater than 255, BYTENUM[7:0] should be configured as 0xFF. Then the master sends the START signal. All the bits above should be configured before the START is set. The slave address will be sent after the START signal when the I2CBSY bit I2C_STAT register is detected as 0. When the arbitration is lost, the master changes to slave mode and the START bit will be cleared by hardware. When the slave address has been sent, the START bit will be cleared by hardware.

In 10-bit addressing mode, if the master receives a NACK after the transmission of 10-bit header, the master will resend it until ACK is received. The ADDSENDC bit must be set to stop sending the slave address.

If the START bit is set, meanwhile the ADDSEND is set by addressing as a slave, the master changes to slave mode. The ADDSENDC bit must be set to clear the START bit.

Figure 16-17. I2C initialization in master mode


START
I2CEN=0
Configure DNF[3:0] in I2C_CTL0
Configure PSC[3:0], SDADELY[3:0], SCLDELY[3:0], SCLL[7:0] and SCLH[7:0] in I2C_TIMING
I2CEN=1
FINISH

Programming model in master transmitting mode

In master transmitting mode, the TI bit is set after the ACK is received of each byte transmission. If the TIE bit in I2C_CTL0 register is set, an interrupt will be generated. The bytes to be transferred is programmed in BYTENUM[7:0] in I2C_CTL0 register. If the bytes to be transferred is greater than 255, RELOAD bit in I2C_CTL0 register must be set to enable the reload mode. In reload mode, when data of BYTENUM[7:0] bytes have been transferred, the TCR bit in I2C_STAT register will be set and the SCL stretches unitil BYTENUM[7:0] is modified with a non-zero value.

When a NACK is received, the TI bit will not set.

- If data of BYTENUM[7:0] bytes have been transferred and RELOAD = 0, the AUTOEND bit in I2C_CTL1 can be set to generate a STOP signal automatically. When AUTOEND is 0, the TC bit in I2C_STAT register will be set and the SCL is stretched. In this case, the master can generate a STOP signal by setting the STOP bit in the I2C_CTL1 register. Or generate a RESTART signal to start a new transfer. The TC bit is cleared when the START / STOP bit is set.
- If a NACK is received, a STOP signal is automatically generated, the NACK is set in I2C_STAT register, if the NACKIE bit is set, an interrupt will be generated.

Note: When the RELOAD bit is 1, the AUTOEND has no effect.

Figure 16-18. Programming model for master transmitting (N<=255)

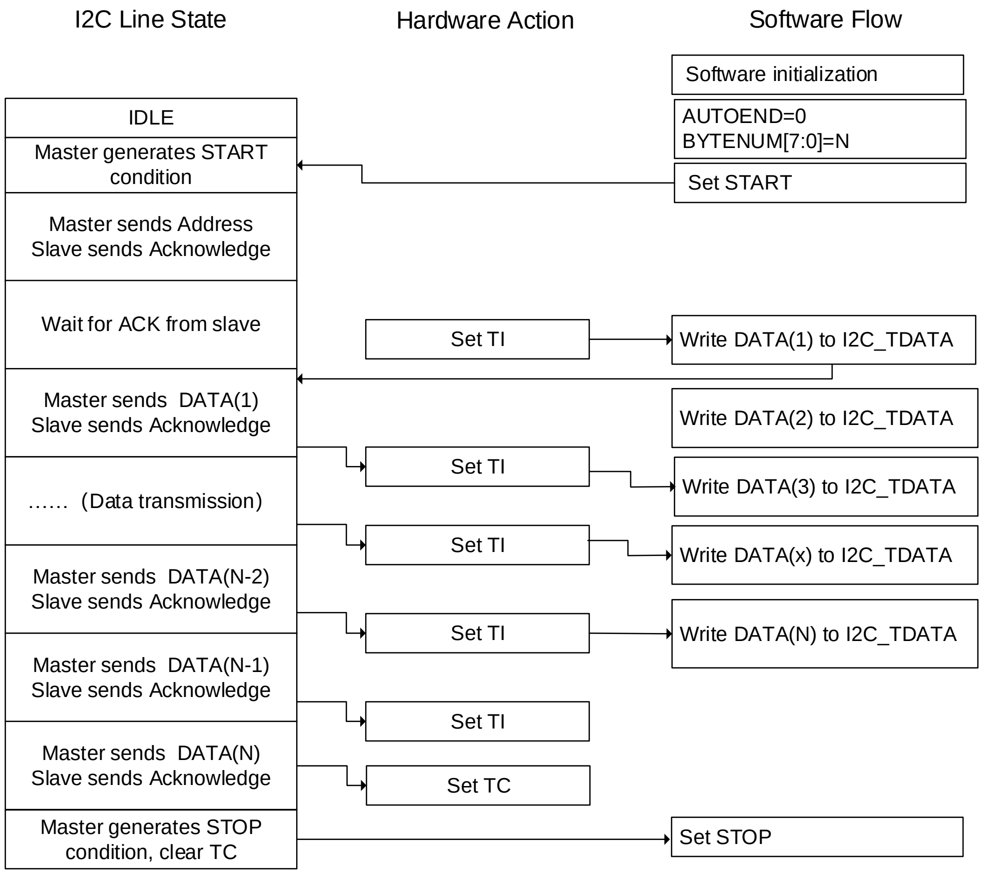
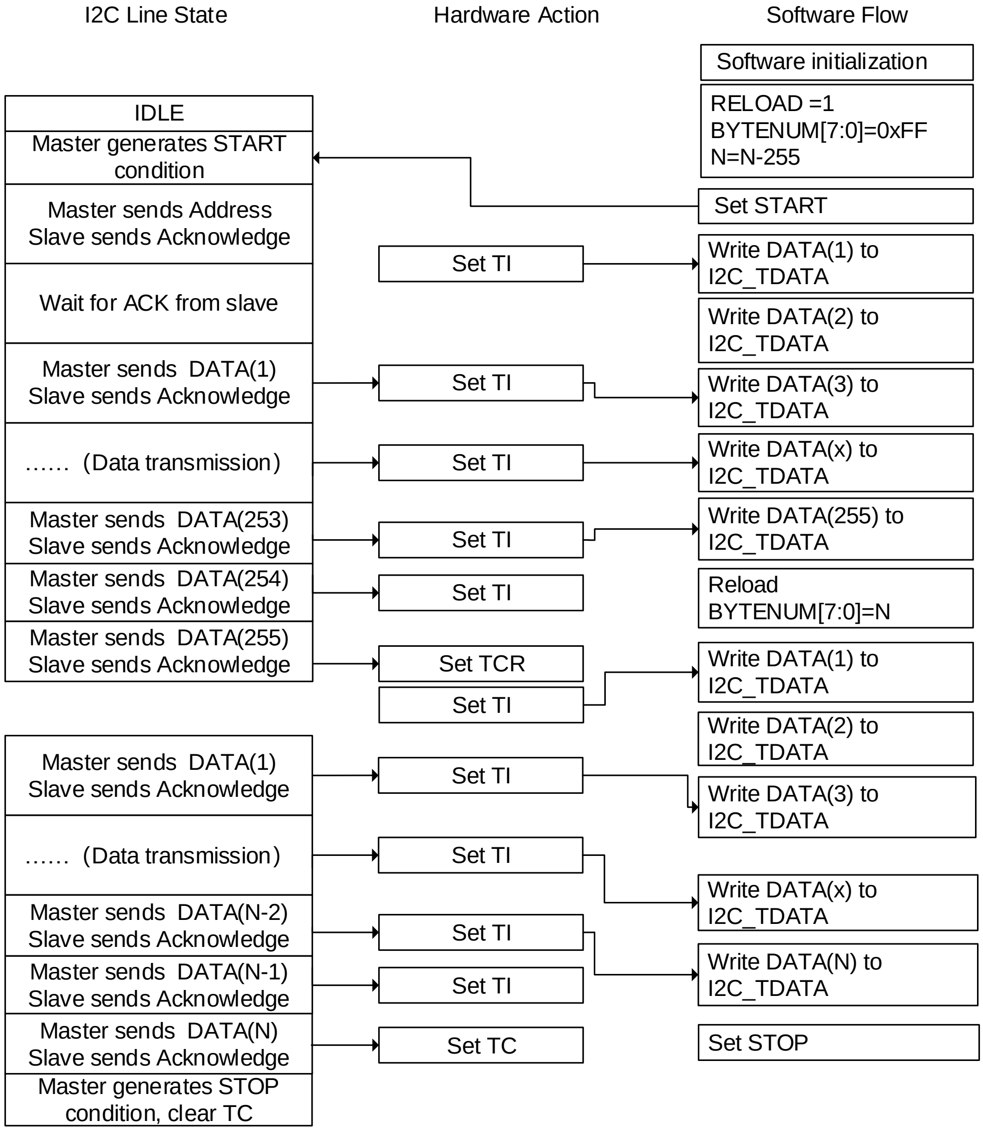


Figure 16-19. Programming model for master transmitting (N>255)



Programming model in master receiving mode

In master receiving mode, the RBNE bit in I2C_STAT register will be set when a byte is received. If the RBNEIE bit is set in I2C_CTL0 register, an interrupt will be generated. If the number of bytes to be received is greater than 255, RELOAD bit in I2C_CTL0 register must be set to enable the reload mode. In reload mode, when data of BYTENUM[7:0] bytes have been transferred, the TCR bit in I2C_STAT register will be set and the SCL stretches unitil BYTENUM[7:0] is modified with a non-zero value.

If data of BYTENUM[7:0] bytes have been transferred and RELOAD = 0, the AUTOEND bit in I2C_CTL1 can be set to generate a STOP signal automatically. When AUTOEND is 0, the TC bit in I2C_STAT register will be set and the SCL is stretched. In this case, the master can generate a STOP signal by setting the STOP bit in the I2C_CTL1 register. Or generate a RESTART signal to start a new transfer. The TC bit is cleared when the START / STOP bit is set.

Figure 16-20. Programming model for master receiving (N<=255)

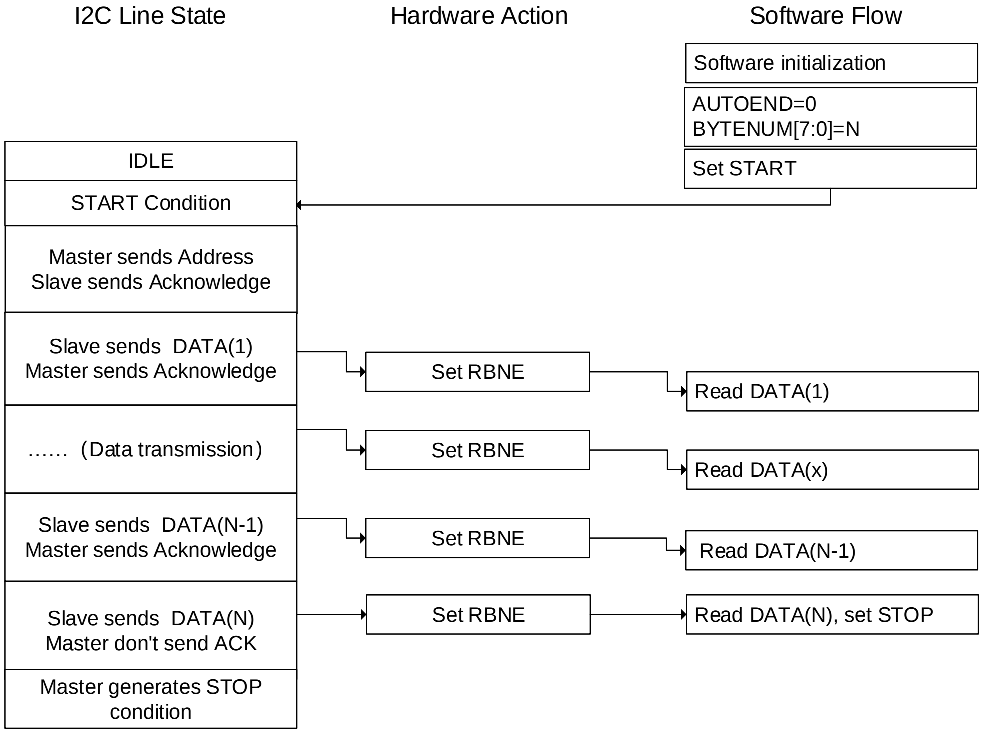
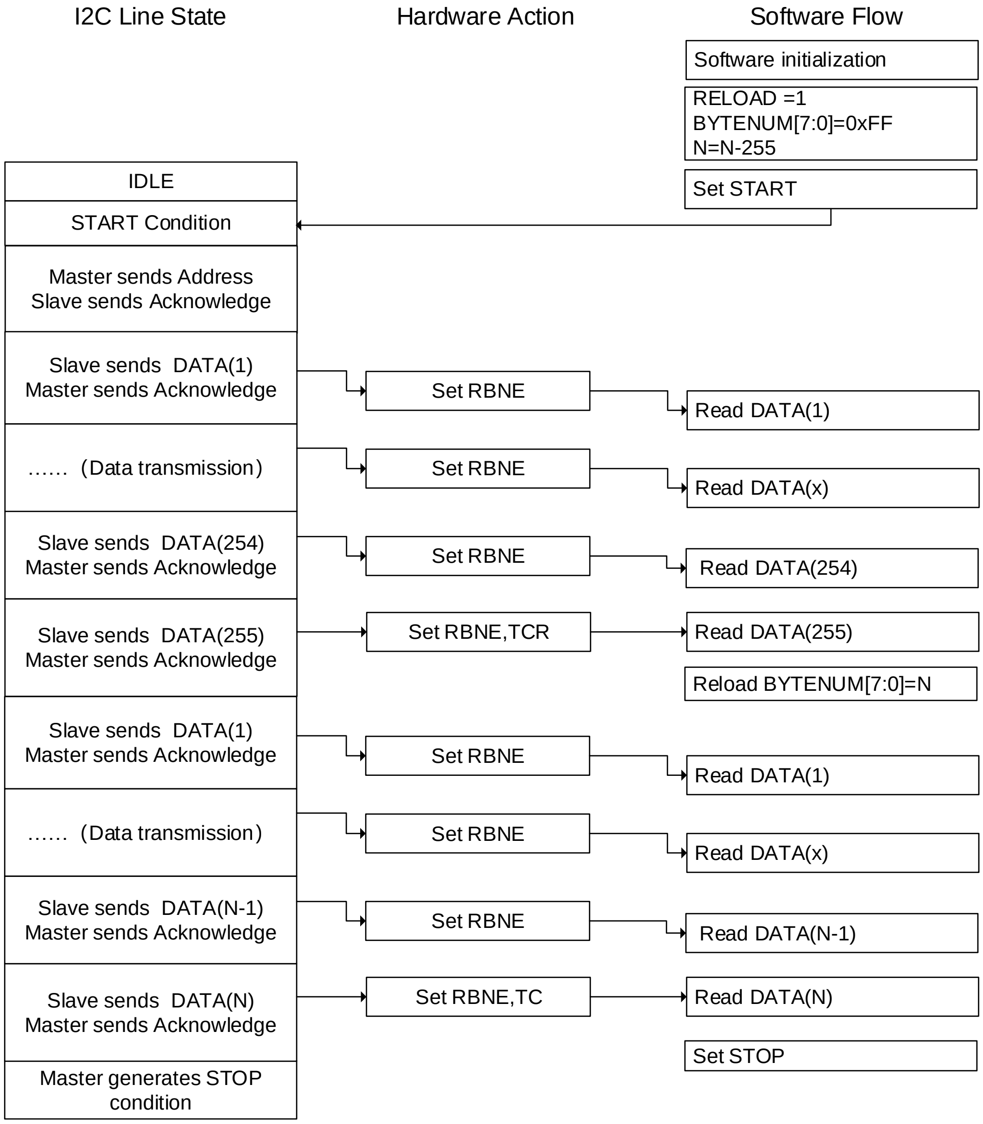


Figure 16-21. Programming model for master receiving (N>255)



16.3.9. SMBus support

The System Management Bus (abbreviated to SMBus or SMB) is a single-ended simple two-wire bus for the purpose of lightweight communication. Most commonly it is found in computer motherboards for communication with power source for ON/OFF instructions. It is derived from I2C for communication with low-bandwidth devices on a motherboard, especially power related chips such as a laptop's rechargeable battery subsystem (see Smart Battery Data).

SMBus protocol

Each message transaction on SMBus follows the format of one of the defined SMBus protocols. The SMBus protocols are a subset of the data transfer formats defined in the I2C specifications. I2C devices that can be accessed through one of the SMBus protocols are compatible with the SMBus specifications. I2C devices that do not adhere to these protocols cannot be accessed by standard methods as defined in the SMBus and Advanced

Configuration and Power Management Interface (abbreviated to ACPI) specifications.

Address resolution protocol

The SMBus uses I2C hardware and I2C hardware addressing, but adds second-level software for building special systems. Additionally, its specifications include an Address Resolution Protocol that can make dynamic address allocations. Dynamic reconfiguration of the hardware and software allow bus devices to be 'hot-plugged' and used immediately, without restarting the system. The devices are recognized automatically and assigned unique addresses. This advantage results in a plug-and-play user interface. In this protocol there is a very useful distinction between a system host and all the other devices in the system, that is the host provides address assignment function.

SMBus slave byte control

The slave byte control of SMBus receiver is the same as I2C. It allows the ACK control of each byte.The Slave Byte Control mode must be enabled by setting SBCTL bit in I2C_CTL0 register.

Host notify protocol

When the SMBHAEN bit in the I2C_CTL0 register is set, the SMBus supports the host notify protocol. In this protocol, the device acts as a master and the host as a slave, and the host will acknowledge the SMBus host address.

Time-out feature

SMBus has a time-out feature which resets devices if a communication takes too long. This explains the minimum clock frequency of 10 kHz to prevent locking up the bus. I2C can be a 'DC' bus, meaning that a slave device stretches the master clock when performing some routine while the master is accessing it. This will notify to the master that the slave is busy but does not want to lose the communication. The slave device will allow continuation after its task is completed. There is no limit in the I2C bus protocol as to how long this delay can be, whereas for a SMBus system, it would be limited to 25~35ms. SMBus protocol just assumes that if something takes too long, then it means that there is a problem on the bus and that all devices must reset in order to clear this mode. Slave devices are not allowed to hold the clock low too long.

The timeout detection can be enabled by setting TOEN and EXTOEN bits in the I2C_TIMEOUT register. The timer must be configured to guarantee that the timeout detected before the maximum time given in the SMBus specification.

The value programmed in BUSTOA[11:0] is used to check the $t_{TIMEOUT}$ parameter. To detect the SCL low level timeout, the TOIDLE bit must be 0. And the timer can be enabled by setting the TOEN bit in the I2C_TIMEOUT register, after the TOEN bit is set, the BUSTOA[11:0] and the TOIDLE bit cannot be changed. If the low level time of SCL is greater than (BUSTOA+1)*2048*t_{I2CCLK}, the TIMEOUT flag will be set in I2C_STAT register.

The BUSTOB[11:0] is used to check the $t_{LOW:SEXT}$ of the slave and the $t_{LOW:MEXT}$ of the master. The timer can be enabled by setting the EXTOEN bit in the I2C_TIMEOUT register, after the EXTOEN bit is set, the BUSTOB[11:0] cannot be changed. If the SCL stretching time of the SMBus peripheral is greater than (BUSTOB+1)*2048*t_{I2CCLK} and within the timeout interval described in the bus idle detection section, the TIMEOUT bit in the I2C_STAT register will be set.

Packet error checking

There is a CRC-8 calculator in I2C block to perform Packet Error Checking for I2C data. A PEC (packet error code) byte is appended at the end of each transfer. The byte is calculated as CRC-8 checksum, calculated over the entire message including the address and read/write bit. The polynomial used is x^8+x^2+x+1 (the CRC-8-ATM HEC algorithm, initialized to zero).

When I2C is disabled, the PEC can be enabled by setting the PECEN bit in I2C_CTL0 register.Since the PEC transmission is managed by BYTENUM[7:0] in I2C_CTL1 register, SBCTL bit must be set when act as a slave. When PECTRANS is set and the RELOAD bit is cleared, PEC is transmitted after the BYTENUM[7:0]-1 data byte. The PECTRANS has no effect if RELOAD is set.

SMBus alert

The SMBus has an extra optional shared interrupt signal called SMBALERT# which can be used by slaves to tell the host to ask its slaves about events of interest. The host processes the interrupt and accesses all SMBALERT# devices through the Alert Response Address at the same time. If the SMBALERT# is pulled low by the devices, the devices will acknowledge the Alert Response Address. When SMBHAEN is 0, it is configured as a slave device, the SMBA pin will be pulled low by setting the SMBALTEN bit in the I2C_CTL0 register. Meanwhile the Alert Response Address is enabled. When SMBHAEN is 1, it is configured as a host, and the SMBALTEN is 1, as soon as a falling edge is detected on the SMBA pin, the SMBALT flag will be set in the I2C_STAT register. If the ERRIE bit is set in the I2C_CTL0 register, an interrupt will be generated. When SMBALTEN is 0, the level of ALERT line is considered high even if the SMBA pin is low. The SMBA pin can be used as a standard GPIO if SMBALTEN is 0.

Bus idle detection

If the master detects that the high level duration of the clock and data signals is greater than $t_{HIGH,MAX}$, the bus can be considered idle.

This timing parameter includes the case of a master that has been dynamically added to the bus and may not have detected a state transition on a SMBCLK or SMBDAT lines. In this case, in order to ensure that there is no ongoing transmission, the master must wait long enough.

The BUSTOA[11:0] bits must be programmed with the timer reload value to enable the t_{IDLE} check in order to obtain the t_{IDLE} parameter. To detect SCL and SDA high level timeouts, the

TOIDLE bit must be set. Then setting the TOEN bit in the I2C_TIMEOUT register to enable the timer, after the TOEN bit is set, the BUSTOA[11:0] bit and the TOIDLE bit cannot be changed. If the high level time of both SCL and SDA is greater than (BUSTOA+1)*4*t_{I2CCLK}, the TIMEOUT flag will be set in the I2C_STAT register.

SMBus slave mode

The SMBus receiver must be able to NACK each command or data it receives. For ACK control in slave mode, slave byte control mode can be enabled by setting SBCTL bit in I2C_CTL0 register.

SMBus-specific addresses should be enabled when needed. The SMBus Device Default address (0b1100 001) is enabled by setting the SMBDAEN bit in the I2C_CTL0 register. The SMBus Host address (0b0001 000) is enabled by setting the SMBHAEN bit in the I2C_CTL0 register. The Alert Response Address (0b0001 100) is enabled by setting the SMBALTEN bit in the I2C_CTL0 register.

16.3.10. SMBus mode

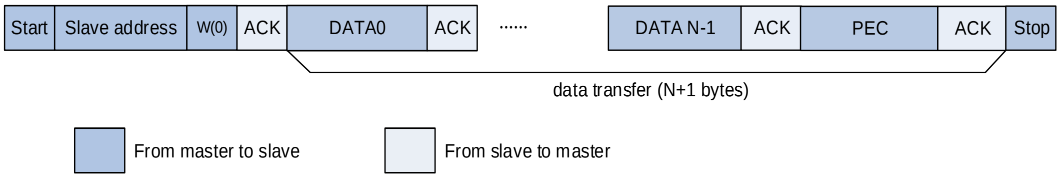
SMBus master transmitter and slave receiver

The PEC in SMBus master mode can be transmitted by setting the PECTRANS bit before setting the START bit, and the number of bytes in the BYTENUM[7:0] field must be configured. In this case, the total number of transmissions when TI interrupt occur is BYTENUM - 1. So if BYTENUM = 0x1 and PECTRANS bit is set, the data in I2C_PEC register will be transmitted automatically. If AUTOEND is 1 the SMBus master will send the STOP signal after the PEC byte automatically. If the AUTOEND is 0, the SMBus master can send a RESTART signal after the PEC. The PEC byte in I2C_PEC register will be sent after BYTENUM - 1 bytes, and the TC flag will be set after PEC is sent, then the SCL line is stretched low. The RESTART must be set in the TC interrupt routine.

