GigaDevice Semiconductor Inc.

GD32L23x

Arm® Cortex®-M23 32-bit MCU

For GD32L233xx, GD32L235x

User Manual

Revision 2.2

(Mar. 2024)

Table of Contents

List of Figures

List of Tables

1. System and memory architecture

The GD32L23x 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.

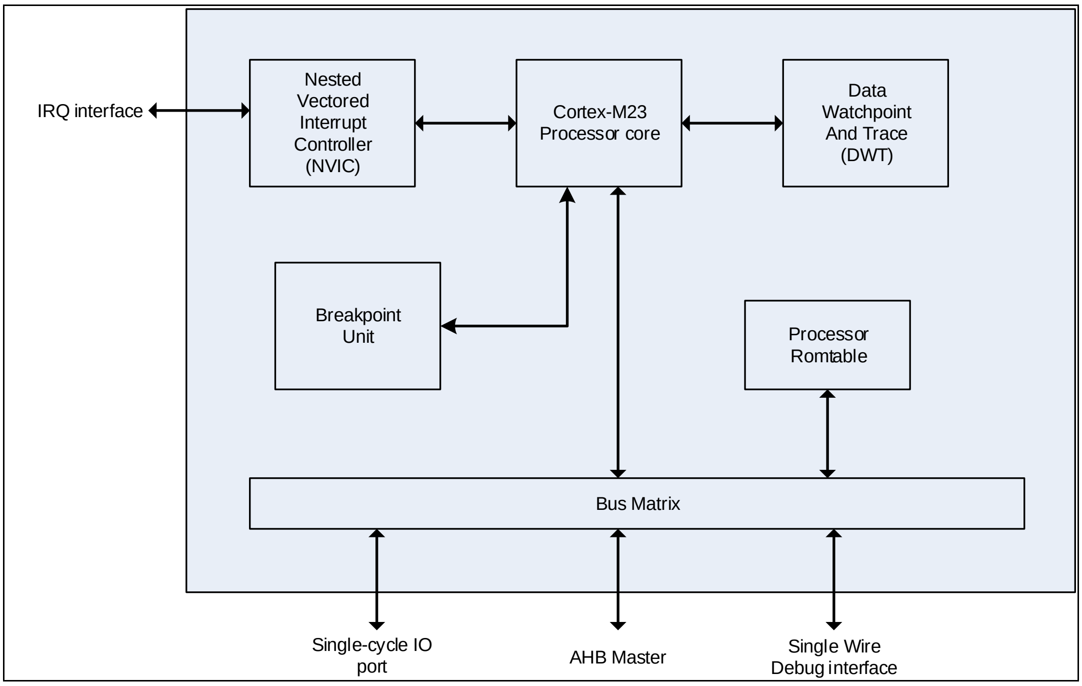
The Arm® Cortex®-M23 processor closely integrates a configurable Nested Vectored Interrupt Controller (NVIC), to deliver industry-leading interrupt performance.

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



1.2. System architecture

The Bus Matrix is implemented in the GD32L23x 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, tow 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
SRAM0	1	1
AHB1	1	1
AHB2	1	1
SRAM1	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 SRAMs, Flash memory and AHB/APB peripherals.

There are also several slaves connected with the AHB interconnect matrix, including FMC, SRAM0, SRAM1, AHB1, AHB2. FMC is the bus interface of the flash memory controller. SRAM0~ SRAM1 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. APB1 is limited to 32Mhz, APB2 is limited to 64Mhz.

The system architecture of GD32L23x 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 five slaves, including the flash memory controller, internal SRAM0, internal SRAM1, AHB1 and AHB2.

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

Figure 1-2. Series system architecture of GD32L23x(x=3) series

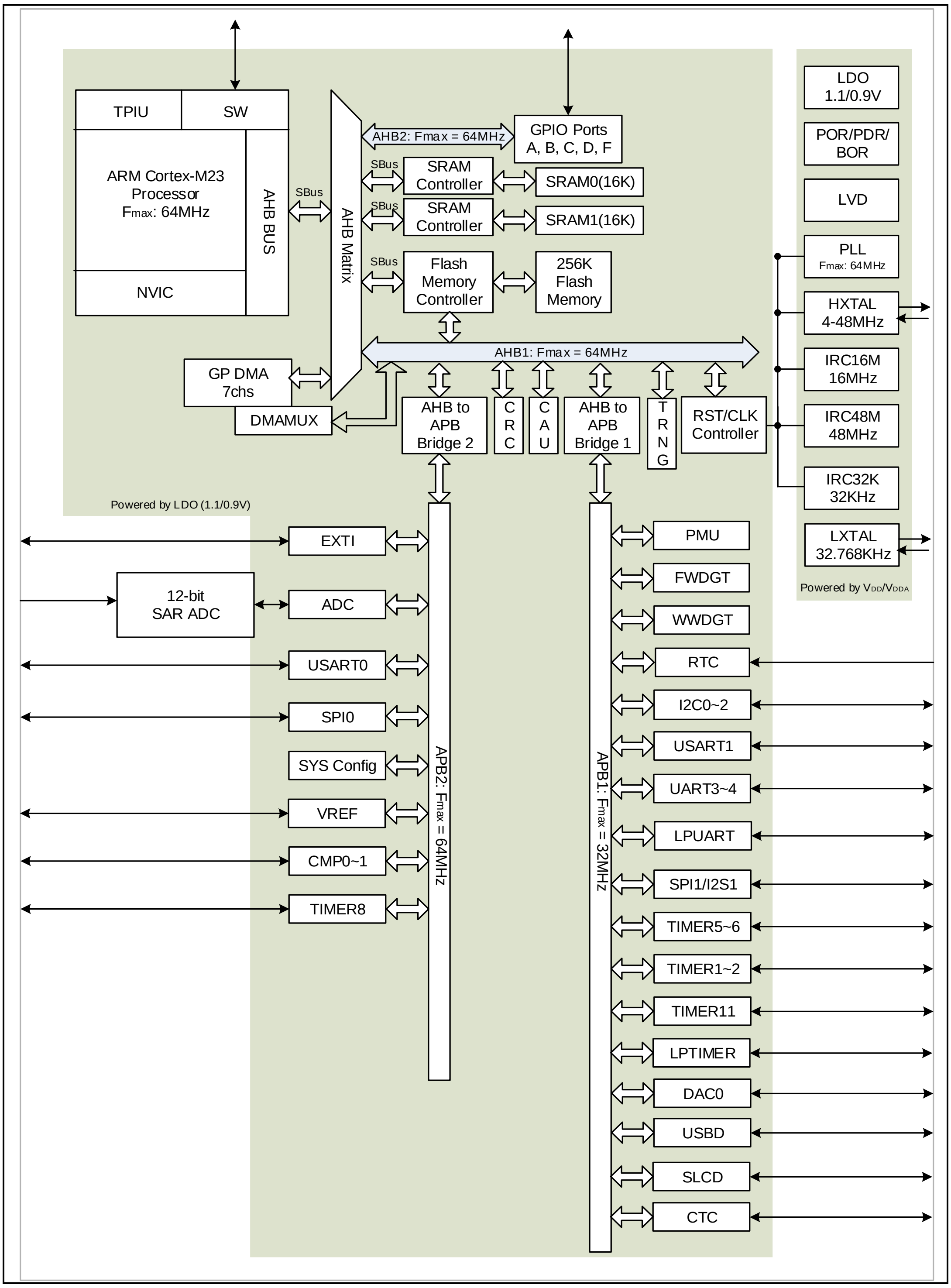
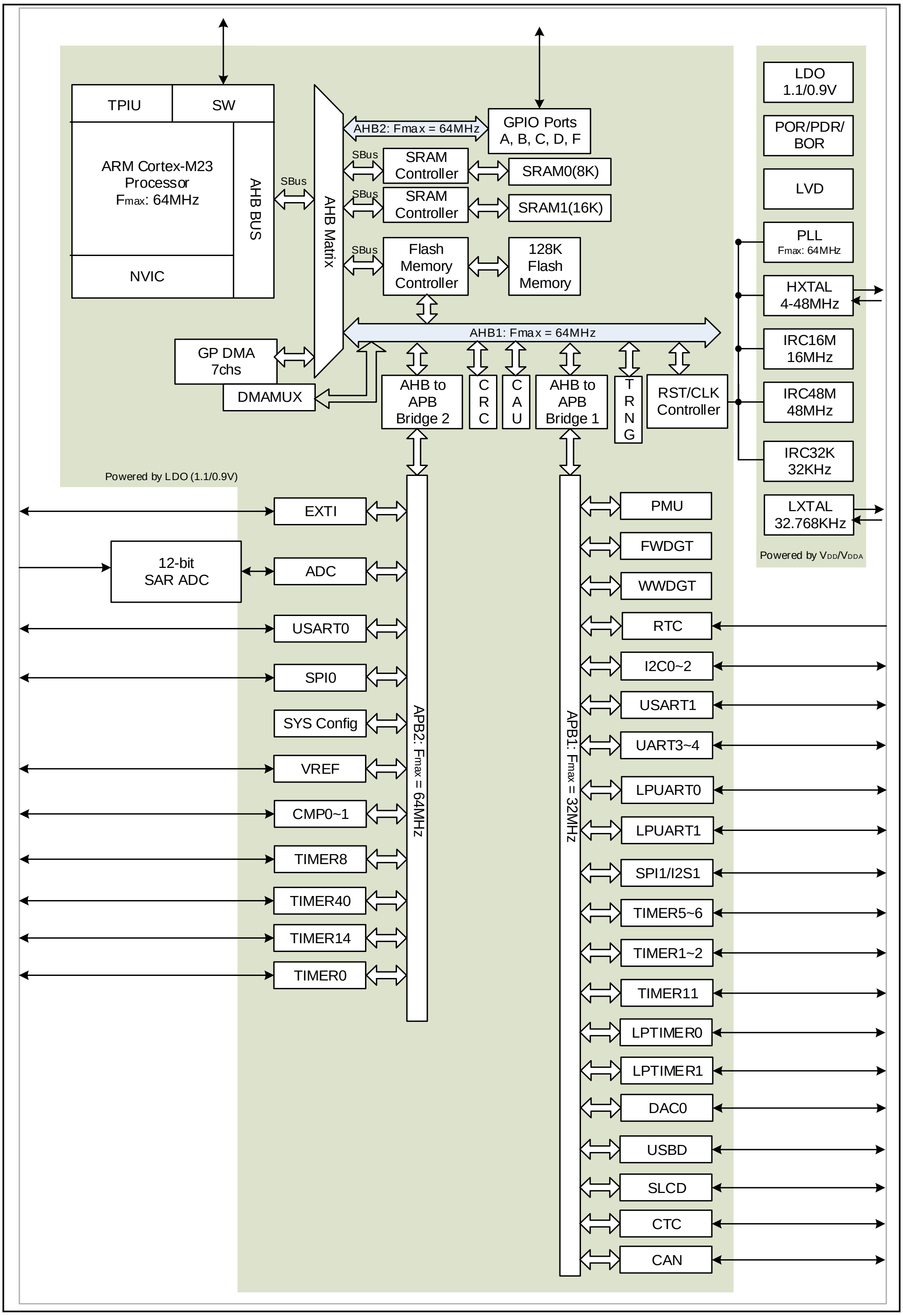


Figure 1-3. Series system architecture of GD32L23x(x=5) series



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 GD32L23x 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 GD32L23x(x=3) 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	AHB1	0x5006 1000 – 0x5FFF FFFF	Reserved
		0x5006 0C00 – 0x5006 0FFF	Reserved
		0x5006 0800 – 0x5006 0BFF	TRNG
		0x5006 0400 – 0x5006 07FF	Reserved
		0x5006 0000 – 0x5006 03FF	CAU
		0x5005 0400 – 0x5005 FFFF	Reserved
		0x5005 0000 – 0x5005 03FF	Reserved
		0x5004 0000 – 0x5004 FFFF	Reserved
		0x5000 0000 – 0x5003 FFFF	Reserved
	AHB2	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	0x4002 4400 – 0x47FF FFFF	Reserved
		0x4002 4000 – 0x4002 43FF	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

Pre-defined Regions	Bus	ADDRESS	Peripherals
	APB2	0x4001 8000 – 0x4001 FFFF	Reserved
		0x4001 7C00 – 0x4001 7FFF	CMP
		0x4001 5C00 – 0x4001 7BFF	Reserved
		0x4001 5800 – 0x4001 5BFF	DBG
		0x4001 5000 – 0x4001 57FF	Reserved
		0x4001 4C00 – 0x4001 4FFF	TIMER8
		0x4001 3C00 – 0x4001 4BFF	Reserved
		0x4001 3800 – 0x4001 3BFF	USART0
		0x4001 3400 – 0x4001 37FF	Reserved
		0x4001 3000 – 0x4001 33FF	SPI0
		0x4001 2C00 – 0x4001 2FFF	Reserved
		0x4001 2800 – 0x4001 2BFF	Reserved
		0x4001 2400 – 0x4001 27FF	ADC
		0x4001 0800 – 0x4001 23FF	Reserved
		0x4001 0400 – 0x4001 07FF	EXTI
		0x4001 0000 – 0x4001 03FF	SYSCFG + VREF
	APB1	0x4000 CC00 – 0x4000 FFFF	Reserved
		0x4000 C800 – 0x4000 CBFF	CTC
		0x4000 C400 – 0x4000 C7FF	Reserved
		0x4000 C000 – 0x4000 C3FF	I2C2
		0x4000 9800 – 0x4000 BFFF	Reserved
		0x4000 9400 – 0x4000 97FF	LPTIMER
		0x4000 8400 – 0x4000 93FF	Reserved
		0x4000 8000 – 0x4000 83FF	LPUART
		0x4000 7C00 – 0x4000 7FFF	Reserved
		0x4000 7800 – 0x4000 7BFF	Reserved
		0x4000 7400 – 0x4000 77FF	DAC0
		0x4000 7000 – 0x4000 73FF	PMU
		0x4000 6400 – 0x4000 6FFF	Reserved
		0x4000 6000 – 0x4000 63FF	USBD RAM (512 bytes)
		0x4000 5C00 – 0x4000 5FFF	USBD
		0x4000 5800 – 0x4000 5BFF	I2C1
		0x4000 5400 – 0x4000 57FF	I2C0
		0x4000 5000 – 0x4000 53FF	UART4
		0x4000 4C00 – 0x4000 4FFF	UART3
		0x4000 4800 – 0x4000 4BFF	Reserved
		0x4000 4400 – 0x4000 47FF	USART1
		0x4000 4000 – 0x4000 43FF	Reserved
		0x4000 3C00 – 0x4000 3FFF	Reserved
		0x4000 3800 – 0x4000 3BFF	SPI1/I2S1

Pre-defined Regions	Bus	ADDRESS	Peripherals
		0x4000 3400 – 0x4000 37FF	Reserved
		0x4000 3000 – 0x4000 33FF	FWDGT
		0x4000 2C00 – 0x4000 2FFF	WWDGT
		0x4000 2800 – 0x4000 2BFF	RTC
		0x4000 2400 – 0x4000 27FF	SLCD
		0x4000 2000 – 0x4000 23FF	Reserved
		0x4000 1C00 – 0x4000 1FFF	Reserved
		0x4000 1800 – 0x4000 1BFF	TIMER11
		0x4000 1400 – 0x4000 17FF	TIMER6
		0x4000 1000 – 0x4000 13FF	TIMER5
		0x4000 0800 – 0x4000 0FFF	Reserved
		0x4000 0400 – 0x4000 07FF	TIMER2
		0x4000 0000 – 0x4000 03FF	TIMER1
		0x4000 0000 – 0x4000 03FF	Reserved
SRAM		0x2000 8000 – 0x3FFF FFFF	Reserved
		0x2000 5000 – 0x2000 7FFF	SRAM1(16KB)
		0x2000 4000 – 0x2000 4FFF	
		0x2000 2000 – 0x2000 3FFF	SRAM0(16KB)
		0x2000 1000 – 0x2000 1FFF	
		0x2000 0000 – 0x2000 0FFF	
Code		0x1FFF F810 – 0x1FFF FFFF	Reserved
		0x1FFF F800 – 0x1FFF F80F	Option bytes(16B)
		0x1FFF D000- 0x1FFF F7FF	System memory(10KB)
		0x1FFF 7200 – 0x1FFF CFFF	Reserved
		0x1FFF 7000 – 0x1FFF 71FF	OTP(512B)
		0x1000 0000 – 0x1FFF 6FFF	Reserved
		0x0804 0000 – 0x0FFF FFFF	Reserved
		0x0802 0000 – 0x0803 FFFF	Main Flash memory(256KB)
		0x0801 0000 – 0x0801 FFFF	
		0x0800 0000 – 0x0800 FFFF	
		0x0004 0000 – 0x07FF FFFF	Reserved
		0x0001 0000 – 0x0003 FFFF	Aliased to Flash or system memory
		0x0000 0000 – 0x0000 FFFF	

Table 1-3. Memory map of GD32L23x(x=5) 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	AHB1	0x5006 1000 – 0x5FFF FFFF	Reserved

Pre-defined Regions	Bus	ADDRESS	Peripherals
		0x5006 0C00 – 0x5006 0FFF	Reserved
		0x5006 0800 – 0x5006 0BFF	TRNG
		0x5006 0400 – 0x5006 07FF	Reserved
		0x5006 0000 – 0x5006 03FF	CAU
		0x5005 0400 – 0x5005 FFFF	Reserved
		0x5005 0000 – 0x5005 03FF	Reserved
		0x5004 0000 – 0x5004 FFFF	Reserved
		0x5000 0000 – 0x5003 FFFF	Reserved
	AHB2	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	0x4002 4400 – 0x47FF FFFF	Reserved
		0x4002 4000 – 0x4002 43FF	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
	APB2	0x4001 8000 – 0x4001 FFFF	Reserved
		0x4001 D000 – 0x4001 D3FF	TIMER40
		0x4001 8000 – 0x4001 CFFF	Reserved
		0x4001 7C00 – 0x4001 7FFF	CMP
		0x4001 5C00 – 0x4001 7BFF	Reserved
		0x4001 5800 – 0x4001 5BFF	DBG
		0x4001 5000 – 0x4001 57FF	Reserved
		0x4001 4C00 – 0x4001 4FFF	TIMER8
		0x4001 4400 – 0x4001 4BFF	Reserved
		0x4001 4000 – 0x4001 43FF	TIMER14
		0x4001 3C00 – 0x4001 3FFF	Reserved
		0x4001 3800 – 0x4001 3BFF	USART0
		0x4001 3400 – 0x4001 37FF	Reserved

Pre-defined Regions	Bus	ADDRESS	Peripherals
		0x4001 3000 – 0x4001 33FF	SPI0
		0x4001 2C00 – 0x4001 2FFF	TIMER0
		0x4001 2800 – 0x4001 2BFF	Reserved
		0x4001 2400 – 0x4001 27FF	ADC
		0x4001 0800 – 0x4001 23FF	Reserved
		0x4001 0400 – 0x4001 07FF	EXTI
		0x4001 0000 – 0x4001 03FF	SYSCFG + VREF
	APB1	0x4000 CC00 – 0x4000 FFFF	Reserved
		0x4000 C800 – 0x4000 CBFF	CTC
		0x4000 C400 – 0x4000 C7FF	Reserved
		0x4000 C000 – 0x4000 C3FF	I2C2
		0x4000 9800 – 0x4000 BFFF	Reserved
		0x4000 9400 – 0x4000 97FF	LPTIMER0
		0x4000 8400 – 0x4000 93FF	Reserved
		0x4000 8000 – 0x4000 83FF	LPUART0
		0x4000 7C00 – 0x4000 7FFF	LPTIMER1
		0x4000 7800 – 0x4000 7BFF	Reserved
		0x4000 7400 – 0x4000 77FF	DAC0
		0x4000 7000 – 0x4000 73FF	PMU
		0x4000 6800 – 0x4000 6FFF	Reserved
		0x4000 6400 – 0x4000 67FF	CAN
		0x4000 6000 – 0x4000 63FF	Shared USBD/CAN RAM (512 bytes)
		0x4000 5C00 – 0x4000 5FFF	USBD
		0x4000 5800 – 0x4000 5BFF	I2C1
		0x4000 5400 – 0x4000 57FF	I2C0
		0x4000 5000 – 0x4000 53FF	UART4
		0x4000 4C00 – 0x4000 4FFF	UART3
		0x4000 4800 – 0x4000 4BFF	LPUART1
		0x4000 4400 – 0x4000 47FF	USART1
		0x4000 4000 – 0x4000 43FF	Reserved
		0x4000 3C00 – 0x4000 3FFF	Reserved
		0x4000 3800 – 0x4000 3BFF	SPI1/I2S1
		0x4000 3400 – 0x4000 37FF	Reserved
		0x4000 3000 – 0x4000 33FF	FWDGT
		0x4000 2C00 – 0x4000 2FFF	WWDGT
		0x4000 2800 – 0x4000 2BFF	RTC
		0x4000 2400 – 0x4000 27FF	SLCD
		0x4000 2000 – 0x4000 23FF	Reserved
		0x4000 1C00 – 0x4000 1FFF	Reserved
		0x4000 1800 – 0x4000 1BFF	TIMER11

Pre-defined Regions	Bus	ADDRESS	Peripherals
		0x4000 1400 – 0x4000 17FF	TIMER6
		0x4000 1000 – 0x4000 13FF	TIMER5
		0x4000 0800 – 0x4000 0FFF	Reserved
		0x4000 0400 – 0x4000 07FF	TIMER2
		0x4000 0000 – 0x4000 03FF	TIMER1
		0x4000 0000 – 0x4000 03FF	Reserved
SRAM		0x2000 8000 – 0x3FFF FFFF	Reserved
		0x2000 6000 – 0x2000 7FFF	Reserved
		0x2000 4000 – 0x2000 5FFF	SRAM1(16KB)
		0x2000 2000 – 0x2000 3FFF	
		0x2000 1000 – 0x2000 1FFF	SRAM0(8KB)
		0x2000 0000 – 0x2000 0FFF	
Code		0x1FFF F810 – 0x1FFF FFFF	Reserved
		0x1FFF F800 – 0x1FFF F80F	Option bytes(16B)
		0x1FFF D000- 0x1FFF F7FF	System memory(10KB)
		0x1FFF 7200 – 0x1FFF CFFF	Reserved
		0x1FFF 7000 – 0x1FFF 71FF	OTP(512B)
		0x1000 0000 – 0x1FFF 6FFF	Reserved
		0x0804 0000 – 0x0FFF FFFF	Reserved
		0x0802 0000 – 0x0803 FFFF	Reserved
		0x0801 0000 – 0x0801 FFFF	Main Flash memory(128KB)
		0x0800 0000 – 0x0800 FFFF	
		0x0004 0000 – 0x07FF FFFF	Reserved
		0x0001 0000 – 0x0003 FFFF	Aliased to Flash or system memory
		0x0000 0000 – 0x0000 FFFF	

1.3.1. On-chip SRAM memory

The GD32L23x series contain up to 32KB of on-chip SRAM which starts at the address 0x2000 0000. It supports byte, half-word (16 bits), and word (32 bits) accesses.

In GD32L235xx devices, parity check is implemented on the SRAM. An NMI interrupt is triggered if a parity error occurs.

1.3.2. On-chip Flash memory

The devices provide up to 256 KB of on-chip flash memory. Refer to ***Flash memory architecture*** for flash module organization.

All of, byte, half-word (16 bits) and word (32 bits) read accesses are supported. The flash memory can be programmed word (32 bits) and half-word (16 bits). Each page of the flash memory can be erased individually. The whole flash memory space except information blocks can be erased at a time.

1.4. Boot configuration

The GD32L23x series provide three kinds of boot sources which can be selected by the BOOT0 and BOOT1 pins. The details are shown in the following table. The value on the two pins is latched on the 4th rising edge of CK_SYS after a reset. It is up to the user to set the BOOT0 and BOOT1 pins after a power-on reset or a system reset to select the required boot source. Once the two pins have been sampled, they are free and can be used for other purposes.

Table1-4. Boot modes

Selected boot source	Boot mode selection pins	
	Boot1	Boot0
Main Flash Memory	x	0
System Memory	0	1
On-chip SRAM	1	1

After power-on sequence or a system reset, the Arm® Cortex®-M23 processor fetches the top-of-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 D000) 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. The boot loader can be activated through one of the following interfaces: USART0, USART1 or USBD.

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)

For GD32L233xx devices

Address offset: 0x00

Reset value: 0x0000 000X (X indicates BOOT_MODE[1:0] may be any value according to the BOOT0 pin and the BOOT1 pin after reset)

This register can be accessed by word(32-bit)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved												PB9_HCCE	PB8_HCCE	PB7_HCCE	PB6_HCCE
												rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved									BOOT0_PD3_RMP	Reserved	PA11_PA12_RMP	Reserved		BOOT_MODE[1:0]	
									rw		rw			r	

Bits	Fields	Descriptions
31:20	Reserved	Must be kept at reset value
19	PB9_HCCE	PB9 pin high current capability enable When it is set, the PB9 pin can be used to control an infrared LED directly. 0: High current capability on the PB9 pin is disabled. 1: High current capability on the PB9 pin is enabled, and the speed control of the pin is bypassed.
18	PB8_HCCE	PB8 pin high current capability enable When it is set, the PB8 pin can be used to control an infrared LED directly. 0: High current capability on the PB8 pin is disabled. 1: High current capability on the PB8 pin is enabled, and the speed control of the pin is bypassed.
17	PB7_HCCE	PB7 pin high current capability enable When it is set, the PB7 pin can be used to control an infrared LED directly. 0: High current capability on the PB7 pin is disabled. 1: High current capability on the PB7 pin is enabled, and the speed control of the pin is bypassed.
16	PB6_HCCE	PB6 pin high current capability enable When it is set, the PB6 pin can be used to control an infrared LED directly.

		0: High current capability on the PB6 pin is disabled. 1: High current capability on the PB6 pin is enabled, and the speed control of the pin is bypassed.
15:7	Reserved	Must be kept at reset value
6	BOOT0_PD3_RMP	BOOT0 and PD3 remapping bit. It controls the mapping of either BOOT0 or PD3 function on the BOOT0 pin. When BOOT0_PD3_RMP is set, the BOOT0 function is tied to 0 by hardware. In this case, the system will boot from main flash without regard to the input value from the BOOT0 pin. 0: No remap (BOOT0 function is mapping on the BOOT0 pin) 1: Remap (PD3 function is mapping on the BOOT0 pin)
5	Reserved	Must be kept at reset value
4	PA11_PA12_RMP	PA11 and PA12 remapping bit for small packages (28 and 20 pins). This bit is set and cleared by software. It controls the mapping of either PA9/10 or PA11/12 pin pair on small pin-count packages. 0: No remap (pin pair PA9/10 mapped on the pins) 1: Remap (pin pair PA11/12 mapped instead of PA9/10)
3:2	Reserved	Must be kept at reset value
1:0	BOOT_MODE[1:0]	Boot mode Bit0 is mapping to the BOOT0 pin; the value of bit1 is mapping to the BOOT1 pin. X0: Boot from the Main Flash 01: Boot from the System Flash memory 11: Boot from the embedded SRAM

For GD32L235xx devices

Address offset: 0x00

Reset value: 0x0000 000X (X indicates BOOT_MODE[1:0] may be any value according to the BOOT0 pin and the BOOT1 pin after reset)

This register can be accessed by word(32-bit)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved												PB9_HCCE	PB8_HCCE	PB7_HCCE	PB6_HCCE
												rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								PA8_RMP	BOOT0_PD3_RMP	PD0_PD1_PD2_RMP	PA11_PA12_RMP	PA11_PA12_PB6_PB8_RMP	Reserved	BOOT_MODE[1:0]	
								rw	rw	rw	rw	rw		r	

Bits	Fields	Descriptions
31:20	Reserved	Must be kept at reset value
19	PB9_HCCE	PB9 pin high current capability enable When it is set, the PB9 pin can be used to control an infrared LED directly. 0: High current capability on the PB9 pin is disabled. 1: High current capability on the PB9 pin is enabled, and the speed control of the pin is bypassed.
18	PB8_HCCE	PB8 pin high current capability enable When it is set, the PB8 pin can be used to control an infrared LED directly. 0: High current capability on the PB8 pin is disabled. 1: High current capability on the PB8 pin is enabled, and the speed control of the pin is bypassed.
17	PB7_HCCE	PB7 pin high current capability enable When it is set, the PB7 pin can be used to control an infrared LED directly. 0: High current capability on the PB7 pin is disabled. 1: High current capability on the PB7 pin is enabled, and the speed control of the pin is bypassed.
16	PB6_HCCE	PB6 pin high current capability enable When it is set, the PB6 pin can be used to control an infrared LED directly. 0: High current capability on the PB6 pin is disabled. 1: High current capability on the PB6 pin is enabled, and the speed control of the pin is bypassed.
15:8	Reserved	Must be kept at reset value
7	PA8_RMP	PA8 remapping bit. It controls the mapping of PA8 to PC14. 0: No remap 1: Remap (PA8 function is mapping on the PC14 pin)
6	BOOT0_PD3_RMP	BOOT0 and PD3 remapping bit. It controls the mapping of either BOOT0 or PD3 function on the BOOT0 pin. When BOOT0_PD3_RMP is set, the BOOT0 function is tied to 0 by hardware. In this case, the system will boot from main flash without regard to the input value from the BOOT0 pin. 0: No remap (BOOT0 function is mapping on the BOOT0 pin) 1: Remap (PD3 function is mapping on the BOOT0 pin)
5	PD0_PD1_PD2_RMP	It controls the mapping of either PD0/PD1/PD2 or PC10/PC11/PC12 pin pair on small pin-count packages. 0: No remap (pin pair mapped on the PC10/PC11/PC12 pins) 1: Remap (pin pair PD0/PD1/PD2 mapped instead of PC10/PC11/PC12)
4	PA11_PA12_RMP	PA11 and PA12 remapping bit for small packages (28 and 20 pins).

		This bit is set and cleared by software. It controls the mapping of either PA9/10 or PA11/12 pin pair on small pin-count packages. 0: No remap (pin pair PA9/10 mapped on the pins) 1: Remap (pin pair PA11/12 mapped instead of PA9/10)
3	PA11_PA12_PB6_PB8_RMP	It controls the mapping of either PA11/PA12/PB6/PB8 or PB3/PB4/PB2/PD3 pin pair on small pin-count packages(WLCSP25). 0: No remap 1: Remap (pin pair PA11/PA12/PB6/PB8 mapped instead of PB3/PB4/PB2/PD3)
2	Reserved	Must be kept at reset value
1:0	BOOT_MODE[1:0]	Boot mode (Refer to ***Boot configuration***) Bit0 is mapping to the BOOT0 pin; the value of bit1 is mapping to the BOOT1 pin. 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

This register can 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
EXTI3_SS [3:0]				EXTI2_SS [3:0]				EXTI1_SS [3:0]				EXTI0_SS [3:0]			
rw				rw				rw				rw			

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value
15:12	EXTI3_SS[3:0]	EXTI 3 sources selection X000: PA3 pin X001: PB3 pin X010: PC3 pin X011: PD3 pin X100: reserved X101: reserved X110: reserved X111: reserved
11:8	EXTI2_SS[3:0]	EXTI 2 sources selection X000: PA2 pin

		X001: PB2 pin X010: PC2 pin X011: PD2 pin X100: reserved X101: reserved X110: reserved X111: reserved
7:4	EXTI1_SS[3:0]	EXTI 1 sources selection X000: PA1 pin X001: PB1 pin X010: PC1 pin X011: PD1 pin X100: reserved X101: PF1 pin X110: reserved X111: reserved
3:0	EXTI0_SS[3:0]	EXTI 0 sources selection X000: PA0 pin X001: PB0 pin X010: PC0 pin X011: PD0 pin X100: reserved X101: PF0 pin X110: reserved X111: reserved

1.6.3. EXTI sources selection register 1 (SYSCFG_EXTISS1)

Address offset: 0x0C

Reset value: 0x0000 0000

This register can 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
EXTI7_SS [3:0]				EXTI6_SS [3:0]				EXTI5_SS [3:0]				EXTI4_SS [3:0]			
rw				rw				rw				rw			

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value
15:12	EXTI7_SS[3:0]	EXTI 7 sources selection

		X000: PA7 pin X001: PB7 pin X010: PC7 pin X011: reserved X100: reserved X101: reserved X110: reserved X111: reserved
11:8	EXTI6_SS[3:0]	EXTI 6 sources selection X000: PA6 pin X001: PB6 pin X010: PC6 pin X011: PD6 pin X100: reserved X101: reserved X110: reserved X111: reserved
7:4	EXTI5_SS[3:0]	EXTI 5 sources selection X000: PA5 pin X001: PB5 pin X010: PC5 pin X011: PD5 pin X100: reserved X101: reserved X110: reserved X111: reserved
3:0	EXTI4_SS[3:0]	EXTI 4 sources selection X000: PA4 pin X001: PB4 pin X010: PC4 pin X011: PD4 pin X100: reserved X101: reserved X110: reserved X111: reserved

1.6.4. EXTI sources selection register 2 (SYSCFG_EXTISS2)

Address offset: 0x10

Reset value: 0x0000 0000

This register can 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
EXTI11_SS [3:0]				EXTI10_SS [3:0]				EXTI9_SS [3:0]				EXTI8_SS [3:0]			
rw				rw				rw				rw			

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value
15:12	EXTI11_SS[3:0]	EXTI 11 sources selection X000: PA11 pin X001: PB11 pin X010: PC11 pin X011: reserved X100: reserved X101: reserved X110: reserved X111: reserved
11:8	EXTI10_SS[3:0]	EXTI 10 sources selection X000: PA10 pin X001: PB10 pin X010: PC10 pin X011: reserved X100: reserved X101: reserved X110: reserved X111: reserved
7:4	EXTI9_SS[3:0]	EXTI 9 sources selection X000: PA9 pin X001: PB9 pin X010: PC9 pin X011: PD9 pin X100: reserved X101: reserved X110: reserved X111: reserved
3:0	EXTI8_SS[3:0]	EXTI 8 sources selection X000: PA8 pin X001: PB8 pin X010: PC8 pin X011: PD8 pin

X100: reserved

X101: reserved

X110: reserved

X111: reserved

1.6.5. EXTI sources selection register 3 (SYSCFG_EXTISS3)

Address offset: 0x14

Reset value: 0x0000 0000

This register can 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
EXTI15_SS [3:0]				EXTI14_SS [3:0]				EXTI13_SS [3:0]				EXTI12_SS [3:0]			
rw				rw				rw				rw			

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value
15:12	EXTI15_SS[3:0]	EXTI 15 sources selection X000: PA15 pin X001: PB15 pin X010: PC15 pin X011: reserved X100: reserved X101: reserved X110: reserved X111: reserved
11:8	EXTI14_SS[3:0]	EXTI 14 sources selection X000: PA14 pin X001: PB14 pin X010: PC14 pin X011: reserved X100: reserved X101: reserved X110: reserved X111: reserved
7:4	EXTI13_SS[3:0]	EXTI 13 sources selection X000: PA13 pin X001: PB13 pin X010: PC13 pin X011: reserved

		X100: reserved X101: reserved X110: reserved X111: reserved
3:0	EXTI12_SS[3:0]	EXTI 12 sources selection X000: PA12 pin X001: PB12 pin X010: PC12 pin X011: reserved X100: reserved X101: reserved X110: reserved X111: reserved

1.6.6. System configuration register 1 (SYSCFG_CFG1)

Only for GD32L235xx devices

Address offset: 0x18

Reset value: 0x0000 0080

This register can 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							SRAM_PCEF	SRAM_LP_WATISTATE	Reserved				LVD_LOCK	SRAM_PARITY_ERROR_LOCK	LOCKUP_LOCK
							rc_w1	rw					rw	rw	rw

Bits	Fields	Descriptions
31:9	Reserved	Must be kept at reset value
8	SRAM_PCEF	SRAM parity check error flag This bit is set by hardware when an SRAM parity check error occurs. It is cleared by software by writing 1. 0: No SRAM parity check error detected 1: SRAM parity check error detected
7	SRAM_LP_WATISTATE	SRAM wait state configurate This bit is set and cleared by software when the LDO is 1.1V. It can only be reset by a system reset. This bit is used to insert wait state in read accesses of SRAM0/SRAM1.

		0: No wait-state 1: Insert wait-state
6:3	Reserved	Must be kept at reset value
2	LVD_LOCK	LVD lock This bit is set by software and cleared by a system reset. 0: The LVD interrupt is disconnected from the break input of TIMER0/14/40. LVDEN and LVDT[2:0] in the PMU_CTL register can be programmed. 1: The LVD interrupt is connected from the break input of TIMER0/14/40. LVDEN and LVDT[2:0] in the PMU_CTL register are read only.
1	SRAM_PARITY_ERROR_LOCK	SRAM parity check error lock This bit is set by software and cleared by a system reset. 0: The SRAM parity check error is disconnected from the break input of TIMER0/14/40 1: The SRAM parity check error is connected from the break input of TIMER0/14/40.
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 TIMER0/14/40. 1: The Cortex-M23 LOCKUP output is connected from the break input of TIMER0/14/40.

1.6.7. IRQ latency register (SYSCFG_CPU_IRQ_LAT)

Address offset: 0x100

Reset value: 0x0000 0000

This register can 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								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 Arm® 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.

1.6.8. TIMERx configuration register (SYSCFG_TIMERxCFG, x=0)

Only for GD32L235xx devices

Address offset: 0x110 for TIMER0

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
TSCFG7[3:0]				TSCFG6[3:0]				TSCFG5[3:0]				TSCFG4[3:0]			
rw				rw				rw				rw			

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

Bits	Fields	Descriptions
31:28	TSCFG7[3:0]	Trigger source configuration, include channels' input signals (Cix) and internal trigger input (ITIx) 0000: Reserved 0001: Internal trigger input 0 (ITI0) 0010: Internal trigger input 1 (ITI1) 0011: Internal trigger input 2 (ITI2) 0100: Internal trigger input 3 (ITI3) 0101: CI0 edge flag (CI0F_ED) Others: Reserved
27:24	TSCFG6[3:0]	External clock mode 0 configuration The counter counts on the rising edges of the selected trigger. 0000: External clock mode 0 disable 0001: Internal trigger input 0 (ITI0) 0010: Internal trigger input 1 (ITI1) 0011: Internal trigger input 2 (ITI2) 0100: Internal trigger input 3 (ITI3) 0101: CI0 edge flag (CI0F_ED) 0110: The filtered output of channel 0 input (CI0FE0) 0111: The filtered output of channel 1 input (CI1FE1) 1000: The filtered output of external trigger input (ETIFP) Others: Reserved

23:20	TSCFG5[3:0]	Event mode configuration A rising edge of the trigger input enables the counter. 0000: Event mode disable 0001: Internal trigger input 0 (ITI0) 0010: Internal trigger input 1 (ITI1) 0011: Internal trigger input 2 (ITI2) 0100: Internal trigger input 3 (ITI3) 0101: CI0 edge flag (CI0F_ED) 0110: The filtered output of channel 0 input (CI0FE0) 0111: The filtered output of channel 1 input (CI1FE1) 1000: The filtered output of external trigger input (ETIFP) Others: Reserved
19:16	TSCFG4[3:0]	Pause mode configuration The trigger input enables the counter clock when it is high and disables the counter clock when it is low. 0000: Pause mode disable 0001: Internal trigger input 0 (ITI0) 0010: Internal trigger input 1 (ITI1) 0011: Internal trigger input 2 (ITI2) 0100: Internal trigger input 3 (ITI3) 0101: CI0 edge flag (CI0F_ED) 0110: The filtered output of channel 0 input (CI0FE0) 0111: The filtered output of channel 1 input (CI1FE1) 1000: The filtered output of external trigger input (ETIFP) Others: Reserved
15:12	TSCFG3[3:0]	Restart mode configuration The counter is reinitialized and an update event is generated on the rising edge of the selected trigger input. 0000: Restart mode disable 0001: Internal trigger input 0 (ITI0) 0010: Internal trigger input 1 (ITI1) 0011: Internal trigger input 2 (ITI2) 0100: Internal trigger input 3 (ITI3) 0101: CI0 edge flag (CI0F_ED) 0110: The filtered output of channel 0 input (CI0FE0) 0111: The filtered output of channel 1 input (CI1FE1) 1000: The filtered output of external trigger input (ETIFP) Others: Reserved
11:8	TSCFG2[3:0]	Quadrature decoder mode 2 configuration 0000: Quadrature decoder mode 2 disable Others: The counter counts on both CI0FE0 and CI1FE1 edge, while the direction depends on each other.

7:4	TSCFG1[3:0]	Quadrature decoder mode 1 configuration 0000: Quadrature decoder mode 1 disable Others: The counter counts on CI1FE1 edge, while the direction depends on CI0FE0 level
3:0	TSCFG0[3:0]	Quadrature decoder mode 0 configuration 0000: Quadrature decoder mode 0 disable Others: The counter counts on CI0FE0 edge, while the direction depends on CI1FE1 level.

1.6.9. TIMERx configuration register (SYSCFG_TIMERxCFG, x=1, 2)

Only for GD32L235xx devices

Address offset: 0x114 for TIMER1

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
TSCFG7[3:0]				TSCFG6[3:0]				TSCFG5[3:0]				TSCFG4[3:0]			
rw				rw				rw				rw			

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

Bits	Fields	Descriptions
31:28	TSCFG7[3:0]	Trigger source configuration, include channels' input signals (Cix) and internal trigger input (ITIx) 0000: Reserved 0001: Internal trigger input 0 (ITI0) 0010: Internal trigger input 1 (ITI1) 0011: Internal trigger input 2 (ITI2) 0100: Internal trigger input 3 (ITI3) Others: Reserved
27:24	TSCFG6[3:0]	External clock mode 0 configuration The counter counts on the rising edges of the selected trigger. 0000: External clock mode 0 disable 0001: Internal trigger input 0 (ITI0) 0010: Internal trigger input 1 (ITI1) 0011: Internal trigger input 2 (ITI2) 0100: Internal trigger input 3 (ITI3) 0101: CI0 edge flag (CI0F_ED)

		0110: The filtered output of channel 0 input (CI0FE0) 0111: The filtered output of channel 1 input (CI1FE1) 1000: The filtered output of external trigger input (ETIFP) Others: Reserved
23:20	TSCFG5[3:0]	Event mode configuration A rising edge of the trigger input enables the counter. 0000: Event mode disable 0001: Internal trigger input 0 (ITI0) 0010: Internal trigger input 1 (ITI1) 0011: Internal trigger input 2 (ITI2) 0100: Internal trigger input 3 (ITI3) 0101: CI0 edge flag (CI0F_ED) 0110: The filtered output of channel 0 input (CI0FE0) 0111: The filtered output of channel 1 input (CI1FE1) 1000: The filtered output of external trigger input (ETIFP) Others: Reserved
19:16	TSCFG4[3:0]	Pause mode configuration The trigger input enables the counter clock when it is high and disables the counter clock when it is low. 0000: Pause mode disable 0001: Internal trigger input 0 (ITI0) 0010: Internal trigger input 1 (ITI1) 0011: Internal trigger input 2 (ITI2) 0100: Internal trigger input 3 (ITI3) 0101: CI0 edge flag (CI0F_ED) 0110: The filtered output of channel 0 input (CI0FE0) 0111: The filtered output of channel 1 input (CI1FE1) 1000: The filtered output of external trigger input (ETIFP) Others: Reserved
15:12	TSCFG3[3:0]	Restart mode configuration The counter is reinitialized and an update event is generated on the rising edge of the selected trigger input. 0000: Restart mode disable 0001: Internal trigger input 0 (ITI0) 0010: Internal trigger input 1 (ITI1) 0011: Internal trigger input 2 (ITI2) 0100: Internal trigger input 3 (ITI3) 0101: CI0 edge flag (CI0F_ED) 0110: The filtered output of channel 0 input (CI0FE0) 0111: The filtered output of channel 1 input (CI1FE1) 1000: The filtered output of external trigger input (ETIFP) Others: Reserved

11:8	TSCFG2[3:0]	Quadrature decoder mode 2 configuration 0000: Quadrature decoder mode 2 disable Others: The counter counts on both CI0FE0 and CI1FE1 edge, while the direction depends on each other.
7:4	TSCFG1[3:0]	Quadrature decoder mode 1 configuration 0000: Quadrature decoder mode 1 disable Others: The counter counts on CI1FE1 edge, while the direction depends on CI0FE0 level
3:0	TSCFG0[3:0]	Quadrature decoder mode 0 configuration 0000: Quadrature decoder mode 0 disable Others: The counter counts on CI0FE0 edge, while the direction depends on CI1FE1 level.

1.6.10. TIMERx configuration register (SYSCFG_TIMERxCFG, x=8, 11)

Only for GD32L235xx devices

Address offset: 0x11C for TIMER8

Address offset: 0x120 for TIMER11

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
TSCFG7[3:0]				TSCFG6[3:0]				TSCFG5[3:0]				TSCFG4[3:0]			
rw				rw				rw				rw			

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

Bits	Fields	Descriptions
31:28	TSCFG7[3:0]	Trigger source configuration, include channels' input signals (Cix) and internal trigger input (ITIx) 0000: Reserved 0001: Internal trigger input 0 (ITI0) 0010: Internal trigger input 1 (ITI1) 0011: Internal trigger input 2 (ITI2) 0100: Internal trigger input 3 (ITI3) 0101: CI0 edge flag (CI0F_ED) Others: Reserved
27:16	TSCFG6[3:0]	External clock mode 0 configuration The counter counts on the rising edges of the selected trigger. 0000: External clock mode 0 disable

		0001: Internal trigger input 0 (ITI0) 0010: Internal trigger input 1 (ITI1) 0011: Internal trigger input 2 (ITI2) 0100: Internal trigger input 3 (ITI3) 0101: CI0 edge flag (CI0F_ED) 0110: The filtered output of channel 0 input (CI0FE0) 0111: The filtered output of channel 1 input (CI1FE1) 1000: Reserved Others: Reserved
23:20	TSCFG5[3:0]	Event mode configuration A rising edge of the trigger input enables the counter. 0000: Event mode disable 0001: Internal trigger input 0 (ITI0) 0010: Internal trigger input 1 (ITI1) 0011: Internal trigger input 2 (ITI2) 0100: Internal trigger input 3 (ITI3) 0101: CI0 edge flag (CI0F_ED) 0110: The filtered output of channel 0 input (CI0FE0) 0111: The filtered output of channel 1 input (CI1FE1) 1000: Reserved Others: Reserved
19:16	TSCFG4[3:0]	Pause mode configuration The trigger input enables the counter clock when it is high and disables the counter clock when it is low. 0000: Pause mode disable 0001: Internal trigger input 0 (ITI0) 0010: Internal trigger input 1 (ITI1) 0011: Internal trigger input 2 (ITI2) 0100: Internal trigger input 3 (ITI3) 0101: CI0 edge flag (CI0F_ED) 0110: The filtered output of channel 0 input (CI0FE0) 0111: The filtered output of channel 1 input (CI1FE1) 1000: Reserved Others: Reserved
15:12	TSCFG3[3:0]	Restart mode configuration The counter is reinitialized and an update event is generated on the rising edge of the selected trigger input. 0000: Restart mode disable 0001: Internal trigger input 0 (ITI0) 0010: Internal trigger input 1 (ITI1) 0011: Internal trigger input 2 (ITI2) 0100: Internal trigger input 3 (ITI3)

		0101: CI0 edge flag (CI0F_ED) 0110: The filtered output of channel 0 input (CI0FE0) 0111: The filtered output of channel 1 input (CI1FE1) 1000: Reserved Others: Reserved
11:0	Reserved	Must be kept at reset value

1.6.11. TIMERx configuration register (SYSCFG_TIMERxCFG, x=14, 40)

Only for GD32L235xx devices

Address offset: 0x124 for TIMER14

Address offset: 0x128 for TIMER40

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
TSCFG7[3:0]				TSCFG6[3:0]				TSCFG5[3:0]				TSCFG4[3:0]			
rw				rw				rw				rw			

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

Bits	Fields	Descriptions
31:28	TSCFG7[3:0]	Trigger source configuration, include channels' input signals (Cix) and internal trigger input (ITIx) 0000: Reserved 0001: Internal trigger input 0 (ITI0) 0010: Internal trigger input 1 (ITI1) 0011: Internal trigger input 2 (ITI2) 0100: Internal trigger input 3 (ITI3) 0101: CI0 edge flag (CI0F_ED) Others: Reserved
27:24	TSCFG6[3:0]	External clock mode 0 configuration The counter counts on the rising edges of the selected trigger. 0000: External clock mode 0 disable 0001: Internal trigger input 0 (ITI0) 0010: Internal trigger input 1 (ITI1) 0011: Internal trigger input 2 (ITI2) 0100: Internal trigger input 3 (ITI3) 0101: CI0 edge flag (CI0F_ED) 0110: The filtered output of channel 0 input (CI0FE0)

		0111: The filtered output of channel 1 input (CI1FE1) Others: Reserved
23:20	TSCFG5[3:0]	Event mode configuration A rising edge of the trigger input enables the counter. 0000: Event mode disable 0001: Internal trigger input 0 (ITI0) 0010: Internal trigger input 1 (ITI1) 0011: Internal trigger input 2 (ITI2) 0100: Internal trigger input 3 (ITI3) 0101: CI0 edge flag (CI0F_ED) 0110: The filtered output of channel 0 input (CI0FE0) 0111: The filtered output of channel 1 input (CI1FE1) Others: Reserved
19:16	TSCFG4[3:0]	Pause mode configuration The trigger input enables the counter clock when it is high and disables the counter clock when it is low. 0000: Pause mode disable 0001: Internal trigger input 0 (ITI0) 0010: Internal trigger input 1 (ITI1) 0011: Internal trigger input 2 (ITI2) 0100: Internal trigger input 3 (ITI3) 0101: CI0 edge flag (CI0F_ED) 0110: The filtered output of channel 0 input (CI0FE0) 0111: The filtered output of channel 1 input (CI1FE1) Others: Reserved
15:12	TSCFG3[3:0]	Restart mode configuration The counter is reinitialized and an update event is generated on the rising edge of the selected trigger input. 0000: Restart mode disable 0001: Internal trigger input 0 (ITI0) 0010: Internal trigger input 1 (ITI1) 0011: Internal trigger input 2 (ITI2) 0100: Internal trigger input 3 (ITI3) 0101: CI0 edge flag (CI0F_ED) 0110: The filtered output of channel 0 input (CI0FE0) 0111: The filtered output of channel 1 input (CI1FE1) Others: Reserved
11:0	Reserved	Must be kept at reset value

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 F7E0

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 F7E8

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 F7EC

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 F7F0

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[95:64]	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 from the 256K bytes of the flash. It also provides page erase, mass erase, and program operations for flash memory.

2.2. Characteristics

- Up to 256KB of on-chip flash memory for instruction and data.
- 0~3 waiting time within 256KB bytes when CPU executes instructions.
- Pre-fetch buffer to speed read operations.
- For GD32L233xx, the flash page size is 4 / 2 / 1KB, depending on different flash density. For GD32L235xx, the flash page size is 1KB.
- For GD32L233xx, word programming, page erase and mass erase operation supported. For GD32L235xx, double-word programming, page erase and mass erase operation supported.
- 512B OTP(one-time program) block used for user data storage.
- 16B option bytes block for user application requirements.
- Option bytes are uploaded to the option bytes control registers when the system is reset.
- Flash security protection to prevent illegal code/data access.
- Page erase/program protection to prevent unexpected operation.
- Fast program supported (only available in GD32L233xx).
- Low-power mode supported.
- ECC check supported (only available in GD32L235xx)

2.3. Function overview

2.3.1. Flash memory architecture

For GD32L233xx, the memory includes a up to 256KB main flash memory and a 10KB information block for the bootloader and the main flash memory is organized into 64 or 32 pages with 4/2/1KB capacity per page. For GD32L235xx, the memory includes a up to 128KB main flash memory and a 12KB information block for the bootloader and the main flash memory is organized into 128 pages with 1KB capacity per page. Each page can be erased individually.

The following table shows the details of flash organization.

For GD32L233xx：

Table 2-1. 256KB flash base address and size for flash memory

Block	Name	Address range	size(bytes)
Main flash block	Page 0	0x0800 0000 – 0x0800 0FFF	4KB
	Page 1	0x0800 1000 – 0x0800 1FFF	4KB
	Page 2	0x0800 2000 – 0x0800 2FFF	4KB
	Page 63	0x0803 F000 – 0x0803 FFFF	4KB
Information block	Boot loader area	0x1FFF D000- 0x1FFF F7FF	10KB
Option bytes block	Option bytes	0x1FFF F800 – 0x1FFF F80F	16B
One-time program block	OTP bytes	0x1FFF_7000~0x1FFF_71FF	512B

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

Table 2-2. 128KB flash base address and size for flash memory

Block	Name	Address range	size(bytes)
Main flash block	Page 0	0x0800 0000 – 0x0800 07FF	2KB
	Page 1	0x0800 0800 – 0x0800 0FFF	2KB
	Page 2	0x0800 1000 – 0x0800 17FF	2KB
	Page 63	0x0801 F800 – 0x0801 FFFF	2KB
Information block	Boot loader area	0x1FFF D000- 0x1FFF F7FF	10KB
Option bytes block	Option bytes	0x1FFF F800 – 0x1FFF F80F	16B
One-time program block	OTP bytes	0x1FFF_7000~0x1FFF_71FF	512B

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

Table 2-3. 64KB flash base address and size for flash memory

Block	Name	Address range	size(bytes)
Main flash block	Page 0	0x0800 0000 – 0x0800 03FF	1KB
	Page 1	0x0800 0400 – 0x0800 07FF	1KB
	Page 2	0x0800 0800 – 0x0800 0BFF	1KB
	Page 63	0x0800 FC00 – 0x0800 FFFF	1KB
Information block	Boot loader area	0x1FFF D000- 0x1FFF F7FF	10KB
Option bytes block	Option bytes	0x1FFF F800 – 0x1FFF F80F	16B

Block	Name	Address range	size(bytes)
One-time program block	OTP bytes	0x1FFF_7000~0x1FFF_71FF	512B

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

Table 2-4. 32KB flash base address and size for flash memory

Block	Name	Address range	size(bytes)
Main flash block	Page 0	0x0800 0000 – 0x0800 03FF	1KB
	Page 1	0x0800 0400 – 0x0800 07FF	1KB
	Page 2	0x0800 0800 – 0x0800 0BFF	1KB
	Page 31	0x08007C00 – 0x0800 7FFF	1KB
Information block	Boot loader area	0x1FFF D000- 0x1FFF F7FF	10KB
Option bytes block	Option bytes	0x1FFF F800 – 0x1FFF F80F	16B
One-time program block	OTP bytes	0x1FFF_7000~0x1FFF_71FF	512B

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

For GD32L235xx：

Table 2-5. Flash base address and size for flash memory

Block	Name	Address range	size(bytes)
Main flash block	Page 0	0x0800 0000 – 0x0800 03FF	1KB
	Page 1	0x0800 0400 – 0x0800 07FF	1KB
	Page 2	0x0800 0800 – 0x0800 0BFF	1KB
	Page 127	0x0801 FC00 – 0x0801 FFFF	1KB
Information block	Boot loader area	0x1FFF D000- 0x1FFF F7FF	10KB
Option bytes block	Option bytes	0x1FFF F800 – 0x1FFF F80F	16B
One-time program block	OTP bytes	0x1FFF_7000~0x1FFF_71FF	512B

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

2.3.2. Error checking and correcting (ECC) (only available in GD32L235xx)

For GD32L233xx, ECC function is not supported. For GD32L235xx, ECC function is enabled. The ECC mechanism supports:

- One error detection and correction

- Two errors detection

When one error is detected and corrected:

- When ECC error occurs, the ECCCOR bit in FMC_ECCCS register will be set. If the ECCCORIE bit in FMC_ECCCS register is set, an interrupt is generated. The OB_ECC / OTP_ECC / SYS_ECC / MF_ECC notice the space where error occurred. The ECCADDR bits notice the offset address in each space. The ECCCOR bit in FMC_ECCCS register can be cleared by writing 1 to ECCCOR bit.

When two errors are detected:

- When occurred in option bytes (load option bytes to register after reset), the OBECCDET bit in FMC_ECCCS register will be set. The ECCADDR bits notice the offset address in option bytes. And the data return all F.
- When occurred in other space including reading option bytes at 0x1FFFF80x, the ECCDET bit in FMC_ECCCS register will be set. The OB_ECC / OTP_ECC / SYS_ECC / MF_ECC notice the space where error occurred. The ECCADDR bits notice the offset address in each space. And the data return all F. The ECCDET bit in FMC_ECCCS register can be cleared by writing 1 to ECCDET bit.

Note:

1. When ECC function is enabled, data in Flash memory are 72-bits words: 8 bits are added per double word (64 bits), but the added 8 bits are calculated by hardware and cannot be accessed by user.
2. For a virgin data 0xFF FFFF FFFF FFFF FFFF, ECC is not supported.
3. The OB_ECC / OTP_ECC / SYS_ECC / MF_ECC / ECCADDR notice the first error occurred in single bit error and double bit errors. If ECCCOR or ECCDET is set, the value will not be changed even if a new error occurred.
4. When an ECC error is reported, a new read at the error address may not generate an ECC error if the data is still present in the current buffer / prefetch buffer, even if ECCCOR and ECCDET bits are cleared.

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 the SBUS from the CPU.

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 below ***Table 2-6. The relation between WSCNT and AHB clock frequency when LDO is 1.1V for*** .

Table 2-6. The relation between WSCNT and AHB clock frequency when LDO is 1.1V

for GD32L233xx

AHB clock frequency	WSCNT configured
<= 32MHz	0 (0 wait state added)
<= 64MHz	1 (1 wait state added)

Table 2-7. The relation between WSCNT and AHB clock frequency when LDO is 0.9V for GD32L233xx

AHB clock frequency	WSCNT configured
<= 16MHz	0 (0 wait state added)
<= 36MHz	1 (1 wait state added)
<= 48MHz	2 (2 wait state added)
<= 64MHz	3 (3 wait state added)

Table 2-8. The relation between WSCNT and AHB clock frequency when LDO is 1.1V for GD32L235xx

AHB clock frequency	WSCNT configured
<= 21MHz	0 (0 wait state added)
<= 42MHz	1 (1 wait state added)
<= 64MHz	2 (2 wait state added)

If system reset occurs, the AHB clock frequency is 16MHz and the WSCNT is 0.

Note:

1. If want to increase the AHB clock frequency. First, refer to the correspondence table between WSCNT bit and AHB clock frequency, 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 configure the WSCNT.
2. If want to decrease the AHB clock frequency. First, decrease the AHB clock frequency to target frequency. Then refer to the correspondence table between WSCNT bit and AHB clock frequency, configure the WSCNT bits according the target AHB clock frequency. It is forbidden to configure the WSCNT bits before decrease the AHB clock frequency.
3.For GD32L235xx, flash only works at LDO 1.1V.

Because the wait state is added, the read efficiency is very low (such as, add 1 wait state when 64MHz and LDO is 1.1V). In order to speed up the read access, there are some functions performed.

Current buffer:

The current buffer is always enabled. Each time read from flash memory, 64-bit data get and store in current buffer. The CPU only need 32-bit or 16-bit 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 set the PFEN bit in the FMC_WS register. In the case of sequential code, when CPU execute 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 finish the current buffer and need execute the next data, the pre-fetch buffer hit.

2.3.4. Unlock the FMC_CTL register

After reset, the FMC_CTL register is not accessible in write mode, and the LK bit in the FMC_CTL register is reset to 1. An unlocking sequence consists of two write operations to the FMC_KEY register to 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 the FMC_CTL register is reset to 0 by hardware. The software can lock the FMC_CTL again by setting the LK bit in the FMC_CTL register to 1. Any wrong operations to the FMC_KEY, will set the LK bit to 1, and lock the FMC_CTL register, and lead to a bus error.

The OBPG bit and OBER bit in the FMC_CTL are still protected even the FMC_CTL is unlocked. The unlocking sequence consists of two write operations, which are writing 0x45670123 and 0xCDEF89AB to the FMC_OBKEY register. Then the hardware sets the OBWEN bit in the FMC_CTL register to 1. The software can reset OBWEN bit to 0 to protect the OBPG bit and OBER bit in the FMC_CTL register again.

2.3.5. Page erase

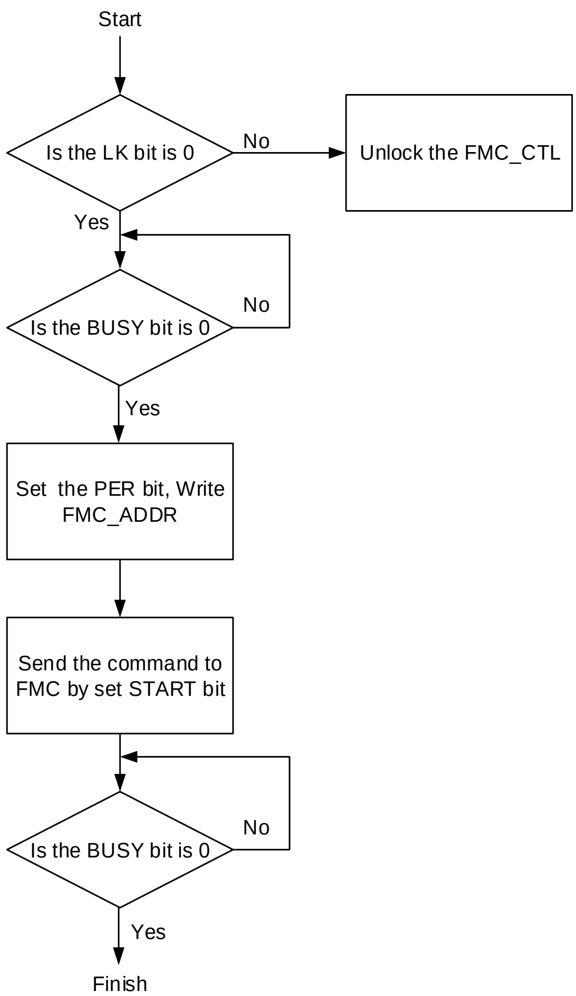
The FMC provides a page erase function which is used to initialize the contents of a main flash memory page to a high state. Each page can be erased independently without affecting the contents of other pages. The following steps show the access sequence of the registers for a page erase operation.

- 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.
- Set the PER bit in the FMC_CTL register.
- Write the page absolute address (0x08XX XXXX) into the FMC_ADDR registers.
- Send the page erase command to the FMC by setting the START bit in the FMC_CTL register.
- Wait until all the operations have finished by checking the value of the BUSY bit in the FMC_STAT register.
- Read and verify the page using a SBUS access if required.

When the operation is executed successfully, the ENDF bit in the 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 a correct target page address must be confirmed. Otherwise, the software may run out of control if the target erase page is being used to fetch codes or access data. The FMC will not provide any notification when that happens. Additionally, the page erase operation will be ignored on erase/program protected pages. 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. ***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 to initialize the main flash block contents. This erase can affect entire flash block by setting the MER bit to 1 in the FMC_CTL register. The following steps show the mass erase register access sequence.

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

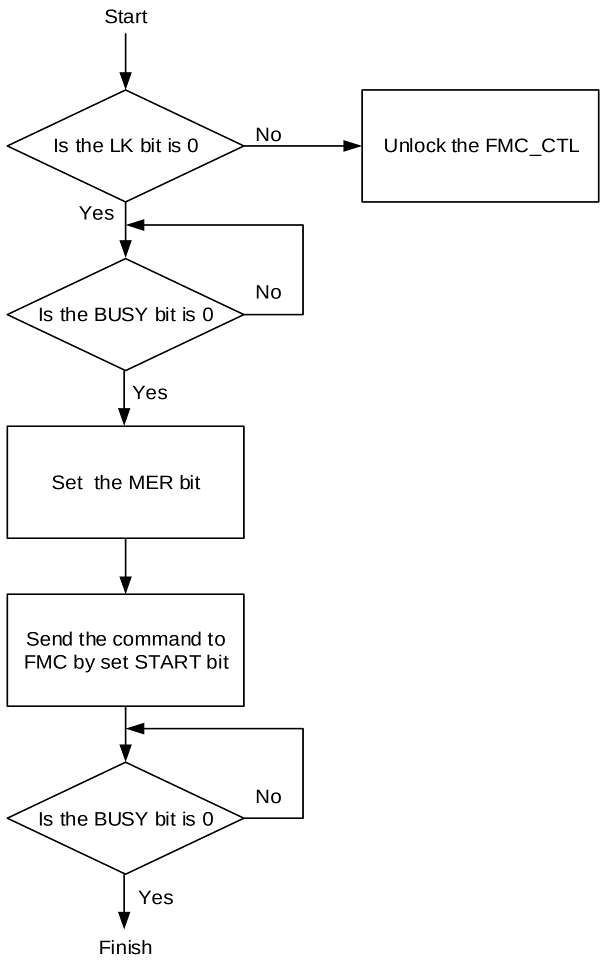
is in progress (BUSY equals to 0). Otherwise, wait until the operation has finished.

- Set the MER bit in the FMC_CTL register if erase entire flash.
- Send the mass erase command to the FMC by setting the START bit in the FMC_CTL register.
- Wait until all the operations have been finished by checking the value of the BUSY bit in the FMC_STAT register.
- Read and verify the flash memory using a SBUS access if required.

When the operation is executed successfully, the ENDF bit in the 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 modified to a value of 0xFFFF_FFFF, the mass erase operation can be implemented using a program that runs in SRAM or using the debugging tool that accesses 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.

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

For GD32L233xx, the FMC provides a 32-bit word/16-bit half word programming function by SBUS which is used to modify the main flash memory contents. While actually, the data programmed to flash memory is 32-bit.

For GD32L235xx, the FMC provides a 32-bit word/16-bit half word programming function by SBUS which is used to modify the main flash memory contents. While actually, the data programmed to flash memory is 64-bit.

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

- 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.
- Set the PG bit in the FMC_CTL register.
- Write the data to be programed by SBUS with desired absolute address (0x08XX XXXX).

For GD32L233xx:

If SBUS program is 32-bit word, the SBUS write once and the data program to flash memory. The data to be programed must be word-aligned. If SBUS program is 16-bit, the SBUS write twice form a 32-bit data and then the 32-bit data program to flash memory. The data to be programed must be word-aligned.

In order to shorten the programming time, it is recommended to use SBUS 32-bit programming.

For GD32L235xx:

If SBUS program is 32-bit, 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 be double-word aligned.

If SBUS program is 16-bit, the SBUS write four times to form a 64-bit data and then the 64-bit data program to flash memory. The data to be programed must be double-word aligned.

In order to shorten the programming time, it is recommended to use SBUS 32-bit programming.

- Wait until all the operations have been finished by checking the value of the BUSY bit in the FMC_STAT register.
- Read and verify the flash memory using a SBUS access if required.

When the operation is executed successfully, the ENDF bit in the 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 there are some program error need caution:

The programming operation checks the address if it has been erased or not. If the address has not been erased, the PGERR bit in the FMC_STAT register will be set. Each word can be programmed only one time after erase and before next erase. Note that the PG bit must be set before the word/half word programming operation.

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

In the following cases, the PGAERR bit in the FMC_STAT register will be set.

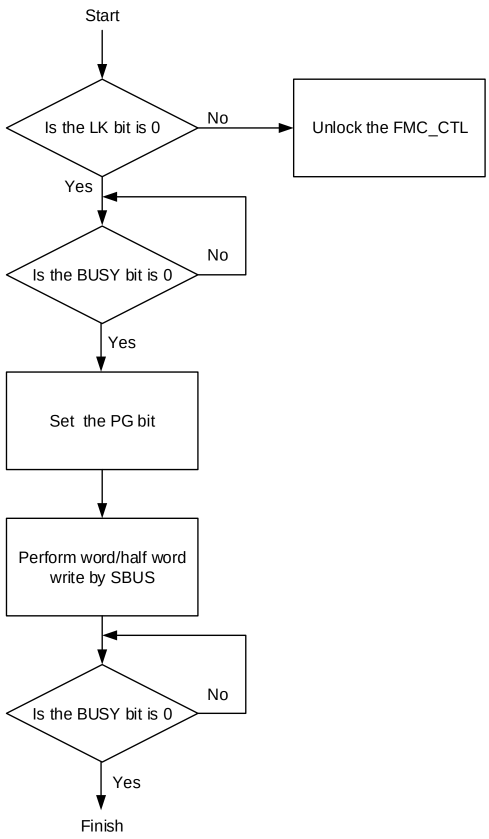
- The SBUS program use byte write (not 32-bit or 16-bit write)
- The SBUS program size is not equal previous size. It not allows mix 32-bit with 16-bit write.
- The SBUS write is not alignment.

Note: If the program operation is not write total 32 bits/64 bits, the data is not programmed to the flash memory without any notice.

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 bit, PGAERR bit or

WPERR bit in the FMC_STAT register to detect which condition occurred in the interrupt handler. *Figure 2-3. Process of program operation* shows the word programming operation flow.

Figure 2-3. Process of program operation



Note: Reading the flash should be avoided when a program/erase operation is ongoing in the same bank. And flash memory accesses will fail if the CPU enters the power saving modes.

2.3.8. Main flash fast programming (only available in GD32L233xx)

FMC provides a function that allows fast programming of one row data (32 double-word). By eliminating the verification of flash memory locations before programming, page programming time is shortened. At the same time, high voltage rise and fall times are avoided for each word.

Only the main memory can be programmed in fast programming mode.

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

- Check the row (32 double-word) in flash to confirm all data in flash is all FF.
- 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.

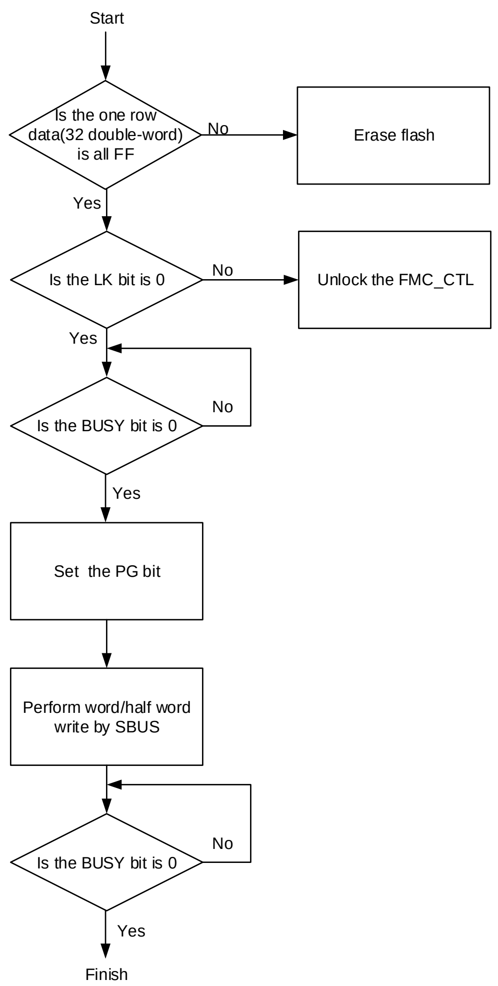
- Set the FSTPG bit in FMC_CTL register.
- Write the one row data (32 double-word) to be programed by BUS with desired absolute address (0x08XX XXXX).
- Wait until all the operations have been finished by checking the value of the BUSY bit in FMC_STAT register.
- Read and verify the Flash memory if required using a BUS access.

When the operation is executed successfully, the END 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 there are some program errors need caution:

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

In these conditions, a flash operation error interrupt will be triggered by the FMC if the ERIE bit in the FMC_CTL register is set. The software can check the PGAERR/WPERR bit in the FMC_STAT register to detect which condition occurred in the interrupt handler. ***Figure 2-4. Process of fast program operation*** displays the word programming operation flow.