When used as slave receiver, in order to allow PEC checking at the end of the number of bytes transmitted, SBCTL must be set. To configure ack control for each byte, the RELOAD must be set. In order to check the PEC byte, it is necessary to clear the RELOAD bit and set PECTRANS bit. After receiving BYTENUM-1 data, the next received byte will be compared with the data in the I2C_PEC register. If the PEC values does not match, the NACK is automatically generated. If the PEC values matches, the ACK is automatically generated, regardless of the NACKEN bit value. When PEC byte is received, it is also copied into the I2C_RDATA register, and RBNE flag will be set. If the ERRIE bit in I2C_CTL0 register is 1, when PEC value does not match, the PECERR flag will be set and the interrupt will be generated. If ACK control is not required, then PECTRANS can be set to 1 and BYTENUM can be programmed according to the number of bytes to be received.

Note: After the RELOAD bit is set, the PECTRANS cannot be changed.

Figure 16-22. SMBus master transmitter and slave receiver communication flow



SMBus master receiver and slave transmitter

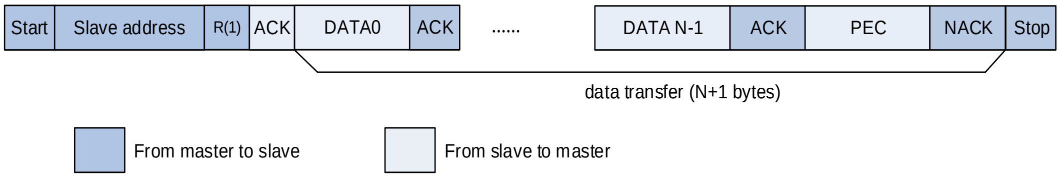
If the SMBus master is required to receive PEC at the end of bytes transfer, automatic end mode can be enabled. Before sending a START signal on the bus, PECTRANS bit must be set and slave addresses must be programmed. After receiving BYTENUM - 1 data, the next received byte will be compared with the data in the I2C_PEC register automatically. A NACK is respond to the PEC byte before STOP signal.

If the SMBus master receiver is required to generate a RESTART signal after receiving PEC byte, automatic end mode must be disabled. Before sending a START signal to the bus, PECTRANS bit must be set and slave addresses must be programmed. After receiving BYTENUM - 1 data, the next received byte will be compared with the data in the I2C_PEC register automatically. The TC flag will be set after PEC is sent, then the SCL line is stretched low. The RESTART can be set in the TC interrupt routine.

When used as slave transmitter, in order to allow PEC transmission at the end of BYTENUM[7:0] bytes, SBCTL must be set. If PECTRANS bit is set, the number of bytes in BYTENUM[7:0] contains PEC byte. In this case, if the number of bytes requested by the master is greater than BYTENUM - 1, the total number of TI interrupts will be BYTENUM - 1, and the data of the I2C_PEC register will be transmitted automatically.

Note: After the RELOAD bit is set, the PECTRANS cannot be changed.

Figure 16-23. SMBus master receiver and slave transmitter communication flow



16.3.11. Wakeup from power saving modes

When the address of I2C matches correctly, it can wake up MCU from Deep-sleep mode and Deep-sleep 1 mode. In order to wake up from these power saving modes, WUEN bit must be set in the I2C_CTL0 register and the CK_IRC48MDIV_PER must be selected as the clock source for I2CCLK. During Deep-sleep mode and Deep-sleep 1 mode, the CK_IRC48MDIV_PER is switched off. The I2C interface switches the CK_IRC48MDIV_PER on, and stretches SCL low until CK_IRC48MDIV_PER is woken up when a START is detected. Then the CK_IRC48MDIV_PER is used as the clock of I2C to receive the address. When address matching is detected, I2C stretches SCL during MCU wake-up. The SCL is released until the software clears the ADDSEND flag and the transmission proceeds normally. If the detected address does not match, CK_IRC48MDIV_PER will be closed again and the MCU will not be wake up.

Only an address match interrupt (ADDMIE = 1) can wakeup the MCU. If the clock source of I2C is the system clock, or WUEN = 0, CK_IRC48MDIV_PER will not switched on after receiving start signal. When wakeup from power saving mode is enabled, the digital filter must be disabled and the SS bit in I2C_CTL0 must be cleared. Before entering power saving mode, the I2C peripheral must be disabled (I2CEN = 0) if wakeup from power saving mode is disabled (WUEN = 0).

16.3.12. Use DMA for data transfer

As is shown in I2C slave mode and I2C master mode, each time TI or RBNE is asserted, software should write or read a byte, this may cause CPU's high overload. The DMA controller can be used to process TI and RBNE flag: each time TI or RBNE is asserted, DMA controller does a read or write operation automatically.

The DMA transmission request is enabled by setting the DENT bit in I2C_CTL0 register. The DMA reception request is enabled by setting the DENR bit in I2C_CTL0 register. In master mode, the slave address, transmission direction, number of bytes and START bit are programmed by software. The DMA must be initialized before setting the START bit. The number of bytes to be transferred is configured in the BYTENUM[7:0] in I2C_CTL1 register. In slave mode, the DMA must be initialized before the address match event or in the ADDSEND interrupt routine, before clearing the ADDSEND flag.

16.3.13. I2C error and interrupts

The I2C error flags are listed in ***Table 16-4. I2C error flags***.

Table 16-4. I2C error flags

I2C Error Name	Description
BERR	Bus error
LOSTARB	Arbitration lost
OUERR	Overrun / Underrun flag
PECERR	CRC value doesn't match
TIMEOUT	Bus timeout in SMBus mode
SMBALT	SMBus Alert

The I2C interrupt events and flags are listed in ***Table 16-5. I2C interrupt events***.

Table 16-5. I2C interrupt events

Interrupt event	Event flag	Enable control bit
I2C_RDATA is not empty during receiving	RBNE	RBNEIE
Transmit interrupt	TI	TIE

Interrupt event	Event flag	Enable control bit
STOP signal detected in slave mode	STPDET	STPDETIE
Transfer complete reload	TCR	TCIE
Transfer complete	TC	
Address match	ADDSEND	ADDMIE
Not acknowledge received	NACK	NACKIE
Bus error	BERR	ERRIE
Arbitration Lost	LOSTARB	
Overrun / Underrun error	OUERR	
PEC error	PECERR	
Timeout error	TIMEOUT	
SMBus Alert	SMBALT	

16.3.14. I2C debug mode

When the microcontroller enters the debug mode (Cortex®-M23 core halted), the SMBus timeout either continues to work normally or stops, depending on the I2Cx_HOLD configuration bits in the DBG module.

16.4. Register definition

I2C0 base address: 0x4000 5400

I2C1 base address: 0x4000 5800

16.4.1. Control register 0 (I2C_CTL0)

Address offset: 0x00

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved								PECEN	SMBALTEN	SMBDAEN	SMBHAEN	GCEN	WUEN	SS	SBCTL
								rw	rw	rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DENR	DENT	Reserved	ANOFF	DNF[3:0]				ERRIE	TCIE	STPDETIE	NACKIE	ADDMIE	RBNEIE	TIE	I2CEN
rw	rw		rw	rw				rw	rw	rw	rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31:24	Reserved	Must be kept at reset value.
23	PECEN	PEC Calculation Switch 0: PEC Calculation off 1: PEC Calculation on Note: If the SMBus feature is not supported, this bit is reserved and forced by hardware to ‘0’.
22	SMBALTEN	SMBus Alert enable 0: SMBA pin is not pulled down (device mode) or SMBus Alert pin SMBA is disabled (host mode) 1: SMBA pin is pulled down (device mode) or SMBus Alert pin SMBA is enabled (host mode) Note: If the SMBus feature is not supported, this bit is reserved and forced by hardware to ‘0’.
21	SMBDAEN	SMBus device default address enable 0: Device default address is disabled, the default address 0b1100001x will be not acknowledged. 1: Device default address is enabled, the default address 0b1100001x will be acknowledged. Note: If the SMBus feature is not supported, this bit is reserved and forced by hardware to ‘0’.
20	SMBHAEN	SMBus host address enable

		0: Host address is disabled, address 0b0001000x will be not acknowledged. 1: Host address is enabled, address 0b0001000x will be acknowledged. Note: If the SMBus feature is not supported, this bit is reserved and forced by hardware to ‘0’.
19	GCEN	Whether or not to response to a General Call (0x00) 0: Slave won’t response to a General Call 1: Slave will response to a General Call
18	WUEN	Wakeup from power saving mode enable, including Deep-sleep mode and Deep-sleep 1 mode. This bit is cleared when mcu wakeup from power saving mode. 0: Wakeup from power saving mode disable. 1: Wakeup from power saving mode enable. Note: WUEN can be set only when DNF[3:0] = 0000.
17	SS	Whether to stretch SCL low when data is not ready in slave mode. This bit is set and cleared by software. 0: SCL Stretching is enabled 1: SCL Stretching is disabled Note: When in master mode, this bit must be 0. This bit can be modified when I2CEN = 0.
16	SBCTL	Slave byte control This bit is used to enable hardware byte control in slave mode. 0: Slave byte control is disabled 1: Slave byte control is enabled
15	DENR	DMA enable for reception 0: DMA is disabled for reception 1: DMA is enabled for reception
14	DENT	DMA enable for transmission 0: DMA is disabled for transmission 1: DMA is enabled for transmission
13	Reserved	Must be kept at reset value.
12	ANOFF	Analog noise filter disable 0: Analog noise filter is enabled 1: Analog noise filter is disabled Note: This bit can only be programmed when the I2C is disabled (I2CEN = 0).
11:8	DNF[3:0]	Digital noise filter 0000: Digital filter is disabled 0001: Digital filter is enabled and filter spikes with a length of up to 1 t_{I2CCLK} ... 1111: Digital filter is enabled and filter spikes with a length of up to15 t_{I2CCLK}

		Note: These bits can only be modified when the I2C is disabled (I2CEN = 0).
7	ERRIE	Error interrupt enable 0: Error interrupt disabled 1: Error interrupt enabled. When BERR, LOSTARB, OUERR, PECERR, TIMEOUT or SMBALT bit is set, an interrupt will be generated.
6	TCIE	Transfer complete interrupt enable 0: Transfer complete interrupt is disabled 1: Transfer complete interrupt is enabled
5	STPDETIE	Stop detection interrupt enable 0: Stop detection (STPDET) interrupt is disabled 1: Stop detection (STPDET) interrupt is enabled
4	NACKIE	NACK received interrupt enable 0: NACK received interrupt is disabled 1: NACK received interrupt is enabled
3	ADDMIE	Address match interrupt enable in slave mode 0: Address matchinterrupt is disabled 1: Address matchnterrupt is enabled
2	RBNEIE	Receive interrupt enable 0: Receive (RBNE) interrupt is disabled 1: Receive (RBNE) interrupt is enabled
1	TIE	Transmit interrupt enable 0: Transmit (TI) interrupt is disabled 1: Transmit (TI) interrupt is enabled
0	I2CEN	I2C peripheral enable 0: I2C is disabled 1: I2C is enabled

16.4.2. Control register 1 (I2C_CTL1)

Address offset: 0x04

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved					PECTRANS	AUTOEND	RELOAD	BYTENUM[7:0]							
					rw	rw	rw	rw							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NACKEN	STOP	START	HEAD10R	ADD10EN	TRDIR	SADDRESS[9:0]									

rw	rw	rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31:27	Reserved	Must be kept at reset value.
26	PECTRANS	PEC Transfer Set by software. Cleared by hardware in the following cases: When PEC byte is transferred or ADDSEND bit is set or STOP signal is detected or I2CEN = 0. 0: Don't transfer PEC value 1: Transfer PEC Note: This bit has no effect when RELOAD = 1, or SBCTL = 0 in slave mode. If the SMBus feature is not supported, this bit is reserved and forced by hardware to '0'.
25	AUTOEND	Automatic end mode in master mode 0: TC bit is set when the transfer of BYTENUM[7:0] bytes is completed. 1: a STOP signal is sent automatically when the transfer of BYTENUM[7:0] bytes is completed. Note: This bit works only when RELOAD = 0. This bit is set and cleared by software.
24	RELOAD	Reload mode 0: After the data of BYTENUM[7:0] bytes transfer, the transfer is completed. 1: After data of BYTENUM[7:0] bytes transfer, the transfer is not completed and the new BYTENUM[7:0] will be reloaded. Every time when the BYTENUM[7:0] bytes have been transferred, the TCR bit in I2C_STAT register will be set. Note: This bit is set and cleared by software.
23:16	BYTENUM[7:0]	Number of bytes to be transferred These bits are programmed with the number of bytes to be transferred. When SBCTL = 0, these bits have no effect. Note: These bits should not be modified when the START bit is set.
15	NACKEN	Generate NACK in slave mode 0: an ACK is sent after receiving a new byte. 1: a NACK is sent after receiving a new byte. Note: The bit can be set by software, and cleared by hardware when the NACK is sent, or when a STOP signal is detected or ADDSEND is set, or when I2CEN = 0. When PEC is enabled, whether to send an ACK or a NACK is not depend on the NACKEN bit. When SS = 1, and the OUERR bit is set, the value of NACKEN is ignored and a NACK will be sent.
14	STOP	Generate a STOP signal on I2C bus This bit is set by software and cleared by hardware when I2CEN = 0 or STOP signal is detected. 0: STOP will not be sent 1: STOP will be sent

13	START	Generate a START signal on I2C bus This bit is set by software and cleared by hardware after the address is sent. When the arbitration is lost, or a timeout error occurred, or I2CEN = 0, this bit can also be cleared by hardware. It can be cleared by software by setting the ADDSENDC bit in I2C_STATC register. 0: START will not be sent 1: START will be sent
12	HEAD10R	10-bit address header executes read direction only in master receive mode 0: The 10 bit master receive address sequence is START + header of 10-bit address (write) + slave address byte 2 + RESTART + header of 10-bit address (read). 1: The 10 bit master receive address sequence is RESTART + header of 10-bit address (read). Note: When the START bit is set, this bit can not be changed.
11	ADD10EN	10-bit addressing mode enable in master mode 0: 7-bit addressing in master mode 1: 10-bit addressing in master mode Note: When the START bit is set, this bit can not be modified.
10	TRDIR	Transfer direction in master mode 0: Master transmit 1: Master receive Note: When the START bit is set, this bit can not be modified.
9:0	SADDRESS[9:0]	Slave address to be sent SADDRESS[9:8]: Slave address bit 9:8 If ADD10EN = 0, these bits have no effect. If ADD10EN = 1, these bits should be written with bits 9:8 of the slave address to be sent. SADDRESS[7:1]: Slave address bit 7:1 If ADD10EN = 0, these bits should be written with the 7-bit slave address to be sent. If ADD10EN = 1, these bits should be written with bits 7:1 of the slave address to be sent. SADDRESS0: Slave address bit 0 If ADD10EN = 0, this bit has no effect. If ADD10EN = 1, this bit should be written with bit 0 of the slave address to be sent Note: When the START bit is set, the bit filed can not be modified.

16.4.3. Slave address register 0 (I2C_SADDR0)

Address offset: 0x08

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDRESSEN	Reserved				ADDFORMAT	ADDRESS[9:8]		ADDRESS[7:1]							ADDRESS0
rw					rw	rw		rw							rw

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15	ADDRESSEN	I2C address enable 0: I2C address disable. 1: I2C address enable.
14:11	Reserved	Must be kept at reset value.
10	ADDFORMAT	Address mode for the I2C slave 0: 7-bit address 1: 10-bit address Note: When ADDRESSEN is set, this bit should not be written.
9:8	ADDRESS[9:8]	Highest two bits of a 10-bit address Note: When ADDRESSEN is set, this bit should not be written.
7:1	ADDRESS[7:1]	7-bit address or bits 7:1 of a 10-bit address Note: When ADDRESSEN is set, this bit should not be written.
0	ADDRESS0	Bit 0 of a 10-bit address Note: When ADDRESSEN is set, this bit should not be written.

16.4.4. Slave address register 1 (I2C_SADDR1)

Address offset: 0x0C

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDRESS2EN	Reserved				ADDMSK2[2:0]			ADDRESS2[7:1]							Reserved
rw					rw			rw							

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15	ADDRESS2EN	Second I2C address enable 0: Second I2C address disable.

		1: Second I2C address enable.
14:11	Reserved	Must be kept at reset value.
10:8	ADDMSK2[2:0]	ADDRESS2[7:1] mask Defines which bits of ADDRESS2[7:1] are compared with an incoming address byte, and which bits are masked (don't care). 000: No mask, all the bits must be compared. n(001~110): ADDRESS2[n:0] is masked. Only ADDRESS2[7:n+1] are compared. 111: ADDRESS2[7:1] are masked. All 7-bit received addresses are acknowledged except the reserved address (0b0000xxx and 0b1111xxx). Note: When ADDRESS2EN is set, these bits should not be written. If ADDMSK2 is not equal to 0, the reserved I2C addresses (0b0000xxx and 0b1111xxx) are not acknowledged even if all the bits are matched.
7:1	ADDRESS2[7:1]	Second I2C address for the slave Note: When ADDRESS2EN is set, these bits should not be written.
0	Reserved	Must be kept at reset value.

16.4.5. Timing register (I2C_TIMING)

Address offset: 0x10

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PSC[3:0]				Reserved				SCLDELY[3:0]				SDADELY[3:0]			
rw								rw				rw			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCLH[7:0]								SCLL[7:0]							
rw								rw							

Bits	Fields	Descriptions
31:28	PSC[3:0]	Timing prescaler In order to generate the clock period t_{PSC} used for data setup and data hold counters, these bits are used to configure the prescaler for I2CCLK. The t_{PSC} is also used for SCL high and low level counters. $t_{PSC}=(PSC+1)*t_{I2CCLK}$
27:24	Reserved	Must be kept at reset value.
23:20	SCLDELY[3:0]	Data setup time A delay $t_{SCLDELY}$ between SDA edge and SCL rising edge can be generated by configuring these bits. And during $t_{SCLDELY}$, the SCL line is stretched low in master mode and in slave mode when SS = 0. $t_{SCLDELY}=(SCLDELY+1)*t_{PSC}$

19:16	SDADELY[3:0]	Data hold time A delay $t_{SDADELY}$ between SCL falling edge and SDA edge can be generated by configuring these bits. And during $t_{SDADELY}$, the SCL line is stretched low in master mode and in slave mode when SS = 0. $t_{SDADELY}$=SDADELY*t_{PSC}
15:8	SCLH[7:0]	SCL high period SCL high period can be generated by configuring these bits. t_{SCLH}=(SCLH+1)*t_{PSC} Note: These bits can only be used in master mode.
7:0	SCLL[7:0]	SCL low period SCL low period can be generated by configuring these bits. t_{SCLL}=(SCLL+1)*t_{PSC} Note: These bits can only be used in master mode.

16.4.6. Timeout register (I2C_TIMEOUT)

Address offset: 0x14

Reset value: 0x0000 0000

Note: If the SMBus feature is not supported, this register is reserved and forced by hardware to “0x00000000”.

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EXTOEN	Reserved			BUSTOB[11:0]											
rw				rw											

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TOEN	Reserved		TOIDLE	BUSTOA[11:0]											
rw			rw	rw											

Bits	Fields	Descriptions
31	EXTOEN	Extended clock timeout detection enable When a cumulative SCL stretch time is greater than $t_{LOW:EXT}$, a timeout error will be occurred. $t_{LOW:EXT}$=(BUSTOB+1)*2048*t_{I2CCLK}. 0: Extended clock timeout detection is disabled. 1: Extended clock timeout detection is enabled.
30:28	Reserved	Must be kept at reset value.
27:16	BUSTOB[11:0]	Bus timeout B Configure the cumulative clock extension timeout. In master mode, the master cumulative clock low extend time $t_{LOW:MEXT}$ is detected. In slave mode, the slave cumulative clock low extend time $t_{LOW:SEXT}$ is detected. $t_{LOW:EXT}$=(BUSTOB+1)*2048*t_{I2CCLK}.

		Note: These bits can be modified only when EXTOEN = 0.
15	TOEN	Clock timeout detection enable If the SCL stretch time greater than $t_{TIMEOUT}$ when TOIDLE = 0 or high for more than t_{IDLE} when TOIDLE = 1, a timeout error is detected. 0: SCL timeout detection is disabled 1: SCL timeout detection is enabled
14:13	Reserved	Must be kept at reset value.
12	TOIDLE	Idle clock timeout detection 0: BUSTOA is used to detect SCL low timeout 1: BUSTOA is used to detect both SCL and SDA high timeout when the bus is idle Note: This bit can be written only when TOEN = 0.
11:0	BUSTOA[11:0]	Bus timeout A When TOIDLE = 0, $t_{TIMEOUT}$=(BUSTOA+1)*2048*t_{I2CCLK}. When TOIDLE = 1, t_{IDLE}=(BUSTOA+1)*4*t_{I2CCLK}. Note: These bits can be written only when TOEN = 0.

16.4.7. Status register (I2C_STAT)

Address offset: 0x18

Reset value: 0x0000 0001

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved								READDR[6:0]							TR
											r				r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
I2CBSY	Reserved	SMBALT	TIMEOUT	PECERR	OUERR	LOSTARB	BERR	TCR	TC	STPDET	NACK	ADDSEND	RBNE	TI	TBE
r		r	r	r	r	r	r	r	r	r	r	r	r	rw	rw

Bits	Fields	Descriptions
31:24	Reserved	Must be kept at reset value.
23:17	READDR[6:0]	Received match address in slave mode When the ADDSEND bit is set, these bits store the matched address. In the case of a 10-bit address, READDR[6:0] stores the header of the 10-bit address followed by the 2 MSBs of the address.
16	TR	Whether the I2C is a transmitter or a receiver in slave mode This bit is updated when the ADDSEND bit is set. 0: Receiver 1: Transmitter
15	I2CBSY	Busy flag

		This bit is set by hardware when a START signal is detected and cleared by hardware after a STOP signal. When I2CEN = 0, this bit is also cleared by hardware. 0: No I2C communication. 1: I2C communication active.
14	Reserved	Must be kept at reset value.
13	SMBALT	SMBus Alert When SMBHAEN = 1, SMBALTEN = 1, and a SMBALERT event (falling edge) is detected on SMBA pin, this bit will be set by hardware. It is cleared by software by setting the SMBALTC bit. This bit is cleared by hardware when I2CEN = 0. 0: SMBALERT event is not detected on SMBA pin 1: SMBALERT event is detected on SMBA pin Note: If the SMBus feature is not supported, this bit is reserved and forced by hardware to ‘0’.
12	TIMEOUT	TIMEOUT flag. When a timeout or extended clock timeout occurred, this bit will be set. It is cleared by software by setting the TIMEOUTC bit and cleared by hardware when I2CEN = 0. 0: no timeout or extended clock timeout occur 1: a timeout or extended clock timeout occur Note: If the SMBus feature is not supported, this bit is reserved and forced by hardware to ‘0’.
11	PECERR	PEC error This flag is set by hardware when the received PEC does not match with the content of I2C_PEC register. Then a NACK is automatically sent. It is cleared by software by setting the PECERRC bit and cleared by hardware when I2CEN = 0. 0: Received PEC and content of I2C_PEC match 1: Received PEC and content of I2C_PEC don’t match, I2C will send NACK regardless of NACKEN bit. Note: If the SMBus feature is not supported, this bit is reserved and forced by hardware to ‘0’.
10	OUERR	Overrun/Underrun error in slave mode In slave mode with SS = 1, when an overrun/underrun error occurs, this bit will be set by hardware. It is cleared by software by setting the OUERRC bit and cleared by hardware when I2CEN = 0. 0: No overrun or underrun occurs 1: Overrun or underrun occurs
9	LOSTARB	Arbitration Lost It is cleared by software by setting the LOSTARBC bit and cleared by hardware when I2CEN = 0. 0: No arbitration lost. 1: Arbitration lost occurs and the I2C block changes back to slave mode.

8	BERR	Bus error When an unexpected START or STOP signal on I2C bus is detected, a bus error occurs and this bit will be set. It is cleared by software by setting BERRC bit and cleared by hardware when I2CEN = 0. 0: No bus error 1: A bus error detected
7	TCR	Transfer complete reload This bit is set by hardware when RELOAD = 1 and data of BYTENUM[7:0] bytes have been transferred. It is cleared by software when BYTENUM[7:0] is written to a non-zero value. 0: When RELOAD = 1, transfer of BYTENUM[7:0] bytes is not completed 1: When RELOAD = 1, transfer of BYTENUM[7:0] bytes is completed
6	TC	Transfer complete in master mode This bit is set by hardware when RELOAD = 0, AUTOEND = 0 and data of BYTENUM[7:0] bytes have been transferred. It is cleared by software when START bit or STOP bit is set. 0: Transfer of BYTENUM[7:0] bytes is not completed 1: Transfer of BYTENUM[7:0] bytes is completed
5	STPDET	STOP signal detected in slave mode This flag is set by hardware when a STOP signal is detected on the bus. It is cleared by software by setting STPDETC bit and cleared by hardware when I2CEN = 0. 0: STOP signal is not detected. 1: STOP signal is detected.
4	NACK	Not Acknowledge flag This flag is set by hardware when a NACK is received. It is cleared by software by setting NACKC bit and cleared by hardware when I2CEN = 0. 0: ACK is received. 1: NACK is received.
3	ADDSEND	Address received matches in slave mode. This bit is set by hardware when the received slave address matched with one of the enabled slave addresses. It is cleared by software by setting ADDSENDC bit and cleared by hardware when I2CEN = 0. 0: Received address not matched 1: Received address matched
2	RBNE	I2C_RDATA is not empty during receiving This bit is set by hardware when the received data is shift into the I2C_RDATA register. It is cleared when I2C_RDATA is read. 0: I2C_RDATA is empty 1: I2C_RDATA is not empty, software can read
1	TI	Transmit interrupt This bit is set by hardware when the I2C_TDATA register is empty and the I2C is

		ready to transmit data. It is cleared when the next data to be sent is written in the I2C_TDATA register.When SS = 1, this bit can be set by software, in order to generate a TI event (interrupt if TIE = 1 or DMA request if DENT = 1). 0: I2C_TDATA is not empty or the I2C is not ready to transmit data 1: I2C_TDATA is empty and the I2C is ready to transmit data
0	TBE	I2C_TDATA is empty during transmitting This bit is set by hardware when the I2C_TDATA register is empty. It is cleared when the next data to be sent is written in the I2C_TDATA register. This bit can be set by software in order to empty the I2C_TDATA register. 0: I2C_TDATA is not empty 1: I2C_TDATA is empty

16.4.8. Status clear register (I2C_STATC)

Address offset: 0x1C

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved		SMBALTC	TIMEOUTC	PECERRC	OUERRC	LOSTARBC	BERRC	Reserved		STPDETC	NACKC	ADDSENDC	Reserved		
		w	w	w	w	w	w			w	w	w			

Bits	Fields	Descriptions
31:14	Reserved	Must be kept at reset value.
13	SMBALTC	SMBus alert flag clear. Software can clear the SMBALT bit of I2C_STAT by writing 1 to this bit. Note: If the SMBus feature is not supported, this bit is reserved and forced by hardware to ‘0’.
12	TIMEOUTC	TIMEOUT flag clear. Software can clear the TIMEOUT bit of I2C_STAT by writing 1 to this bit. Note: If the SMBus feature is not supported, this bit is reserved and forced by hardware to ‘0’.
11	PECERRC	PEC error flag clear. Software can clear the PECERR bit of I2C_STAT by writing 1 to this bit. Note: If the SMBus feature is not supported, this bit is reserved and forced by hardware to ‘0’.
10	OUERRC	Overrun/Underrun flag clear.

		Software can clear the OUERR bit of I2C_STAT by writing 1 to this bit.
9	LOSTARBC	Arbitration Lost flag clear. Software can clear the LOSTARB bit of I2C_STAT by writing 1 to this bit.
8	BERRC	Bus error flag clear. Software can clear the BERR bit of I2C_STAT by writing 1 to this bit.
7:6	Reservced	Must be kept at reset value.
5	STPDETC	STPDET flag clear Software can clear the STPDET bit of I2C_STAT by writing 1 to this bit.
4	NACKC	Not Acknowledge flag clear Software can clear the NACK bit of I2C_STAT by writing 1 to this bit.
3	ADDSENDC	ADDSEND flag clear Software can clear the ADDSEND bit of I2C_STAT by writing 1 to this bit.
2:0	Reserved	Must be kept at reset value.

16.4.9. PEC register (I2C_PEC)

Address offset: 0x20

Reset value: 0x0000 0000

Note: If the SMBus feature is not supported, this register is reserved and forced by hardware to “0x00000000”.

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								PECV[7:0]							
								r							

Bits	Fields	Descriptions
31:8	Reserved	Must be kept at reset value.
7:0	PECV[7:0]	Packet Error Checking Value that calculated by hardware when PEC is enabled. PECV is cleared by hardware when I2CEN = 0.

16.4.10. Receive data register (I2C_RDATA)

Address offset: 0x24

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								RDATA[7:0]							
								r							

Bits	Fields	Descriptions
31:8	Reserved	Must be kept at reset value.
7:0	RDATA[7:0]	Receive data value

16.4.11. Transmit data register (I2C_TDATA)

Address offset: 0x28

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								TDATA [7:0]							
								rw							

Bits	Fields	Descriptions
31:8	Reserved	Must be kept at reset value.
7:0	TDATA[7:0]	Transmit data value

16.4.12. Control register 2 (I2C_CTL2)

Address offset: 0x90

Reset value: 0x0000 FE00

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDM[6:0]							Reserved								
rw															

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:9	ADDM[6:0]	Defines which bits of ADDRESS[7:1] are compared with an incoming address byte,

		and which bits are ignored. Any bit set to 1 in ADDM[6:0] enables comparisons with the corresponding bit in ADDRESS[7:1]. Bits set to 0 are ignored (can be either 0 or 1 in the incoming address).
8:0	Reserved	Must be kept at reset value.

17. Serial peripheral interface / Inter-IC sound (SPI / I2S)

17.1. Overview

The SPI/I2S module can communicate with external devices using the SPI protocol or the I2S audio protocol.

The serial peripheral interface (SPI) provides a SPI protocol of data transmission and reception function in master or slave mode. Both full-duplex and simplex communication modes are supported, with hardware CRC calculation and checking. Quad-SPI master mode is also supported in SPI1.

The inter-IC sound (I2S) supports four audio standards: I2S Phillips standard, MSB justified standard, LSB justified standard, and PCM standard. I2S works at either master or slave mode for transmission and reception.

17.2. Characteristics

17.2.1. SPI characteristics

- Master or slave operation with full-duplex or simplex mode.
- Separate transmit and receive buffer, 16 bits wide (only in SPI0).
- Separate transmission and reception 32-bit FIFO (only in SPI1).
- Data frame size can be 8 or 16 bits (only in SPI0).
- Data frame size can be 4 to 16 bits (only in SPI1).
- Bit order can be LSB or MSB.
- Software and hardware NSS management.
- Hardware CRC calculation, transmission and checking.
- Transmission and reception using DMA.
- SPI TI mode supported.
- SPI NSS pulse mode supported.
- Quad-SPI configuration available in master mode (only in SPI1).

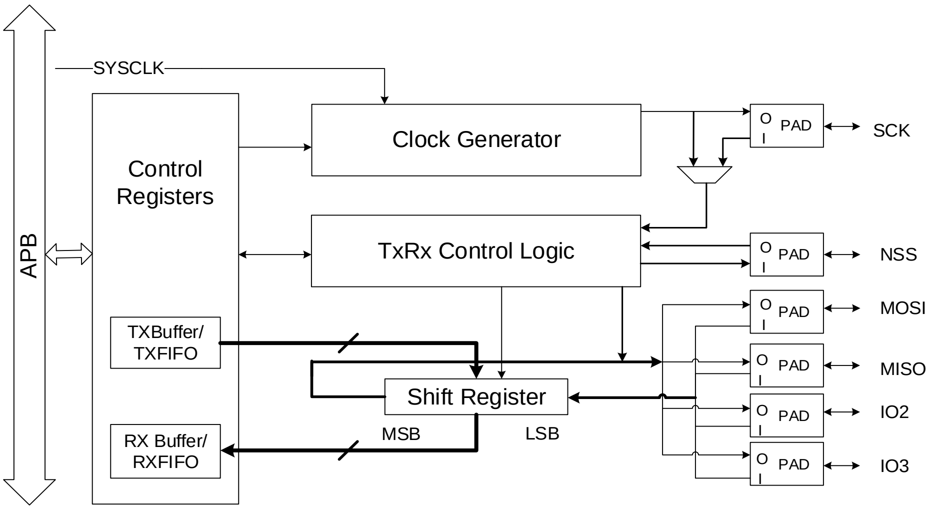
17.2.2. I2S characteristics

- Master or slave operation for transmission / reception.
- Four I2S standards supported: Phillips, MSB justified, LSB justified and PCM standard.
- Data length can be 16 bits, 24 bits or 32 bits.
- Channel length can be 16 bits or 32 bits.
- Transmission and reception using a 16 bits wide buffer.
- Audio sample frequency can be 8 kHz to 192 kHz using I2S clock divider.
- Programmable idle state clock polarity.
- Transmission and reception using DMA.

17.3. SPI function overview

17.3.1. SPI block diagram

Figure 17-1. Block diagram of SPI



17.3.2. SPI signal description

Normal configuration (Not Quad-SPI Mode)

Table 17-1. SPI signal description

Pin name	Direction	Description
SCK	I/O	Master: SPI clock output Slave: SPI clock input
MISO	I/O	Master: Data reception line Slave: Data transmission line Master with bidirectional mode: Not used Slave with bidirectional mode: Data transmission and reception line.
MOSI	I/O	Master: Data transmission line Slave: Data reception line Master with bidirectional mode: Data transmission and reception line. Slave with bidirectional mode: Not used
NSS	I/O	Software NSS mode: not used Master in hardware NSS mode: when NSSDRV=1, it is NSS output, suitable for single master application; when NSSDRV=0, it is NSS input, suitable for multi-master application.

Pin name	Direction	Description
		Slave in hardware NSS mode: NSS input, as a chip select signal for slave.

Quad-SPI configuration

SPI is in single wire mode by default and enters into Quad-SPI mode after QMOD bit in SPI_QCTL register is set (only available in SPI1). Quad-SPI mode can only work in master mode.

The IO2 and IO3 pins can be driven high in normal Non-Quad-SPI mode by configuring IO23_DRV bit in SPI_QCTL register.

The SPI is connected to external devices through 6 pins in Quad-SPI mode:

Table 17-2. Quad-SPI signal description

Pin name	Direction	Description
SCK	O	SPI clock output
MOSI	I/O	Transmission/Reception data 0
MISO	I/O	Transmission/Reception data 1
IO2	I/O	Transmission/Reception data 2
IO3	I/O	Transmission/Reception data 3
NSS	O	NSS output

17.3.3. SPI clock timing and data format

CKPL and CKPH bits in SPI_CTL0 register decide the timing of SPI clock and data signal. The CKPL bit decides the SCK level when idle and CKPH bit decides either first or second clock edge is a valid sampling edge. These bits take no effect in TI mode.

In SPI1 normal mode, the length of data is configured by the DZ bits in the SPI_CTL1 register. It can be set from 4-bit up to 16-bit length and the setting applies for both transmission and reception, and the read access to the FIFO must be aligned with the BYTEN bit setting in the SPI_CTL1 register. The data frame length is fixed to 8 bits in Quad-SPI mode.

Data order is configured by LF bit in SPI_CTL0 register, and SPI will first send the LSB if LF=1, or the MSB if LF=0. The data order is fixed to MSB first in TI mode.

When the SPI_DATA register is accessed, data frames are always right-aligned into either a byte (if the data fits into a byte) or a half-word. During communication, only bits within the data frame are clocked and transferred.

Figure 17-2. SPI1 timing diagram in normal mode

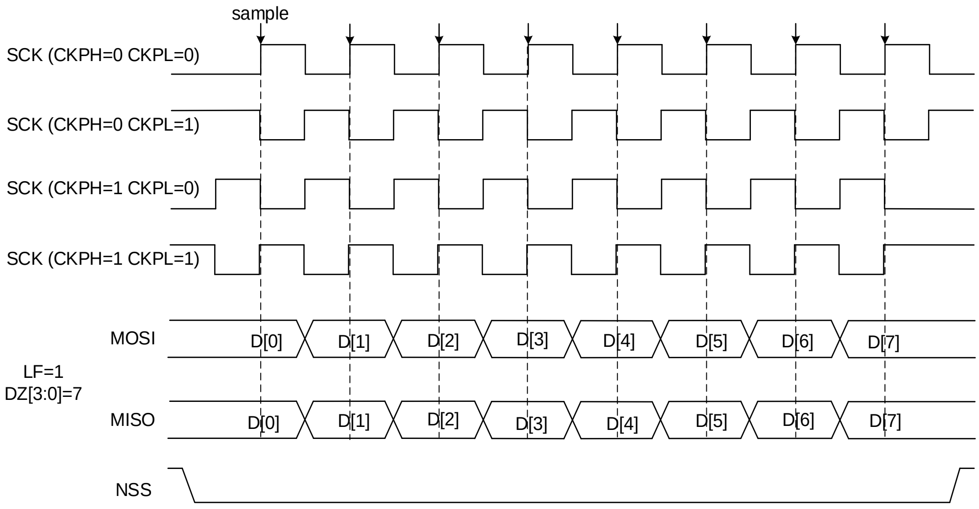
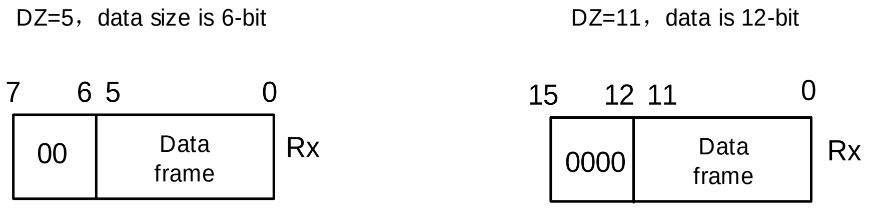


Figure 17-3. SPI1 data frame right-aligned diagram



In SPI0 normal mode, the length of data is configured by the FF16 bit in the SPI_CTL0 register. Data length is 16 bits if FF16=1, otherwise is 8 bits.

Data order is configured by LF bit in SPI_CTL0 register, and SPI0 will first send the LSB if LF=1, or the MSB if LF=0. The data order is fixed to MSB first in TI mode.

Figure 17-4. SPI0 timing diagram in normal mode

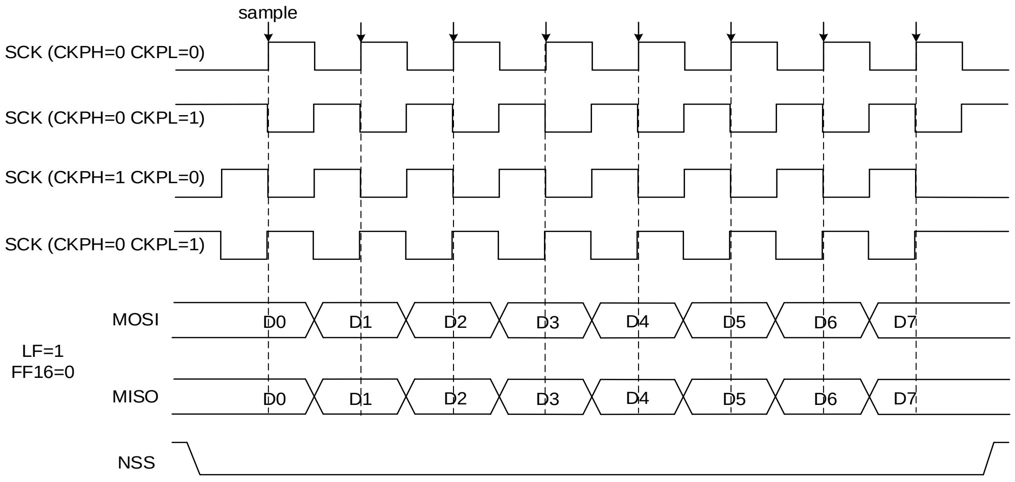
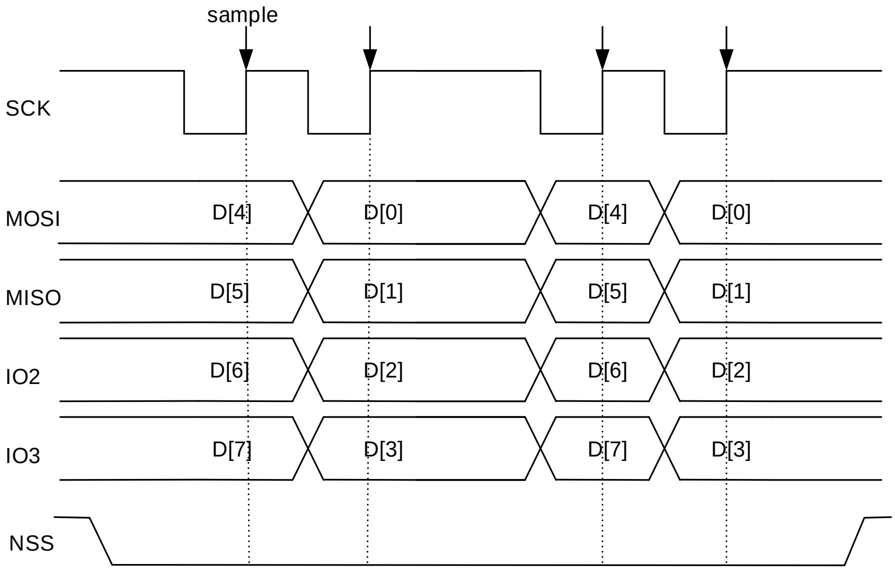


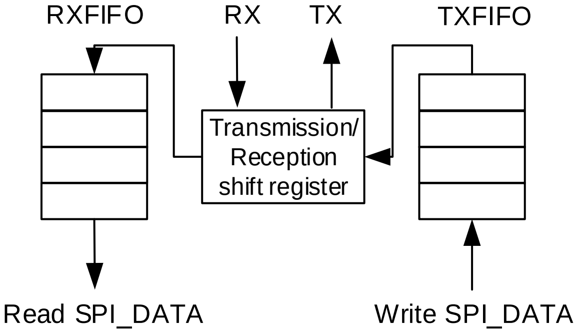
Figure 17-5. SPI1 timing diagram in Quad-SPI mode (CKPL=1, CKPH=1, LF=0)



17.3.4. Separate transmission and reception FIFO

The separate 32-bit reception FIFO (RXFIFO) and transmission FIFO (TXFIFO) are used in different directions for SPI data transactions, and they can enable the SPI to work in a continuous flow (only available in SPI1).

Figure 17-6. Transmission and reception FIFO



When the current TXFIFO level is less than or equal to half of its capacity, the TXFIFO is considered empty(1) and TBE is set to 1 by hardware at this time. When the TBE bit is set, writing data to the SPI_DATA register will store the data at the end of the TXFIFO. Hardware sets the RBNE bit when the RXFIFO is considered non-empty(2). When the RBNE bit is set, reading data from the SPI_DATA register will get the oldest data from the RXFIFO.

Note: (1) For SPI1, the TXFIFO empty means that the TXFIFO level is less than or equal to half of its capacity. The meaning of TXFIFO full is the opposite. Therefore, when the data frame format is not greater than 8 bits, the TXFIFO can store up to three data frames. If the TXFIFO empty or full appears below and there is no special explanation, the meaning is the same as that described here.

(2) For SPI1, the meaning of RXFIFO empty is divided into the following two conditions: If BYTEN bit in SPI_CTL1 is set, the RXFIFO empty means the RXFIFO level is less than quarter of its capacity. At this time, when the data frame format is not more than 8 bits, the RXFIFO can store up to 4 data frames. If BYTEN is cleared, the RXFIFO empty means the RXFIFO level is less than half of its capacity. The meaning of RXFIFO full is the opposite. If the RXFIFO empty or full appears below and there is no special explanation, the meaning is the same as that described here.

Data merging (Only for SPI1)

When DZ[3:0] in the SPI_CTL1 register configures the transmission data bit width to be 8 bits or less than 8 bits, by configuring the BYTEN bit in the SPI_CTL1 register to 0, the data merge transmission mode function is enabled. When DZ[3:0] in the configuration SPI_CTL1 register configures the transmission data bit width to be less than or equal to 8 bits, this function can realize that when 16-bit write access is performed to the SPI_DATA register, two data frames are sent in parallel instead of serial line method.

Similarly, at the receiving end, the receiver obtains these two data frames through a 16-bit read access to SPI_DATA, and only one RBNE event will be generated when the two frames of data are received.

Note: when an odd number of data bytes will be transferred, on the transmitter side, writing the last data frame of any odd sequence with an 8-bit access to SPI_DATA is enough. The receiver has to change BYTEN for the last data frame received in the odd sequence of frames in order to generate the RBNE event.

17.3.5. NSS function

Slave mode

When slave mode is configured (MSTMOD=0), SPI gets NSS level from NSS pin in hardware NSS mode (SWNSSEN = 0) or from SWNSS bit in software NSS mode (SWNSSEN = 1) and transmits/receives data only when NSS level is low. In software NSS mode, NSS pin is not used.

Table 17-3. NSS function in slave mode

Mode	Register configuration	Description

Slave hardware NSS mode	MSTMOD = 0
	SWNSSEN = 0
		SPI slave gets NSS level from NSS pin.

Slave software NSS mode	MSTMOD = 0
	SWNSSEN = 1
		SPI slave NSS level is determined by the SWNSS bit.
		SWNSS = 0: NSS level is low
		SWNSS = 1: NSS level is high

Master mode

In master mode (MSTMOD=1) if the application uses multi-master connection, NSS can be configured to hardware input mode (SWNSSEN=0, NSSDRV=0) or software mode (SWNSSEN=1). Then, once the NSS pin (in hardware NSS mode) or the SWNSS bit (in software NSS mode) goes low, the SPI automatically enters slave mode and triggers a master fault flag CONFERR.

If the application wants to use NSS line to control the SPI slave, NSS should be configured to hardware output mode (SWNSSEN=0, NSSDRV=1). NSS stays high after SPI is enabled and goes low when transmission or reception process begins. When SPI is disabled, the NSS goes high.

The application may also use a general purpose IO as NSS pin to realize more flexible NSS.

Table 17-4. NSS function in master mode

Mode	Register configuration	Description

Master hardware NSS output mode	MSTMOD = 1
	SWNSSEN = 0
	NSSDRV=1
		Applicable to single-master mode. The master uses the NSS pin to control the SPI slave device. At this time, the NSS is configured as the hardware output mode. NSS goes low after enabling SPI.

Master hardware NSS input mode	MSTMOD = 1
	SWNSSEN = 0
	NSSDRV=0
		Applicable to multi-master mode. At this time, NSS is configured as hardware input mode. Once the NSS pin is pulled low, SPI will automatically enter slave mode, and a master configuration error will occur and the CONFERR bit will be set to 1.

Master software NSS mode	MSTMOD = 1
	SWNSSEN = 1
	SWNSS = 0
	NSSDRV: Don’t care
		Applicable to multi-master mode. Once SWNSS = 0, SPI will automatically enter slave mode, and a master configuration error will occur and the CONFERR bit will be 1.

	MSTMOD = 1
	SWNSSEN = 1
	SWNSS = 1
	NSSDRV: Don’t care
		The slave can use hardware or software NSS mode.