Figure 2-4. Process of fast program operation



Note:

1. The 32 double-word must be aligned.
2. If try to read the flash memory while fast programming mode is ongoing, the fast programming will be aborted and the END bit in FMC_STAT register will be set.
3. When the flash interface has received the first word, programming starts automatically. When the high voltage is applied for the first word, the BUSY bit is set, and when the last word has been programmed, the BUSY bit is cleared.
4. The 32 double-word must be written successively. For all programming, the flash memory maintains a high voltage. The longest time between two words write requests is the programming time (approximately 8.8us). If a second word arrives after this time programming, fast programming is interrupted and the END bit will set and BUSY bit will be cleared. At this point, you need to clear the END bit first, and then write the next word to continue fast programming.
5. The duration of the full row of high voltage between two erase operations must not exceed 3ms. This can be guaranteed by a sequence of 32 double words written successively.

6. Because fast program mode cannot check whether the flash memory is FF through hardware, the software must first check whether the flash memory is FF, and a row must not be programmed twice or more between two erase operations. If program one row twice or more between two erase operations, unpredictable result may occur.

2.3.9. OTP programming

The OTP programming method is 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 corrupt the data stored in flash memory.

2.3.10. Option bytes erase

The FMC provides an erase function which is used to initialize the option bytes block in flash. The following steps show the erase 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 equal to 0). Otherwise, wait until the operation has finished.
- Unlock the option bytes operation bits in the FMC_CTL register if necessary.
- Wait until the OBWEN bit is set in the FMC_CTL register.
- Set the OBER bit in the FMC_CTL register.
- Send the option bytes erase command to the FMC by setting the START bit in the FMC_CTL register.
- Wait until all the operations have been finished by checking the value of the BUSY bit in the FMC_STAT register.
- Read and verify the flash memory using a SBUS access if required.

When the operation is executed successful, the ENDF bit in the FMC_STAT register is set, and an interrupt will be triggered by FMC if the ENDIE bit in the FMC_CTL register is set.

2.3.11. Option bytes modify

For GD32L233xx, the FMC provides a 32-bit word programming function which is used for modifying the option bytes block contents. For GD32L235xx, the FMC provides a 64-bit double-word programming function which is used for modifying the option bytes block contents. There are 8 pairs of option bytes. The MSB is the complement of the LSB in each pair. When the option bytes are modified, the MSB is generated by FMC automatically, not the value of input data. The following steps show the programming operation 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 option bytes operation bits in the FMC_CTL register if necessary.
- Wait until the OBWEN bit is set in the FMC_CTL register.
- Set the OBPG bit in the FMC_CTL register.
- A 32-bit word/16-bit half word written at desired address by SBUS. The program method is similar to main flash programming. (32-bit or 16-bit are SBUS programming width, and the actual data width to be programmed to option bytes is 32-bit or 64-bit. ECC check function is enabled only when the data programmed to option bytes is 64-bit.)
- Wait until all the operations have been finished by checking the value of the BUSY bit in the FMC_STAT register.
- Read and verify the flash memory if required using a SBUS access.

When the operation is executed successfully, the ENDF bit in the 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 their programming errors may occur. The PGERR bit and PGAERR bit can be set which is similar to main flash programming.

The modified option bytes only take effect after a system reset.

2.3.12. Option bytes description

The option bytes block is reloaded to the FMC_OBSTAT and FMC_WP registers after each system reset, then the option bytes take effect. The complement option bytes are the opposite of the option bytes. When reload the option bytes, if the complement option bytes and option bytes do not match, the OBERR bit in the FMC_OBSTAT register will be set, and the option bytes will be set to 0xFF. The OBERR bit will not be set if both the option bytes and its complement bytes are 0xFF.The following table shows the detail of option bytes.

Table 2-9. Option bytes

Address	Name	Description
0x1fff f800	SPC	option bytes security protection value 0xA5 : no security protection any value except 0xA5 or 0xCC : protection level low 0xCC : protection level high
0x1fff f801	SPC_N	SPC complement value
0x1fff f802	USER	[7:5]: BOR_TH (Brown out reset threshold) 000/101/110/111: BOR Level 0, brownout threshold level 0 (1.6V) 001: BOR Level 1, brownout threshold level 1 (2.0V) 010: BOR Level 2, brownout threshold level 2 (2.2V) 011: BOR Level 3, brownout threshold level 3 (2.5V) 100: BOR Level 4, brownout threshold level 4 (2.8V) [4]: reserved [3]: SRAM_PARITY_CHECK 0: disable SRAM parity check 1: enable SRAM parity check

Address	Name	Description
		Note: only available in GD32L235xx [2]: nRST_STDBY 0: generate a reset instead of entering standby mode 1: no reset when entering standby mode [1]: nRST_DPSLP 0: generate a reset instead of entering deep-sleep mode 1: no reset when entering deep-sleep mode [0]: nWDG_HW 0: hardware free watchdog 1: software free watchdog
0x1fff f803	USER_N	USER complement value
0x1fff f804	DATA[7:0]	user defined data bit 7 to 0
0x1fff f805	DATA_N[7:0]	DATA complement value bit 7 to 0
0x1fff f806	DATA[15:8]	user defined data bit 15 to 8
0x1fff f807	DATA_N[15:8]	DATA complement value bit 15 to 8
0x1fff f808	WP[7:0]	Page erase/program protection bit 7 to 0 0: protection active 1: unprotected
0x1fff f809	WP_N[7:0]	WP complement value bit 7 to 0
0x1fff f80a	WP[15:8]	Page erase/program protection bit 15 to 8
0x1fff f80b	WP_N[15:8]	WP complement value bit 15 to 8
0x1fff f80c	WP[23:16]	Page erase/program protection bit 23 to 16
0x1fff f80d	WP_N[23:16]	WP complement value bit 23 to 16
0x1fff f80e	WP[31:24]	Page erase/program protection bit 31 to 24 WP[30:24]: Each bit is related to 4KB flash protection. WP[31]: Bit 31 controls the protection of the rest flash memory.
0x1fff f80f	WP_N[31:24]	WP complement value bit 31 to 24

2.3.13. Page erase / program protection

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, the WPERR bit in the FMC_STAT register will be set by the FMC. If the WPERR bit is set and the ERRIE bit is also set to 1 to enable the corresponding interrupt, then the flash operation error interrupt will be triggered by the FMC to draw the attention of the CPU. The page protection function can be individually enabled by configuring the WP [31:0] bit field to 0 in the option bytes. If a page erase operation is executed on the option bytes block, all the flash Memory page protection functions will be disabled. When WP in the option bytes is modified, then a system reset is necessary.

2.3.14. Security protection

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

No protection: when setting SPC byte and its complement value to 0x5AA5, no protection performed. The main flash and option bytes block are accessible by all operations.

Protection level low: when setting SPC byte value to any value except 0xA5 or 0xCC, the low security protection is 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 SPC byte and its complement value to 0x5AA5, then a mass erase for main flash will be performed.

Protection level high: when setting SPC byte to 0xCC, protection level high 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 SPC byte cannot be reprogrammed. So, if protection level high is programmed, it cannot move back to protection level low or no protection level.

2.3.15. LVE sequence (only available in GD32L233xx)

If want to change CORE voltage from 1.1V to 0.9V. First set LVE to 1, and use Read_LV timing (set the corresponding wait state when LDO is 0.9V) to read. Then actually change CORE voltage 1.1V to 0.9V (reset LDOVS bit in PMU).

If want to change CORE voltage from 0.9V to 1.1V. First actually change CORE voltage 0.9V to 1.1V (set LDOVS bit in PMU). Then set LVE to 0, and use Read_HV timing (set the corresponding wait state when LDO is 1.1V) to read.

2.4. Register definition

FMC base address: 0x4002 2000

2.4.1. Wait state register (FMC_WS)

For GD32L233xx devices

Address offset: 0x00

Reset value: 0x0000 0630

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	SLEEP_SLP	RUN_SLP	Reserved					LVE	Reserved		PFEN	Reserved	WSCNT[2:0]		
	rw	rw						rw			rw		rw		

Bits	Fields	Descriptions
31:15	Reserved	Must be kept at reset value.
14	SLEEP_SLP	Flash enter sleep mode or power-down mode when MCU enter deep sleep mode or RUN_SLP bit is set. 0: Flash enter power-down mode. 1: Flash enter sleep mode.
13	RUN_SLP	Flash enter sleep/power-down mode (set by SLEEP_SLP bit) or IDLE mode during MCU run/low-power run mode. The flash memory can be put in sleep/power-down mode only when the code is executed from RAM. This bit is write-protected by FMC_SLPKEY. 0: Flash is in IDLE mode or wakeup from sleep/power-down mode (around 5us). 1: Flash enter sleep/power-down mode when no program/erase is on-going.
12:8	Reserved	Must be kept at reset value.
7	LVE	Low power enable When core is 0.9V or 1.1V 1: Core is 0.9V 0: Core is 1.1V
6:5	Reserved	Must be kept at reset value.
4	PFEN	Pre-fetch enable 0: Pre-fetch disable

		1: Pre-fetch enable
3	Reserved	Must be kept at reset value.
2:0	WSCNT[2:0]	Wait state counter register These bits are set and reset by software. 000: 0 wait state added 001: 1 wait state added 010: 2 wait state added 011: 3 wait state added 010 ~111: reserved

For GD32L235xx devices

Address offset: 0x00

Reset value: 0x0000 0010

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	SLEEP_SLP	RUN_SLP	Reserved								PFEN	Reserved		WSCNT[1:0]	
	rw	rw									rw			rw	

Bits	Fields	Descriptions
31:15	Reserved	Must be kept at reset value.
14	SLEEP_SLP	Flash enter sleep mode or IDLE mode when MCU enter deep sleep mode or RUN_SLP bit is set. 0: Flash enter IDLE mode. 1: Flash enter sleep mode.
13	RUN_SLP	Flash enter sleep mode (set by SLEEP_SLP bit) or IDLE mode during MCU run mode. The flash memory can be put in sleep mode only when the code is executed from RAM. This bit is write-protected by FMC_SLPKEY. 0: Flash is in IDLE mode or wakeup from sleep mode (around 5us). 1: If SLEEP_SLP bit is set, Flash enter sleep mode when no program/erase is on-going. **Note:** There are two ways to set Flash into sleep mode. 1.When MCU enter deep sleep mode, set SLEEP_SLP bit to put Flash in sleep mode. 2.When MCU is in running mode and no program/erase is on-going, set RUN_SLP bit and SLEEP_SLP bit to put Flash in sleep mode.

12:5	Reserved	Must be kept at reset value.
4	PFEN	Pre-fetch enable 0: Pre-fetch disable 1: Pre-fetch enable
3:2	Reserved	Must be kept at reset value.
1:0	WSCNT[1:0]	Wait state counter register These bits are set and reset by software. 00: 0 wait state added 01: 1 wait state added 10: 2 wait state added 11: reserved

2.4.2. Unlock key register (FMC_KEY)

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
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 are only be written by software. Write KEY[31:0] with keys to unlock FMC_CTL register.

2.4.3. Option bytes unlock key register (FMC_OBKEY)

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
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 are only be written by software. Write OBKEY[31:0] with keys to unlock option bytes command in the FMC_CTL register.

2.4.4. Status register (FMC_STAT)

For GD32L233xx devices

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
FSTAT	Reserved									ENDF	WPERR	PGAERR	PGERR	Reserved	BUSY
r										rc_w1	rc_w1	rc_w1	rc_w1		r

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15	FSTAT	Flash status 0: Flash is in IDLE mode. 1: Flash is in sleep mode or power-down mode.
14:6	Reserved	Must be kept at reset value.
5	ENDF	End of operation flag bit When the operation executed successfully, this bit is set by hardware. The software can clear it by writing 1.
4	WPERR	Erase/Program protection error flag bit When erase/program on protected pages, this bit is set by hardware. The software can clear it by writing 1.
3	PGAERR	Program alignment error flag bit This bit is set by hardware when SBUS write data is not alignment. The software can clear it by writing 1.
2	PGERR	Program error flag bit When program to the flash while it is not 0xFFFF, this bit is set by hardware. The software can clear it by writing 1.
1	Reserved	Must be kept at reset value.

0	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 is generated, this bit is cleared to 0.

For GD32L235xx devices

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
FSTAT[1:0]		Reserved								ENDF	WPERR	PGAERR	PGERR	Reserved	BUSY
r										rc_w1	rc_w1	rc_w1	rc_w1		r

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:14	FSTAT[1:0]	Flash status 00: Flash is in IDLE mode. 01: Flash is in power-down mode. The PMU controls whether the Flash enters power-down mode or not. Please refer to ***1.1V power domain***. 10: Flash is in sleep mode. 11: Reserved
13:6	Reserved	Must be kept at reset value.
5	ENDF	End of operation flag bit When the operation executed successfully, this bit is set by hardware. The software can clear it by writing 1.
4	WPERR	Erase/Program protection error flag bit When erase/program on protected pages, this bit is set by hardware. The software can clear it by writing 1.
3	PGAERR	Program alignment error flag bit This bit is set by hardware when SBUS write data is not alignment. The software can clear it by writing 1.
2	PGERR	Program error flag bit When program to the flash while it is not 0xFFFF, this bit is set by hardware. The software can clear it by writing 1.
1	Reserved	Must be kept at reset value.
0	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 is generated, this bit is cleared to 0.

2.4.5. Control register (FMC_CTL)

For GD32L233xx devices

Address offset: 0x10

Reset value: 0x0000 0080

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			ENDIE	Reserved	ERRIE	OBWEN	FSTPG	LK	START	OBER	OBPG	Reserved	MER	PER	PG
			rw		rw	rw	rw	rs	rs	rw	rw		rw	rw	rw

Bits	Fields	Descriptions
31:13	Reserved	Must be kept at reset value.
12	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
11	Reserved	Must be kept at reset value.
10	ERRIE	Error interrupt enable bit This bit is set or cleared by software 0: no interrupt generated by hardware. 1: error interrupt enable
9	OBWEN	Option bytes erase/program enable bit This bit is set by hardware when right sequence written to the FMC_OBKEY register. This bit can be cleared by software.
8	FSTPG	Main flash fast program command bit This bit is set or clear by software 0: no effect 1: main flash fast program command
7	LK	FMC_CTL lock bit This bit is cleared by hardware when right sequence written to the FMC_KEY register. This bit can be set by software.
6	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.
5	OBER	Option bytes erase command bit This bit is set or clear by software 0: no effect 1: option bytes erase command
4	OBPG	Option bytes program command bit This bit is set or clear by software 0: no effect 1: option bytes program command
3	Reserved	Must be kept at reset value.
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 clear by software 0: no effect 1: main flash page erase command
0	PG	Main flash program command bit This bit is set or clear by software 0: no effect 1: main flash program command

Note: This register should be reset after the corresponding flash operation completed.

For GD32L235xx devices

Address offset: 0x10

Reset value: 0x0000 0080

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			ENDIE	Reserved	ERRIE	OBWEN	Reserved	LK	START	OBER	OBPG	Reserved	MER	PER	PG
			rw		rw	rw		rs	rs	rw	rw		rw	rw	rw

Bits	Fields	Descriptions
31:13	Reserved	Must be kept at reset value.
12	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
11	Reserved	Must be kept at reset value.
10	ERRIE	Error interrupt enable bit This bit is set or cleared by software 0: no interrupt generated by hardware. 1: error interrupt enable
9	OBWEN	Option bytes erase/program enable bit This bit is set by hardware when right sequence written to the FMC_OBKEY register. This bit can be cleared by software.
8	Reserved	Must be kept at reset value.
7	LK	FMC_CTL lock bit This bit is cleared by hardware when right sequence written to the FMC_KEY register. This bit can be set by software.
6	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.
5	OBER	Option bytes erase command bit This bit is set or clear by software 0: no effect 1: option bytes erase command
4	OBPG	Option bytes program command bit This bit is set or clear by software 0: no effect 1: option bytes program command
3	Reserved	Must be kept at reset value.
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 clear by software 0: no effect 1: main flash page erase command
0	PG	Main flash program command bit This bit is set or clear by software 0: no effect

1: main flash program command

Note: This register should be reset after the corresponding flash operation completed.

2.4.6. Address register (FMC_ADDR)

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
ADDR[31:16]															
w															

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

Bits	Fields	Descriptions
31:0	ADDR[31:0]	Flash erase/program command address bits These bits are configured by software. ADDR bits are the address of flash to be erased/programmed.

2.4.7. ECC control and status register (FMC_ECCCS)

Only for GD32L235xx devices

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
ECCADDR[15:0]															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ECCCORIE	ECCDETIE	Reserved							OBECCDET	OB_ECC	OTP_ECC	MF_ECC	SYS_ECC	ECCDET	ECCCOR
rw	rw								rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1

Bits	Field	Descriptions
31:16	ECCADDR[15:0]	The offset address of double word where an ECC error is detected. Error address = base address + ECCADDR[15:0] * 8, the base address can be the start address of main Flash, system area, option bytes and OTP. Main flash: 0 ~ 0x3FFF (128K/8 – 1) Option bytes: 0 ~ 0x01 (16/8 -1) OTP: 0 ~ 0x3F (512/8 -1)

		System memory: 0 ~ 0x4FF (10K/8 – 1)
15	ECCCORIE	One bit error correct interrupt enable. 0: One bit error correct interrupt disable. 1: One bit error correct interrupt enable.
14	ECCDETIE	Two bits errors detect interrupt enable. 0: Two bits errors detect interrupt disable. 1: Two bits errors detect interrupt enable.
13:7	Reserved	Must be kept at reset value.
6	OBECCDET	Option bytes two bits errors detect flag. This bit is set when an ECC error occurs when loading option bytes to register after reset. This bit is cleared by writing 1. 0: Two ECC errors of option bytes are not detected. 1: Two ECC errors of option bytes are detected.
5	OB_ECC	If an ECC bit error is detected in reading option bytes, this bit will be set. And the ECCADDR records the offset address of option bytes. ECCDET bit and ECCCOR bit indicate there is a one bit error or a two bits error. This bit is cleared by writing 1. This bit is cleared by writing 1. 0: No ECC error is detected in reading option bytes. 1: An ECC bit error is detected in reading option bytes.
4	OTP_ECC	If an ECC bit error is detected in OTP, this bit will be set. And the ECCADDR records the offset address of OTP. ECCDET bit and ECCCOR bit indicate there is a one bit error or a two bits error. This bit is cleared by writing 1. 0: No ECC error is detected in OTP. 1: An ECC bit error is detected in OTP.
3	MF_ECC	If an ECC bit error is detected in main Flash, this bit will be set. And the ECCADDR records the offset address of main Flash. ECCDET bit and ECCCOR bit indicate there is a one bit error or a two bits error. This bit is cleared by writing 1. 0: No ECC error is detected in main Flash. 1: An ECC bit error is detected in main Flash.
2	SYS_ECC	If an ECC bit error is detected in system memory, this bit will be set. And the ECCADDR records the offset address of system memory. ECCDET bit and ECCCOR bit indicate there is a one bit error or a two bits error. This bit is cleared by writing 1. 0: No ECC error is detected in system memory. 1: An ECC bit error is detected in system memory.
1	ECCDET	Two bit errors detect flag. This bit set when two bit errors is detected. This bit is cleared by writing 1. 0: Two ECC errors are not detected. 1: Two ECC errors are detected.

0	ECCCOR	One bit error detected and correct flag. This bit is cleared by writing 1. 0: No ECC error is detected and corrected. 1: An ECC error is detected and corrected.

2.4.8. Option bytes status register (FMC_OBSTAT)

Address offset: 0x1C

Reset value: 0x0XXX 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						DATA[15:6]									
						r									

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

Bits	Fields	Descriptions
31:26	Reserved	Must be kept at reset value.
25:10	DATA[15:0]	Store DATA[15:0] of option bytes block after system reset.
9:2	USER[7:0]	Store USER of option bytes block after system reset.
1	SPC	Option bytes security protection code 0: no protection 1: protection
0	OBERR	Option bytes read error bit. This bit is set by hardware when the option bytes and its complement byte do not match, then the option bytes is set to 0xFF.

2.4.9. Erase/Program protection register (FMC_WP)

Address offset: 0x20

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
WP[31:16]															
r															

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

Bits	Fields	Descriptions
31:0	WP[31:0]	Store WP[31:0] of option bytes block after system reset

2.4.10. Unlock flash sleep/power-down mode key register (FMC_SLPKEY)

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
SLPKEY[31:16]															
w															

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

Bits	Fields	Descriptions
31:0	SLPKEY[31:0]	RUN_SLP unlock register These bits are only be written by software. Write SLPKEY[31:0] with keys to unlock RUN_SLP bit in FMC_WS register. SLPKEY1: 0x04152637 SLPKEY2: 0xBCAD9E8F

2.4.11. Product ID register (FMC_PID)

Address offset: 0x100

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 GD32L23x series. For GD32L233xx devices, power management unit (PMU) provides ten types of power saving modes, including Run, Run1, Run2, Sleep, Sleep1, Sleep2, Deep-sleep, Deep-sleep 1, Deep-sleep 2 and Standby mode. For GD32L235xx devices, power management unit (PMU) provides six types of power saving modes, including Run, Sleep, Deep-sleep, Deep-sleep 1, Deep-sleep 2 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 GD32L23x devices, there are three power domains, including V_{DD} / V_{DDA} domain, 1.1V domain, and Backup domain, as are shown in ***Figure 3-1. Power supply overview of GD32L233xx devices*** and ***Figure 3-2. Power supply overview of GD32L235xx devices***. The power of the V_{DD} domain is supplied directly by V_{DD}. An embedded LDO in the V_{DD} / V_{DDA} domain is used to supply the 1.1V domain power. A power switch is implemented for the Backup domain. It can be powered from the V_{BAT} voltage when the main V_{DD} supply is shut down.

3.2. Characteristics

- Three power domains: V_{BAK}, V_{DD} / V_{DDA} and 1.1V power domains.
- Low Voltage Detector can issue an interrupt or event when the power is lower than a programmed threshold.
- Battery power (V_{BAT}) for Backup domain when V_{DD} is shut down.
- LDO output voltage select for power saving.
- Ultra power saving for low-driver mode in Deep-sleep / Deep-sleep 1 / Deep-sleep 2 mode.
- SRAM1 can be power-off alone.
- CPU can retent its registers' values when COREOFF0 is power off so that MCU can continue to execute instructions when it wakes up without reloading the program.

For GD32L233xx devices:

- Internal Voltage regulator (LDO) supplies around 1.1V or 0.9V voltage source for 1.1V domain.
- Ten power saving modes: Run, Run1, Run2, Sleep, Sleep1, Sleep2, Deep-sleep, Deep-sleep 1, Deep-sleep 2 and Standby modes.
- CAU can be power off alone.

For GD32L235xx devices:

- Internal Voltage regulator (LDO) supplies around 1.1V source for 1.1V domain.
- Six power saving modes: Run, Sleep, Deep-sleep, Deep-sleep 1, Deep-sleep 2 and

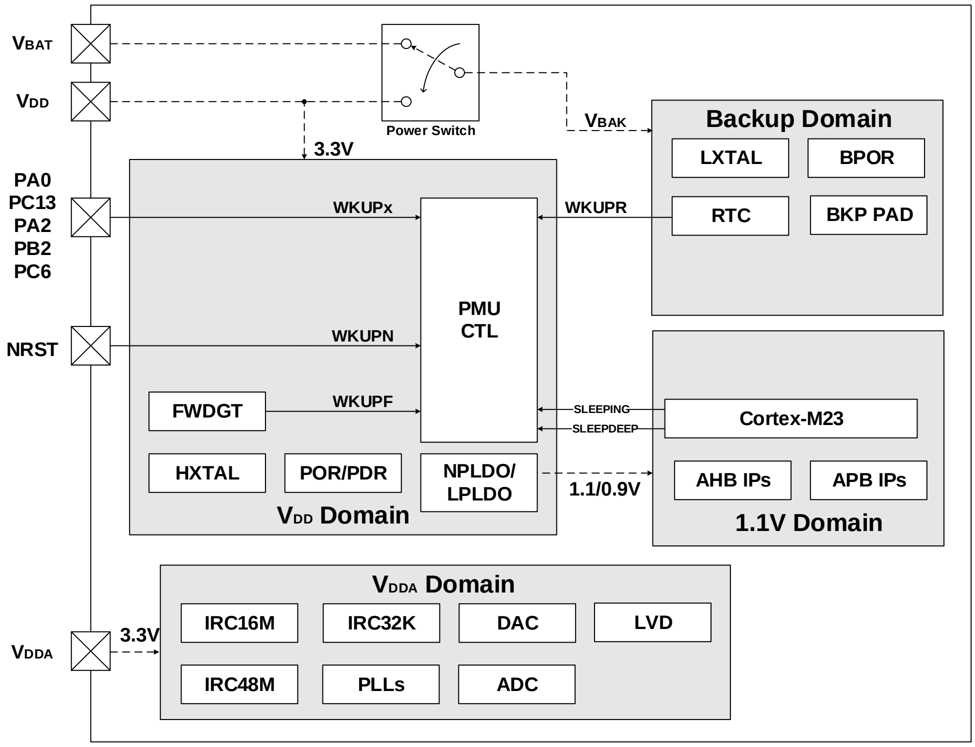
Standby modes.

- Low power Internal Voltage regulator (LDO) supplies around 0.9V voltage source for 1.1V domain when in Deep-sleep 1 or Deep-sleep 2 mode.
- EFLASH can be power-off alone when in run or Deep-sleep mode.

3.3. Function overview

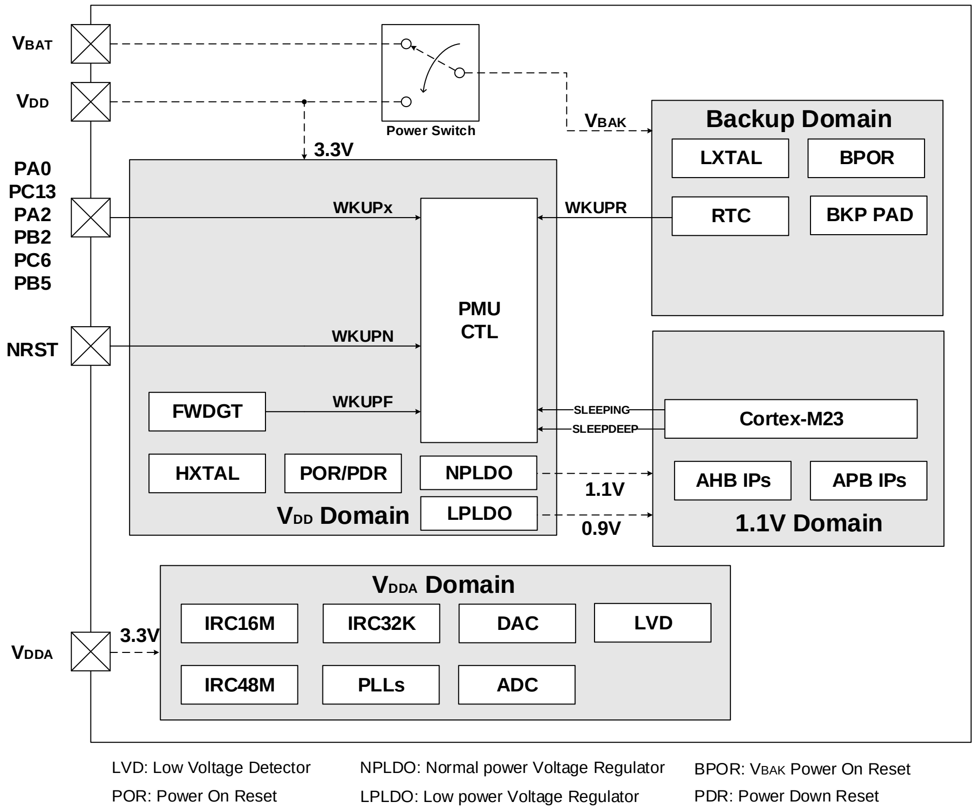
Figure 3-1. Power supply overview of GD32L233xx devices and *Figure 3-2. Power supply overview of GD32L235xx devices* provide details on the internal configuration of the PMU and the relevant power domains.

Figure 3-1. Power supply overview of GD32L233xx devices



LVD: Low Voltage Detector
POR: Power On Reset
NPLDO: Normal power Voltage Regulator
LPLDO: Low power Voltage Regulator
BPOR: VBAK Power On Reset
PDR: Power Down Reset

Figure 3-2. Power supply overview of GD32L235xx devices



3.3.1. Battery backup domain

The Backup domain is powered by the V_{DD} or the battery power source (V_{BAT}) selected by the internal power switch, and the V_{BAK} pin which drives Backup domain, supplies power for RTC unit, LXTAL oscillator and BPOR, and three pads, including PC13 to PC15. In order to ensure the content of the Backup domain registers and the RTC supply, when V_{DD} supply is shut down, V_{BAT} pin can be connected to an optional standby voltage supplied by a battery or by another source. The power switch is controlled by the Power Down Reset circuit in the V_{DD} / V_{DDA} domain. If no external battery is used in the application, it is recommended to connect V_{BAT} pin externally to V_{DD} pin with a 100nF external ceramic decoupling capacitor.

The Backup domain reset sources include the Backup domain power-on-reset (BPOR) and the Backup domain software reset. The BPOR signal forces the device to stay in the reset mode until V_{BAK} is completely powered up. Also the application software can trigger the Backup domain software reset by setting the BKPRST bit in the RCU_BDCTL register to reset the Backup domain.

The clock source of the Real Time Clock (RTC) circuit can be derived from the Internal 32KHz RC oscillator (IRC32K) or the Low Speed Crystal oscillator (LXTAL), or HXTAL clock divided by 32. When V_{DD} is shut down, only LXTAL is valid for RTC. Before entering the power saving mode by executing the WFI / WFE instruction, the Cortex®-M23 can setup the RTC register with an expected alarm time and enable the alarm function to achieve the RTC alarm event. After entering the power saving mode for a certain amount of time, the RTC alarm will wake up the device when the time match event occurs. The details of the RTC configuration and operation will be described in the ***Real time clock (RTC)***.

When the Backup domain is supplied by V_{DD} (V_{BAK} pin is connected to V_{DD}), the following functions are available:

- PC13 can be used as GPIO or RTC function pin described in the ***Real time clock (RTC)***.
- PC14 and PC15 can be used as either GPIO or LXTAL Crystal oscillator pins.

When the Backup domain is supplied by V_{BAT} (V_{BAK} pin is connected to V_{BAT}), the following functions are available:

- PC13 can be used as RTC function pin described in the ***Real time clock (RTC)***.
- PC14 and PC15 can be used as LXTAL Crystal oscillator pins only.

Note: Since PC13, PC14, PC15 are supplied through the Power Switch, which can only be obtained by a small current, the speed of GPIOs PC13 to PC15 should not exceed 2MHz when they are in output mode(maximum load: 30pF).

The external V_{BAT} battery can be charged by the V_{DD} through an internal resistor. The charging resistor can be selected by configuring the VCRSEL bit in PMU_CTL0 register. A 5 kOhms resistor or a 1.5 kOhms resistor can be selected for external V_{BAT} battery charing. The external V_{BAT} battery charing is enabled by setting the VCEN bit in PMU_CTL0 register. When in BKP_ONLY mode, the V_{BAT} battery charing is disabled by hardware.

Note: In BKP_ONLY mode, V_{DD} is power-off, and the backup domain is power by V_{BAT} pin.

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

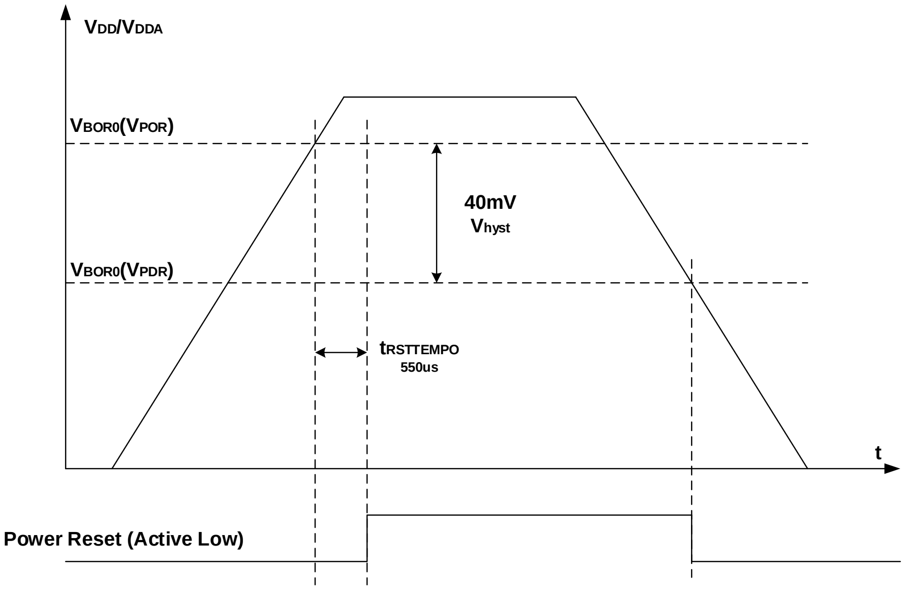
V_{DD} / V_{DDA} domain includes two parts: V_{DD} domain and V_{DDA} domain. V_{DD} domain includes HXTAL (high speed crystal oscillator), LDO (voltage regulator), POR / PDR (power on / down Reset), FWDGT (free watchdog timer), all pads except PC13 / PC14 / PC15, etc. V_{DDA} domain includes ADC / DAC (AD / DA converter), IRC16M (internal 16MHz RC oscillator), IRC48M (internal 48MHz RC oscillator at 48MHz frequency), IRC32K (internal 32KHz RC oscillator), PLLs (phase locking loop), LVD (low voltage detector), etc.

V_{DD} domain

The LDO, which is implemented to supply power for the 1.1V domain, is always enabled after reset. It can be configured to operate in different status, including in the Sleep mode (1.1V full power on, 0.9V full power on, low power), in the Deep-sleep / Deep-sleep 1 / Deep-sleep 2 mode (on or low power), and in the Standby mode (power off).

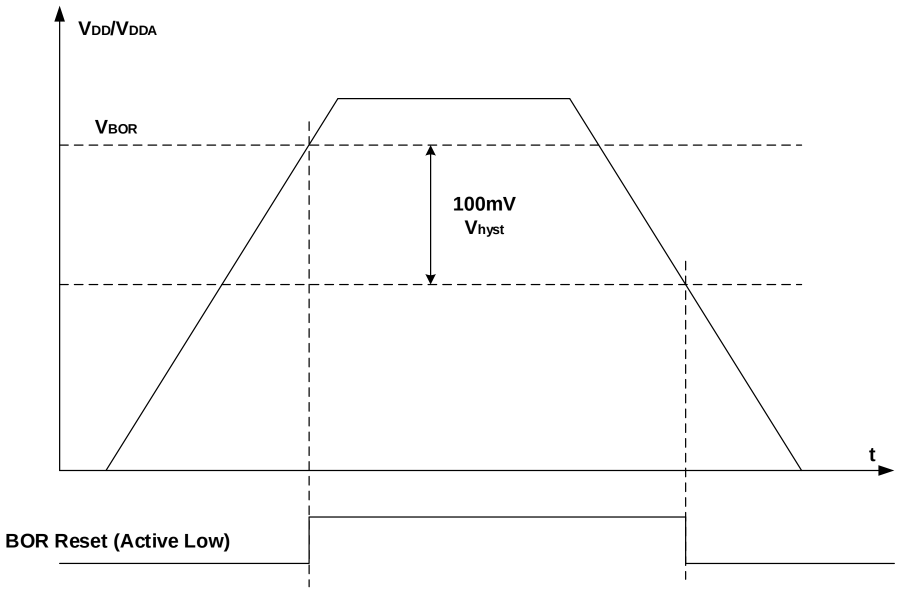
The POR / PDR circuit is implemented to detect V_{DD} / V_{DDA} and generate the power reset signal which resets the whole chip except the Backup domain when the supply voltage is lower than the specified threshold. ***Figure 3-3. Waveform of the BOR0*** shows the relationship between the supply voltage and the power reset signal. V_{BOR0}, indicates the threshold of BOR0 on reset. The hysteresis voltage (V_{hyst}) is around 40mV.

Figure 3-3. Waveform of the BOR0



The BOR circuit is used to detect V_{DD} / V_{DDA} and generate the power reset signal which resets the whole chip except the Backup domain when the supply voltage is lower than the specified threshold which defined in the BOR_TH bits in option bytes. Notice that the BOR0 circuit is always implemented. ***Figure 3-4. Waveform of the BOR*** shows the relationship between the supply voltage and the BOR reset signal. V_{BOR}, which defined in the BOR_TH bits in option bytes, indicates the threshold of BOR on reset. The hysteresis voltage (V_{hyst}) is 100mV.

Figure 3-4. Waveform of the BOR

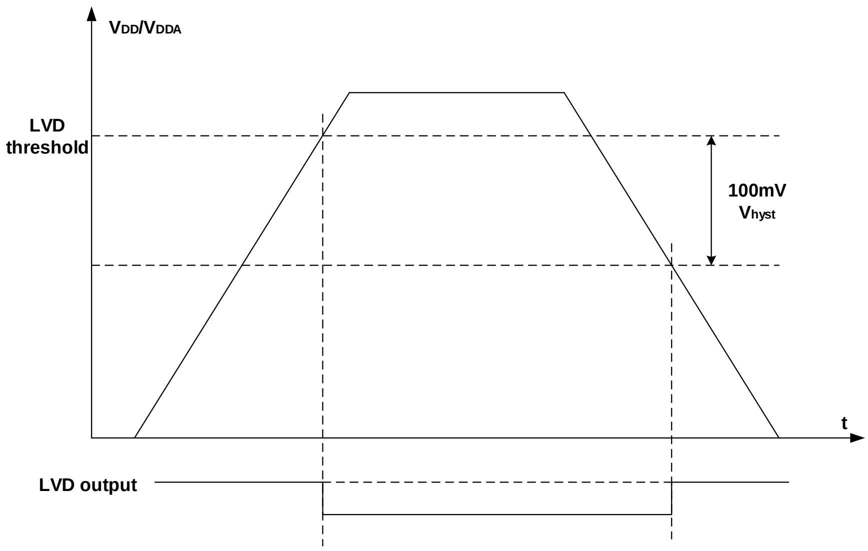


V_{DDA} domain

The LVD is used to detect whether the V_{DD} / V_{DDA} supply voltage is lower than a programmed threshold selected by the LVDT[2:0] bits in the power control register0 (PMU_CTL0). The LVD is enabled by setting the LVDEN bit, and LVDF bit, which in power control and status register (PMU_CS), indicates if V_{DD} / V_{DDA} is higher or lower than the LVD threshold. This event is internally connected to the EXTI line 16 and can generate an interrupt if it is enabled through the EXTI registers. *Figure 3-5. Waveform of the LVD threshold* shows the relationship between the LVD threshold and the LVD output (LVD interrupt signal depends on EXTI line 16 rising or falling edge configuration). The following figure shows the relationship between the supply voltage and the LVD signal. The hysteresis voltage (V_{hyst}) is 100mV.

Note: When LVDT[2:0] is configured as “111”, the input voltage on PB7 is compared with 0.8V, LVDF indicates if the input voltage is higher or lower than 0.8V.

Figure 3-5. Waveform of the LVD threshold



Generally, digital circuits are powered by V_{DD}, while most of analog circuits are powered by V_{DDA}. To improve the ADC and DAC conversion accuracy, the independent power supply V_{DDA} is implemented to achieve better performance of analog circuits. V_{DDA} can be externally connected to V_{DD} through the external filtering circuit that avoids noise on V_{DDA}, and V_{SSA} should be connected to V_{SS} through the specific circuit independently. Otherwise, when the V_{DD} and V_{DDA} are provided by different power supplies, the difference between V_{DD} and V_{DDA} during power-up and running time should not exceed 0.3V.

3.3.3. 1.1V power domain

The main functions that include Cortex®-M23 logic, AHB / APB peripherals, the APB interfaces for the Backup domain and the V_{DD} / V_{DDA} domain, etc, are located in this power domain. Once the 1.1V is powered up, the POR will generate a reset sequence on the 1.1V 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.

SRAM1 power domain

SRAM1(For GD32L233xx devices, 0x20004000~0x20007FFF. For GD32L235xx devices, 0x20002000~0x20005FFF) can be power-off alone and it is power on after system reset by default. SRAM1 can be powered off in order to reduce the power consumption in Run / Run1 / Run2 mode for GD32L233xx devices and reduce the power consumption in Run mode for GD32L235xx devices. To further reduce power consumption in low-power mode (Sleep / Sleep1 / Sleep2 / Deep-sleep / Deep-sleep1 / Deep-sleep2 for GD32L233xx devices and Sleep / Deep-sleep / Deep-sleep1 / Deep-sleep2 for GD32L235xx devices), SRAM1 can be powered off before entering the low-power mode.

COREOFF1 power domain for GD32L233xx devices

The COREOFF1 can be power-off alone. COREOFF1 is power off after system reset by default. The COREOFF1 domain needs to be powered on when using modules in the COREOFF1 domain. The COREOFF1 domain can be powered off in order to reduce the power consumption in Run / Run1 / Run2 mode. To further reduce power consumption in low-power mode (Sleep / Sleep1 / Sleep2 / Deep-sleep / Deep-sleep1 / Deep-sleep2), the COREOFF1 domain can be powered off before entering the low-power mode.

The COREOFF1 power domain includes CAU module.

EFLASH power domain for GD32L235xx devices

The EFLASH can be power-off alone. The default power status is power on after system reset. The core voltage can be switched from 1.1v to 0.9V or even lower voltage, and the EFLASH needs to be powered off by setting the EFPSLEEP bit in PMU_CTL1 register as the EFLASH cannot work properly. When the mcu enters deep-sleep mode, EFLASH can be controlled to switch off by setting the EFDSPSLEEP bit in PMU_CTL1 register.

The EFLASH power domain can work only when the voltage of LDO is 1.1V. In this case, the code should be stored in SRAM.

COREOFF0 power domain

The COREOFF0 power domain is power-off when enter Deep-sleep2 mode and power-on when exit Deep-sleep2 mode.

The COREOFF0 power domain includes the following module:

CPU / BUS / ADC / CMP / CRC / CTC / DAC / DMA / I2C0 / I2C1 / SLCD / TRNG / SPI0 / SPI1 / TIMER1 / TIMER2 / TIMER5 / TIMER6 / TIMER8 / TIMER11 / USART0 / USART1 / UART3 / UART4 / USBD. For GD32L235xx devices, the COREOFF0 power domain also includes TIMER0 / TIMER14 / TIMER40 / CAU.

Note: For GD32L233xx devices, the CPU registers can be retention or not by configuring NRRD2 bit in PMU_CTL1 register when enter / exit Deep-sleep2 mode.

3.3.4. Power saving modes

After a system reset or a power reset, the GD32L23x MCU operates at full function and all power domains are active. Users can achieve lower power consumption through slowing down the system clocks (HCLK, PCLK1, and PCLK2) or gating the clocks of the unused peripherals. Besides, for GD32L233xx devices, ten power saving modes are provided to achieve even lower power consumption, they are Run, Run1, Run2, Sleep mode, Sleep 1 mode, Sleep2 mode, Deep-sleep mode, Deep-sleep 1 mode, Deep-sleep 2 mode and Standby mode. For GD32L235xx devices, six power saving modes are provided to achieve even lower power consumption, they are Run, Sleep mode, Deep-sleep mode, Deep-sleep 1 mode, Deep-sleep 2 mode and Standby mode.

Run mode

After system reset / power reset or wakeup from standby mode, the MCU enters Run mode. And the NPLDO (normal power LDO) works in 1.1V mode.

Run1 mode

When in Run mode, the NPLDO should be selected as 0.9V by configuring the LDOVS bits in PMU_CTL0. In this mode, the system clock frequency should not exceed 16MHz.

Note: Run1 mode is only available in GD32L233xx devices.

Run2 mode

When in Run mode or Run1 mode, the NPLDO can be selected as 0.9V by configuring the LDOVS bits in PMU_CTL0. The LDNP in PMU_CTL0 register should be configured to select the low-dirver mode. In this mode, the system clock frequency should not exceed 16MHz and the clock frequency of AHB should not exceed 2MHz.

Note: Run2 mode is only available in GD32L233xx devices.

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. The NPLDO should be selected as 0.9V by configuring the LDOVS bits in PMU_CTL0.

Note: Sleep1 mode is only available in GD32L233xx devices.

Sleep2 mode

The Sleep2 mode is corresponding to the SLEEPING mode of the Cortex®-M23 When in Run2 mode. The NPLDO should be selected as 0.9V by configuring the LDOVS bits in PMU_CTL0. The LDNP in PMU_CTL0 should be configured to select the low-dirver mode.

Note: Sleep2 mode is only available in GD32L233xx devices.

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 1.1V domain are off, and all of IRC16M, IRC48M, HXTAL and PLLs are disabled. The contents of SRAM and registers are preserved. The NPLDO can operate normally or in low driver mode depending on the LDNPDSP bit in the PMU_CTL0 register. 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 IRC16M is selected as the system clock. Notice that an additional wakeup delay will be incurred if the LDO operates in low driver mode.

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 6-4. EXTI source***. If not, the program will skip the entry process of Deep-sleep mode to continue to execute the following procedure.

Deep-sleep 1 mode

The Deep-sleep 1 mode is based on the SLEEPDEEP mode of the Cortex®-M23. In Deep-sleep 1 mode, all clocks in the 1.1V domain are off, and all of IRC16M, IRC48M, HXTAL and PLLs are disabled. The LPLDO (low power LDO) can operate normally instead of NPLDO. Before entering the Deep-sleep 1 mode, it is necessary to set the SLEEPDEEP bit in the Cortex®-M23 System Control Register, set LPMOD bits to “01” in the PMU_CTL0 register.

Then, the device enters the Deep-sleep 1 mode after a WFI or WFE instruction is executed. If the Deep-sleep 1 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 1 mode, the IRC16M is selected as the system clock. Waking up from Deep-sleep 1 mode needs an additional delay to wakeup NPLDO.

Note: If power-on or exit from standby, it needs to wait more than 600us before entering Deep-sleep 1 mode.

Deep-sleep 2 mode

The Deep-sleep 2 mode is based on the SLEEPDEEP mode of the Cortex®-M23. In Deep-sleep 2 mode, all clocks in the 1.1V domain are off, and all of IRC16M, IRC48M, HXTAL and PLLs are disabled. For GD32L233xx devices, the power of COREOFF0 / SRAM1 / COREOFF1 domain is cut off and the contents of COREOFF0 / SRAM1 / COREOFF1 domain are lost. For GD32L235xx devices, the power of COREOFF0 / SRAM1 domain is cut off and the contents of COREOFF0 / SRAM1 domain are lost. The LPLDO can operate normally instead of NPLDO. Before entering the Deep-sleep 2 mode, it is necessary to set the SLEEPDEEP bit in the Cortex®-M23 System Control Register, set LPMOD bits to “10” in the PMU_CTL0 register. Then, the device enters the Deep-sleep 2 mode after a WFI or WFE instruction is executed. If the Deep-sleep 2 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 2 mode, the IRC16M is selected as the system clock. Waking up from Deep-sleep 2 mode needs an additional delay to wakeup NPLDO.

Note: If power-on or exit from standby, it is need to wait more than 600us before entering Deep-sleep 2 mode.

Standby mode

The Standby mode is based on the SLEEPDEEP mode of the Cortex®-M23, too. In Standby mode, the whole 1.1V domain is power off, the NPLDO / LPLDO is shut down, and all of IRC16M, IRC48M, HXTAL and PLLs 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 / time stamp / tamper / auto wakeup events, the FWDGT reset, and the rising edge on WKUP pins. The Standby mode achieves the lowest power consumption, but spends longest time to wake up. Besides, the contents of SRAM and registers in 1.1V 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 (for GD32L233xx devices)

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 <= 16Mhz	NPLDO on LPLDO on	LDOVS set to 0.9V	Clear LDVOS	-	-
Run2	System clock <= 2Mhz	NPLDO in low driver mode LPLDO on	LDOVS set to 0.9V and LDNP set to 1	Clear LDVOS and LDNP	-	-
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 mode	-
Sleep1	Only CPU clock is off	NPLDO on LPLDO on	SLEEPDEEP = 0, WFI or WFE from Run1	Any interrupt for WFI Any event (or interrupt when SEVONPEND is 1) for WFE	Run1 mode	-
Sleep2	Only CPU clock is off	NPLDO in low driver mode LPLDO on	SLEEPDEEP = 0, WFI or WFE from Run2	Any interrupt for WFI Any event (or interrupt when SEVONPEND is 1) for WFE	Run2 mode	-
Deep-sleep	1. All clocks in the 1.1V domain are off 2. Disable IRC16M, IRC48M, HXTAL and PLLs	NPLDO in low driver mode or normal driver mode 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 / Run1 / Run2	IRC16M wakeup time, + Flash wakeup time
Deep-sleep 1	1. All clocks in the 1.1V domain are off 2. Disable IRC16M, IRC48M, HXTAL and PLLs 3. LPLDO instead of NPLDO	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 / Run1 / Run2	IRC16M wakeup time, + NPLDO wakeup time+Flash wakeup time
Deep-sleep 2	1. All clocks in the 1.1V domain are off 2. Disable IRC16M,	NPLDO off LPLDO on	SLEEPDEEP = 1, LPMOD = 10, WFI or WFE	Any interrupt from EXTI lines for WFI Any event(or interrupt	Run / Run1 / Run2	IRC16M wakeup time, + NPLDO

Mode	Description	LDO	Entry	Wakeup	Wakeup status	Wakeup Latency
	IRC48M, HXTAL and PLLs 3. LPLDO instead of NPLDO 4. COREOFF0 / SRAM1 power-off. 5. For GD32L233xx devices, COREOFF1 power-off.			when SEVONPEND is 1) from EXTI for WFE		wakeup time+Flash wakeup time
Standby	1. The 1.1V domain is power off 2. Disable IRC16M, IRC48M, HXTAL and PLLs	NPLDO off LPLDO off	SLEEPDEEP = 1, LPMOD = 11, WFI or WFE	1. NRST pin 2. WKUP pins 3. FWDGT reset 4. RTC	Run	IRC16M wakeup time, + NPLDO wakeup time+Flash wakeup time
BKP_ONLY	All V_{DD} / 1.1V core domain power off	NPLDO off LPLDO off	V_{DD} off	V_{DD} on	Run	V_{DD} power on sequence

Table 3-2. Power saving mode summary(for GD32L235xx devices)

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	-	-	-
Sleep	Only CPU clock is off	NPLDO on	SLEEPDEEP = 0, WFI or WFE from Run	Any interrupt for WFI Any event (or interrupt when SEVONPEND is 1) for WFE	Run mode	-
Deep-sleep	1. All clocks in the 1.1V domain are off 2. Disable IRC16M, IRC48M, HXTAL and PLLs	NPLDO in low driver mode or normal driver mode 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	IRC16M wakeup time, + Flash wakeup time
Deep-sleep 1	1. All clocks in the 1.1V domain are off 2. Disable IRC16M, IRC48M, HXTAL and PLLs 3. LPLDO instead of NPLDO	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	IRC16M wakeup time, + NPLDO wakeup time+Flash wakeup time

Mode	Description	LDO	Entry	Wakeup	Wakeup status	Wakeup Latency
Deep-sleep 2	1. All clocks in the 1.1V domain are off 2. Disable IRC16M, IRC48M, HXTAL and PLLs 3. LPLDO instead of NPLDO 4. COREOFF0 / SRAM1 power-off. 5. For GD32L233xx devices, COREOFF1 power-off.	NPLDO off LPLDO on	SLEEPDEEP = 1, LPMOD = 10, WFI or WFE	Any interrupt from EXTI lines for WFI Any event(or interrupt when SEVONPEND is 1) from EXTI for WFE	Run	IRC16M wakeup time, + NPLDO wakeup time+Flash wakeup time
Standby	1. The 1.1V domain is power off 2. Disable IRC16M, IRC48M, HXTAL and PLLs	NPLDO off LPLDO off	SLEEPDEEP = 1, LPMOD = 11, WFI or WFE	1. NRST pin 2. WKUP pins 3. FWDGT reset 4. RTC	Run	IRC16M wakeup time, + NPLDO wakeup time+Flash wakeup time
BKP_ONLY	All V_{DD} / 1.1V core domain power off	NPLDO off LPLDO off	V_{DD} off	V_{DD} on	Run	V_{DD} power on sequence

Note:

(2) 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.

3.4. Register definition

PMU base address: 0x4000 7000

3.4.1. Control register 0 (PMU_CTL0)

For GD32L233xx devices

Address offset: 0x00

Reset value: 0x0000 C000 (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
LDOVS[1:0]		VCRSEL	VCEN	LDNP	LDNPDSP	Reserved	BKPWEN	LVDT[2:0]			LVDEN	STBRST	WURST	LPMOD[1:0]	
rw		rw	rw	rw	rw		rw	rw			rw	rc_w1	rc_w1	rw	

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:14	LDOVS[1:0]	LDO output voltage select These bits are set by software when the main PLL closed. And the LDO output voltage selected by LDOVS bits takes effect when the main PLL enabled. If the main PLL closed, the LDO output voltage low mode selected (value of this bit filed not changed). 0x: LDO output voltage low mode (0.9V). 1x: LDO output voltage high mode (1.1V).
13	VCRSEL	V_{BAT} battery charging resistor selection 0: 5 kOhms resistor is selected for charing V_{BAT} battery. 1: 1.5 kOhms resistor is selected for charing V_{BAT} battery.
12	VCEN	V_{BAT} battery charging enable 0: Disable V_{BAT} battery charging. 1: Enable V_{BAT} battery charging.
11	LDNP	Low-driver mode when use NPLDO in Run / Sleep mode. 0: normal driver when use NPLDO. 1: Low-driver mode enabled when use NPLDO.
10	LDNPDSP	Low-driver mode when use NPLDO in Deep-sleep mode. 0: normal driver when use NPLDO.

		1: Low-driver mode enabled when use NPLDO.
9	Reserved	Must be kept at reset value.
8	BKPWEN	Backup Domain Write Enable 0: Disable write access to the registers in Backup domain. 1: Enable write access to the registers in Backup domain. After reset, any write access to the registers in Backup domain is ignored. This bit has to be set to enable write access to these registers.
7:5	LVDT[2:0]	Low Voltage Detector Threshold 000: 2.1V 001: 2.3V 010: 2.4V 011: 2.6V 100: 2.7V 101: 2.9V 110: 3.0V 111: input analog voltage on PB7 (compared with 0.8V)
4	LVDEN	Low Voltage Detector Enable 0: Disable Low Voltage Detector 1: Enable Low Voltage Detector
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-sleep 1 10: Deep-sleep 2 11: Standby

For GD32L235xx devices

Address offset: 0x00

Reset value: 0x0000 C000 (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
LDOVS[1:0]		VCRSEL	VCEN	LDNP	LDNPDSP	Reserved	BKPWEN	LVDT[2:0]			LVDEN	STBRST	WURST	LPMOD[1:0]	
rw		rw	rw	rw	rw		rw	rw			rw	rc_w1	rc_w1	rw	

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:14	LDOVS[1:0]	LDO output voltage select These bits are set by software when the main PLL closed. And the LDO output voltage selected by LDOVS bits takes effect when the main PLL enabled. If the main PLL closed, the LDO output voltage low mode selected (value of this bit filed not changed). 0x: reserved. 10: The temperature self-adaptive mode (1.1V). 11: LDO output voltage high mode (1.1V). **Note:** The temperature self-adaptive mode is suitable for applications that require high voltage stability.
13	VCRSEL	V_{BAT} battery charging resistor selection 0: 5 kOhms resistor is selected for charing V_{BAT} battery. 1: 1.5 kOhms resistor is selected for charing V_{BAT} battery.
12	VCEN	V_{BAT} battery charging enable 0: Disable V_{BAT} battery charging. 1: Enable V_{BAT} battery charging.
11	LDNP	Low-driver mode when use NPLDO in Run / Sleep mode. 0: normal driver when use NPLDO. 1: Low-driver mode enabled when use NPLDO.
10	LDNPDSP	Low-driver mode when use NPLDO in Deep-sleep mode. 0: normal driver when use NPLDO. 1: Low-driver mode enabled when use NPLDO.
9	Reserved	Must be kept at reset value.
8	BKPWEN	Backup Domain Write Enable 0: Disable write access to the registers in Backup domain. 1: Enable write access to the registers in Backup domain. After reset, any write access to the registers in Backup domain is ignored. This bit has to be set to enable write access to these registers.
7:5	LVDT[2:0]	Low Voltage Detector Threshold 000: 2.1V

		001: 2.3V 010: 2.4V 011: 2.6V 100: 2.7V 101: 2.9V 110: 3.0V 111: input analog voltage on PB7 (compared with 0.8V)
4	LVDEN	Low Voltage Detector Enable 0: Disable Low Voltage Detector 1: Enable Low Voltage Detector
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-sleep 1 10: Deep-sleep 2 11: Standby

3.4.2. Control and status register (PMU_CS)

For GD32L233xx devices

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	Reserved	WUPEN4	WUPEN3	WUPEN2	WUPEN1	WUPEN0	Reserved					LVDF	STBF	WUF
	r		rw	rw	rw	rw	rw						r	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	Reserved	Must be kept at reset value.
12	WUPEN4	WKUP Pin4(PC6) enable 0: Disable WKUP pin4 function. 1: Enable WKUP pin4 function. If WUPEN4 is set before entering the power saving mode, a rising edge on the WKUP pin4 wakes up the system from the power saving mode. As the WKUP pin4 is active high, the WKUP pin4 is internally configured to input pull down mode. And set this bit will trigger a wakup event when the input is aready high.
11	WUPEN3	WKUP Pin3(PB2) 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(PA2) 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) 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.
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.
7:3	Reserved	Must be kept at reset value.
2	LVDF	Low Voltage Detector Status Flag 0: Low Voltage event has not occurred (V_{DD} is higher than the specified LVD threshold). 1: Low Voltage event occurred (V_{DD} is equal to or lower than the specified LVD threshold). **Note:** The LVD function is stopped in Standby mode.
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.

For GD32L235xx devices

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	WUPEN4	WUPEN3	WUPEN2	WUPEN1	WUPEN0	Reserved					LVDF	STBF	WUF
	r	rw	rw	rw	rw	rw	rw						r	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	WUPEN4	WKUP Pin4 (PC6) enable 0: Disable WKUP pin4 function. 1: Enable WKUP pin4 function. If WUPEN4 is set before entering the power saving mode, a rising edge on the WKUP pin4 wakes up the system from the power saving mode. As the WKUP pin4 is active high, the WKUP pin4 is internally configured to input pull down mode. And set this bit will trigger a wakup event when the input is aready high.
11	WUPEN3	WKUP Pin3 (PB2) 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 (PA2) 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) 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.
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.
7:3	Reserved	Must be kept at reset value.
2	LVDF	Low Voltage Detector Status Flag 0: Low Voltage event has not occurred (V_{DD} is higher than the specified LVD threshold). 1: Low Voltage event occurred (V_{DD} is equal to or lower than the specified LVD threshold). **Note:** The LVD function is stopped in Standby mode.
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)

For GD32L233xx devices

Address offset: 0x08

Reset value: 0x0001 0000 (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														SRAM1PD2	NRRD2
														rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Reserved	CORE1WAKE	CORE1SLEEP	Reserved	SRAM1PWAKE	SRAM1PSLEEP
	rw	rw		rw	rw

Bits	Fields	Descriptions
31:18	Reserved	Must be kept at reset value.
17	SRAM1PD2	Power state of SRAM1 when enters Deep-sleep2 mode 0: SRAM1 power-off. 1: SRAM1 power same as Run / Run1 / Run2 mode. **Note:** When wakeup from the Deep-sleep2 mode, the power state of SRAM1 is the same as the power state before entering the Deep-sleep2 mode.
16	NRRD2	No retention register in Deep-sleep 2 mode 0: CPU have retention register. 1: No retention register.
15:6	Reserved	Must be kept at reset value.
5	CORE1WAKE	COREOFF1 domain wakeup. This bit is set by software only in Run / Run1 / Run2 mode and COREOFF1 in sleep mode, and cleared by hardware.
4	CORE1SLEEP	COREOFF1 domain go to power-off. This bit is set by software only in Run / Run1 / Run2 mode and COREOFF1 in active mode, and cleared by hardware.
3:2	Reserved	Must be kept at reset value.
1	SRAM1PWAKE	SRAM1 wakeup. This bit is set by software only in Run / Run1 / Run2 mode and SRAM1 in sleep mode, and cleared by hardware.
0	SRAM1PSLEEP	SRAM1 go to power-off. This bit is set by software only in Run / Run1 / Run2 mode and SRAM1 in active mode, and cleared by hardware.

For GD32L235xx devices

Address offset: 0x08

Reset value: 0x0001 0000 (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														SRAM1PD2	Reserved
														rw	rw

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

Reserved	EFDSPSLEEP	EFPSLEEP	Reserved	SRAM1PWAKE	SRAM1PSLEEP
	rw	rw		rw	rw

Bits	Fields	Descriptions
31:18	Reserved	Must be kept at reset value.
17	SRAM1PD2	Power state of SRAM1 when enters Deep-sleep2 mode 0: SRAM1 power-off. 1: SRAM1 power same as Run mode. **Note:** When wakeup from the Deep-sleep2 mode, the power state of SRAM1 is the same as the power state before entering the Deep-sleep2 mode.
16:6	Reserved	Must be kept at reset value.
5	EFDSPSLEEP	EFLASH power control when in Deep-sleep / Deep-sleep 1 / Deep-sleep 2 mode. 0: EFLASH power on in Deep-sleep / Deep-sleep 1 / Deep-sleep 2 mode. 1: EFLASH power off in Deep-sleep / Deep-sleep 1 / Deep-sleep 2 mode.
4	EFPSLEEP	EFLASH power control when in Run mode. 0: EFLASH power on in Run mode. 1: EFLASH power off in Run mode.
3:2	Reserved	Must be kept at reset value.
1	SRAM1PWAKE	SRAM1 wakeup. This bit is set by software only in Run mode and SRAM1 in sleep mode, and cleared by hardware.
0	SRAM1PSLEEP	SRAM1 go to power-off. This bit is set by software only in Run mode and SRAM1 in active mode, and cleared by hardware.

3.4.4. Status register (PMU_STAT)

For GD32L233xx devices

Address offset: 0x0C

Reset value: 0x0000 0018 (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															
								r			r			r	

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

Reserved	CORE1PS_ACTIVE	CORE1PS_SLEEP	SRAM1PS_ACTIVE	SRAM1PS_SLEEP	DPF2	Reserved
	r	r	r	r	rc_w0	

Bits	Fields	Descriptions
31:6	Reserved	Must be kept at reset value.
5	CORE1PS_ACTIVE	COREOFF1 domain is in active state.
4	CORE1PS_SLEEP	COREOFF1 domain is in sleep state.
3	SRAM1PS_ACTIVE	SRAM1 is in active state.
2	SRAM1PS_SLEEP	SRAM1 is in sleep state.
1	DPF2	This bit is Deep-sleep2 mode status. This bit is set by hardware when enter Deep-sleep2 mode. And clear by software when write 0.
0	Reserved	Must be kept at reset value.

For GD32L235xx devices

Address offset: 0x0C

Reset value: 0x0000 0028 (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	SRAM1PS_ACTIVE	SRAM1PS_SLEEP	DPF2	Reserved
										r	r	r	r	rc_w0	

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	SRAM1PS_ACTIVE	SRAM1 is in active state.
2	SRAM1PS_SLEEP	SRAM1 is in sleep state.
1	DPF2	This bit is Deep-sleep2 mode status. This bit is set by hardware when enter Deep-sleep2 mode. And clear by software when write 0.

0	Reserved	Must be kept at reset value.

3.4.5. Parameter register (PMU_PAR)

For GD32L233xx devices

Address offset: 0x10

Reset value: 0x040A 2064

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
TWKEN	TWKSRAM1EN	TWKCORE1EN	TWK_CORE1[7:0]								TSW_IRC16MCNT[4:0]				
rw	rw	rw	rw								rw				

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TWK_SRAM1[7:0]								TWK_CORE0[7:0]							
rw								rw							

Bits	Fields	Descriptions
31	TWKEN	Use software value when wake up Deep-sleep2 or not 0: use hardware ack signal when wake up Deep-sleep2. 1: use software value when wake up Deep-sleep2, the value is set by TWK_CORE0[7:0].
30	TWKSRAM1EN	Use software value when wake up SRAM1 power domain or not 0: use hardware ack signal when wake up SRAM1. 1: use software value when wake up SRAM1, the value is set by TWK_SRAM1[7:0].
29	TWKCORE1EN	Use software value when wake up COREOFF1 or not 0: use hardware ack signal when wake up COREOFF1. 1: use software value when wake up COREOFF1, the value is set by TWK_CORE1[7:0].
28:21	TWK_CORE1[7:0]	Wakeup time of power switch of COREOFF1 domain. 4 clock step and the max value is 64us.
20:16	TSW_IRC16MCNT[4:0]	When enter deep-sleep mode, switch to IRC16M clock. Wait the IRC16M COUNTER and then set deep-sleep signal. The default is 10 IRC16M clock.
15:8	TWK_SRAM1[7:0]	Wakeup time of power switch of SRAM1 domain. 4 IRC16M clock step and the max value is 64us.
7:0	TWK_CORE0[7:0]	Wakeup time of power switch of COREOFF0 domain. 2 IRC16M clock step and the max value is 32us.

For GD32L235xx devices

Address offset: 0x10

Reset value: 0x000A 2064

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
TWKEN	Reserved		TWK_EFLASH[7:0]								TSW_IRC16MCNT[4:0]				
rw			rw								rw				

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TWK_SRAM1[7:0]								TWK_CORE0[7:0]							
rw								rw							

Bits	Fields	Descriptions
31	TWKEN	Use software value when wake up Deep-sleep2 or not 0: use hardware ack signal when wake up Deep-sleep2. 1: use software value when wake up Deep-sleep2, the value is set by TWK_CORE0[7:0].
30:29	Reserved	Must be kept at reset value.
28:21	TWK_EFLASH[7:0]	EFLASH wake up counter. Wakeup time of power switch of EFLASH. TWK_EFLASH IRC16M clock.
20:16	TSW_IRC16MCNT[4:0]	When enter deep-sleep mode, switch to IRC16M clock. Wait the IRC16M COUNTER and then set deep-sleep signal. The default is 10 IRC16M clock.
15:8	TWK_SRAM1[7:0]	Wakeup time of power switch of SRAM1 domain. 4 IRC16M clock step and the max value is 64us.
7:0	TWK_CORE0[7:0]	Wakeup time of power switch of COREOFF0 domain. 2 IRC16M clock step and the max value is 32us.

4. Reset and clock unit (RCU)

4.1. Reset control unit (RCTL)

4.1.1. Overview

GD32L23x reset control includes the control of three kinds of reset: power reset, system reset and backup domain reset. The power on reset, known as a cold reset, resets the full system except the backup domain 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 domain. A backup domain reset resets the backup domain. 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 except the backup domain. The power reset which active signal is low will be de-asserted when the internal LDO voltage regulator is ready to provide 1.1V power for GD32L23x series. The RESET service routine vector is fixed at address 0x0000_0004 in the memory map.

System Reset

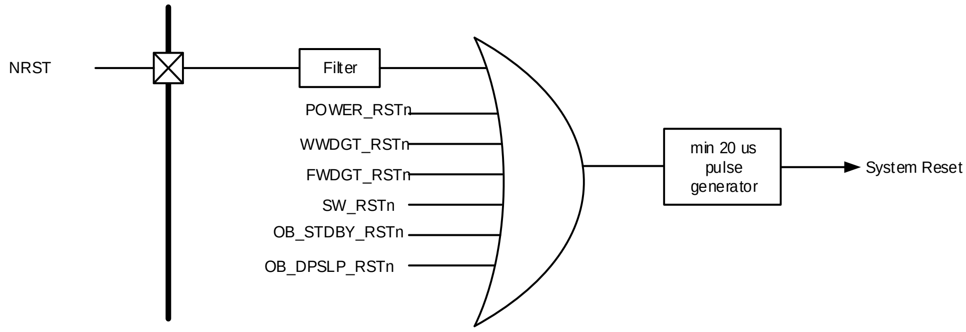
A system reset is generated by the following events:

- 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).
- 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).

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

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 domain reset

A backup domain reset is generated by setting the BKPRST bit in the backup domain control register or backup domain 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 16 MHz RC oscillator (IRC16M), 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 Phase Lock Loop (PLL), a HXTAL clock monitor, a LXTAL clock monitor, clock prescalers, clock multiplexers and clock gating circuitry.

The clocks of the AHB, APB and Cortex®-M23 are derived from the system clock (CK_SYS) which can source from the IRC16M, IRC48M, HXTAL, IRC32K (Only for GD32L235 devices) or PLL. The maximum operating frequency of the system clock (CK_SYS) can be up to 64 MHz.

Figure 4-2. Clock tree of GD32L233xx devices

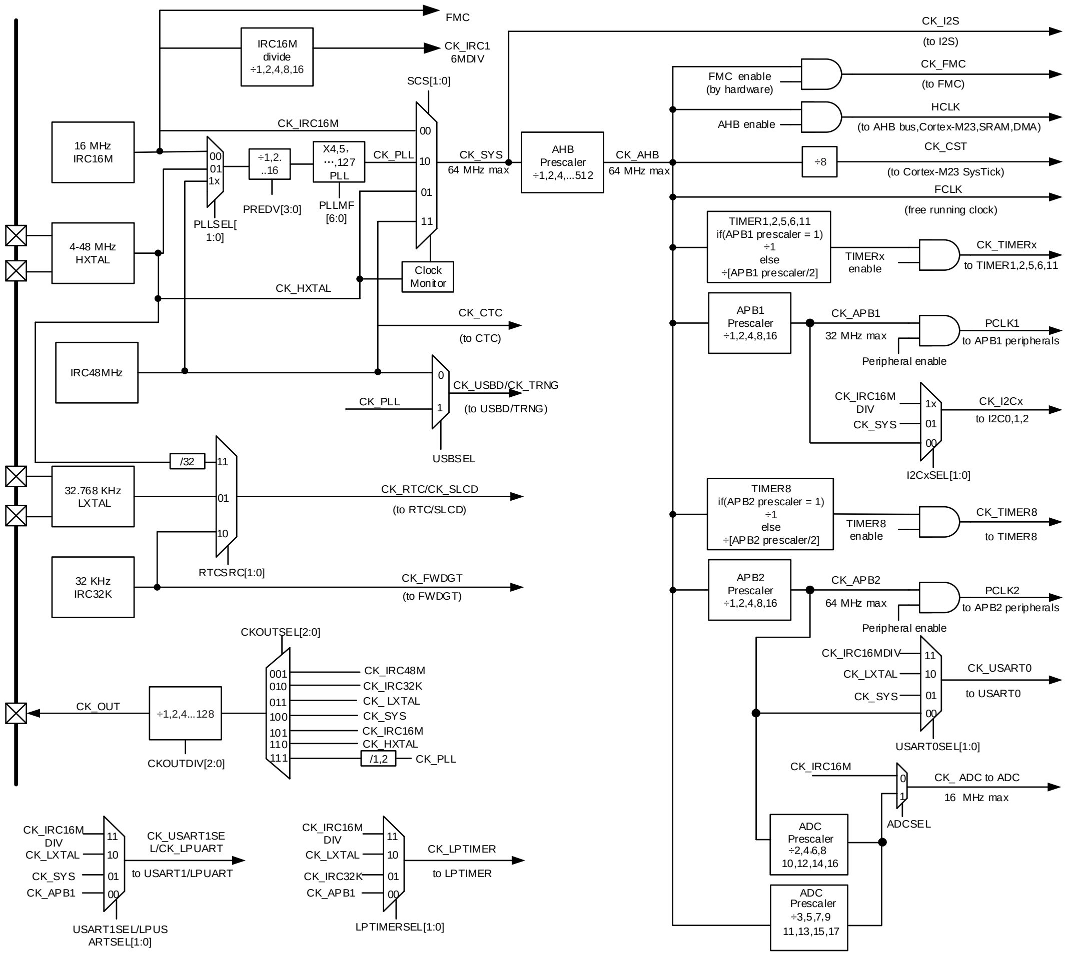
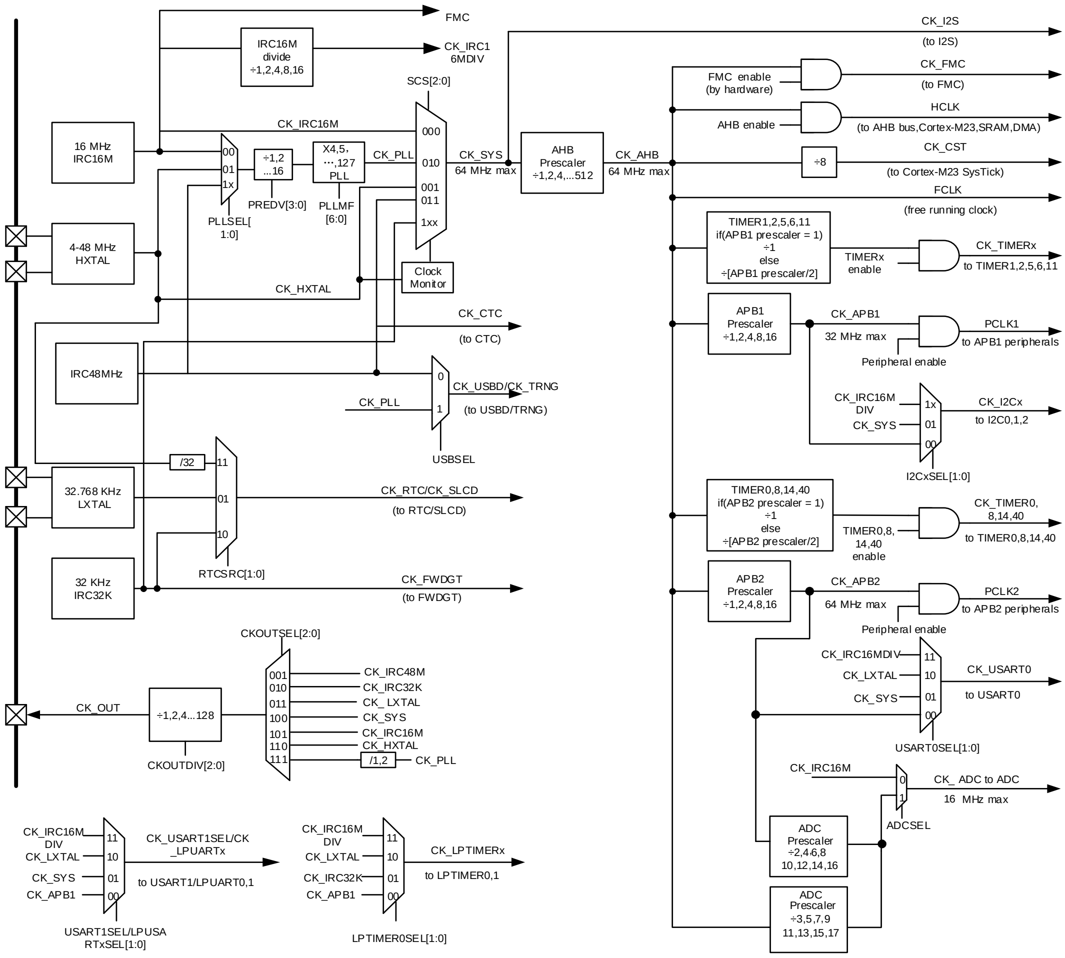


Figure 4-3. Clock tree of GD32L235xx devices



The frequency of AHB, APB2 and the APB1 domains can be configured by each prescaler. The maximum frequency of the AHB, APB2 and APB1 domains is 64MHz / 64MHz / 32MHz. 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 are clocked by the clock of APB2 divided by 2, 4, 6, 8, 10, 12, 14, 16 or by the clock of AHB divided by 3, 5, 7, 9, 11, 13, 15, 17 or IRC16M clock for GD32L23x series selected by ADCSEL bit in configuration register 2 (RCU_CFG2).

The USART0 is clocked by IRC16MDIV clock or LXTAL clock or system clock or APB2 clock, which selected by USART0SEL bits in configuration register 2 (RCU_CFG2).

The USART1 is clocked by IRC16MDIV clock or LXTAL clock or system clock or APB1 clock, which selected by USART1SEL bits in configuration register 2 (RCU_CFG2).

For GD32L233xx devices, the The LPUART is clocked by IRC16MDIV clock or LXTAL clock or system clock or APB1 clock, which selected by LPUARTSEL bits in configuration register 2 (RCU_CFG2). For GD32L235xx devices, the LPUARTx(x = 0, 1) is clocked by IRC16MDIV clock or LXTAL clock or system clock or APB1 clock, which selected by LPUARTxSEL(x = 0,

1) bits in configuration register 2 (RCU_CFG2).

The I2Cx(x = 0, 1, 2) is clocked by IRC16MDIV clock or system clock or APB1 clock, which selected by I2CxSEL(x = 0, 1, 2) bits in configuration register 2 (RCU_CFG2).

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

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

For GD32L233xx devices, the LPTIMER is clocked by IRC16MDIV clock or LXTAL clock or system clock or APB1 clock, which selected by LPTIMERSEL bits in configuration register 2 (RCU_CFG2). For GD32L235xx devices, the LPTIMERx(x = 0, 1) is clocked by IRC16MDIV clock or LXTAL clock or system clock or APB1 clock, which selected by LPTIMERxSEL(x = 0, 1) bits in configuration register 2 (RCU_CFG2).

If the APB prescaler is 1, the timer clock frequencies are set to AHB frequency divide by 1. Otherwise, they are set to the AHB frequency divide by half of APB prescaler.

4.2.2. Characteristics

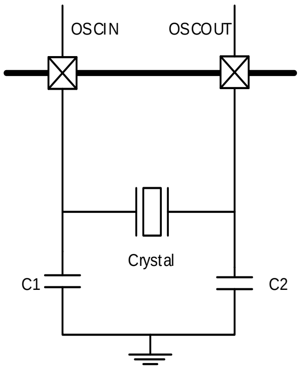
- 4 to 48 MHz High speed crystal oscillator (HXTAL).
- Internal 16 MHz RC oscillator (IRC16M).
- Internal 48 MHz RC oscillator (IRC48M).
- 32.768 KHz Low speed crystal oscillator (LXTAL).
- Internal 32 KHz RC oscillator (IRC32K).
- PLL clock source can be HXTAL, IRC16M, IRC32K or IRC48M.
- HXTAL and 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 32 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.

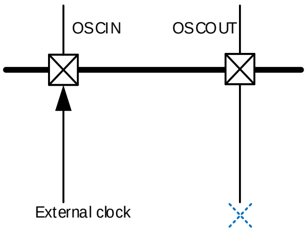
Figure 4-4. HXTAL clock source



The HXTAL crystal oscillator can be switched on or off using the HXTALEN bit in the control register, RCU_CTL. The HXTALSTB flag in control register, RCU_CTL 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 or the PLL input clock.

Select external clock bypass mode by setting the HXTALBPS and HXTALEN bits in the control register RCU_CTL. During bypass mode, the signal is connected to OSCIN, and OSCOUT remains in the suspended state, as shown in ***Figure 4-5. HXTAL clock source in bypass mode***. The CK_HXTAL is equal to the external clock which drives the OSCIN pin.

Figure 4-5. HXTAL clock source in bypass mode



Internal 16 MHz RC oscillator (IRC16M)

The Internal 16 MHz RC oscillator, IRC16M, has a fixed frequency of 16 MHz and is the default clock source selection for the CPU when the device is powered up. The IRC16M oscillator provides a lower cost type clock source as no external components are required. The IRC16M RC oscillator can be switched on or off using the IRC16MEN bit in the control register, RCU_CTL. The IRC16MSTB flag in the control register, RCU_CTL, is used to indicate if the internal RC oscillator is stable. The start-up time of the IRC16M oscillator is shorter than the HXTAL crystal oscillator. An interrupt can be generated if the related interrupt enable bit, IRC16MSTBIE, in the interrupt register, RCU_INT, is set when the IRC16M becomes stable. The IRC16M clock can also be used as the PLL input clock.

The frequency accuracy of the IRC16M can be calibrated by the manufacturer, but its operating frequency is still less accurate than HXTAL. The application requirements, environment and cost will determine which oscillator type is selected.

If the HXTAL or PLL is the system clock source, to minimize the time required for the system to recover from the Deep-sleep Mode, the hardware forces the IRC16M clock to be the system clock when the system initially wakes-up.

IRC16M can be switched on by LPUART / LPUART0 / LPUART1 / USART0 / USART1 / I2C0 / I2C1 / I2C2 during deep-sleep mode. If the IRC16M switch on during deep-sleep state， the unworked peripheral should been disabled for save power.

Internal 48M RC oscillators (IRC48M)

The internal 48M RC oscillator, IRC48M, has a fixed frequency of 48 MHz. The IRC48M oscillator provides a lower cost type clock source as no external components are required when USBD used. 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. The IRC48M clock is used for the clocks of USBD.

The frequency accuracy of the IRC48M can be calibrated by the manufacturer, but its operating frequency is still not enough accurate because the USB need the frequency must between 48MHz with 500ppm accuracy. A hardware automatically dynamic trim performed in CTC unit adjust the IRC48M to the needed frequency.

Phase Locked Loop (PLL)

The internal Phase Locked Loop, PLL, can provide 16 ~ 64 MHz clock output which is 2 ~ 64 multiples of a fundamental reference frequency of 4 ~ 48 MHz.

The PLL can be switched on or off by using the PLLEN bit in the control registe, RCU_CTL. The PLLSTB flag in the control register, RCU_CTL will indicate if the PLL clock is stable. An interrupt can be generated if the related interrupt enable bit, PLLSTBIE, in the interrupt register, RCU_INT, is set as the PLL 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 backup domain control Register (RCU_BDCTL). The LXTALSTB flag in the backup domain control register (RCU_BDCTL) 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 backup domain control register (RCU_BDCTL). The CK_LXTAL is equal to the external clock which drives the OSC32IN pin.

LXTAL can be switched on when LPUART / LPUART0 / LPUART1 / USART0 / USART1 uses LXTAL as function clock.

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 IRC16M and can be switched to HXTAL, PLL, IRC32K(only for GD32L235 devices) or IRC48M 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 or the PLL, 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_CTL. 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 or PLL, the HXTAL failure will force the CK_SYS source to IRC16M and the PLL will be disabled automatically

LXTAL clock monitor (LCKM)

A clock monitor on LXTAL can be activated by software writing the LCKMEN bit in the control register (RCU_CTL). 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_{BAT}. If a failure is detected on the external 32 KHz oscillator, an interrupt can be sent to CPU.

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 64 MHz. There are several clock signals can be selected via the CK_OUT clock source selection bits, CKOUTSEL, 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
001	CK_IRC48M
010	CK_IRC32K
011	CK_LXTAL
100	CK_SYS
101	CK_IRC16M
110	CK_HXTAL
111	CK_PLL or CK_PLL/2

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

Deep-sleep 1 / 2 mode clock control

When the MCU is in deep-sleep 1 / 2 mode, the LPUART / LPUART0 / LPUART1 / USART0 / USART1 can wake up the MCU, when their clock is provided by LXTAL clock and LXTAL clock is enable.

If the LPUART / LPUART0 / LPUART1 / USART0 / USART1 clock is selected IRC16M_DIV clock in deep-sleep 1 / 2 mode, they have capable of open IRC16M clock or close IRC16M clock, which used to the LPUART / USART0 / USART1 / I2C0 / I2C1 / I2C2 to wake up the Deep-sleep mode.

If the LPUART / LPUART0 / LPUART1 / USART0 / USART1 clock is selected LXTAL clock in deep-sleep 1/2 mode, they have capable of open LXTAL clock or close LXTAL clock (if LXTAL is opened by softer, LPUART / LPUART0 / LPUART1 / USART0 / USART1 can't close the LXTAL), which used to the LPUART to wake up the deep-sleep 1/2 mode.

If the I2C0 / I2C1 / I2C2 clock is selected IRC16M_DIV clock in deep-sleep 1/2 mode, they have capable of open IRC16M clock or close IRC16M clock, which used to the I2C0 / I2C1 / I2C2 to wake up the deep-sleep 1/2 mode.

FMC and PMU also have capable of open IRC16M clock or close IRC16M clock, if they work in deep-sleep 1 / 2 mode.

To save power in deep-sleep 1 / 2 mode. CK_FMC and LPUART / LPUART0 / LPUART1 / USART0 / USART1 function clock can be gated individually, if they don’t work in deep-sleep 1/2 mode mode. But I2C0 / I2C1 / I2C2, ADC, LPTIMER, LPTIMER0, LPTIMER1 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 (RCU_CTL)

Address offset: 0x00

Reset value: 0x0000 XX83 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
Reserved						PLLSTB	PLLEN	LCKMD	LCKMEN	IRC48MSTB	IRC48MEN	CKMEN	HXTALBPS	HXTALSTB	HXTALEN
						r	rw	r	rw	r	rw	rw	rw	r	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IRC16MCALIB[7:0]								IRC16MADJ[4:0]					Reserved	IRC16MSTB	IRC16MEN
r								rw						r	rw

Bits	Fields	Descriptions
31:26	Reserved	Must be kept at reset value.
25	PLLSTB	PLL clock stabilization flag Set by hardware to indicate if the PLL output clock is stable and ready for use. 0: PLL is not stable 1: PLL is stable
24	PLLEN	PLL enable Set and reset by software. This bit cannot be reset if the PLL clock is used as the system clock. Reset by hardware when entering Deep-sleep or Standby mode. 0: PLL is switched off 1: PLL is switched on
23	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 disable LXTALEN or LXTAL gets right. 0: No failure detected on LXTAL (32 KHz oscillator) 1: Failure detected on LXTAL (32 KHz oscillator)
22	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).
21	IRC48MSTB	IRC48M 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
20	IRC48MEN	Internal high speed oscillator enable Set and reset by software. 0: Internal 48 MHz RC oscillator disabled 1: Internal 48 MHz RC oscillator enabled
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 IRC16M 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 IRC16M internal RC oscillator regardless of the control bit, IRC16MEN, 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 or the PLL input 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	IRC16MCALIB[7:0]	Internal 16M RC oscillator calibration value register These bits are load automatically at power on.
7:3	IRC16MADJ[4:0]	Internal 16M RC oscillator clock trim adjust value These bits are set by software. The trimming value is there bits (IRC16MADJ) added to the IRC16MCALIB[7:0] bits. The trimming value should trim the IRC16M to 16

		MHz ± 1%.
2	Reserved	Must be kept at reset value.
1	IRC16MSTB	IRC16M high speed internal oscillator stabilization flag Set by hardware to indicate if the IRC16M oscillator is stable and ready for use. 0: IRC16M oscillator is not stable 1: IRC16M oscillator is stable
0	IRC16MEN	Internal high speed oscillator enable Set and reset by software. This bit cannot be reset if the IRC16M 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 16 MHz RC oscillator disabled 1: Internal 16 MHz RC oscillator enabled

4.3.2. Configuration register 0 (RCU_CFG0)

Address offset: 0x04

Reset value: 0x003C 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
PLLDV	CKOUTDIV[2:0]			PLLMF[6]	CKOUTSEL[2:0]			PLLMF[5:0]						PLLSEL	
rw	rw			rw	rw			rw						rw	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADCPSC[1:0]		APB2PSC[2:0]			APB1PSC[2:0]			AHBPSC[3:0]				SCSS[1:0]		SCS[1:0]	
rw		rw			rw			rw				r		rw	

Bits	Fields	Descriptions
31	PLLDV	The CK_PLL divide by 1 or 2 for CK_OUT 0: CK_PLL divide by 2 for CK_OUT 1: CK_PLL divide by 1 for CK_OUT
30:28	CKOUTDIV[2:0]	The CK_OUT divider which the CK_OUT frequency can be reduced, see bits 26:24 of RCU_CFG0 for CK_OUT. 000: The CK_OUT is divided by 1 001: The CK_OUT is divided by 2 010: The CK_OUT is divided by 4 011: The CK_OUT is divided by 8 100: The CK_OUT is divided by 16 101: The CK_OUT is divided by 32 110: The CK_OUT is divided by 64 111: The CK_OUT is divided by 128
27	PLLMF[6]	Bit 6 of PLLMF

		see bits 23:18 of RCU_CFG0
26:24	CKOUTSEL[2:0]	CK_OUT clock source selection Set and reset by software. 000: No clock selected 001: Internal 48MHz RC oscillator clock selected 010: Internal 32K RC oscillator clock selected 011: External low speed oscillator clock selected 100: System clock selected 101: Internal 16MHz RC oscillator clock selected 110: External high speed oscillator clock selected 111: (CK_PLL / 2) or CK_PLL selected depend on PLLDV
23:18	PLLMF[5:0]	PLL multiply factor These bits are written by software to define the PLL multiplication factor. 0000000~0000001: Reserved 0000010~0001110: (PLL source clock x (PLLMF[6:0] + 2)) 0001111~1111110: (PLL source clock x (PLLMF[6:0] + 1)) 1111111: Reserved **Note:** The PLL output frequency must not exceed 64 MHz.
17:16	PLLSEL	PLL clock source selection Set and reset by software to control the PLL clock source. 00: IRC16M clock selected as source clock of PLL 01: HXTAL selected as source clock of PLL 1x: IRC48M clock selected as source clock of PLL
15:14	ADCPSC[1:0]	ADC clock prescaler selection These bits and bit [31:30] of RCU_CFG2 are written by software to define the ADC clock prescaler. Set and cleared by software. 0000: (CK_APB2 / 2) selected 0001: (CK_APB2 / 4) selected 0010: (CK_APB2 / 6) selected 0011: (CK_APB2 / 8) selected 0100: (CK_APB2 / 10) selected 0101: (CK_APB2 / 12) selected 0110: (CK_APB2 / 14) selected 0111: (CK_APB2 /16) selected 1000: (CK_AHB / 3) selected 1001: (CK_AHB / 5) selected 1010: (CK_AHB / 7) selected 1011: (CK_AHB / 9) selected 1100: (CK_AHB / 11) selected 1101: (CK_AHB / 13) selected 1110: (CK_AHB / 15) selected

		1111: (CK_AHB / 17) selected
13:11	APB2PSC[2:0]	APB2 prescaler selection Set and reset by software to control the APB2 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	APB1PSC[2:0]	APB1 prescaler selection Set and reset by software to control the APB1 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
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. **For GD32L233xx devices** 00: select CK_IRC16M as the CK_SYS source 01: select CK_HXTAL as the CK_SYS source 10: select CK_PLL as the CK_SYS source 11: Select CK_IRC48M as the CK_SYS source **For GD32L235xx devices** 000: Select CK_IRC16M as the CK_SYS source 001: Select CK_HXTAL as the CK_SYS source 010: Select CK_PLL as the CK_SYS source 011: Select CK_IRC48M as the CK_SYS source 1xx: Select CK_IRC32K as the CK_SYS source **Note:** SCSS[2] is in RCU_CFG1 bit[17]
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. The switch will be forced to IRC16M when leaving Deep-sleep and Standby mode or by HXTAL clock monitor when the HXTAL failure is detected and the HXTAL is selected as the clock source of CK_SYS or PLL.

For GD32L233xx devices

00: Select CK_IRC16M as the CK_SYS source

01: Select CK_HXTAL as the CK_SYS source

10: Select CK_PLL as the CK_SYS source

11: Select CK_IRC48M as the CK_SYS source

For GD32L235xx devices

000: Select CK_IRC16M as the CK_SYS source

001: Select CK_HXTAL as the CK_SYS source

010: Select CK_PLL as the CK_SYS source

011: Select CK_IRC48M as the CK_SYS source

1xx: Select CK_IRC32K as the CK_SYS source

Note: SCS[2] is in RCU_CFG1 bit[16]

4.3.3. Interrupt register (RCU_INT)

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								CKMIC	LCKMIC	IRC48M STBIC	PLL STBIC	HXTAL STBIC	IRC16M STBIC	LXTAL STBIC	IRC32K STBIC
								w	w	w	w	w	w	w	w

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	LCKMIE	IRC48M STBIE	PLL STBIE	HXTAL STBIE	IRC16M STBIE	LXTAL STBIE	IRC32K STBIE	CKMIF	LCKMIF	IRC48M STBIF	PLL STBIF	HXTAL STBIF	IRC16M STBIF	LXTAL STBIF	IRC32K STBIF
	rw	rw	rw	rw	rw	rw	rw	r	r	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	IRC48MSTBIC	IRC48M stabilization interrupt clear Write 1 by software to reset the IRC48MSTBIF flag. 0: Not reset IRC48MSTBIF flag 1: Reset IRC48MSTBIF flag
20	PLLSTBIC	PLL stabilization interrupt clear Write 1 by software to reset the PLLSTBIF flag. 0: Not reset PLLSTBIF flag 1: Reset PLLSTBIF flag
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	IRC16MSTBIC	IRC16M stabilization interrupt clear Write 1 by software to reset the IRC16MSTBIF flag. 0: Not reset IRC16MSTBIF flag 1: Reset IRC16MSTBIF 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	Reserved	Must be kept at reset value.
14	LCKMIE	LXTAL clock stuck interrupt enable Set and reset by software to enable/disable the LXTAL clock stuck interrupt. 0: Disable the LXTAL clock stuck interrupt 1: Enable the LXTAL clock stuck interrupt
13	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
12	PLLSTBIE	PLL stabilization interrupt enable Set and reset by software to enable/disable the PLL stabilization interrupt. 0: Disable the PLL stabilization interrupt 1: Enable the PLL stabilization interrupt

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	IRC16MSTBIE	IRC16M stabilization interrupt enable Set and reset by software to enable/disable the IRC16M stabilization interrupt 0: Disable the IRC16M stabilization interrupt 1: Enable the IRC16M 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	IRC48MSTBIF	IRC48M stabilization interrupt flag Set by hardware when the IRC48M 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
4	PLLSTBIF	PLL stabilization interrupt flag Set by hardware when the PLL is stable and the PLLSTBIE bit is set. Reset by software when setting the PLLSTBIC bit. 0: No PLL stabilization interrupt generated 1: PLL stabilization interrupt generated
3	HXTALSTBIF	HXTAL stabilization interrupt flag Set by hardware when the external 4 ~ 48 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	IRC16MSTBIF	IRC16M stabilization interrupt flag Set by hardware when the internal 16 MHz RC oscillator clock is stable and the IRC16MSTBIE bit is set. Reset by software when setting the IRC16MSTBIC bit. 0: No IRC16M stabilization interrupt generated 1: IRC16M 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. APB2 reset register (RCU_APB2RST)

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														TIMER40RST	TIMER14RST
														rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	USART0RST	Reserved.	SPI0RST	TIMER8RST	TIMER0RST	ADCRST	Reserved							CMPRST	SYSCFGRST
	rw		rw	rw	rw	rw								rw	rw

Bits	Fields	Descriptions
31:18	Reserved	Must be kept at reset value.
17	TIMER40RST	TIMER40 reset This bit is set and reset by software. 0: No reset

		1: Reset the TIMER40
16	TIMER14RST	TIMER14 reset This bit is set and reset by software. 0: No reset 1: Reset the TIMER14
14	USART0RST	USART0 Reset This bit is set and reset by software. 0: No reset 1: Reset the USART0
13	Reserved	Must be kept at reset value.
12	SPI0RST	SPI0 Reset This bit is set and reset by software. 0: No reset 1: Reset the SPI0
11	TIMER8RST	TIMER8 reset This bit is set and reset by software. 0: No reset 1: Reset the TIMER8
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
8:2	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

4.3.5. APB1 reset register (RCU_APB1RST)

For GD32L233xx devices

Address offset: 0x10

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	CTCRST	DACRST	PMURST	Reserved			I2C2RST	USBDRST	I2C1RST	I2C0RST	UART4RST	UART3RST	LPUARTRST	USART1RST	Reserved
	rw	rw	rw				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	SPI1RST	Reserved		WWDGTRST	SLCDRST	LPTIMERRST	TIMER11RST	Reserved		TIMER6RST	TIMER5RST	Reserved		TIMER2RST	TIMER1RST
	rw			rw	rw	rw	rw			rw	rw			rw	rw

Bits	Fields	Descriptions
31	Reserved	Must be kept at reset value.
30	CTCRST	CTC reset This bit is set and reset by software. 0: No reset 1: Reset CTC
29	DACRST	DAC reset This bit is set and reset by software. 0: No reset 1: Reset DAC
28	PMURST	Power control reset This bit is set and reset by software. 0: No reset 1: Reset power control unit
27:25	Reserved	Must be kept at reset value.
24	I2C2RST	I2C2 reset This bit is set and reset by software. 0: No reset 1: Reset I2C2
23	USBDRST	USBD reset This bit is set and reset by software. 0: No reset 1: Reset USBD

22	I2C1RST	I2C1 reset This bit is set and reset by software. 0: No reset 1: Reset I2C1
21	I2C0RST	I2C0 reset This bit is set and reset by software. 0: No reset 1: Reset I2C0
20	UART4RST	UART4 reset This bit is set and reset by software. 0: No reset 1: Reset UART4
19	UART3RST	UART3 reset This bit is set and reset by software. 0: No reset 1: Reset UART3
18	LPUARTRST	LPUART reset This bit is set and reset by software. 0: No reset 1: Reset LPUART
17	USART1RST	USART1 reset This bit is set and reset by software. 0: No reset 1: Reset USART1
16:15	Reserved	Must be kept at reset value.
14	SPI1RST	SPI1 reset This bit is set and reset by software. 0: No reset 1: Reset SPI1
13:12	Reserved	Must be kept at reset value.
11	WWDGTRST	Window watchdog timer reset This bit is set and reset by software. 0: No reset 1: Reset window watchdog timer
10	SLCDRST	SLCD reset This bit is set and reset by software. 0: No reset 1: Reset SLCD
9	LPTIMERRST	LPTIMER timer reset

		This bit is set and reset by software. 0: No reset 1: Reset LPTIMER timer
8	TIMER11RST	TIMER11 timer reset This bit is set and reset by software. 0: No reset 1: Reset TIMER11 timer
7:6	Reserved	Must be kept at reset value.
5	TIMER6RST	TIMER6 timer reset This bit is set and reset by software. 0: No reset 1: Reset TIMER6 timer
4	TIMER5RST	TIMER5 timer reset This bit is set and reset by software. 0: No reset 1: Reset TIMER5 timer
3:2	Reserved	Must be kept at reset value.
1	TIMER2RST	TIMER2 timer reset This bit is set and reset by software. 0: No reset 1: Reset TIMER2 timer
0	TIMER1RST	TIMER1 timer reset This bit is set and reset by software. 0: No reset 1: Reset TIMER1 timer

For GD32L235xx devices

Address offset: 0x10

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	CTCRST	DACRST	PMURST	Reserved		LPUART1RST	I2C2RST	USBDRST	I2C1RST	I2C0RST	UART4RST	UART3RST	LPUART0RST	USART1RST	CANRST
	rw	rw	rw			rw	rw	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	SPI1RST	Reserved	LPTIMER1RST	WWDGTRST	SLCDRST	LPTIMER0RST	TIMER11RST	Reserved		TIMER6RST	TIMER5RST	Reserved		TIMER2RST	TIMER1RST
	rw		rw	rw	rw	rw	rw			rw	rw			rw	rw

Bits	Fields	Descriptions
31	Reserved	Must be kept at reset value
30	CTCRST	CTC reset This bit is set and reset by software. 0: No reset 1: Reset CTC
29	DACRST	DAC reset This bit is set and reset by software. 0: No reset 1: Reset DAC
28	PMURST	Power control reset This bit is set and reset by software. 0: No reset 1: Reset power control unit
27:26	Reserved	Must be kept at reset value.
25	LPUART1RST	LPUART1 reset This bit is set and reset by software. 0: No reset 1: Reset LPUART1
24	I2C2RST	I2C2 reset This bit is set and reset by software. 0: No reset 1: Reset I2C2
23	USBDRST	USBD reset This bit is set and reset by software. 0: No reset 1: Reset USBD
22	I2C1RST	I2C1 reset This bit is set and reset by software. 0: No reset 1: Reset I2C1
21	I2C0RST	I2C0 reset This bit is set and reset by software. 0: No reset 1: Reset I2C0
20	UART4RST	UART4 reset This bit is set and reset by software. 0: No reset 1: Reset UART4

19	UART3RST	UART3 reset This bit is set and reset by software. 0: No reset 1: Reset UART3
18	LPUART0RST	LPUART0 reset This bit is set and reset by software. 0: No reset 1: Reset LPUART0
17	USART1RST	USART1 reset This bit is set and reset by software. 0: No reset 1: Reset USART1
16	CANRST	CAN reset This bit is set and reset by software. 0: No reset 1: Reset CAN
15	Reserved	Must be kept at reset value.
14	SPI1RST	SPI1 reset This bit is set and reset by software. 0: No reset 1: Reset SPI1
13	Reserved	Must be kept at reset value.
12	LPTIMER1RST	LPTIMER1 timer reset This bit is set and reset by software. 0: No reset 1: Reset LPTIMER1 timer
11	WWDGTRST	Window watchdog timer reset This bit is set and reset by software. 0: No reset 1: Reset window watchdog timer
10	SLCDRST	SLCD reset This bit is set and reset by software. 0: No reset 1: Reset SLCD
9	LPTIMER0RST	LPTIMER timer reset This bit is set and reset by software. 0: No reset 1: Reset LPTIMER0 timer
8	TIMER11RST	TIMER11 timer reset

		This bit is set and reset by software. 0: No reset 1: Reset TIMER11 timer
7:6	Reserved	Must be kept at reset value.
5	TIMER6RST	TIMER6 timer reset This bit is set and reset by software. 0: No reset 1: Reset TIMER6 timer
4	TIMER5RST	TIMER5 timer reset This bit is set and reset by software. 0: No reset 1: Reset TIMER5 timer
3:2	Reserved	Must be kept at reset value.
1	TIMER2RST	TIMER2 timer reset This bit is set and reset by software. 0: No reset 1: Reset TIMER2 timer
0	TIMER1RST	TIMER1 timer reset This bit is set and reset by software. 0: No reset 1: Reset TIMER1 timer

4.3.6. AHB enable register (RCU_AHBEN)

Address offset: 0x14

Reset value: 0x0000 0014

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								SRAM1SPEN	CRCEN	Reserved	FMCSPEN	Reserved	SRAM0SPEN	Reserved	DMAEN
								rw	rw		rw		rw		rw

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
16:8	Reserved	Must be kept at reset value.
7	SRAM1SPEN	SRAM1 interface clock enable This bit is set and reset by software to enable/disable SRAM1 interface clock during seep mode. 0: Disabled SRAM1 interface clock during seep mode. 1: Enabled SRAM1 interface clock during seep mode
6	CRCEN	CRC clock enable This bit is set and reset by software. 0: Disabled CRC clock 1: Enabled CRC clock
5	Reserved	Must be kept at reset value.
4	FMCSPEN	FMC clock enable This bit is set and reset by software to enable/disable FMC clock during Sleep mode. 0: Disabled FMC clock during sleep mode 1: Enabled FMC clock during sleep mode
3	Reserved	Must be kept at reset value.
2	SRAM0SPEN	SRAM0 interface clock enable This bit is set and reset by software to enable / disable SRAM0 interface clock

		during Sleep mode. 0: Disabled SRAM0 interface clock during Sleep mode. 1: Enabled SRAM0 interface clock during Sleep mode
1	Reserved	Must be kept at reset value.
0	DMAEN	DMA clock enable This bit is set and reset by software. 0: Disabled DMA clock 1: Enabled DMA clock

4.3.7. APB2 enable register (RCU_APB2EN)

Address offset: 0x18

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									DBGMCUEN	Reserved				TIMER40EN	TIMER14EN
									rw					rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	USART0EN	Reserved	SPI0EN	TIMER8EN	TIMER0EN	ADCEN	Reserved							CMPEN	SYSCFGEN
	rw		rw	rw	rw	rw								rw	rw

Bits	Fields	Descriptions
31:23	Reserved	Must be kept at reset value.
22	DBGMCUEN	DBGMCU clock enable This bit is set and reset by software. 0: Disabled DBGMCU clock 1: Enabled DBGMCU clock
21:18	Reserved	Must be kept at reset value.
17	TIMER40EN	TIMER40 timer clock enable This bit is set and reset by software. 0: Disabled TIMER40 timer clock 1: Enabled TIMER40 timer clock
16	TIMER14EN	TIMER14 timer clock enable This bit is set and reset by software. 0: Disabled TIMER14 timer clock 1: Enabled TIMER14 timer clock
15	Reserved	Must be kept at reset value.

14	USART0EN	USART0 clock enable This bit is set and reset by software. 0: Disabled USART0 clock 1: Enabled USART0 clock
13	Reserved	Must be kept at reset value.
12	SPI0EN	SPI0 clock enable This bit is set and reset by software. 0: Disabled SPI0 clock 1: Enabled SPI0 clock
11	TIMER8EN	TIMER8 timer clock enable This bit is set and reset by software. 0: Disabled TIMER8 timer clock 1: Enabled TIMER8 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
8:2	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

4.3.8. APB1 enable register (RCU_APB1EN)

For GD32L233xx devices

Address offset:0x1C

Reset value: 0x1000 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

BKPEN	CTCEN	DACEN	PMUEN	Reserved			I2C2EN	USBDEN	I2C1EN	I2C0EN	UART4EN	UART3EN	LPUARTEN	USART1EN	Reserved
rw	rw	rw	rw				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	SPI1EN	Reserved		WWDGTEN	SLCDEN	LPTIMEREN	TIMER11EN	Reserved		TIMER6EN	TIMER5EN	Reserved		TIMER2EN	TIMER1EN
	rw			rw	rw	rw	rw			rw	rw			rw	rw

Bits	Fields	Descriptions
31	BKPEN	BKP (RTC) clock enable This bit is set and reset by software. 0: Disabled BKP(RTC) clock 1: Enabled BKP (RTC) clock
30	CTCEN	CTC clock enable This bit is set and reset by software. 0: Disabled CTC clock 1: Enabled CTC clock
29	DACEN	DAC clock enable This bit is set and reset by software. 0: Disabled DAC clock 1: Enabled DAC clock
28	PMUEN	Power interface clock enable This bit is set and reset by software. 0: Disabled Power interface clock 1: Enabled Power interface clock
27:25	Reserved	Must be kept at reset value.
24	I2C2EN	I2C2 clock enable This bit is set and reset by software. 0: Disabled I2C2 clock 1: Enabled I2C2 clock
23	USBDEN	USBDclock enable This bit is set and reset by software. 0: Disabled USBD clock 1: Enabled USBD clock
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	UART4EN	UART4 clock enable This bit is set and reset by software. 0: Disabled UART4 clock 1: Enabled UART4 clock
19	UART3EN	UART3 clock enable This bit is set and reset by software. 0: Disabled UART3 clock 1: Enabled UART3 clock
18	LPUARTEN	LPUART clock enable This bit is set and reset by software. 0: Disabled LPUART clock 1: Enabled LPUART clock
17	USART1EN	USART1 clock enable This bit is set and reset by software. 0: Disabled USART1 clock 1: Enabled USART1 clock
16:15	Reserved	Must be kept at reset value.
14	SPI1EN	SPI1 clock enable This bit is set and reset by software. 0: Disabled SPI1 clock 1: Enabled SPI1 clock
13:12	Reserved	Must be kept at reset value.
11	WWDGTEN	Window watchdog timer clock enable This bit is set and reset by software. 0: Disabled window watchdog timer clock 1: Enabled window watchdog timer clock
10	SLCDEN	SLCD clock enable This bit is set and reset by software. 0: Disabled SLCD clock 1: Enabled SLCD clock
9	LPTIMEREN	LPTIMER timer clock enable This bit is set and reset by software. 0: Disabled LPTIMER timer clock 1: Enabled LPTIMER timer clock
8	TIMER11EN	TIMER11 timer clock enable This bit is set and reset by software.

		0: Disabled TIMER11 timer clock 1: Enabled TIMER11 timer clock
7:6	Reserved	Must be kept at reset value.
5	TIMER6EN	TIMER6 timer clock enable This bit is set and reset by software. 0: Disabled TIMER6 timer clock 1: Enabled TIMER6 timer clock
4	TIMER5EN	TIMER5 timer clock enable This bit is set and reset by software. 0: Disabled TIMER5 timer clock 1: Enabled TIMER5 timer clock
3:2	Reserved	Must be kept at reset value.
1	TIMER2EN	TIMER2 timer clock enable This bit is set and reset by software. 0: Disabled TIMER2 timer clock 1: Enabled TIMER2 timer clock
0	TIMER1EN	TIMER1 timer clock enable This bit is set and reset by software. 0: Disabled TIMER1 timer clock 1: Enabled TIMER1 timer clock

For GD32L235xx devices

Address offset:0x1C

Reset value: 0x1000 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
BKPEN	CTCEN	DACEN	PMUEN	Reserved		LPUART1EN	I2C2EN	USBDEN	I2C1EN	I2C0EN	UART4EN	UART3EN	LPUART0EN	USART1EN	CANEN
rw	rw	rw	rw			rw	rw	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	SPI1EN	Reserved	LPTIMER1EN	WWDGTEN	SLCDEN	LPTIMER0EN	TIMER11EN	Reserved		TIMER6EN	TIMER5EN	Reserved		TIMER2EN	TIMER1EN
	rw		rw	rw	rw	rw	rw			rw	rw			rw	rw

Bits	Fields	Descriptions
31	BKPEN	BKP (RTC) clock enable This bit is set and reset by software. 0: Disabled BKP(RTC) clock 1: Enabled BKP (RTC) clock

30	CTCEN	CTC clock enable This bit is set and reset by software. 0: Disabled CTC clock 1: Enabled CTC clock
29	DACEN	DAC clock enable This bit is set and reset by software. 0: Disabled DAC clock 1: Enabled DAC clock
28	PMUEN	Power interface clock enable This bit is set and reset by software. 0: Disabled Power interface clock 1: Enabled Power interface clock
27:26	Reserved	Must be kept at reset value.
25	LPUART1EN	LPUART1 clock enable This bit is set and reset by software. 0: Disabled LPUART1 clock 1: Enabled LPUART1 clock
24	I2C2EN	I2C2 clock enable This bit is set and reset by software. 0: Disabled I2C2 clock 1: Enabled I2C2 clock
23	USBDEN	USBDclock enable This bit is set and reset by software. 0: Disabled USBD clock 1: Enabled USBD clock
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	UART4EN	UART4 clock enable This bit is set and reset by software. 0: Disabled UART4 clock 1: Enabled UART4 clock
19	UART3EN	UART3 clock enable This bit is set and reset by software.

		0: Disabled UART3 clock 1: Enabled UART3 clock
18	LPUART0EN	LPUART0 clock enable This bit is set and reset by software. 0: Disabled LPUART0 clock 1: Enabled LPUART0 clock
17	USART1EN	USART1 clock enable This bit is set and reset by software. 0: Disabled USART1 clock 1: Enabled USART1 clock
16	CANEN	CAN clock enable This bit is set and reset by software. 0: Disabled CAN clock 1: Enabled CAN clock
15	Reserved	Must be kept at reset value.
14	SPI1EN	SPI1 clock enable This bit is set and reset by software. 0: Disabled SPI1 clock 1: Enabled SPI1 clock
13	Reserved	Must be kept at reset value.
12	LPTIMER1EN	LPTIMER1 timer clock enable This bit is set and reset by software. 0: Disabled LPTIMER1 timer clock 1: Enabled LPTIMER1 timer clock
11	WWDGTEN	Window watchdog timer clock enable This bit is set and reset by software. 0: Disabled window watchdog timer clock 1: Enabled window watchdog timer clock
10	SLCDEN	SLCD clock enable This bit is set and reset by software. 0: Disabled SLCD clock 1: Enabled SLCD clock
9	LPTIMER0EN	LPTIMER0 timer clock enable This bit is set and reset by software. 0: Disabled LPTIMER0 timer clock 1: Enabled LPTIMER0 timer clock
8	TIMER11EN	TIMER11 timer clock enable This bit is set and reset by software. 0: Disabled TIMER11 timer clock

		1: Enabled TIMER11 timer clock
7:6	Reserved	Must be kept at reset value.
5	TIMER6EN	TIMER6 timer clock enable This bit is set and reset by software. 0: Disabled TIMER6 timer clock 1: Enabled TIMER6 timer clock
4	TIMER5EN	TIMER5 timer clock enable This bit is set and reset by software. 0: Disabled TIMER5 timer clock 1: Enabled TIMER5 timer clock
3:2	Reserved	Must be kept at reset value.
1	TIMER2EN	TIMER2 timer clock enable This bit is set and reset by software. 0: Disabled TIMER2 timer clock 1: Enabled TIMER2 timer clock
0	TIMER1EN	TIMER1 timer clock enable This bit is set and reset by software. 0: Disabled TIMER1 timer clock 1: Enabled TIMER1 timer clock

4.3.9. Backup domain control register (RCU_BDCTL)

Address offset: 0x20

Reset value: 0x0000 0018, reset by backup domain 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 Backup domain control register (BDCTL) are only reset after a Backup domain Reset. These bits can be modified only when the BKPWEN bit in the power control register (PMU_CTL) has to be set.

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

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

Bits	Fields	Descriptions
31:17	Reserved	Must be kept at reset value.
16	BKPRST	Backup domain reset

		This bit is set and reset by software. 0: No reset 1: Resets backup domain
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 domain 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
7:5	Reserved	Must be kept at reset value.
4:3	LXTALDRI[1:0]	LXTAL drive capability Set and reset by software. Backup domain reset reset this value. 00: Lower driving capability 01: Medium low driving capability 10: Medium high driving capability 11: 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

4.3.10. Reset source /clock register (RCU_RSTSCK)

Address offset: 0x24

Reset value: 0x0C80 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	V11RSTF	Reserved						
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	V11RSTF	1.1V domain Power reset flag Set by hardware when a 1.1V domain power reset generated. Reset by writing 1 to the RSTFC bit. 0: No 1.1V domain Power reset generated 1: 1.1V domain Power 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

4.3.11. AHB reset register (RCU_AHBRST)

Address offset: 0x28

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									PFRST	Reserved	PDRST	PCRST	PBRST	PARST	Reserved
									rw		rw	rw	rw	rw	

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

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
16:7	Reserved	Must be kept at reset value.
6	CRCRST	CRC reset This bit is set and reset by software. 0: No reset CRC module 1: Reset CRC module
5:0	Reserved	Must be kept at reset value.

4.3.12. Configuration register 1 (RCU_CFG1)

Address offset: 0x2C

Reset value: 0x0000 0007

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														SSCS[2]	SCS[2]
														r	rw

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

Bits	Fields	Descriptions
31:18	Reserved	Must be kept at reset value.

Bits	Fields	Descriptions
17	SSCS[2]	Bit 2 of SSCS, only for GD32L235xx devieces see bits 3:2 of RCU_CFG0
16	SCS[2]	Bit 2 of SCS, only for GD32L235xx devieces see bits 1:0 of RCU_CFG0
15:4	Reserved	Must be kept at reset value.
3:0	PREDV[3:0]	PLL source clocks pre-divider This bit is set and reset by software. These bits can be written when PLL is disable The source clock is divided by (PREDV + 1). 0000: PREDV input source clock not divided 0001: PREDV input source clock divided by 2 0010: PREDV input source clock divided by 3 0011: PREDV input source clock divided by 4 0100: PREDV input source clock divided by 5 0101: PREDV input source clock divided by 6 0110: PREDV input source clock divided by 7 0111: PREDV input source clock divided by 8 1000: PREDV input source clock divided by 9 1001: PREDV input source clock divided by 10 1010: PREDV input source clock divided by 11 1011: PREDV input source clock divided by 12 1100: PREDV input source clock divided by 13 1101: PREDV input source clock divided by 14 1110: PREDV input source clock divided by 15 1111: PREDV input source clock divided by 16

4.3.13. Configuration register 2 (RCU_CFG2)

For GD32L233xx devices

Address offset: 0x30

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
ADCPSC[3:2]		Reserved									IRC16MDIVSEL			USART1SEL[1:0]	
rw											rw			rw	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved		USBDSEL	LPUARTSEL[1:0]		LPTIMERSEL[1:0]		ADCSEL	I2C2SEL[1:0]		I2C1SEL[1:0]		I2C0SEL[1:0]		USART0SEL[1:0]	
		rw	rw		rw		rw	rw		rw		rw		rw	

Bits	Fields	Descriptions

31:30	ADCPSC[3:2]	Bit 3 and bit 2 of ADCPSC see bits 15:14 of RCU_CFG0
29:21	Reserved	Must be kept at reset value.
20:18	IRC16MDIVSEL	CK_IRC16M divided clock selection 0xx: CK_IRC16MDIV select CK_IRC16M 100: CK_IRC16MDIV select CK_IRC16M divided by 2 101: CK_IRC16MDIV select CK_IRC16M divided by 4 110: CK_IRC16MDIV select CK_IRC16M divided by 8 111: CK_IRC16MDIV select CK_IRC16M divided by 16
17:16	USART1SEL[1:0]	CK_USART1 clock source selection This bit is set and reset by software. 00: CK_USART1 select CK_APB1 01: CK_USART1 select CK_SYS 10: CK_USART1 select CK_LXTAL 11: CK_USART1 select CK_IRC16MDIV
15:14	Reserved	Must be kept at reset value.
13	USBDSEL	CK_USBD clock source selection This bit is set and reset by software. 0: CK_USBD select CK_IRC48M 1: CK_ USBD select CK_PLL
12:11	LPUARTSEL[1:0]	LPUART clock source selection This bit is set and reset by software. 00: CK_LPUART select CK_APB1 01: CK_LPUART select CK_SYS 10: CK_LPUART select CK_LXTAL 11: CK_LPUART select CK_IRC16MDIV
10:9	LPTIMERSEL[1:0]	CK_LPTIMER clock source selection This bit is set and reset by software. 00: CK_LPTIMER select CK_APB1 01: CK_LPTIMER select CK_IRC32K 10: CK_LPTIMER select CK_LXTAL 11: CK_LPTIMER select CK_IRC16MDIV
8	ADCSEL	CK_ADC clock source selection This bit is set and reset by software. 0: CK_ADC select CK_IRC16M 1: CK_ADC select CK_APB2 which is divided by 2, 4, 6, 8,10,12,14,16 or by the clock of AHB divided by 3, 5, 7, 9, 11, 13, 15, 17
7:6	I2C2SEL[1:0]	CK_I2C2 clock source selection 00: CK_I2C2 select CK_APB1

		01: CK_I2C2 select CK_SYS 10 / 11: CK_I2C2 select CK_IRC16MDIV
5:4	I2C1SEL[1:0]	CK_I2C1 clock source selection 00: CK_I2C1 select CK_APB1 01: CK_I2C1 select CK_SYS 10 / 11: CK_I2C1 select CK_IRC16MDIV
3:2	I2C0SEL[1:0]	CK_I2C0 clock source selection 00: CK_I2C0 select CK_APB1 01: CK_I2C0 select CK_SYS 10 / 11: CK_I2C0 select CK_IRC16MDIV
1:0	USART0SEL[1:0]	CK_USART0 clock source selection This bit is set and reset by software. 00: CK_USART0 select CK_APB2 01: CK_USART0 select CK_SYS 10: CK_USART0 select CK_LXTAL 11: CK_USART0 select CK_IRC16MDIV

For GD32L235xx devices

Address offset: 0x30

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
ADCPSC[3:2]		Reserved				LPUART1SEL[1:0]		Reserved	LPTIMER1SEL[1:0]		IRC16MDIVSEL			USART1SEL[1:0]	
rw						rw			rw		rw			rw	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved		USBDSEL	LPUART0SEL[1:0]		LPTIMER0SEL[1:0]		ADCSEL	I2C2SEL[1:0]		I2C1SEL[1:0]		I2C0SEL[1:0]		USART0SEL[1:0]	
		rw	rw		rw		rw	rw		rw		rw		rw	

Bits	Fields	Descriptions
31:30	ADCPSC[3:2]	Bit 3 and bit 2 of ADCPSC see bits 15:14 of RCU_CFG0
29:26	Reserved	Must be kept at reset value.
25:24	LPUART1SEL[1:0]	LPUART1 clock source selection This bit is set and reset by software. 00: CK_LPUART1 select CK_APB1 01: CK_LPUART1 select CK_SYS 10: CK_LPUART1 select CK_LXTAL 11: CK_LPUART1 select CK_IRC16MDIV

23	Reserved	Must be kept at reset value.
22:21	LPTIMER1SEL[1:0]	CK_LPTIMER1 clock source selection This bit is set and reset by software. 00: CK_LPTIMER1 select CK_APB1 01: CK_LPTIMER1 select CK_IRC32K 10: CK_LPTIMER1 select CK_LXTAL 11: CK_LPTIMER1 select CK_IRC16MDIV
20:18	IRC16MDIVSEL	CK_IRC16M divided clock selection 0xx: CK_IRC16MDIV select CK_IRC16M 100: CK_IRC16MDIV select CK_IRC16M divided by 2 101: CK_IRC16MDIV select CK_IRC16M divided by 4 110: CK_IRC16MDIV select CK_IRC16M divided by 8 111: CK_IRC16MDIV select CK_IRC16M divided by 16
17:16	USART1SEL[1:0]	CK_USART1 clock source selection This bit is set and reset by software. 00: CK_USART1 select CK_APB1 01: CK_USART1 select CK_SYS 10: CK_USART1 select CK_LXTAL 11: CK_USART1 select CK_IRC16MDIV
15:14	Reserved	Must be kept at reset value.
13	USBDSEL	CK_USBD clock source selection This bit is set and reset by software. 0: CK_USBD select CK_IRC48M 1: CK_ USBD select CK_PLL
12:11	LPUART0SEL[1:0]	LPUART0 clock source selection This bit is set and reset by software. 00: CK_LPUART0 select CK_APB1 01: CK_LPUART0 select CK_SYS 10: CK_LPUART0 select CK_LXTAL 11: CK_LPUART0 select CK_IRC16MDIV
10:9	LPTIMER0SEL[1:0]	CK_LPTIMER0 clock source selection This bit is set and reset by software. 00: CK_LPTIMER0 select CK_APB1 01: CK_LPTIMER0 select CK_IRC32K 10: CK_LPTIMER0 select CK_LXTAL 11: CK_LPTIMER select CK_IRC16MDIV
8	ADCSEL	CK_ADC clock source selection This bit is set and reset by software. 0: CK_ADC select CK_IRC16M 1: CK_ADC select CK_APB2 which is divided by 2, 4, 6, 8,10,12,14,16 or by the

		clock of AHB divided by 3, 5, 7, 9, 11, 13, 15, 17
7:6	I2C2SEL[1:0]	CK_I2C2 clock source selection 00: CK_I2C2 select CK_APB1 01: CK_I2C2 select CK_SYS 10/11: CK_I2C2 select CK_IRC16MDIV
5:4	I2C1SEL[1:0]	CK_I2C1 clock source selection 00: CK_I2C1 select CK_APB1 01: CK_I2C1 select CK_SYS 10/11: CK_I2C1 select CK_IRC16MDIV
3:2	I2C0SEL[1:0]	CK_I2C0 clock source selection 00: CK_I2C0 select CK_APB1 01: CK_I2C0 select CK_SYS 10/11: CK_I2C0 select CK_IRC16MDIV
1:0	USART0SEL[1:0]	CK_USART0 clock source selection This bit is set and reset by software. 00: CK_USART0 select CK_APB2 01: CK_USART0 select CK_SYS 10: CK_USART0 select CK_LXTAL 11: CK_USART0 select CK_IRC16MDIV

4.3.14. 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															

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

Bits	Fields	Descriptions
31:4	Reserved	Must be kept at reset value.
3	TRNGEN	TRNG clock enable This bit is set and reset by software. 0: Disabled TRNG clock 1: Enabled TRNG clock
2	Reserved	Must be kept at reset value.

1	CAUEN	CAU clock enable This bit is set and reset by software. 0: Disabled CAU clock 1: Enabled CAU clock
0	Reserved	Must be kept at reset value.

4.3.15. AHB2 reset register (RCU_AHB2RST)

Address offset: 0x38

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
												TRNGRST	Reserved	CAURST	Reserved
												rw		rw	

Bits	Fields	Descriptions
31:4	Reserved	Must be kept at reset value.
3	TRNGST	TRNG reset This bit is set and reset by software. 0: No reset TRNG module 1: Reset TRNG module
2	Reserved	Must be kept at reset value.
1	CAURST	CAU reset This bit is set and reset by software. 0: No reset CAU module 1: Reset CAU module
0	Reserved	Must be kept at reset value.

4.3.16. Voltage key register (RCU_VKEY)

Address offset: 0x100

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
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]	**For GD32L233xx devices** The key of RCU_LPB register These bits are written only by software and read as 0. Only after write 0x1A2B3C4D to the RCU_VKEY, the RCU_LPB registers can be written. **For GD32L235xx devices** The key of RCU_LPB register These bits are written only by software and read as 0. Only after write 0x00007432 to the RCU_VKEY, the RCU_LPB registers can be written.

4.3.17. Low power bandgap mode register (RCU_LPB)

For GD32L233xx devices

Offset: 0x12C

Reset value: 0x0000 0007

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
Reserved													LPBMSEL[2:0]		
													rw		

Bits	Fields	Descriptions
31:3	Reserved	Must be kept at reset value
2:0	LPBMSEL[2:0]	Low power bandgap mode selection signal. This field can only be written only when right password is written to RCU_VKEY. The length of holding phase of sample and hold circuit is controlled. 011: The length of holding phase is 3.2ms, 32 clock cycles 010: The length of holding phase is 6.4ms, 64 clock cycles 001: The length of holding phase is 12.8ms, 128 clock cycles 000: The length of holding phase is 25.6ms, 256 clock cycles 111: The length of holding phase is 51.2ms, 512 clock cycles 110: The length of holding phase is 102.4ms, 1024 clock cycles 101: The length of holding phase is 204.8ms, 2048 clock cycles 100: The length of holding phase is 204.8ms, 2048 clock cycles

For GD32L235xx devices

Offset: 0x12C

Reset value: 0x0000 000F

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
Reserved												LPBMSEL[3:0]			
												rw			

Bits	Fields	Descriptions
31:4	Reserved	Must be kept at reset value
3:0	LPBMSEL[3:0]	Low power bandgap mode selection signal. This field can only be written only when right password is written to RCU_VKEY The length of holding phase of sample and hold circuit is controlled. 1011: The length of holding phase is 3.2ms, 32 clock cycles 1010: The length of holding phase is 6.4ms, 64 clock cycles 1001: The length of holding phase is 12.8ms, 128 clock cycles 1000: The length of holding phase is 25.6ms, 256 clock cycles 1111: The length of holding phase is 51.2ms, 512 clock cycles 1110: The length of holding phase is 102.4ms, 1024 clock cycles 1101: The length of holding phase is 204.8ms, 2048 clock cycles 1100: The length of holding phase is 307.2ms, 3072 clock cycles 0011: The length of holding phase is 409.6ms, 4096 clock cycles 0010: The length of holding phase is 512ms, 5120 clock cycles 0001: The length of holding phase is 614.4ms, 6144 clock cycles 0000: The length of holding phase is 716.8ms, 7168 clock cycles 0111: The length of holding phase is 819.2ms, 8192 clock cycles 0110: The length of holding phase is 1024ms, 10240 clock cycles 0101: The length of holding phase is 1228.8ms, 12288 clock cycles 0100: The length of holding phase is 1638.4ms, 16384 clock cycles

5. Clock trim controller (CTC)

5.1. Overview

The Clock Trim Controller (CTC) is used to trim internal 48MHz RC oscillator (IRC48M) automatically by hardware. The CTC unit trim the frequency of the IRC48M based on an external accurate reference signal source. It can automatically adjust the trim value to provide a precise IRC48M clock.

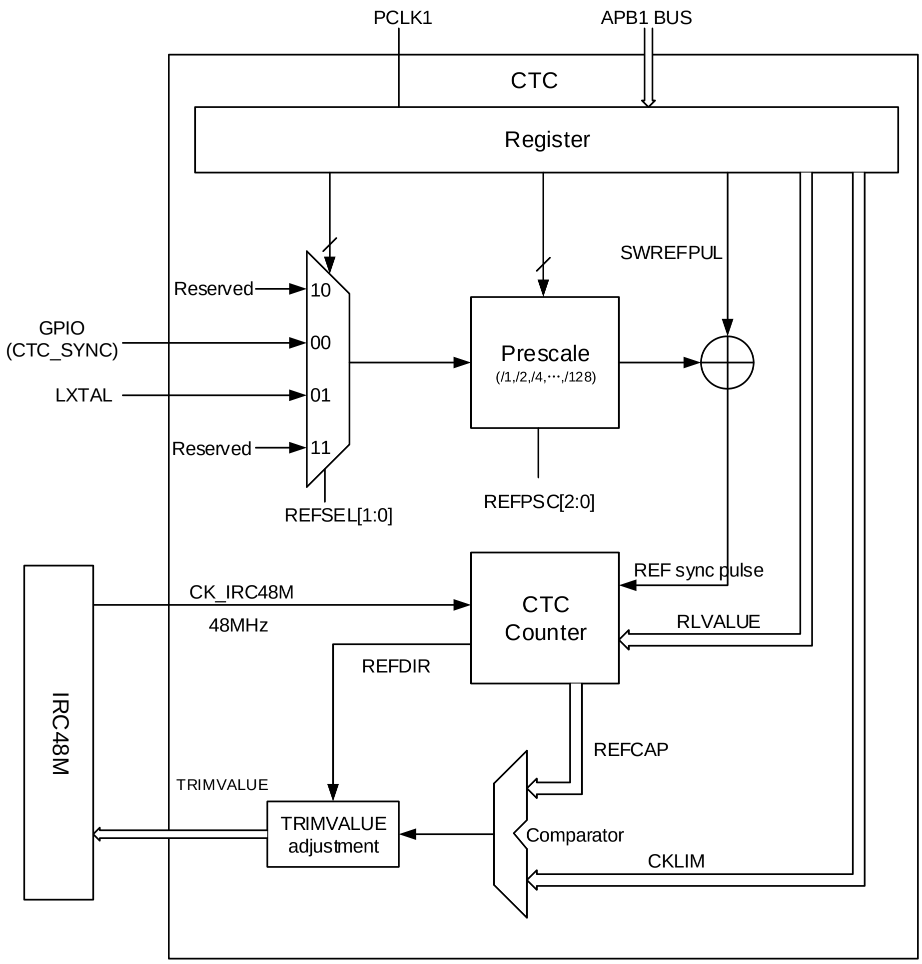
5.2. Characteristics

- Two external reference signal source: GPIO(CTC_SYNC), LXTAL clock.
- Provide software reference sync pulse.
- Automatically trimmed by hardware without any software action.
- 16 bits trim counter with reference signal source capture and reload.
- 8 bits clock trim base value to frequency evaluation and automatically trim.
- Enough flag or interrupt to indicate the clock is OK (CKOKIF), warning (CKWARNIF) or error (ERRIF).

5.3. Function overview

Figure 5-1. CTC overview provides details on the internal configuration of the CTC.

Figure 5-1. CTC overview



5.3.1. REF sync pulse generator

Firstly, the reference signal source can select GPIO(CTC_SYNC) or LXTAL clock output by setting REFSEL bits in CTC_CTL1 register.

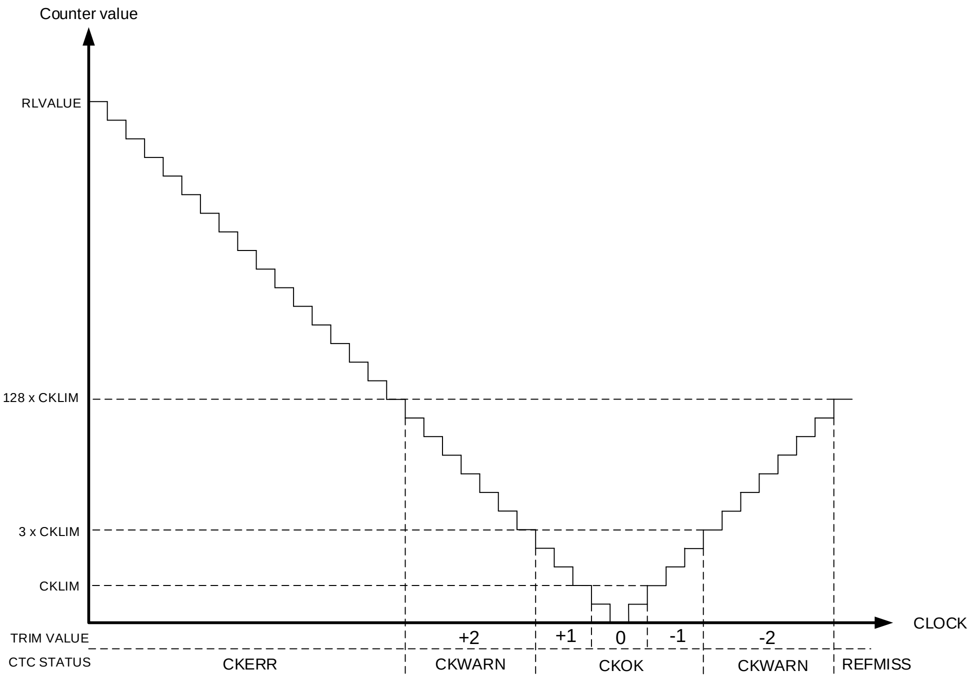
Secondly, the selected reference signal source use a configurable polarity by setting REFPOL bit in CTC_CTL1 register, and can be divided to a suitable frequency with a configurable prescaler by setting REFPSC bits in CTC_CTL1 register.

Thirdly, if a software reference pulse needed, write 1 to SWREFPUL bit in CTC_CTL0 register. The software reference pulse generated in last step is logical OR with the external reference pulse.

5.3.2. CTC trim counter

The CTC trim counter is clocked by CK_IRC48M. After CNTEN bit in CTC_CTL0 register set, and a first REF sync pulse detected, the counter start down-counting from RLVALUE (defined in CTC_CTL1 register). If any REF sync pulse detected, the counter reload the RLVALUE and start down-counting again. If no REF sync pulse detected, the counter down-count to zero, and then up-count to 128 x CKLIM (defined in CTC_CTL1 register), and then stop until next REF sync pulse detected. If any REF sync pulse detected, the current CTC trim counter value is captured to REFCAP in status register (CTC_STAT), and the counter direction is captured to REFDIR in status register (CTC_STAT). The detail is showing in ***Figure 5-2. CTC trim counter***.

Figure 5-2. CTC trim counter



5.3.3. Frequency evaluation and automatically trim process

The clock frequency evaluation is performed when a REF sync pulse occur. If a REF sync pulse occurs on down-counting, it means the current clock is slower than correct clock (the frequency of 48M).It needs to improve TRIMVALUE in CTC_CTL0 register. If a REF sync pulse occurs on up-counting, it means the current clock is faster than correct clock (the frequency of 48M).It needs to reduce TRIMVALUE in CTC_CTL0 register. The CKOKIF, CKWARNIF, CKERR and REFMISS in CTC_STAT register shows the frequency evaluation scope.

If the AUTOTRIM bit in CTC_CTL0 register is setting, the automatically hardware trim mode enabled. In this mode, if a REF sync pulse occurs on down-counting, it means the current clock is slower than correct clock, the TRIMVALUE will be increased automatically to raise the clock frequency. Vice versa when it occurs on up-counting, the TRIMVALUE will be reduced automatically to reduce the clock frequency.

- Counter < CKLIM when REF sync pulse is detected.

  The CKOKIF in CTC_STAT register set, and an interrupt generated if CKOKIE bit in CTC_CTL0 register is 1.

If the AUTOTRIM bit in CTC_CTL0 register set, the TRIMVALUE in CTC_CTL0 register is not changed.

- CKLIM ≤ Counter < 3 x CKLIM when REF sync pulse is detected.

  The CKOKIF in CTC_STAT register set, and an interrupt generated if CKOKIE bit in CTC_CTL0 register is 1.

  If the AUTOTRIM bit in CTC_CTL0 register set, the TRIMVALUE in CTC_CTL0 register add 1 when down-counting or sub 1 when up-counting.

- 3 x CKLIM ≤ Counter < 128 x CKLIM when REF sync pulse is detected.

  The CKWARNIF in CTC_STAT register set, and an interrupt generated if CKWARNIE bit in CTC_CTL0 register is 1.

  If the AUTOTRIM bit in CTC_CTL0 register set, the TRIMVALUE in CTC_CTL0 register add 2 when down-counting or sub 2 when up-counting.

- Counter ≥ 128 x CKLIM when down-counting when a REF sync pulse is detected.

  The CKERR in CTC_STAT register set, and an interrupt generated if ERRIE bit in CTC_CTL0 register is 1.

  The TRIMVALUE in CTC_CTL0 register is not changed

- Counter = 128 x CKLIM when up-counting.

  The REFMISS in CTC_STAT register set, and an interrupt generated if ERRIE bit in CTC_CTL0 register is 1.

  The TRIMVALUE in CTC_CTL0 register is not changed.

If adjusting the TRIMVALUE in CTC_CTL0 register over the value of 127, the overflow will be occurred, while adjusting the TRIMVALUE under the value of 0, the underflow will be occurred. The TRIMVALUE is in the range 0 to 127 (the TRIMVALUE is 127 if overflow, the TRIMVALUE is 0 if underflow). Then, the TRIMERR in CTC_STAT register will be set, and an interrupt generated if ERRIE bit in CTC_CTL0 register is 1.

5.3.4. Software program guide

The RLVALUE and CKLIM bits in CTC_CTL1 register is critical to evaluate the clock frequency and automatically hardware trim. The value is calculated by the correct clock frequency (IRC48M:48 MHz) and the frequency of REF sync pulse. The ideal case is REF sync pulse occur when the CTC counter is zero, so the RLVALUE is:

$$RLVALUE = (F_{clock} \div F_{REF}) - 1 \qquad (5\text{-}1)$$

The CKLIM is set by user according to the clock accuracy. It is recommend to set to the half of the step size, so the CKLIM is:

$$CKLIM = (F_{clock} \div F_{REF}) \times 0.12\% \div 2 \qquad (5\text{-}2)$$

The typical step size is 0.12%. Where the F_{clock} is the frequency of correct clock (IRC48M), the F_{REF} is the frequency of reference sync pulse.

5.4. Register definition

CTC base address: 0x4000 C800

5.4.1. Control register 0 (CTC_CTL0)

Address offset: 0x00

Reset value: 0x0000 4000

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	TRIMVALUE[6:0]							SWREFPUL	AUTOTRIM	CNTEN	Reserved	EREFIE	ERRIE	CKWARNIE	CKOKIE
	rw							w	rw	rw		rw	rw	rw	rw

Bits	Fields	Descriptions
31:15	Reserved	Must be kept at reset value.
14:8	TRIMVALUE[6:0]	IRC48M trim value When AUTOTRIM in CTC_CTL0 register is 0, these bits are set and cleared by software. This mode used to software calibration. When AUTOTRIM in CTC_CTL0 register is 1, these bits are read only. The value automatically modified by hardware. This mode used to hardware trim. The middle value is 64. When increase 1, the IRC48M clock frequency add around 57KHz. When decrease 1, the IRC48M clock frequency sub around 57KHz.
7	SWREFPUL	Software reference source sync pulse This bit is set by software, and generates a reference sync pulse to CTC counter. This bit is cleared by hardware automatically and read as 0. 0: No effect 1: generates a software reference source sync pulse
6	AUTOTRIM	Hardware automatically trim mode This bit is set and cleared by software. When this bit is set, the hardware automatic trim enabled, the TRIMVALUE bits in CTC_CTL0 register are modified by hardware automatically, until the frequency of IRC48M clock is close to 48MHz. 0: Hardware automatic trim disabled 1: Hardware automatic trim enabled
5	CNTEN	CTC counter enable This bit is set and cleared by software. This bit used to enable or disable the CTC trim counter. When this bit is set, the CTC_CTL1 register cannot be modified. 0: CTC trim counter disabled

		1: CTC trim counter enabled.
4	Reserved	Must be kept at reset value.
3	EREFIE	EREFIF interrupt enable 0: EREFIF interrupt disable 1: EREFIF interrupt enable
2	ERRIE	Error (ERRIF) interrupt enable 0: ERRIF interrupt disable 1: ERRIF interrupt enable
1	CKWARNIE	Clock trim warning (CKWARNIF) interrupt enable 0: CKWARNIF interrupt disable 1: CKWARNIF interrupt enable
0	CKOKIE	Clock trim OK (CKOKIF) interrupt enable 0: CKOKIF interrupt disable 1: CKOKIF interrupt enable

5.4.2. Control register 1 (CTC_CTL1)

Address offset: 0x04

Reset value: 0x2022 BB7F

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

Note: This register cannot be modified when CNTEN is 1.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
REF POL	Reserved	REFSEL[1:0]		Reserved	REFPSC[2:0]			CKLIM[7:0]							
rw		rw			rw			rw							

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

Bits	Fields	Descriptions
31	REFPOL	Reference signal source polarity This bit is set and cleared by software to select reference signal source polarity 0: rising edge selected 1: falling edge selected
30	Reserved	Must be kept at reset value.
29:28	REFSEL[1:0]	Reference signal source selection These bits are set and cleared by software to select reference signal source. 00: GPIO(CTC_SYNC) selected 01: LXTAL clock selected

		10: Reserved. 11: Reserved.
27	Reserved	Must be kept at reset value.
26:24	REFPSC[2:0]	Reference signal source prescaler These bits are set and cleared by software 000: Reference signal not divided 001: Reference signal divided by 2 010: Reference signal divided by 4 011: Reference signal divided by 8 100: Reference signal divided by 16 101: Reference signal divided by 32 110: Reference signal divided by 64 111: Reference signal divided by 128
23:16	CKLIM[7:0]	Clock trim base limit value These bits are set and cleared by software to define the clock trim base limit value. These bits used to frequency evaluation and automatically trim process. Please refer to the ***Frequency evaluation and automatically trim process*** for detail.
15:0	RLVALUE[15:0]	CTC counter reload value These bits are set and cleared by software to define the CTC counter reload value. These bits reload to CTC trim counter when a reference sync pulse received to start or restart the counter.