17.3.6. SPI operation modes

Table 17-5. SPI operation modes

Mode	Description	Register configuration	Data pin usage
MFD	Master full-duplex	MSTMOD = 1	MOSI: Transmission

Mode	Description	Register configuration	Data pin usage

		RO = 0
		BDEN = 0
		BDOEN: Don't care
			MISO: Reception

MTU	Master transmission with unidirectional connection	MSTMOD = 1
		RO = 0
		BDEN = 0
		BDOEN: Don't care
			MOSI: Transmission
			MISO: Not used

MRU	Master reception with unidirectional connection	MSTMOD = 1
		RO = 1
		BDEN = 0
		BDOEN: Don't care
			MOSI: Not used
			MISO: Reception

MTB	Master transmission with bidirectional connection	MSTMOD = 1
		RO = 0
		BDEN = 1
		BDOEN = 1
			MOSI: Transmission
			MISO: Not used

MRB	Master reception with bidirectional connection	MSTMOD = 1
		RO = 0
		BDEN = 1
		BDOEN = 0
			MOSI: Reception
			MISO: Not used

SFD	Slave full-duplex	MSTMOD = 0
		RO = 0
		BDEN = 0
		BDOEN: Don't care
			MOSI: Reception
			MISO: Transmission

STU	Slave transmission with unidirectional connection	MSTMOD = 0
		RO = 0
		BDEN = 0
		BDOEN: Don't care
			MOSI: Not used
			MISO: Transmission

SRU	Slave reception with unidirectional connection	MSTMOD = 0
		RO = 1
		BDEN = 0
		BDOEN: Don't care
			MOSI: Reception
			MISO: Not used

STB	Slave transmission with bidirectional connection	MSTMOD = 0
		RO = 0
		BDEN = 1
		BDOEN = 1
			MOSI: Not used
			MISO: Transmission

SRB	Slave reception with bidirectional connection	MSTMOD = 0
		RO = 0
		BDEN = 1
		BDOEN = 0
			MOSI: Not used
			MISO: Reception

Figure 17-7. A typical full-duplex connection

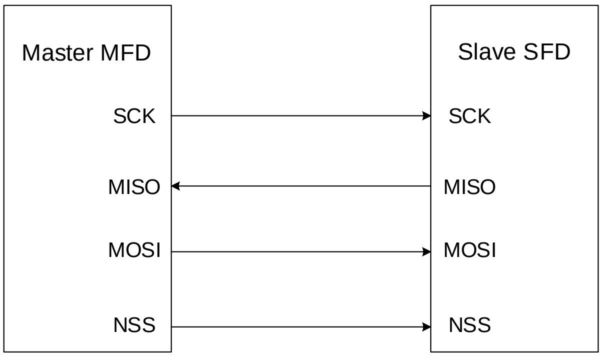


Figure 17-8. A typical simplex connection (Master: Receive, Slave: Transmit)

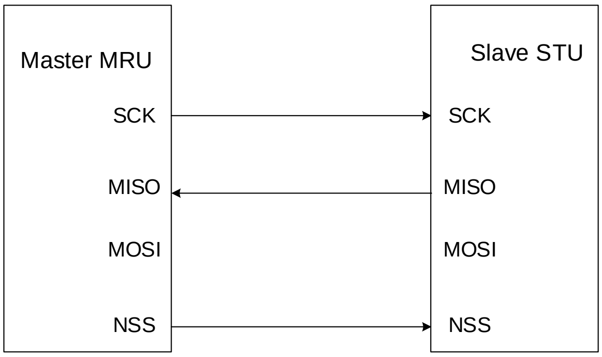


Figure 17-9. A typical simplex connection (Master: Transmit only, Slave: Receive)

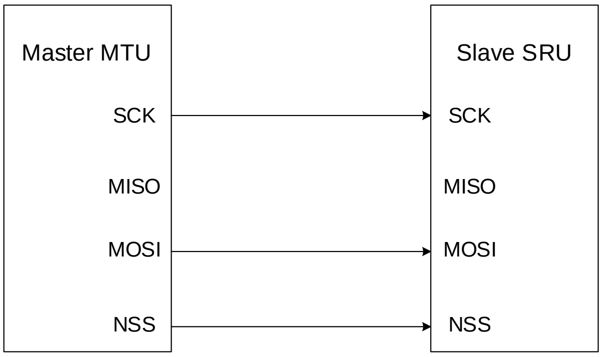
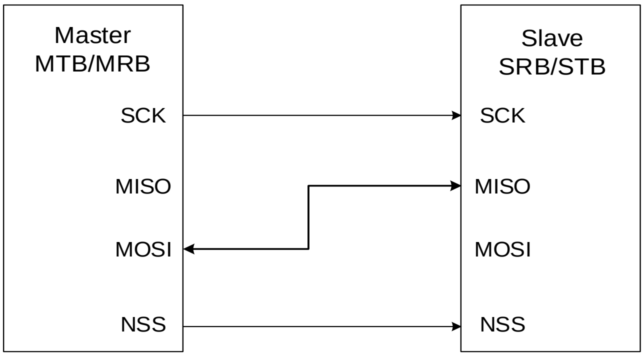


Figure 17-10. A typical bidirectional connection



Initialization sequence

SPI1:

Before transmitting or receiving data, application should follow the SPI initialization sequence described below:

1. If master mode or slave TI mode is used, program the PSC [2:0] bits in SPI_CTL0 register to generate SCK with desired baud rate or configure the Td time in TI mode, otherwise, ignore this step.
2. Program the clock timing register (CKPL and CKPH bits in the SPI_CTL0 register).
3. Program the frame format (LF bit in the SPI_CTL0 register).
4. Program data format (DZ bits in the SPI_CTL1 register) and the access size for the SPI_DATA register (BYTEN bit in the SPI_CTL1 register).
5. Program the NSS mode (SWNSSEN and NSSDRV bits in the SPI_CTL0 register) according to the application's demand as described above in ***NSS function*** section.
6. If TI mode is used, set TMOD bit in SPI_CTL1 register, otherwise, ignore this step.
7. If NSSP mode is used, set NSSP bit in SPI_CTL1 register, otherwise, ignore this step.
8. Configure MSTMOD, RO, BDEN and BDOEN depending on the operation modes described in ***SPI operation modes*** section.
9. Initialize TXDMA_ODD/RXDMA_ODD bits if they are needed when DMA is used in packed mode.
10. If Quad-SPI mode is used, set the QMOD bit in SPI_QCTL register. Ignore this step if Quad-SPI mode is not used.
11. Enable the SPI (set the SPIEN bit).

Note: During communication, CKPH, CKPL, MSTMOD, PSC[2:0], LF and DZ[3:0] bits should not be changed.

SPI0:

Before transmitting or receiving data, application should follow the SPI initialization sequence described below:

1. If master mode or slave TI mode is used, program the PSC [2:0] bits in SPI_CTL0 register to generate SCK with desired baud rate or configure the Td time in TI mode, otherwise, ignore this step.
2. Program data format (FF16 bit in the SPI_CTL0 register).
3. Program the clock timing register (CKPL and CKPH bits in the SPI_CTL0 register).
4. Program the frame format (LF bit in the SPI_CTL0 register).
5. Program the NSS mode (SWNSSEN and NSSDRV bits in the SPI_CTL0 register) according to the application's demand as described above in ***NSS function*** section.
6. If TI mode is used, set TMOD bit in SPI_CTL1 register, otherwise, ignore this step.
7. If NSSP mode is used, set NSSP bit in SPI_CTL1 register, otherwise, ignore this step.
8. Configure MSTMOD, RO, BDEN and BDOEN depending on the operating modes described in ***SPI operation modes*** section.
9. Enable the SPI (set the SPIEN bit).

Note: During communication, CKPH, CKPL, MSTMOD, PSC[2:0] and LF bits should not be changed.

Basic transmission and reception sequence

Transmission sequence

After the initialization sequence, the SPI is enabled and stays at idle state. In master mode, the transmission starts when the application writes a data into the transmit buffer/TXFIFO. In slave mode the transmission starts when SCK clock signal begins to toggle at SCK pin and NSS level is low, so application should ensure that data is already written into transmit buffer/TXFIFO before the transmission starts in slave mode.

When SPI begins to send a data frame, it first loads this data frame from the data buffer/TXFIFO to the shift register and then begins to transmit the loaded data frame. After TBE flag is set, which means the transmit buffer/TXFIFO is empty, the application should write SPI_DATA register again if it has more data to transmit.

In master mode, software should write the next data into SPI_DATA register before the transmission of current data frame is completed if it desires to generate continuous transmission.

Reception sequence

After the last valid sample clock, the incoming data will be moved from shift register to the receive buffer/RXFIFO and RBNE will be set. The application should read SPI_DATA register to get the received data and this will clear the RBNE flag automatically when receive buffer/RXFIFO is empty. In MRU and MRB modes, hardware continuously sends clock signal to receive the next data frame, while in full-duplex master mode (MFD), hardware only receives the next data frame when the transmit buffer/TXFIFO is not empty.

SPI operation sequence in different modes (Not Quad-SPI, TI mode or NSSP mode)

In full-duplex mode, either MFD or SFD, the RBNE and TBE flags should be monitored and then follow the sequences described above.

The transmission mode (MTU, MTB, STU or STB) is similar to the transmission sequence of full-duplex mode regardless of the RBNE and OVRE bits.

The master reception mode (MRU or MRB) is different from the reception sequence of full-duplex mode. In MRU or MRB mode, after SPI is enabled, the SPI continuously generates SCK until the SPI is disabled. So the application should ignore the TBE flag and read out reception buffer/RXFIFO in time after the RBNE flag is set, otherwise a data overrun fault will occur.

The slave reception mode (SRU or SRB) is similar to the reception sequence of full-duplex mode regardless of the TBE flag.

SPI TI mode

SPI TI mode takes NSS as a special frame header flag signal and its operation sequence is similar to normal mode described above. The modes described above (MFD, MTU, MRU, MTB, MRB, SFD, STU, SRU, STB and SRB) are still supported in TI mode. While, in TI mode the CKPL and CKPH bits in SPI_CTL0 registers take no effect and the SCK sample edge is falling edge.

Figure 17-11. Timing diagram of TI master mode with discontinuous transfer

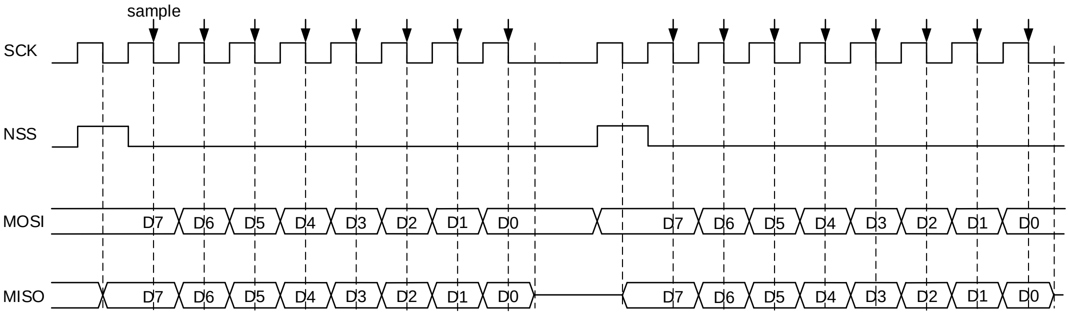
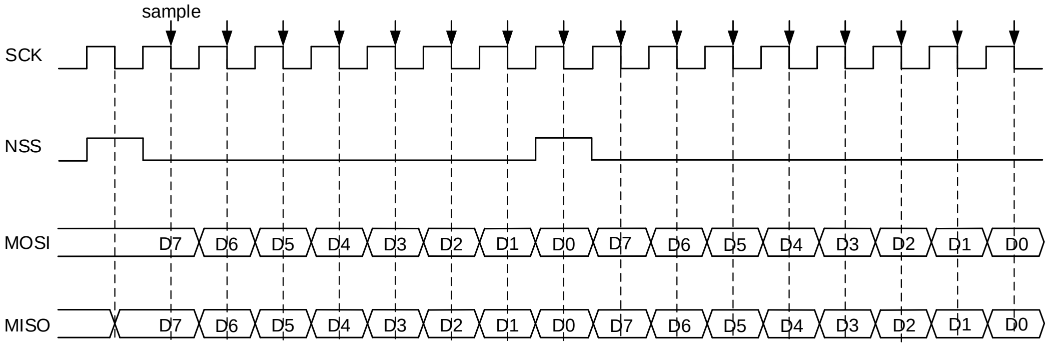
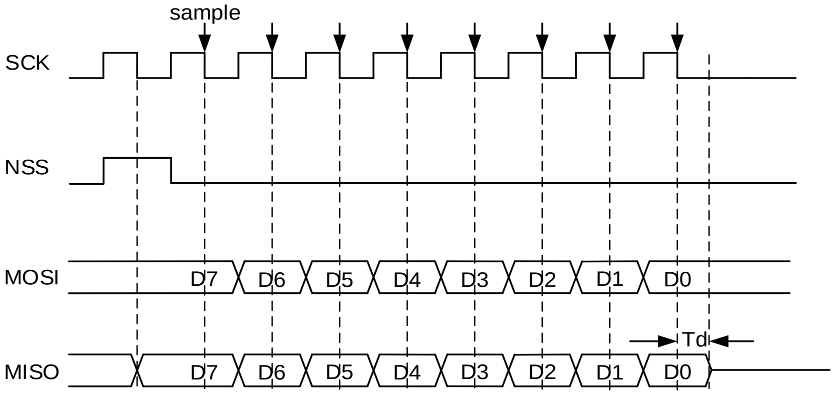


Figure 17-12. Timing diagram of TI master mode with continuous transfer



In master TI mode, SPI can perform continuous or non-continuous transfer. If the master writes SPI_DATA register fast enough, the transfer is continuous, otherwise non-continuous. In non-continuous transfer there is an extra header clock cycle before each byte. While in continuous transfer, the extra header clock cycle only exists before the first byte and the following bytes' header clock is overlaid at the last bit of pervious bytes.

Figure 17-13. Timing diagram of TI slave mode



In slave TI mode, after the last rising edge of SCK in transfer, the slave begins to transmit the LSB bit of the last data byte, and after a half-bit time, the master begins to sample the line. To make sure that the master samples the right value, the slave should continue to drive this bit after the falling sample edge of SCK for a period of time before releasing the pin. This time is called T_d. T_d is decided by PSC [2:0] bits in SPI_CTL0 register.

$$T_d=\frac{T_{bit}}{2}+5*T_{pclk} \qquad (23\text{-}1)$$

For example, if PSC [2:0] = 010, $T_d$is 9*Tpclk.

In slave mode, the slave also monitors the NSS signal and sets an error flag FERR if it detects an incorrect NSS behavior, for example: toggles at the middle bit of a byte.

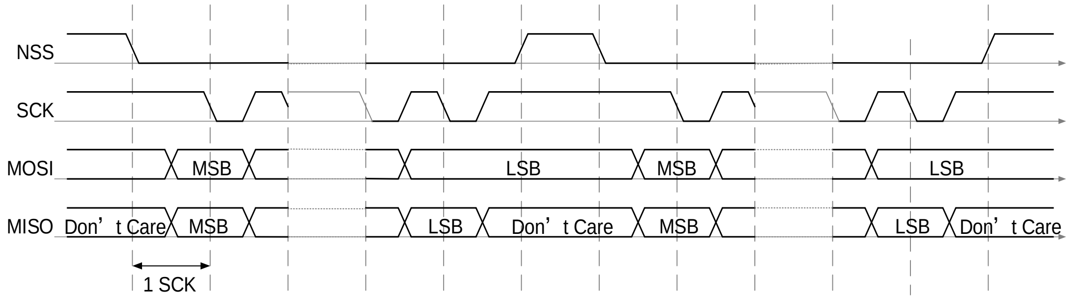
NSS pulse mode operation sequence

This function is controlled by NSSP bit in SPI_CTL1 register. In order to implement this function, several additional conditions must be met: configure the device to master mode, frame format should follow the normal SPI protocol, select the first clock transition as the data capture edge.

In summary, MSTMOD = 1, NSSP = 1, CKPH = 0.

When NSS pulse mode is enabled, a pulse duration of at least 1 SCK clock period is inserted between two successive data frames depending on the status of internal data transmit buffer/TXFIFO. Multiple SCK clock cycle intervals are possible if the transfer buffer/TXFIFO stays empty. This function is designed for single master-slave configuration for the slave to latch data. The following diagram depicts its timing diagram.

Figure 17-14. Timing diagram of NSS pulse with continuous transmit



Quad-SPI mode operation sequence

The Quad-SPI mode is designed to control Quad-SPI flash.

In order to enter Quad-SPI mode, the software should first verify that the TBE bit is set and TRANS bit is cleared, then set QMOD bit in SPI_QCTL register. In Quad-SPI mode, BDEN, BDOEN, CRCEN, CRCNT, CRCL, RO and LF in SPI_CTL0 register should be kept cleared and DZ[3:0] should be set to ensure that SPI data size is 8-bit, MSTMOD should be set to ensure that SPI is in master mode. SPIEN, PSC, CKPL and CKPH should be configured as desired.

There are two operation modes in Quad-SPI mode: quad write and quad read, decided by QRD bit in SPI_QCTL register.

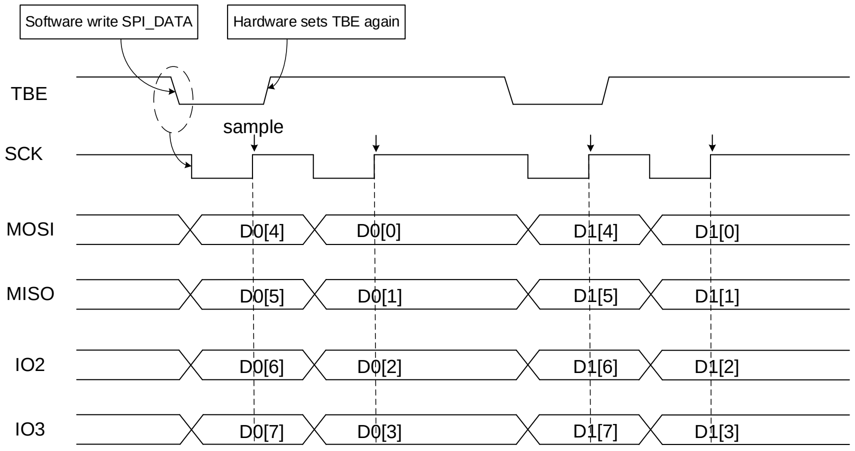
Quad write operation

SPI works in quad write mode when QMOD is set and QRD is cleared in SPI_QCTL register. In this mode, MOSI, MISO, IO2 and IO3 are all used as output pins. SPI begins to generate clock on SCK line and transmit data on MOSI, MISO, IO2 and IO3 as soon as data is written into SPI_DATA (TBE is cleared) and SPIEN is set. Once SPI starts transmission, it always checks TBE status at the end of a frame and stops when condition is not met.

The operation flow for transmitting in quad mode:

1. Configure clock prescaler, clock polarity, phase, etc. in SPI_CTL0 and SPI_CTL1 based on your application requirements.
2. Set QMOD bit in SPI_QCTL register and then enable SPI by setting SPIEN in SPI_CTL0.
3. Write a byte to SPI_DATA register and the TBE will be cleared.
4. Wait until TBE is set by hardware again before writing the next byte.

Figure 17-15. Timing diagram of quad write operation in Quad-SPI mode



Quad read operation

SPI works in quad read mode when QMOD and QRD are both set in SPI_QCTL register. In this mode, MOSI, MISO, IO2 and IO3 are all used as input pins. SPI begins to generate clock on SCK line as soon as a data is written into SPI_DATA (TBE is cleared) and SPIEN is set. Writing data into SPI_DATA is only to generate SCK clocks, so the written data can be any value. Once SPI starts transmission, it always checks SPIEN and TBE status at the end of a frame and stops when condition is not met. So, dummy data should always be written into SPI_DATA to generate SCK.

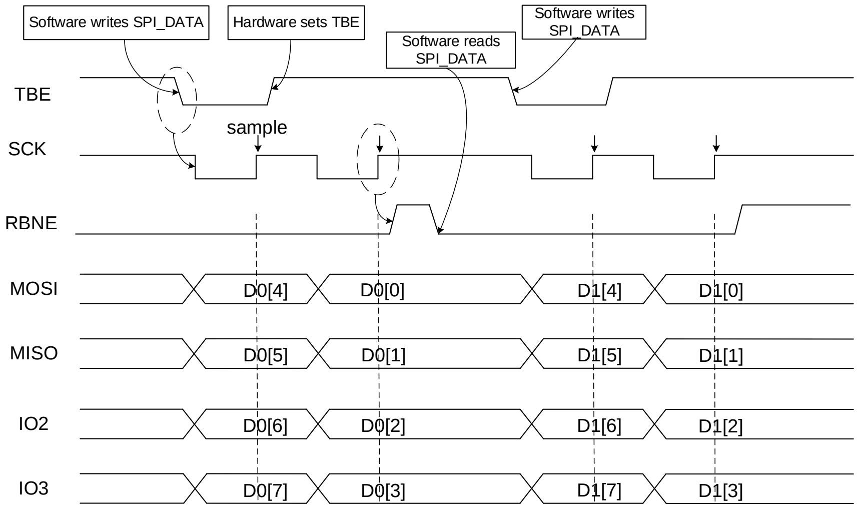
The operation flow for receiving in quad mode is shown below:

1. Configure clock prescaler, clock polarity, phase, etc. in SPI_CTL0 and SPI_CTL1 register based on your application requirements.
2. Set QMOD and QRD bits in SPI_QCTL register and then enable SPI by setting SPIEN

in SPI_CTL0 register.

3. Write an arbitrary byte (for example, 0xFF) to SPI_DATA register.
4. Wait until the RBNE flag is set and read SPI_DATA to get the received byte.
5. Write an arbitrary byte (for example, 0xFF) to SPI_DATA to receive the next byte.

Figure 17-16. Timing diagram of quad read operation in Quad-SPI mode



SPI disabling sequence

Different sequences are used to disable the SPI in different operation modes:

MFD SFD

For SPI1, wait until TXLVL[1:0]=00 and confirm TRANS=0. Then disable the SPI by clearing SPIEN bit. At last, read data until RXTVL[1:0]=00.

For SPI0, wait for the last RBNE flag and then receive the last data. Confirm that TBE=1 and TRANS=0. At last, disable the SPI by clearing SPIEN bit.

MTU MTB STU STB

For SPI1, wait until TXLVL[1:0]=00 and confirm TRANS=0. Then disable the SPI by clearing SPIEN bit.

For SPI0, write the last data into SPI_DATA and wait until the TBE flag is set and then wait until the TRANS flag is cleared. Disable the SPI by clearing SPIEN bit.

MRU MRB

For SPI1, application can disable the SPI when it doesn’t want to receive data, and then confirm the TRANS=0 and read data until RXLVL[1:0]=00.

For SPI0, after getting the second last RBNE flag, read out this data and delay for a SCK clock time and then, disable the SPI by clearing SPIEN bit. Wait until the last RBNE flag is set and read out the last data.

SRU SRB

For SPI1, application can disable the SPI when it doesn't want to receive data, and then confirm the TRANS=0 and read data until RXLVL[1:0]=00.

For SPI0, application can disable the SPI when it doesn't want to receive data, and then wait until the TRANS=0 to ensure the ongoing transfer completes.

TI mode

The disabling sequence of TI mode is the same as the sequences described above.

NSS pulse mode

The disabling sequence of NSSP mode is the same as the sequences described above.

Quad-SPI mode

Before leaving quad wire mode or disabling SPI, software should first check that TBE bit is set and TRANS bit is cleared, then the QMOD bit in SPI_QCTL register and SPIEN bit in SPI_CTL0 register are cleared.

17.3.7. DMA function

The DMA frees the application from data writing and reading process during transfer, to improve the system efficiency.

DMA function in SPI is enabled by setting DMATEN and DMAREN bits in SPI_CTL1 register. To use DMA function, application should first correctly configure DMA modules, then configure SPI module according to the initialization sequence, at last enable SPI.

After being enabled, If DMATEN is set, SPI will generate a DMA request each time when TBE=1, then DMA will acknowledge to this request and write data into the SPI_DATA register automatically. If DMAREN is set, SPI will generate a DMA request each time when RBNE=1, then DMA will acknowledge to this request and read data from the SPI_DATA register automatically.

Data merging with DMA (Only for SPI1)

In the case of using DMA for data transmission, when BYTEN is set to 0 and the data length configured by DZ[3:0] is less than or equal to 8 bits and the data merging mode is enabled, the DMA will access the SPI_DATA register in 16-bit mode, automatically Complete the data transmission.

In the case that the data packetization mode is enabled and the frame number of the data frame is not an even multiple, in order to avoid the problem of one more frame of data in the last DMA transmission, the TXDMA_ODD/RXDMA_ODD bit in the SPI_CTL1 register needs to be set to 1

17.3.8. CRC function

There are two CRC calculators in SPI: one for transmission and the other for reception. The CRC calculation uses the polynomial defined in SPI_CRCPOLY register.

Application can enable the CRC function by setting CRCEN bit in SPI_CTL0 register. The CRC calculators continuously calculate CRC for each bit transmitted and received on lines, and the calculated CRC values can be read from SPI_TCRC and SPI_RCRC registers.

To transmit the calculated CRC value, application should set the CRCNT bit in SPI_CTL0 register after the last data is written to the transmit buffer/TXFIFO. In full-duplex mode (MFD or SFD), when the SPI transmits a CRC and prepares to check the received CRC value, the SPI treats the incoming data as a CRC value. In reception mode (MRB, MRU, SRU and SRB), the application should set the CRCNT bit after the second last data frame is received. When CRC checking fails, the CRCERR flag will be set.

For SPI0, if DMA function is enabled, application doesn't need to operate CRCNT bit and hardware will automatically process the CRC transmitting and checking.

For SPI1, a CRC-format transaction usually takes one more data frame to communicate at the end of data sequence. However, when setting an 8-bit data frame checked by 16-bit CRC, two more frames are necessary to send the complete CRC. If DMA function is enabled, the counter for the SPI transmission DMA channel has to be set to the number of data frames to transmit excluding the CRC frame. On the receiver side, the DMA counter should be configured as follows:

1. Full duplex mode: Suppose the amount of data received by SPI is L, when CRCL = 0 and DZ = 8, then the count of the DMA receive channel is L + 1, otherwise the count of the DMA receive channel is L + 2.

2. Receive only mode: DMA receive channel count is only equal to the amount of data received .After receiving data, the CRC value is obtained by reading SPI_RCRC register by software.

Note: When SPI is in slave mode and CRC function is enable, the CRC calculator is sensitive to input SCK clock whether SPI is enable or not. The software must enable CRC only when the clock is stable to avoid wrong CRC calculation. And when SPI works as a slave, the NSS internal signal needs to be kept low between the data phase and CRC phase.

17.3.9. SPI interrupts

Status flags

- **Transmit buffer/TXFIFO empty flag (TBE)**

This bit is set when the transmit buffer is empty or the TXFIFO level is lower or equal to 1/2 of FIFO depth, the software can write the next data to the transmit buffer/TXFIFO by writing the SPI_DATA register.

- **Receive buffer/RXFIFO not empty flag (RBNE)**

For SPI1, this bit is set depending on the BYTEN bit in the SPI_CTL1: If BYTEN = 0, the RBNE is set when the RXFIFO level is greater or equal to 1/2(16-bit). If BYTEN = 1, the RBNE is set when the RXFIFO level is greater or equal to 1/4(8-bit).

For SPI0, this bit is set when receive buffer is not empty, which means that one data is received and stored in the receive buffer, and software can read the data by reading the SPI_DATA register.

- **SPI transmitting ongoing flag (TRANS)**

TRANS is a status flag to indicate whether the transfer is ongoing or not. It is set and cleared by hardware and not controlled by software. This flag doesn't generate any interrupt.

Error conditions

- **Configuration fault error (CONFERR)**

CONFERR is an error flag in master mode. In NSS hardware mode and the NSSDRV is not enabled, the CONFERR is set when the NSS pin is pulled low. In NSS software mode, the CONFERR is set when the SWNSS bit is 0. When the CONFERR is set, the SPIEN bit and the MSTMOD bit are cleared by hardware, the SPI is disabled and the device is forced into slave mode.

The SPIEN and MSTMOD bit are write protection until the CONFERR is cleared. The CONFERR bit of the slave cannot be set. In a multi-master configuration, the device can be in slave mode with CONFERR bit set, which means there might have been a multi-master conflict for system control.

- **Rx overrun error (RXORERR)**

The RXORERR bit is set if a data is received when the RBNE is set. For SPI0, that means the last data has not been read out and the newly incoming data is received. For SPI1, that means the RXFIFO has not enough space to store this received data. The receive buffer/RXFIFO contents won't be covered with the newly incoming data, so the newly incoming data is lost.

- **Format error (FERR)**

In slave TI mode, the slave also monitors the NSS signal and set an error flag if it detects an incorrect NSS behavior, for example: toggles at the middle bit of a byte.

- **CRC error (CRCERR)**

When the CRCEN bit is set, the CRC calculation result of the received data in the SPI_RCRC register is compared with the received CRC value after the last data, the CRCERR is set when they are different.

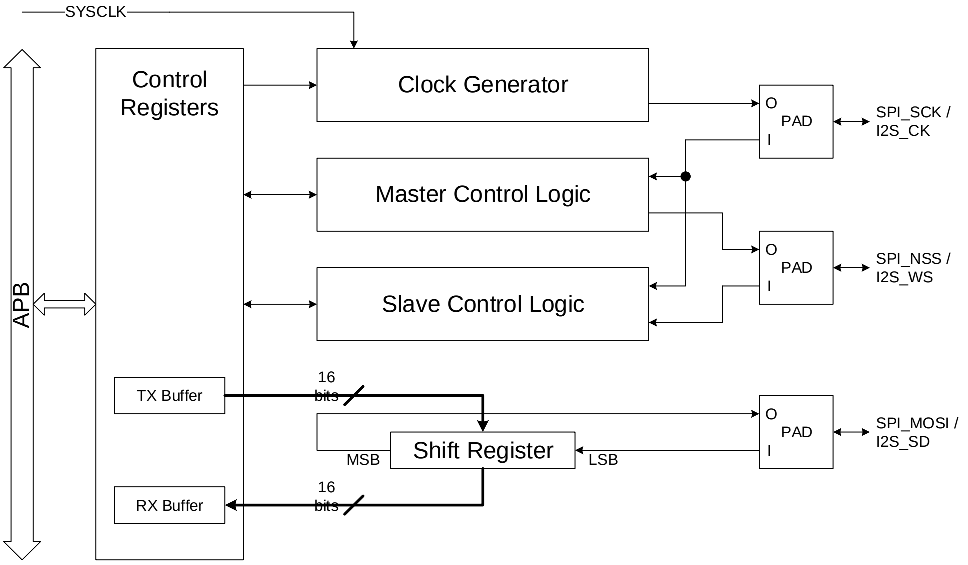
Table 17-6. SPI interrupt requests

Flag	Description	Clear method	Interrupt enable bit
TBE	Transmit buffer/TXFIFO empty	Write SPI_DATA register.	TBEIE
RBNE	Receive buffer/RXFIFO not empty	Read SPI_DATA register.	RBNEIE
CONFERR	Configuration fault error	Read or write SPI_STAT register, then write SPI_CTL0 register.	ERRIE
RXORERR	Rx overrun error	Read SPI_DATA register, then read SPI_STAT register.	
CRCERR	CRC error	Write 0 to CRCERR bit	
FERR	TI mode format error	Write 0 to FERR bit	

17.4. I2S function overview

17.4.1. I2S block diagram

Figure 17-17. Block diagram of I2S



There are five sub modules to support I2S function, including control registers, clock generator, master control logic, slave control logic and shift register. All the user configuration registers are implemented in the control registers module, including the TX buffer and RX buffer. The clock generator is used to produce I2S communication clock in master mode. The master control logic is implemented to generate the I2S_WS signal and control the communication in master mode. The slave control logic is implemented to control the communication in slave mode according to the received I2SCK and I2S_WS. The shift register handles the serial data transmission and reception on I2S_SD.

17.4.2. I2S signal description

There are three pins on the I2S interface, including I2S_CK, I2S_WS and I2S_SD. I2S_CK is the serial clock signal, which shares the same pin with SPI_SCK. I2S_WS is the frame control signal, which shares the same pin with SPI_NSS. I2S_SD is the serial data signal, which shares the same pin with SPI_MOSI.

17.4.3. I2S audio standards

The I2S audio standard is selected by the I2SSTD bits in the SPI_I2SCTL register. Four audio standards are supported, including I2S Phillips standard, MSB justified standard, LSB justified standard, and PCM standard. All standards except PCM handle audio data time-multiplexed on two channels (the left channel and the right channel). For these standards, the I2S_WS signal indicates the channel side. For PCM standard, the I2S_WS signal indicates frame synchronization information.

The data length and the channel length are configured by the DTLEN bits and CHLEN bit in the SPI_I2SCTL register. Since the channel length must be greater than or equal to the data length, four packet types are available. They are 16-bit data packed in 16-bit frame, 16-bit data packed in 32-bit frame, 24-bit data packed in 32-bit frame, and 32-bit data packed in 32-bit frame. The data buffer for transmission and reception is 16-bit wide. In the case that the data length is 24 bits or 32 bits, two write or read operations to or from the SPI_DATA register are needed to complete the transmission of a frame. In the case that the data length is 16 bits, only one write or read operation to or from the SPI_DATA register is needed to complete the transmission of a frame. When using 16-bit data packed in 32-bit frame, 16-bit 0 is inserted by hardware automatically to extend the data to 32-bit format.

For all standards and packet types, the most significant bit (MSB) is always sent first. For all standards based on two channels time-multiplexed, the channel left is always sent first followed by the channel right.

I2S Phillips standard

For I2S Phillips standard, I2S_WS and I2S_SD are updated on the falling edge of I2S_CK. The timing diagrams for each configuration are shown below.

Figure 17-18. I2S Phillips standard timing diagram (DTLEN=00, CHLEN=0, CKPL=0)

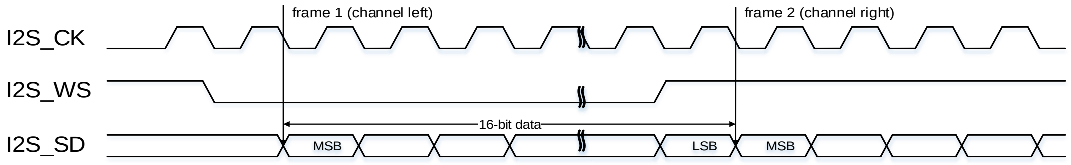
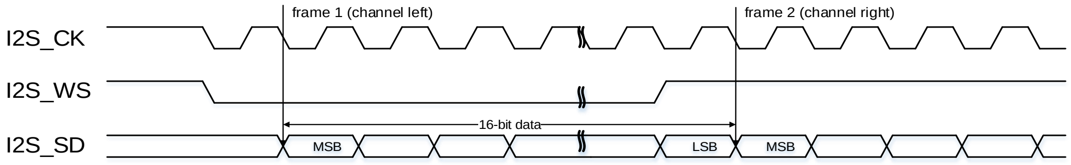


Figure 17-19. I2S Phillips standard timing diagram (DTLEN=00, CHLEN=0, CKPL=1)



When the packet type is 16-bit data packed in 16-bit frame, only one write or read operation to or from the SPI_DATA register is needed to complete the transmission of a frame.

Figure 17-20. I2S Phillips standard timing diagram (DTLEN=10, CHLEN=1, CKPL=0)

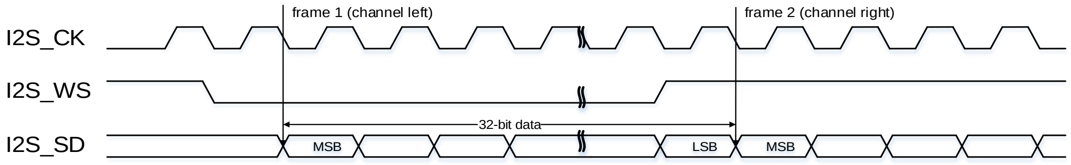
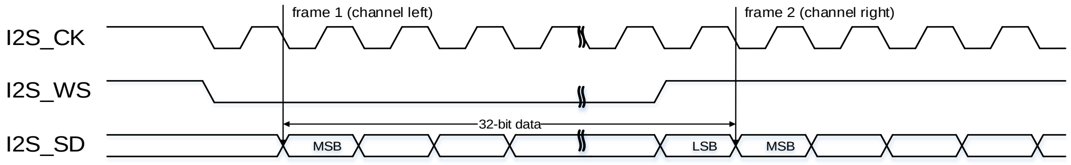


Figure 17-21. I2S Phillips standard timing diagram (DTLEN=10, CHLEN=1, CKPL=1)



When the packet type is 32-bit data packed in 32-bit frame, two write or read operations to or from the SPI_DATA register are needed to complete the transmission of a frame. In transmission mode, if a 32-bit data is going to be sent, the first data written to the SPI_DATA register should be the higher 16 bits, and the second one should be the lower 16 bits. In reception mode, if a 32-bit data is received, the first data read from the SPI_DATA register should be higher 16 bits, and the second one should be the lower 16 bits.

Figure 17-22. I2S Phillips standard timing diagram (DTLEN=01, CHLEN=1, CKPL=0)

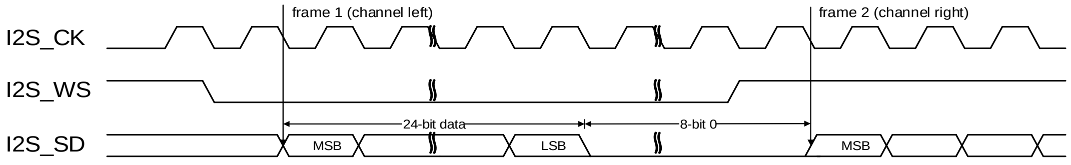
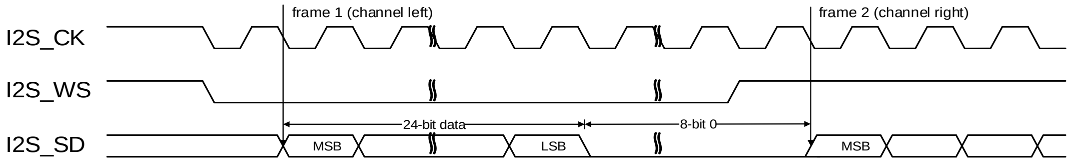


Figure 17-23. I2S Phillips standard timing diagram (DTLEN=01, CHLEN=1, CKPL=1)



When the packet type is 24-bit data packed in 32-bit frame, two write or read operations to or from the SPI_DATA register are needed to complete a frame. In transmission mode, if a 24-bit data D[23:0] is going to be sent, the first data written to the SPI_DATA register should be the higher 16 bits: D[23:8], and the second one should be a 16-bit data. The higher 8 bits of this 16-bit data should be D[7:0] and the lower 8 bits can be any value. In reception mode, if a 24-bit data D[23:0] is received, the first data read from the SPI_DATA register is D[23:8], and the second one is a 16-bit data. The higher 8 bits of this 16-bit data are D[7:0] and the lower 8 bits are zeros.

Figure 17-24. I2S Phillips standard timing diagram (DTLEN=00, CHLEN=1, CKPL=0)

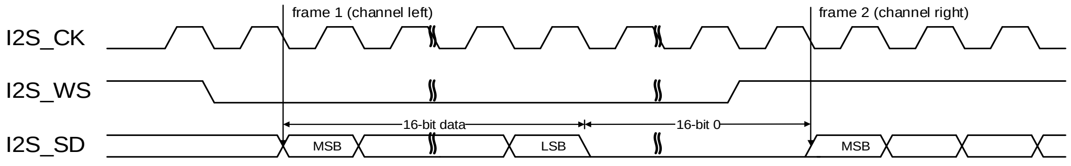
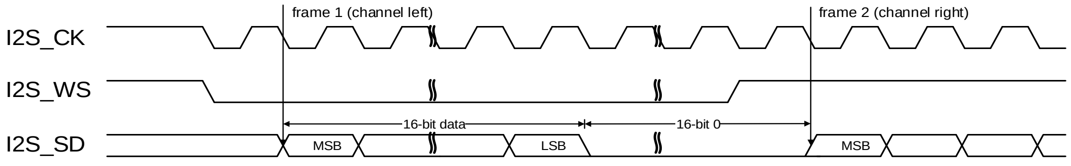


Figure 17-25. I2S Phillips standard timing diagram (DTLEN=00, CHLEN=1, CKPL=1)



When the packet type is 16-bit data packed in 32-bit frame, only one write or read operation to or from the SPI_DATA register is needed to complete the transmission of a frame. The remaining 16 bits are forced by hardware to 0x0000 to extend the data to 32-bit format.

MSB justified standard

For MSB justified standard, I2S_WS and I2S_SD are updated on the falling edge of I2S_CK. The SPI_DATA register is handled in the exactly same way as that for I2S Phillips standard. The timing diagrams for each configuration are shown below.

Figure 17-26. MSB justified standard timing diagram (DTLEN=00, CHLEN=0, CKPL=0)

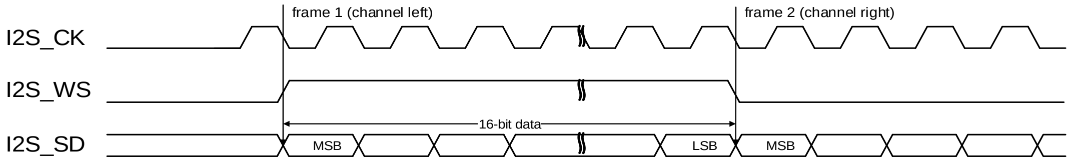


Figure 17-27. MSB justified standard timing diagram (DTLEN=00, CHLEN=0, CKPL=1)

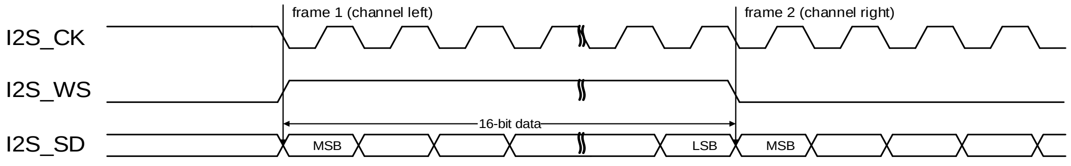


Figure 17-28. MSB justified standard timing diagram (DTLEN=10, CHLEN=1, CKPL=0)

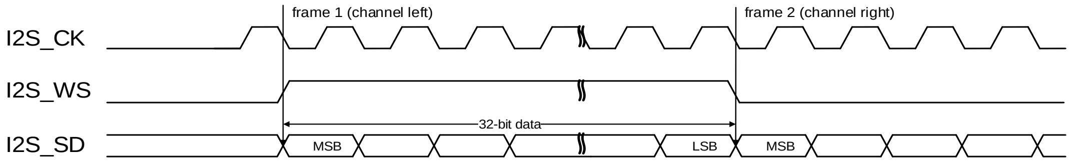


Figure 17-29. MSB justified standard timing diagram (DTLEN=10, CHLEN=1, CKPL=1)

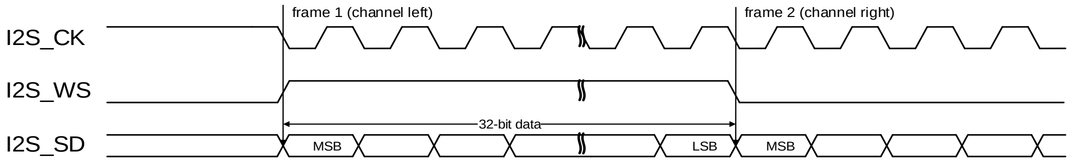


Figure 17-30. MSB justified standard timing diagram (DTLEN=01, CHLEN=1, CKPL=0)

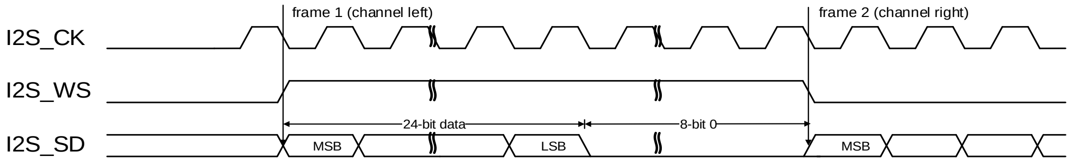


Figure 17-31. MSB justified standard timing diagram (DTLEN=01, CHLEN=1, CKPL=1)

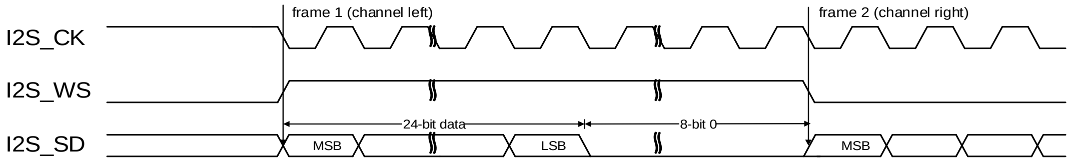


Figure 17-32. MSB justified standard timing diagram (DTLEN=00, CHLEN=1, CKPL=0)

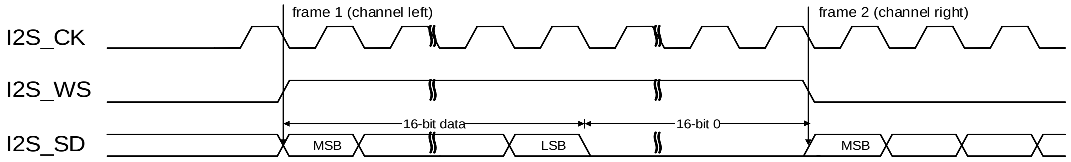
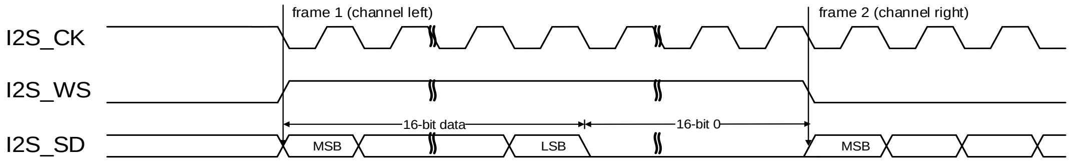


Figure 17-33. MSB justified standard timing diagram (DTLEN=00, CHLEN=1, CKPL=1)



LSB justified standard

For LSB justified standard, I2S_WS and I2S_SD are updated on the falling edge of I2S_CK. In the case that the channel length is equal to the data length, LSB justified standard and MSB justified standard are exactly the same. In the case that the channel length is greater than the data length, the valid data is aligned to LSB for LSB justified standard while the valid data is aligned to MSB for MSB justified standard. The timing diagrams for the cases that the channel length is greater than the data length are shown below.

Figure 17-34. LSB justified standard timing diagram (DTLEN=01, CHLEN=1, CKPL=0)

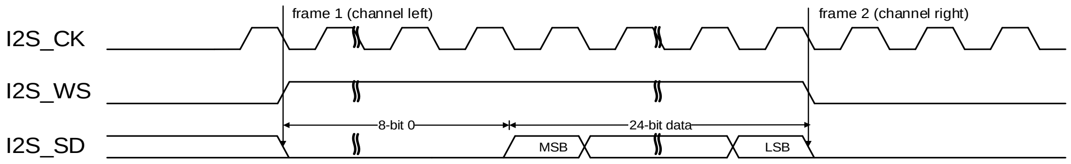
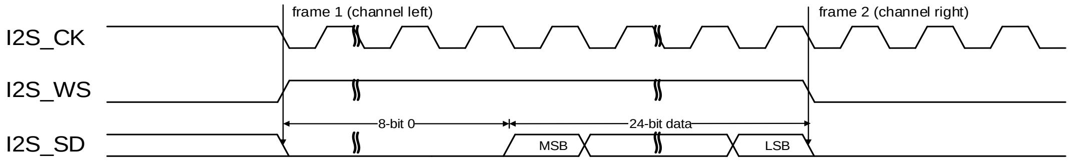


Figure 17-35. LSB justified standard timing diagram (DTLEN=01, CHLEN=1, CKPL=1)



When the packet type is 24-bit data packed in 32-bit frame, two write or read operations to or from the SPI_DATA register are needed to complete the transmission of a frame. In transmission mode, if a 24-bit data D [23:0] is going to be sent, the first data written to the SPI_DATA register should be a 16-bit data. The higher 8 bits of the 16-bit data can be any value and the lower 8 bits should be D [23:16]. The second data written to the SPI_DATA register should be D [15:0]. In reception mode, if a 24-bit data D [23:0] is received, the first data read from the SPI_DATA register is a 16-bit data. The high 8 bits of this 16-bit data are zeros and the lower 8 bits are D [23:16]. The second data read from the SPI_DATA register is D [15:0].