5.4.3. Status register (CTC_STAT)

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
REFCAP[15:0]															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REFDIR	Reserved				TRIMERR	REFMISS	CKERR	Reserved				EREFIF	ERRIF	CKWARNIF	CKOKIF
r					r	r	r					r	r	r	r

Bits	Fields	Descriptions
31:16	REFCAP[15:0]	CTC counter capture when reference sync pulse. When a reference sync pulse occurred, the CTC trim counter value is captured to REFCAP bits.
15	REFDIR	CTC trim counter direction when reference sync pulse

		When a reference sync pulse occurred during the counter is working, the CTC trim counter direction is captured to REFDIR bit. 0: Up-counting 1: Down-counting
14:11	Reserved	Must be kept at reset value.
10	TRIMERR	Trim value error bit This bit is set by hardware when the TRIMVALUE in CTC_CTL0 register overflow or underflow. When the ERRIE in CTC_CTL0 register is set, an interrupt occur. This bit is cleared by writing 1 to ERRIC bit in CTC_INTC register. 0: No trim value error occur 1: Trim value error occur
9	REFMISS	Reference sync pulse miss This bit is set by hardware when the reference sync pulse miss. This is occur when the CTC trim counter reach to 128 x CKLIM during up counting and no reference sync pulse detected. This means the clock is too fast to be trimmed to correct frequency or other error occur. When the ERRIE in CTC_CTL0 register is set, an interrupt occur. This bit is cleared by writing 1 to ERRIC bit in CTC_INTC register. 0: No Reference sync pulse miss occur 1: Reference sync pulse miss occur
8	CKERR	Clock trim error bit This bit is set by hardware when the clock trim error occur. This is occur when the CTC trim counter greater or equal to 128 x CKLIM during down counting when a reference sync pulse detected. This means the clock is too slow and cannot be trimmed to correct frequency. When the ERRIE in CTC_CTL0 register is set, an interrupt occur. This bit is cleared by writing 1 to ERRIC bit in CTC_INTC register. 0: No Clock trim error occur 1: Clock trim error occur
7:4	Reserved	Must be kept at reset value.
3	EREFIF	Expect reference interrupt flag This bit is set by hardware when the CTC counter reach to 0. When the EREFIE in CTC_CTL0 register is set, an interrupt occur. This bit is cleared by writing 1 to EREFIC bit in CTC_INTC register. 0 : No Expect reference occur 1: Expect reference occur
2	ERRIF	Error interrupt flag This bit is set by hardware when an error occurred. If any error of TRIMERR, REFMISS or CKERR occurred, this bit will be set. When the ERRIE in CTC_CTL0 register is set, an interrupt occur. This bit is cleared by writing 1 to ERRIC bit in CTC_INTC register. 0 : No Error occur

		1: An error occur
1	CKWARNIF	Clock trim warning interrupt flag This bit is set by hardware when a clock trim warning occurred. If the CTC trim counter greater or equal to 3 x CKLIM and smaller to 128 x CKLIM when a reference sync pulse detected, this bit will be set. This means the clock is too slow or too fast, but can be trim to correct frequency. The TRIMVALUE add 2 or sub 2 when a clock trim warning occurred. When the CKWARNIE in CTC_CTL0 register is set, an interrupt occur. This bit is cleared by writing 1 to CKWARNIC bit in CTC_INTC register. 0 : No Clock trim warning occur 1: Clock trim warning occur
0	CKOKIF	Clock trim OK interrupt flag This bit is set by hardware when the clock trim is OK. If the CTC trim counter smaller to 3 x CKLIM when a reference sync pulse detected, this bit will be set. This means the clock is OK to use. The TRIMVALUE need not to adjust or adjust one step. When the CKOKIE in CTC_CTL0 register is, an interrupt occur. This bit is cleared by writing 1 to CKOKIC bit in CTC_INTC register. 0 : No Clock trim OK occur 1: Clock trim OK occur

5.4.4. Interrupt clear register (CTC_INTC)

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												EREFIC	ERRIC	CKWARNIC	CKOKIC
												w	w	w	w

Bits	Fields	Descriptions
31:4	Reserved	Must be kept at reset value.
3	EREFIC	EREFIF interrupt clear bit This bit is written by software and read as 0. Write 1 to clear EREFIF bit in CTC_STAT register. Write 0 is no effect.
2	ERRIC	ERRIF interrupt clear bit This bit is written by software and read as 0. Write 1 to clear ERRIF, TRIMERR,

		REFMISS and CKERR bits in CTC_STAT register. Write 0 is no effect.
1	CKWARNIC	CKWARNIF interrupt clear bit This bit is written by software and read as 0. Write 1 to clear CKWARNIF bit in CTC_STAT register. Write 0 is no effect.
0	CKOKIC	CKOKIF interrupt clear bit This bit is written by software and read as 0. Write 1 to clear CKOKIF bit in CTC_STAT register. Write 0 is no effect.

6. Interrupt / event controller (EXTI)

6.1. 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 30 independent edge detectors (for GD32L233xx devices) or 32 independent edge detectors (for GD32L235xx devices) 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.

6.2. Characteristics

- Cortex®-M23 system exception.
- Up to 69 maskable peripheral interrupts (for GD32L233xx devices) or 72 maskable peripheral interrupts (for GD32L235xx devices).
- 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 30 independent edge detectors (for GD32L233xx devices) or 32 independent edge detectors (for GD32L235xx devices) in EXTI.
- Three trigger types: rising, falling and both edges.
- Software interrupt or event trigger.
- Trigger sources configurable.

6.3. 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 6-1. NVIC exception types in Cortex®-M23***, ***Table 6-2. Interrupt vector table for GD32L233xx devices*** and ***Table 6-3. Interrupt vector table for GD32L235xx devices*** list all exception types.

Table 6-1. 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 systemservice
SysTick	15	Programmable	0x0000_003C	System tick timer

Table 6-2. Interrupt vector table for GD32L233xx devices

Interrupt number	Vector number	Peripheral interrupt description	Vector address
IRQ 0	16	WWDGT interrupt	0x0000_0040
IRQ 1	17	LVD from EXTI interrupt	0x0000_0044
IRQ 2	18	RTC Tamper and Timestamp from EXTI interrupts	0x0000_0048
IRQ 3	19	RTC wakeup from EXTI interrupt	0x0000_004C
IRQ 4	20	FMC global interrupt	0x0000_0050
IRQ 5	21	RCU or CTC global interrupt	0x0000_0054
IRQ 6	22	EXTI Line0 interrupt	0x0000_0058
IRQ 7	23	EXTI Line1 interrupt	0x0000_005C
IRQ 8	24	EXTI Line2 interrupt	0x0000_0060
IRQ 9	25	EXTI Line3 interrupt	0x0000_0064
IRQ 10	26	EXTI Line4 interrupt	0x0000_0068
IRQ 11	27	DMA Channel0 global interrupt	0x0000_006C
IRQ 12	28	DMA Channel1 global interrupt	0x0000_0070
IRQ 13	29	DMA Channel2 global interrupt	0x0000_0074
IRQ 14	30	DMA Channel3 global interrupt	0x0000_0078
IRQ 15	31	DMA Channel4 global interrupt	0x0000_007C
IRQ 16	32	DMA Channel5 global interrupt	0x0000_0080
IRQ 17	33	DMA Channel6 global interrupt	0x0000_0084
IRQ 18	34	ADC interrupt	0x0000_0088
IRQ 19	35	USBD High Priority Interrupt	0x0000_008C
IRQ 20	36	USBD Low Priority Interrupt	0x0000_0090
IRQ 21	37	TIMER1 global interrupt	0x0000_0094

Interrupt number	Vector number	Peripheral interrupt description	Vector address
IRQ 22	38	TIMER2 global interrupt	0x0000_0098
IRQ 23	39	TIMER8 global interrupt	0x0000_009C
IRQ 24	40	TIMER11 global interrupt	0x0000_00A0
IRQ 25	41	TIMER5 global interrupt	0x0000_00A4
IRQ 26	42	TIMER6 global interrupt	0x0000_00A8
IRQ 27	43	USART0 global interrupt	0x0000_00AC
IRQ 28	44	USART1 global interrupt	0x0000_00B0
IRQ 29	45	UART3 global interrupt	0x0000_00B4
IRQ 30	46	UART4 global interrupt	0x0000_00B8
IRQ 31	47	I2C0 event interrupt	0x0000_00BC
IRQ 32	48	I2C0 error interrupt	0x0000_00C0
IRQ 33	49	I2C1 event interrupt	0x0000_00C4
IRQ 34	50	I2C1 error interrupt	0x0000_00C8
IRQ 35	51	SPI0 global interrupt	0x0000_00CC
IRQ 36	52	SPI1 global interrupt	0x0000_00D0
IRQ 37	53	DAC interrupt	0x0000_00D4
IRQ 38	54	Reserved	0x0000_00D8
IRQ 39	55	I2C2 event interrupt	0x0000_00DC
IRQ 40	56	I2C2 error interrupt	0x0000_00E0
IRQ 41	57	RTC alarm from EXTI interrupt	0x0000_00E4
IRQ 42	58	USBD wakeup from EXTI interrupt	0x0000_00E8
IRQ 43	59	EXTI line[9:5] interrupts	0x0000_00EC
IRQ 44-46	60-62	Reserved	0x0000_00F0- 0x0000_00F8
IRQ 47	63	EXTI line[15:10] interrupts	0x0000_00FC
IRQ 48-54	64-70	Reserved	0x0000_0100- 0x0000_0118
IRQ 55	71	DMA MUX interrupt	0x0000_011C
IRQ 56	72	CMP0 output from EXTI interrupt	0x0000_0120
IRQ 57	73	CMP1 output from EXTI interrupt	0x0000_0124
IRQ 58	74	I2C0 wakeup from EXTI interrupt	0x0000_0128
IRQ 59	75	I2C2 wakeup from EXTI interrupt	0x0000_012C
IRQ 60	76	USART0 wakeup from EXTI interrupt	0x0000_0130
IRQ 61	77	LPUART global interrupt	0x0000_0134
IRQ 62	78	CAU global interrupt	0x0000_0138
IRQ 63	79	TRNG global Interrupt	0x0000_013C
IRQ 64	80	SLCD global interrupt	0x0000_0140
IRQ 65	81	USART1 wakeup from EXTI interrupt	0x0000_0144
IRQ 66	82	I2C1 wakeup from EXTI interrupt	0x0000_0148
IRQ 67	83	LPUART wakeup from EXTI interrupt	0x0000_014C

Interrupt number	Vector number	Peripheral interrupt description	Vector address
IRQ 68	84	LPTIMER global interrupt	0x0000_0150

Table 6-3. Interrupt vector table for GD32L235xx devices

Interrupt number	Vector number	Peripheral interrupt description	Vector Address
IRQ 0	16	WWDGT interrupt	0x0000_0040
IRQ 1	17	LVD from EXTI interrupt	0x0000_0044
IRQ 2	18	RTC Tamper and Timestamp from EXTI interrupts	0x0000_0048
IRQ 3	19	RTC wakeup from EXTI interrupt	0x0000_004C
IRQ 4	20	FMC global interrupt	0x0000_0050
IRQ 5	21	RCU or CTC global interrupt	0x0000_0054
IRQ 6	22	EXTI Line0 interrupt	0x0000_0058
IRQ 7	23	EXTI Line1 interrupt	0x0000_005C
IRQ 8	24	EXTI Line2 interrupt	0x0000_0060
IRQ 9	25	EXTI Line3 interrupt	0x0000_0064
IRQ 10	26	EXTI Line4 interrupt	0x0000_0068
IRQ 11	27	DMA Channel0 global interrupt	0x0000_006C
IRQ 12	28	DMA Channel1 global interrupt	0x0000_0070
IRQ 13	29	DMA Channel2 global interrupt	0x0000_0074
IRQ 14	30	DMA Channel3 global interrupt	0x0000_0078
IRQ 15	31	DMA Channel4 global interrupt	0x0000_007C
IRQ 16	32	DMA Channel5 global interrupt	0x0000_0080
IRQ 17	33	DMA Channel6 global interrupt	0x0000_0084
IRQ 18	34	ADC interrupt	0x0000_0088
IRQ 19	35	USBD High Priority or CAN TX Interrupt	0x0000_008C
IRQ 20	36	USBD Low Priority or CAN RX0 Interrupt	0x0000_0090
IRQ 21	37	TIMER1 global interrupt	0x0000_0094
IRQ 22	38	TIMER2 global interrupt	0x0000_0098
IRQ 23	39	TIMER8 global interrupt	0x0000_009C
IRQ 24	40	TIMER11 global interrupt	0x0000_00A0
IRQ 25	41	TIMER5 global interrupt	0x0000_00A4
IRQ 26	42	TIMER6 global interrupt	0x0000_00A8
IRQ 27	43	USART0 global interrupt	0x0000_00AC
IRQ 28	44	USART1 global interrupt	0x0000_00B0
IRQ 29	45	UART3 global interrupt	0x0000_00B4
IRQ 30	46	UART4 global interrupt	0x0000_00B8
IRQ 31	47	I2C0 event interrupt	0x0000_00BC
IRQ 32	48	I2C0 error interrupt	0x0000_00C0
IRQ 33	49	I2C1 event interrupt	0x0000_00C4
IRQ 34	50	I2C1 error interrupt	0x0000_00C8

Interrupt number	Vector number	Peripheral interrupt description	Vector Address
IRQ 35	51	SPI0 global interrupt	0x0000_00CC
IRQ 36	52	SPI1 global interrupt	0x0000_00D0
IRQ 37	53	DAC interrupt	0x0000_00D4
IRQ 38	54	Reserved	0x0000_00D8
IRQ 39	55	I2C2 event interrupt	0x0000_00DC
IRQ 40	56	I2C2 error interrupt	0x0000_00E0
IRQ 41	57	RTC alarm from EXTI interrupt	0x0000_00E4
IRQ 42	58	USBD wakeup from EXTI interrupt	0x0000_00E8
IRQ 43	59	EXTI line[9:5] interrupts	0x0000_00EC
IRQ 44	60	TIMER0 trigger and channel commutation interrupts or TIMER0 update interrupt or TIMER0 break interrupt	0x0000_00F0
IRQ 45	61	TIMER0 capture compare interrupt	0x0000_00F4
IRQ 46	62	TIMER14 global interrupt	0x0000_00F8
IRQ 47	63	EXTI line[15:10] interrupts	0x0000_00FC
IRQ 48	64	TIMER40 global interrupt	0x0000_0100
IRQ 49	65	CAN RX1 interrupt	0x0000_0104
IRQ 50	66	CAN EWMC interrupt	0x0000_0108
IRQ 51-54	67-70	Reserved	0x0000_010C–0x0000_0118
IRQ 55	71	DMA MUX interrupt	0x0000_011C
IRQ 56	72	CMP0 output from EXTI interrupt	0x0000_0120
IRQ 57	73	CMP1 output from EXTI interrupt	0x0000_0124
IRQ 58	74	I2C0 wakeup from EXTI interrupt	0x0000_0128
IRQ 59	75	I2C2 wakeup from EXTI interrupt	0x0000_012C
IRQ 60	76	USART0 Wakeup from EXTI interrupt	0x0000_0130
IRQ 61	77	LPUART0 global interrupt	0x0000_0134
IRQ 62	78	CAU global interrupt	0x0000_0138
IRQ 63	79	TRNG global Interrupt	0x0000_013C
IRQ 64	80	SLCD global interrupt	0x0000_0140
IRQ 65	81	USART1 wakeup from EXTI interrupt	0x0000_0144
IRQ 66	82	I2C1 wakeup from EXTI interrupt	0x0000_0148
IRQ 67	83	LPUART0 wakeup from EXTI interrupt	0x0000_014C
IRQ 68	84	LPTIMER0 global interrupt	0x0000_0150
IRQ 69	85	LPUART1 wakeup from EXTI interrupt	0x0000_0164
IRQ 70	86	LPTIMER1 global interrupt	0x0000_0168
IRQ 71	87	LPUART1 global interrupt	0x0000_016C

6.4. External interrupt and event block diagram

Figure 6-1. Block diagram of EXTI for GD32L233xx devices

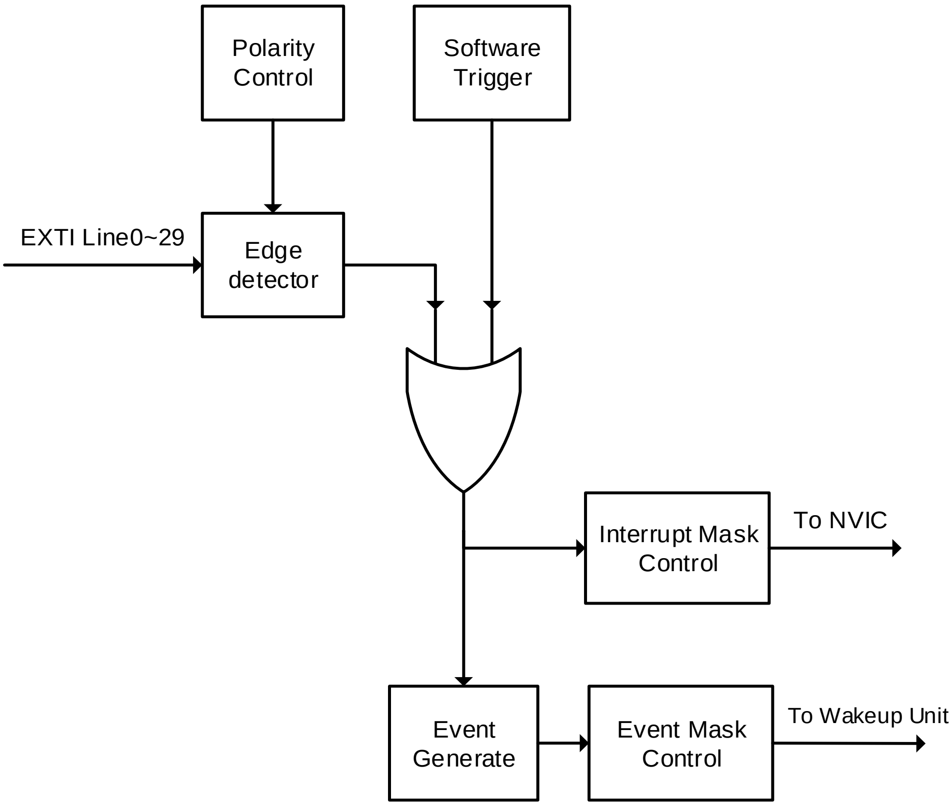
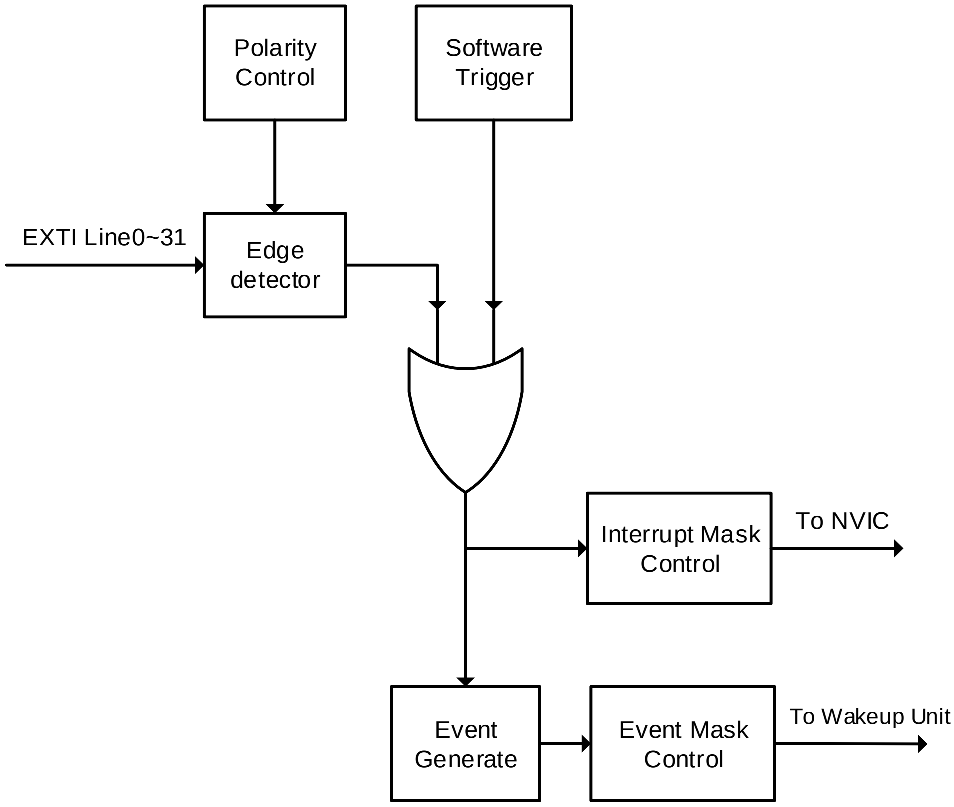


Figure 6-2. Block diagram of EXTI for GD32L235xx devices



6.5. External interrupt and Event function overview

The EXTI contains up to 30 independent edge detectors (for GD32L233xx devices) or 32 independent edge detectors (for GD32L235xx devices) 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 14 lines (for GD32L233xx devices which refer to ***Table 6-4. EXTI source for GD32L233xx devices*** for detail) or 16 lines (for GD32L235xx devices which refer to ***Table 6-5. EXTI source for GD32L235xx devices*** for detail) from internal modules. 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 6-4. EXTI source for GD32L233xx devices

EXTI line number	Source
0	PA0 / PB0 / PC0 / PD0 / PF0
1	PA1 / PB1 / PC1 / PD1 / PF1
2	PA2 / PB2 / PC2 / PD2
3	PA3 / PB3 / PC3 / PD3
4	PA4 / PB4 / PC4 / PD4
5	PA5 / PB5 / PC5 / PD5
6	PA6 / PB6 / PC6 / PD6
7	PA7 / PB7 / PC7
8	PA8 / PB8 / PC8 / PD8
9	PA9 / PB9 / PC9 / PD9
10	PA10 / PB10 / PC10
11	PA11 / PB11 / PC11
12	PA12 / PB12 / PC12
13	PA13 / PB13 / PC13
14	PA14 / PB14 / PC14
15	PA15 / PB15 / PC15
16	LVD
17	RTC Alarm

EXTI line number	Source
18	USBD wakeup
19	RTC Tamper and Timestamp
20	RTC wakeup
21	CMP0 output
22	CMP1 output
23	I2C0 wakeup
24	I2C2 wakeup
25	USART0 wakeup
26	USART1 wakeup
27	I2C1 wakeup
28	LPUART wakeup
29	LPTIMER wakeup

Table 6-5. EXTI source for GD32L235xx devices

EXTI line number	Source
0	PA0 / PB0 / PC0 / PD0 / PF0
1	PA1 / PB1 / PC1 / PD1 / PF1
2	PA2 / PB2 / PC2 / PD2
3	PA3 / PB3 / PC3 / PD3
4	PA4 / PB4 / PC4 / PD4
5	PA5 / PB5 / PC5 / PD5
6	PA6 / PB6 / PC6 / PD6
7	PA7 / PB7 / PC7
8	PA8 / PB8 / PC8 / PD8
9	PA9 / PB9 / PC9 / PD9
10	PA10 / PB10 / PC10
11	PA11 / PB11 / PC11
12	PA12 / PB12 / PC12
13	PA13 / PB13 / PC13
14	PA14 / PB14 / PC14
15	PA15 / PB15 / PC15
16	LVD
17	RTC Alarm
18	USBD wakeup
19	RTC Tamper and Timestamp
20	RTC wakeup
21	CMP0 output
22	CMP1 output
23	I2C0 wakeup
24	I2C2 wakeup

EXTI line number	Source
25	USART0 wakeup
26	USART1 wakeup
27	I2C1 wakeup
28	LPUART0 wakeup
29	LPTIMER0 wakeup
30	LPUART1 wakeup
31	LPTIMER1 wakeup

6.6. Register definition

EXTI base address: 0x4001 0400

6.6.1. Interrupt enable register (EXTI_INTEN)

For GD32L233xx devices

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		INTEN29	INTEN28	INTEN27	INTEN26	INTEN25	INTEN24	INTEN23	INTEN22	INTEN21	INTEN20	INTEN19	INTEN18	INTEN17	INTEN16
		rw	rw	rw	rw	rw	rw	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
31:30	Reserved	Must be kept at reset value.
29:0	INTENx	Interrupt enable bit x(x = 0..29) 0: Interrupt from Linex is disabled 1: Interrupt from Linex is enabled

For GD32L235xx devices

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
INTEN31	INTEN30	INTEN29	INTEN28	INTEN27	INTEN26	INTEN25	INTEN24	INTEN23	INTEN22	INTEN21	INTEN20	INTEN19	INTEN18	INTEN17	INTEN16
rw	rw	rw	rw	rw	rw	rw	rw	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
31:0	INTENx	Interrupt enable bit x(x = 0..31) 0: Interrupt from Linex is disabled 1: Interrupt from Linex is enabled

6.6.2. Event enable register (EXTI_EVEN)

For GD32L233xx devices

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		EVEN29	EVEN28	EVEN27	EVEN26	EVEN25	EVEN24	EVEN23	EVEN22	EVEN21	EVEN20	EVEN19	EVEN18	EVEN17	EVEN16
		rw	rw	rw	rw	rw	rw	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
31:30	Reserved	Must be kept at reset value.
29:0	EVENx	Event enable bit x(x = 0..29) 0: Event from Linex is disabled 1: Event from Linex is enabled

For GD32L235xx devices

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
EVEN31	EVEN30	EVEN29	EVEN28	EVEN27	EVEN26	EVEN25	EVEN24	EVEN23	EVEN22	EVEN21	EVEN20	EVEN19	EVEN18	EVEN17	EVEN16
rw	rw	rw	rw	rw	rw	rw	rw	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
31:0	EVENx	Event enable bit x(x = 0..31) 0: Event from Linex is disabled 1: Event from Linex is enabled

6.6.3. Rising edge trigger enable register (EXTI_RTEN)

For GD32L233xx devices

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		RTEN29	RTEN28	RTEN27	RTEN26	RTEN25	RTEN24	RTEN23	RTEN22	RTEN21	RTEN20	RTEN19	RTEN18	RTEN17	RTEN16
		rw	rw	rw	rw	rw	rw	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
31:30	Reserved	Must be kept at reset value.
29:0	RTENx	Rising edge trigger enable (x = 0..29) 0: Rising edge of Linex is invalid 1: Rising edge of Linex is valid as an interrupt / event request

For GD32L235xx devices

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
RTEN31	RTEN30	RTEN29	RTEN28	RTEN27	RTEN26	RTEN25	RTEN24	RTEN23	RTEN22	RTEN21	RTEN20	RTEN19	RTEN18	RTEN17	RTEN16
rw	rw	rw	rw	rw	rw	rw	rw	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
31:0	RTENx	Rising edge trigger enable (x = 0..31) 0: Rising edge of Linex is invalid 1: Rising edge of Linex is valid as an interrupt / event request

6.6.4. Falling edge trigger enable register (EXTI_FTEN)

For GD32L233xx devices

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		FTEN29	FTEN28	FTEN27	FTEN26	FTEN25	FTEN24	FTEN23	FTEN22	FTEN21	FTEN20	FTEN19	FTEN18	FTEN17	FTEN16
		rw	rw	rw	rw	rw	rw	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
31:30	Reserved	Must be kept at reset value.
29:0	FTENx	Falling edge trigger enable (x = 0..29) 0: Falling edge of Linex is invalid 1: Falling edge of Linex is valid as an interrupt / event request

For GD32L235xx devices

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
FTEN31	FTEN30	FTEN29	FTEN28	FTEN27	FTEN26	FTEN25	FTEN24	FTEN23	FTEN22	FTEN21	FTEN20	FTEN19	FTEN18	FTEN17	FTEN16
rw	rw	rw	rw	rw	rw	rw	rw	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
31:0	FTENx	Falling edge trigger enable (x = 0..31) 0: Falling edge of Linex is invalid 1: Falling edge of Linex is valid as an interrupt / event request

6.6.5. Software interrupt event register (EXTI_SWIEV)

For GD32L233xx devices

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		SWIEV29	SWIEV28	SWIEV27	SWIEV26	SWIEV25	SWIEV24	SWIEV23	SWIEV22	SWIEV21	SWIEV21	SWIEV19	SWIEV18	SWIEV17	SWIEV16
		rw	rw	rw	rw	rw	rw	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
31:30	Reserved	Must be kept at reset value.
29:0	SWIEVx	Interrupt / Event software trigger (x = 0..29) 0: Deactivate the EXTIx software interrupt / event request 1: Activate the EXTIx software interrupt / event request

For GD32L235xx devices

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
SWIEV31	SWIEV30	SWIEV29	SWIEV28	SWIEV27	SWIEV26	SWIEV25	SWIEV24	SWIEV23	SWIEV22	SWIEV21	SWIEV21	SWIEV19	SWIEV18	SWIEV17	SWIEV16
rw	rw	rw	rw	rw	rw	rw	rw	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
31:0	SWIEVx	Interrupt / Event software trigger (x = 0..31) 0: Deactivate the EXTIx software interrupt / event request 1: Activate the EXTIx software interrupt / event request

6.6.6. Pending register (EXTI_PD)

For GD32L233xx devices

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		PD29	PD28	PD27	PD26	PD25	PD24	PD23	PD22	PD21	PD21	PD19	PD19	PD17	PD16
		rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rw	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
31:30	Reserved	Must be kept at reset value.
29:0	PDx	Interrupt pending status (x = 0..29) 0: EXTI Linex is not triggered 1: EXTI Linex is triggered. This bit is cleared to 0 by writing 1 to it.

For GD32L235xx devices

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
PD31	PD30	PD29	PD28	PD27	PD26	PD25	PD24	PD23	PD22	PD21	PD21	PD19	PD19	PD17	PD16
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	rw	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
31:0	PDx	Interrupt pending status (x = 0..31) 0: EXTI Linex is not triggered 1: EXTI Linex is triggered. This bit is cleared to 0 by writing 1 to it.

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

7.1. Overview

There are up to 59 general purpose I/O pins (GPIO), named PA0 ~ PA15, PB0 ~ PB15, PC0 ~ PC15, PD0~PD6，PD8~PD9，PF0, PF1 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.

7.2. 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.

7.3. Function overview

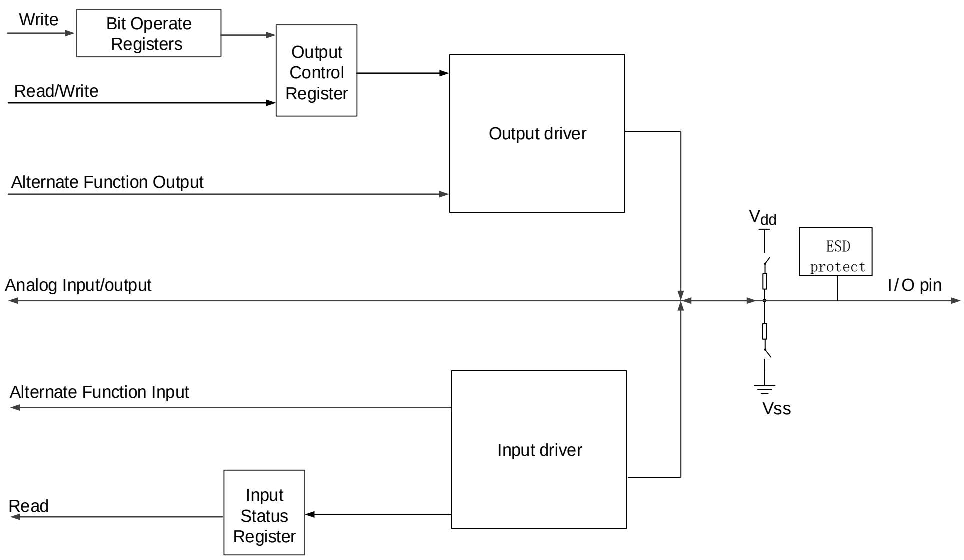
Each of the general-purpose I/O ports can be configured as GPIO inputs, GPIO outputs, AF function or analog mode by GPIO 32-bit configuration registers (GPIOx_CTL). When select AF function, the pad input or output is decided by selected AF function output enable. When the port is output (GPIO output 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 7-1. 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 7-1. Basic structure of a general-pupose I/O shows the basic structure of an I/O Port bit.

Figure 7-1. Basic structure of a general-pupose I/O



7.3.1. 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.

7.3.2. External interrupt / event lines

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

7.3.3. 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.

7.3.4. Additional functions

Some pins have additional functions, which have priority over the configuration in the standard GPIO registers. When for ADC, DAC, CMP 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.

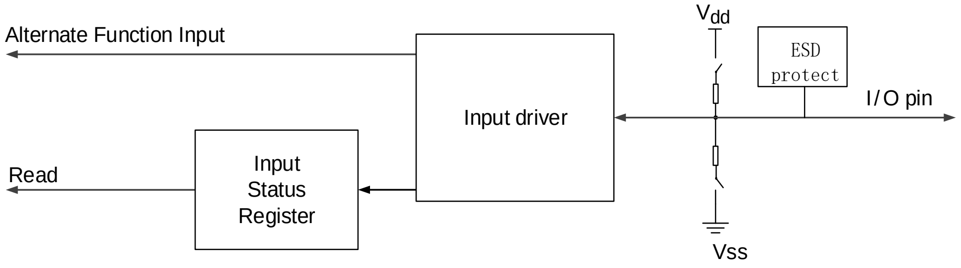
7.3.5. Input configuration

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 7-2. Basic structure of Input configuration shows the input configuration.

Figure 7-2. Basic structure of Input configuration



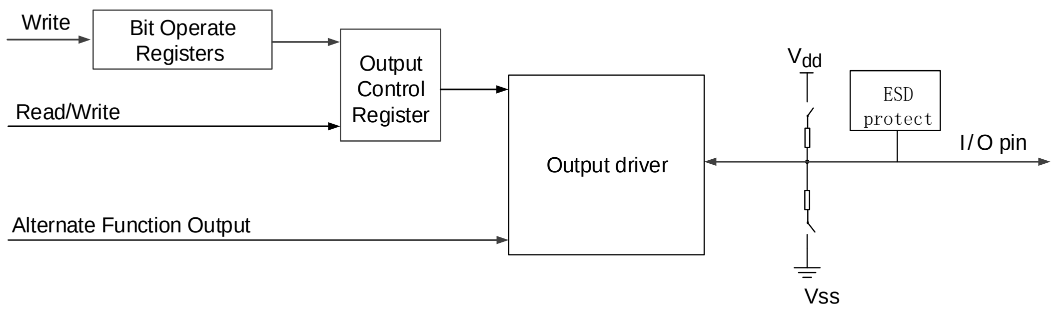
7.3.6. Output configuration

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 7-3. Basic structure of Output configuration shows the output configuration.

Figure 7-3. Basic structure of Output configuration



7.3.7. 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 7-4. Basic structure of Analog configuration shows the analog configuration.

Figure 7-4. Basic structure of Analog configuration

7.3.8. Alternate function (AF) configuration

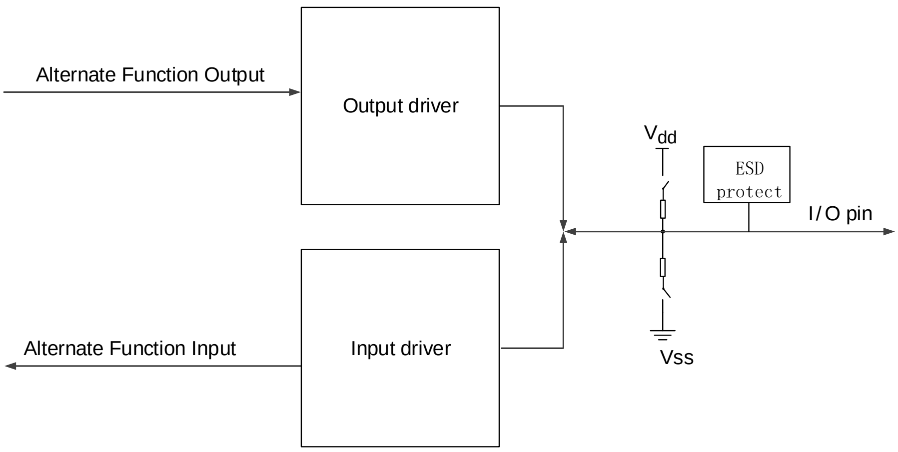
To suit for different device packages, the GPIO supports some alternate functions mapped to some other pins by software.

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 7-5. Basic structure of Alternate function configuration shows the alternate function configuration.

Figure 7-5. Basic structure of Alternate function configuration



7.3.9. 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.

7.3.10. 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.

7.4. 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

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

Address offset: 0x00

Reset value: 0x2800 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
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

7.4.2. 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

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

Address offset: 0x08

Reset value: 0x0C00 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
OSPD15[1:0]	OSPD14[1:0]	OSPD13[1:0]	OSPD12[1:0]	OSPD11[1:0]	OSPD10[1:0]	OSPD9[1:0]	OSPD8[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
OSPD7[1:0]	OSPD6[1:0]	OSPD5[1:0]	OSPD4[1:0]	OSPD3[1:0]	OSPD2[1:0]	OSPD1[1:0]	OSPD0[1:0]
rw	rw	rw	rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31:30	OSPD15[1:0]	Pin 15 output max speed bits These bits are set and cleared by software. Refer to OSPD0[1:0] description
29:28	OSPD14[1:0]	Pin 14 output max speed bits

		These bits are set and cleared by software. Refer to OSPD0[1:0] description
27:26	OSPD13[1:0]	Pin 13 output max speed bits These bits are set and cleared by software. Refer to OSPD0[1:0] description
25:24	OSPD12[1:0]	Pin 12 output max speed bits These bits are set and cleared by software. Refer to OSPD0[1:0] description
23:22	OSPD11[1:0]	Pin 11 output max speed bits These bits are set and cleared by software. Refer to OSPD0[1:0] description
21:20	OSPD10[1:0]	Pin 10 output max speed bits These bits are set and cleared by software. Refer to OSPD0[1:0] description
19:18	OSPD9[1:0]	Pin 9 output max speed bits These bits are set and cleared by software. Refer to OSPD0[1:0] description
17:16	OSPD8[1:0]	Pin 8 output max speed bits These bits are set and cleared by software. Refer to OSPD0[1:0] description
15:14	OSPD7[1:0]	Pin 7 output max speed bits These bits are set and cleared by software. Refer to OSPD0[1:0] description
13:12	OSPD6[1:0]	Pin 6 output max speed bits These bits are set and cleared by software. Refer to OSPD0[1:0] description
11:10	OSPD5[1:0]	Pin 5 output max speed bits These bits are set and cleared by software. Refer to OSPD0[1:0] description
9:8	OSPD4[1:0]	Pin 4 output max speed bits These bits are set and cleared by software. Refer to OSPD0[1:0] description
7:6	OSPD3[1:0]	Pin 3 output max speed bits These bits are set and cleared by software. Refer to OSPD0[1:0] description
5:4	OSPD2[1:0]	Pin 2 output max speed bits These bits are set and cleared by software.

		Refer to OSPD0[1:0] description
3:2	OSPD1[1:0]	Pin 1 output max speed bits These bits are set and cleared by software. Refer to OSPD0[1:0] description
1:0	OSPD0[1:0]	Pin 0 output max speed bits These bits are set and cleared by software. X0: Output max speed 2M (reset value) 01: Output max speed 10M 11: Output max speed 50M

7.4.4. 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

7.4.5. 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

7.4.6. 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	Pin output control (y=0..15) These bits are set and cleared by software. 0: Pin output low 1: Pin output high

7.4.7. 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

7.4.8. 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

7.4.9. 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

7.4.10. 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 1 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

7.4.11. 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

7.4.12. 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	Pin 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

8. Cyclic redundancy checks management unit (CRC)

8.1. Overview

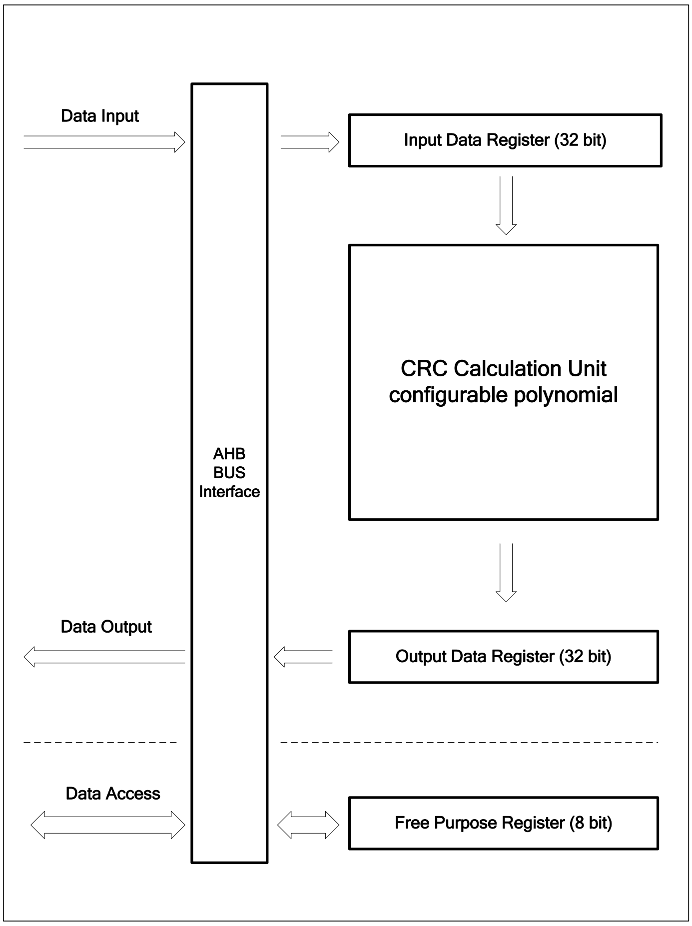
A cyclic redundancy check (CRC) is an error-detecting code commonly used in digital networks and storage devices to detect accidental changes to raw data.

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

8.2. 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 8-1. Block diagram of CRC calculation unit



8.3. 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.

8.4. Register definition

CRC base address: 0x4002 3000

8.4.1. Data register (CRC_DATA)

Address offset: 0x00

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
DATA[31:16]															
rw															

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

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.

8.4.2. Free data register (CRC_FDATA)

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								FDATA[7:0]							
								rw							

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.

8.4.3. 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.

8.4.4. 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
IDATA[31:16]															
rw															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
IDATA[15:0]															
rw															

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.

8.4.5. Polynomial register (CRC_POLY)

Address offset: 0x14

Reset value: 0x04C1 1DB7

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
POLY[31:16]															
rw															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
POLY[15:0]															
rw															

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

9. True random number generator (TRNG)

9.1. Overview

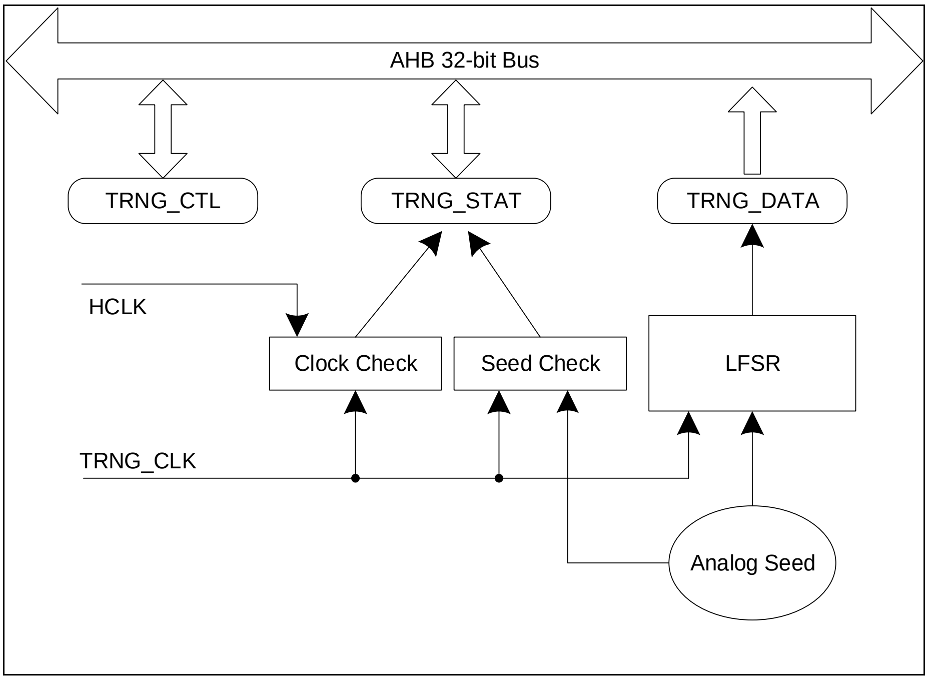
The true random number generator (TRNG) module can generate a 32-bit random value by using continuous analog noise.

9.2. Characteristics

- About 40 periods of TRNG_CLK are needed between two consecutive random numbers.
- 32-bit random value seed is generated from analog noise, so the random number is a true random number.

9.3. Function overview

Figure 9-1. TRNG block diagram



The random number seed comes from analog circuit. This analog seed is then plugged into a linear feedback shift register (LFSR), where a 32-bit width random number is generated.

The analog seed is generated by several ring oscillators. The LFSR is driven by a configurable TRNG_CLK (refer to ***Reset and clock unit (RCU)*** chapter), so that the quality of the generated random number depends on TRNG_CLK exclusively, no matter what HCLK frequency was set or not.

The 32-bit value of LFSR will transfer into TRNG_DATA register after a sufficient number of seeds have been sent to the LFSR.

At the same time, the analog seed and TRNG_CLK clock are monitored. When an analog seed error or a clock error occurs, the corresponding status bit in TRNG_STAT will be set and an interrupt will generate if the IE bit in TRNG_CTL is set.

9.3.1. Operation flow

The following steps are recommended for using TRNG block:

1). Enable the interrupt as necessary, so that when a random number or an error occurs, an interrupt will be generated.
2). Enable IRC48M clock or select CK_PLL for USBSEL[1:0] to enable CK_TRNG clock.
3). Enable the TRNGEN bit.
4). When an interrupt occurs, check the status register TRGN_STAT, if SEIF=0, CEIF=0 and DRDY=1, then the random value in the data register could be read.

As required by the FIPS PUB 140-2, the first random data in data register should be saved but not be used. Every subsequent new random data should be compared to the previously random data. The data can only be used if it is not equal to the previously one.

9.3.2. Error flags

(2) Clock error

When the TRNG_CLK frequency is lower than the 1/16 of HCLK, the CECS and CEIF bit will be set. In this case, the application should check TRNG_CLK and HCLK frequency configurations and then clear CEIF bit. Clock error will not impact the previous random data.

(2) Seed error

When the analog seed is not changed or always changing during 64 TRNG_CLK periods, the SECS and SEIF bit will be set. In this case, the random data in data register should not be used. The application needs to clear the SEIF bit, then clear and set TRNGEN bit for restarting the TRNG.

9.4. Register definition

TRNG base address: 0x5006 0800

9.4.1. Control register (TRNG_CTL)

Address offset: 0x00

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
Reserved												TRNGIE	TRNGEN	Reserved	
												rw	rw		

Bits	Fields	Descriptions
31:4	Reserved	Must be kept at reset value.
3	TRNGIE	Interrupt enabled bit. This bit controls the generation of an interrupt when DRDY, SEIF or CEIF was set. 0: Disable TRNG interrupt 1: Enable TRNG interrupt
2	TRNGEN	TRNG enabled bit. 0: Disable TRNG module (reduce power consuming) 1: Enable TRNG module
1:0	Reserved	Must be kept at reset value.

9.4.2. Status register (TRNG_STAT)

Address offset: 0x04

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
Reserved									SEIF	CEIF	Reserved		SECS	CECS	DRDY
									rc_w0	rc_w0			r	r	r

Bits	Fields	Descriptions

31:7	Reserved	Must be kept at reset value.
6	SEIF	Seed error interrupt flag This bit will be set if more than 64 consecutive same bit or more than 32 consecutive 01 (or 10) changing are detected. 0: No fault detected 1: Seed error has been detected. The bit is cleared by writing 0.
5	CEIF	Clock error interrupt flag This bit will be set if TRNG_CLK frequency is lower than 1 / 16 HCLK frequency. 0: No fault detected 1: Clock error has been detected. The bit is cleared by writing 0.
4:3	Reserved	Must be kept at reset value.
2	SECS	Seed error current status 0: Seed error is not detected at current time. In case of SEIF = 1 and SECS = 0, it means seed error has been detected before but now is recovered. 1: Seed error is detected at current time if more than 64 consecutive same bits or more than 32 consecutive 01(or 10) changing are detected
1	CECS	Clock error current status 0: Clock error is not detected at current time. In case of CEIF = 1 and CECS = 0, it means clock error has been detected before but now is recovered. 1: Clock error is detected at current time. TRNG_CLK frequency is lower than 1 / 16 HCLK frequency.
0	DRDY	Random data ready status bit. This bit is cleared by reading the TRNG_DATA register and set when a new random number is generated. 0: The content of TRNG data register is not available. 1: The content of TRNG data register is available.

9.4.3. Data register (TRNG_DATA)

Address offset: 0x08

Reset value: 0x0000 0000

Application must make sure DRDY is set before reading 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
TRNGDATA[31:16]															
r															

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

Bits	Fields	Descriptions
31:0	TRNGDATA[31:0]	32-bit random data

10. Cryptographic Acceleration Unit (CAU)

10.1. Overview

The cryptographic acceleration unit (CAU) is used to encipher and decipher data with DES, Triple-DES or AES (128, 192, or 256) algorithms. It is fully compliant implementation of the following standards:

- The Data Encryption Standard (DES) and the Triple Data Encryption Algorithm (TDEA) are announced by Federal Information Processing Standards Publication (FIPS) 46-3, October 25, 1999. It follows the American National Standards Institute (ANSI) X9.52 standard.
- The Advanced Encryption Standard (AES) is announced by Federal Information Processing Standards Publication 197, November 26, 2001.

DES / TDES / AES algorithms with different key sizes are supported to perform data encryption and decryption in the CAU in multiple modes.

The CAU is a 32-bit peripheral, DMA transfer is supported and data can be accessed in the input and output FIFO.

10.2. Characteristics

- DES, TDES and AES encryption/decryption algorithms are supported.
- Multiple modes are supported respectively in DES, TDES and AES, including Electronic codebook (ECB), Cipher block chaining (CBC), Counter mode (CTR), Galois/counter mode (GCM), Galois message authentication code mode (GMAC), Counter with CBC-MAC (CCM), Cipher Feedback mode (CFB) and Output Feedback mode(OFB).
- DMA transfer for incoming and outgoing data is supported.

DES / TDES

- Supports the ECB and CBC chaining algorithms.
- two 32-bit initialization vectors (IV) are used in CBC mode.
- 8*32-bit input and output FIFO.
- Multiple data types are supported, including No swapping, Half-word swapping Byte swapping and Bit swapping.
- Data are transferred by DMA, CPU during interrupts, or without both of them.

AES

- Supports the ECB, CBC, CTR, GCM, GMAC, CCM, CFB and OFB chaining algorithms.
- Supports 128-bit, 192-bit and 256-bit keys.
- four 32-bit initialization vectors (IV) are used in CBC, CTR, GCM, GMAC, CCM, CFB

and OFB modes.

- 8*32-bit input and output FIFO.
- Multiple data types are supported, including No swapping, Half-word swapping Byte swapping and Bit swapping.
- Data can be transferred by DMA, CPU during interrupts, or without both of them.

10.3. CAU data type and initialization vectors

10.3.1. Data type

The cryptographic acceleration unit receives data of 32 bits at a time, while they are processed in 64 / 128 bits for DES / AES algorithms. For each data block, according to the data type, the data could be bit/byte/half-word/no swapped before they are transferred into the cryptographic acceleration processor. The same swapping operation should be also performed on the processor output data before they are collected. Note the least-significant data always occupies the lowest address location no matter which data type is configured, because the system memory is little-endian.

Figure 10-1. DATAM No swapping and Half-word swapping and *Figure 10-2. DATAM Byte swapping and Bit swapping* illustrate the 128-bit AES block data swapping according to different data types. (For DES, the data block is two 32-bit words, please refer to the first two words data swapping in the figure).

Figure 10-1. DATAM No swapping and Half-word swapping

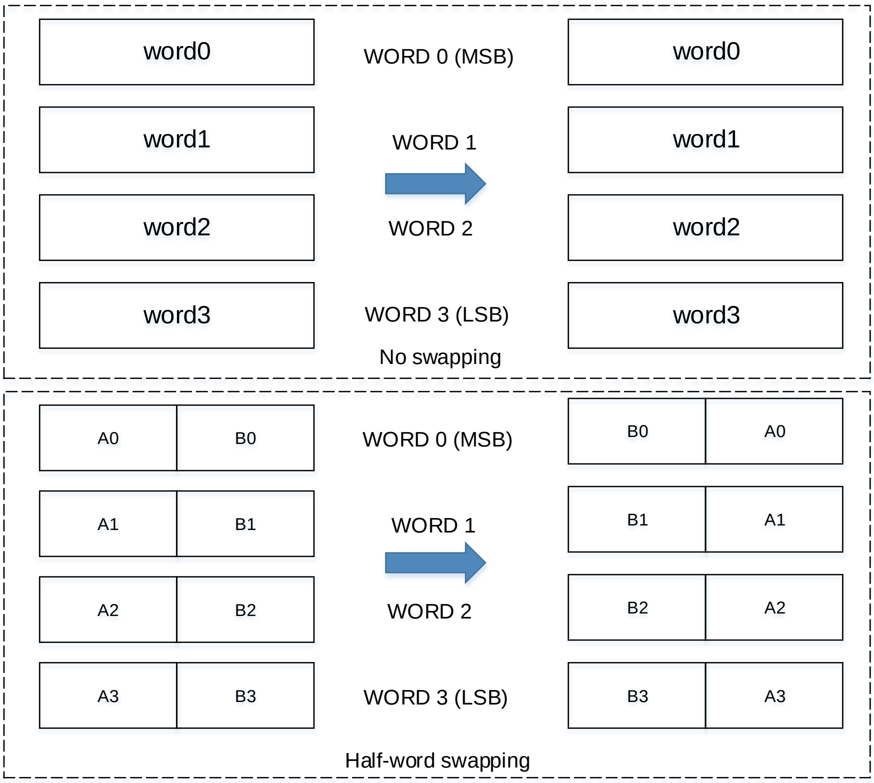
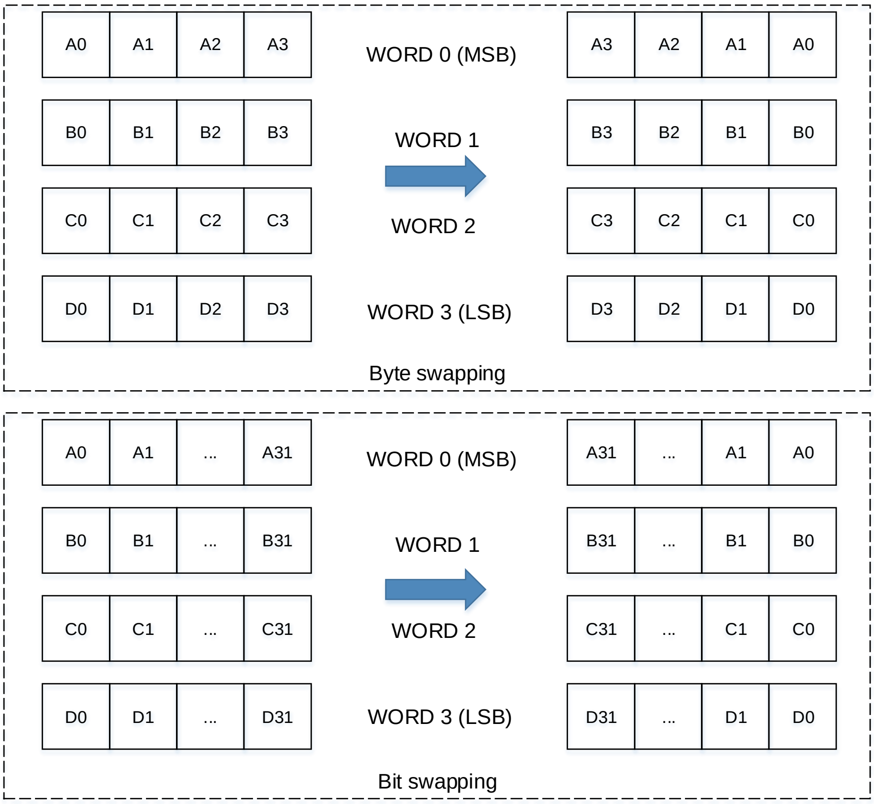


Figure 10-2. DATAM Byte swapping and Bit swapping



10.3.2. Initialization vectors

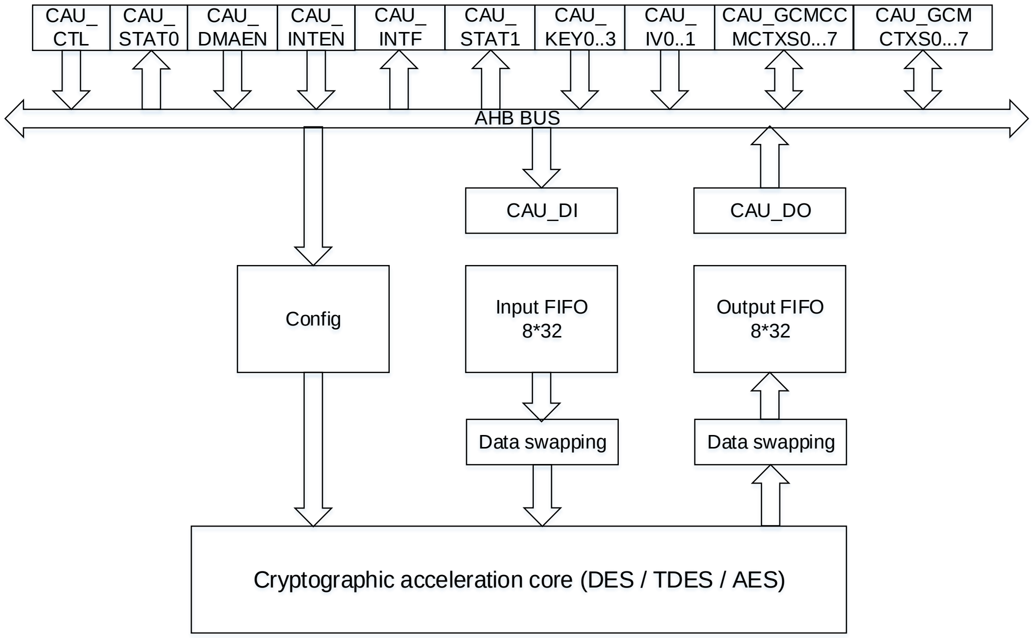
The initialization vectors are used in CBC, CTR, GCM, GMAC, CCM, CFB and OFB modes to XOR with data blocks. They are independent of plaintext and ciphertext, and the DATAM value will not affect them. Note the initialization vector registers CAU_IV0..1(H/L) can only be written when BUSY is 0, otherwise the write operations are invalid.

10.4. Cryptographic acceleration processor

The cryptographic acceleration unit implements DES and AES acceleration processors, which are detailed described in section ***DES / TDES cryptographic acceleration*** processor and ***AES cryptographic acceleration processor***.

Figure 10-3. CAU diagram shows the block diagram of the cryptographic acceleration unit.

Figure 10-3. CAU diagram



10.4.1. DES / TDES cryptographic acceleration processor

The DES/TDES cryptographic acceleration processor contains the DES algorithm (DEA), cryptographic keys (1 for DES algorithm and 3 for TDES algorithm), and initialization vectors in CBC mode.

DES / TDES key

[KEY1] is used in DES and [KEY3 KEY2 KEY1] are used in TDES respectively.
When TDES algorithm is configured, three different keying options are allowed:

1. Three same keys

The three keys KEY3, KEY2 and KEY1 are completely equal, which means KEY3 = KEY2 = KEY1. FIPS PUB 46-3 – 1999 (and ANSI X9.52 -1998) refers to this option. It is easy to understand that this mode is equivalent to DES.

2. Two different keys

In this option, KEY2 is different from KEY1, and KEY3 is equal to KEY1, which means, KEY1 and KEY2 are independent while KEY3 = KEY1. FIPS PUB 46-3 – 1999 (and ANSI X9.52 – 1998) refers to this option.

3. Three different keys

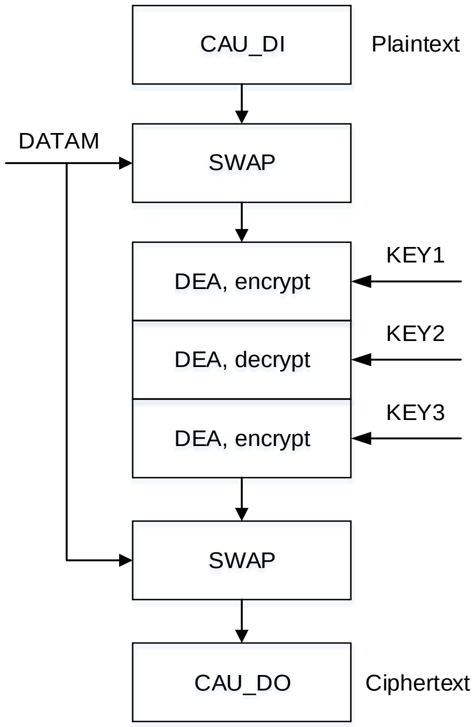
In this option, KEY1, KEY2 and KEY3 are completely independent. FIPS PUB 46-3 -1999 (and ANSI X9.52 – 1998) refers to this option.

More information of the thorough explanation of the key used in the DES / TDES please refer to FIPS PUB 46-3 (and ANSI X9.52 -1998), and the explanation process is omitted in this manual.

DES / TDES ECB encryption

The 64-bit input plaintext is first obtained after data swapping according to the data type. When the TDES algorithm is configured, the input data block is read in the DEA and encrypted using KEY1. The output is fed back directly to next DEA and then decrypted using KEY2. After that, the output is fed back directly to the last DEA and encrypted with KEY3. The output after above processes is then swapped back according to the data type again, and a 64-bit ciphertext is produced. When the DES algorithm is configured, the result of the first DEA encrypted using KEY1 is swapped directly according to the data type, and a 64-bit ciphertext is produced. The procedure of DES/TDES ECB mode encryption is illustrated in ***Figure 10-4. DES / TDES ECB*** **encryption**.

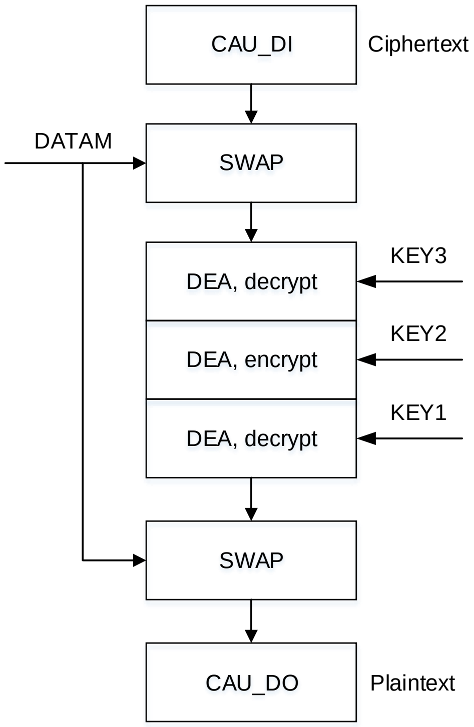
Figure 10-4. DES / TDES ECB encryption



DES / TDES ECB decryption

The 64-bit input ciphertext is first obtained after data swapping according to the data type. When the TDES algorithm is configured, the input data block is read in the DEA and decrypted using KEY3. The output is fed back directly to next DEA and then encrypted using KEY2. After that, the output is fed back directly to the last DEA and decrypted with KEY1. The output after above process is then swapped back according to the data type again, and a 64-bit plaintext is produced. When the DES algorithm is configured, the result of the first DEA decrypted using KEY1 is swapped directly according to the data type, and a 64-bit plaintext is produced. The procedure of DES/TDES ECB mode decryption is illustrated in ***Figure 10-5. DES / TDES ECB*** **decryption**.

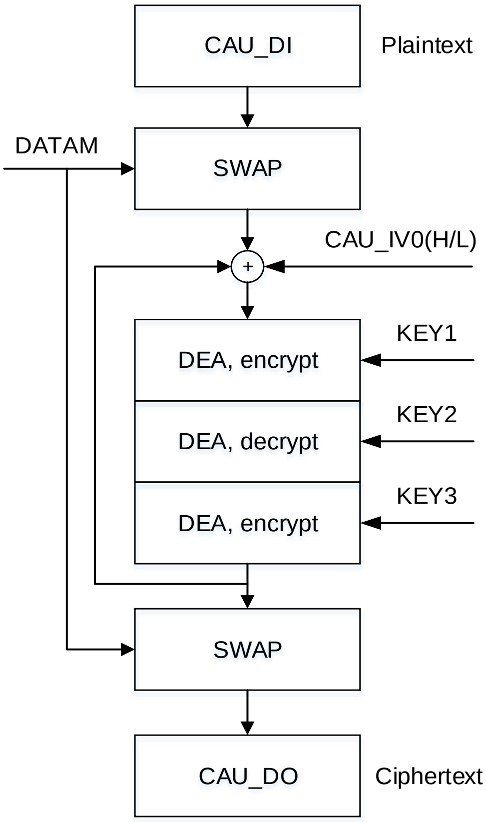
Figure 10-5. DES / TDES ECB decryption



DES / TDES CBC encryption

The input data of the DEA block in CBC mode consists of two aspects: the input plaintext after data swapping according to the data type, and the initialization vectors. When the TDES algorithm is configured, the XOR result of the swapped plaintext data block and the 64-bit initialization vector CAU_IV0..1 is read in the DEA and encrypted using KEY1. The output is fed back directly to next DEA and then decrypted using KEY2. After that, the output is fed back directly to the last DEA and encrypted with KEY3. The result is then used as the next initialization vector and exclusive-Ored with the next plaintext data block to process next encryption. The above operations are repeated until the last plaintext block is encrypted. Note if the plaintext message does not consist of an integral number of data blocks, the final partial data block should be encrypted in a specified manner. At last, the output ciphertext is also obtained after data swapping according to the data type. When the DES algorithm is configured, the state and process of the second and third block of DEA should be omitted. The procedure of DES/TDES CBC mode encryption is illustrated in ***Figure 10-6. DES / TDES CBC* encryption**.

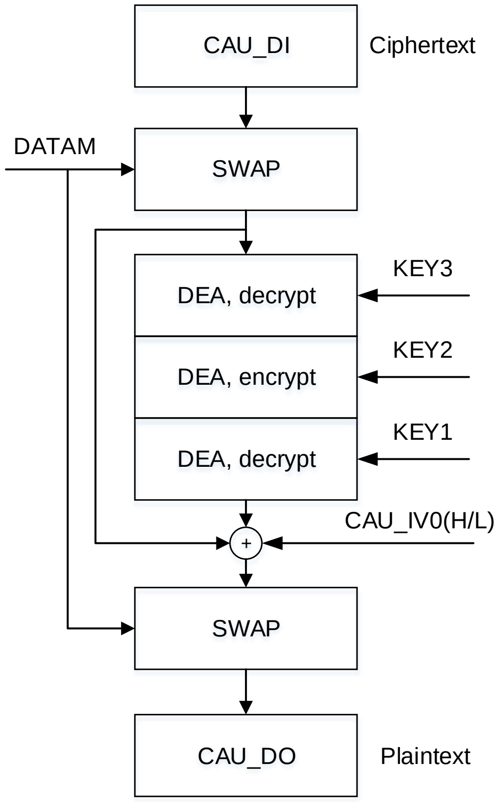
Figure 10-6. DES / TDES CBC encryption



DES / TDES CBC decryption

In DES/TDES CBC decryption, when the TDES algorithm is configured, the first ciphertext block is used directly after data swapping according to the data type, it is read in the DEA and decrypted using KEY3. The output is fed back directly to next DEA and then encrypted using KEY2. After that, the output is fed back directly to the last DEA and decrypted with KEY1. The first result of above process is then XORed with the initialization vector which is the same as that used during encryption. At the same time, the first ciphertext is then used as the next initialization vector and exclusive-Ored with the next result after DEA blocks. The above operations are repeated until the last ciphertext block is decrypted. Note if the ciphertext message does not consist of an integral number of data blocks, the final partial data block should be decrypted in a specified manner same to that in encryption. At last, the output plaintext is also obtained after data swapping according to the data type. When the DES algorithm is configured, the state and process of the second and third block of DEA should also be omitted. The procedure of DES/TDES CBC mode decryption is illustrated in ***Figure 10-7. DES / TDES CBC*** **decryption**.

Figure 10-7. DES / TDES CBC decryption



10.4.2. AES cryptographic acceleration processor

The AES cryptographic acceleration processor consists of three components, including the AES algorithm (AEA), multiple keys and the initialization vectors or Nonce.

Three lengths of AES keys are supported: 128, 192 and 256 bits, and different initialization vectors or nonce are used depends on the operation mode.

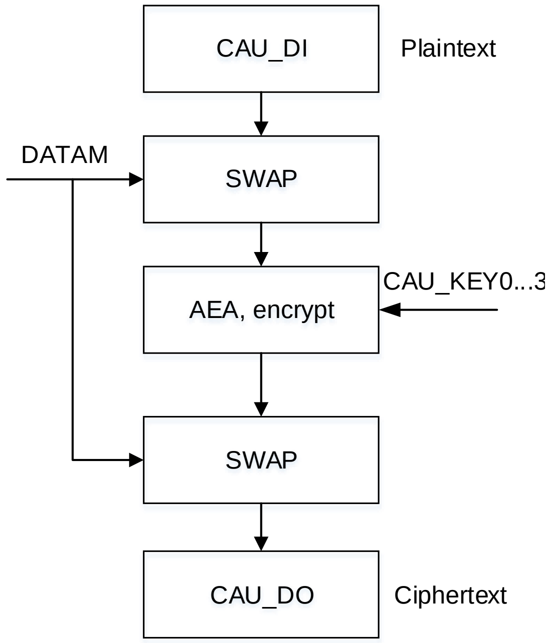
The AES key is used as [KEY3 KEY2] when the key size is configured as 128, [KEY3 KEY2 KEY1] when the key size is configured as 192 and [KEY3 KEY2 KEY1 KEY0] when the key size is configured as 256.

The thorough explanation of the key used in the AES is provided in FIPS PUB 197 (November 26, 2001), and the explanation process is omitted in this manual.

AES-ECB mode encryption

The 128-bit input plaintext is first obtained after data swapping according to the data type. The input data block is read in the AEA and encrypted using the 128, 192 or 256-bit key. The output after above process is then swapped back according to the data type again, and a 128-bit ciphertext is produced and stored in the out FIFO. The procedure of AES ECB mode encryption is illustrated in ***Figure 10-8. AES ECB encryption***.

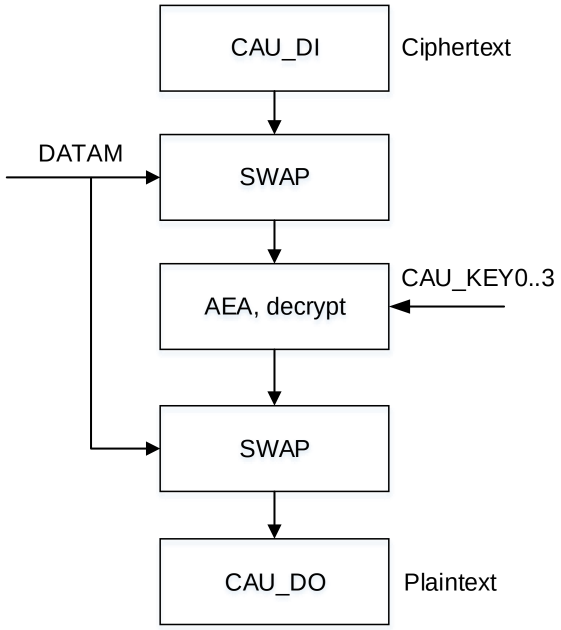
Figure 10-8. AES ECB encryption



AES-ECB mode decryption

First of all, the key derivation must be completed to prepare the decryption keys, the input key of the key schedule is the same to that used in encryption. The last round key obtained from the above operation is then used as the first round key in the decryption. After the key derivation, the 128-bit input ciphertext is first obtained after data swapping according to the data type. The input data block is read in the AEA and decrypted using keys prepared above. The output is then swapped back according to the data type again, and a 128-bit plaintext is produced. The procedure of AES ECB mode decryption is illustrated in ***Figure 10-9. AES ECB decryption***.

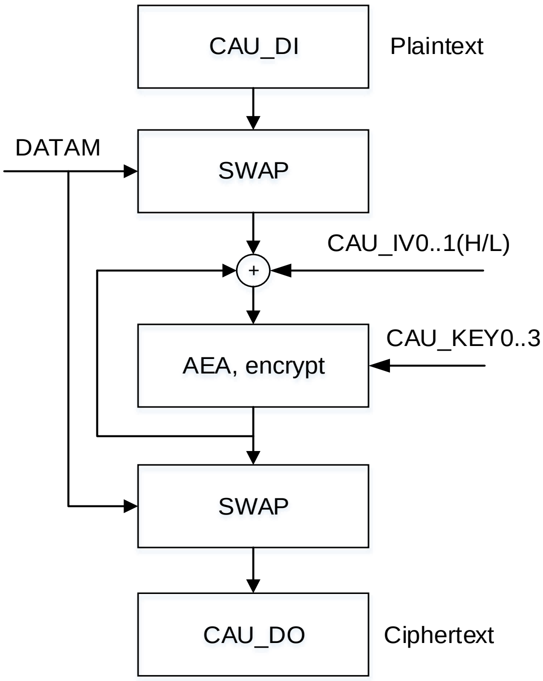
Figure 10-9. AES ECB decryption



AES-CBC mode encryption

The input data of the AEA block in CBC mode consists of two aspects: the input plaintext after data swapping according to the data type, and the initialization vectors. The XOR result of the swapped plaintext data block and the 128-bit initialization vector CAU_IV0..1 is read in the AEA and encrypted using the 128-, 192-, 256-bit key. The result is then used as the next initialization vector and exclusive-Ored with the next plaintext data block to process next encryption. The above operations are repeated until the last plaintext block is encrypted. Note if the plaintext message does not consist of an integral number of data blocks, the final partial data block should be encrypted in a specified manner. At last, the output ciphertext is also obtained after data swapping according to the data type. The procedure of AES CBC mode encryption is illustrated in ***Figure 10-10. AES CBC encryption***.

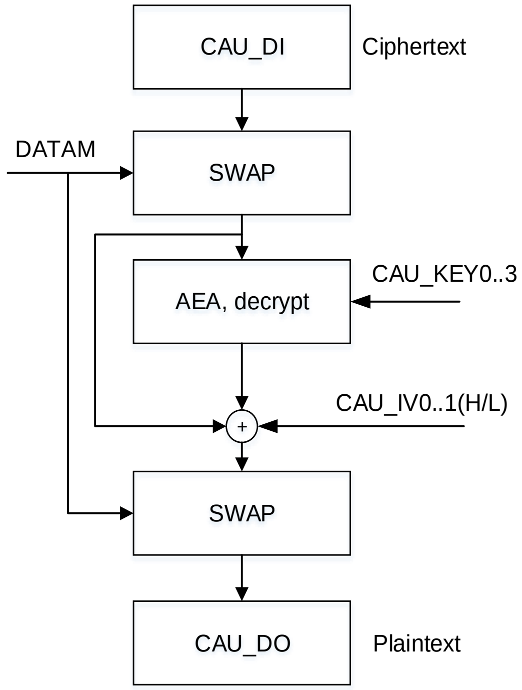
Figure 10-10. AES CBC encryption



AES-CBC mode decryption

Similar to that in AES-ECB mode decryption, the key derivation also must be completed first to prepare the decryption keys, the input of the key schedule should be the same to that used in encryption. The last round key obtained from the above operation is then used as the first round key in the decryption. After the key derivation, the 128-bit input ciphertext is first obtained after data swapping according to the data type. The input data block is read in the AEA and decrypted using keys prepared above. At the same time, the first ciphertext is then used as the next initialization vector and exclusive-Ored with the next result after AEA blocks (The first initialization is obtained directly from the CAU_IV0..1 registers). The above operations are repeated until the last ciphertext block is decrypted. Note if the ciphertext message does not consist of an integral number of data blocks, the final partial data block should be decrypted in a specified manner same to that in encryption. At last, the output plaintext is also obtained after data swapping according to the data type. The procedure of AES CBC mode decryption is illustrated in ***Figure 10-11. AES CBC decryption***.

Figure 10-11. AES CBC decryption



AES-CTR mode

In counter mode, a counter is used in addition with a nonce value to be encrypted and decrypted in AEA, and the result will be used for the XOR operation with the plaintext or the ciphertext. As the counter is incremented from the same initialized value for each block in encryption and decryption, the key schedule during the encryption and decryption are the same. Then decryption operation acts exactly in the same way as the encryption operation. Only the 32-bit LSB of the 128-bit initialization vector represents the counter, which means the other 96 bits are unchanged during the operation, and the initial value should be set to 1. Nonce is 32-bit single-use random value and should be updated to each communication block. And the 64-bit initialization vector is used to ensure that a given value is used only once for a given key. ***Figure 10-12. Counter block structure*** illustrates the counter block structure and ***Figure 10-13. AES CTR encryption/decryption*** shows the AES CTR encryption/decryption.

Figure 10-12. Counter block structure

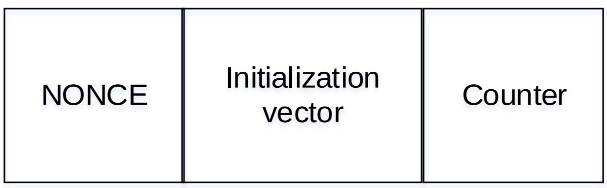
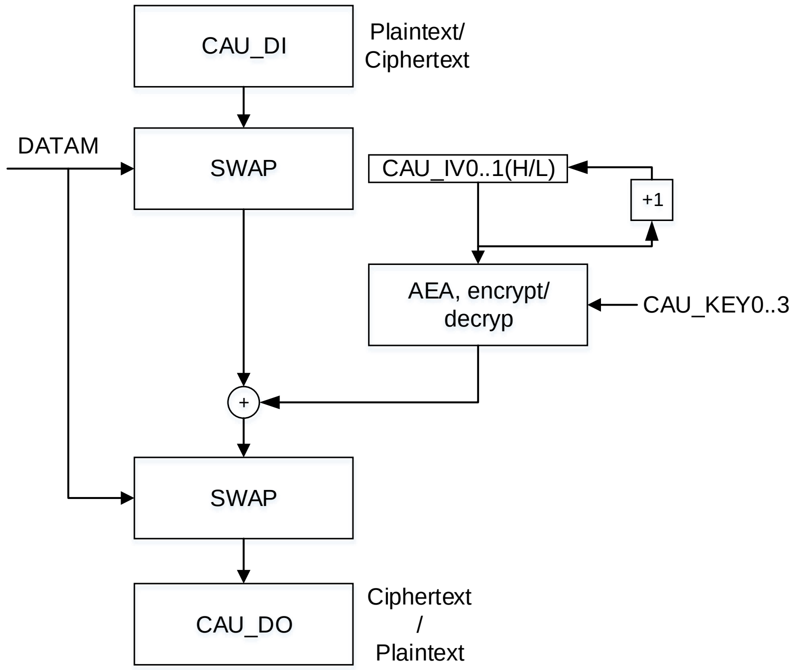


Figure 10-13. AES CTR encryption/decryption



AES-GCM mode

The AES Galois/counter mode (GCM) can be used to encrypt or authenticate message, and then ciphertext and tag can be obtained. This algorithm is based on AES CTR mode to ensure confidentiality. A multiplier over a fixed finite field is used to generate the tag.

In this mode, four steps are required to perform an encryption/decryption:

1. GCM prepare phase

   The hash key is calculated and saved internally to be used later.

   (a) Clear the CAUEN bit to make sure CAU is disabled.
   (b) Configure the ALGM[3:0] bits to ‘1000’.
   (c) Configure GCM_CCMPH[1:0] bits to ‘00’.
   (d) Configure key registers and initialization vectors.
   (e) Enable CAU by writing 1 to CAUEN bit.
   (f) Wait until CAUEN bit is cleared by hardware, and then enable CAU again for following phases.

2. GCM AAD (additional authenticated data) phase

   This phase must be performed after GCM prepare phase and also precede the encryption/decryption phase. In this phase, data is authenticated but not protected.

   (g) Configure GCM_CCMPH[1:0] bits to ‘01’.
   (h) Write data into CAU_DI register, INF and IEM flags can be used to determine if the input FIFO can receive data. The size of the header must be a multiple of 128bits. DMA can also be used.

(i) Repeat (h) until all AAD data are supplied, wait until BUSY bit is cleared.

3. GCM encryption/decryption phase

   This phase must be performed after GCM AAD phase. In this phase, the message is authenticated and encrypted/decrypted.

   (j) Configure GCM_CCMPH[1:0] bits to ‘10’.
   (k) Configure the computation direction in CAUDIR.
   (l) Write data into CAU_DI register, INF and IEM flags can be used to determine if the input FIFO can receive data. ONE and OFU flags can be used to check if the output FIFO is not empty. If so, read the CAU_DO register. DMA can also be used.
   (m) Repeat (l) step until all payload blocks are processed.

4. GCM tag phase

   In this phase, the final authentication tag is generated.

   (n) Configure GCM_CCMPH[1:0] bits to ‘11’.
   (o) Write the input into the CAU_DI register, 4 times write operation is needed. The input consists of the header data size (64bits) and the payload data size (64bits).
   (p) Wait until the ONE flag is set to 1, and then read CAU_DO 4 times. The output corresponds to the authentication tag.
   (q) Disable the CAU.

Note: The key should be prepared at the beginning when a decryption is performed.

AES-GMAC mode

The AES Galois message authentication code mode is also supported to authenticate the message. It is processing based on the AES-GCM mode, while the payload phase is by-passed.

AES-CCM mode

The AES combined cipher machine mode, which is similar to AES-GCM mode, also allows encrypting and authenticating message. It is also based on AES-CTR mode to ensure confidentiality. In this mode, AES-CBC is used to generate a 128-bit tag.

The CCM standard (RFC 3610 Counter with CBC-MAC (CCM) dated September 2003) defines particular encoding rules for the first authentication block (B0 in the standard). In particular, the first block includes flags, a nonce and the payload length expressed in bytes. The CCM standard specifies another format, called A or counter, for encryption/decryption. The counter is incremented during the payload phase and its 32 LSB bits are initialized to ‘1’ during the tag generation (A0 packet in the CCM standard).

Note: The formatting operation of B0 packet should be handled by software.

In this mode, four steps are required to perform an encryption/decryption:

1. CCM prepare phase

   In this phase, B0 packet (the first packet) is programmed into the CAU_DI register. CAU_DO never contain data in this phase.

   (a) Clear the CAUEN bit to make sure CAU is disabled.
   (b) Configure the ALGM[3:0] bits to ‘1001’.
   (c) Configure GCM_CCMPH[1:0] bits to ‘00’.
   (d) Configure key registers and initialization vectors.
   (e) Enable CAU by writing 1 to CAUEN bit.
   (f) Program the B0 packet into the CAU_DI.
   (g) Wait until CAUEN is cleared by hardware, and then enable CAU again for following phases.

2. CCM AAD (additional authenticated data) phase

   This phase must be performed after CCM prepare phase and also precede the encryption/decryption phase. In this phase, CAU_DO never contain data in this phase.

   This phase can be by-passed if there is no additional authenticated data.

   (h) Configure GCM_CCMPH[1:0] bits to ‘01’
   (i) Write data into CAU_DI register, INF and IEM flags can be used to determine if the input FIFO can receive data. The size of the header must be a multiple of 128 bits. DMA can also be used.
   (j) Repeat (i) until all AAD data are supplied, wait until BUSY bit is cleared

3. CCM encryption/decryption phase

   This phase must be performed after CCM AAD phase. In this phase, the message is authenticated and encrypted / decrypted.

   (k) Configure GCM_CCMPH[1:0] bits to ‘10’
   (l) Configure the computation direction in CAUDIR
   (m) Write data into CAU_DI register, INF and IEM flags can be used to determine if the input FIFO can receive data. ONE and OFU flags can be used to check if the output FIFO is not empty. If so, read the CAU_DO register. DMA can also be used.
   (n) Repeat (m) step until all payload blocks are processed.

4. CCM tag phase

   In this phase, the final authentication tag is generated.

   (o) Configure GCM_CCMPH[1:0] bits to ‘11’
   (p) Write the 128 bit input into the CAU_DI register, 4 times of write operation to CAU_DI is needed. The input is the A0 value.
   (q) Wait until the ONE flag is set to 1, and then read CAU_DO 4 times. The output corresponds to the authentication tag.
   (r) Disable the CAU

AES-CFB mode

The Cipher Feedback (CFB) mode is a confidentiality mode that features the feedback of successive ciphertext segments into the input blocks of the forward cipher to generate output blocks that are exclusive-Ored with the plaintext to produce the ciphertext, and vice versa.

AES-OFB mode

The Output Feedback (OFB) mode is a confidentiality mode that features the iteration of the forward cipher on an IV to generate a sequence of output blocks that are exclusive-Ored with the plaintext to produce the ciphertext, and vice versa.

10.5. Operating modes

Encryption

1. Disable the CAU by resetting the CAUEN bit in the CAU_CTL register.
2. Select and configure the key length with the KEYM bits in the CAU_CTL register if AES algorithm is chosen.
3. Configure the CAU_KEY0..3(H / L) registers according to the algorithm.
4. Configure the DATAM bit in the CAU_CTL register to select the data swapping type.
5. Configure the algorithm (DES / TDES / AES) and the chaining mode (ECB / CBC / CTR / GCM / GMAC / CCM / CFB / OFB) by writing the ALGM[3:0] bit in the CAU_CTL register.
6. Configure the encryption direction by writing 0 to the CAUDIR bit in the CAU_CTL register.
7. Configure the initialization vectors by writing the CAU_IV0..1 registers.
8. Flush the input FIFO and output FIFO by configure the FFLUSH bit in the CAU_CTL register when CAUEN is 0.
9. Enable the CAU by set the CAUEN bit as 1 in the CAU_CTL register.
10. If the INF bit in the CAU_STAT0 register is 1, then write data blocks into the CAU_DI register. The data can be transferred by DMA/CPU during interrupts/no DMA or interrupts.
11. Wait for ONE bit in the CAU_STAT0 register is 1 then read the CAU_DO registers. The output data can also be transferred by DMA/CPU during interrupts/no DMA or interrupts.
12. Repeat steps 10, 11 until all data blocks has been encrypted.

Decryption

1. Disable the CAU by resetting the CAUEN bit in the CAU_CTL register.
2. Select and configure the key length with the KEYM bits in the CAU_CTL register if AES algorithm is chosen.
3. Configure the CAU_KEY0..3(H / L) registers according to the algorithm.
4. Configure the DATAM bit in the CAU_CTL register to select the data swapping type.
5. Configure the ALGM[3:0] bits to “0111” in the CAU_CTL register to complete the key derivation.

6. Enable the CAU by set the CAUEN bit as 1.
7. Wait until the BUSY and CAUEN bit return to 0 to make sure that the decryption keys are prepared.
8. Configure the algorithm (DES / TDES / AES) and the chaining mode (ECB / CBC / CTR / GCM / GMAC / CCM / CFB / OFB) by writing the ALGM[3:0] bit in the CAU_CTL register.
9. Configure the decryption direction by writing 1 to the CAUDIR bit in the CAU_CTL register.
10. Configure the initialization vectors by writing the CAU_IV0..1 registers.
11. Flush the input FIFO and output FIFO by configure the FFLUSH bit in the CAU_CTL register when CAUEN is 0.
12. Enable the CAU by set the CAUEN bit as 1 in the CAU_CTL register.
13. If the INF bit in the CAU_STAT0 register is 1, then write data blocks into the CAU_DI register. The data can be transferred by DMA/CPU during interrupts/no DMA or interrupts.
14. Wait for ONE bit in the CAU_STAT0 register is 1, then read the CAU_DO registers. The output data can also be transferred by DMA/CPU during interrupts/no DMA or interrupts.
15. Repeat steps 13, 14 until all data blocks has been decrypted.

Data append

For GCM payload encryption or CCM payload decryption, CAU supports non 128 bit integer multiple data block processing. When the last data block is less than 128bit, use ‘0’ to fill the remaining bits, and then configure the number of bytes to be filled in the NBPILB bitfield of the CAU_CTL register. AES will automatically remove the the number of filled pads and encrypt it. It should be noted that the NBPILB bitfield should be configured after the encryption of the penultimate data block is completed.

10.6. CAU DMA interface

The DMA can be used to transfer data blocks with the interface of the cryptographic acceleration unit. The operations can be controlled by the CAU_DMAEN register. DMAIEN is used to enable the DMA request during the input phase, then a word is written into CAU_DI from DMA. DMAOEN is used to enable the DMA request during the output phase, then a word is read from the CAU.

Single and Burst transfers are both supported to ensure the data transfer if the number of words is not an integral multiple of burst size. Note the DMA controller should be configured to perform burst of 4 words or less to make sure no data will be lost. DMA channel for output data has a higher priority than that channel for input data so that the output FIFO can be empty earlier than that the input FIFO is full.

10.7. CAU interrupts

There are two types of interrupt registers in CAU, which are CAU_STAT1 and CAU_INTF. In CAU, the interrupt is used to indicate the situation of the input and output FIFO.

Any of input and output FIFO interrupt can be enabled or disabled by configuring the Interrupt Enable register CAU_INTEN. Value 1 of the register enable the interrupts.

Input FIFO interrupt

The input FIFO interrupt is asserted when the number of words in the input FIFO is less than four words, then ISTA is asserted. And if the input FIFO interrupt is enabled by IINTEN with a 0 value, the IINTF is also asserted. Note if the CAUEN is low, then the ISTA and IINTF are also always low.

Output FIFO interrupt

The output FIFO interrupt is asserted when the number of words in the output FIFO is more than one words, then OSTA is asserted. And if the output FIFO interrupt is enabled by OINTEN with a 0 value, the OINTF is also asserted. Note Unlike that of Input FIFO interrupt, the value of CAUEN will never affect the situation of OSTA and OINTF.

10.8. CAU suspended mode

It is possible to suspend a data block if another new data block with a higher priority needs to be processed in CAU. The following steps can be performed to complete the encryption/decryption acceleration of the suspended data blocks.

When DMA transfer is used:

1. Stop the current input transfer. Clear the DMAIEN bit in the CAU_DMAEN register.
2. When it is DES or AES, wait until both the input and output FIFO are both empty if the input FIFO is not empty (IEM = 0), then write a word of data into CAU_DI register, do as so until the IEM is checked to be 1, then wait until the BUSY bit is cleared, so that the next data block will not be affected by the last one. Case of TDES is similar to that of AES except that it does not need to wait until the input FIFO is empty.
3. Stop the output transfer by clearing the DMAOEN bit in the CAU_DMAEN register. And disable the CAU by clearing the CAUEN bit in the CAU_CTL register.
4. Save the configuration, including the key size, data type, operation mode, direction, GCM CCM phase and the key values. When it is CBC, CTR, GCM, GMAC, CCM, CFB or OFB chaining mode, the initialization vectors should also be stored. When it is GCM, GMAC, or CCM mode, the context switch registers should also be stored.
5. Configure and process the new data block.
6. Restore the process before. Configure the CAU with the parameters stored before, and prepare the key and initialization vectors. Then enable CAU by setting the CAUEN bit in the CAU_CTL register.

When data transfer is done by CPU access to CAU_DI and CAU_DO:

1. When the data transfer is done by CPU access, then wait for the fourth read of the

CAU_DO register and before the next CAU_DI write access so that the message is suspended at the end of a block processing.

2. Disable the CAU by clearing the CAUEN bit in the CAU_CTL register.
3. Save the configuration, including the key size, data type, operation mode, direction, GCM CCM phase and the key values. When it is CBC, CTR, GCM, GMAC, CCM, CFB or OFB chaining mode, the initialization vectors should also be stored. When it is GCM, GMAC, or CCM mode, the context switch registers should also be stored.
4. Configure and process the new data block.
5. Restore the process before. Configure the CAU with the parameters stored before, and prepare the key and initialization vectors. Then enable CAU by setting the CAUEN bit in the CAU_CTL register.

10.9. Register definition

CAU access base address: 0x5006 0000

10.9.1. Control register (CAU_CTL)

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								NBPILB[3:0]				ALGM[3]	Reserved	GCM_CCMPH[1:0]	
												rw		rw	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CAUEN	FFLUSH	Reserved				KEYM[1:0]		DATAM[1:0]		ALGM[2:0]			CAUDIR	Reserved	
rw	w					rw		rw		rw			rw		

Bits	Fields	Descriptions
31:24	Reserved	Must be kept at reset value.
23:20	NBPILB[3:0]	Number of bytes padding in last block of payload 0000: All bytes are valid (no padding) 0001: One padding byte of last block ... 1111: 15 padding bytes of last block
19	ALGM[3]	Encryption / decryption algorithm mode bit 3
18	Reserved	Must be kept at reset value.
17:16	GCM_CCMPH[1:0]	GCM CCM phase 00: Prepare phase 01: AAD phase 10: Encryption / decryption phase 11: Tag phase
15	CAUEN	CAU Enable 0: CAU is disabled 1: CAU is enabled **Note:** the CAUEN can be cleared automatically when the key derivation (ALGM=0111b) is finished or the AES-GCM or AES-CCM initial phase finished.
14	FFLUSH	Flush FIFO 0: No effect 1: When CAUEN=1, flush the input and output FIFO Reading this bit always returns 0

13:10	Reserved	Must be kept at reset value.
9:8	KEYM[1:0]	AES key size mode configuration, must be configured when BUSY=0 00: 128-bit key length 01: 192-bit key length 10: 256-bit key length 11: never use
7:6	DATAM[1:0]	Data swapping type mode configuration, must be configured when BUSY=0 00: No swapping 01: Half-word swapping 10: Byte swapping 11: Bit swapping
5:3	ALGM[2:0]	Encryption/decryption algorithm mode bit 0 to bit 2 These bits and bit 19 of CAU_CTL must be configured when BUSY=0 0000: TDES-ECB with CAU_KEY1, 2, 3. Initialization vectors (CAU_IV0..1) are not used 0001: TDES-CBC with CAU_KEY1, 2, 3. Initialization vectors (CAU_IV0) is used to XOR with data blocks 0010: DES-ECB with only CAU_KEY1 Initialization vectors (CAU_IV0..1) are not used 0011: DES-CBC with only CAU_KEY1 Initialization vectors (CAU_IV0) is used to XOR with data blocks 0100: AES-ECB with CAU_KEY0, 1, 2, 3. Initialization vectors (CAU_IV0..1) are not used 0101: AES-CBC with CAU_KEY0, 1, 2, 3. Initialization vectors (CAU_IV0..1) are used to XOR with data blocks 0110: AES_CTR with CAU_KEY0, 1, 2, 3. Initialization vectors (CAU_IV0..1) are used to XOR with data blocks In this mode, encryption and decryption are same, then the CAUDIR is disregarded. 0111: AES key derivation for decryption mode. The input key must be same to that used in encryption. The BUSY bit is set until the process has been finished, and CAUEN is then cleared. 1000: Galois Counter Mode (GCM). This algorithm mode is also used for GMAC algorithm. 1001: Counter with CBC-MAC (CCM). 1010: Cipher Feedback (CFB) mode 1011: Output Feedback (OFB) mode
2	CAUDIR	CAU direction, must be configured when BUSY=0 0: Encryption 1: Decryption
1:0	Reserved	Must be kept at reset value.

10.9.2. Status register 0 (CAU_STAT0)

Address offset: 0x04

Reset value: 0x0000 0003

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											BUSY	OFU	ONE	INF	IEM
											r	r	r	r	r

Bits	Fields	Descriptions
31:5	Reserved	Must be kept at reset value.
4	BUSY	Busy bit 0: No processing. This is because: - CAU is disabled by CAUEN=0 or the processing has been completed. - No enough data or no enough space in the input/output FIFO to perform a data block 1: CAU is processing data or key derivation.
3	OFU	Output FIFO is full 0: Output FIFO is not full 1: Output FIFO is full
2	ONE	Output FIFO is not empty 0: Output FIFO is empty 1: Output FIFO is not empty
1	INF	Input FIFO is not full 0: Input FIFO is full 1: Input FIFO is not full
0	IEM	Input FIFO is empty 0: Input FIFO is not empty 1: Input FIFO is empty

10.9.3. Data input register (CAU_DI)

Address offset: 0x08

Reset value: 0x0000 0000

The data input register is used to transfer plaintext or ciphertext blocks into the input FIFO for processing. The MSB is firstly written into the FIFO and the LSB is the last one. If the CAUEN is 0 and the input FIFO is not empty, when it is read, then the first data in the FIFO is popped out and returned. If the CAUEN is 1, the returned value is undefined. Once it is read, then the FIFO must be flushed.

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
DI[31:16]															
rw															

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

Bits	Fields	Descriptions
31:0	DI[31:0]	Data input Write these bits will write data to IN FIFO, read these bits will return IN FIFO value if CAUEN is 0, or it will return an undefined value

10.9.4. Data output register (CAU_DO)

Address offset: 0x0C

Reset value: 0x0000 0000

The data output register is a read only register. It is used to receive plaintext or ciphertext results from the output FIFO. Similar to CAU_DI, the MSB is read at first while the LSB is read at last.

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
DO[31:16]															
r															

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

Bits	Fields	Descriptions
31:0	DO[31:0]	Data output These bits are read only, read these bits return OUT FIFO value.

10.9.5. DMA enable register (CAU_DMAEN)

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														DMAOEN	DMAIEN
														rw	rw

Bits	Fields	Descriptions
31:2	Reserved	Must be kept at reset value.
1	DMAOEN	DMA output enable 0: DMA for OUT FIFO data is disabled 1: DMA for OUT FIFO data is enabled
0	DMAIEN	DMA input enable 0: DMA for IN FIFO data is disabled 1: DMA for IN FIFO data is enabled

10.9.6. Interrupt enable register (CAU_INTEN)

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														OINTEN	IINTEN
														rw	rw

Bits	Fields	Descriptions
31:2	Reserved	Must be kept at reset value.
1	OINTEN	OUT FIFO interrupt enable 0: OUT FIFO interrupt is disable 1: OUT FIFO interrupt is enable
0	IINTEN	IN FIFO interrupt enable 0: IN FIFO interrupt is disable 1: IN FIFO interrupt is enable

10.9.7. Status register 1 (CAU_STAT1)

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															

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

Bits	Fields	Descriptions
31:2	Reserved	Must be kept at reset value.
1	OSTA	OUT FIFO interrupt status 0: OUT FIFO interrupt status not pending 1: OUT FIFO interrupt status pending
0	ISTA	IN FIFO interrupt status 0: IN FIFO interrupt not pending 1: IN FIFO interrupt flag pending

10.9.8. Interrupt flag register (CAU_INTF)

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														OINTF	IINTF
														r	r

Bits	Fields	Descriptions
31:2	Reserved	Must be kept at reset value.
1	OINTF	OUT FIFO enabled interrupt flag 0: OUT FIFO Interrupt not pending 1: OUT FIFO Interrupt pending
0	IINTF	IN FIFO enabled interrupt flag 0: IN FIFO Interrupt not pending 1: IN FIFO Interrupt pending when CAUEN is 1

10.9.9. Key registers (CAU_KEY0..3(H/L))

Address offset: 0x20 to 0x3C

Reset value: 0x0000 0000

This registers have to be accessed by word (32-bit), and all of them must be written when BUSY is 0.

In DES mode, only CAU_KEY1 is used.

In TDES mode, CAU_KEY1, CAU_KEY2 and CAU_KEY3 are used.

In AES-128 mode, KEY2H[31:0] || KEY2L[31:0] is used as AES_KEY[0:63], and KEY3H[31:0] || KEY3L[31:0] is used as AES_KEY[64:127].

In AES-192 mode, KEY1H[31:0] || KEY1L[31:0] is used as AES_KEY[0:63], KEY2H[31:0] || KEY2L[31:0] is used as AES_KEY[64:127], and KEY3H[31:0] || KEY3L[31:0] is used as AES_KEY[128:191].

In AES-256 mode, KEY0H[31:0] || KEY0L[31:0] is used as AES_KEY[0:63], KEY1H[31:0] || KEY1L[31:0] is used as AES_KEY[64:127], KEY2H[31:0] || KEY2L[31:0] is used as AES_KEY[128:191], and KEY3H[31:0] || KEY3L[31:0] is used as AES_KEY[192:255].

NOTE: "||" is a concatenation operator. For example, X || Y denotes the concatenation of two bit strings X and Y.

CAU_KEY0H

Address offset: 0x20

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
KEY0H[31:16]															
w															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
KEY0H[15:0]															
w															

CAU_KEY0L

Address offset: 0x24

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
KEY0L[31:16]															
w															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
KEY0L[15:0]															
w															

CAU_KEY1H

Address offset: 0x28

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
KEY1H[31:16]															
w															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
KEY1H[15:0]															
w															

CAU_KEY1L

Address offset: 0x2C

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
KEY1L[31:16]															
w															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
KEY1L[15:0]															
w															

CAU_KEY2H

Address offset: 0x30

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
KEY2H[31:16]															
w															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
KEY2H[15:0]															
w															

CAU_KEY2L

Address offset: 0x34

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
KEY2L[31:16]															
w															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
KEY2L[15:0]															
w															

CAU_KEY3H

Address offset: 0x38

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
KEY3H[31:16]															
w															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
KEY3H[15:0]															
w															

CAU_KEY3L

Address offset: 0x3C

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
KEY3L[31:16]															
w															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
KEY3L[15:0]															
w															

Bits	Fields	Descriptions
31:0	KEY0...3(H / L)	The key for DES, TDES, AES

10.9.10. Initial vector registers (CAU_IV0..1(H/L))

Address offset: 0x40 to 0x4C

Reset value: 0x0000 0000

This registers have to be accessed by word (32-bit), and all of them must be written when BUSY is 0.

In DES/TDES mode, IV0H is the leftmost bits, and IV0L is the rightmost bits of the initialization vectors.

In AES mode, IV0H is the leftmost bits, and IV1L is the rightmost bits of the initialization vectors.

CAU_IV0H

Address offset: 0x40

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IV0H[31:16]															
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IV0H[15:0]															
rw															

CAU_IV0L

Address offset: 0x44

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IV0L[31:16]															
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IV0L[15:0]															
rw															

CAU_IV1H

Address offset: 0x48

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IV1H[31:16]															
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IV1H[15:0]															
rw															

CAU_IV1L

Address offset: 0x4C

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IV1L[31:16]															
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IV1L[15:0]															
rw															

Bits	Fields	Descriptions
31:0	IV0...1(H / L)	The initialization vector for DES, TDES, AES

10.9.11. GCM or CCM mode context switch register x (CAU_GCMCCMCTXSx) (x=0..7)

Address offset: 0x50 to 0x6C

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
CTXx[31:16]															
rw															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CTXx[15:0]															
rw															

Bits	Fields	Descriptions
31:0	CTXx[31:0]	The internal status of the CAU core. Read and save the register data when a high-priority task is coming to be processed, and restore the saved data back to the registers to resume the suspended processing. **Note:** These registers are used only when GCM, GMAC or CCM mode is selected.

10.9.12. GCM mode context switch register x (CAU_GCMCTXSx) (x=0..7)

Address offset: 0x70 to 0x8C

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
CTXx[31:16]															
rw															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CTXx[15:0]															
rw															

Bits	Fields	Descriptions
31:0	CTXx[31:0]	The internal status of the CAU core. Read and save the register data when a high-priority task is coming to be processed, and restore the saved data back to the registers to resume the suspended processing. **Note:** These registers are used only when GCM or GMAC mode is selected.

11. Direct memory access controller (DMA)

11.1. 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 7 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.

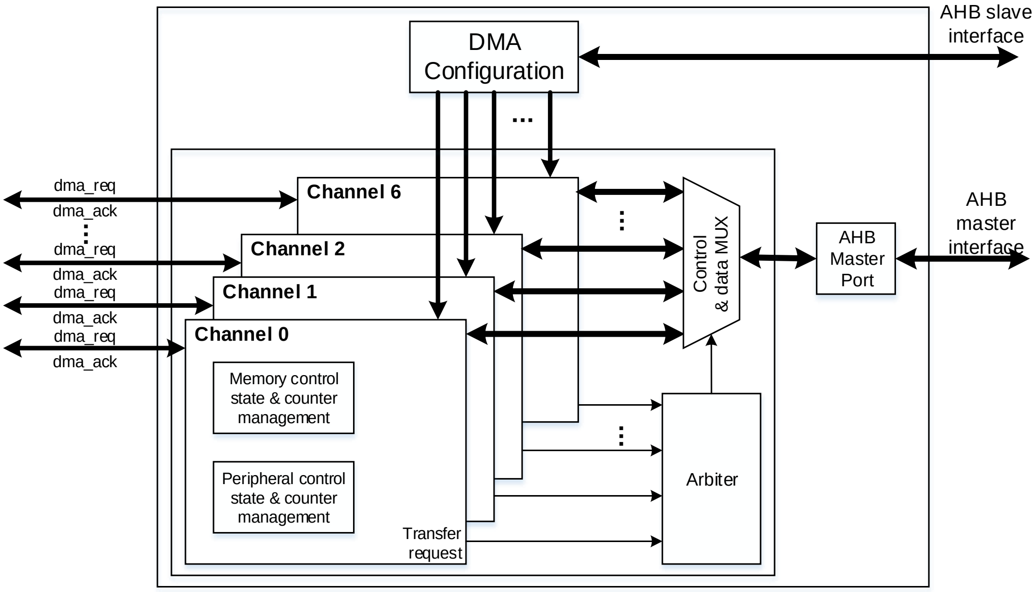
The system bus is shared by the DMA controller and the Cortex®-M23 core. When the DMA and the CPU are targeting the same destination, the DMA access may stop the CPU access to the system bus for some bus cycles. Round-robin scheduling is implemented in the bus matrix to ensure at least half of the system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

11.2. Characteristics

- Programmable length of data to be transferred, max to 65536.
- 7 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 6 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.

11.3. Block diagram

Figure 11-1. Block diagram of DMA



As shown in ***Figure 11-1. 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.

11.4. Function overview

11.4.1. 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 11-1. DMA transfer operation***.

Table 11-1. 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.

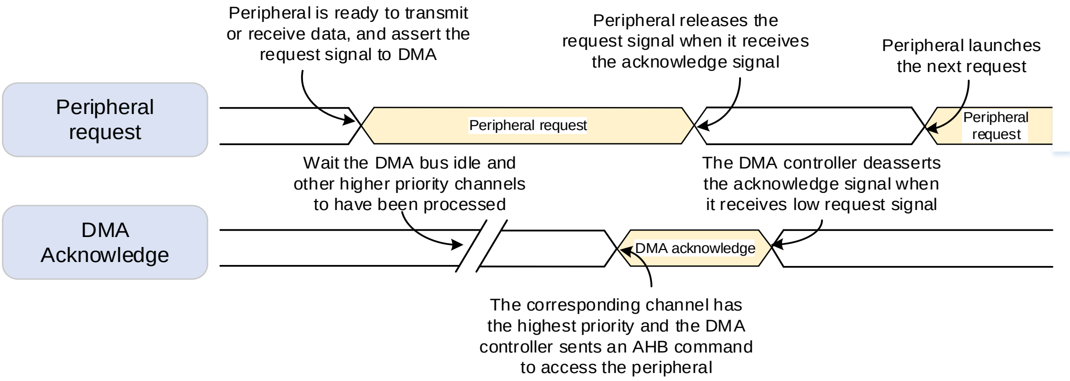
11.4.2. Peripheral handshake

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 11-2. Handshake mechanism shows how the handshake mechanism works between the DMA controller and peripherals.

Figure 11-2. Handshake mechanism



11.4.3. 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.

11.4.4. 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.

11.4.5. 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.

11.4.6. 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.

11.4.7. 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.

11.4.8. 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.

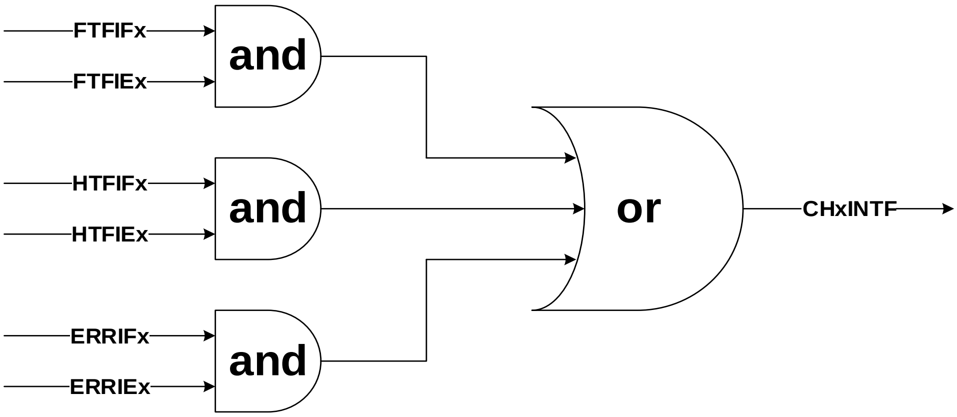
Each interrupt event has a dedicated flag bit in the DMA_INTF register, a dedicated clear bit in the DMA_INTC register, and a dedicated enable bit in the DMA_CHxCTL register. The relationship is described in the following ***Table 11-2. interrupt events***.

Table 11-2. 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 11-3. DMA interrupt logic***, an interrupt can be produced when any type of interrupt event occurs and enabled on the channel.

Figure 11-3. DMA interrupt logic



Note: "x" indicates channel number (x = 0…6).

11.4.9. 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 12-2. Request multiplexer input mapping***.

11.5. Register definition

DMA base address: 0x4002 0000

11.5.1. 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				ERRIF6	HTFIF6	FTFIF6	GIF6	ERRIF5	HTFIF5	FTFIF5	GIF5	ERRIF4	HTFIF4	FTFIF4	GIF4
				r	r	r	r	r	r	r	r	r	r	r	r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ERRIF3	HTFIF3	FTFIF3	GIF3	ERRIF2	HTFIF2	FTFIF2	GIF2	ERRIF1	HTFIF1	FTFIF1	GIF1	ERRIF0	HTFIF0	FTFIF0	GIF0
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

Bits	Fields	Descriptions
31:20	Reserved	Must be kept at reset value.
27 / 23 / 19 / 15 / 11 / 7 / 3	ERRIFx	Error flag of channel x (x = 0…6) 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
26 / 22 / 18 / 14 / 10 / 6 / 2	HTFIFx	Half transfer finish flag of channel x (x = 0…6) 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
25 / 21 / 17 / 13 / 9 / 5 / 1	FTFIFx	Full Transfer finish flag of channel x (x = 0…6) 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
24 / 20 / 16 / 12 / 8 / 4 / 0	GIFx	Global interrupt flag of channel x (x = 0…6) 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

11.5.2. 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				ERRIFC6	HTFIFC6	FTFIFC6	GIFC6	ERRIFC5	HTFIFC5	FTFIFC5	GIFC5	ERRIFC4	HTFIFC4	FTFIFC4	GIFC4
				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
ERRIFC3	HTFIFC3	FTFIFC3	GIFC3	ERRIFC2	HTFIC2	FTFIFC2	GIFC2	ERRIFC1	HTFIFC1	FTFIFC1	GIFC1	ERRIFC0	HTFIFC0	FTFIFC0	GIFC0
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

Bits	Fields	Descriptions
31:20	Reserved	Must be kept at reset value.
27 / 23 / 19 / 15 / 11 / 7 / 3	ERRIFCx	Clear bit for error flag of channel x (x = 0…6) 0: No effect 1: Clear error flag
26 / 22 / 18 / 14 / 10 / 6 / 2	HTFIFCx	Clear bit for half transfer finish flag of channel x (x = 0…6) 0: No effect 1: Clear half transfer finish flag
25 / 21 / 17 / 13 / 9 / 5 / 1	FTFIFCx	Clear bit for full transfer finish flag of channel x (x = 0…6) 0: No effect 1: Clear full transfer finish flag
24 / 20 / 16 / 12 / 8 / 4 / 0	GIFCx	Clear global interrupt flag of channel x (x = 0…6) 0: No effect 1: Clear GIFx, ERRIFx, HTFIFx and FTFIFx bits in the DMA_INTF register

11.5.3. Channel x control register (DMA_CHxCTL)

x = 0...6, 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 can 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 can 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 can 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 can 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 can 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

11.5.4. Channel x counter register (DMA_CHxCNT)

x = 0...6, 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.

11.5.5. Channel x peripheral base address register (DMA_CHxPADDR)

x = 0...6, 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.

11.5.6. Channel x memory base address register (DMA_CHxMADDR)

x = 0...6, 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.

12. DMA request multiplexer (DMAMUX)

12.1. Overview

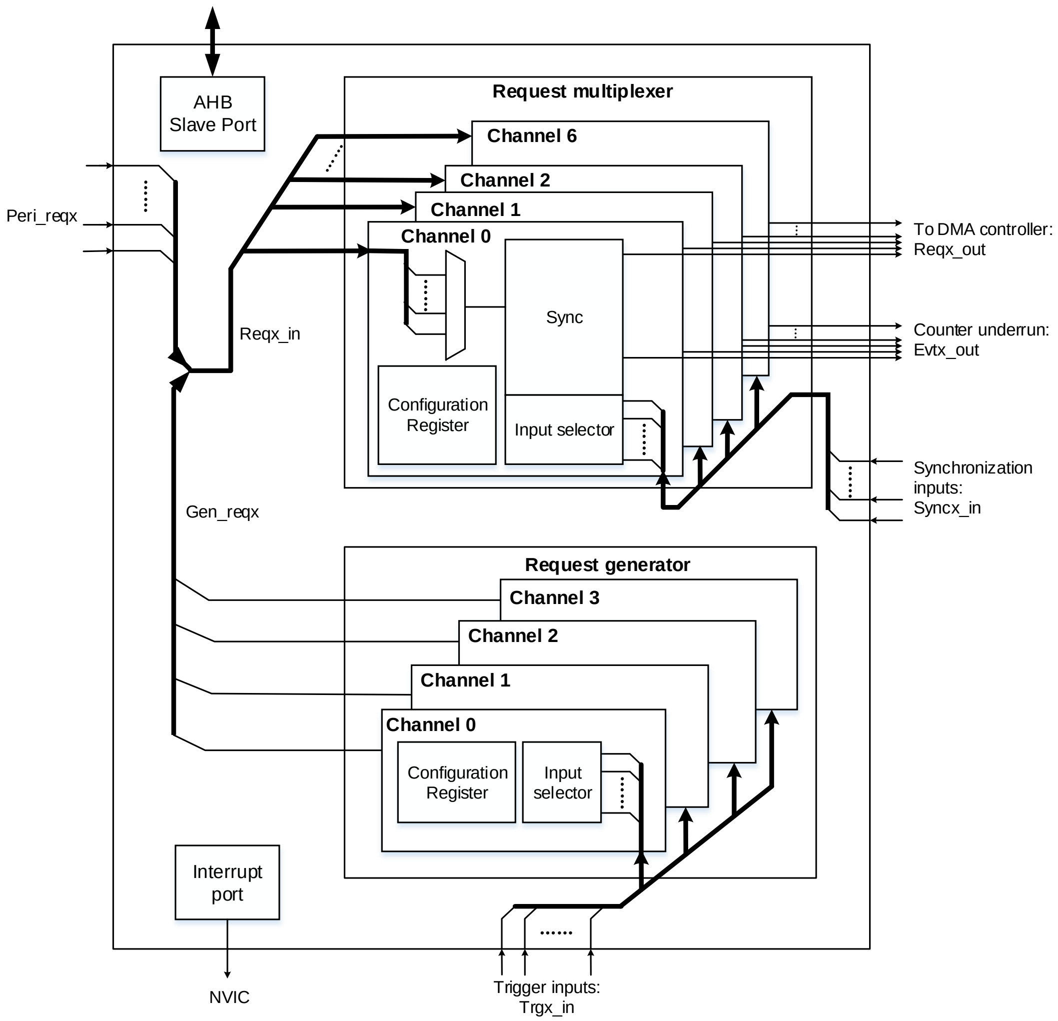
DMAMUX is a transmission scheduler for DMA requests. The DMAMUX request multiplexer is used for routing a DMA request line between the peripherals / generated DMA request (from the DMAMUX request generator) and the DMA controller. Each DMAMUX request multiplexer channel selects a unique DMA request line, 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).

12.2. Characteristics

- 7 channels for DMAMUX request multiplexer.
- 4 channels for DMAMUX request generator.
- Support 21 trigger inputs.
- Support 21 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 60 input DMA request lines from peripherals, 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.

12.3. Block diagram

Figure 12-1. Block diagram of DMAMUX



12.4. 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.

12.5. Function overview

As shown in ***Figure 12-1. 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.

12.5.1. 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 12-2. 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 MUXID[5:0] / MUXID[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 MUXID[5:0] / MUXID[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.

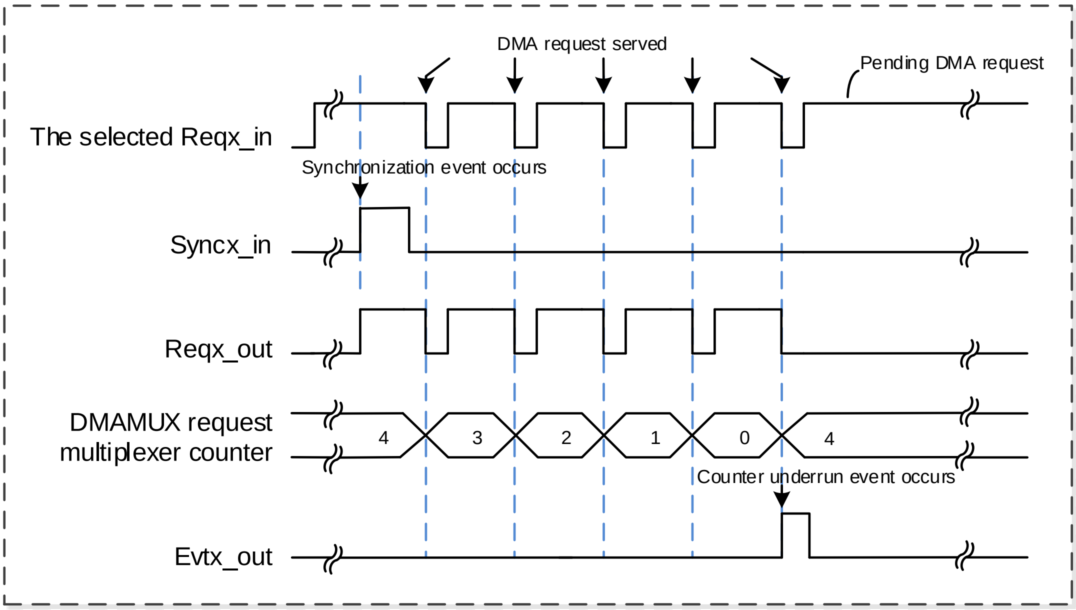
Note: The NBR[4:0] bits value shall only be written by software when both synchronization enable bit SYNCEN and event generation enable EVGEN bit of the corresponding request multiplexer channel x are disabled.

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 12-2. Synchronization mode shows an example when NBR[4:0] = 4, SYNCEN = 1, EVGEN = 1, SYNCP[1:0] = 01.

Figure 12-2. Synchronization mode



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 12-5. Synchronization input mapping***. The synchronization input valid edge is configured by the SYNCP[1:0] bits of the DMAMUX_RM_CHxCFG register.

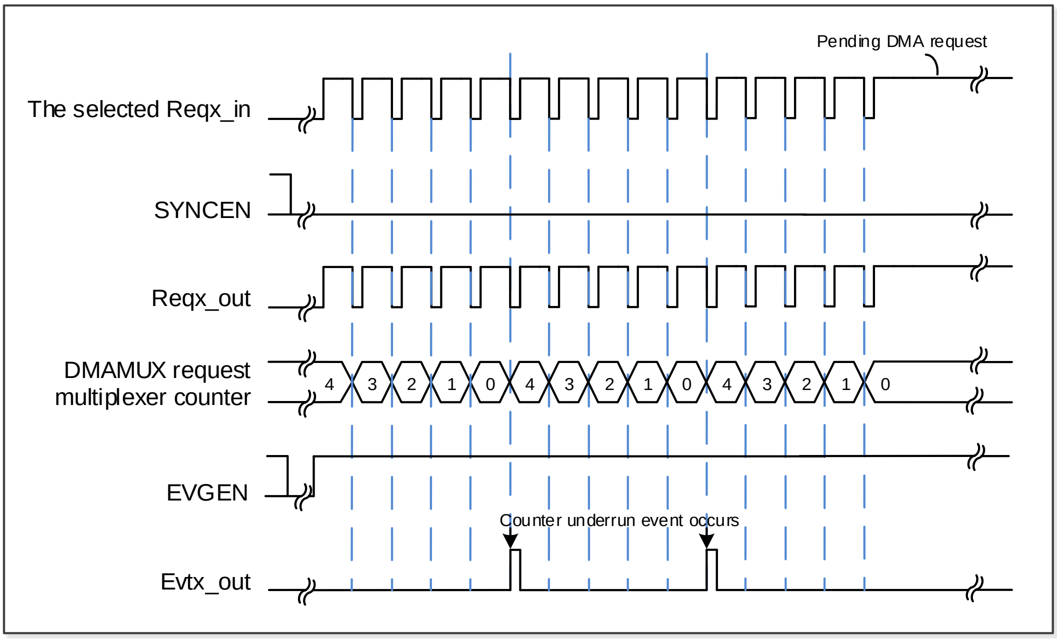
Note: If a synchronization input event occurs when there is no pending selected input DMA request line, the input event is discarded. The following asserted input request lines will not be routed to the DMAMUX multiplexer channel output until a synchronization input event occurs again.

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 12-3. Event generation shows an example when NBR[4:0] = 4, SYNCEN = 0, EVGEN = 1.

Figure 12-3. 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.

12.5.2. 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 12-4. 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.

12.5.3. 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.

12.5.4. 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 12-1. Interrupt events***.

Table 12-1. 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.

12.5.5. 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 12-2. Request multiplexer input mapping for GD32L233x*** and ***Table 12-3. Request multiplexer input mapping for*** , configured by the MUXID[5:0] /

MUXID[6:0] bits in the DMAMUX_RM_CHxCFG register for the DMAMUX request multiplexer channel x.

Table 12-2. Request multiplexer input mapping for GD32L233x

Request multiplexer channel input identification MUXID[5:0]	Source
1	Gen_req0
2	Gen_req1
3	Gen_req2
4	Gen_req3
5	ADC
6	DAC
7	Reserved
8	Reserved
9	Reserved
10	I2C0_RX
11	I2C0_TX
12	I2C1_RX
13	I2C1_TX
14	I2C2_RX
15	I2C2_TX
16	SPI0_RX
17	SPI0_TX
18	SPI1_RX
19	SPI1_TX
20	Reserved
21	Reserved
22	Reserved
23	Reserved
24	Reserved
25	TIMER1_CH0
26	TIMER1_CH1
27	TIMER1_CH2
28	TIMER1_CH3
29	Reserved
30	TIMER1_UP
31	Reserved
32	TIMER2_CH0
33	TIMER2_CH1
34	TIMER2_CH2
35	TIMER2_CH3
36	TIMER2_TRIG

Request multiplexer channel input identification MUXID[5:0]	Source
37	TIMER2_UP
38	Reserved
39	Reserved
40	Reserved
41	Reserved
42	TIMER5_UP
43	TIMER6_UP
44	CAU_IN
45	CAU_OUT
46	Reserved
47	Reserved
48	Reserved
49	Reserved
50	USART0_RX
51	USART0_TX
52	USART1_RX
53	USART1_TX
54	UART3_RX
55	UART3_TX
56	UART4_RX
57	UART4_TX
58	LPUART_RX
59	LPUART_TX
60	Reserved
61	Reserved
62	Reserved
63	Reserved

Table 12-3. Request multiplexer input mapping for GD32L235xx

Request multiplexer channel input identification MUXID[6:0]	Source
1	Gen_reqx0
2	Gen_reqx1
3	Gen_reqx2
4	Gen_reqx3
5	ADC
6	DAC_CH0
7	Reserved
8	Reserved

Request multiplexer channel input identification MUXID[6:0]	Source
9	Reserved
10	I2C0_RX
11	I2C0_TX
12	I2C1_RX
13	I2C1_TX
14	I2C2_RX
15	I2C2_TX
16	SPI0_RX
17	SPI0_TX
18	SPI1_RX
19	SPI1_TX
20	Reserved
21	Reserved
22	Reserved
23	Reserved
24	Reserved
25	TIMER1_CH0
26	TIMER1_CH1
27	TIMER1_CH2
28	TIMER1_CH3
29	TIMER1_TRIG
30	TIMER1_UP
31	Reserved
32	TIMER2_CH0
33	TIMER2_CH1
34	TIMER2_CH2
35	TIMER2_CH3
36	TIMER2_TRIG
37	TIMER2_UP
38	Reserved
39	Reserved
40	Reserved
41	Reserved
42	TIMER5_UP
43	TIMER6_UP
44	CAU_IN
45	CAU_OUT
46	Reserved
47	Reserved

Request multiplexer channel input identification MUXID[6:0]	Source
48	Reserved
49	Reserved
50	USART0_RX
51	USART0_TX
52	USART1_RX
53	USART1_TX
54	UART3_RX
55	UART3_TX
56	UART4_RX
57	UART4_TX
58	LPUART0_RX
59	LPUART0_TX
60	LPUART1_RX
61	LPUART1_TX
62	Reserved
63	Reserved
64	TIMER0_CH0
65	TIMER0_CH1
66	TIMER0_CH2
67	TIMER0_CH3
68	TIMER0_TRIG
69	TIMER0_UP
70	TIMER0_COM
71	TIMER14_CH0
72	TIMER14_CH1
73	TIMER14_TRIG
74	TIMER14_UP
75	TIMER14_COM
76	TIMER40_CH0
77	TIMER40_CH1
78	TIMER40_TRIG
79	TIMER40_UP
80	TIMER40_COM

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 12-4. Trigger input mapping***.

Table 12-4. 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
15	EXTI_15
16	Evt0_out
17	Evt1_out
18	Evt2_out
19	Evt3_out
20	Reserved
21	Reserved
22	TIMER11_CH0_O
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 12-5. Synchronization input mapping***.

Table 12-5. 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

Synchronization input identification SYNCID[4:0]	Source
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	Evt0_out
17	Evt1_out
18	Evt2_out
19	Evt3_out
20	Reserved
21	Reserved
22	TIMER11_CH0_O
23	Reserved

12.6. Register definition

DMAMUX base address: 0x4002 0800

12.6.1. Request multiplexer channel x configuration register (DMAMUX_RM_CHxCFG)

For GD32L233xx devices

x = 0...6, 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		MUXID[5:0]					
						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 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
7:6	Reserved	Must be kept at reset value.
5:0	MUXID[5:0]	Multiplexer input identification Selects the input DMA request in multiplexer input sources.

For GD32L235xx devices

x = 0...6, 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	MUXID[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 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	MUXID[6:0]	Multiplexer input identification Selects the input DMA request in multiplexer input sources.

12.6.2. 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									SOIF6	SOIF5	SOIF4	SOIF3	SOIF2	SOIF1	SOIF0
									r	r	r	r	r	r	r

Bits	Fields	Descriptions
31:7	Reserved	Must be kept at reset value.
6	SOIF6	Synchronization overrun event flag of request multiplexer channel 6 Refers to SOIF0 descriptions.
5	SOIF5	Synchronization overrun event flag of request multiplexer channel 5 Refers to SOIF0 descriptions.
4	SOIF4	Synchronization overrun event flag of request multiplexer channel 4 Refers to SOIF0 descriptions.
3	SOIF3	Synchronization overrun event flag of request multiplexer channel 3 Refers to SOIF0 descriptions.
2	SOIF2	Synchronization overrun event flag of request multiplexer channel 2 Refers to SOIF0 descriptions.
1	SOIF1	Synchronization overrun event flag of request multiplexer channel 1

		Refers to SOIF0 descriptions.
0	SOIF0	Synchronization overrun event flag of request multiplexer channel 0 If a synchronization event occurs when the DMAMUX request counter value is lower than NBR[4:0], the flag is set. It is cleared by writing 1 to the corresponding SOIFC0 bit in DMAMUX_RM_INTC register.

12.6.3. 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									SOIFC6	SOIFC5	SOIFC4	SOIFC3	SOIFC2	SOIFC1	SOIFC0
									w	w	w	w	w	w	w

Bits	Fields	Descriptions
31:7	Reserved	Must be kept at reset value.
6	SOIFC6	Clear bit for synchronization overrun event flag of request multiplexer channel 6 Refers to SOIFC0 descriptions.
5	SOIFC5	Clear bit for synchronization overrun event flag of request multiplexer channel 5 Refers to SOIFC0 descriptions.
4	SOIFC4	Clear bit for synchronization overrun event flag of request multiplexer channel 4 Refers to SOIFC0 descriptions.
3	SOIFC3	Clear bit for synchronization overrun event flag of request multiplexer channel 3 Refers to SOIFC0 descriptions.
2	SOIFC2	Clear bit for synchronization overrun event flag of request multiplexer channel 2 Refers to SOIFC0 descriptions.
1	SOIFC1	Clear bit for synchronization overrun event flag of request multiplexer channel 1 Refers to SOIFC0 descriptions.
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.

12.6.4. 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]	DMAMUX request generator trigger polarity 00: No event trigger detection 01: Rising edge 10: Falling edge 11: Rising and falling edges
16	RGEN	DMAMUX request generator channel x enable 0: Disable DMAMUX request generator channel x 1: Enable DMAMUX 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.

12.6.5. Request generator channel 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	r

Bits	Fields	Descriptions
31:4	Reserved	Must be kept at reset value.
3	TOIF3	Trigger overrun event flag of request generator channel 3 Refers to TOIF0 descriptions.
2	TOIF2	Trigger overrun event flag of request generator channel 2 Refers to TOIF0 descriptions.
1	TOIF1	Trigger overrun event flag of request generator channel 1 Refers to TOIF0 descriptions.
0	TOIF0	Trigger overrun event flag of request generator channel 0 If a new trigger event occurs before the request generator counter underrun, the flag is set. It is cleared by writing 1 to the corresponding TOIFC0 bit in the DMAMUX_RG_INTC register.

12.6.6. Rquest generator channel 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 Refers to TOIFC0 descriptions.
2	TOIFC2	Clear bit for trigger overrun event flag of request generator channel 2 Refers to TOIFC0 descriptions.
1	TOIFC1	Clear bit for trigger overrun event flag of request generator channel 1 Refers to TOIFC0 descriptions.
0	TOIFC0	Clear bit for trigger overrun event flag of request generator channel 0 Writing 1 clears the corresponding trigger overrun flag TOIF0 in the DMAMUX_RG_INTF register.

13. Debug (DBG)

13.1. Overview

The GD32L23x 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, LPTIMER, I2C, CAN, RTC, WWDGT, and FWDGT. When corresponding bit is set, it provides a clock in power saving mode or holds the state for TIMER, LPTIMER, I2C, CAN, RTC, WWDGT and FWDGT.

13.2. SW function overview

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

13.2.1. 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-pin can be released to other GPIO functions. Please refer to ***General-purpose and alternate-function I/Os (GPIO and AFIO)***.

13.3. Debug hold function overview

13.3.1. Debug support for power saving 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_IRC16M, and the debugger can debug in standby mode. When exiting the standby mode, a system reset generated.

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_IRC16M, 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.

13.3.2. Debug support for TIMER, LPTIMER, 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 and LPTIMER, 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.

13.4. Register definition

DBG base address: 0x4001 5800

13.4.1. 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.

13.4.2. Control register 0 (DBG_CTL0)

For GD32L233xx devices

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					TIMER11_HOLD	Reserved		TIMER8_HOLD	Reserved	TIMER6_HOLD	TIMER5_HOLD	Reserved			I2C1_HOLD
					rw			rw		rw	rw				rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
I2C0_HOLD	Reserved	TIMER2_HOLD	TIMER1_HOLD	Reserved		WWDGT_HOLD	FWDGT_HOLD	Reserved					STB_HOLD	DSLP_HOLD	SLP_HOLD
rw		rw	rw			rw	rw						rw	rw	rw

Bits	Fields	Descriptions
31:27	Reserved	Must be kept at reset value.
26	TIMER11_HOLD	TIMER 11 hold bit This bit is set and reset by software. 0: no effect 1: hold the TIMER 11 counter for debugging when the core is halted.
25:24	Reserved	Must be kept at reset value.

23	TIMER8_HOLD	TIMER 8 hold bit This bit is set and reset by software. 0: no effect 1: hold the TIMER 8 counter for debugging when the core is halted.
22	Reserved	Must be kept at reset value.
21	TIMER6_HOLD	TIMER 6 hold bit This bit is set and reset by software. 0: no effect 1: hold the TIMER 6 counter for debugging when the core is halted.
20	TIMER5_HOLD	TIMER 5 hold bit This bit is set and reset by software. 0: no effect 1: hold the TIMER 5 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	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.
12	TIMER1_HOLD	TIMER 1 hold bit This bit is set and reset by software. 0: no effect 1: hold the TIMER 1 counter for debugging when the core is halted.
11:10	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_IRC16M, 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_IRC16M.
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.

For GD32L235xx devices

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.			TIMER40_HOLD	TIMER14_HOLD	TIMER11_HOLD	Reserved		TIMER8_HOLD	Reserved	TIMER6_HOLD	TIMER5_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	CAN_HOLD	TIMER2_HOLD	TIMER1_HOLD	TIMER0_HOLD	Reserved	WWDGT_HOLD	FWDGT_HOLD	Reserved					STB_HOLD	DSLP_HOLD	SLP_HOLD
rw			rw		rw	rw	rw						rw	rw	rw

Bits	Fields	Descriptions
31:29	Reserved	Must be kept at reset value.
28	TIMER40_HOLD	TIMER 40 hold bit This bit is set and reset by software. 0: no effect 1: hold the TIMER 40 counter for debugging when the core is halted.
27	TIMER14_HOLD	TIMER 14 hold bit This bit is set and reset by software. 0: no effect

		1: hold the TIMER 14 counter for debugging when the core is halted.
26	TIMER11_HOLD	TIMER 11 hold bit This bit is set and reset by software. 0: no effect 1: hold the TIMER 11 counter for debugging when the core is halted.
25:24	Reserved	Must be kept at reset value.
23	TIMER8_HOLD	TIMER 8 hold bit This bit is set and reset by software. 0: no effect 1: hold the TIMER 8 counter for debugging when the core is halted.
22	Reserved	Must be kept at reset value.
21	TIMER6_HOLD	TIMER 6 hold bit This bit is set and reset by software. 0: no effect 1: hold the TIMER 6 counter for debugging when the core is halted.
20	TIMER5_HOLD	TIMER 5 hold bit This bit is set and reset by software. 0: no effect 1: hold the TIMER 5 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 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	CAN_HOLD	CAN hold bit This bit is set and reset by software. 0: no effect 1: Hold the CAN counter for debug when core halted.
13	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.
12	TIMER1_HOLD	TIMER 1 hold bit

		This bit is set and reset by software. 0: no effect 1: hold the TIMER 1 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_IRC16M, 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_IRC16M.
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.

13.4.3. Control register 1 (DBG_CTL1)

For GD32L233xx devices

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.														I2C2_HOLD	LPTIMER_HOLD
														rw	rw

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

Bits	Fields	Descriptions
31:18	Reserved	Must be kept at reset value.
17	I2C2_HOLD	I2C2 hold bit This bit is set and reset by software. 0: no effect 1: hold the I2C2 status to avoid SMBUS timeout for debugging when the core is halted.
16	LPTIMER_HOLD	LPTIMER hold bit This bit is set and reset by software. 0: no effect 1: hold the LPTIMER counter for debugging when the core is halted.
15:11	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.

For GD32L235xx devices

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.													LPTIMER1_HOLD	I2C2_HOLD	LPTIMER0_HOLD
														rw	rw

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

Bits	Fields	Descriptions
31:19	Reserved	Must be kept at reset value.
18	LPTIMER1_HOLD	LPTIMER1 hold bit This bit is set and reset by software.

		0: no effect 1: hold the LPTIMER1 counter for debugging when the core is halted.
17	I2C2_HOLD	I2C2 hold bit This bit is set and reset by software. 0: no effect 1: hold the I2C2 status to avoid SMBUS timeout for debugging when the core is halted.
16	LPTIMER0_HOLD	LPTIMER0 hold bit This bit is set and reset by software. 0: no effect 1: hold the LPTIMER0 counter for debugging when the core is halted.
15:11	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.

14. Analog to digital converter (ADC)

14.1. Overview

A 12-bit successive approximation analog-to-digital converter module(ADC) is integrated on the MCU chip, which can sample analog signals from 16 external channels and 4 internal channels. The 20 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 alignment or the most significant bit alignment. An on-chip hardware oversample scheme improves performances and reduces the computational burden of MCU.

14.2. Characteristics

- High performance:
  - ADC sampling rsolution: 12-bit, 10-bit,8-bit, or 6-bit.
  - Foreground calibration function.
  - Programmable sampling time.
  - Data storage mode: the most significant bit and the least significant bit.
  - DMA support.
- Dual clock domain architecture (APB clock and ADC clock).
- Analog input channels:
  - 16 external analog inputs.
  - 1 channel for internal temperature sensor (V_{SENSE}).
  - 1 channel for internal reference voltage (V_{REFINT}).
  - 1 channel for monitoring external V_{BAT} power supply pin.
  - 1 channel for monitoring SLCD voltage (V_{SLCD}).
- 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 the selected inputs once for 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 conversions.
  - Analog watchdog event.
- Module supply requirements: 1.8V to 3.6V, and typical power supply voltage is 3.3V.
- Oversampling:

– 16-bit data register.
– Oversampling ratio adjustable from 2x to 256x.
– Programmable data shift up to 8-bits.

- Channel input range: V_{SSA}/ V_{SS} ≤ V_{IN} ≤ V_{DDA}/ V_{DD}.

14.3. Pins and internal signals

Figure 14-1. ADC module block diagram shows the ADC block diagram. *Table 14-1. ADC internal input signals* and *Table 14-2. ADC input pins definition* give the ADC internal signals and pins description.

Table 14-1. ADC internal input signals

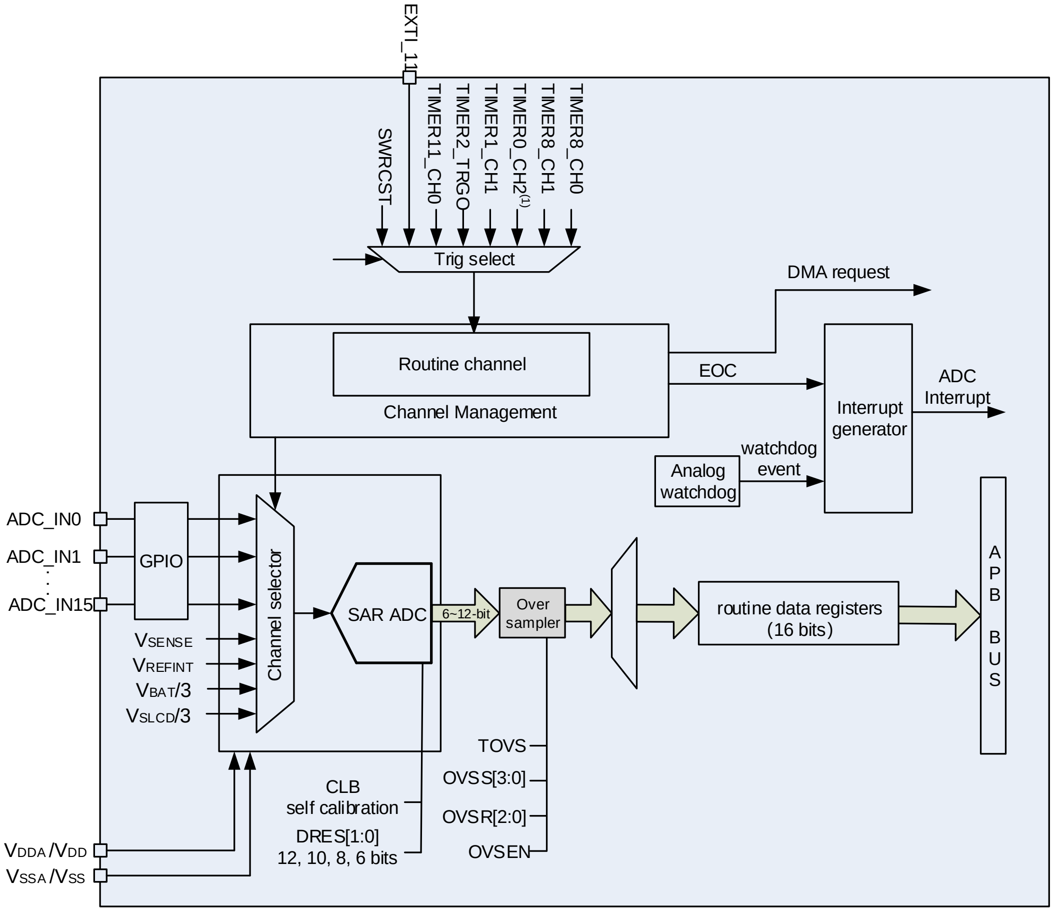
Internal signal name	Description
V_{SENSE}	Internal temperature sensor output voltage
V_{REFINT}	Internal voltage reference output voltage
V_{BAT}	V_{BAT} pin input voltage divided by 3
V_{SLCD}	V_{SLCD} pin input voltage divided by 3

Table 14-2. ADC input pins definition

Name	Description
V_{DDA}/ V_{DD}	Analog power supply equals to V_{DD} and 1.8V ≤V_{DDA}≤ 3.6 V
V_{SSA}/ V_{SS}	Ground for analog power supply equals to V_{SS}
ADCx_IN [15:0]	Up to 16 external channels

14.4. Function overview

Figure 14-1. ADC module block diagram



(1) The trigger source is only available for GD32L235xx devices. In GD32L233xx series products, the corresponding value is reserved.

14.4.1. Foreground calibration function

During the foreground calibration procedure, the ADC calculates a calibration factor which is internally applied to the ADC until the next ADC power-off. The application must not use the ADC during calibration and must wait until it is completed. Calibration should be performed before starting A/D conversion. The calibration is initiated by setting the CLB bit to 1. CLB bit stays at 1 during the calibration sequence. It is then cleared by hardware as soon as the calibration is completed.

When the ADC operating conditions change (such as supply power voltage V_{DDA}, temperature and so on), it is recommended to re-run a calibration cycle.

The internal analog calibration can be reset by setting the RSTCLB bit in ADC_CTL1 register.

Calibration procedure by software:

1. Ensure that ADCON=1.

2. Delay 14 CK_ADC to wait for ADC stability.
3. Set RSTCLB (optional).
4. Set CLB=1.
5. Wait for CLB =0.

14.4.2. Dual clock domain architecture

The ADC sub-module, with exception of the APB interface block, is feed by an ADC clock, which can be asynchronous and independent from the APB clock.

Application can reduce PLCK frequency for low power operation while still keeping optimum ADC performance.

Refer to RCU Section ***4.2.1*** for more information on generating this clock source.

14.4.3. 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 be put into power off mode. After ADC is enabled, you need delay $t_{ST(ADC)}$ time for sampling, the value of $t_{ST(ADC)}$ please refer to the chip datasheet.