Figure 17-36. LSB justified standard timing diagram (DTLEN=00, CHLEN=1, CKPL=0)

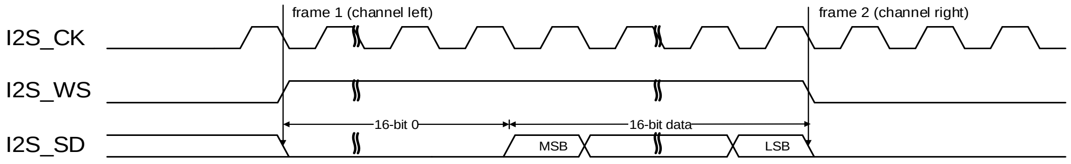
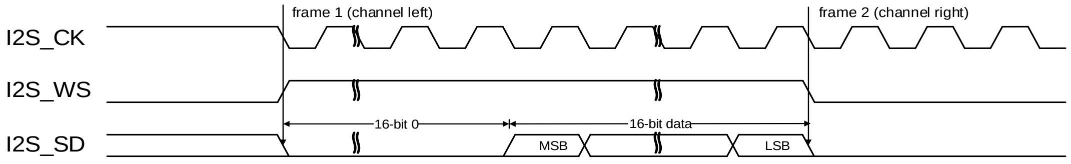


Figure 17-37. LSB justified standard timing diagram (DTLEN=00, CHLEN=1, CKPL=1)



When the packet type is 16-bit data packed in 32-bit frame, only one write or read operation to or from the SPI_DATA register is needed to complete the transmission of a frame. The remaining 16 bits are forced by hardware to 0x0000 to extend the data to 32-bit format.

PCM standard

For PCM standard, I2S_WS and I2S_SD are updated on the rising edge of I2S_CK, and the I2S_WS signal indicates frame synchronization information. Both the short frame synchronization mode and the long frame synchronization mode are available and configurable using the PCMSMOD bit in the SPI_I2SCTL register. The SPI_DATA register is handled in the exactly same way as that for I2S Phillips standard. The timing diagrams for each configuration of the short frame synchronization mode are shown below.

Figure 17-38. PCM standard short frame synchronization mode timing diagram (DTLEN=00, CHLEN=0, CKPL=0)

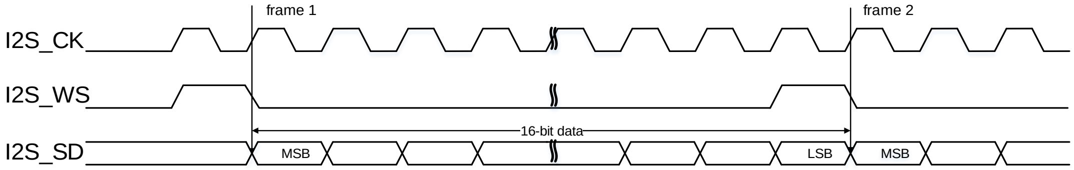


Figure 17-39. PCM standard short frame synchronization mode timing diagram (DTLEN=00, CHLEN=0, CKPL=1)

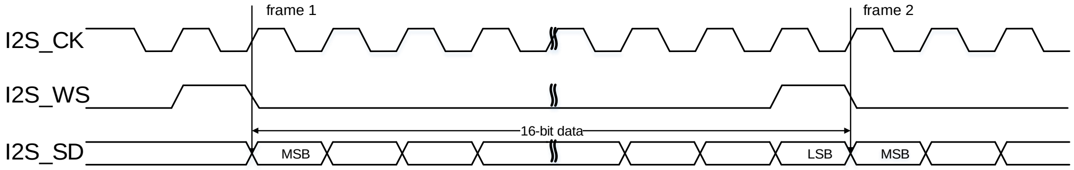


Figure 17-40. PCM standard short frame synchronization mode timing diagram (DTLEN=10, CHLEN=1, CKPL=0)

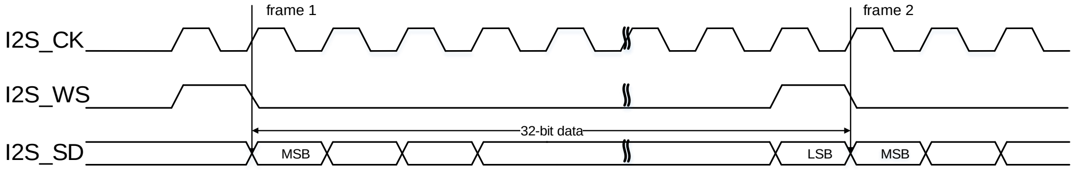


Figure 17-41. PCM standard short frame synchronization mode timing diagram (DTLEN=10, CHLEN=1, CKPL=1)

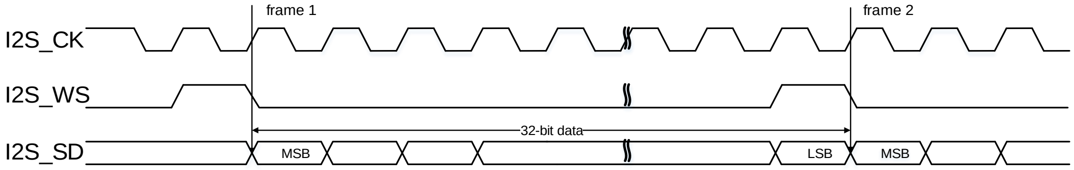


Figure 17-42. PCM standard short frame synchronization mode timing diagram

(DTLEN=01, CHLEN=1, CKPL=0)

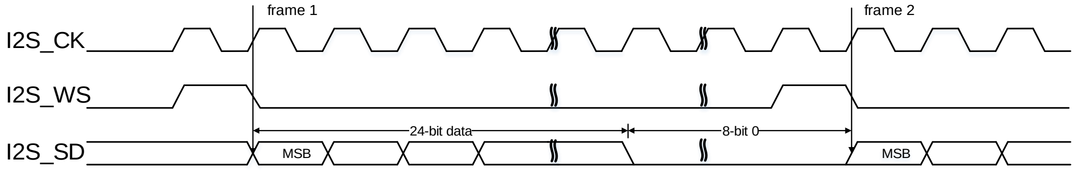


Figure 17-43. PCM standard short frame synchronization mode timing diagram (DTLEN=01, CHLEN=1, CKPL=1)

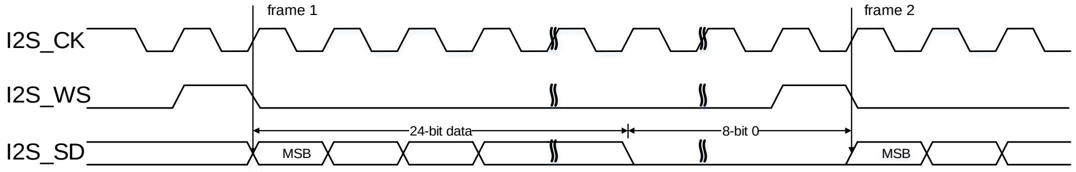


Figure 17-44. PCM standard short frame synchronization mode timing diagram (DTLEN=00, CHLEN=1, CKPL=0)

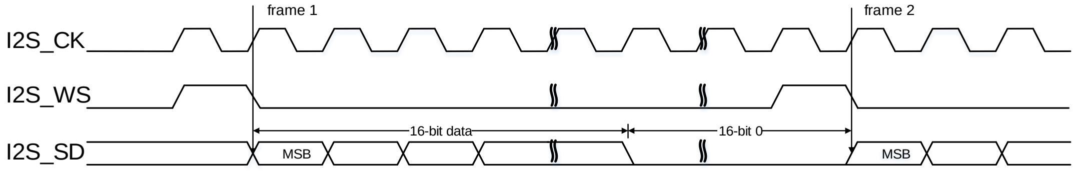
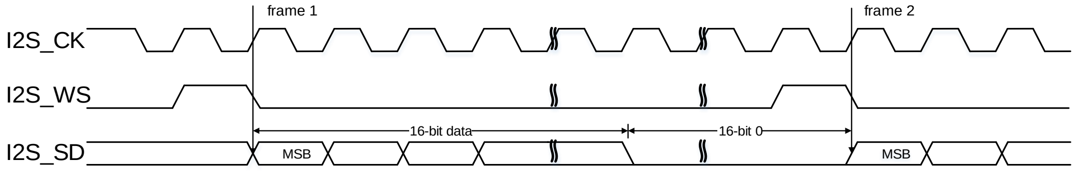


Figure 17-45. PCM standard short frame synchronization mode timing diagram (DTLEN=00, CHLEN=1, CKPL=1)



The timing diagrams for each configuration of the long frame synchronization mode are shown below.

Figure 17-46. PCM standard long frame synchronization mode timing diagram (DTLEN=00, CHLEN=0, CKPL=0)

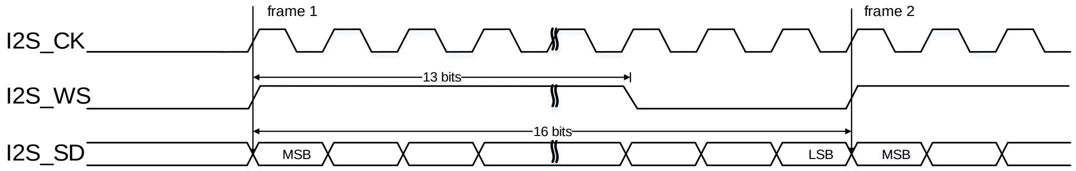


Figure 17-47. PCM standard long frame synchronization mode timing diagram

(DTLEN=00, CHLEN=0, CKPL=1)

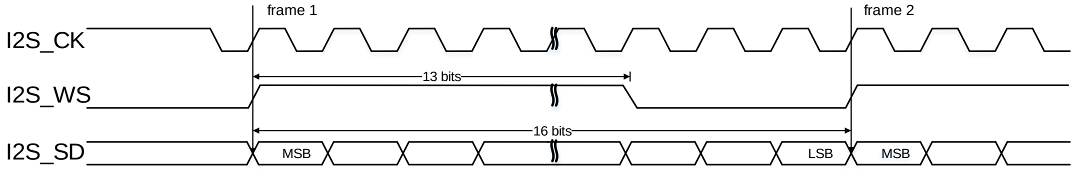


Figure 17-48. PCM standard long frame synchronization mode timing diagram (DTLEN=10, CHLEN=1, CKPL=0)

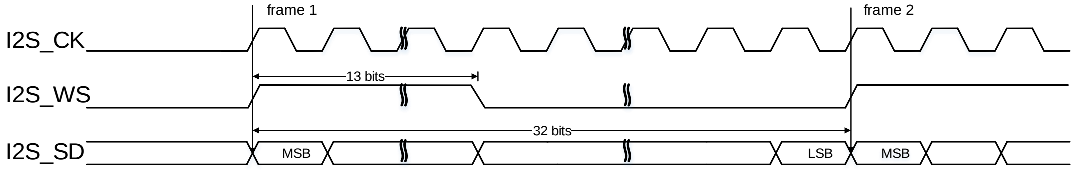


Figure 17-49. PCM standard long frame synchronization mode timing diagram (DTLEN=10, CHLEN=1, CKPL=1)

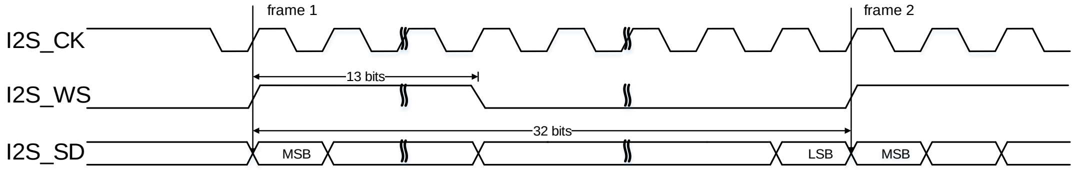


Figure 17-50. PCM standard long frame synchronization mode timing diagram (DTLEN=01, CHLEN=1, CKPL=0)

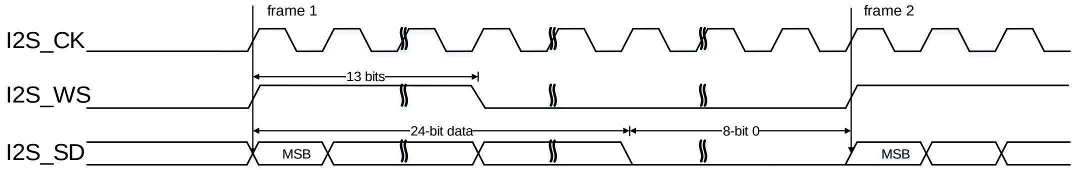


Figure 17-51. PCM standard long frame synchronization mode timing diagram (DTLEN=01, CHLEN=1, CKPL=1)

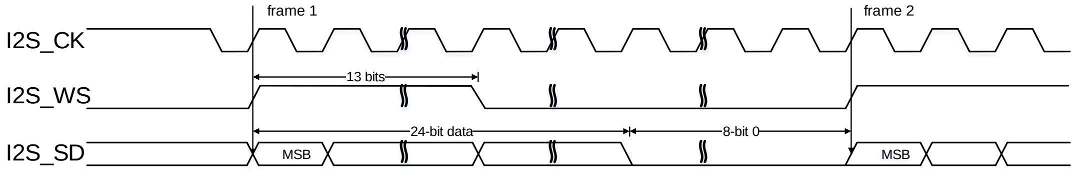


Figure 17-52. PCM standard long frame synchronization mode timing diagram (DTLEN=00, CHLEN=1, CKPL=0)

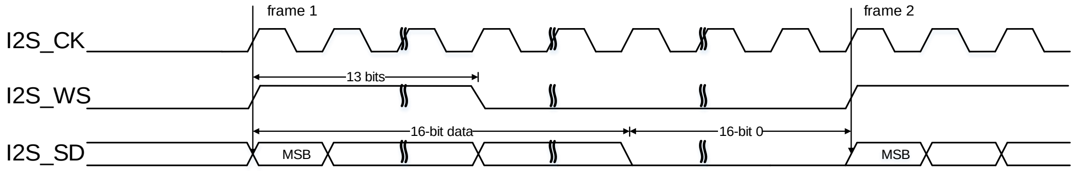
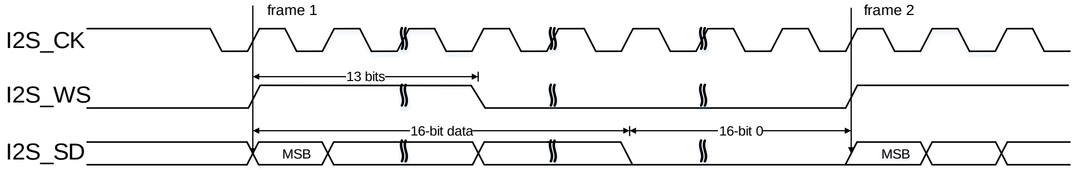
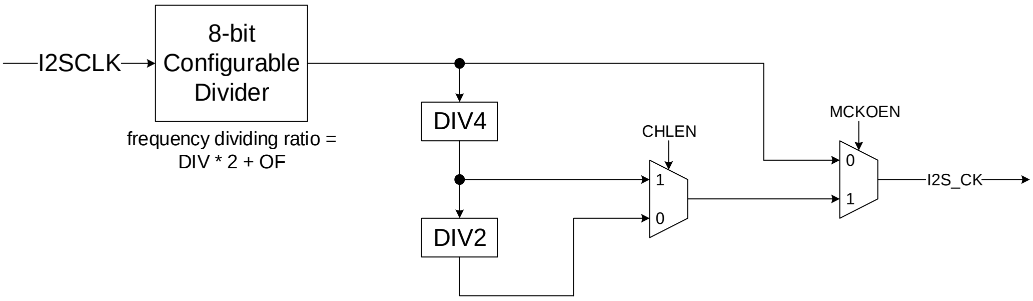


Figure 17-53. PCM standard long frame synchronization mode timing diagram (DTLEN=00, CHLEN=1, CKPL=1)



17.4.4. I2S clock

Figure 17-54. Block diagram of I2S clock generator


I2SCLK
8-bit Configurable Divider
frequency dividing ratio = DIV * 2 + OF
DIV4
DIV2
CHLEN
1
0
MCKOEN
0
1
I2S_CK

The block diagram of I2S clock generator is shown as ***Figure 17-54. Block diagram of I2S clock generator***.The I2S interface clocks are configured by the DIV bits, the OF bit, the MCKOEN bit in the SPI_I2SPSC register and the CHLEN bit in the SPI_I2SCTL register. The I2S source clock (I2SCLK) is from the CK_I2S. The I2S bitrate can be calculated by the formulas shown in ***Table 17-7. I2S bitrate calculation formulas***.

Note: The I2SCLK clock is derived from CK_I2S of the RCU module.

Table 17-7. I2S bitrate calculation formulas

MCKOEN	CHLEN	Formula
0	0	I2SCLK / (DIV * 2 + OF)
0	1	I2SCLK / (DIV * 2 + OF)
1	0	I2SCLK / (8 * (DIV * 2 + OF))
1	1	I2SCLK / (4 * (DIV * 2 + OF))

The relationship between audio sampling frequency (Fs) and I2S bitrate is defined by the following formula:

Fs = I2S bitrate / (number of bits per channel * number of channels)

So, in order to get the desired audio sampling frequency, the clock generator needs to be configured according to the formulas listed in ***Table 17-8. Audio sampling frequency calculation formulas***.

Table 17-8. Audio sampling frequency calculation formulas

MCKOEN	CHLEN	Formula
0	0	I2SCLK / (32 * (DIV * 2 + OF))
0	1	I2SCLK / (64 * (DIV * 2 + OF))
1	0	I2SCLK / (256 * (DIV * 2 + OF))
1	1	I2SCLK / (256 * (DIV * 2 + OF))

17.4.5. Operation

Operation modes

The operation mode is selected by the I2SOPMOD bits in the SPI_I2SCTL register. There are four available operation modes, including master transmission mode, master reception mode, slave transmission mode, and slave reception mode. The direction of I2S interface signals for each operation mode is shown in the ***Table 17-9. Direction of I2S interface signals for each operation mode***.

Table 17-9. Direction of I2S interface signals for each operation mode

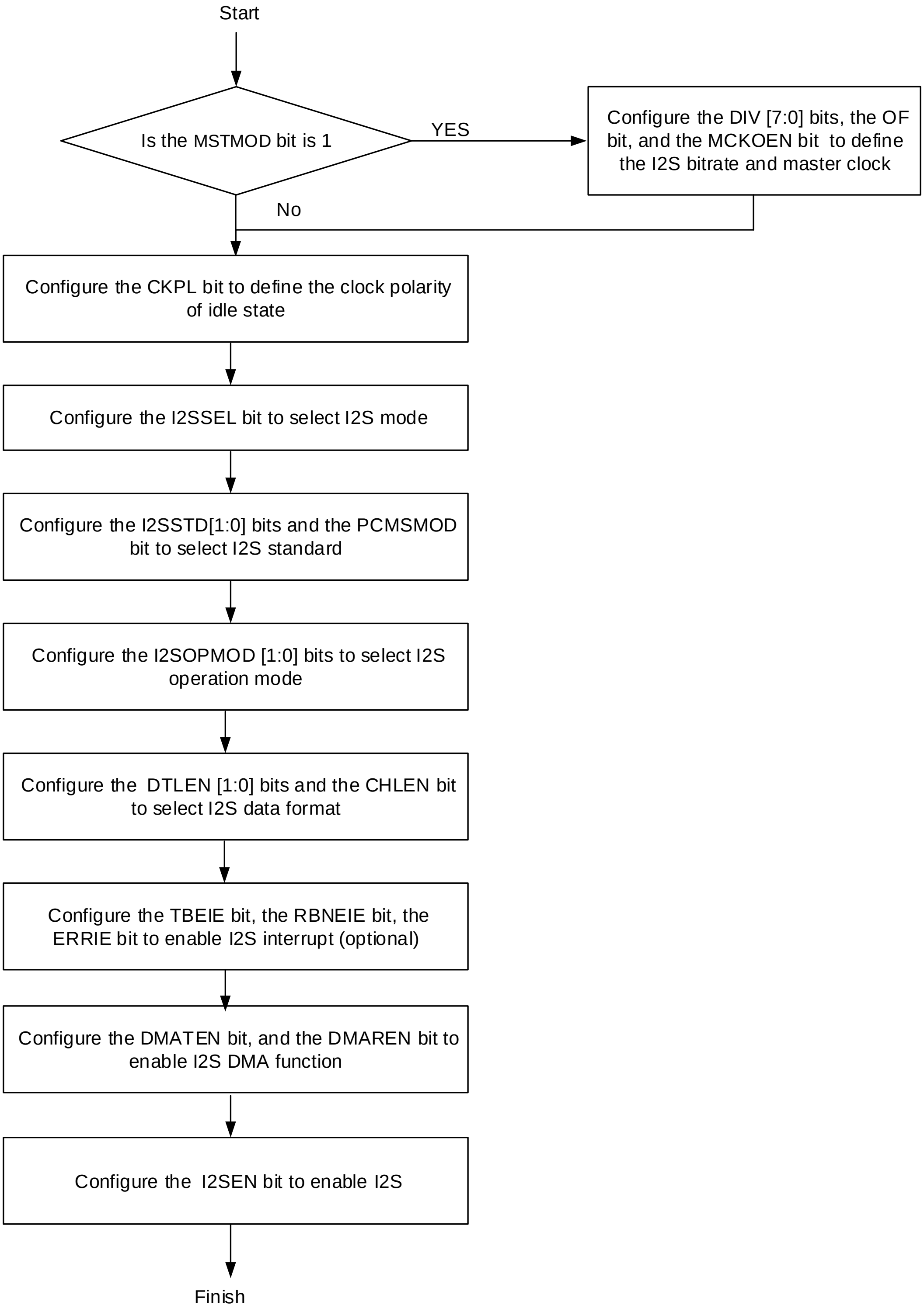
Operation mode	I2S_CK	I2S_WS	I2S_SD
Master transmission	Output	Output	Output
Master reception	Output	Output	Input
Slave transmission	Input	Input	Output
Slave reception	Input	Input	Input

(1) NU means the pin is not used by I2S and can be used by other functions.

I2S initialization sequence

I2S initialization sequence shown as below ***Figure 17-55. I2S initialization sequence***.

Figure 17-55. I2S initialization sequence


Start
Is the MSTMOD bit is 1
YES
Configure the DIV [7:0] bits, the OF bit, and the MCKOEN bit to define the I2S bitrate and master clock
No
Configure the CKPL bit to define the clock polarity of idle state
Configure the I2SSEL bit to select I2S mode
Configure the I2SSTD[1:0] bits and the PCMSMOD bit to select I2S standard
Configure the I2SOPMOD [1:0] bits to select I2S operation mode
Configure the DTLEN [1:0] bits and the CHLEN bit to select I2S data format
Configure the TBEIE bit, the RBNEIE bit, the ERRIE bit to enable I2S interrupt (optional)
Configure the DMATEN bit, and the DMAREN bit to enable I2S DMA function
Configure the I2SEN bit to enable I2S
Finish

I2S master transmission sequence

The TBE flag is used to control the transmission sequence. As is mentioned before, the TBE flag indicates that the transmit buffer is empty, and an interrupt will be generated if the TBEIE bit in the SPI_CTL1 register is set. At the beginning, the transmit buffer is empty (TBE is high) and no transmission sequence is processing in the shift register. When a half word is written to the SPI_DATA register (TBE goes low), the data is transferred from the transmit buffer to the shift register (TBE goes high) immediately. At the moment, the transmission sequence begins.

The data is parallel loaded into the 16-bit shift register, and shifted out serially to the I2S_SD pin, MSB first. The next data should be written to the SPI_DATA register, when the TBE flag is high. After a write operation to the SPI_DATA register, the TBE flag goes low. When the current transmission finishes, the data in the transmit buffer is loaded into the shift register, and the TBE flag goes back high. Software should write the next audio data into SPI_DATA register before the current data finishes, otherwise, the audio data transmission is not continuous.

For all standards except PCM, the I2SCH flag is used to distinguish which channel side the data to transfer belongs to. The I2SCH flag is refreshed at the moment when the TBE flag goes high. At the beginning, the I2SCH flag is low, indicating the left channel data should be written to the SPI_DATA register.