Note: When the ADC uses the internal voltage reference VREF (VREFEN=1, HIPM=0), ensure that VREFRDY bit in the VREF_CS register is set before enabling the ADC.

14.4.4. Single-ended and differential input channels

In GD32L23xx series, this function is only available for GD32L235xx devices. By writing into bits DIFCTL[14:0] in the ADC_DIFCTL register, the user can configure channels as differential input or single-ended input,and the ADC must be disabled (ADCON =0) when the user configurate these bits.

The channel n voltage is the difference between positive input and negative input. The positive input is external voltage V_{INn}, and there is a difference of the negative input between single-ended mode and differential input mode. In single-ended input mode, the negative input is V_{REFN}, in differential input mode, the negative input is $V_{IN(n+1)}$. And therefore, channel (n+1) is no longer usable in single-ended mode or in differential mode and must never be configured to be converted.

Channel 15, 16, 17, 18 and 19 are forced to single-ended configuration (corresponding bits DIFCTL[n] is always zero), because they are connected to internal channels.

When the channel is used in differential input mode, the input voltages should be differential signals (common mode voltage is V_{REFP}/ 2), and the input ranges are still (V_{REFN} ~ V_{REFP}).

Taking the right-aligned, 12-bit resolution as an example,

1) When V_{INn} is V_{REFP} and $V_{IN(n+1)}$ is V_{REFN}, the conversion result of channel n is 0x0FFF;

2) When V_{INn} is V_{REFN} and $V_{IN(n+1)}$ is V_{REFP}, The conversion result of channel n is 0x0000;
3) When V_{INn} is V_{REFP}/ 2 and $V_{IN(n+1)}$ is V_{REFP}/ 2, the conversion result of channel n is 0x07FF.

D_{out} is the conversion result of channel n, then the differential voltage is：

$$V_{INn}-V_{IN(n+1)} = V_{REFP}*(2*D_{out}/4095-1) \quad (14-1)$$

14.4.5. 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, and 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.

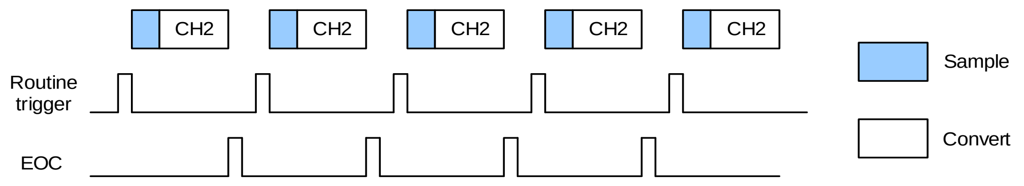
Note: Although the ADC supports 20 multiplexed channels, the maximum length of the sequence is only 16.

14.4.6. Operation modes

Single operation mode

In the single operation mode, the ADC performs conversion on the channel specified in the RSQ0[4:0] bits in 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 14-2. Single operation mode



After the 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:

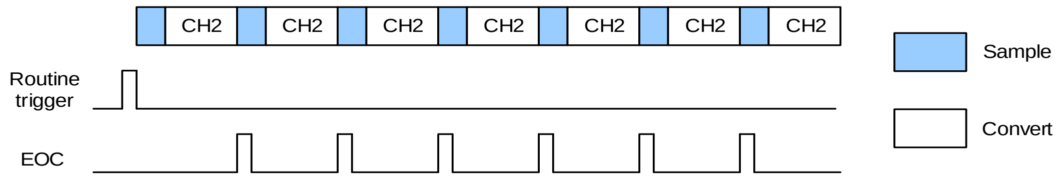
1. Make sure the DISRC, SM bits in the ADC_CTL0 register and CTN bit in the ADC_CTL1 register are reset.
2. Configure the RSQ0 with the analog channel number.
3. Configure the ADC_SAMPTx register.
4. Configure the ETERC and ETSRC bits in the ADC_CTL1 register if it is needed.
5. Set the SWRCST bit, or generate an external trigger for the routine sequence.
6. Wait for the EOC flag to be set.

7. Read the converted data from the ADC_RDATA register.
8. Clear the EOC flag by writing 0.

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[4:0]. 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 14-3. 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 with the analog channel number.
3. Configure the ADC_SAMPTx register.
4. Configure the ETERC and ETSRC bits in the ADC_CTL1 register if it is needed.
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.

To get rid of checking, DMA can be used to transfer the converted data:

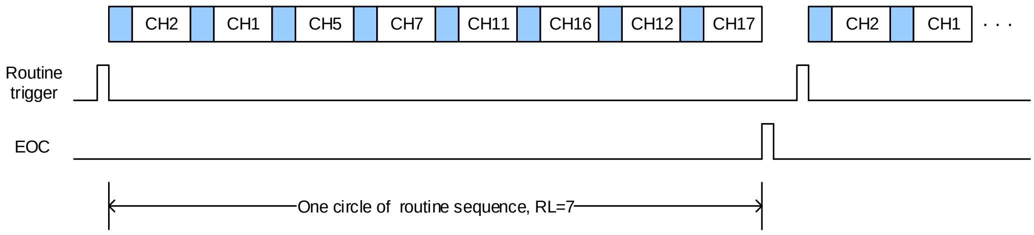
1. Set the CTN and DMA bits in the ADC_CTL1 register.
2. Configure the RSQ0 with the analog channel number.
3. Configure the ADC_SAMPTx register.
4. Configure the ETERC and ETSRC bits in the ADC_CTL1 register if it is needed.
5. Prepare the DMA module to transfer data from the ADC_RDATA.
6. Set the SWRCST 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 all 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 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.

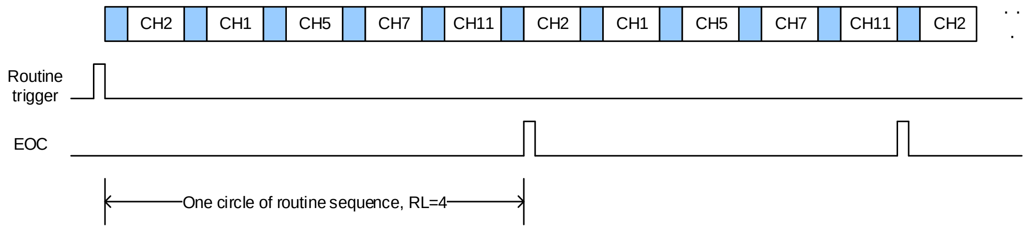
Figure 14-4. Scan operation mode, continuous disable



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 it is needed.
4. Prepare the 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 for the EOC flag to be set.
7. Clear the EOC flag by writing 0.

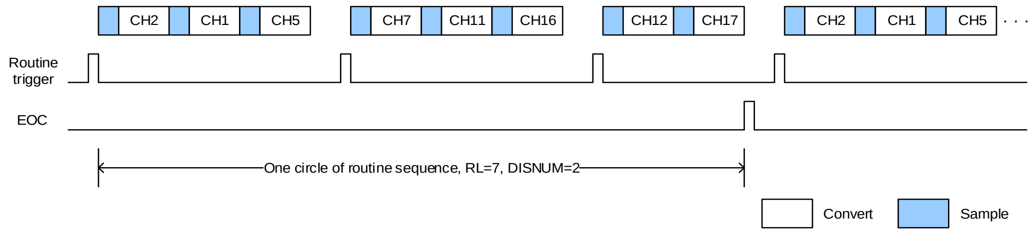
Figure 14-5. Scan operation mode, continuous enable



Discontinuous operation mode

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 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 14-6. 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 it is needed;
5. Prepare the DMA module to transfer data from the ADC_RDATA (refer to the spec of the DMA module);
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.

14.4.7. Conversion result threshold monitor function

The analog watchdog is enabled when the RWDEN 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, and the WDE bit in ADC_STAT register will be set. An interrupt will be generated if the WDEIE bit is set. The ADC_WDHT and ADC_WDLT registers are used to specify the high and low threshold. The comparison is done before the alignment, so the threshold values are 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 RWDEN, WDSC and WDCHSEL [4:0] bits in ADC_CTL0 register, can be monitored by the analog watchdog.

14.4.8. Data storage mode

The alignment of data stored after conversion can be specified by DAL bit in the ADC_CTL1 register.

When the most significant bit, the 12/10/8-bit data are aligned on a half-word, while the 6-bit data are aligned on a byte basis as shown below ***Figure 14-7. Data storage mode of 12-bit resolution***, ***Figure 14-8. Data storage mode of 10-bit resolution***, ***Figure 14-9. Data storage mode of 8-bit resolution*** and ***Figure 14-10. Data storage mode of 6-bit resolution***.

Figure 14-7. 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

Routine channel data

D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	0	0	0	0

DAL=1

Figure 14-8. Data storage mode of 10-bit resolution

Routine channel data

0	0	0	0	0	0	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0

DAL=0

Routine channel data

D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	0	0	0	0	0	0

DAL=1

Figure 14-9. Data storage mode of 8-bit resolution

Routine channel data

0	0	0	0	0	0	0	0	D7	D6	D5	D4	D3	D2	D1	D0

DAL=0

Routine channel data

D7	D6	D5	D4	D3	D2	D1	D0	0	0	0	0	0	0	0	0

DAL=1

Figure 14-10. 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

Routine channel data

0	0	0	0	0	0	0	0	D5	D4	D3	D2	D1	D0	0	0

DAL=1

14.4.9. 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 sampling 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.

Example:

CK_ADC = 16MHz and sampling time is 2.5 cycles, the total conversion time is “2.5+12.5” CK_ADC cycles, that means 0.9375us.

14.4.10. 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 14-3. External trigger source for ADC

ETSRC[2:0]	Trigger Source	Trigger Type
000	TIMER8_CH0	Hardware trigger
001	TIMER8_CH1	
010	TIMER0_CH2(1)	
011	TIMER1_CH1	
100	TIMER2_TRGO	
101	TIMER11_CH0	
110	EXTI_11	
111	SWRCST	Software trigger

(1) The trigger source is only available for GD32L235xx devices. In GD32L233xx series products, the corresponding value is reserved.

14.4.11. 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.

14.4.12. ADC internal channels

When the TSVEN bit in ADC_CTL1 register is set, the temperature sensor channel (ADC_IN16) 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 be set in power down mode by resetting the TSVEN bit.

The output voltage of the temperature sensor changes linearly with temperature. Due to the diversification of production process, the deviation of the temperature change curve is different on chip to chip (refer to the device datasheet for more information).

To use the temperature sensor:

1. Configure the ADC clock (not greater than 5MHz).
2. Configure the conversion sequence (ADC_IN16) and the sampling time (t_{s_temp}) for the channel.
3. Enable the temperature sensor by setting the TSVREN bit in the ADC control register 1 (ADC_CTL1).
4. Start the ADC conversion by setting the ADCON bit or by the triggers.
5. Read the internal temperature sensor output voltage ($V_{temperature}$), and get the temperature with the following equation.
   For GD32L233xx series devices:

$$\text{Temperature (°C)} = ((V_{30} - V_{temperature}) / \text{Avg\_Slope}) + 30 \qquad (14\text{-}2)$$

   For GD32L235xx series devices:

$$\text{Temperature (°C)} = ((V_{25} - V_{temperature}) / \text{Avg\_Slope}) + 25 \qquad (14\text{-}3)$$

   $V_{temperature}$: The output voltage of temperature sensor.

   V_{30} / V_{25}: Internal temperature sensor output voltage at 30°C / 25°C. The ADC conversion results of the temperature sensor corresponding to 30°C / 25°C were calibrated during chip production. The factory-calibrated offset value is stored in the read-only area of flash, and the address please refer to the datasheet.

   Avg_Slope: Average slope for curve between temperatures vs. internal temperature sensor output voltage, the typical value refer to the datasheet.

Note:

1) After the temperature sensor is enabled, it is necessary to wait for at least 3 ADC sampling cycles before the ADC conversion code value is considered valid, and the first 3 conversion data should be discarded;
2) The sampling accuracy of temperature sensor can be improved by means of hardware on-chip over-sampling or software averaging.

When the INREFEN bit in ADC_CTL1 register is set, the V_{REFINT} channel (ADC_IN17) is enabled. The internal reference voltage (V_{REFINT}) provides a stable (bandgap) voltage output for the ADC and comparators. V_{REFINT} is internally connected to the ADC_IN17 input channel.

14.4.13. Battery voltage monitoring

The V_{BAT} channel can be used to measure the backup battery voltage on the V_{BAT} pin. When the VBATEN bit in ADC_CTL1 register is set, V_{BAT} channel (ADC_IN18) is enabled and abridge divider by 3 integrated on the V_{BAT} pin is also enabled automatically with it. As V_{BAT} may be higher than V_{DDA}, this bridge is used to ensure the ADC correct operation. And it connects V_{BAT}/3 to the ADC_IN18 input channel. So, the converted digital value is V_{BAT}/3. In order to prevent unnecessary battery energy consumption, it is recommended that the bridge will be enabled only when it is required.

14.4.14. SLCD voltage monitoring

The V_{SLCD} channel can be used to measure the voltage on the V_{SLCD} pin. When the VSLCDEN bit in ADC_CTL1 register is set, V_{SLCD} channel (ADC_IN19) is enabled and abridge divider by 3 integrated on the V_{SLCD} pin is also enabled automatically with it. As V_{SLCD} may be higher than VDDA, this bridge is used to ensure the ADC correct operation. And it connects V_{SLCD} /3 to the ADC_IN19 input channel. So, the converted digital value is V_{SLCD} /3.

14.4.15. 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.

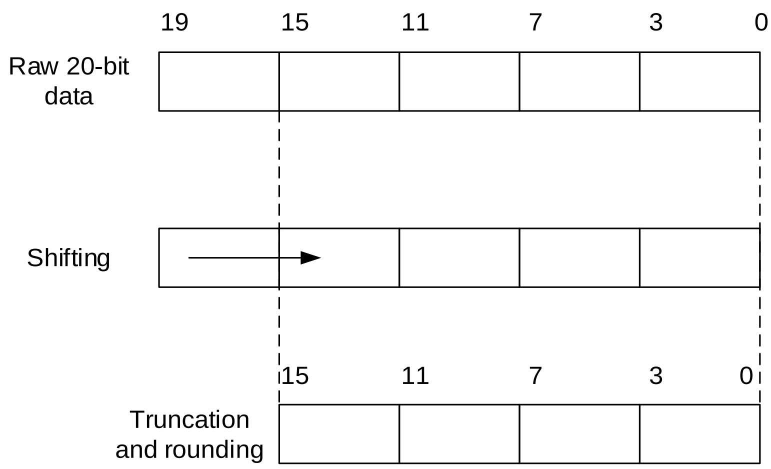
It provides a result with the following form, where N and M can be adjusted, and $D_{out}(n)$ is the n-th output digital signal of the ADC:

$$\text{Result}=\frac{1}{M}*\sum_{n=0}^{N-1} D_{out}(n) \qquad (14\text{-}3)$$

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 bits. 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 14-11. 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 14-12. 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 14-12. A numerical example with 5-bit shifting and rounding

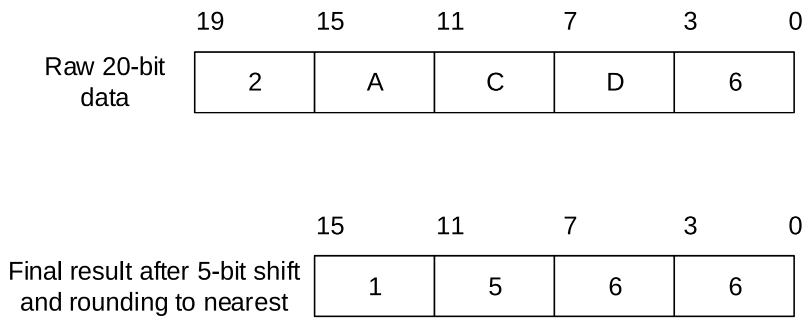


Table 14-4. Maximum output results for N and M combimations (grayed values indicates truncation) below gives the data format for the various N and M combinations, and the raw conversion data equals 0xFFF.

Table 14-4. 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
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:

$$N \times t_{ADC} = N \times (t_{SMPL} + t_{CONV}) \quad (14\text{-}4)$$

Oversampling work with ADC modes

Most of the ADC work modes are available when oversampling is enabled.

- Routine sequence.
- ADC started by software or external triggers.
- Single or scan, continuous or discontinuous operation modes.
- Programmable sample time.
- Analog watchdog.

The oversampling configuration can only be changed when ADCON is reset. Make sure configuring the oversampling before setting ADCON to 1.

14.4.16. 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.Lower resolution reduces the conversion time needed for the successive approximation steps as shown in ***Table 14-5. tCONV timings depending on resolution.***

Table 14-5. t_{CONV} timings depending on resolution

DRES[1:0] bits	t_{CONV} (ADC clock cycles)	t_{CONV}(ns) at f_{ADC}=16MHz	t_{SMPL}(min) (ADC clock cycles)	t_{ADC}(ADC clock cycles)	t_{ADC}(ns) at f_{ADC}=16MHz
12	12.5	781ns	2.5	15	937.5ns
10	10.5	656ns	2.5	13	812.5ns
8	8.5	531ns	2.5	11	687.5ns
6	6.5	406ns	2.5	9	562.5ns

14.4.17. 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.

14.5. Register definition

ADC base address: 0x4001 2400

14.5.1. 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
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved											STRC	Reserved		EOC	WDE
											rc_w0			rc_w0	rc_w0

Bits	Fields	Descriptions
31:5	Reserved	Must be kept at reset value.
4	STRC	Start flag of routine sequence conversion 0: Conversion is not started 1: 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 routine sequence conversion 0: No end of routine sequence conversion 1: End of routine sequence conversion Set by hardware at the end of a routine sequence conversion. Cleared by software writing 0 to it or by reading the ADC_RDATA register.
0	WDE	Analog watchdog event flag 0: No analog watchdog event is not happened 1: Analog watchdog event is happening Set by hardware when the converted voltage crosses the values programmed in the ADC_WDLT and ADC_WDHT registers. Cleared by software writing 0 to it.

14.5.2. Control register 0 (ADC_CTL0)

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						DRES [1:0]		RWDEN	Reserved						
						rw		rw							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DISNUM [2:0]			Reserved	DISRC	Reserved	WDSC	SM	Reserved	WDEIE	EOCIE	WDCHSEL [4:0]				
rw				rw		rw	rw		rw	rw	rw				

Bits	Fields	Descriptions
31:26	Reserved	Must be kept at reset value.
25:24	DRES[1:0]	ADC resolution 00: 12bit 01: 10bit 10: 8bit 11: 6bit
23	RWDEN	Routine channel analog watchdog enable 0: Analog watchdog routine channel disable 1: Analog watchdog routine channel 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 disable 1: Discontinuous operation mode enable
10	Reserved	Must be kept at reset value.
9	WDSC	When in scan mode, analog watchdog is effective on a single channel 0: All channels have analog watchdog function 1: A single channel has analog watchdog function
8	SM	Scan mode 0: Scan operation mode disable 1: Scan operation mode enable
7	Reserved	Must be kept at reset value.
6	WDEIE	Interrupt enable for WDE 0: Interrupt disable 1: Interrupt enable

5	EOCIE	Interrupt enable for EOC 0: Interrupt disable 1: Interrupt enable
4:0	WDCHSEL[4:0]	Analog watchdog channel select 00000: ADC channel 0 00001: ADC channel 1 00010: ADC channel 2 …… 10000: ADC channel16 10001: ADC channel17 10010: ADC channel18 10011: ADC channel19 Other values are reserved. **Note:** ADC analog inputs Channel16, Channel17, Channel 18 and Channel 19 are internally connected to the temperature sensor, V_{REFINT}, V_{BAT} and V_{SLCD} analog inputs.

14.5.3. Control register 1 (ADC_CTL1)

For GD32L233xx devices

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					VSLCDEN	VBATEN	INREFEN	TSVEN	SWRCST	Reserved	ETERC	ETSRC [2:0]			Reserved
					rw	rw	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				RSTCLB	CLB	CTN	ADCON
				rw			rw					rw	rw	rw	rw

Bits	Fields	Descriptions
31:27	Reserved	Must be kept at reset value.
26	VSLCDEN	Channel 19 (1/3 voltage of V_{SLCD}) enable of ADC. 0: V_{SLCD} channel disabled 1: V_{SLCD} channel enabled
25	VBATEN	Channel 18 (1/3 voltage of external battery) enable of ADC. 0: V_{BAT} channel disabled 1: V_{BAT} channel enabled
24	INREFEN	Channel 17 (internal reference voltage) enable of ADC.

		0: Channel 17 of ADC disable 1: Channel 17 of ADC enable
23	TSVEN	Channel 16 (temperature sensor) enable of ADC. 0: Channel 16 of ADC disable 1: Channel 16 of ADC enable
22	SWRCST	Software start conversion of routine sequence. Set 1 on this bit starts the conversion of a routine sequence if ETSRC is 111. It is set by software and cleared by software or by hardware 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 select for routine sequence 000: TIMER8 CH0 001: TIMER8 CH1 010: reserved 011: TIMER1 CH1 100: TIMER2 TRGO 101: TIMER11 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
7:4	Reserved	Must be kept at reset value.
3	RSTCLB	Reset calibration This bit is set by software and cleared by hardware after the calibration registers are initialized. 0: Calibration register initialization done 1: Calibration register initialization start
2	CLB	ADC calibration 0: Calibration done

		1: Calibration start
1	CTN	Continuous mode 0: Continuous operation mode disable 1: Continuous operation mode enable
0	ADCON	ADC ON. The ADC will be waked up when this bit is changed from low to high and take a stabilization time. When this bit is high and “1” is written to it with other bits of this register unchanged, the conversion will start. 0: ADC disable and power down 1: ADC enable

For GD32L235xx devices

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					VSLCDEN	VBATEN	INREFEN	TSVEN	SWRCST	Reserved	ETERC	ETSRC [2:0]			Reserved
					rw	rw	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	CALNUM[2:0]			RSTCLB	CLB	CTN	ADCON
				rw			rw		rw			rw	rw	rw	rw

Bits	Fields	Descriptions
31:27	Reserved	Must be kept at reset value.
26	VSLCDEN	Channel 19 (1/3 voltage of V_{SLCD}) enable of ADC. 0: V_{SLCD} channel disabled 1: V_{SLCD} channel enabled
25	VBATEN	Channel 18 (1/3 voltage of external battery) enable of ADC. 0: V_{BAT} channel disabled 1: V_{BAT} channel enabled
24	INREFEN	Channel 17 (internal reference voltage) enable of ADC. 0: Channel 17 of ADC disable 1: Channel 17 of ADC enable
23	TSVEN	Channel 16 (temperature sensor) enable of ADC. 0: Channel 16 of ADC disable 1: Channel 16 of ADC enable
22	SWRCST	Software start conversion of routine sequence. Set 1 on this bit starts the conversion of a routine sequence if ETSRC is 111. It is set by software and cleared by software or by hardware 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 select for routine sequence 000: TIMER8 CH0 001: TIMER8 CH1 010: TIMER0_CH2 011: TIMER1 CH1 100: TIMER2 TRGO 101: TIMER11 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	Reserved	Must be kept at reset value.
6:4	CALNUM[2:0]	Calibration times These bits define the calibration times for ADC. 000: 1 time 001: 2 times 010: 4times 011: 8times 100: 16times 101: 32times Others: reserved.
3	RSTCLB	Reset calibration This bit is set by software and cleared by hardware after the calibration registers are initialized. 0: Calibration register initialization done 1: Calibration register initialization start
2	CLB	ADC calibration 0: Calibration done

		1: Calibration start
1	CTN	Continuous mode 0: Continuous operation mode disable 1: Continuous operation mode enable
0	ADCON	ADC ON. The ADC will be waked up when this bit is changed from low to high and take a stabilization time. When this bit is high and “1” is written to it with other bits of this register unchanged, the conversion will start. 0: ADC disable and power down 1: ADC enable

14.5.4. 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		SPT19[2:0]			SPT18[2:0]			SPT17[2:0]			SPT16[2:0]			SPT15[2:0]	
			rw			rw			rw			rw		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
31:30	Reserved	Must be kept at reset value.
29:27	SPT19[2:0]	Refer to SPT10[2:0] description.
26:24	SPT18[2:0]	Refer to SPT10[2:0] description.
23:21	SPT17[2:0]	Refer to SPT10[2:0] description.
20:18	SPT16[2:0]	Refer to SPT10[2:0] description.
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 sampling time 000: channel sampling time is 2.5 cycles 001: channel sampling time is 7.5 cycles 010: channel sampling time is 13.5 cycles

011: channel sampling time is 28.5 cycles
100: channel sampling time is 41.5 cycles
101: channel sampling time is 55.5 cycles
110: channel sampling time is 71.5 cycles
111: channel sampling time is 239.5 cycles

14.5.5. 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 sampling time 000: channel sampling time is 2.5 cycles 001: channel sampling time is 7.5 cycles 010: channel sampling time is 13.5 cycles 011: channel sampling time is 28.5 cycles 100: channel sampling time is 41.5 cycles 101: channel sampling time is 55.5 cycles 110: channel sampling time is 71.5 cycles

111: channel sampling time is 239.5 cycles

14.5.6. Watchdog high threshold register (ADC_WDHT)

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				WDHT [11:0]											
				rw											

Bits	Fields	Descriptions
31:12	Reserved	Must be kept at reset value.
11:0	WDHT[11:0]	High threshold for analog watchdog These bits define the high threshold for the analog watchdog.

14.5.7. Watchdog low threshold register (ADC_WDLT)

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				WDLT [11:0]											
				rw											

Bits	Fields	Descriptions
31:12	Reserved	Must be kept at reset value.
11:0	WDLT[11:0]	Low threshold for analog watchdog These bits define the low threshold for the analog watchdog.

14.5.8. 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]				RSQ15[4:1]			
								rw				rw			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSQ15[0]	RSQ14[4:0]					RSQ13[4:0]					RSQ12[4:0]				
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.
19:15	RSQ15[4:0]	Refer to RSQ0[4:0] description
14:10	RSQ14[4:0]	Refer to RSQ0[4:0] description
9:5	RSQ13[4:0]	Refer to RSQ0[4:0] description
4:0	RSQ12[4:0]	Refer to RSQ0[4:0] description

14.5.9. 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[4:0]					RSQ10[4:0]					RSQ9[4:1]			
		rw					rw					rw			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSQ9[0]	RSQ8[4:0]					RSQ7[4:0]					RSQ6[4:0]				
rw	rw					rw					rw				

Bits	Fields	Descriptions
31:30	Reserved	Must be kept at reset value.
29:25	RSQ11[4:0]	Refer to RSQ0[4:0] description
24:20	RSQ10[4:0]	Refer to RSQ0[4:0] description
19:15	RSQ9[4:0]	Refer to RSQ0[4:0] description
14:10	RSQ8[4:0]	Refer to RSQ0[4:0] description
9:5	RSQ7[4:0]	Refer to RSQ0[4:0] description
4:0	RSQ6[4:0]	Refer to RSQ0[4:0] description

14.5.10. Routine sequence register 2 (ADC_RSQ2)

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		RSQ5[4:0]					RSQ4[4:0]					RSQ3[4:1]			
		rw					rw					rw			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSQ3[0]	RSQ2[4:0]					RSQ1[4:0]					RSQ0[4:0]				
rw	rw					rw					rw				

Bits	Fields	Descriptions
31:30	Reserved	Must be kept at reset value.
29:25	RSQ5[4:0]	Refer to RSQ0[4:0] description
24:20	RSQ4[4:0]	Refer to RSQ0[4:0] description
19:15	RSQ3[4:0]	Refer to RSQ0[4:0] description
14:10	RSQ2[4:0]	Refer to RSQ0[4:0] description
9:5	RSQ1[4:0]	Refer to RSQ0[4:0] description
4:0	RSQ0[4:0]	The channel number (0..19) is written to these bits to select a channel as the nth conversion in the routine sequence.

14.5.11. 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 the conversion result from routine channel, which is read only.

14.5.12. Oversampling 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 oversampling 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 bits 0011: Shift 3 bits 0100: Shift 4 bits 0101: Shift 5 bits 0110: Shift 6 bits 0111: Shift 7 bits 1000: Shift 8 bits Other: 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	Oversampling enable This bit is set and cleared by software. 0: Oversampling disabled 1: Oversampling enabled **Note:** The software allows this bit to be written only when ADCON = 0 (this ensures that no conversion is in progress).

14.5.13. Charge control register (ADC_CCTL)

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															CHARGE
															ro

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

Bits	Fields	Descriptions
31:17	Reserved	Must be kept at reset value.
16	CHARGE	ADC charge status 0: not charging 1: charging Set and reset by hardware.
15:12	Reserved	Must be kept at reset value.
11:0	CCNT [11:0]	ADC charge pulse width counter This bit-field controls the value of the ADC charge pulse width. The relationship between CCNT value and the pulse width is as follow: Pulse Width = 5us = CCNT [11:0] * t_{PCLK2}. **Note:** Software is allowed to write this bit only when ADCON =0 (which ensures that no conversion is ongoing).

14.5.14. Differential mode control register (ADC_DIFCTL)

Only for GD32L235xx devices

Address offset: 0xC4

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												DIFCTL[19:16]			
												r			

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

Bits	Fields	Descriptions
31:20	Reserved	Must be kept at reset value.
19:15	DIFCTL[19:15]	Differential mode for channel 19..15. These bits are read only. These channels are forcedto single-ended input mode (either connected to a single-ended I/O port or to an internal channel).
14:0	DIFCTL[14:0]	Differential mode for channel 14..0. These bits are configured to select whether a channel is in single-ended or differential mode. DIFCTL[i] = 0: ADC analog input channel-i is configured in single-ended mode DIFCTL[i] = 1: ADC analog input channel-i is configured in differential mode **Note:** Software is allowed to write these bits only when the ADC is disabled (ADCON =0).

15. Digital-to-analog converter (DAC)

15.1. Overview

The Digital-to-analog converter converts 12-bit digital data to a voltage on the external pins. The digital data can be configured to 8-bit or 12-bit mode, left-aligned or right-aligned mode. DMA can be used to update the digital data on external triggers.

The output voltage can be optionally buffered for higher drive capability.

15.2. Characteristics

- 8-bit or 12-bit resolution.
- Left or right data alignment.
- DMA capability for each channel and underrun function.
- Conversion update synchronously.
- Conversion triggered by external triggers.
- Configurable internal buffer.
- Extern voltage reference, V_{REFP}.
- Noise wave generation (LFSR noise mode and triangle noise mode).

Figure 15-1. DAC block diagram and ***Table 15-1. DAC I/O description*** show the block diagram of DAC and the pin description of DAC, respectively.

Figure 15-1. DAC block diagram

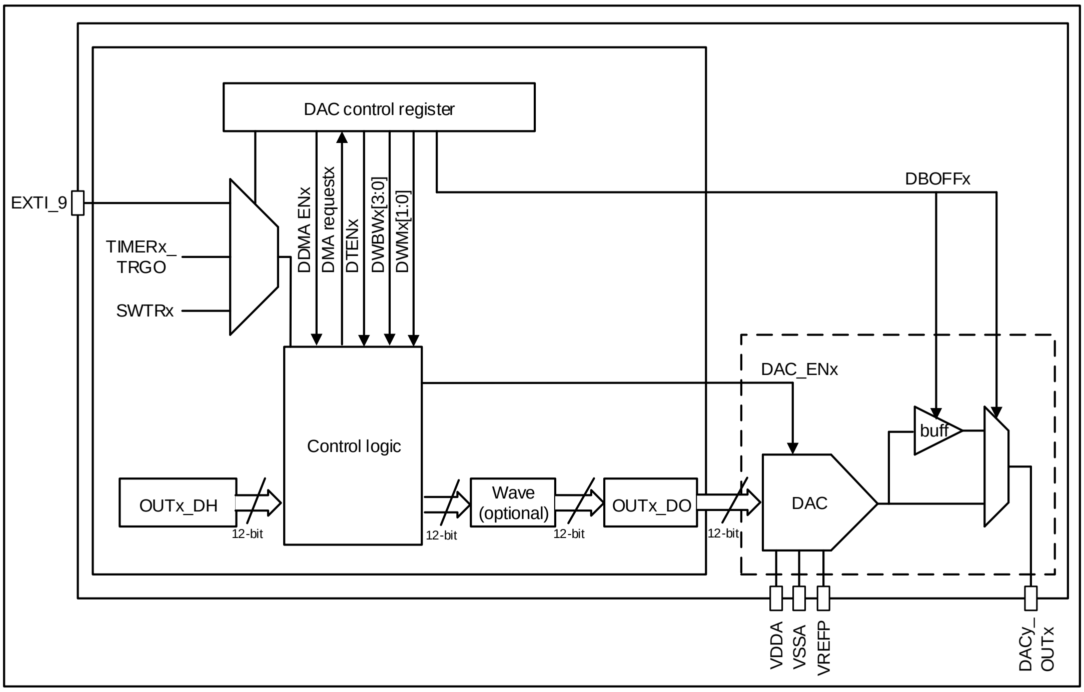


Table 15-1. DAC I/O description

Name	Description	Signal type
V_{DDA}	Analog power supply	Input, analog supply
V_{SSA}	Ground for analog power supply	Input, analog supply ground
V_{REFP}	Positive reference voltage of DAC	Input, analog positive reference
DACy_OUTx	DAC analog output	Analog output signal

The below table details the triggers and outputs of the DAC.

Table 15-2. DAC triggers and outputs summary

	DAC0
Channel	**Channel0**
DAC outputs connected to I / Os	PA4
DAC output buffer	●
DAC software trigger	●
DAC trigger signals from EXTI	EXTI_9
DAC trigger signals from TIMER	TIMER1_TRGO TIMER2_TRGO TIMER5_TRGO TIMER6_TRGO

Note: The GPIO pins should be configured to analog mode before enable the DAC module.

15.3. Function overview

15.3.1. DAC enable

The DAC can be turned on by setting the DENx bit in the DAC_CTL0 register. A t_{WAKEUP} time is needed to startup the analog DAC submodule.

15.3.2. DAC output buffer

For reducing output impedance and driving external loads without an external operational amplifier, an output buffer is integrated inside each DAC module.

The output buffer, which is turned on by default to reduce the output impedance and improve the driving capability, can be turned off by setting the DBOFFx bit in the DAC_CTL0 register.

When DAC output buffer turns off, DAC can connect to on chip peripherals (CMP) independently by setting the DDISC bit in the DAC_CTL register.

15.3.3. DAC data configuration

The 12-bit DAC holding data (OUTx_DH) can be configured by writing any one of the OUTx_R12DH, OUTx_L12DH and OUTx_R8DH registers. When the data is loaded by OUTx_R8DH register, only the MSB 8 bits are configurable, the LSB 4 bits are forced to 4'b0000.

15.3.4. DAC trigger

The DAC conversion can be triggered by software or rising edge of external trigger source. The DAC external trigger is enabled by setting the DTENx bits in the DAC_CTL0 register. The DAC external triggers are selected by the DTSELx bits in the DAC_CTL0 register, which is shown as ***Table 15-3. Triggers of DAC.***

Table 15-3. Triggers of DAC

DTSELx[2:0]	Trigger Source	Trigger Type
3b'000	TIMER1_TRGO	Hardware trigger
3b'001	TIMER2_TRGO	
3b'010	reserved	
3b'011	reserved	
3b'100	TIMER6_TRGO	
3b'101	TIMER5_TRGO	
3b'110	EXTI_9	
3b'111	SWTR	Software trigger

The TIMERx_TRGO signals are generated from the timers, while the software trigger can be generated by setting the SWTRx bits in the DAC_SWT register.

15.3.5. DAC conversion

If the external trigger is enabled by setting the DTENx bit in DAC_CTL0 register, the DAC holding data is transferred to the DAC output data (OUTx_DO) register when the selected trigger event happened. When the external trigger is disabled, the transfer is performed automatically.

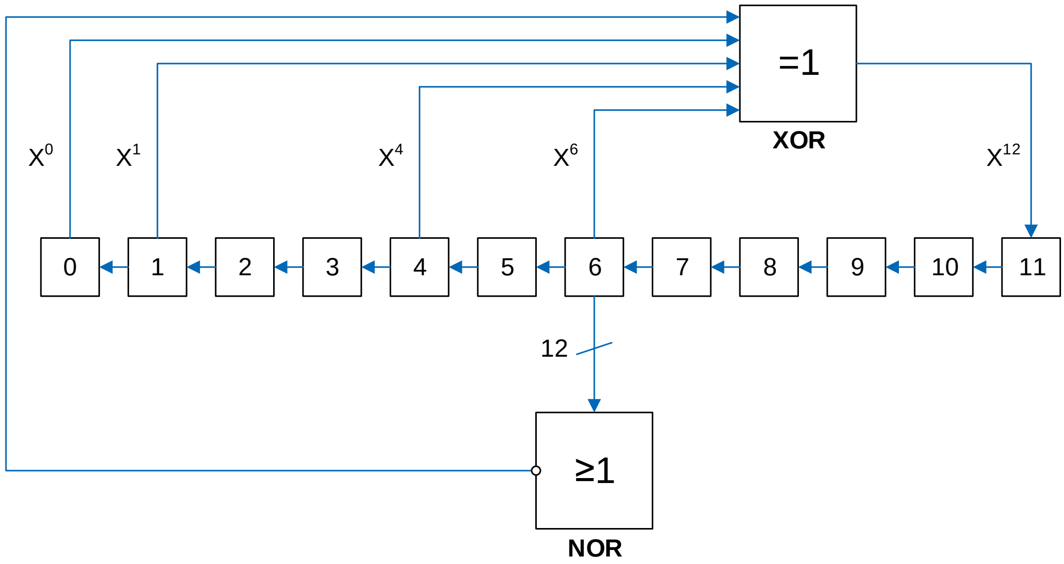
When the DAC holding data (OUTx_DH) is loaded into the OUTx_DO register, after the time $t_{SETTLING}$ which is determined by the analog output load and the power supply voltage, the analog output is valid.

15.3.6. DAC noise wave

There are two methods of adding noise wave to the DAC output data: LFSR noise wave mode and Triangle wave mode. The noise wave mode can be selected by the DWMx bits in the DAC_CTL0 register. The amplitude of the noise can be configured by the DAC noise wave bit width (DWBWx) bits in the DAC_CTL0 register.

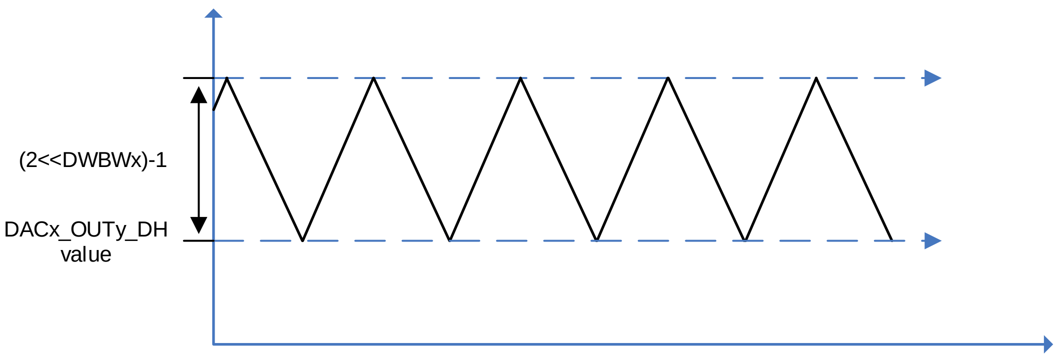
LFSR noise wave mode: there is a Linear Feedback Shift Register (LFSR) in the DAC control logic, it controls the LFSR noise signal which is added to the OUTx_DH value, and then the result is stored into the OUTx_DO register When the configured DAC noise wave bit width is less than 12, the noise signal equals to the LSB DWBWx bits of the LFSR register, while the MSB bits are masked.

Figure 15-2. DAC LFSR algorithm



Triangle noise mode: a triangle signal is added to the OUTx_DH value, and then the result is stored into the OUTx_DO register. The minimum value of the triangle signal is 0, while the maximum value of the triangle signal is (2 << DWBWx) - 1.

Figure 15-3. DAC triangle noise wave



15.3.7. DAC output voltage

The following equation determines the analog output voltage on the DAC pin.

$$V_{DAC\_OUT}=V_{REFP}*OUTx\_DO/4096 \qquad (15\text{-}1)$$

The digital input is linearly converted to an analog output voltage, and its range is 0 to V_{REFP}.

15.3.8. DMA request

When the external trigger is enabled, the DMA request is enabled by setting the DDMAENx bit of the DAC_CTL0 register. A DMA request will be generated when an external hardware trigger (not a software trigger) occurs.

If the second external trigger arrives before confirming the previous request, the new request will not be serviced, and an underrun error event occurs. The DDUDRx bit in the DAC_STAT0 register is set, an interrupt will be generated if the DDUDRIEx bit in the DAC_CTL0 register is set. The DMA request will be stalled until the DDUDRx bit is cleared.

15.4. Register definition

DAC0 base address: 0x4000 7400

15.4.1. DACx control register 0 (DAC_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	DDISC0	DDUDRIE0	DDMAEN0	DWBW0[3:0]				DWM0[1:0]		DTSEL0[2:0]			DTEN0	DBOFF0	DEN0
	rw	rw	rw	rw				rw		rw			rw	rw	rw

Bits	Fields	Descriptions
31:15	Reserved	Must be kept at reset value.
14	DDISC0	DACx_OUT0 connect GPIO selection 0: DACx_OUT0 is connected to the external pin and on chip peripherals (CMP). 1: DACx_OUT0 is connected to on chip peripherals (CMP) independently only if output buffer turns off (DBOFF=1). Otherwise it is connected to the external pin and to on chip peripherals (CMP).
13	DDUDRIE0	DACx_OUT0 DMA underrun interrupt enable 0: DACx_OUT0 DMA underrun interrupt disabled 1: DACx_OUT0 DMA underrun interrupt enabled
12	DDMAEN0	DACx_OUT0 DMA enable 0: DACx_OUT0 DMA mode disabled 1: DACx_OUT0 DMA mode enabled
11:8	DWBW0[3:0]	DACx_OUT0 noise wave bit width These bits specify bit width of the noise wave signal of DACx_OUT0. These bits indicate that unmask LFSR bit [n-1, 0] in LFSR noise mode or the amplitude of the triangle is ((2<<(n-1))-1) in triangle noise mode, where n is the bit width of wave. 0000: The bit width of the wave signal is 1 0001: The bit width of the wave signal is 2 0010: The bit width of the wave signal is 3 0011: The bit width of the wave signal is 4 0100: The bit width of the wave signal is 5 0101: The bit width of the wave signal is 6 0110: The bit width of the wave signal is 7

		0111: The bit width of the wave signal is 8 1000: The bit width of the wave signal is 9 1001: The bit width of the wave signal is 10 1010: The bit width of the wave signal is 11 ⩾1011: The bit width of the wave signal is 12
7:6	DWM0[1:0]	DACx_OUT0 noise wave mode These bits specify the mode selection of the noise wave signal of DACx_OUT0 when external trigger of DACx_OUT0 is enabled (DTEN0=1). 00: Wave disabled 01: LFSR noise mode 1x: Triangle noise mode
5:3	DTSEL0[2:0]	DACx_OUT0 trigger selection These bits are only used if bit DTEN = 1 and select the external event used to trigger DAC. 000: TIMER1 TRGO 001: TIMER2 TRGO 010: Reserved 011: Reserved 100: TIMER6 TRGO 101: TIMER5 TRGO 110: EXTI line 9 111: Software trigger
2	DTEN0	DACx_OUT0 trigger enable 0: DACx_OUT0 trigger disabled 1: DACx_OUT0 trigger enabled
1	DBOFF0	DACx_OUT0 output buffer turn off 0: DACx_OUT0 output buffer turns on to reduce the output impedance and improve the driving capability 1: DACx_OUT0 output buffer turns off
0	DEN0	DACx_OUT0 enable 0: DACx_OUT0 disabled 1: DACx_OUT0 enabled

15.4.2. DACx software trigger register (DAC_SWT)

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															SWTR0
															w

Bits	Fields	Descriptions
31:1	Reserved	Must be kept at reset value.
0	SWTR0	DACx_OUT0 software trigger, cleared by hardware. 0: Software trigger disabled 1: Software trigger enabled

15.4.3. DACx_OUT0 12-bit right-aligned data holding register (DAC_OUT0_R12DH)

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				OUT0_DH[11:0]											
				rw											

Bits	Fields	Descriptions
31:12	Reserved	Must be kept at reset value.
11:0	OUT0_DH[11:0]	DACx_OUT0 12-bit right-aligned data. These bits specify the data that is to be converted by DACx_OUT0.

15.4.4. DACx_OUT0 12-bit left-aligned data holding register (DAC_OUT0_L12DH)

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
OUT0_DH[11:0]												Reserved			
rw															

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:4	OUT0_DH[11:0]	DACx_OUT0 12-bit left-aligned data. These bits specify the data that is to be converted by DACx_OUT0.
3:0	Reserved	Must be kept at reset value.

15.4.5. DACx_OUT0 8-bit right-aligned data holding register (DAC_OUT0_R8DH)

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								OUT0_DH[7:0]							
								rw							

Bits	Fields	Descriptions
31:8	Reserved	Must be kept at reset value.
7:0	OUT0_DH[7:0]	DACx_OUT0 8-bit right-aligned data. These bits specify the MSB 8-bit of the data that is to be converted by DACx_OUT0.

15.4.6. DACx_OUT0 data output register (DAC_OUT0_DO)

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				OUT0_DO [11:0]											
				r											

Bits	Fields	Descriptions

31:12	Reserved	Must be kept at reset value.
11:0	OUT0_DO [11:0]	DACx_OUT0 12-bit output data These bits, which are read only, storage the data that is being converted by DACx_OUT0.

15.4.7. DACx status register 0 (DAC_STAT0)

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		DDUDR0	Reserved												
		rc_w1													

Bits	Fields	Descriptions
31:14	Reserved	Must be kept at reset value.
13	DDUDR0	DACx_OUT0 DMA underrun flag. This bit is set by hardware and cleared by software (by writing it to 1). 0: no underrun occurred. 1: underrun occurred (Speed of DAC trigger is high than the DMA transfer).
12:0	Reserved	Must be kept at reset value.

16. 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.

16.1. Free watchdog timer (FWDGT)

16.1.1. 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.

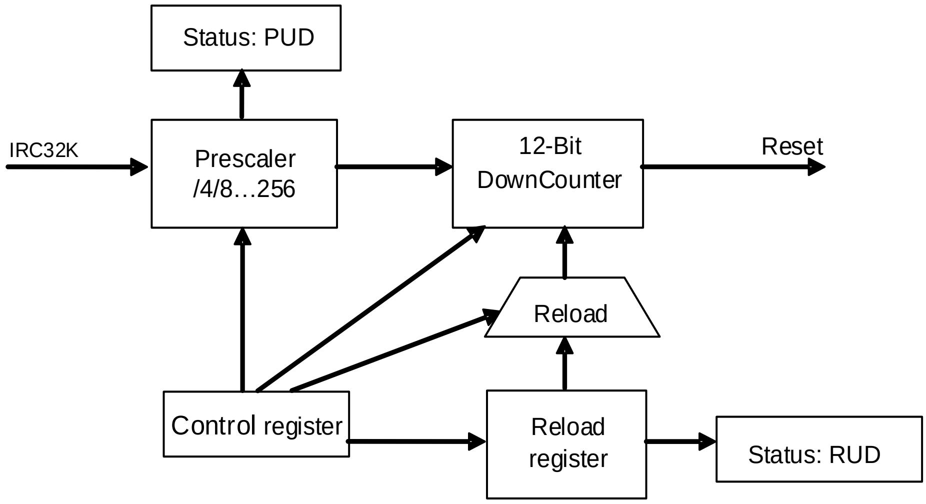
16.1.2. Characteristics

- 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.

16.1.3. Function overview

The free watchdog consists of an 8-stage prescaler and a 12-bit down counter. ***Figure 16-1. Free watchdog block diagram*** shows the functional block of the free watchdog module.

Figure 16-1. 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 16-1. 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.

16.1.4. 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).

16.2. Window watchdog timer (WWDGT)

16.2.1. 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.Interrup occurs if it is enable.

The window watchdog timer clock is prescaled from the APB1 clock. The window watchdog timer is suitable for the situation that requires an accurate timing.

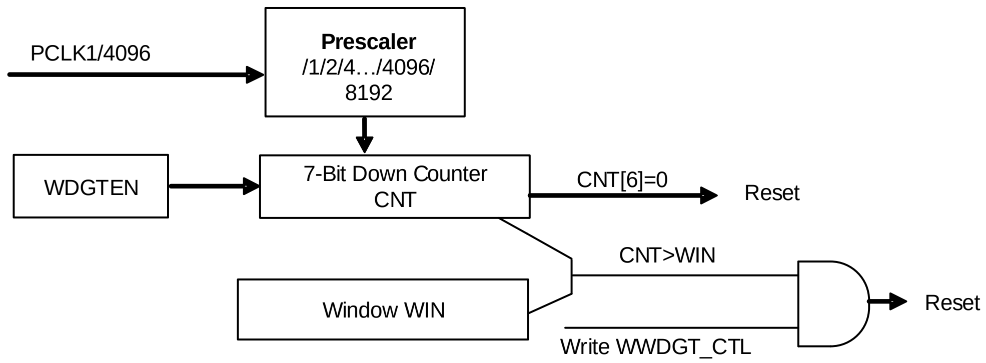
16.2.2. Characteristics

- 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.

16.2.3. 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 16-2. Window watchdog timer block diagram



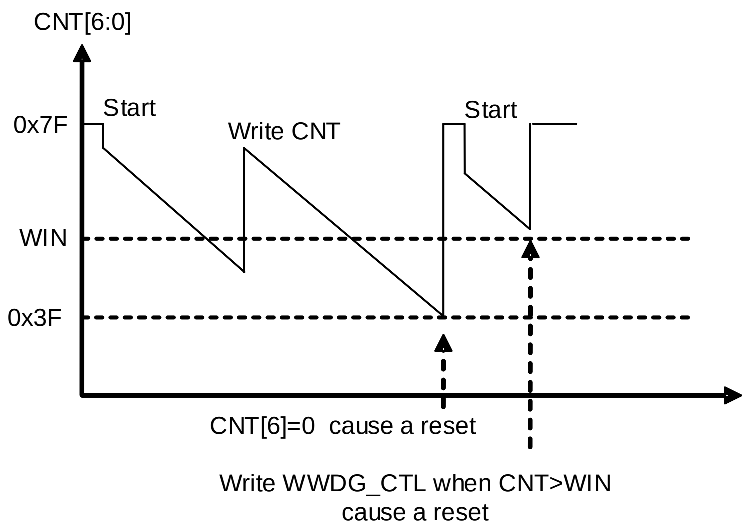
The window watchdog timer is always disabled after power on reset. The software starts the watchdog by setting the WDGTEN bit in the WWDGT_CTL register. When window watchdog timer is enabled, the counter counts down all the time, the configured value of the counter should be greater than 0x3F(it implies 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 APB1 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 16-3. Window watchdog timing diagram



Calculate the WWDGT timeout by using the formula below.

$$T_{WWDGT}=t_{PCLK1} \times 4096 \times 2^{PSC} \times (CNT[5:0]+1) \quad (ms) \qquad (15\text{-}1)$$

where:

t_{WWDGT}: WWDGT timeout

t_{PCLK1}: APB1 clock period measured in ms

The ***Table 16-2. Min-max timeout value at 64 MHz (fPCLK1)*** shows the minimum and maximum values of the t_{WWDGT}.

Table 16-2. Min-max timeout value at 64 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	64μs	4.096ms
1 / 2	0001	128μs	8.192ms
1 / 4	0010	256μs	16.384ms
1 / 8	0011	512μs	32.768ms
1 / 16	0100	1.024ms	65.536ms
1 / 32	0101	2.048ms	131.072ms
1 / 64	0110	4.096ms	262.144ms
1 / 128	0111	8.192ms	524.288ms
1 / 256	1000	16.384ms	1049.576ms
1 / 512	1001	32.768ms	2097.152ms
1 / 1024	1010	65.536ms	4194.304ms
1 / 2048	1011	131.072ms	8388.608ms
1 / 4096	1100	262.144ms	16777.216ms
1 / 8192	1101	524.288ms	33554.432ms
1 / 1	1110	64μs	4.096ms
1 / 1	1111	64μs	4.096ms

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.

16.2.4. 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.

17. Real time clock (RTC)

17.1. 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 and smart wakeup is software configurable. Support improving the calendar accuracy using extern accurate low frequency clock.

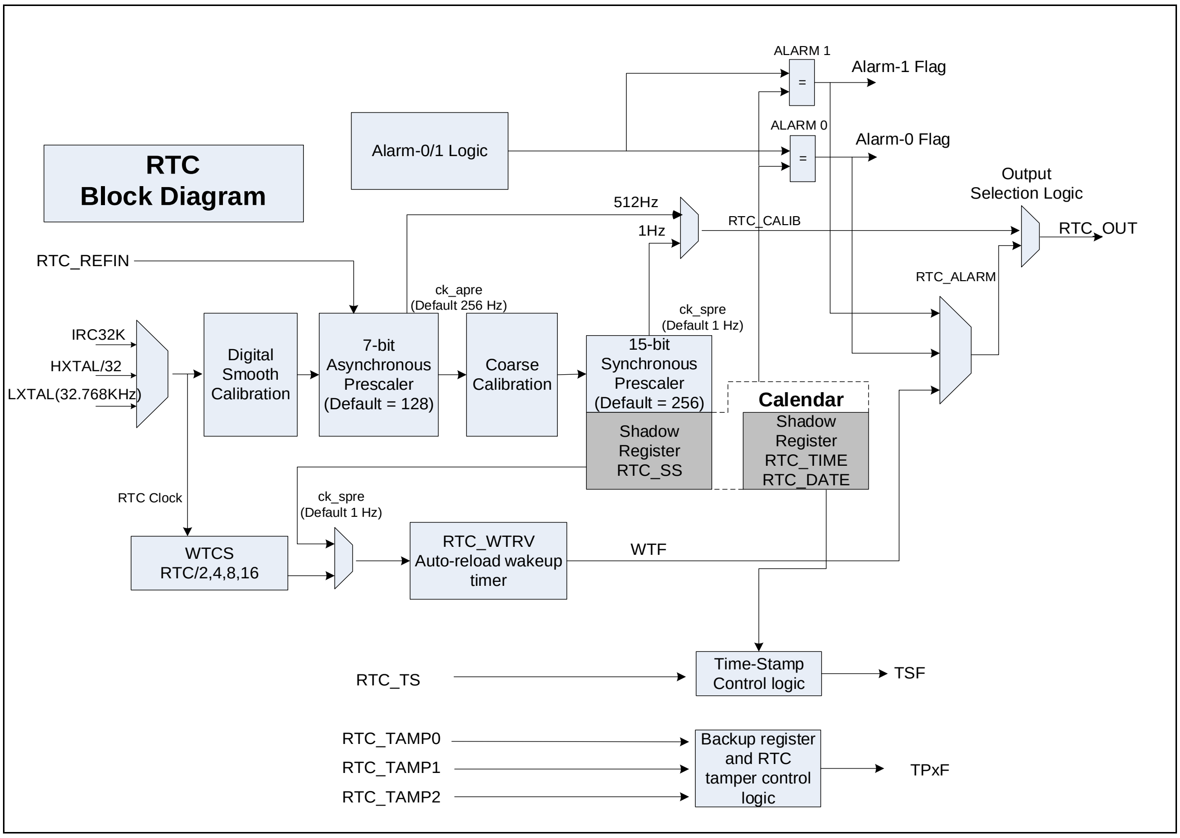
17.2. Characteristics

- 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
- Three Tamper sources can be chosen and tamper type is configurable
- Programmable calendar and two field maskable alarms
- Maskable interrupt source:
  - Alarm 0 and Alarm 1
  - Time-stamp detection
  - Tamper detection
  - Auto wakeup event
- Five 32-bit (20 bytes total) universal backup registers which can keep data under power saving mode. Backup register will be reset if tamper event detected

17.3. Function overview

17.3.1. Block diagram

Figure 17-1. Block diagram of RTC



The RTC unit includes:

- Alarm event/interrupt
- Tamper event/interrupt
- Tamper detection erases the backup registers
- Timestamp can be generated when a switch to VBAT occurs
- 32-bit backup registers
- Optional RTC output function:
  - 512Hz (default prescale) :RTC_OUT(PC13/PB2/PB14)
  - 1Hz(default prescale): RTC_OUT(PC13/PB2/PB14)
  - Alarm event(polarity is configurable): RTC_OUT(PC13/PB2/PB14)
  - Automatic wakeup event(polarity is configurable): RTC_OUT(PC13/PB2/PB14)
- Optional RTC input function:
  - time stamp event detection: RTC_TS(PC13)
  - tamper 0 event detection: RTC_TAMP0(PC13)
  - tamper 1 event detection: RTC_TAMP1(PA0)
  - tamper 2 event detection: RTC_TAMP2(PA2)
  - reference clock input: RTC_REFIN(PB15)

17.3.2. Clock source and prescalers

RTC unit has three independent clock sources: LXTAL, IRC32K and HXTAL with divided by 32(configured in RCU_CFG 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:

$$f_{ck\_apre} = \frac{f_{rtcclk}}{FACTOR\_A + 1} \tag{17-1}$$

$$f_{ck\_spre} = \frac{f_{ck\_apre}}{FACTOR\_S + 1} = \frac{f_{rtcclk}}{(FACTOR\_A + 1)*(FACTOR\_S + 1)} \tag{17-2}$$

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.

17.3.3. 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.

17.3.4. 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=0,1) bit in RTC_CTL. If all the alarm fields value match the corresponding calendar value when ALRMxEN=1(x=0,1), 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=0,1).

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=0,1) is set.

17.3.5. Configurable periodic auto-wakeup counter

In the RTC block, there is a 16-bit down counter designed to generate periodic wakeup flag. This function is enabled by set the WTEN to 1 and can be running in power saving mode.

Two clock sources can be chose for the down counter:

1) RTC clock divided by 2/4/8/16

   Assume RTC clock comes from LXTAL (32.768 KHz), this can periodically assert wakeup interrupt from 122us to 32s under the resolution down to 61us.

2) Internal clock ck_spre

   Assume ck_spre is 1Hz, this can periodically assert wakeup interrupt from 1s to 36 hours under the resolution down to 1s.

   - WTCS[2:1] = 0b10.This will make period to be 1s to 18 hours
   - WTCS[2:1] = 0b11.This will make period to be 18 to 36 hours

When this function is enabled, the down counter is running. When it reaches 0, the WTF flag is set and the wakeup counter is automatically reloaded with RTC_WUT value.

When WTF asserts, software must then clear it.

If WTIE is set and this counter reaches 0, a wakeup interrupt will make system exit from the power saving mode. System reset has no influence on this function.

WTF is also can be output to RTC_OUT from RTC_ALARM channel.

17.3.6. 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_WUT, RTC_ALRM0TD, RTC_ALRM1TD, RTC_SHIFTCTL, RTC_HRFC, RTC_ALRM0SS, RTC_ALRM1SS

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 tautologically 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=0,1) in RTC_CTL)
2. Set the Alarm registers needed(RTC_ALRMxTD/RTC_ALRMxSS(x=0,1))
3. Enable Alarm function (by setting ALRMxEN(x=0,1) in the RTC_CTL)

17.3.7. 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 APB1 bus clock frequency must be equal to or greater than 7 times the RTC clock frequency.APB1 bus clock frequency lower than RTC clock frequency is not allowed in any case whatever happens.

When APB1 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.

17.3.8. Resetting the RTC

There are two reset sources used in RTC unit: system reset and backup domain 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 domain 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 Wakeup timer register (RTC_WUT)
- 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 Tamper register (RTC_TAMP)
- RTC Backup registers (RTC_BKPx)
- RTC Alarm registers (RTC_ALRMxSS/RTC_ALRMxTD(x=0,1))

The RTC unit will go on running when system reset occurs or enter power saving mode, but if backup domain reset occurs, RTC will stop counting and all registers will reset.

17.3.9. 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.

17.3.10. 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.

17.3.11. 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 211/210/29(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:

$$f_{cal} = f_{rtcclk} \times (1 + \frac{FREQI \times 512 - CMSK}{2^N + CMSK - FREQI \times 512}) \qquad (17\text{-}3)$$

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 = 2: 2 less than nominal FACTOR_S (8189 with 32.768 KHz)

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:

$$f_{cal} = f_{rtcclk} \times (1 + \frac{256 - CMSK}{2^N + CMSK - 256}) \qquad (17\text{-}4)$$

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

17.3.12. Time-stamp function

Time-stamp function is performed on RTC_TS pin and is enabled by control bit TSEN. It is also enabled by control bit ITSEN

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 event can generate an interrupt if time-stamp interrupt enable (TSIE) is set.

When an internal time-stamp event detected (ITSEN = 1), the calendar value will be saved in time-stamp registers (RTC_DTS/RTC_TTS/RTC_SSTS), the time-stamp flag (TSF) and internal time-stamp flag (ITSF) is set to 1 by hardware. Time-stamp event can generate an interrupt if internal time-stamp interrupt enable (TSIE) is set.The internal timestamp event is generated by the switch to the V_{BAT} supply

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.

To extend the time-stamp event source, one optional feature is provided: tamper function can also be considered as time-stamp function if TPTS is set.

Note: When the time-stamp event occurs, TSF is set 2 ck_apre cycles delay because of synchronization mechanism.

17.3.13. Tamper detection

The RTC_TAMPx pin input can be used for tamper event detection under edge detection mode or level detection mode with configurable filtering setting.

The purposes of the tamper detect configuration are the following:

1. The default configuration will erase the RTC backup registers
2. It can wakeup from DeepSleep and Standby modes, and generate an interrupt
3. It is used for the low-power timers to generate a hardware trigger

RTC backup registers (RTC_BKPx)

The RTC backup registers are located in the VDD backup domain that remains powered-on by V_{BAT} even if V_{DD} power is switched off. The wake up action from Standby Mode or System Reset does not affect these registers.

These registers are only reset by detected tamper event and backup domain reset except if the TPxNOERASE bit is set, or if TPxMASK is set in the RTC_TAMP register.

Tamper detection function initialization

RTC tamper detection function can be independently enabled on tamper input pin by setting corresponding TpxEN bit. Tamper detection configuration is set before enable TpxEN bit.

TPxMASK =0:

The TPxF flag is set after the tamper event occurs on the pin with the following latency:

2. When FLT is different from 0x0 (Level detection mode with configurable filtering), there are three ck_apre cycles
3. When TPTS is set (Timestamp on tamper event), there are three ck_apre cycles
4. When FLT is reset (Edge detection mode on tamper input detection) and TPTS is reset, there is no latency.

When TPxF is set during the latency, new tamper cannot be detected occurring on the same pin.

TPxMASK=1:

When TPxF is set during the latency and 2.5 RTC clock additional, new tamper cannot be detected occurring on the same pin.

Tamper event can generate an interrupt if tamper interrupt enable (TPIE) is set.when one or more TPxMASK is set, TPIE can not be setting.

When TPIE is cleared, each tamper pin event interrupt can be enabled independently by setting the corresponding TpxIE bit. When the corresponding TPxMASK is set, TpxIE cannot be set.

Trigger output generation on tamper event

The tamper event detection can be used as trigger input for the low-power timers

To allow a new tamper detection on the same pin, the TPxF flag must be cleared by software When TPxMASK bit is cleared.

The TPxF flag is masked, and kept cleared in RTC_STAT register if TPxMASK bit is set.

This configuration can trigger the low-power timers in Deep-sleep mode automatically without the system wakeup to clear the TPxF flag. The backup registers are not cleared in this case.

Timestamp on tamper event

The TPTS bit can control whether the tamper detection function is used as time-stamp function. If the bit is set to 1, the TSF bit will be set when the tamper event detected as if "enable" the time-stamp function. Whatever the TPTS bit is, the TPxF will assert when tamper event detected.

Edge detection mode on tamper input detection

When FLT bit is set to 0x0, the tamper detection is set to edge detection mode and TpxEG bit determines the rising edge or falling edge is the detecting edge. When tamper detection is under edge detection mode, the internal pull-up resistors on the tamper detection input pin are deactivated.

Because of detecting the tamper event will reset the backup registers (RTC_BKPx), writing to the backup register should ensure that the tamper event reset and the writing operation will not occur at the same time, a recommend way to avoid this situation is disable the tamper detection before writing to the backup register and re-enable tamper detection after finish writing.

Note: Tamper detection is still running when V_{DD} power is switched off if tamper is enabled.

Level detection mode with configurable filtering on tamper input detection

When FLT bit is not reset to 0x0, the tamper detection is set to level detection mode and FLT bit determines the consecutive number of samples (2, 4 or 8) needed for valid level. When DISPU is set to 0x0(this is default), the internal pull-up resistance will pre-charge the tamper input pin before each sampling and thus larger capacitance is allowed to connect to the tamper input pin. The pre-charge duration is configured through PRCH bit. Higher capacitance needs long pre-charge time.

The time interval between each sampling is also configurable. Through adjusting the sampling frequency (FREQ), software can balance between the power consuming and tamper detection latency.

17.3.14. Calibration clock output

Calibration clock can be output on the RTC_OUT 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:

$$f_{rtc\_calib} = \frac{f_{rtcclk}}{(FACTOR\_A+1)\times(FACTOR\_S+1)} \tag{16-5}$$

When the RTCCLK is 32.768 KHz, RTC_CALIB output is corresponding to 1Hz if prescaler are default values.

17.3.15. 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 or auto wakeup flag bit in RTC_STAT.

The OPOL bit in RTC_CTL can configure the polarity of the alarm or auto wakeup flag output which means that the RTC_ALARM output is the opposite of the corresponding flag bit or not.

17.3.16. RTC pin configuration

RTC_OUT, RTC_TS and RTC_TAMP0 use the same pin (PC13). Function of PC13 is controlled by the RTC and regardless of PC13 GPIO configuration. The RTC functions of PC13 are available in all low-power modes and in VBAT only mode.

The priority of the PC13 output shown in ***Table 17-1 RTC pin PC13 configuration***

Table 17-1 RTC pin PC13 configuration

function configuration and pin function	OS[1:0] (output selection)	COEN (calibration output)	TP0EN (tamper enabled)	TSEN (time stamp enabled)	ALARMOUTTYPE(RTC_ALARM output type)
Alarm out output open drain	01 or 10 or 11	-	-	-	0
Alarm out output push-pull	01 or 10 or 11	-	-	-	1
Calibration output push-pull	00	1	-	-	-
TAMP0 input floating	00	0	1	0	-
TIMESTAMP and TAMP0 input floating	00	0	1	1	Don't care

function configuration and pin function	OS[1:0] (output selection)	COEN (calibration output)	TP0EN (tamper enabled)	TSEN (time stamp enabled)	ALARMOUTTYPE(RTC_ALARM output type)
TIMESTAMP input floating	00	0	0	1	Don't care
Standard GPIO	00	0	0	0	Don't care

It is possible to output RTC_OUT on PB2/PB14 pin thanks to OUT2EN bit in RTC_CTL[31]. This output is not available in VBAT only mode.

Table 17-2 RTC functions in all lowpower modes

Pin	RTC functions	all lowpower modes except Standby mode	Standby mode	VBAT only mode
PC13	RTC_TAMP0 RTC_TS RTC_OUT	YES	YES	YES
PA0	RTC_TAMP1	YES	YES	YES
PA2	RTC_TAMP2	YES	YES	YES
PB2	RTC_OUT	YES	NO	NO
PB14	RTC_OUT	YES	NO	NO
PB15	RTC_REFIN	YES	NO	NO

17.3.17. RTC power saving mode management

Table 17-3 RTC power saving mode management

Mode	Active in Mode	Exit Mode
Sleep	Yes	RTC Interrupts
Sleep1	Yes	RTC Interrupts
LPSleep	Yes	RTC Interrupts
Deep-sleep	Yes: if clock source is LXTAL or IRC32K	RTC Interrupts
Deep-sleep1	Yes: if clock source is LXTAL or IRC32K	RTC Interrupts
Deep-sleep2	Yes: if clock source is LXTAL or IRC32K	RTC Interrupts
Standby	Yes: if clock source is LXTAL or IRC32K	RTC Alarm / Tamper Event / Timestamp Event / Wake up

17.3.18. RTC interrupts

All RTC interrupts are connected to the EXTI controller.

Below steps should be followed if you want to use the RTC alarm/tamper/timestamp/auto wakeup interrupt:

1. Configure and enable the corresponding interrupt line to RTC alarm/tamper/timestamp/auto wakeup event of EXTI and set the rising edge for triggering
2. Configure and enable the RTC alarm/tamper/timestamp/auto wakeup interrupt
3. Configure and enable the RTC alarm/tamper/timestamp/auto wakeup function

Table 17-4 RTC interrupts control

Interrupt	Event flag	Control Bit	Exit Sleep	Exit Deep-sleep And Standby
Alarm 0	ALRM0F	ALRM0IE	Y	Y(1)
Alarm 1	ALRM1F	ALRM1IE	Y	Y(1)
Wakeup	WTF	WTIE	Y	Y(1)
Timestamp	TSF	TSIE	Y	Y(1)
Tamper 0	TP0F	TPIE	Y	Y(1)
Tamper 1	TP1F	TPIE	Y	Y(1)
Tamper 2	TP2F	TPIE	Y	Y(1)

(2) Only active when RTC clock source is LXTAL or IRC32K.

17.4. Register definition

RTC base address: 0x4000 2800

17.4.1. 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

17.4.2. 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

17.4.3. Control register (RTC_CTL)

Address offset: 0x08

System reset: not affected

Backup domain 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
OUT2EN	Reserved						ITSEN	COEN	OS[1:0]		OPOL	COS	DSM	S1H	A1H
rw							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	WTIE	ALRM1IE	ALRM0IE	TSEN	WTEN	ALRM1EN	ALRM0EN	Reserved	CS	BPSHAD	REFEN	TSEG	WTCS[2:0]		
rw	rw	rw	rw	rw	rw	rw	rw		rw	rw	rw	rw	rw		

Bits	Fields	Descriptions
31:24	OUT2EN	RTC_OUT pin select 0:RTC_OUT is output on PC13

		1: RTC_OUT is output on PB2/PB14
30:25	Reserved	Must be kept at reset value.
24	ITSEN	Internal timestamp event enable 0: Disable Internal timestamp event 1: Enable Internal timestamp event
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: Enable alarm1 flag output 0x3: Enable wakeup flag output
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	WTIE	Auto-wakeup timer interrupt enable 0: Disable auto-wakeup timer interrupt

		1: Enable auto-wakeup timer interrupt
13	ALRM1IE	RTC alarm-1 interrupt enable 0: Disable alarm interrupt 1: Enable alarm interrupt
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	WTEN	Auto-wakeup timer function enable 0: Disable function 1: Enable function
9	ALRM1EN	Alarm-1 function enable 0: Disable alarm function 1: Enable alarm function
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 APB1 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	WTCS[2:0]	Auto-wakeup timer clock selection 0x0:RTC Clock divided by 16

0x1:RTC Clock divided by 8

0x2:RTC Clock divided by 4

0x3:RTC Clock divided by 2

0x4:0x5: ck_spre (default 1Hz) clock

0x6:0x7: ck_spre (default 1Hz) clock and 2^{16} is added to wake-up counter.

17.4.4. Status register (RTC_STAT)

Address offset: 0x0C

System reset: Only INITM, INITF and RSYNF bits are set to 0. Others are not affected

Backup domain reset value: 0x0000 0007

This register is writing protected except RTC_STAT[14:8].