In order to disable I2S, it is mandatory to clear the I2SEN bit after the TBE flag is high and the TRANS flag is low.

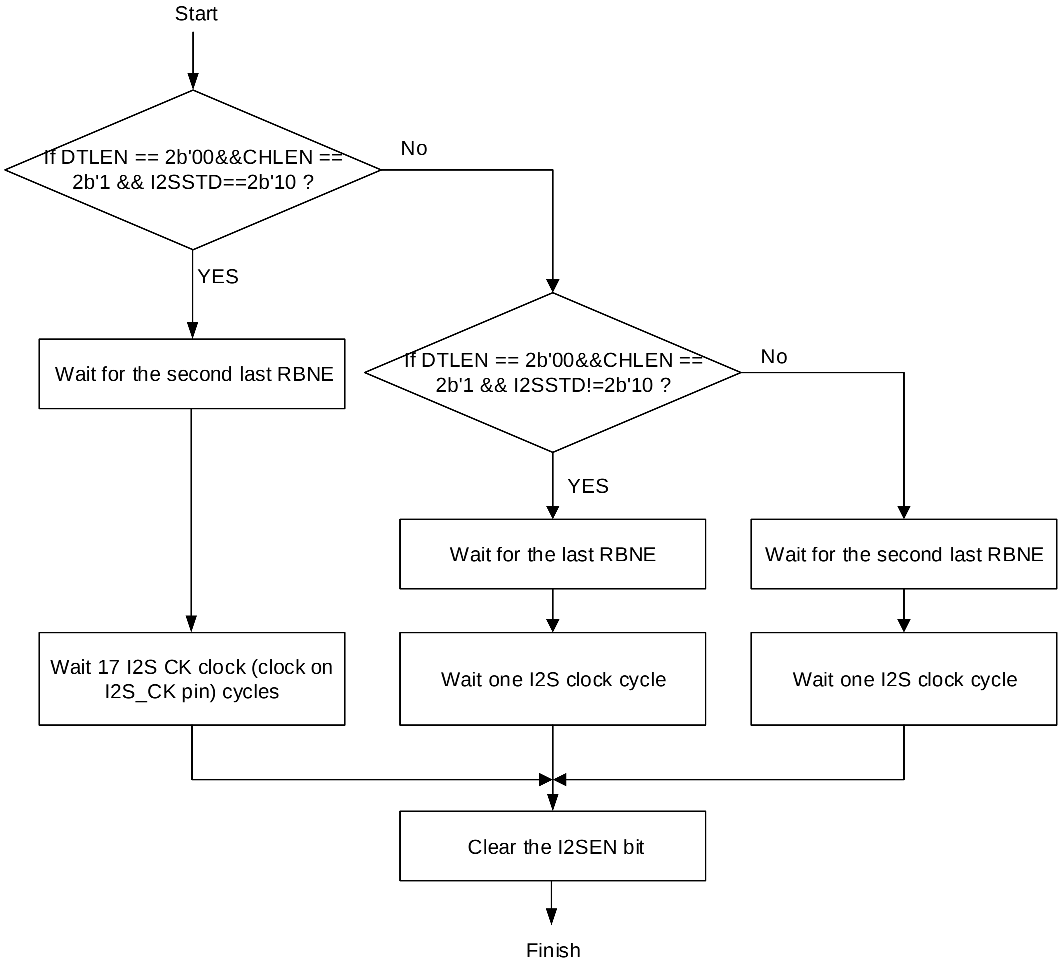
I2S master reception sequence

The RBNE flag is used to control the reception sequence. As is mentioned before, the RBNE flag indicates the receive buffer is not empty, and an interrupt will be generated if the RBNEIE bit in the SPI_CTL1 register is set. The reception sequence begins immediately when the I2SEN bit in the SPI_I2SCTL register is set. At the beginning, the receive buffer is empty (RBNE is low). When a reception sequence finishes, the received data in the shift register is loaded into the receive buffer (RBNE goes high). The data should be read from the SPI_DATA register, when the RBNE flag is high. After a read operation to the SPI_DATA register, the RBNE flag goes low. It is mandatory to read the SPI_DATA register before the end of the next reception. Otherwise, reception overrun error occurs. The RXORERR flag is set and an interrupt may be generated if the ERRIE bit in the SPI_CTL1 register is set. In this case, it is necessary to disable and then enable I2S before resuming the communication.

For all standards except PCM, the I2SCH flag is used to distinguish the channel side which the received data belongs to. The I2SCH flag is refreshed at the moment when the RBNE flag goes high.

Different sequences are used to disable the I2S in different standards, data length and channel length. The sequences for each case are shown as below ***Figure 17-56. I2S master reception disabling sequence***.

Figure 17-56. I2S master reception disabling sequence


Start
If DTLEN == 2b'00&&CHLEN == 2b'1 && I2SSTD==2b'10 ?
No
YES
Wait for the second last RBNE
If DTLEN == 2b'00&&CHLEN == 2b'1 && I2SSTD!=2b'10 ?
No
YES
Wait for the last RBNE
Wait for the second last RBNE
Wait 17 I2S CK clock (clock on I2S_CK pin) cycles
Wait one I2S clock cycle
Wait one I2S clock cycle
Clear the I2SEN bit
Finish

I2S slave transmission sequence

The transmission sequence in slave mode is similar to that in master mode. The difference between them is described below.

In slave mode, the slave has to be enabled before the external master starts the communication. The transmission sequence begins when the external master sends the clock and when the I2S_WS signal requests the transfer of data. The data has to be written to the SPI_DATA register before the master initiates the communication. Software should write the next audio data into SPI_DATA register before the current data finishes. Otherwise, transmission underrun error occurs. The TXURERR flag is set and an interrupt may be generated if the ERRIE bit in the SPI_CTL1 register is set. In this case, it is mandatory to disable and enable I2S to resume the communication. In slave mode, I2SCH is sensitive to the I2S_WS signal coming from the external master.

In order to disable I2S, it is mandatory to clear the I2SEN bit after the TBE flag is high and the TRANS flag is low.

I2S slave reception sequence

The reception sequence in slave mode is similar to that in master mode. The differences between them are described below.

In slave mode, the slave has to be enabled before the external master starts the communication. The reception sequence begins when the external master sends the clock and when the I2S_WS signal indicates a start of the data transfer. In slave mode, I2SCH is sensitive to the I2S_WS signal coming from the external master.

In order to disable I2S, it is mandatory to clear the I2SEN bit immediately after receiving the last RBNE.

17.4.6. DMA function

DMA function is the same as SPI mode. The only difference is that the CRC function is not available in I2S mode.

17.4.7. I2S interrupts

Status flags

There are four status flags implemented in the SPI_STAT register, including TBE, RBNE, TRANS and I2SCH. The user can use them to fully monitor the state of the I2S bus.

- Transmit buffer empty flag (TBE)

This bit is set when the transmit buffer is empty, the software can write the next data to the transmit buffer by writing the SPI_DATA register.

- Receive buffer not empty flag (RBNE)

This bit is set when receive buffer is not empty, which means that one data is received and stored in the receive buffer, and software can read the data by reading the SPI_DATA register.

- I2S transmitting ongoing flag (TRANS)

TRANS is a status flag to indicate whether the transfer is ongoing or not. It is set and cleared by hardware and not controlled by software. This flag will not generate any interrupt.

- I2S channel side flag (I2SCH)

This flag indicates the channel side information of the current transfer and has no meaning in PCM mode. It is updated when TBE rises in transmission mode or RBNE rises in reception mode. This flag will not generate any interrupt.

Error conditions

There are three error flags:

- Transmission underrun error flag (TXURERR)

This situation occurs when the transmit buffer is empty when the valid SCK signal starts in slave transmission mode.

- Reception overrun error flag (RXORERR)

This situation occurs when the receive buffer is full and a newly incoming data has been completely received. When overrun occurs, the data in receive buffer is not updated and the newly incoming data is lost.

- Format error (FERR)

In slave I2S mode, the I2S monitors the I2S_WS signal and an error flag will be set if I2S_WS toggles at an unexpected position.

I2S interrupt events and corresponding enabled bits are summed up in the ***Table 17-10. I2S interrupt.***

Table 17-10. I2S interrupt

Interrupt flag	Description	Clear method	Interrupt enable bit
TBE	Transmit buffer empty	Write SPI_DATA register	TBEIE
RBNE	Receive buffer not empty	Read SPI_DATA register	RBNEIE
TXURERR	Transmission underrun error	Read SPI_STAT register	ERRIE
RXORERR	Reception overrun error	Read SPI_DATA register and then read SPI_STAT register.	
FERR	I2S format error	Read SPI_STAT register	

17.5. Register definition

SPI0 / I2S0 base address: 0x4001 3000

SPI1 base address: 0x4000 3800

17.5.1. Control register 0 (SPI_CTL0)

Address offset: 0x00

Reset value: 0x0000 0000

This register can be accessed by half-word (16-bit) or word (32-bit).

This register has no meaning in I2S mode.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BDEN	BDOEN	CRCEN	CRCNT	FF16 CRCL	RO	SWNSS EN	SWNSS	LF	SPIEN	PSC [2:0]			MSTMOD	CKPL	CKPH
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw			rw	rw	rw

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15	BDEN	Bidirectional enable 0: 2 line unidirectional transmit mode 1: 1 line bidirectional transmit mode. The information transfers between the MOSI pin in master and the MISO pin in slave.
14	BDOEN	Bidirectional transmit output enable When BDEN is set, this bit determines the direction of transfer. 0: Work in receive-only mode 1: Work in transmit-only mode
13	CRCEN	CRC calculation enable 0: CRC calculation is disabled. 1: CRC calculation is enabled.
12	CRCNT	CRC next transfer 0: Next transfer is data 1: Next transfer is CRC value (TCRC) When the transfer is managed by DMA, CRC value is transferred by hardware. This bit should be cleared. In full-duplex or transmit-only mode, set this bit after the last data is written to SPI_DATA register. In receive only mode, set this bit after the second last data is received.

11	FF16	Data frame format (for SPI0) 0: 8-bit data frame format 1: 16-bit data frame format
	CRCL	CRC length (only for SPI1) 0: 8-bit crc length. 1: 16-bit crc length.
10	RO	Receive only When BDEN is cleared, this bit determines the direction of transfer. 0: Full-duplex mode 1: Receive-only mode
9	SWNSSEN	NSS software mode selection 0: NSS hardware mode. The NSS level depends on NSS pin. 1: NSS software mode. The NSS level depends on SWNSS bit. This bit has no meaning in SPI TI mode.
8	SWNSS	NSS pin selection in NSS software mode 0: NSS pin is pulled low. 1: NSS pin is pulled high. This bit has an effect only when the SWNSSEN bit is set. This bit has no meaning in SPI TI mode.
7	LF	LSB first mode 0: Transmit MSB first 1: Transmit LSB first This bit has no meaning in SPI TI mode.
6	SPIEN	SPI enable 0: SPI peripheral is disabled. 1: SPI peripheral is enabled.
5:3	PSC[2:0]	Master clock prescaler selection 000: PCLK/2 001: PCLK/4 010: PCLK/8 011: PCLK/16 100: PCLK/32 101: PCLK/64 110: PCLK/128 111: PCLK/256
2	MSTMOD	Master mode enable 0: Slave mode 1: Master mode
1	CKPL	Clock polarity selection 0: CLK pin is pulled low when SPI is idle.

		1: CLK pin is pulled high when SPI is idle.
0	CKPH	Clock phase selection 0: Capture the first data at the first clock transition. 1: Capture the first data at the second clock transition.

17.5.2. Control register 1 (SPI_CTL1)

Address offset: 0x04

Reset value: 0x0000 0700 for SPI1, 0x0000 0000 for SPI0

This register can be accessed by half-word (16-bit) or word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	TXDMA_ ODD	RXDMA_ ODD	BYTEN	DZ[3:0]				TBEIE	RBNEIE	ERRIE	TMOD	NSSP	NSSDRV	DMATEN	DMAREN
	rw	rw	rw	rw				rw	rw	rw	rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31:15	Reserved	Must be kept at reset value.
14	TXDMA_ODD	Odd bytes in TX DMA channel (only for SPI1) In data merging mode, this bit is set if the total number of data to transmit by DMA is odd. It has effect only when DMATEN is set and data merging mode enable (data size is less than or equal to 8-bit and write access to SPI_DATA is 16-bit wide). This field can be written only when SPI is disabled. 0: The total number of data to transmit by DMA is even. 1: The total number of data to transmit by DMA is odd.
13	RXDMA_ODD	Odd bytes in RX DMA channel (only for SPI1) In data merging mode, this bit is set if the total number of data to receive by DMA is odd. It has effect only when DMAREN is set and data merging mode enable (data size is less than or equal to 8-bit and write access to SPI_DATA is 16-bit wide). This field can be written only when SPI is disabled. 0: The total number of data to receive by DMA is even. 1: The total number of data to receive by DMA is odd.
12	BYTEN	Byte access enable (only for SPI1) This bit is used to indicate the access size to FIFO, and set the threshold of the RXFIFO that generate RBNE. 0: Half-word access, and RBNE is generated when RXLVL >= 2. 1: Byte access, and RBNE is generated when RXLVL >= 1.
11:8	DZ[3:0]	Date size (only for SPI1) This field indicates the data size for transfer.

		0000: Force to “0111” 0001: Force to “0111” 0010: Force to “0111” 0011: 4-bit 0100: 5-bit …… 1111: 16-bit
7	TBEIE	Transmit buffer / TXFIFO empty interrupt enable 0: TBE interrupt is disabled. 1: TBE interrupt is enabled. An interrupt is generated when the TBE bit is set
6	RBNEIE	Receive buffer / RXFIFO not empty interrupt enable 0: RBNE interrupt is disabled. 1: RBNE interrupt is enabled. An interrupt is generated when the RBNE bit is set.
5	ERRIE	Errors interrupt enable. 0: Error interrupt is disabled. 1: Error interrupt is enabled. An interrupt is generated when the CRCERR bit or the CONFERR bit or the RXORERR bit or the TXURERR bit is set.
4	TMOD	SPI TI mode enable. 0: SPI TI mode disabled. 1: SPI TI mode enabled.
3	NSSP	SPI NSS pulse mode enable. 0: SPI NSS pulse mode disable. 1: SPI NSS pulse mode enable.
2	NSSDRV	Drive NSS output 0: NSS output is disabled. 1: NSS output is enabled. If the NSS pin is configured as output, the NSS pin is pulled low in master mode when SPI is enabled. If the NSS pin is configured as input, the NSS pin should be pulled high in master mode, and this bit has no effect.
1	DMATEN	Transmit buffer / TXFIFO DMA enable 0: Transmit buffer / TXFIFO DMA is disabled. 1: Transmit buffer / TXFIFO DMA is enabled, when the TBE bit in SPI_STAT is set, it will be a DMA request on corresponding DMA channel.
0	DMAREN	Receive buffer / RXFIFO DMA enable 0: Receive buffer / RXFIFO DMA is disabled. 1: Receive buffer / RXFIFO DMA is enabled, when the RBNE bit in SPI_STAT is set, it will be a DMA request on corresponding DMA channel.

17.5.3. Status register (SPI_STAT)

Address offset: 0x08

Reset value: 0x0000 0002

This register can be accessed by half-word (16-bit) or word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved			TXLVL[1:0]		RXLVL[1:0]		FERR	TRANS	RXORERR	CONFERR	CRCERR	TXURERR	I2SCH	TBE	RBNE
			r		r		rc_w0	r	r	r	rc_w0	r	r	r	r

Bits	Fields	Descriptions
31:13	Reserved	Must be kept at reset value.
12:11	TXLVL[1:0]	TXFIFO level (only for SPI1) 00: Empty 01: 1/4 full 10: 1/2 full 11: Full **Note:** The FIFO level here refers to the current actual storage of the FIFO. Here, the FIFO is considered full when the FIFO level is greater than 1/2.
10:9	RXLVL[1:0]	RXFIFO level (only for SPI1) 00: Empty 01: 1/4 full 10: 1/2 full 11: Full This field has no meaning when SPI is in receive-only mode with CRC function enabled. **Note:** The FIFO level here refers to the current actual storage of the FIFO. Here, the FIFO is considered full when the FIFO level is greater than 1/2.
8	FERR	Format error SPI TI mode: 0: No TI mode format error 1: TI mode format error occurs. This bit is set by hardware and is able to be cleared by writing 0.
7	TRANS	Transmitting ongoing bit 0: SPI is idle. 1: SPI is currently transmitting and/or receiving a frame. This bit is set and cleared by hardware.
6	RXORERR	Reception overrun error bit

		0: No reception overrun error occurs. 1: Reception overrun error occurs. This bit is set by hardware and cleared by a read operation on the SPI_DATA register followed by a read access to the SPI_STAT register.
5	CONFERR	SPI configuration error 0: No configuration fault occurs. 1: Configuration fault occurred. (In master mode, the NSS pin is pulled low in NSS hardware mode or SWNSS bit is low in NSS software mode.) This bit is set by hardware and cleared by a read or write operation on the SPI_STAT register followed by a write access to the SPI_CTL0 register.
4	CRCERR	SPI CRC error bit 0: The SPI_RCRC value is equal to the received CRC data at last. 1: The SPI_RCRC value is not equal to the received CRC data at last. This bit is set by hardware and is able to be cleared by writing 0.
3	TXURERR	Transmission underrun error bit 0: No transmission underrun error occurs. 1: Transmission underrun error occurs. This bit is set by hardware and cleared by a read operation on the SPI_STAT register. This bit is not used in SPI mode.
2	I2SCH	I2S channel side 0: The next data needs to be transmitted or the data just received is channel left. 1: The next data needs to be transmitted or the data just received is channel right. This bit is set and cleared by hardware. This bit is not used in SPI mode, and has no meaning in the I2S PCM mode.
1	TBE	Transmit buffer / TXFIFO empty 0: Transmit buffer / TXFIFO is not empty. 1: Transmit buffer / TXFIFO is empty.
0	RBNE	Receive buffer / RXFIFO not empty 0: Receive buffer / RXFIFO is empty. 1: Receive buffer / RXFIFO is not empty.

17.5.4. Data register (SPI_DATA)

Address offset: 0x0C
Reset value: 0x0000 0000

For SPI1, this register can be accessed by byte (8-bit) or half-word (16-bit).For SPI0, this register can be accessed by half-word (16-bit) or word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SPI_DATA[15:0]															
rw															

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	SPI_DATA[15:0]	Data transfer register. For SPI1, the hardware has two FIFOs, including TXFIFO and RXFIFO. The SPI_DATA register serves as an interface between the Rx and Tx FIFOs. Write data to SPI_DATA will save the data to TXFIFO and read data from SPI_DATA will get the data from RXFIFO. For SPI0, the hardware has two buffers, including transmit buffer and receive buffer. Write data to SPI_DATA will save the data to transmit buffer and read data from SPI_DATA will get the data from receive buffer. When the data frame format is set to 8-bit data, the SPI_DATA [15:8] is forced to 0 and the SPI_DATA [7:0] is used for transmission and reception, transmit buffer and receive buffer are 8-bits. If the Data frame format is set to 16-bit data, the SPI_DATA [15:0] is used for transmission and reception, transmit buffer and receive buffer are 16-bit. **Note:** In fact, SPI1 hardware determines the size of each access to SPI_DATA only based on the BYTEN bit in SPI_CTL1, regardless of the size of the software's current operation.

17.5.5. CRC polynomial register (SPI_CRCPOLY)

Address offset: 0x10

Reset value: 0x0000 0007

This register can be accessed by half-word (16-bit) or word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CRCPOLY[15:0]															
rw															

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value
15:0	CRCPOLY[15:0]	CRC polynomial register This register contains the CRC polynomial and it is used for CRC calculation. The default value is 0007h.

17.5.6. Receive CRC register (SPI_RCRC)

Address offset: 0x14

Reset value: 0x0000 0000

This register can be accessed by half-word (16-bit) or word (32-bit).

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="16">Reserved</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="16">RCRC[15:0]</td></tr>
<tr><td colspan="16">r</td></tr>
</table>

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	RCRC[15:0]	RX CRC value When the CRCEN bit of SPI_CTL0 is set, the hardware computes the CRC value of the received bytes and saves them in RCRC register. For SPI0, if the data frame format is set to 8-bit data, CRC calculation is based on CRC8 standard, and saves the value in RCRC[7:0], when the data frame format is set to 16-bit data, CRC calculation is based on CRC16 standard, and saves the value in RCRC[15:0]. For SPI1, CRC function is valid only when the data length is 8 bits or 16 bits. And if the CRC length is set to 8-bit and the data size is equal to 8-bit, the CRC calculation is based on CRC8 standard, and saves the value in RCRC [7:0]. In addition to this, the calculation is based on CRC16 standard, and saves the value in RCRC [15:0]. The hardware computes the CRC value after each received bit, when the TRANS is set, a read to this register could return an intermediate value. This register is reset when the CRCEN bit in SPI_CTL0 register or the SPIxRST bit in RCU reset register is set.

17.5.7. Transmit CRC register (SPI_TCRC)

Address offset: 0x18

Reset value: 0x0000 0000

This register can be accessed by half-word (16-bit) or word (32-bit).

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="16">Reserved</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="16">TCRC[15:0]</td></tr>
<tr><td colspan="16">r</td></tr>
</table>

Bits	Fields	Descriptions

31:16	Reserved	Must be kept at reset value.
15:0	TCRC[15:0]	TX CRC value When the CRCEN bit of SPI_CTL0 is set, the hardware computes the CRC value of the transmitted bytes and saves them in TCRC register. For SPI0, if the data frame format is set to 8-bit data, CRC calculation is based on CRC8 standard, and saves the value in TCRC[7:0], when the data frame format is set to 16-bit data, CRC calculation is based on CRC16 standard, and saves the value in TCRC[15:0]. For SPI1, CRC function is valid only when the data length is 8 bits or 16 bits. And if the CRC length is set to 8-bit and the data size is equal to 8-bit, the CRC calculation is based on CRC8 standard, and saves the value in TCRC[7:0]. In addition to this, the calculation is based on CRC16 standard, and saves the value in TCRC[15:0]. The hardware computes the CRC value after each transmitted bit, when the TRANS is set, a read to this register could return an intermediate value. The different frame formats (LF bit of the SPI_CTL0) will get different CRC values. This register is reset when the CRCEN bit in SPI_CTL0 register or the SPIxRST bit in RCU reset register is set.

17.5.8. I2S control register (SPI_I2SCTL)

Address offset: 0x1C

Reset value: 0x0000 0000

This register can be accessed by half-word (16-bit) or word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				I2SSEL	I2SEN	I2SOPMOD[1:0]		PCMSMOD	Reserved	I2SSTD[1:0]		CKPL	DTLEN[1:0]		CHLEN
				rw	rw	rw		rw		rw		rw	rw		rw

Bits	Fields	Descriptions
31:12	Reserved	Must be kept at reset value.
11	I2SSEL	I2S mode selection 0: SPI mode 1: I2S mode This bit should be configured when SPI/I2S is disabled.
10	I2SEN	I2S enable 0: I2S is disabled 1: I2S is enabled This bit is not used in SPI mode.
9:8	I2SOPMOD[1:0]	I2S operation mode

		00: Slave transmission mode 01: Slave reception mode 10: Master transmission mode 11: Master reception mode This bit should be configured when I2S mode is disabled. This bit is not used in SPI mode.
7	PCMSMOD	PCM frame synchronization mode 0: Short frame synchronization 1: long frame synchronization This bit has a meaning only when PCM standard is used. This bit should be configured when I2S mode is disabled. This bit is not used in SPI mode.
6	Reserved	Must be kept at reset value.
5:4	I2SSTD[1:0]	I2S standard selection 00: I2S Phillips standard 01: MSB justified standard 10: LSB justified standard 11: PCM standard These bits should be configured when I2S mode is disabled. These bits are not used in SPI mode.
3	CKPL	Idle state clock polarity 0: The idle state of I2S_CK is low level. 1: The idle state of I2S_CK is high level. This bit should be configured when I2S mode is disabled. This bit is not used in SPI mode.
2:1	DTLEN[1:0]	Data length 00: 16 bits 01: 24 bits 10: 32 bits 11: Reserved These bits should be configured when I2S mode is disabled. These bits are not used in SPI mode.
0	CHLEN	Channel length 0: 16 bits 1: 32 bits The channel length must be equal to or greater than the data length. This bit should be configured when I2S mode is disabled. This bit is not used in SPI mode.

17.5.9. I2S clock prescaler register (SPI_I2SPSC)

Address offset: 0x20

Reset value: 0x0000 0002

This register can be accessed by half-word (16-bit) or word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved						MCKOEN	OF	DIV[7:0]							
						rw	rw	rw							

Bits	Fields	Descriptions
31:10	Reserved	Must be kept at reset value.
9	MCKOEN	I2S_MCK output enable 0: I2S_MCK output is disabled. 1: I2S_MCK output is enabled. This bit should be configured when I2S mode is disabled. This bit is not used in SPI mode. **Note:** This bit is only used for I2S_SCK configuration.
8	OF	Odd factor for the prescaler 0: Real divider value is DIV * 2 1: Real divider value is DIV * 2 + 1 This bit should be configured when I2S mode is disabled. This bit is not used in SPI mode.
7:0	DIV[7:0]	Dividing factor for the prescaler Real divider value is DIV * 2 + OF. DIV must not be 0. These bits should be configured when I2S mode is disabled. These bits are not used in SPI mode.

17.5.10. Quad-SPI mode control register (SPI_QCTL) of SPI1

Address offset: 0x80

Reset value: 0x0000 0000

This register can be accessed by half-word (16-bit) or word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved													IO23_DRV	QRD	QMOD

rw	rw	rw

Bits	Fields	Descriptions
31:3	Reserved	Must be kept at reset value.
2	IO23_DRV	Drive IO2 and IO3 enable 0: IO2 and IO3 are not driven in single wire mode. 1: IO2 and IO3 are driven to high in single wire mode. This bit is only available in SPI1.
1	QRD	Quad-SPI mode read select. 0: SPI is in quad wire write mode. 1: SPI is in quad wire read mode. This bit should be only be configured when SPI is not busy (TRANS bit cleared) This bit is only available in SPI1.
0	QMOD	Quad-SPI mode enable. 0: SPI is in single wire mode. 1: SPI is in Quad-SPI mode. This bit should only be configured when SPI is not busy (TRANS bit cleared). This bit is only available in SPI1.

18. Comparator (CMP)

18.1. Overview

The general purpose CMP can work either standalone (all terminal are available on I / Os) or together with the timers.

It can be used to wake up the MCU from low-power mode by an analog signal, provide a trigger source when an analog signal is in a certain condition.

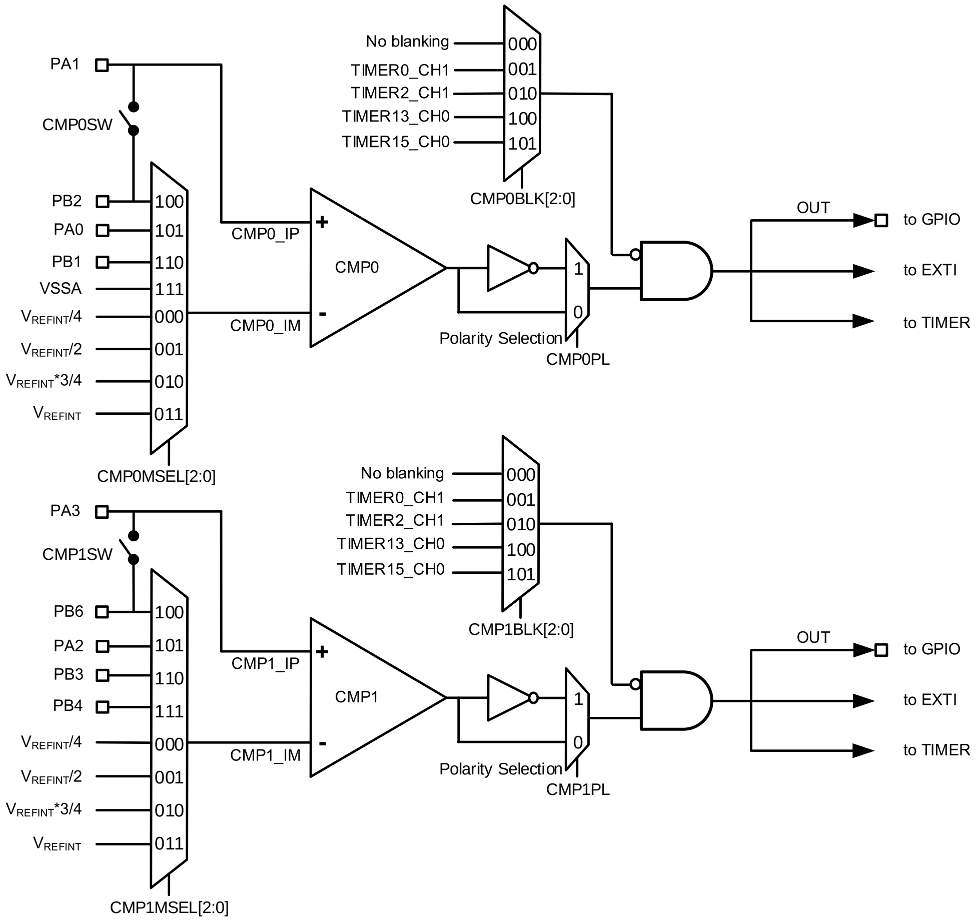
18.2. Characteristics

- Rail-to-rail comparators.
- Configurable hysteresis.
- Configurable speed and consumption.
- Configurable analog input source.
  - DAC output.
  - Multiplexed I / O pins.
  - The whole or sub-multiple values of internal reference voltage.
- Outputs with blanking source.
- Outputs to I / O.
- Outputs to timers for triggering.
- Outputs to EXTI.

18.3. Function overview

The block diagram of CMP is shown below:

Figure 18-1. CMP block diagram



Note: V_{REFINT} is 1.2V.

18.3.1. CMP clock

The clock of the CMP which is connected to APB bus, is synchronous with PCLK.

18.3.2. CMP I / O configuration

These I / Os must be configured in analog mode in the GPIOs registers before they are selected as CMP inputs.

The CMP output can be redirected internally and externally simultaneously.

Refer to pin definitions in datasheet, and the CMP output can be connected to the corresponding I/O port via the alternate function of the GPIO.

CMP output internally connect to the TIMER and the connections between them are as follows:

- CMP output to the TIMER input channel, configured by the TIMER0_INSEL register.

In order to work even in Deep-sleep mode, the polarity selection logic and the output redirection to the port work independently from PCLK.

Table 18-1. CMP inputs and outputs summary details the inputs and outputs of the CMP.

Table 18-1. CMP inputs and outputs summary

	CMP0	CMP1
CMP non inverting inputs connected to I / Os	PA1	PA3
CMP inverting inputs connected to I / Os	PA0 PB1 PB2 VSSA	PA2 PB3 PB4 PB6
CMP inverting inputs connected to internal signals	V_{REFINT}/4, V_{REFINT}/2, V_{REFINT}*3/4, V_{REFINT},	V_{REFINT}/4, V_{REFINT}/2, V_{REFINT}*3/4, V_{REFINT},
CMP outputs connected to I/Os	PA0 PA6 PB0 PB10 PA11	PA2 PA7 PB11 PA12 PB5
CMP outputs connected to EXTI	•	
CMP outputs connected to internal signals	TIMER0_CH0	TIMER0_CH1

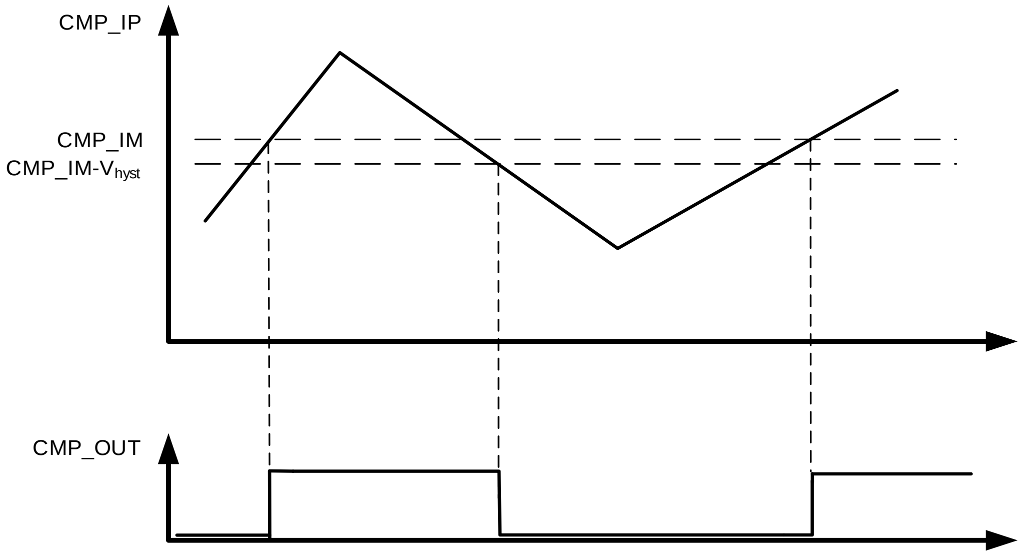
18.3.3. CMP operating mode

For a given application, there is a trade-off between the CMP power consumption versus propagation delay, which is adjusted by configuring bits CMPxM [1:0] in CMPx_CS register. The CMP works fastest with highest power consumption when CMPxM [1:0] = 2’b00, while works slowest with lowest power consumption when CMPxM [1:0] = 2’b11.

18.3.4. CMP hysteresis

In order to avoid spurious output transitions that caused by the noise signal, a programmable hysteresis is designed to force the hysteresis value by configuring CMPx_CS register. This function could be shut down if it is unnecessary.

Figure 18-2. CMP hysteresis



18.3.5. CMP register write protection

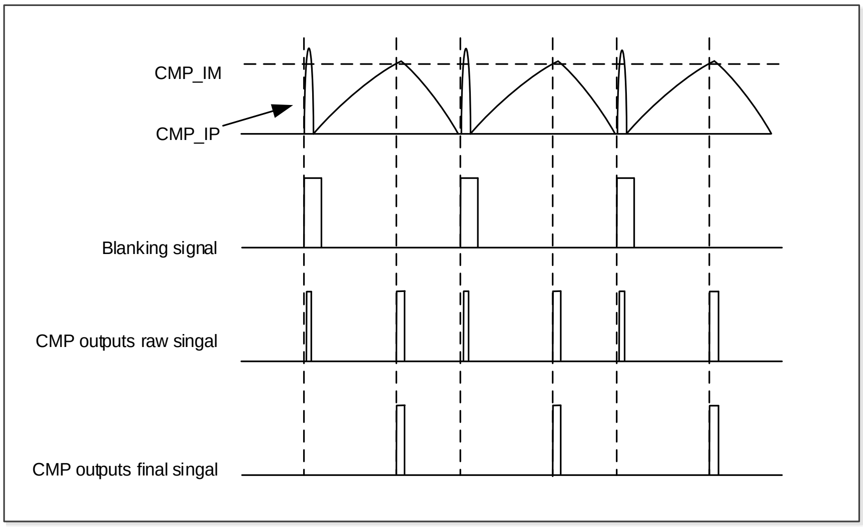
The CMP control and status register (CMPx_CS) can be protected from writing by setting CMPxLK bit to 1. The CMPx_CS register, including the CMPxLK bit will be read-only, and can only be reset by the MCU reset.

18.3.6. CMP output blanking

CMP output blanking function can be used to avoid interference of short pulses in the input signal to CMP output signal. If the CMPxBLK[2:0] bits in the CMPx_CS register are setting to an available value, the CMP output final signal is obtained by ANDing the complementary signal of the selected blanking signal with the raw output of the comparator. The blanking function can be used for false overcurrent detection in motor control applications.

Figure 18-3. The CMP outputs signal blanking shows the comparator output blank function.

Figure 18-3. The CMP outputs signal blanking



18.3.7. CMP voltage scaler function

The voltage scaler function can provide selectable 1 / 4, 1 / 2, 3 / 4 reference voltage for CMP input. It is controlled by CMPxSEN and CMPxBEN bits in CMPx control / status register. The CMPxSEN and CMPxBEN bits are used to enable the V_{REFINT} voltage output and the divider circuit, respectively, to generate the selected voltage.

18.3.8. CMP interrupt

The CMP output is connected to the EXTI and the EXTI line is exclusive to CMP. With this function, CMP can generate either interrupt or event which could be used to exit from low-power modes.

18.4. Register definition

CMP base address: 0x4001 7C00

18.4.1. CMP0 Control / Status register (CMP0_CS)

Address offset: 0x00

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CMP0LK	CMP0O	Reserved						CMP0SEN	CMP0BEN	Reserved	CMP0BLK[2:0]			CMP0HST[1:0]	
rwo	r							rw	rw		rw			rw	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CMP0PL	Reserved			CMP0SW					CMP0MSEL[2:0]			CMP0M[1:0]		Reserved	CMP0EN
rw				rw					rw			rw			rw

Bits	Fields	Descriptions
31	CMP0LK	CMP0 lock This bit can set all control bits of CMP0 as read-only. It can only be set once by software and cleared by a system reset. 0: CMP0_CS[31:0] bits are read-write 1: CMP0_CS[31:0] bits are read-only
30	CMP0O	CMP0 output state This bit is a copy of CMP0 output state, which is read only. 0: Non-inverting input below inverting input and the output is low 1: Non-inverting input above inverting input and the output is high
29:24	Reserved	Must be kept at reset value.
23	CMP0SEN	Voltage scaler enable bit This bit is set and cleared by software. This bit enables the outputs of the VREFINT divider, which is treated as the minus input of the CMP0. 0: Disable bandgap scaler in case that CMP1SEN bit of CMP1_CS is also reset 1: Enable bandgap scaler
22	CMP0BEN	Scaler bridge enable bit 0: Disable scaler resistor bridge in case that CMP1BEN bit of CMP1_CS is also reset 1: Enable scaler resistor bridge
21	Reserved	Must be kept at reset value.
20:18	CMP0BLK[2:0]	CMP0 output blanking source This bit is used to select which timer output controls the CMP0 output blanking. 000: No blanking

		001: Select TIMER0_CH1 output compare signal as blanking source 010: Select TIMER2_CH1 output compare signal as blanking source 011: Reserved 100: Select TIMER13_CH0 output compare signal as blanking source 101: Select TIMER15_CH0 output compare signal as blanking source 110~111: Reserved
17:16	CMP0HST[1:0]	CMP0 hysteresis These bits are used to control the hysteresis level. 00: No hysteresis 01: Low hysteresis 10: Medium hysteresis 11: High hysteresis
15	CMP0PL	Polarity of CMP0 output This bit is used to select the polarity of CMP0 output. 0: Output is not inverted 1: Output is inverted
14:12	Reserved	Must be kept at reset value.
11	CMP0SW	CMP0 switch This bit is used to close a switch between CMP0 non-inverting input on PA1 and PB2 I / O. 0: Switch open 1: Switch closed
10:7	Reserved	Must be kept at reset value.
6:4	CMP0MSEL[2:0]	CMP0_IM input selection These bits are used to select the source connected to the CMP0_IM input of the CMP0. 000: V_{REFINT} / 4 001: V_{REFINT} / 2 010: V_{REFINT} * 3 / 4 011: V_{REFINT} 100: PB2 101: PA0 110: PB1 111: VSSA
3:2	CMP0M[1:0]	CMP0 mode These bits are used to control the operating mode of the CMP0 adjust the speed / consumption. 00: High speed / full power 01: Medium speed / medium power 10: Low speed / low power

		11: Very-low speed / ultra-low power
1	Reserved	Must be kept at reset value.
0	CMP0EN	CMP0 enable 0: CMP0 disabled 1: CMP0 enabled

18.4.2. CMP1 Control / Status register (CMP1_CS)

Address offset: 0x04

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CMP1LK	CMP1O	Reserved						CMP1SEN	CMP1BEN	Reserved	CMP1BLK[2:0]			CMP1HST1:0]	
rwo	r							rw	rw		rw			rw	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CMP1PL	Reserved			CMP1SW	Reserved				CMP1MSEL[2:0]			CMP1M[1:0]		Reserved	CMP1EN
rw				rw					rw			rw			rw

Bits	Fields	Descriptions
31	CMP1LK	CMP1 lock This bit can set all control bits of CMP1 as read-only. It can only be set once by software and cleared by a system reset. 0: CMP1_CS[31:0] bits are read-write 1: CMP1_CS[31:0] bits are read-only
30	CMP1O	CMP1 output state This bit is a copy of CMP1 output state, which is read only. 0: Non-inverting input below inverting input and the output is low 1: Non-inverting input above inverting input and the output is high
29:24	Reserved	Must be kept at reset value.
23	CMP1SEN	Voltage scaler enable bit This bit is set and cleared by software. This bit enables the outputs of the V_{REFINT} divider, which is treated as the minus input of the CMP1. 0: Disable bandgap scaler in case that CMP0SEN bit of CMP0_CS is also reset 1: Enable bandgap scaler
22	CMP1BEN	Scaler bridge enable bit 0: Disable scaler resistor bridge in case that CMP0BEN bit of CMP0_CS is also reset 1: Enable scaler resistor bridge
21	Reserved	Must be kept at reset value.

20:18	CMP1BLK[2:0]	CMP1 output blanking source This bit is used to select which timer output controls the CMP1 output blanking. 000: No blanking 001: Select TIMER0_CH1 output compare signal as blanking source 010: Select TIMER2_CH1 output compare signal as blanking source 011: Reserved 100: Select TIMER13_CH0 output compare signal as blanking source 101: Select TIMER15_CH0 output compare signal as blanking source 110~111: Reserved
17:16	CMP1HST[1:0]	CMP1 hysteresis These bits are used to control the hysteresis level. 00: No hysteresis 01: Low hysteresis 10: Medium hysteresis 11: High hysteresis
15	CMP1PL	Polarity of CMP1 output This bit is used to select the polarity of CMP1 output. 0: Output is not inverted 1: Output is inverted
14:12	Reserved	Must be kept at reset value.
11	CMP1SW	CMP1 switch This bit is used to close a switch between CMP1 non-inverting input on PA3 and PB6 I / O. 0: Switch open 1: Switch closed
10:7	Reserved	Must be kept at reset value.
6:4	CMP1MSEL[2:0]	CMP1_IM input selection These bits are used to select the source connected to the CMP1_IM input of the CMP1. 000: V_{REFINT} / 4 001: V_{REFINT} / 2 010: V_{REFINT} * 3 / 4 011: V_{REFINT} 100: PB6 101: PA2 110: PB3 111: PB4
3:2	CMP1M[1:0]	CMP1 mode These bits are used to control the operating mode of the CMP1 adjust the speed / consumption. 00: High speed / full power

		01: Medium speed / medium power 10: Low speed / low power 11: Very-low speed / ultra-low power
1	Reserved	Must be kept at reset value.
0	CMP1EN	CMP1 enable. 0: CMP1 disabled 1: CMP1 enabled

19. Document appendix

19.1. List of abbreviations used in registers

Table 19-1. List of abbreviations used in register

abbreviations for registers	Descriptions
read/write (rw)	Software can read and write to this bit.
read-only (r)	Software can only read this bit.
write-only (w)	Software can only write to this bit. Reading this bit returns the reset value.
read/clear write 1 (rc_w1)	Software can read as well as clear this bit by writing 1. Writing 0 has no effect on the bit value.
read/clear write 0 (rc_w0)	Software can read as well as clear this bit by writing 0. Writing 1 has no effect on the bit value.
toggle (t)	Software can toggle this bit by writing 1. Writing 0 has no effect.
read-only/set by write 1 (rt_w1)	Software can only read to this bit. Writing 1 trigges the event but has no effect on the bit value.
read/set (rs)	Software can read as well as set this bit to 1. Writing 0 has no effect on the bit value.
read/clear by read (rc_r)	Software can read this bit. Reading this bit automatically clears it to '0'. Writing '0' has no effect on the bit value.
read/set by read (rs_r)	Software can read this bit, and set this bit by reading. Writing has no effect on the bit value.
read/write once (rwo)	Software can only write once to this bit and can also read it at any time. Only a reset can return the bit to its reset value.
read/clear write (rc_w)	Software can read as well as clear this bit by writing. Writing 0 or 1 has the same effect to this bit.
read-only/write trigger (rt_w)	Software can read this bit. Writing 0 or 1 triggers an event but has no effect on the bit value.

19.2. List of terms

Table 19-2. List of terms

Glossary	Descriptions
Word	Data of 32-bit length.
Half-word	Data of 16-bit length.
Byte	Data of 8-bit length.
IAP (in-application programming)	Writing 0 has no effect IAP is the ability to re-program the Flash memory of a microcontroller while the user program is running.

Glossary	Descriptions
ICP (in-circuit programming)	ICP is the ability to program the Flash memory of a microcontroller using the JTAG protocol, the SWD protocol or the boot loader while the device is mounted on the user application board.
Option bytes	Product configuration bits stored in the Flash memory.
AHB	Advanced high-performance bus.
APB	Advanced peripheral bus.
RAZ	Read-as-zero.
WI	Writes ignored.
RAZ/WI	Read-as-zero, writes ignored.

19.3. Available peripherals

For availability of peripherals and their number across all MCU series types,refer to the corresponding device data datasheet.

20. Revision history

Table 20-1. Revision history

Revision No.	Description	Date
1.0	Initial Release	Jun.03, 2025

Important Notice

This document is the property of GigaDevice Semiconductor Inc. and its subsidiaries (the "Company"). This document, including any product of the Company described in this document (the “Product”), is owned by the Company according to the laws of the People’s Republic of China and other applicable laws. The Company reserves all rights under such laws and no Intellectual Property Rights are transferred (either wholly or partially) or licensed by the Company (either expressly or impliedly) herein. The names and brands of third party referred thereto (if any) are the property of their respective owner and referred to for identification purposes only.

To the maximum extent permitted by applicable law, the Company makes no representations or warranties of any kind, express or implied, with regard to the merchantability and the fitness for a particular purpose of the Product, nor does the Company assume any liability arising out of the application or use of any Product. Any information provided in this document is provided only for reference purposes. It is the sole responsibility of the user of this document to determine whether the Product is suitable and fit for its applications and products planned, and properly design, program, and test the functionality and safety of its applications and products planned using the Product. The Product is designed, developed, and/or manufactured for ordinary business, industrial, personal, and/or household applications only, and the Product is not designed or intended for use in (i) safety critical applications such as weapons systems, nuclear facilities, atomic energy controller, combustion controller, aeronautic or aerospace applications, traffic signal instruments, pollution control or hazardous substance management; (ii) life-support systems, other medical equipment or systems (including life support equipment and surgical implants); (iii) automotive applications or environments, including but not limited to applications for active and passive safety of automobiles (regardless of front market or aftermarket), for example, EPS, braking, ADAS (camera/fusion), EMS, TCU, BMS, BSG, TPMS, Airbag, Suspension, DMS, ICMS, Domain, ESC, DCDC, e-clutch, advanced-lighting, etc.. Automobile herein means a vehicle propelled by a self-contained motor, engine or the like, such as, without limitation, cars, trucks, motorcycles, electric cars, and other transportation devices; and/or (iv) other uses where the failure of the device or the Product can reasonably be expected to result in personal injury, death, or severe property or environmental damage (collectively "Unintended Uses"). Customers shall take any and all actions to ensure the Product meets the applicable laws and regulations. The Company is not liable for, in whole or in part, and customers shall hereby release the Company as well as its suppliers and/or distributors from, any claim, damage, or other liability arising from or related to all Unintended Uses of the Product. Customers shall indemnify and hold the Company, and its officers, employees, subsidiaries, affiliates as well as its suppliers and/or distributors harmless from and against all claims, costs, damages, and other liabilities, including claims for personal injury or death, arising from or related to any Unintended Uses of the Product.

Information in this document is provided solely in connection with the Product. The Company reserves the right to make changes, corrections, modifications or improvements to this document and the Product described herein at any time without notice. The Company shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them. Information in this document supersedes and replaces information previously supplied in any prior versions of this document.