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														ITSF	SCPF
														rc_w0	r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TP2F	TP1F	TP0F	TSOVRF	TSF	WTF	ALRM1F	ALRM0F	INITM	INITF	RSYNF	YCM	SOPF	WTWF	ALRM1WF	ALRM0WF
rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rw	r	rc_w0	r	r	r	r	r

Bits	Fields	Descriptions
31:18	Reserved	Must be kept at reset value.
17	ITSF	Internal timestamp flag Set by hardware when internal time-stamp event is detected. Cleared by software writing 0, and must be cleared together with TSF bit by writing 0 in both bits.
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	TP2F	RTC_TAMP2 detected flag Set to 1 by hardware when tamper detection is found on tamper2 input pin. Software can clear this bit by writing 0 into this bit.
14	TP1F	RTC_TAMP1 detected flag Set to 1 by hardware when tamper detection is found on tamper1 input pin. Software can clear this bit by writing 0 into this bit.
13	TP0F	RTC_TAMP0 detected flag Set to 1 by hardware when tamper detection is found on tamper0 input pin. Software can clear this bit by writing 0 into this bit.
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	WTF	Wakeup timer flag Set by hardware when wakeup timer decreased to 0. Cleared by software writing 0. This flag must be cleared at least 1.5 RTC Clock periods before WTF is set to 1 again.
9	ALRM1F	Alarm-1 occurs flag Set to 1 by hardware when current time/date matches the time/date of alarm 1 setting value. Cleared by software writing 0.
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	WTWF	Wakeup timer write enable flag 0: Wakeup timer update is not allowed 1: Wakeup timer update is allowed
1	ALRM1WF	Alarm 1 configuration can be write flag Set by hardware if alarm register can be wrote after ALRM1EN bit has reset. 0: Alarm registers programming is not allowed 1: Alarm registers programming is allowed
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.

17.4.5. Prescaler register (RTC_PSC)

Address offset: 0x10

System reset: not effected

Backup domain 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)

17.4.6. Wakeup timer register (RTC_WUT)

Address offset: 0x14

System reset: not effected

Backup domain reset value: 0x0000 FFFF

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
Reserved															

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

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	WTRV[15:0]	Auto-wakeup timer reloads value. Every (WTRV[15:0]+1) ck_wut period the WTF bit is set after WTEN=1.The ck_wut is selected by WTCS[2:0] bits. Note: This configure case is forbidden: WTRV=0x0000 with WTCS[2:0]=0b011. This register can be written only when WTWF=1.

17.4.7. Alarm 0 time and date register (RTC_ALRM0TD)

Address offset: 0x1C

System reset: not effect

Backup domain reset value: 0x0000 0000

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
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

17.4.8. Alarm 1 time and date register (RTC_ALRM1TD)

Address offset: 0x20

System reset: not effect

Backup domain reset value: 0x0000 0000

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
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[3:0] has no means.
29:28	DAYT[1:0]	Day tens in BCD code
27:24	DAYU[3:0]	Day 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

17.4.9. 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

17.4.10. 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)

17.4.11. Shift function control register (RTC_SHIFTCTL)

Address offset: 0x2C

System reset: not effect

Backup 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.

17.4.12. Time of time stamp register (RTC_TTS)

Address offset: 0x30

Backup domain 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

17.4.13. Date of time stamp register (RTC_DTS)

Address offset: 0x34

Backup domain 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		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

17.4.14. Sub second of time stamp register (RTC_SSTS)

Address offset: 0x38

Backup domain 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.

17.4.15. High resolution frequency compensation register (RTC_HRFC)

Address offset: 0x3C

Backup domain 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.

17.4.16. Tamper register (RTC_TAMP)

Address offset: 0x40

Backup domain 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		TP2IE	TP1IE	TP0IE	Reserved	TP2MASK	TP1MASK	TP0MASK	Reserved	TP2NOERASE	TP1NOERASE	TP0NOERASE	ALRMOUTTYPE	Reserved	
		rw	rw	rw		rw	rw	rw		rw	rw	rw	rw		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DISPU	PRCH[1:0]		FLT[1:0]		FREQ[2:0]			TPTS	TP2EG	TP2EN	TP1EG	TP1EN	TPIE	TP0EG	TP0EN
rw	rw		rw		rw			rw	rw	rw	rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31:30	Reserved	Must be kept at reset value.
29	TP2IE	Tamper 2 interrupt enable 0: Disable tamper 2interrupt 1: Enable tamper 2 interrupt
28	TP1IE	Tamper 1 interrupt enable 0: Disable tamper 1 interrupt 1: Enable tamper 1 interrupt
27	TP0IE	Tamper 0 interrupt enable 0: Disable tamper 0 interrupt 1: Enable tamper 0 interrupt
26	Reserved	Must be kept at reset value.
25	TP2MASK	Tamper 2 mask flag 0:Tamper 2 event generates a trigger event and TP2F must be cleared by software to allow next tamper event detection 1:Tamper 2 event generates a trigger event. TP2F is masked and internally cleared by hardware. The backup registers are not erased *Note: The Tamper 2 interrupt must not be enabled when TP2MASK is set.*
24	TP1MASK	Tamper 1 mask flag 0:Tamper 1 event generates a trigger event and TP1F must be cleared by software to allow next tamper event detection 1:Tamper 1 event generates a trigger event. TP1F is masked and internally cleared by hardware. The backup registers are not erased *Note: The Tamper 1 interrupt must not be enabled when TP1MASK is set.*
23	TP0MASK	Tamper 0 mask flag 0:Tamper 0 event generates a trigger event and TP0F must be cleared by software to allow next tamper event detection 1:Tamper 0 event generates a trigger event. TP0F is masked and internally cleared by hardware. The backup registers are not erased

		Note: The Tamper 0 interrupt must not be enabled when TP0MASK is set.
22	Reserved	Must be kept at reset value.
21	TP2NOERASE	Tamper 2 no erase: 0:Tamper 2 event erases the backup registers 1:Tamper 2 event does not erase the backup registers
20	TP1NOERASE	Tamper 1 no erase: 0:Tamper 1 event erases the backup registers 1:Tamper 1 event does not erase the backup registers
19	TP0NOERASE	Tamper 0 no erase: 0:Tamper 0 event erases the backup registers 1:Tamper 0 event does not erase the backup registers
18	ALRMOUTTYPE	RTC_ALARM Output Type 0: Open-drain output type 1: Push-pull output type
16:17	Reserved	Must be kept at reset value.
15	DISPU	RTC_TAMPx pull up disable bit 0: Enable inner pull-up before sampling for pre-charge RTC_TAMPx pin 1: Disable pre-charge duration
14:13	PRCH[1:0]	Pre-charge duration time of RTC_TAMPx This setting determines the pre-charge time before each sampling. 0x0: 1 RTC clock 0x1: 2 RTC clock 0x2: 4 RTC clock 0x3: 8 RTC clock
12:11	FLT[1:0]	RTC_TAMPx filter count setting This bit determines the tamper sampling type and the number of consecutive sample. 0x0: Detecting tamper event using edge mode. Pre-charge duration is disabled automatically 0x1: Detecting tamper event using level mode.2 consecutive valid level samples will make an effective tamper event 0x2: Detecting tamper event using level mode.4 consecutive valid level samples will make an effective tamper event 0x3: Detecting tamper event using level mode.8 consecutive valid level samples will make an effective tamper event
10:8	FREQ[2:0]	Sampling frequency of tamper event detection 0x0: Sample once every 32768 RTCCLK(1Hz if RTCCLK=32.768KHz) 0x1: Sample once every 16384 RTCCLK(2Hz if RTCCLK=32.768KHz) 0x2: Sample once every 8192 RTCCLK(4Hz if RTCCLK=32.768KHz)

		0x3: Sample once every 4096 RTCCLK(8Hz if RTCCLK=32.768KHz) 0x4: Sample once every 2048 RTCCLK(16Hz if RTCCLK=32.768KHz) 0x5: Sample once every 1024 RTCCLK(32Hz if RTCCLK=32.768KHz) 0x6: Sample once every 512 RTCCLK(64Hz if RTCCLK=32.768KHz) 0x7: Sample once every 256 RTCCLK(128Hz if RTCCLK=32.768KHz)
7	TPTS	Make tamper function used for timestamp function 0:No effect 1:TSF is set when tamper event detected even TSEN=0
6	TP2EG	Tamper 2 event trigger edge If tamper detection is in edge mode(FLT =0): 0: Rising edge triggers a tamper detection event 1: Falling edge triggers a tamper detection event If tamper detection is in level mode(FLT !=0): 0: Low level triggers a tamper detection event 1: High level triggers a tamper detection event
5	TP2EN	Tamper 2 detection enable 0:Disable tamper 2 detection function 1:Enable tamper 2 detection function
4	TP1EG	Tamper 1 event trigger edge If tamper detection is in edge mode(FLT =0): 0: Rising edge triggers a tamper detection event 1: Falling edge triggers a tamper detection event If tamper detection is in level mode(FLT !=0): 0: Low level triggers a tamper detection event 1: High level triggers a tamper detection event
3	TP1EN	Tamper 1 detection enable 0: Disable tamper 1 detection function 1: Enable tamper 1 detection function
2	TPIE	Tamper detection interrupt enable 0: Disable tamper interrupt 1: Enable tamper interrupt
1	TP0EG	Tamper 0 event trigger edge If tamper detection is in edge mode(FLT =0): 0: Rising edge triggers a tamper detection event 1: Falling edge triggers a tamper detection event If tamper detection is in level mode(FLT !=0): 0: Low level triggers a tamper detection event 1: High level triggers a tamper detection event
0	TP0EN	Tamper 0 detection enable 0:Disable tamper 0 detection function

1:Enable tamper 0 detection function

Note: It's strongly recommended that reset the TpxEN before change the tamper configuration.

17.4.17. Alarm 0 sub second register (RTC_ALRM0SS)

Address offset: 0x44

Backup domain 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.

17.4.18. Alarm 1 sub second register (RTC_ALRM1SS)

Address offset: 0x48

Backup domain reset: 0x0000 0000

System reset: no effect

This register is write protected and can only be wrote when ALRM1EN=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.

17.4.19. Backup registers (RTC_BKPx) (x=0..4)

Address offset: 0x50~0x64

Backup domain reset: 0x0000 0000

System reset: no effect

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

Bits	Fields	Descriptions
31:0	DATA[31:0]	Data These registers can be wrote or read by software. The content remains valid even in power saving mode because they can powered-on by VBAT. Tamper detection flag TPxF assertion will reset these registers.

18. Timer (TIMERx)

Table 18-1. Timers (TIMERx) are devided into five sorts

TIMER	TIMER0	TIMER1/2	TIMER8/11	TIMER14/40	TIMER5/6
TYPE	Advanced	General-L0	General-L1	General-L3	Basic
Prescaler	16-bit	16-bit	16-bit	16-bit	16-bit
Counter	16-bit	16-bit	16-bit	16-bit	16-bit
Count mode	UP,DOWN, Center-aligned	UP,DOWN, Center-aligned	UP ONLY	UP ONLY	UP ONLY
Repetition	●	×	×	●	×
CH Capture/ Compare	4	4	2	2	0
Complementary & Dead-time	3	×	×	1	×
Break	●	×	×	●	×
Single Pulse	●	●	●	●	●
Quadrature Decoder	●	●	×	×	×
Master-slave management	●	●	●	●	×
Inter connection	●(1)	●(2)	●(3)	●(4)	TRGO TO DAC
DMA	●	●	×	●	●(5)
Debug Mode	●	●	●	●	●

(1)	TIMER0	**ITI0:** TIMER14_TRGO	**ITI1:** TIMER1_TRGO	**ITI2:** TIMER2_TRGO	**ITI3:** TIMER40_TRGO
(2)	TIMER1	**ITI0:** TIMER2_TRGO	**ITI1:** TIMER14_TRGO(6)	**ITI2:** TIMER40_TRGO(6)	**ITI3:** TIMER0_TRGO(6)
	TIMER2	**ITI0:** TIMER1_TRGO	**ITI1:** TIMER0_TRGO(6)	**ITI2:** TIMER14_TRGO(6)	**ITI3:** TIMER40_TRGO(6)
(3)	TIMER8	**ITI0:** TIMER1_TRGO	**ITI1:** TIMER2_TRGO	**ITI2:** TIMER0_TRGO(6)	**ITI3:** TIMER14_TRGO(6)
	TIMER11	**ITI0:** TIMER0_TRGO(6)	**ITI1:** TIMER1_TRGO	**ITI2:** TIMER2_TRGO	**ITI3:** TIMER14_TRGO(6)
(4)	TIMER14	**ITI0:** TIMER1_TRGO	**ITI1:** TIMER2_TRGO	**ITI2:** TIMER0_TRGO	**ITI3:** TIMER40_TRGO
	TIMER40	**ITI0:** TIMER1_TRGO	**ITI1:** TIMER2_TRGO	**ITI2:** TIMER14_TRGO	**ITI3:** TIMER0_TRGO

(5) Only update events will generate a DMA request. TIMER5/6 do not have DMAS bit (DMA request source selection).

(6) Only valid for GD32L235

18.1. Advanced timer (TIMERx,x=0)

18.1.1. 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.

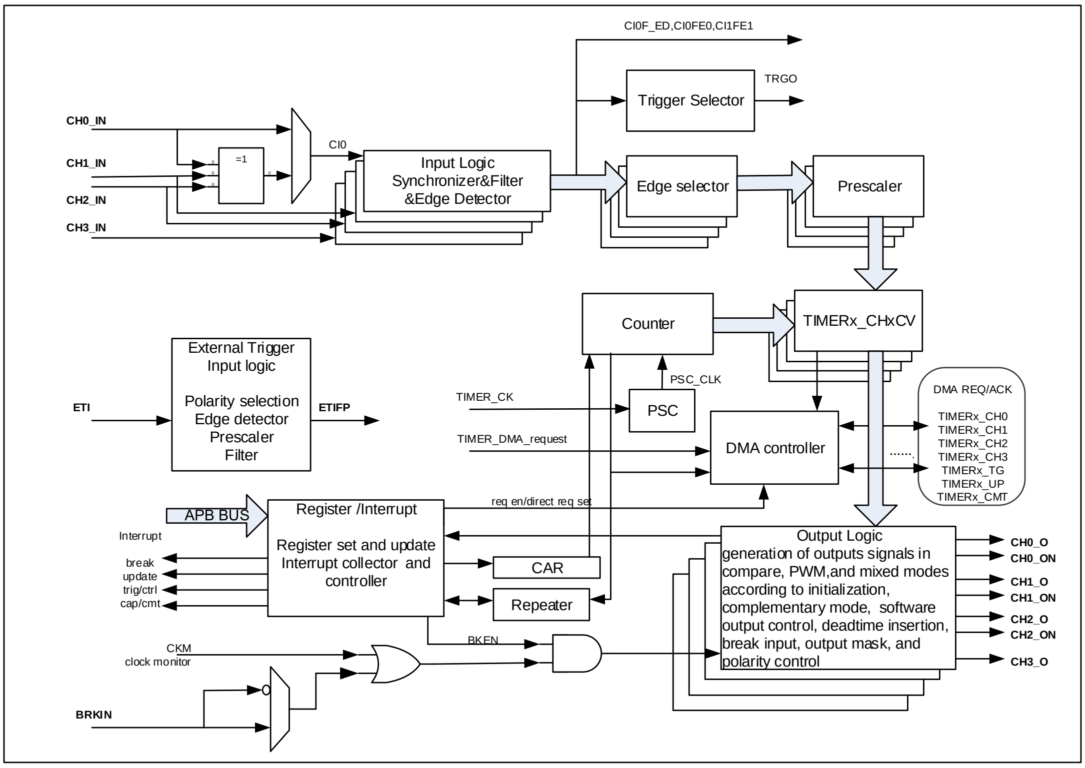
18.1.2. Characteristics

- 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.

18.1.3. Block diagram

Figure 18-1. Advanced timer block diagram provides details of the internal configuration of the advanced timer.

Figure 18-1. Advanced timer block diagram



18.1.4. 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[3:0] in SYSCFG_TIMER0CFG(y=0,1…7).

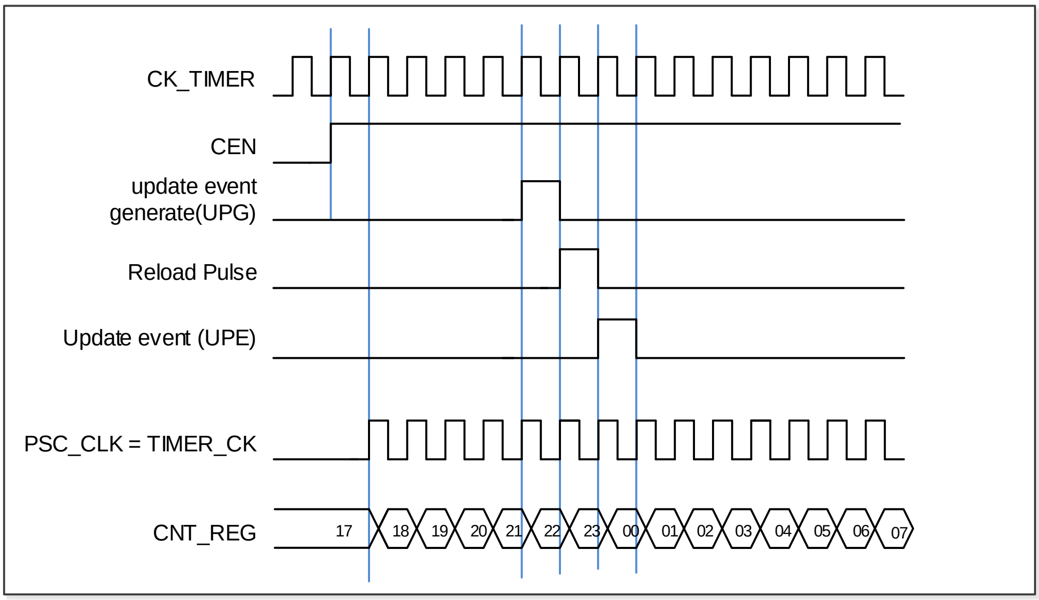
- TSCFGy[3:0] = 4'b0000 in SYSCFG_TIMER0CFG(y=0,1…7). Internal clock CK_TIMER is selected as timer clock source which is from module RCU.

The default clock source is the CK_TIMER for driving the counter prescaler when TSCFGy[3:0] = 4'b0000 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[3:0] !=4'b0000 in SYSCFG_TIMER0CFG(y=0,1,2,6), the prescaler is clocked by other clock sources selected in the TSCFG6[3:0] register, more details will be introduced later. When the TSCFGy[3:0] (y=3,4,5) are setting to an available value, the internal clock TIMER_CK is the counter prescaler driving clock source.

Figure 18-2. Normal mode, internal clock divided by 1



- TSCFG6[3:0] 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[3:0] to 0x5, 0x6, 0x7.

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 TSCFG6[3:0] to 0x1,0x2,0x3,0x4.

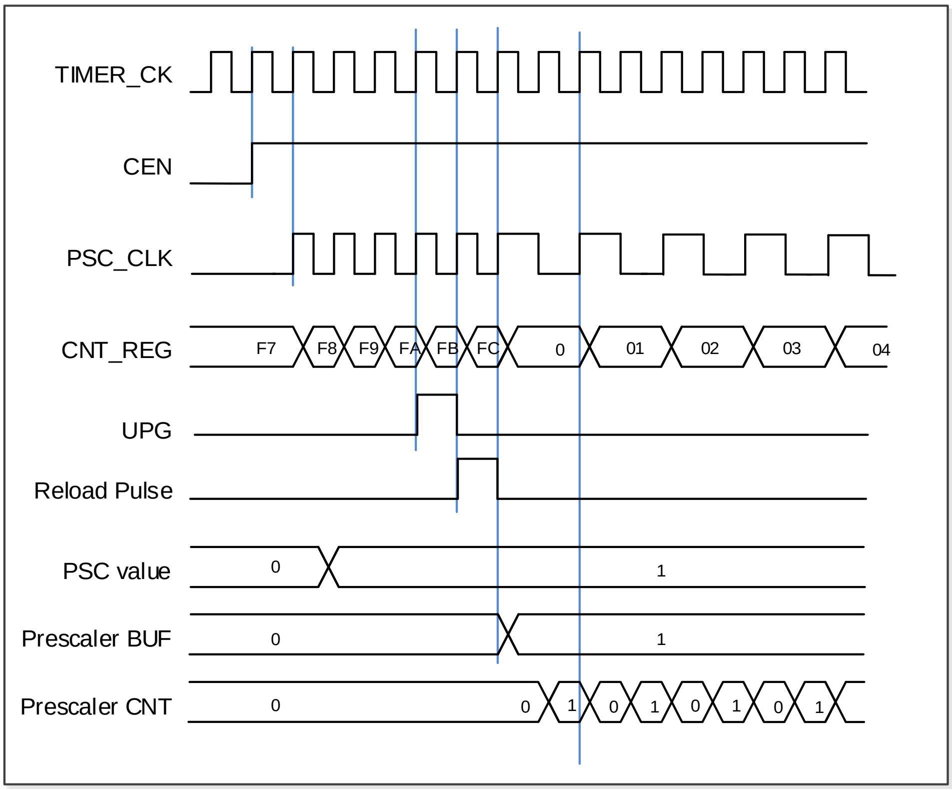
- SMC1= 1'b1 (external clock mode 1). External input ETI is selected as timer clock source.

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[3:0] to 0x8. 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 18-3. 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 18-4. Timing chart of up counting mode, PSC=0/1 and ***Figure 18-5. 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 18-4. Timing chart of up counting mode, PSC=0/1

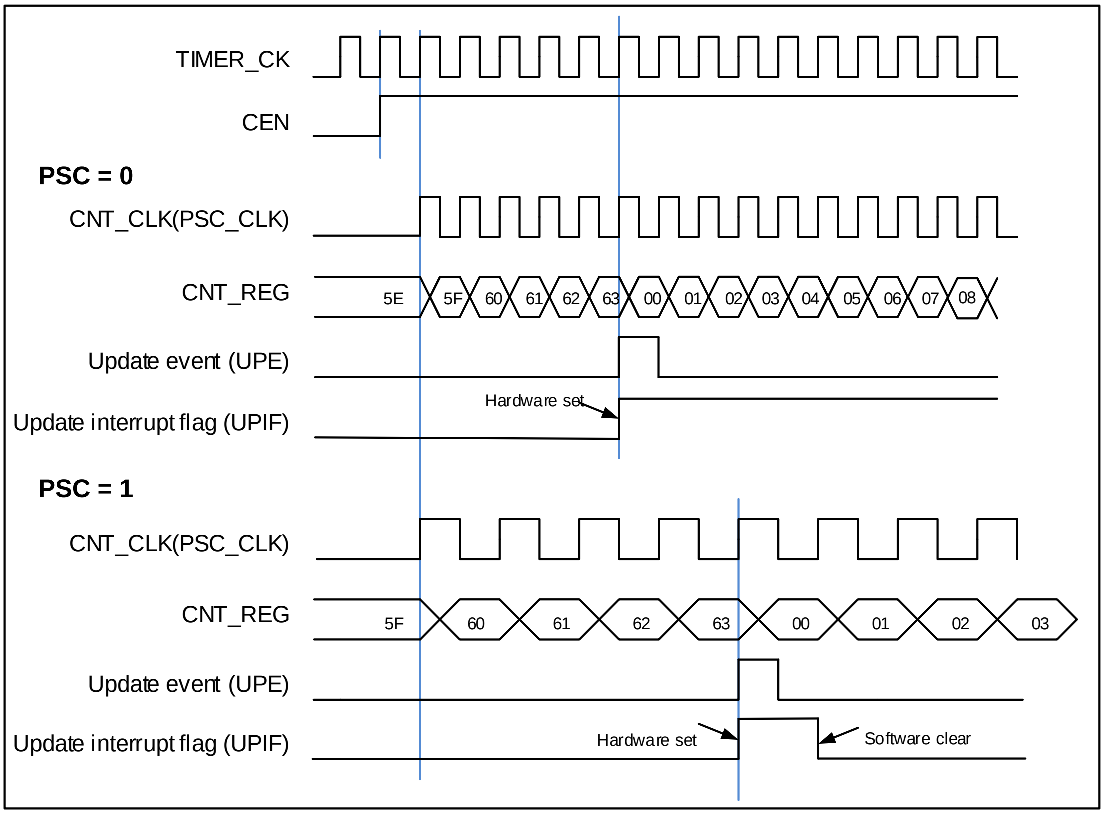
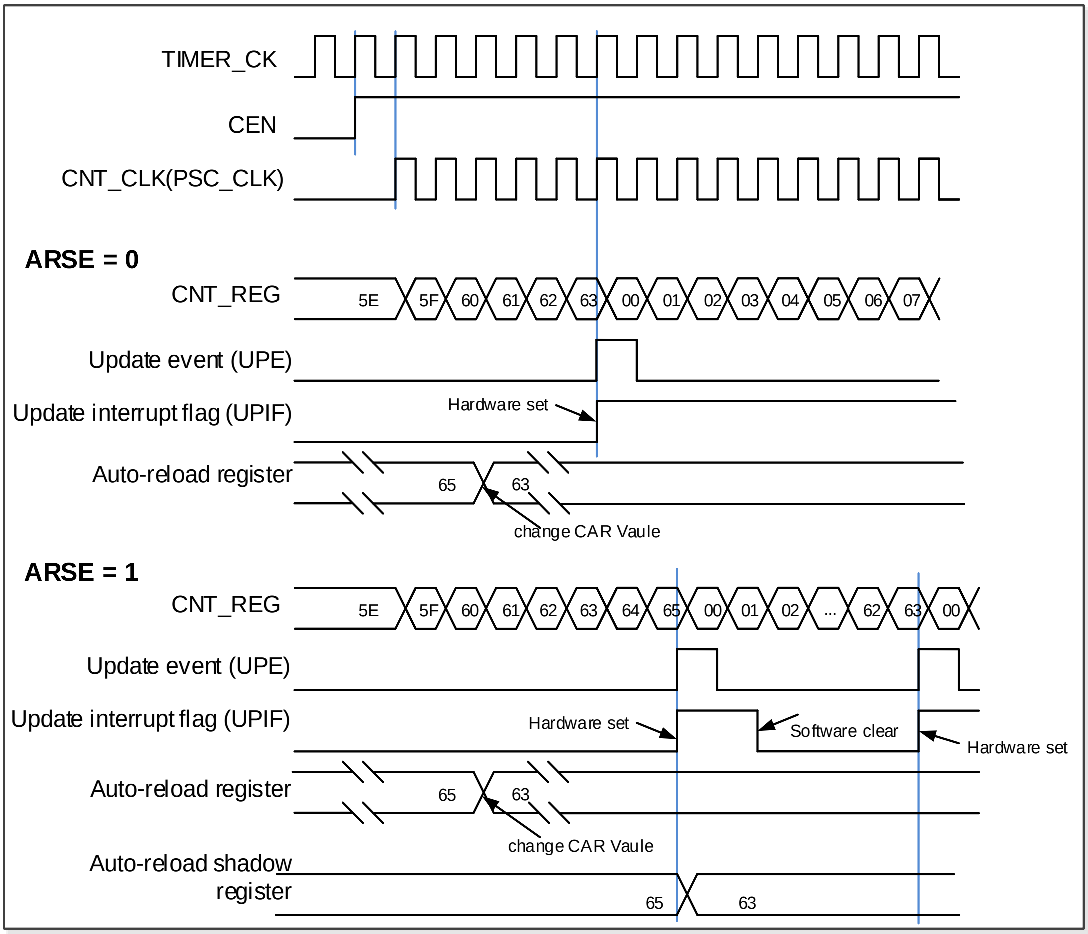


Figure 18-5. Timing chart of up counting mode, change TIMERx_CAR ongoing



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 18-6. Timing chart of down counting mode, PSC=0/1 and Figure 18-7. 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 18-6. Timing chart of down counting mode, PSC=0/1

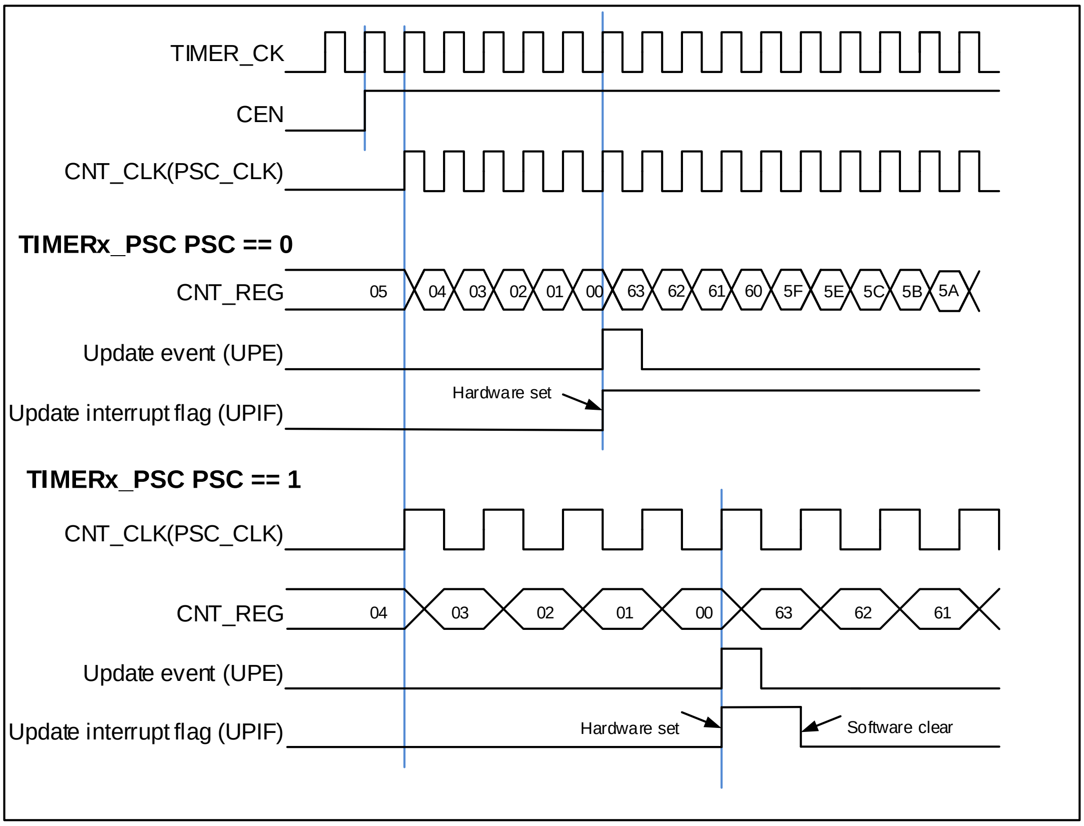
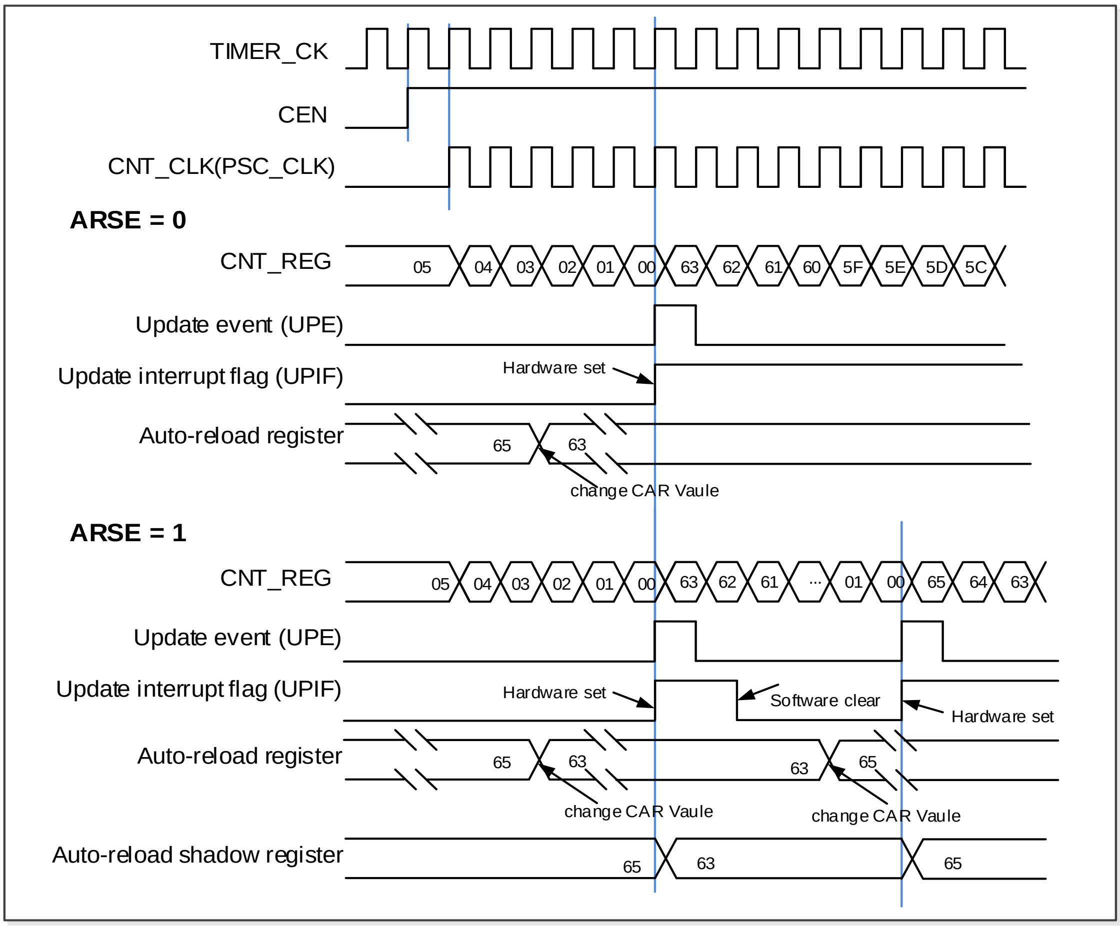


Figure 18-7. Timing chart of down counting mode, change TIMERx_CAR ongoing



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.

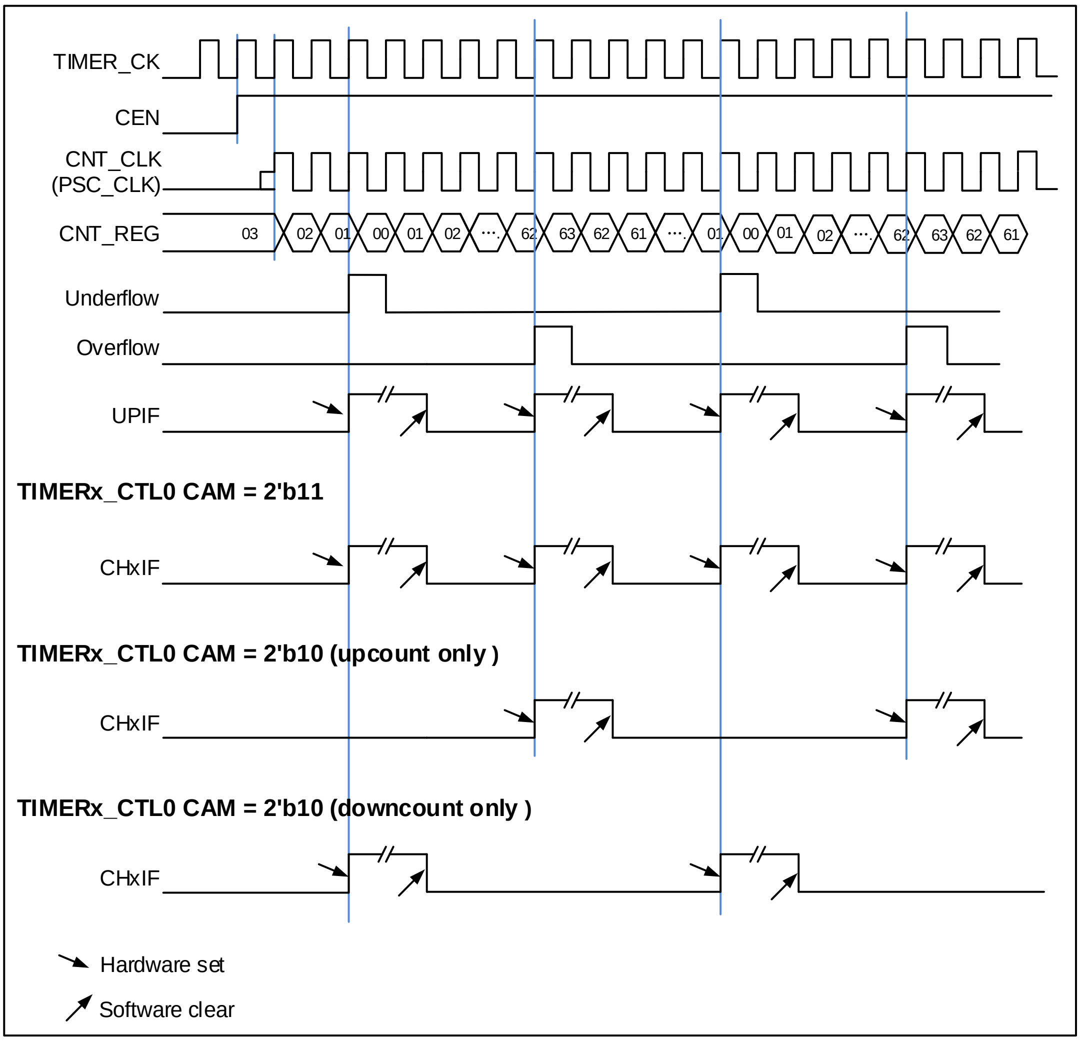
The UPIF bit in the TIMERx_INTF register will be set to 1 either when an underflow event or an overflow event occurs. While the ChxIF bit is associated with the value of CAM in TIMERx_CTL0. The details refer to *Figure 18-8. Timing chart of center-aligned counting* .

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 18-8. Timing chart of center-aligned counting shows some examples of the counter behavior when TIMERx_CAR=0x63. TIMERx_PSC=0x0.

Figure 18-8. Timing chart of center-aligned counting mode



Repetition counter

Repetition counter 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 18-9. Repetition counter timing chart of center-aligned counting mode

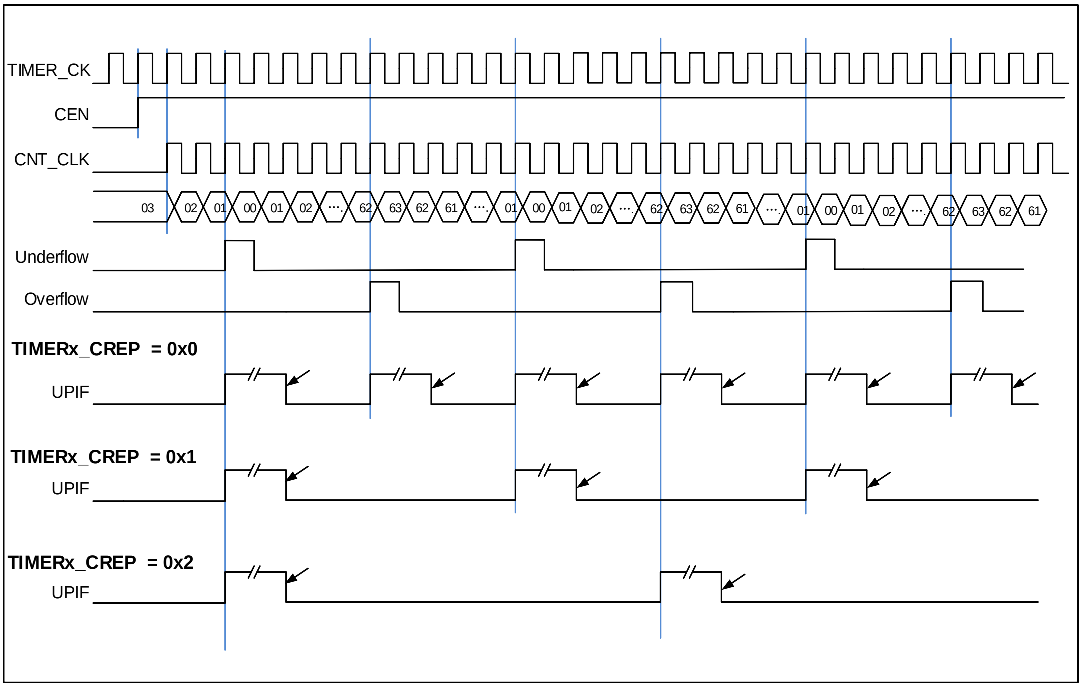


Figure 18-10. Repetition counter timing chart of up counting mode

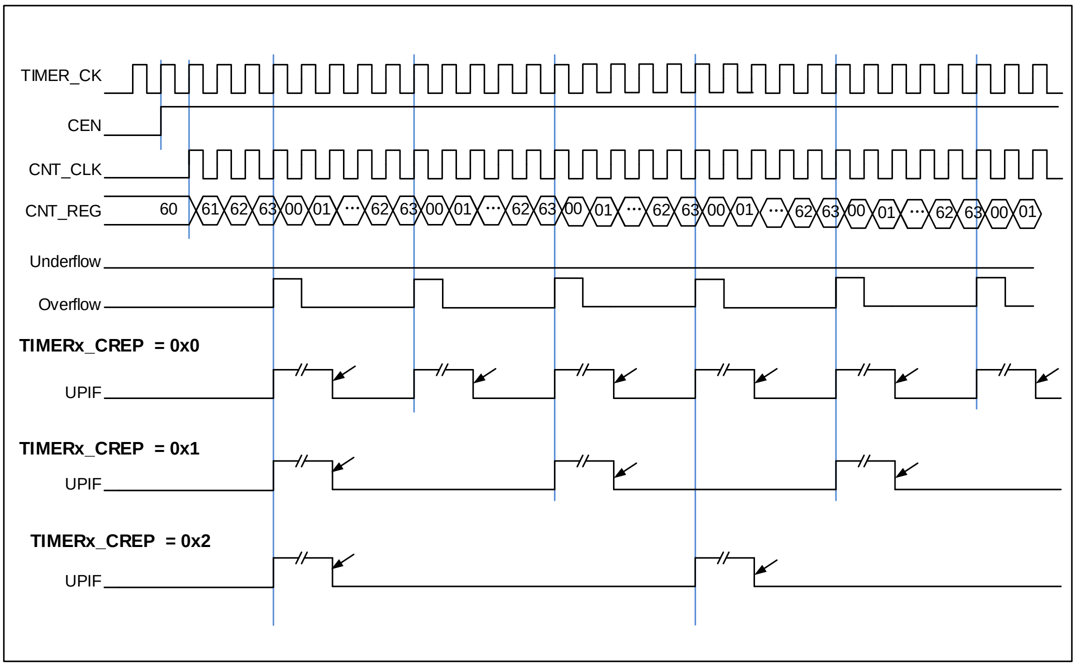
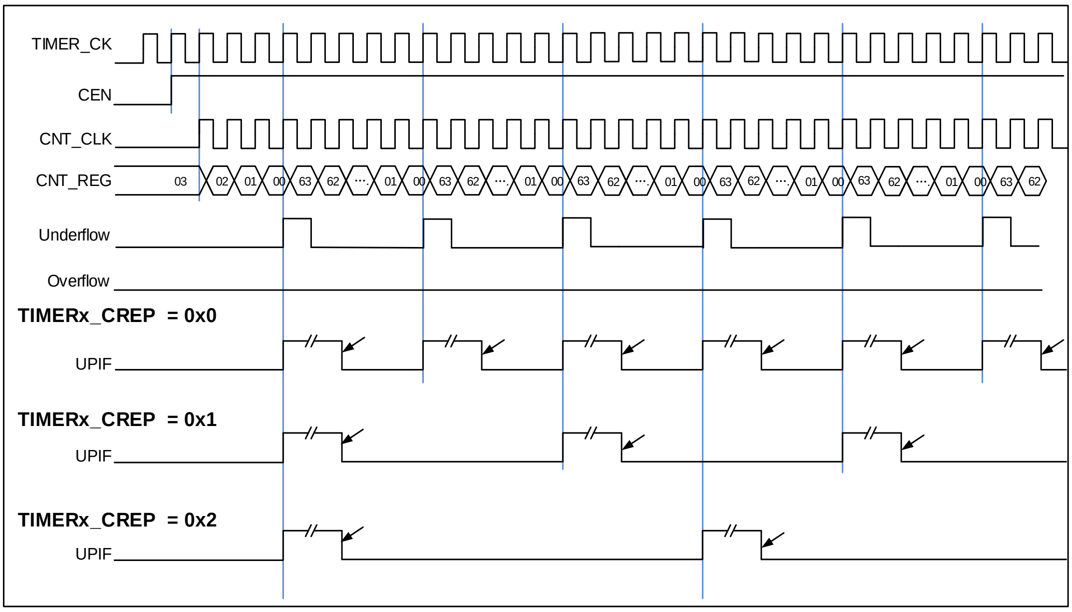


Figure 18-11. Repetition counter timing chart of down counting mode



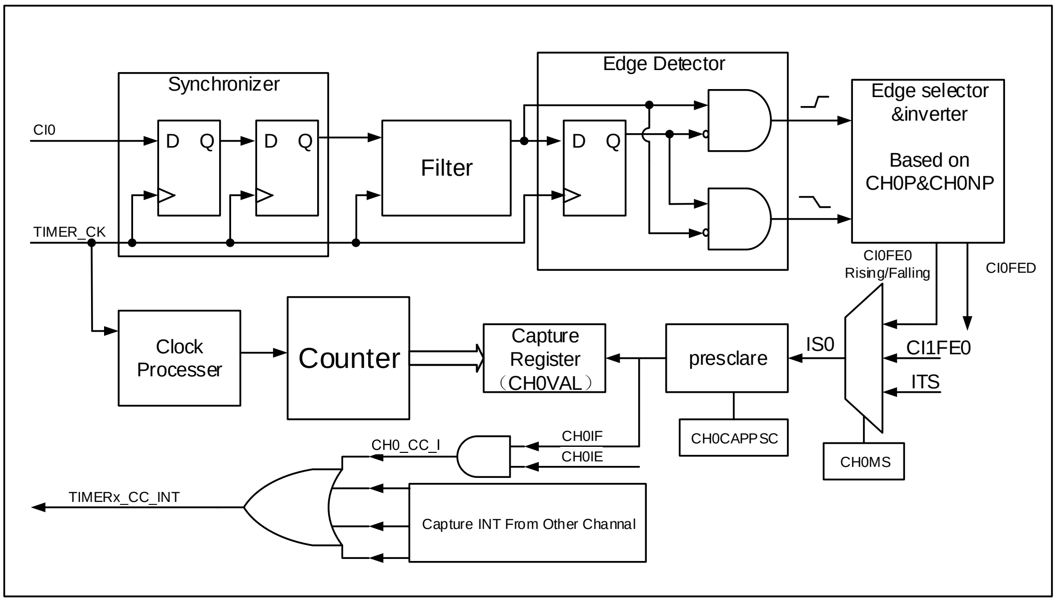
Input capture and output compare channels

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 18-12. 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 18-13. Output compare logic (with complementary output, x=0,1,2)

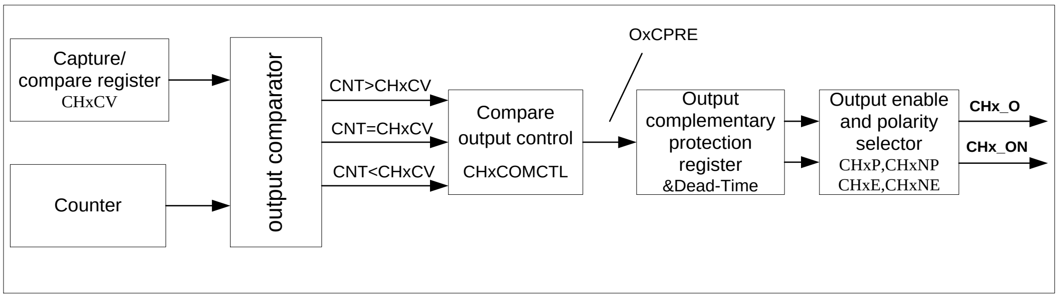


Figure 18-14. Output compare logic (CH3_O)

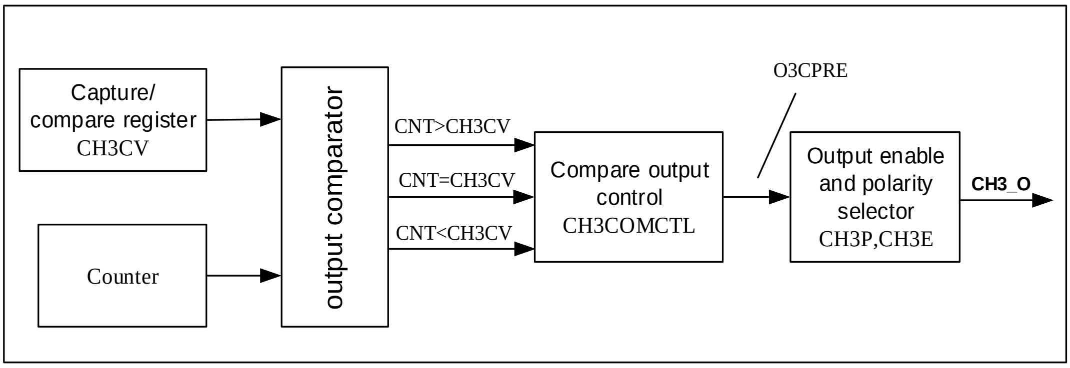


Figure 18-13. Output compare logic (with complementary output, x=0,1,2) and

Figure 18-14. 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). 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 ***Complementary outputs*** 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.

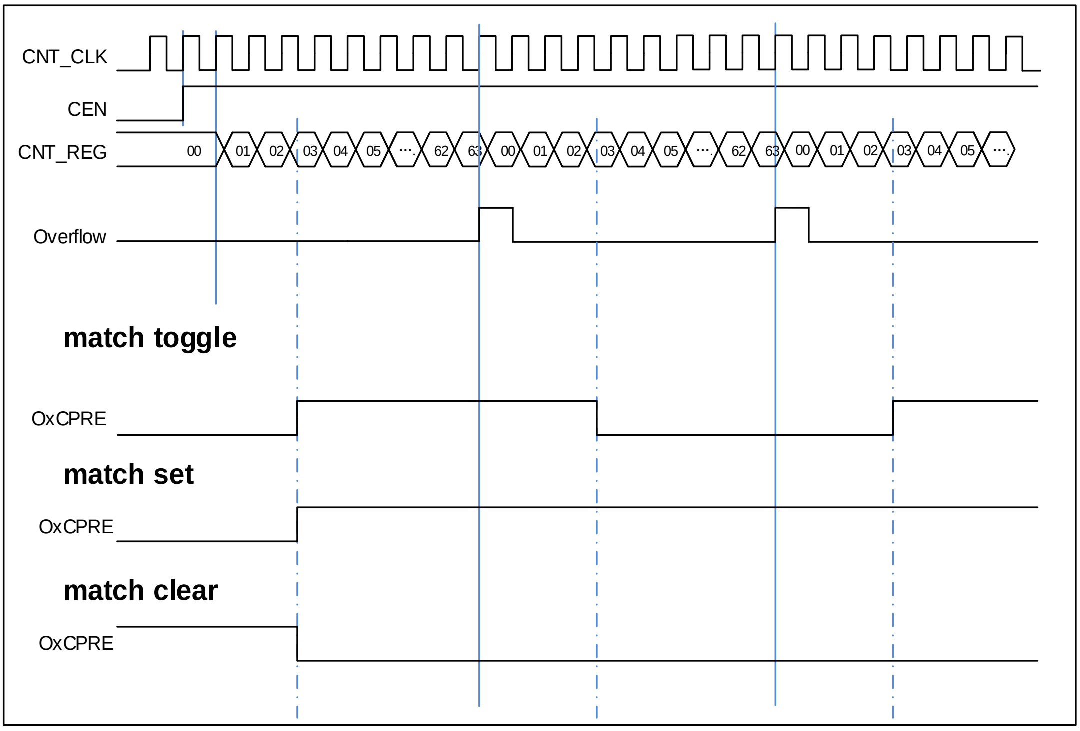
Step3: Interrupt/DMA request enable configuration by ChxIE/CxCDE.

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 18-15. Output-compare in three modes shows the three compare modes: toggle/set/clear. CAR=0x63, CHxVAL=0x3.

Figure 18-15. 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 18-16. 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 18-17. 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 18-16. Timing chart of EAPWM

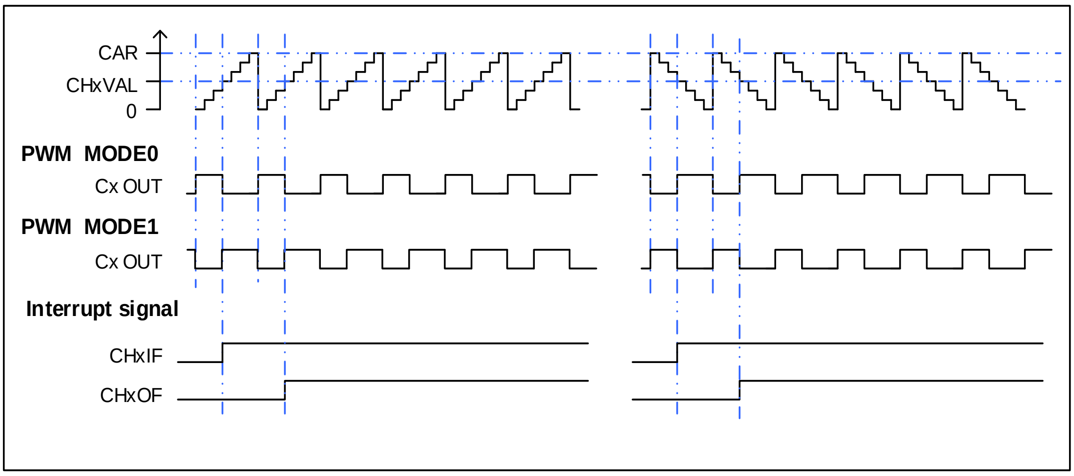
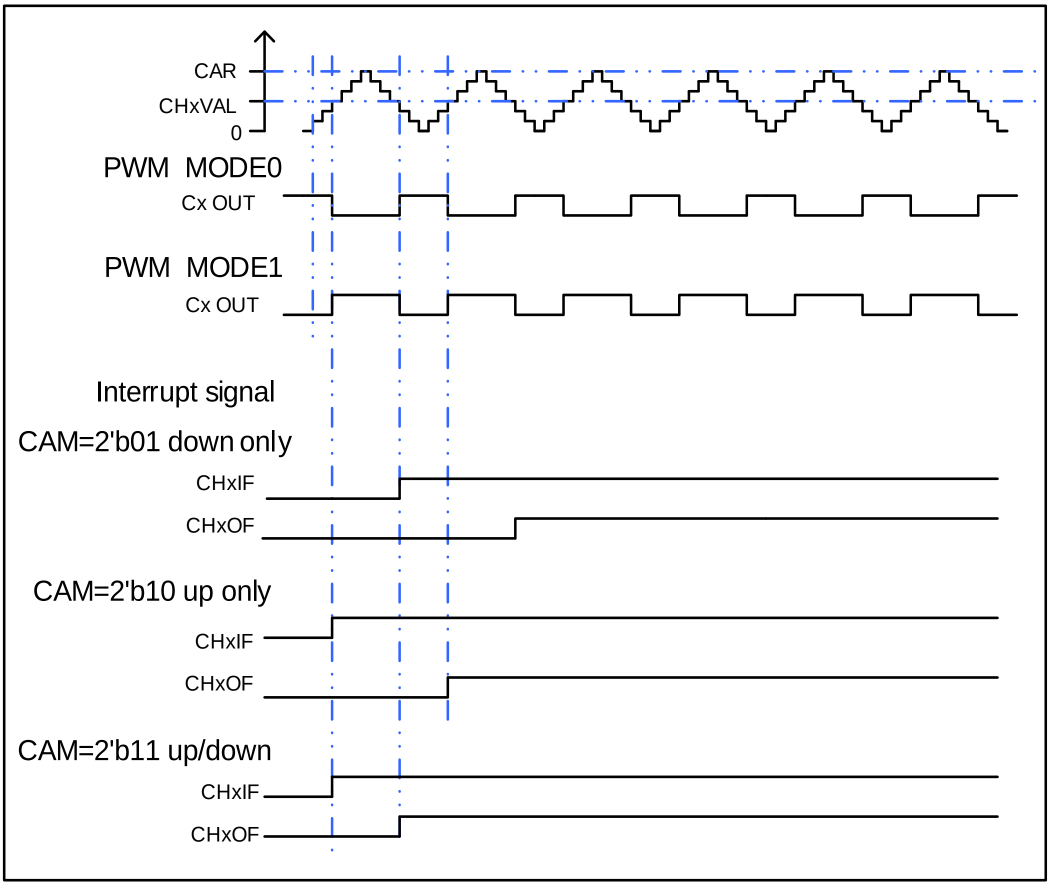


Figure 18-17. Timing chart of CAPWM



Channel output prepare signal

As is shown in *Figure 18-13. 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.

Complementary outputs

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 4 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 18-2. Complementary outputs controlled by parameters

<table>
<tr><th colspan="5">Complementary Parameters</th><th colspan="2">Output Status</th></tr>
<tr><th>POEN</th><th>ROS</th><th>IOS</th><th>ChxEN</th><th>CHxNEN</th><th>CHx_O</th><th>CHx_ON</th></tr>
<tr><td rowspan="5">0</td><td rowspan="5">0/1</td><td rowspan="4">0</td><td rowspan="2">0</td><td>0</td><td colspan="2">CHx_O / CHx_ON = LOW
CHx_O / CHx_ON output disable[1].</td></tr>
<tr><td>1</td><td colspan="2" rowspan="3">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]</td></tr>
<tr><td rowspan="2">1</td><td>0</td></tr>
<tr><td>1</td></tr>
<tr><td>1</td><td>x</td><td>x</td><td colspan="2">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.</td></tr>
<tr><td rowspan="8">1</td><td rowspan="4">0</td><td rowspan="8">0/1</td><td rowspan="2">0</td><td>0</td><td colspan="2">CHx_O/CHx_ON = LOW
CHx_O/CHx_ON output disable.</td></tr>
<tr><td>1</td><td>CHx_O = LOW
CHx_O output disable.</td><td>CHx_ON =OxCPRE⊕[4]CHxNP
CHx_ON output enable.</td></tr>
<tr><td rowspan="2">1</td><td>0</td><td>CHx_O=OxCPRE⊕CHxP
CHx_O output enable.</td><td>CHx_ON = LOW
CHx_ON output disable.</td></tr>
<tr><td>1</td><td>CHx_O=OxCPRE⊕CHxP
CHx_O output enable.</td><td>CHx_ON =(!OxCPRE)[5]⊕ CHxNP.
CHx_ON output enable.</td></tr>
<tr><td rowspan="4">1</td><td rowspan="2">0</td><td>0</td><td>CHx_O = CHxP
CHx_O output “off-state”.</td><td>CHx_ON = CHxNP
CHx_ON output “off-state”.</td></tr>
<tr><td>1</td><td>CHx_O = CHxP
CHx_O output “off-state”</td><td>CHx_ON =OxCPRE⊕CHxNP
CHx_ON output enable</td></tr>
<tr><td rowspan="2">1</td><td>0</td><td>CHx_O=OxCPRE⊕CHxP
CHx_O output enable</td><td>CHx_ON = CHxNP
CHx_ON output “off-state”.</td></tr>
<tr><td>1</td><td>CHx_O=OxCPRE⊕CHxP
CHx_O output enable</td><td>CHx_ON =(!OxCPRE)⊕ CHxNP
CHx_ON output enable.</td></tr>
</table>

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.

Dead time insertion

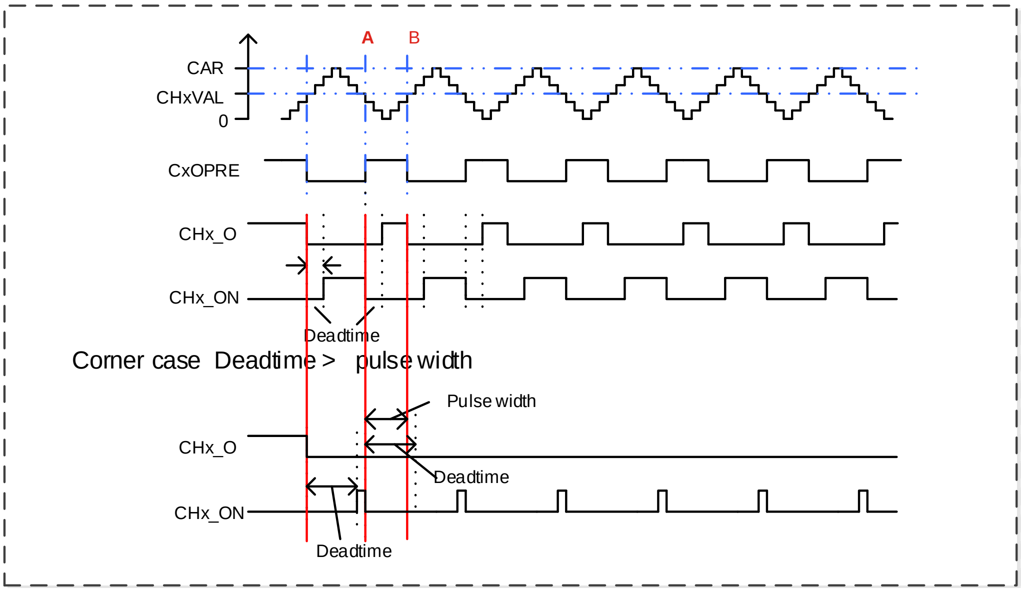
The dead time insertion is enabled when both ChxEN and CHxNEN are configured to 1'b1, it is also necessary to configure POEN to 1. The field named DTCFG defines the dead time delay that can be used for all channels except channel 3. Refer to the TIMERx_CCHP register for details about the delay time.

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 18-18. 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 18-18. Complementary output with dead time insertion)***.

Figure 18-18. Complementary output with dead time insertion



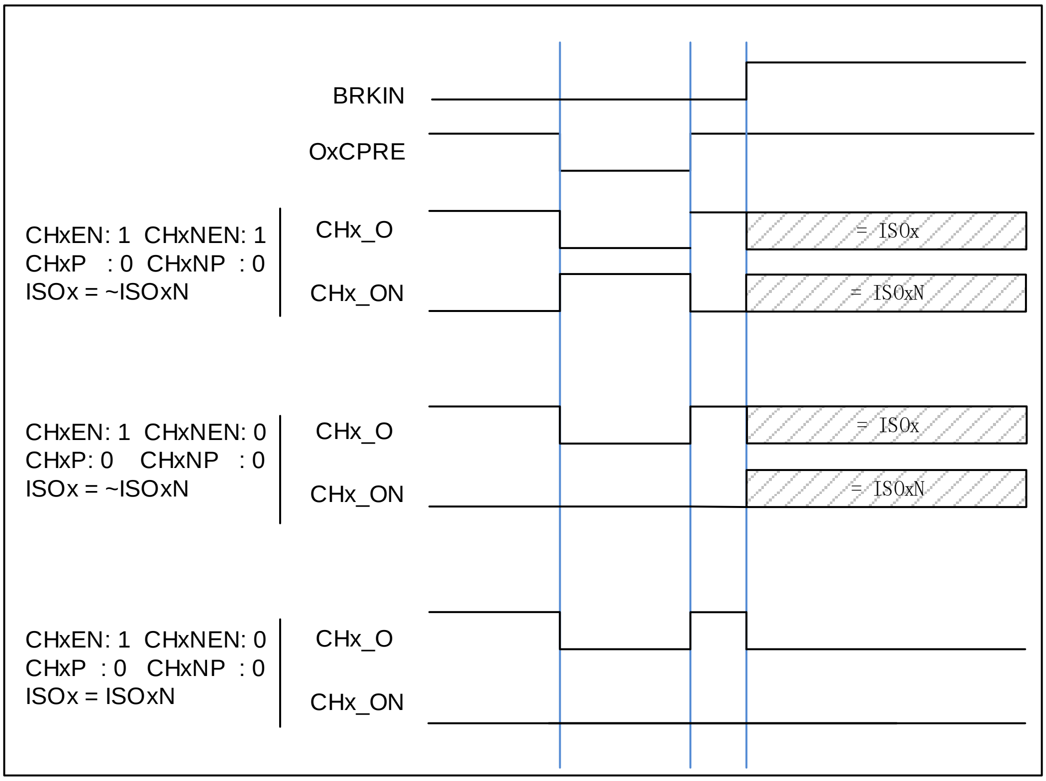
Break function

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. In any case, CHx_O and CHx_ON signals cannot be set to active level at the same time. The break sources are input break pin and HXTAL stuck event which is generated by Clock Monitor (CKM) in RCU. The break function is enabled by setting the BRKEN bit in the TIMERx_CCHP register. The break input polarity is configured by the BRKP bit in TIMERx_CCHP register.

When a break occurs, the POEN bit is cleared asynchronously. As soon as POEN is 0, the level of the CHx_O and CHx_ON outputs are determined by the ISOx and ISOxN bits in the TIMERx_CTL1 register. If IOS is 0, the timer releases the enable output, otherwise, the enable output remains high. The complementary outputs are first in the reset state, and then the dead time generator is reactivated 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 will be set. If BRKIE is 1, an interrupt will be generated.

Figure 18-19. 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[3:0] != 4'b0000 (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 18-3. 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 18-3. 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
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 18-20. Example of counter operation in quadrature decoder interface mode

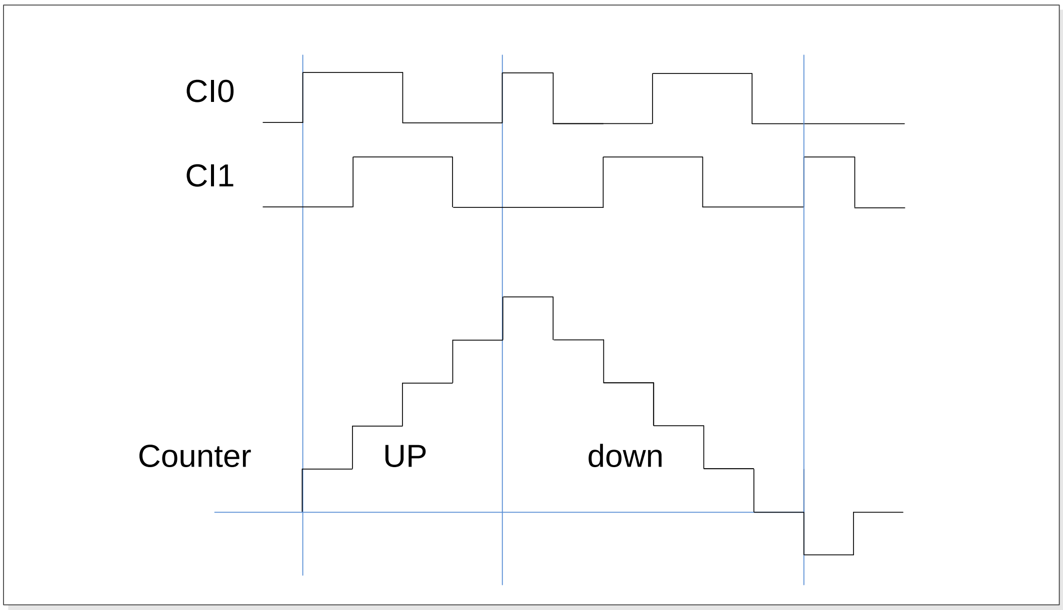
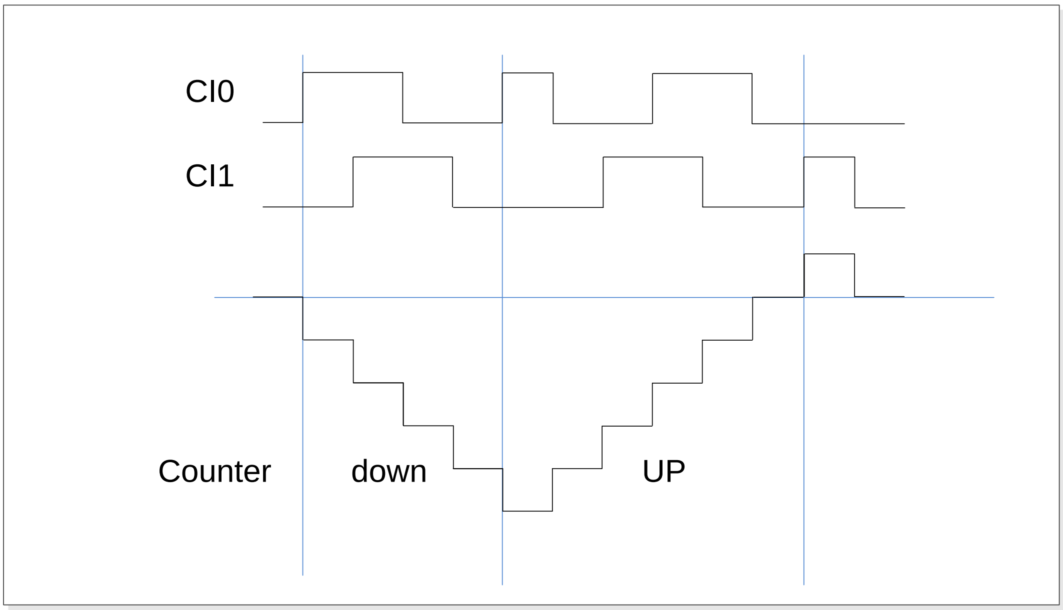


Figure 18-21. Example of quadrature decoder interface mode with CI0FE0 polarity

inverted



Hall sensor function

Hall sensor is generally used to control BLDC Motor; advanced timer can support this function.

Figure 18-22. 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[3:0] 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 18-22. Hall sensor is used to BLDC motor

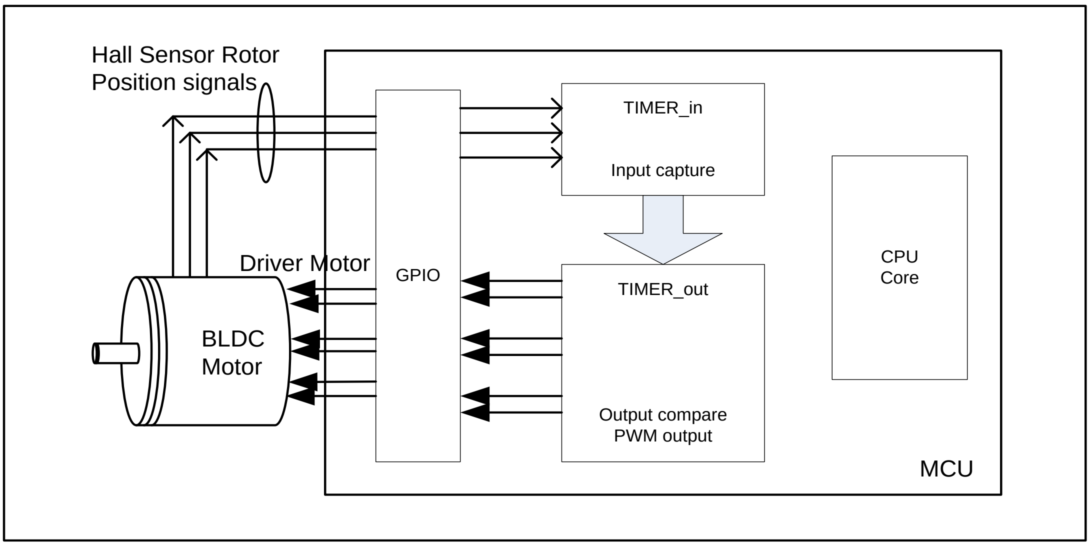
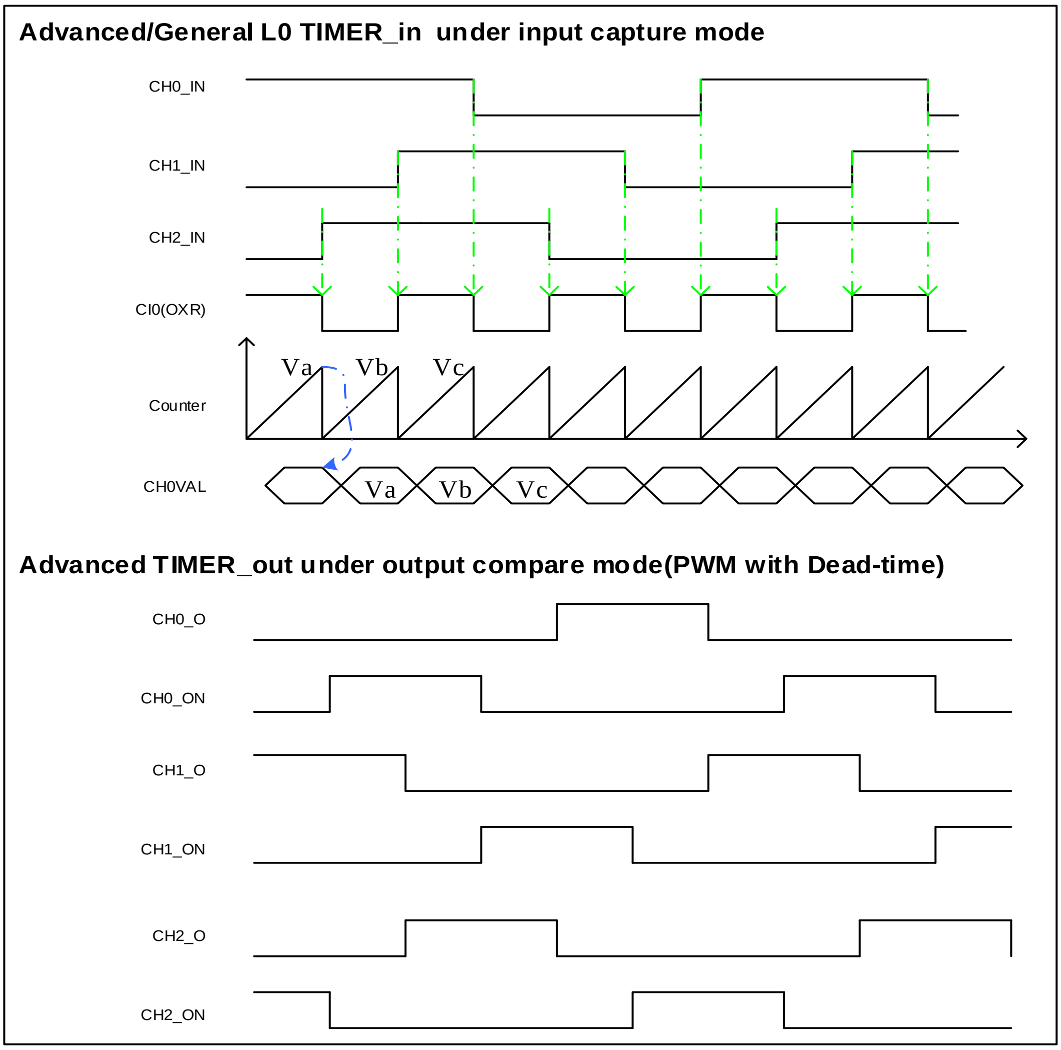


Figure 18-23. Hall sensor timing between two timers

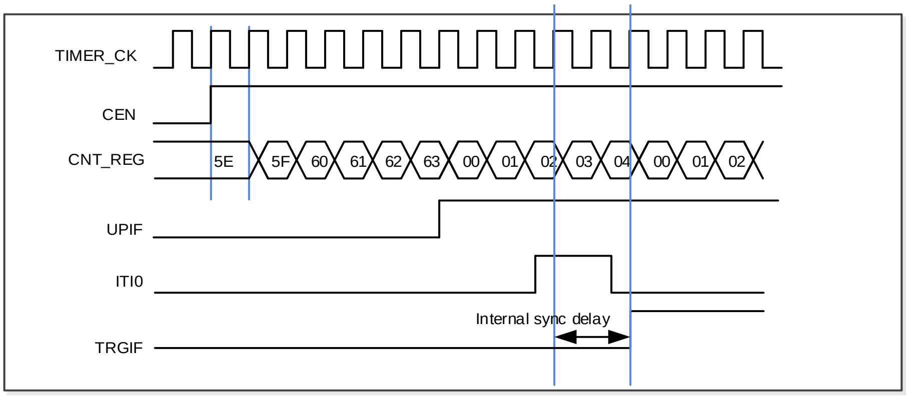


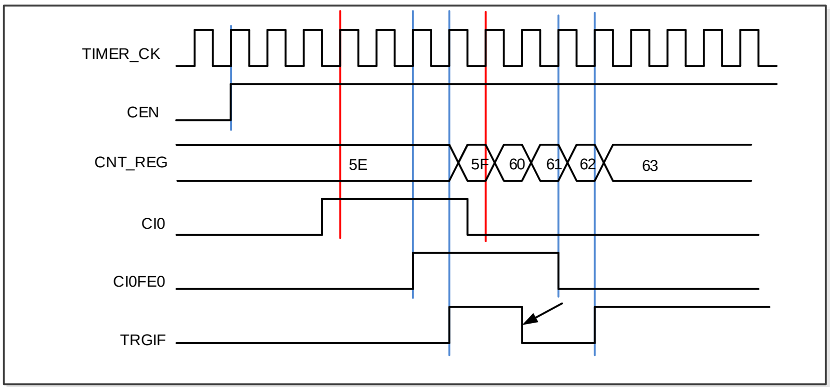
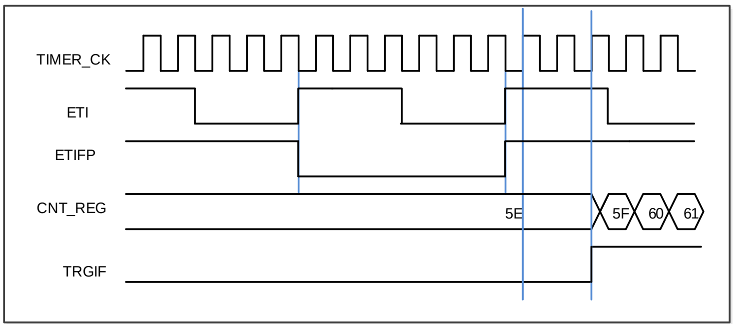
Master-slave management

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[3:0] in SYSCFG_TIMER0CFG(x=3,4,5).

Table 18-4. Examples of slave mode

	Mode Selection	Source Selection	Polarity Selection	Filter and Prescaler
LIST	TSCFGy[3:0] y=3 (restart mode) y=4 (pause mode) y=5 (event mode)	TSCFGy[3:0] 0001: ITI0 0010: ITI1 0011: ITI2 0100: ITI3 0101: CI0F_ED 0110: CI0FE0 0111: CI1FE1	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,	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

	Mode Selection	Source Selection	Polarity Selection	Filter and Prescaler
		1000: ETIFP	configure the ETP for polarity selection and inversion.	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[3:0] = 4’b 0001.ITI0 is selected.	For ITI0, no polarity selector can be used.	For the ITI0, no filter and prescaler can be used.
	Figure 18-24. Restart mode 			
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[3:0] = 4’b 0110 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.

<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 18-25. Pause mode

</td></tr>
<tr><td>Exam3</td><td>Event mode
The counter will start to count when a rising edge of trigger input comes.</td><td>TSCFG5[3:0] = 4’b 1000
ETIFP is selected.</td><td>ETP = 0, the polarity of ETI does not change.</td><td>ETPSC = 1, ETI is divided by 2.
ETFC = 0, ETI does not filter.</td></tr>
<tr><td></td><td colspan="4">Figure 18-26. Event mode

</td></tr>
</table>

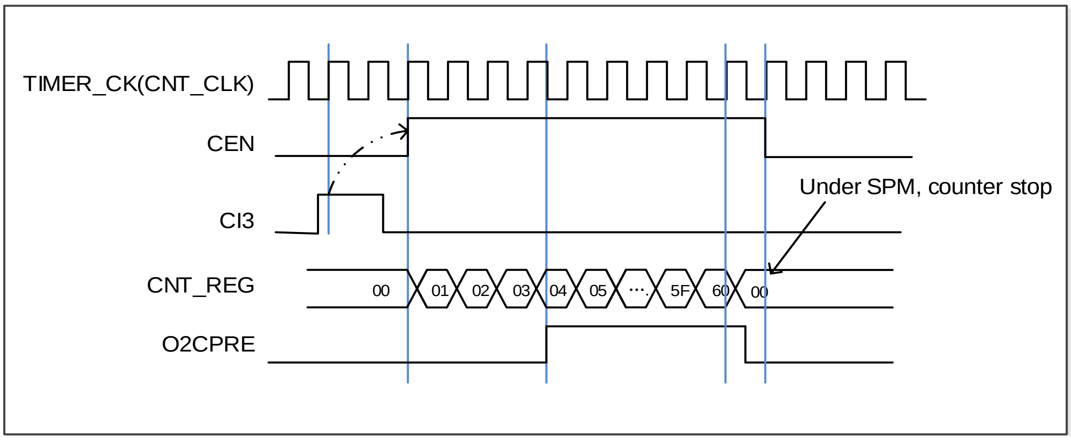
Single pulse mode

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 18-27. 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[3:0] = 0001 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).

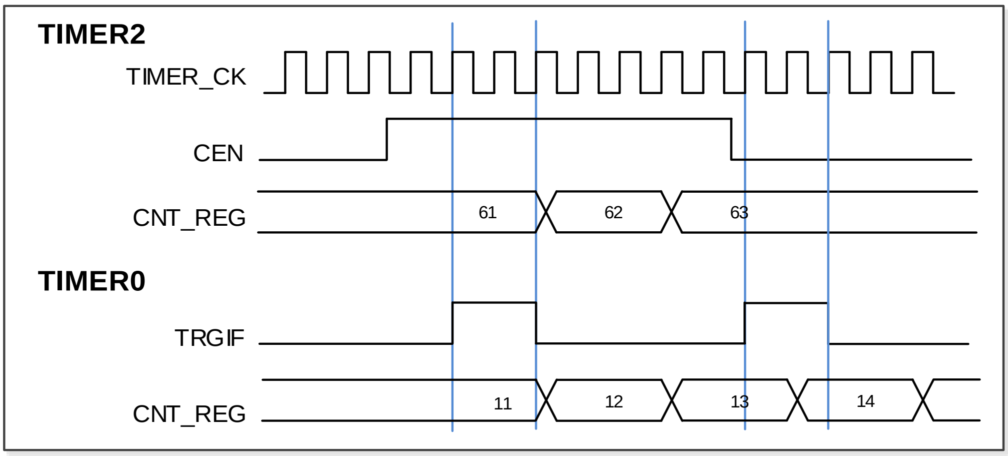
■ Start TIMER0 with TIMER2's Enable/Update signal

First, we enable TIMER0 with the enable out of TIMER2. Refer to ***Figure 18-28. 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[3:0] = 0011 in the SYSCFG_TIMERxCFG register).
2. Start TIMER2 by writing 1 in the CEN bit (TIMER2_CTL0 register).

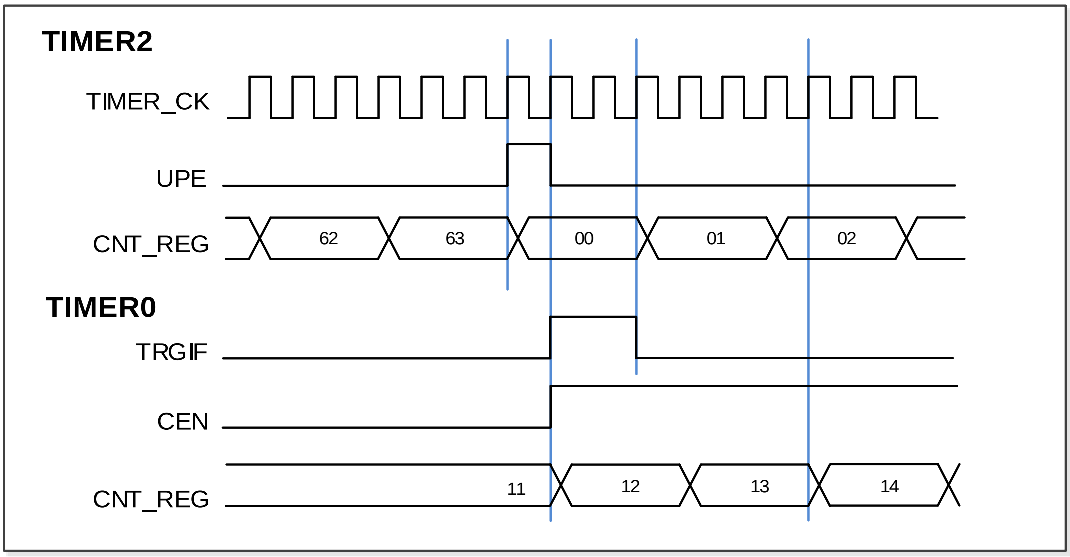
Figure 18-28. Triggering TIMER0 with Enable of TIMER2



In this example, we also can use update Event as trigger source instead of enable signal. Refer to *Figure 18-29. 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=010 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[3:0] = 0011 in the_ SYSCFG_TIMERxCFG register).
4. Start TIMER2 by writing '1 in the CEN bit (TIMER2_CTL0 register).

Figure 18-29. Triggering TIMER0 with update signal of TIMER2



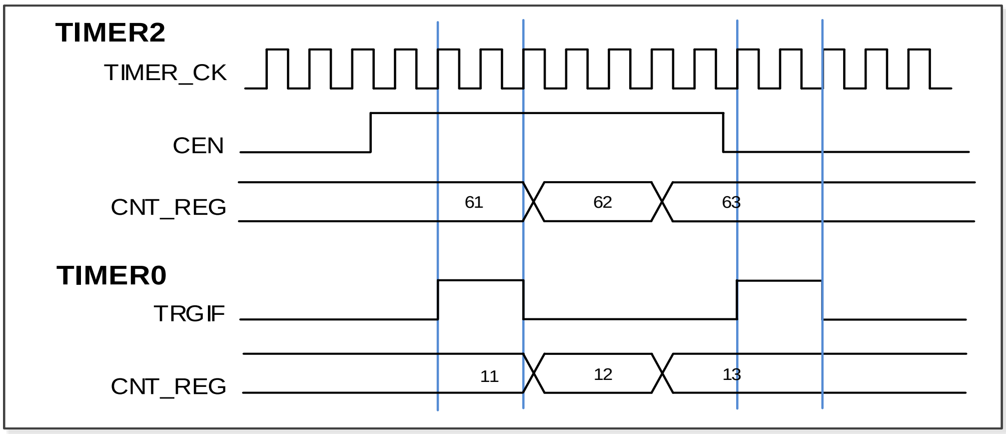
- Enable TIMER0 count with TIMER2's enable/O0CPRE signal

In this example, we control the enable of TIMER0 with the enable output of TIMER2 .Refer to

Figure 18-30. 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[3:0] = 0011 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).

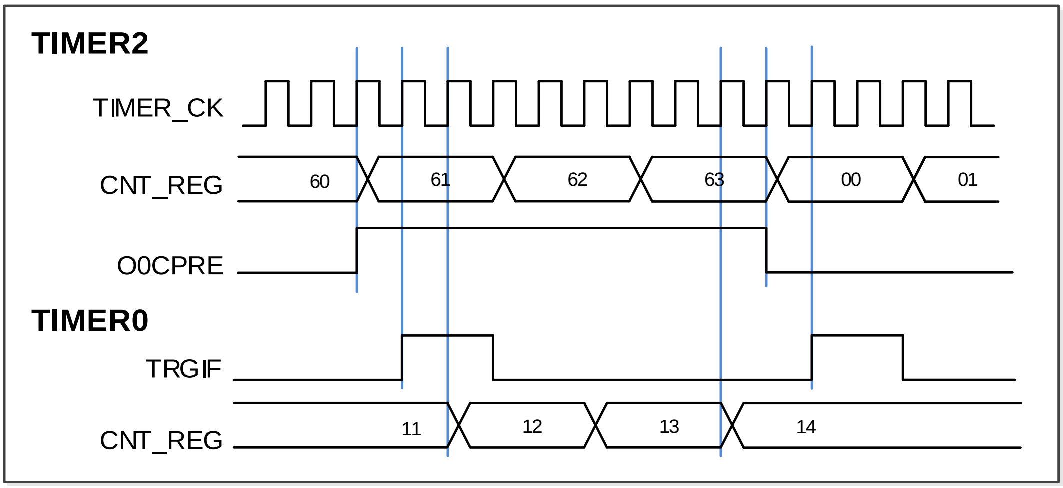
Figure 18-30. Pause TIMER0 with enable of TIMER2



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[3:0] = 0011 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 18-31. 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™-M3 is halted, and the TIMERx_HOLD configuration bit in DBG_CTL0 register is set to 1, the TIMERx counter stops.

18.1.5. 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 ratio between the timer clock (TIMER_CK) and the dead-time and sampling clock (DTS), which is used by the dead-time generators and the digital filters. 00: $f_{DTS}=f_{TIMER\_CK}$ 01: $f_{DTS}= f_{TIMER\_CK} /2$ 10: $f_{DTS}= f_{TIMER\_CK} /4$ 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															

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
31:15	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[3:0]=1000. It is possible to simultaneously use external clock mode 1 with the reset mode, pause mode or event mode. But the TSCFGy[3:0](y=3,4,5) bits must not be 1000 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 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 ETI signal and the length of the digital filter applied to ETI. 0000: Filter disalble. fSAMP= fDTS, N=1. 0001: fSAMP= fTIMER_CK, N=2.

		0010: fSAMP= fTIMER_CK, N=4. 0011: fSAMP= fTIMER_CK, 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.
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															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved			CH3OF	CH2OF	CH1OF	CH0OF	Reserved	BRKIF	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

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	Reserved	Must be kept at reset value.
7	BRKIF	Break interrupt flag This flag is set by hardware when the break input goes active, and cleared by software if the break input is not active. 0: No active level break has been detected.

		1: An active level 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								BRKG	TRGG	CMTG	CH3G	CH2G	CH1G	CH0G	UPG
								w	w	w	w	w	w	w	w

Bits	Fields	Descriptions

31:8	Reserved	Must be kept at reset value.
7	BRKG	Break 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)

<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>CH1COMCEN</td><td colspan="3">CH1COMCTL[2:0]</td><td>CH1COMSEN</td><td>CH1COMFEN</td><td colspan="2" rowspan="2">CH1MS[1:0]</td><td>CH0COMCEN</td><td colspan="3">CH0COMCTL[2:0]</td><td>CH0COMSEN</td><td>CH0COMFEN</td><td colspan="2" rowspan="2">CH0MS[1:0]</td></tr>
<tr><td colspan="4">CH1CAPFLT[3:0]</td><td colspan="2">CH1CAPPSC[1:0]</td><td colspan="4">CH0CAPFLT[3:0]</td><td colspan="2">CH0CAPPSC[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
31:16	Reserved	Must be kept at reset value
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 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 TSCFG7[3:0] bit-field in SYSCFG_TIMERxCFG (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. 000: Frozen. 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 matches the output compare register TIMERx_CH0CV. 010: Clear the channel output. O0CPRE signal is forced low when the counter matches the output compare register TIMERx_CH0CV. 011: Toggle on match. O0CPRE toggles when the counter matches the output compare register TIMERx_CH0CV. 100: Force low. O0CPRE is forced low level. 101: Force high. O0CPRE is forced high level. 110: 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. 111: 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. 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 TSCFG7[3:0] bit-field in SYSCFG_TIMERxCFG (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)

<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>CH3COMCEN</td><td colspan="3">CH3COMCTL[2:0]</td><td>CH3COMSEN</td><td>CH3COMFEN</td><td colspan="2" rowspan="2">CH3MS[1:0]</td><td>CH2COMCEN</td><td colspan="3">CH2COMCTL[2:0]</td><td>CH2COMSEN</td><td>CH2COMFEN</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
31:16	Reserved	Must be kept at reset value
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 TSCFG7[3:0] bit-field in SYSCFG_TIMERxCFG (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. 000: Frozen. The O2CPRE signal keeps stable, independent of the comparison between the output compare register TIMERx_CH2CV and the counter TIMERx_CNT. 001: Set high on match. O2CPRE signal is forced high when the counter matches the output compare register TIMERx_CH2CV. 010: Set low on match. O2CPRE signal is forced low when the counter matches the output compare register TIMERx_CH2CV. 011: Toggle on match. O2CPRE toggles when the counter matches the output compare register TIMERx_CH2CV. 100: Force low. O2CPRE is forced low level. 101: Force high. O2CPRE is forced high level. 110: 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. 111: 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 O2CPRE level changes only when the output compare mode switches from “frozen” mode to “PWM” mode or when the result of the comparison changes. 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 TSCFG7[3:0] bit-field in SYSCFG_TIMERxCFG (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															

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
31:16	Reserved	Must be kept at reset value

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 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.

Counter repetition register (TIMERx_CREP)

Address offset: 0x30

Reset value: 0x0000 0000

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

31:8	7:0
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)

31:16
Reserved

15:0
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															

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
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	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 CCS 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_CHCTL0/1 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 t_{DT}, t_{DT}=t_{DTS}. DTCFG [7:5] =3'b 10x: Dtvalue = (64+DTCFG [5:0])xt_{DT}, t_{DT} =t_{DTS}*2. DTCFG [7:5] =3'b 110: Dtvalue = (32+DTCFG [4:0])xt_{DT}, t_{DT}=t_{DTS}*8. DTCFG [7:5] =3'b 111: Dtvalue = (32+DTCFG [4:0])xt_{DT}, t_{DT} =t_{DTS}*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.

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

18.2. General level0 timer (TIMERx, x=1, 2)

18.2.1. Overview

The general level0 timer module (TIMER1, 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.

18.2.2. 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.

18.2.3. Block diagram

Figure 18-32. General Level 0 timer block diagram for GD32L233 provides details on the internal configuration of the general level0 timer.

Figure 18-32. General Level 0 timer block diagram for GD32L233

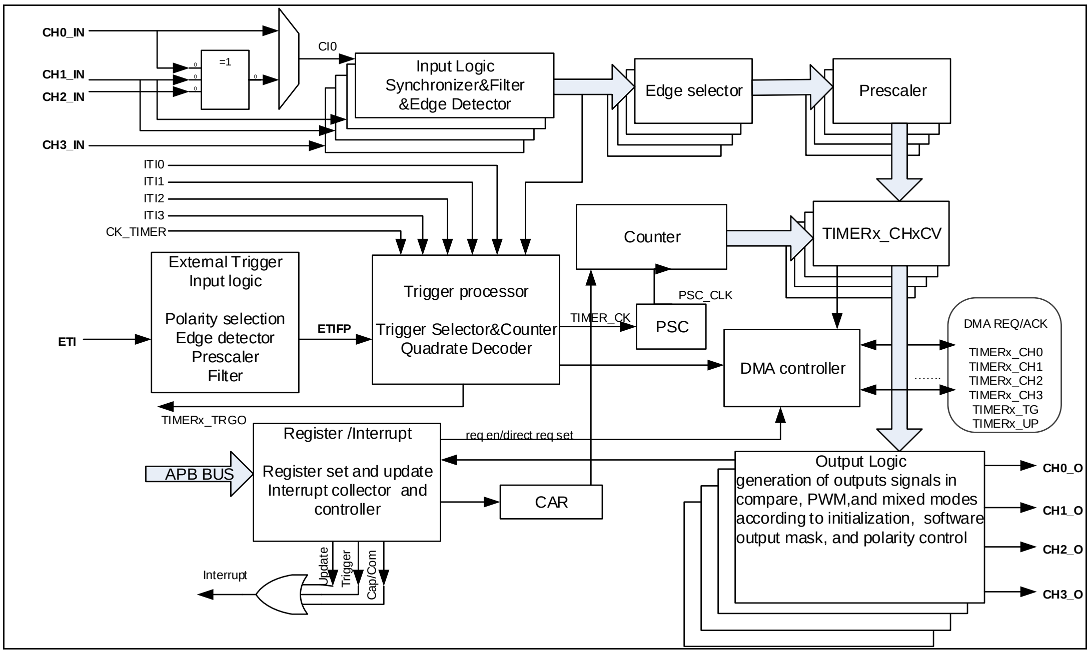
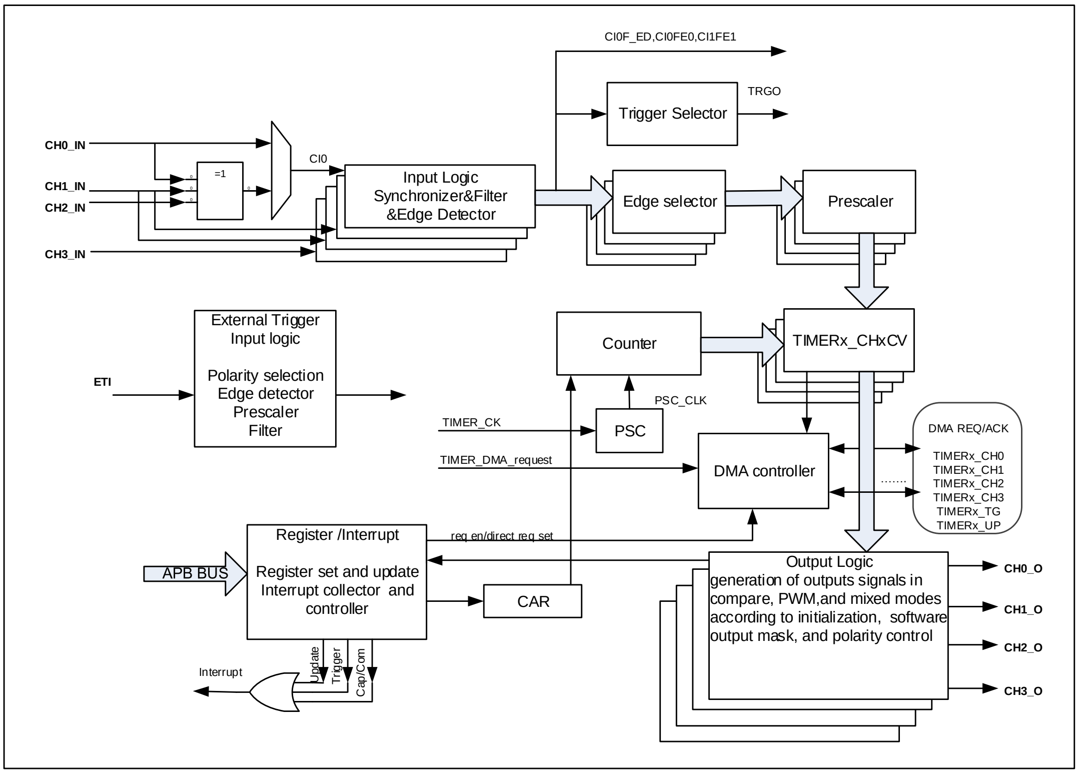


Figure 18-33. General Level 0 timer block diagram for GD32L235 provides details on the internal configuration of the general level0 timer.

Figure 18-33. General Level 0 timer block diagram for GD32L235



18.2.4. Function overview

Clock source configuration for GD32L233

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]).

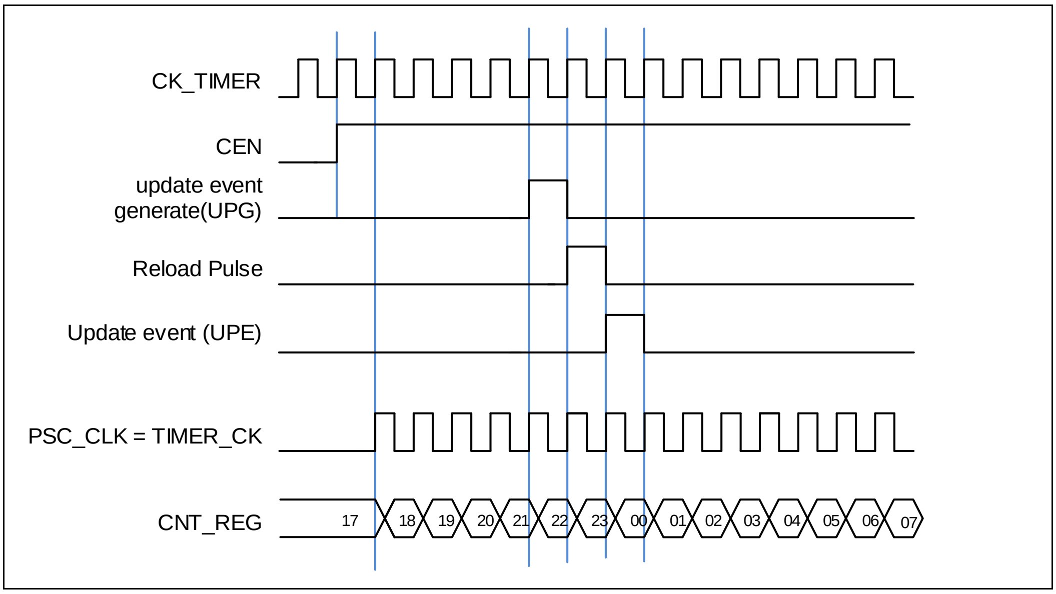
- SMC[2:0] = 3'b000. Internal clock CK_TIMER is selected as timer clock source which is from module RCU.

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 18-34. 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.

- 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 the SMC[2:0] to 0x7 and the TRGS[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 source configuration for GD32L235

The general level0 TIMER has the capability of being clocked by either the CK_TIMER or an alternate clock source controlled by TSCFGy[3:0] in SYSCFG_TIMER1CFG or SYSCFG_TIMER2CFG (y=0,1…7)

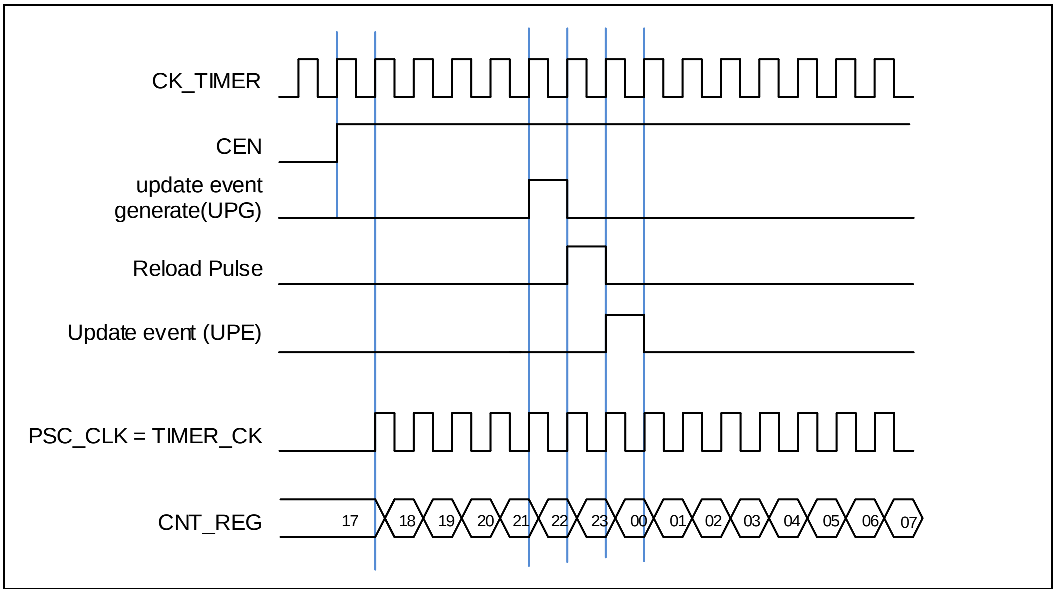
- TSCFGy[3:0] = 4’b0000 in SYSCFG_TIMER1CFG or SYSCFG_TIMER2CFG (y=0,1…7). Internal clock CK_TIMER is selected as timer clock source which is from module RCU.

The default clock source is the CK_TIMER for driving the counter prescaler when the slave mode is disabled (TSCFGy[3:0] = 4’b0000 in SYSCFG_TIMER1CFG or SYSCFG_TIMER2CFG (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 the slave mode controller is enabled by setting TSCFGy[3:0] != 4’b0000 in SYSCFG_TIMER1CFG or SYSCFG_TIMER2CFG(y=0,1,2,6), the prescaler is clocked by other clock sources selected in the TSCFGy[3:0] (y=6) register, more details will be introduced later. When the TSCFGy[3:0] != 4’b0000 (y=3,4,5), the internal clock CK_TIMER is the counter prescaler driving clock source.

Figure 18-35. Timing chart of internal clock divided by 1



- TSCFG6[3:0] != 4'b0000 (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[3:0] to 0x5, 0x6, 0x7.

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 TSCFG6[3:0] to 0x1,0x2,0x3,0x4.

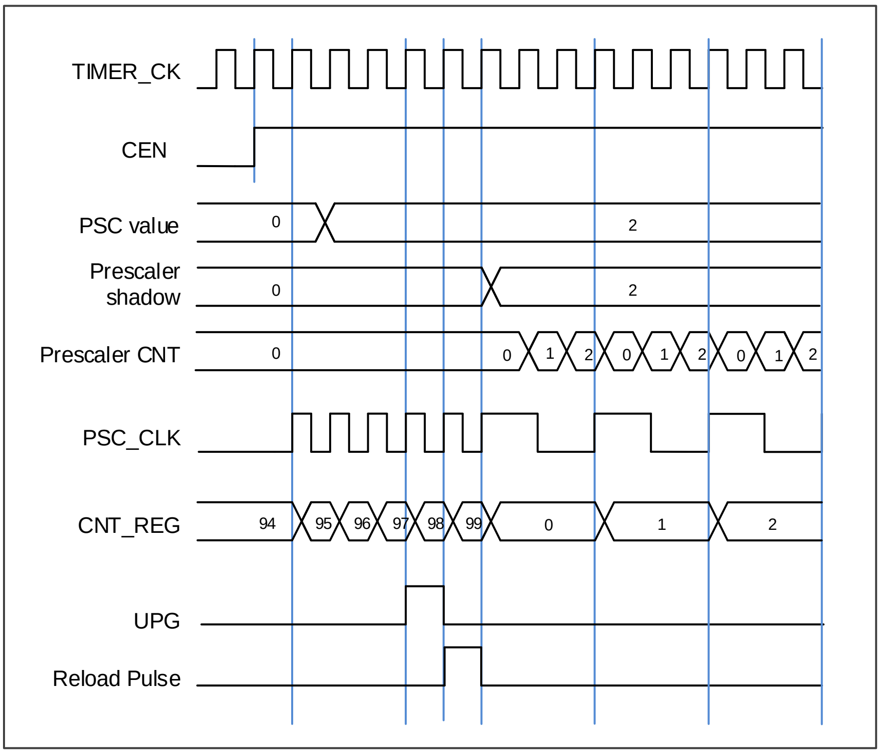
- SMC1= 1'b1 (external clock mode 1). External input ETI is selected as timer clock source.

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[3:0] to 0x8. 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 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 18-36. 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 18-37. Timing chart of up counting mode, PSC=0/2 and ***Figure 18-38. 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 18-37. Timing chart of up counting mode, PSC=0/2

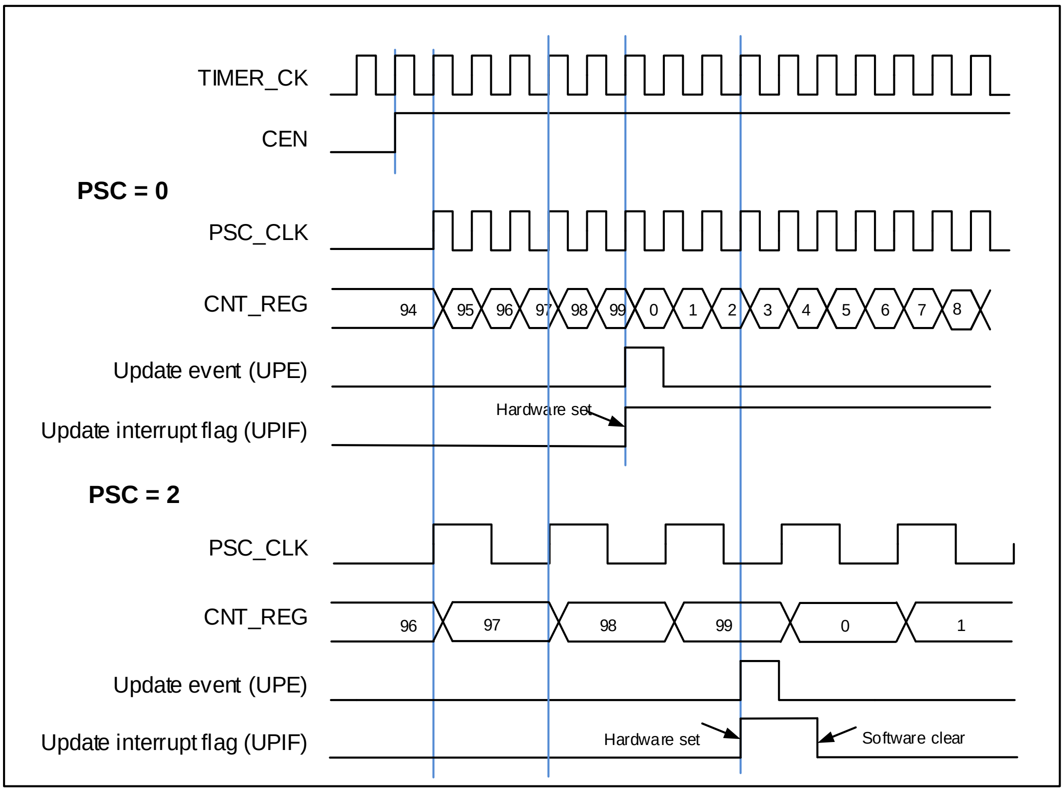
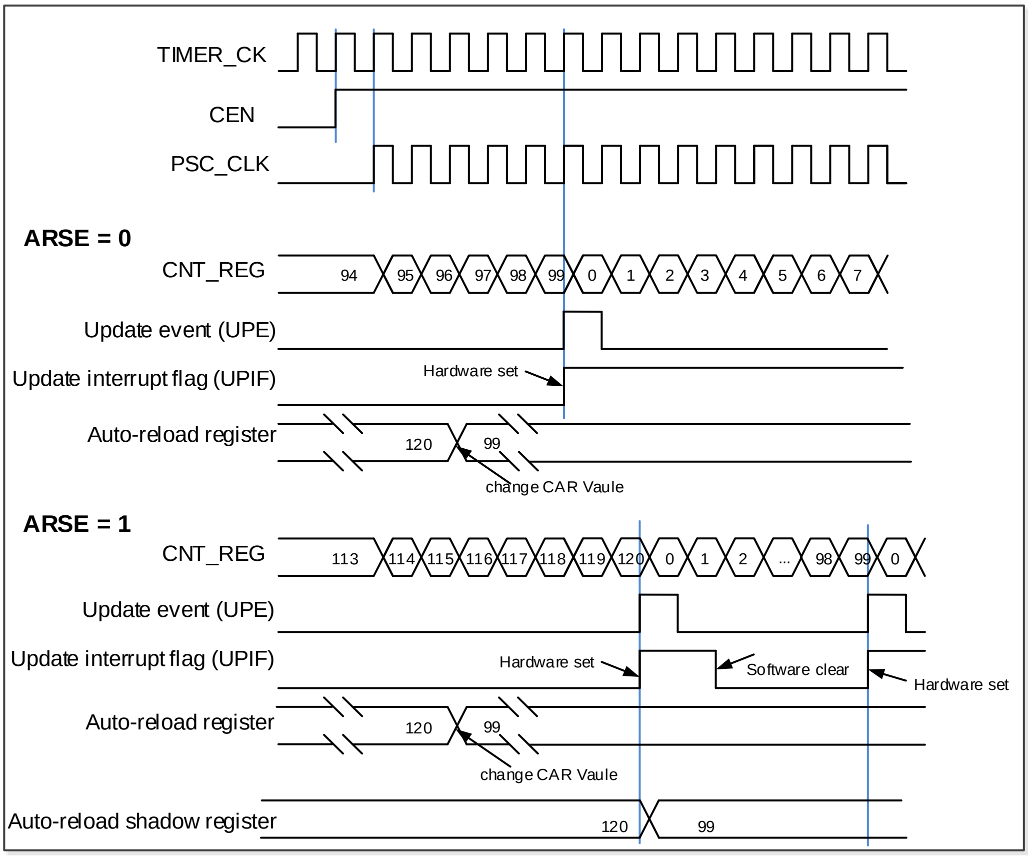


Figure 18-38. Timing chart of up counting, change TIMERx_CAR ongoing



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 18-39. Timing chart of down counting mode, PSC=0/2 and *Figure 18-40. 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 18-39. Timing chart of down counting mode, PSC=0/2

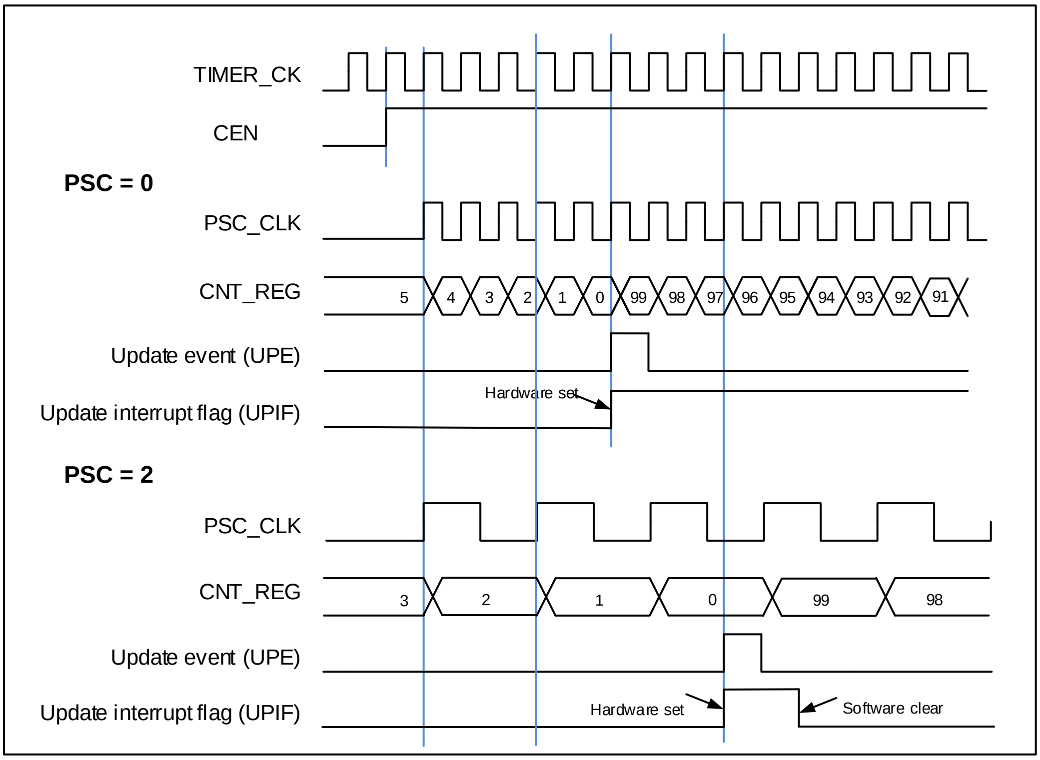
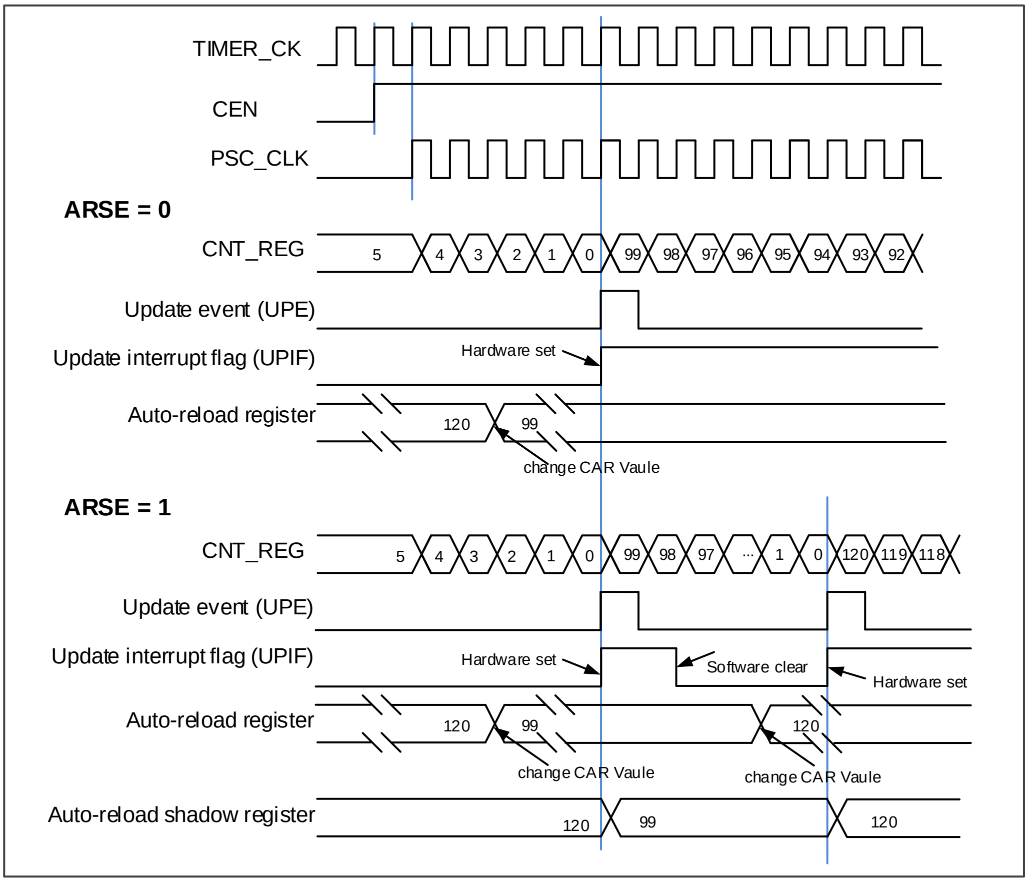


Figure 18-40. Timing chart of down counting mode, change TIMERx_CAR ongoing



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.

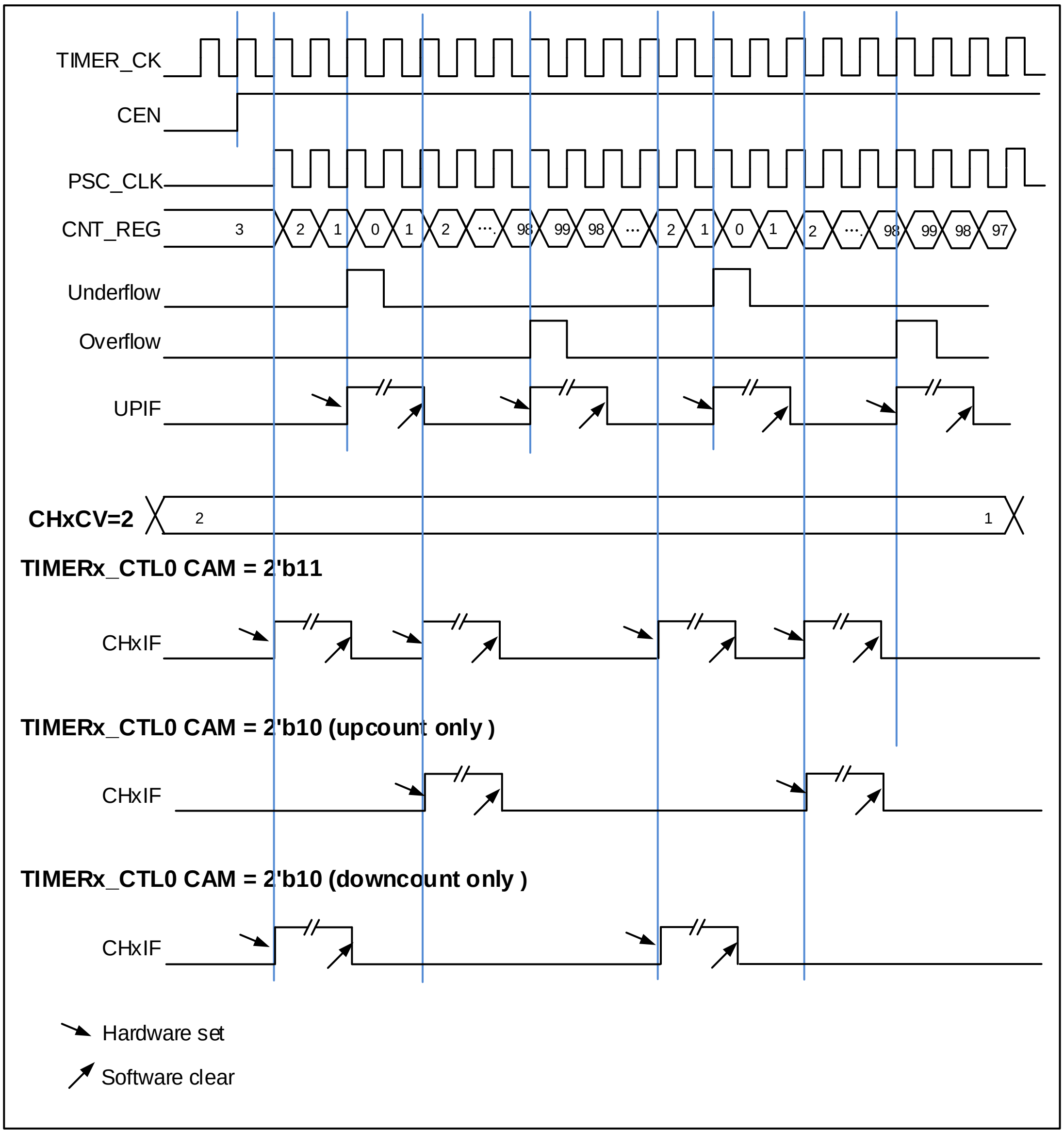
The UPIF bit in the TIMERx_INTF register will be set to 1 either when an underflow event or an overflow event occurs. While the ChxIF bit is associated with the value of CAM in TIMERx_CTL0. The details refer to ***Figure 18-41. Timing chart of center-aligned counting mode***.

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 18-41. Timing chart of center-aligned counting mode shows the example of the counter behavior when TIMERx_CAR=0x99, TIMERx_PSC=0x0

Figure 18-41. Timing chart of center-aligned counting mode



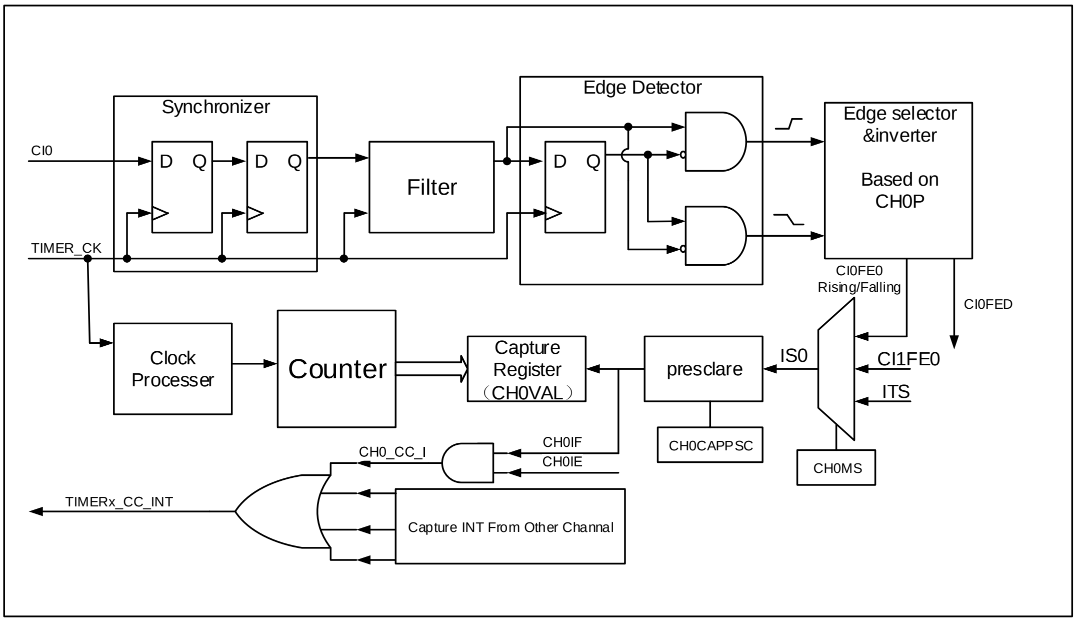
Input capture and output compare channels

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 18-42. 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 18-43. Channel output compare principle (x=0,1,2,3)

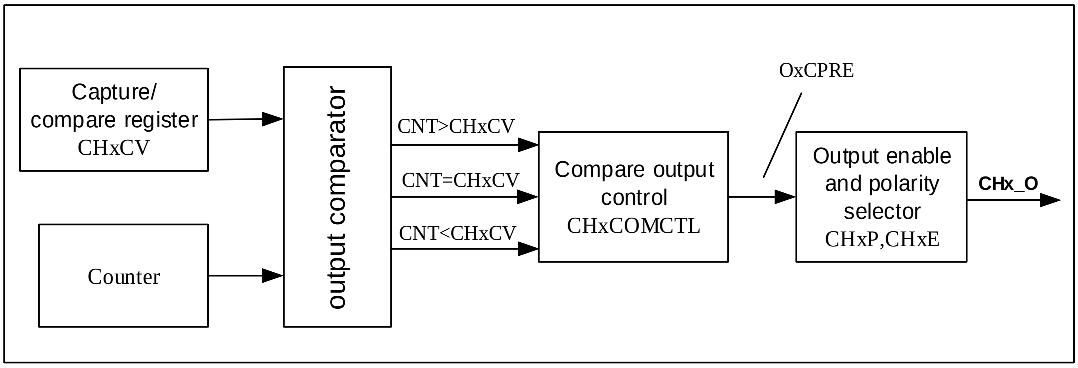


Figure 18-43. Channel output compare principle (x=0,1,2,3) shows the principle circuit of channels output compare function. The relationship between the channel output signal CHx_O and the OxCPRE signal (more details refer to *Channel output prepare signal*) is described as blew: The active level of 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).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.

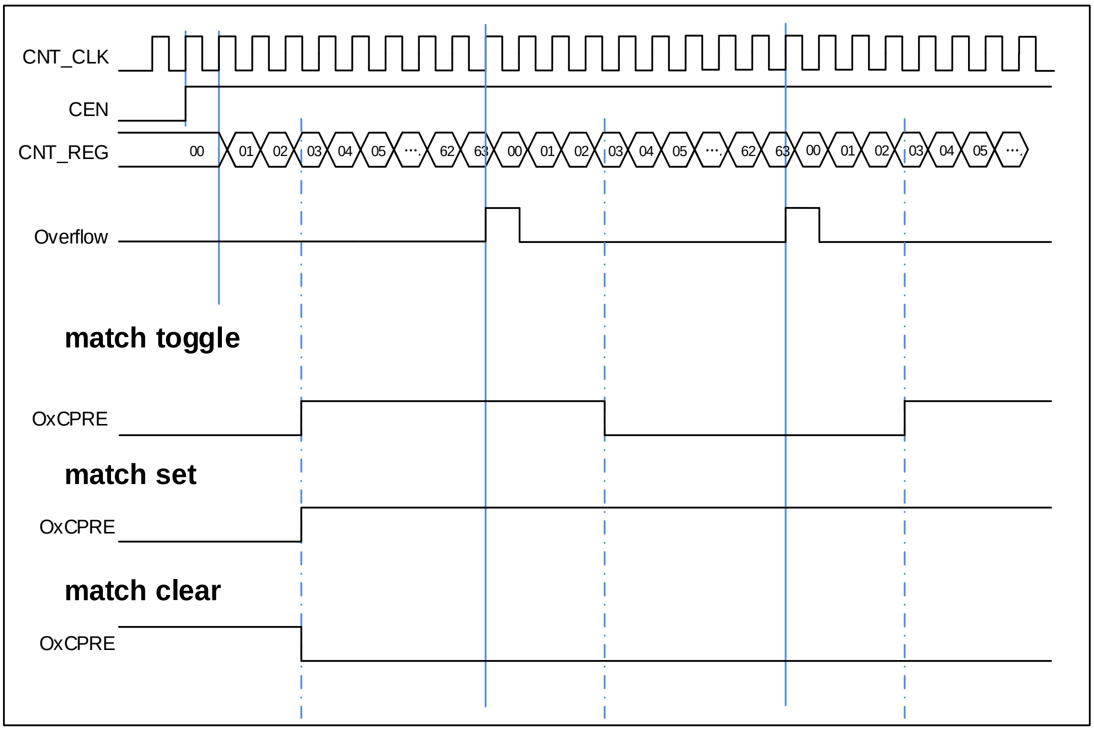
Step3: Interrupt/DMA-request enables configuration by ChxIE/CxCDE.

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 18-44. 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 18-45. 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 18-46. 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 18-45. Timing chart of EAPWM

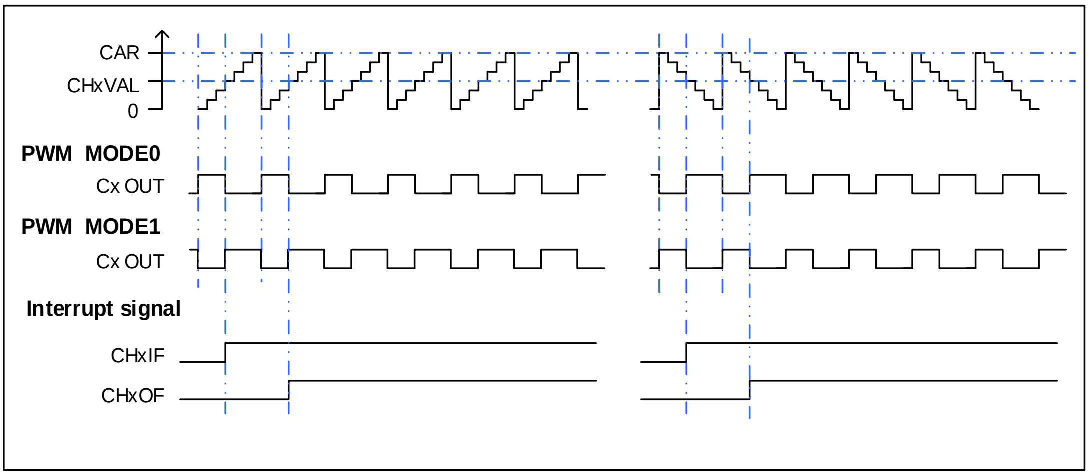
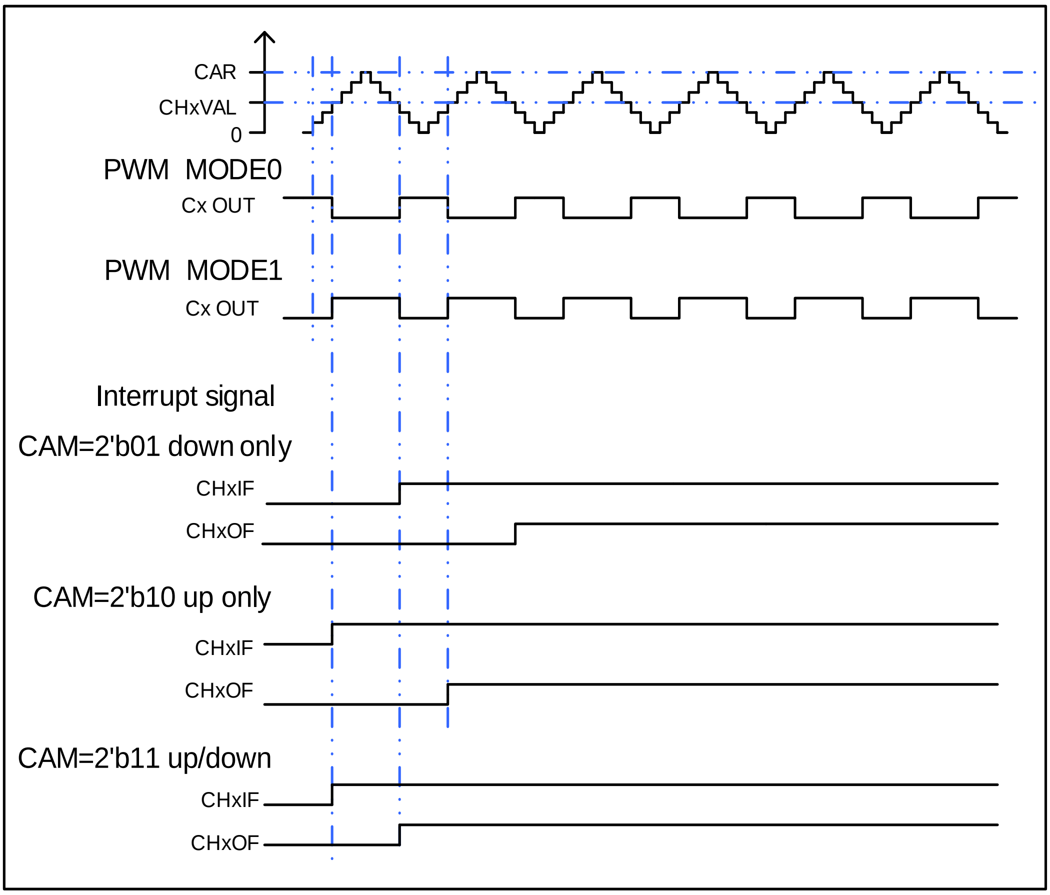


Figure 18-46. Timing chart of CAPWM



Channel output prepare signal

As is shown in *Figure 18-43. Channel output compare principle (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 CI0FE0 and CI1FE1 derived from the TIMERx_CH0 and TIMERx_CH1 pins respectively to interact to control the counter value. The DIR bit is modified during each input source transition. The counter can be changed by the edges of CI0FE0 only, CI1FE1 only or both CI0FE0 and CI1FE1.

- For GD32L233, the mode is selected by setting the SMC[2:0] to 0x01, 0x02 or 0x03.
- For GD32L235, the mode is selected by setting TSCFGy[3:0] != 4'b0000 (y=0,1,2).

The mechanism for changing the counter direction is shown in ***Table 18-5. Counting direction in different quadrature decoder mode***. The quadrature decoder can be regarded as an external clock with a directional selection. This means that the counter counts continuously in the interval between 0 and the counter-period value. Therefore, TIMERx_CAR register must be configured before the counter starts to count.

Table 18-5. Counting direction in different quadrature decoder mode

Counting mode	Level	CI0FE0		CI1FE1	
		Rising	Falling	Rising	Falling
Quadrature decoder mode 0 SMC[2:0]=3'b001	CI1FE1=1	Down	Up	-	-
	CI1FE1=0	Up	Down	-	-
Quadrature decoder mode 1 SMC [2:0]=3'b010	CI0FE0=1	-	-	Up	Down
	CI0FE0=0	-	-	Down	Up
Quadrature decoder mode 2 SMC [2:0]=3'b011	CI1FE1=1	Down	Up	X	X
	CI1FE1=0	Up	Down	X	X
	CI0FE0=1	X	X	Up	Down
	CI0FE0=0	X	X	Down	Up

Note: "-" means "no counting"; "X" means impossible. "0" means "low level", "1" means "high level".

Figure 18-47. Counter behavior with CI0FE0 polarity non-inverted in mode 2

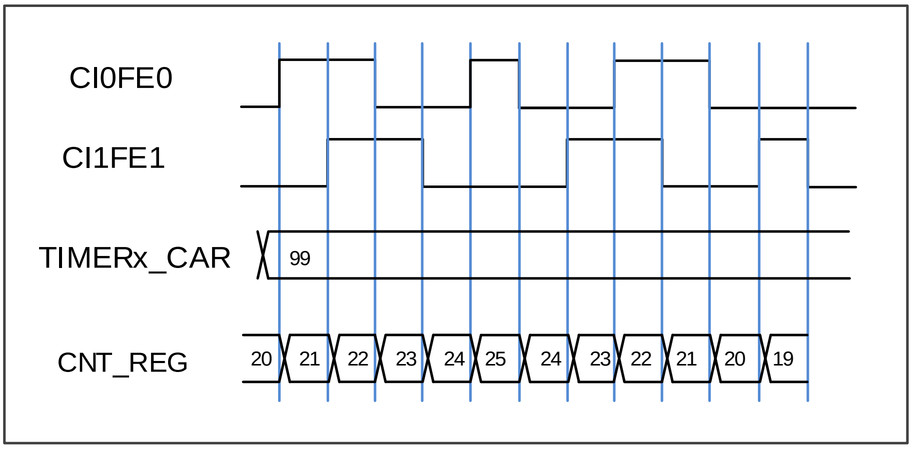
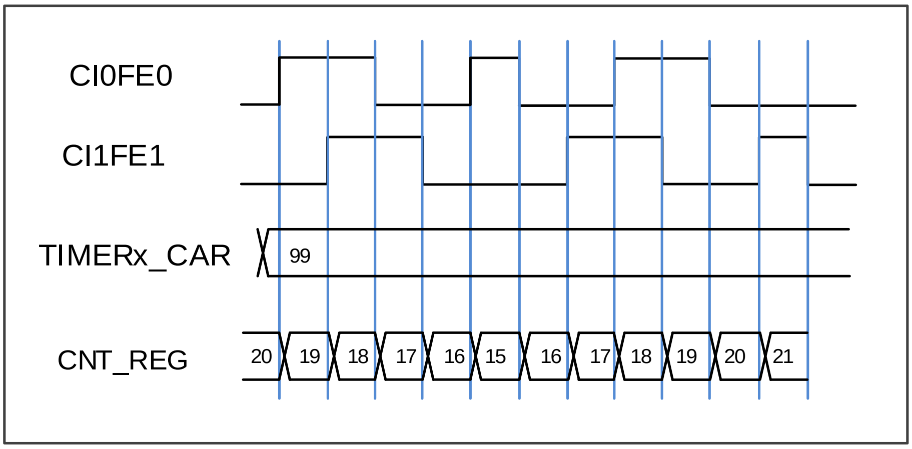


Figure 18-48. Counter behavior with CI0FE0 polarity inverted in mode 2



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.

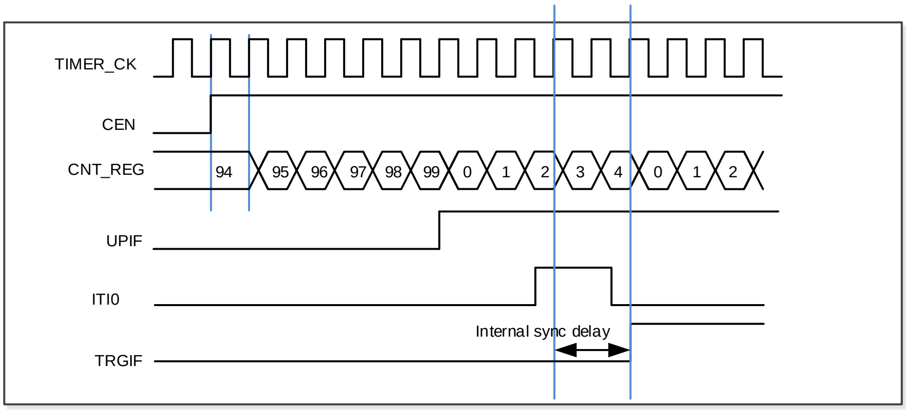
Master-slave management

The TIMERx can be synchronized with a trigger in several modes including the restart mode, the pause mode and the event mode.

- For GD32L233, these modes are 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.
- For GD32L235, these modes are selected by the TSCFGy[3:0] != 4b'0000 in

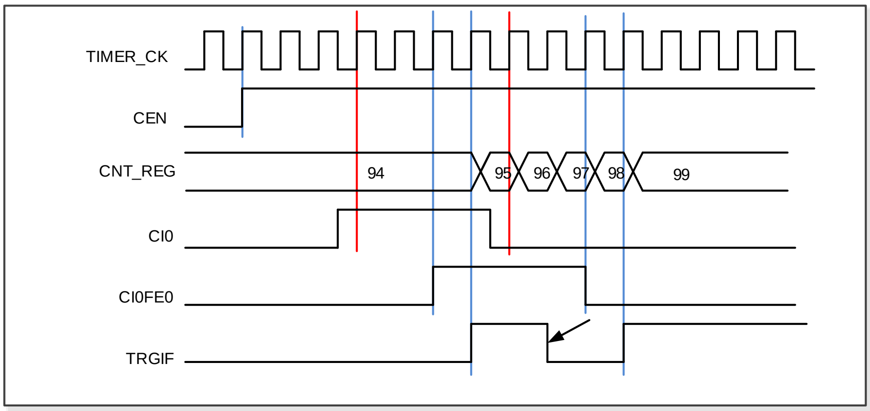
SYSCFG_TIMER1CFG or SYSCFG_TIMER2CFG (y=3,4,5), The trigger input of these modes can be selected by the TSCFGy[3:0] in SYSCFG_TIMER1CFG or SYSCFG_TIMER2CFG, different TSCFGy[3:0] is correspond to different trigger input.

Table 18-6. Slave mode example table for GD32L233

<table>
<tr><th></th><th>Mode Selection</th><th>Source Selection</th><th>Polarity Selection</th><th>Filter and Prescaler</th></tr>
<tr><td>LIST</td><td>SMC[2:0]
3’b100 (restart mode)
3’b101 (pause mode)
3’b110 (event mode)</td><td>TRGS[2:0]
000: ITI0
001: ITI1
010: ITI2
011: ITI3
100: CI0F_ED
101: CI0FE0
110: CI1FE1
111: ETIFP</td><td>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.</td><td>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.</td></tr>
<tr><td rowspan="2">Exam1</td><td>Restart mode
The counter can be clear and restart when a rising trigger input.</td><td>TRGS[2:0]=3’b000
ITI0 is the selection.</td><td>For ITI0, no polarity selector can be used.</td><td>For the ITI0, no filter and prescaler can be used.</td></tr>
<tr><td colspan="4">Figure 18-49. Restart mode

</td></tr>
<tr><td>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>
</table>

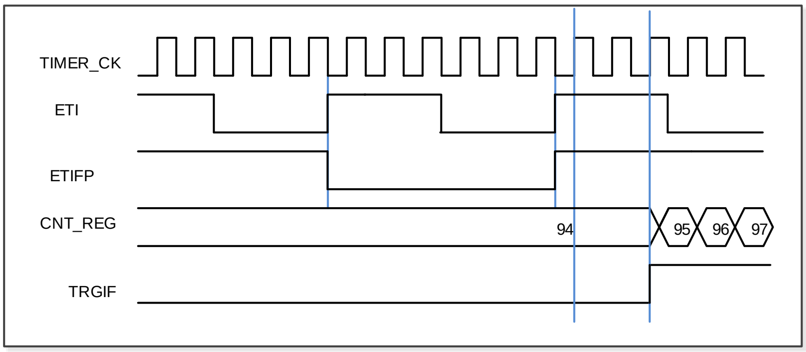
<table>
<tr><th></th><th>Mode Selection</th><th>Source Selection</th><th>Polarity Selection</th><th>Filter and Prescaler</th></tr>
<tr><td></td><td colspan="4">

Figure 18-50. Pause mode



</td></tr>
<tr><td rowspan="2">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>
<tr><td colspan="4">

Figure 18-51. Event mode



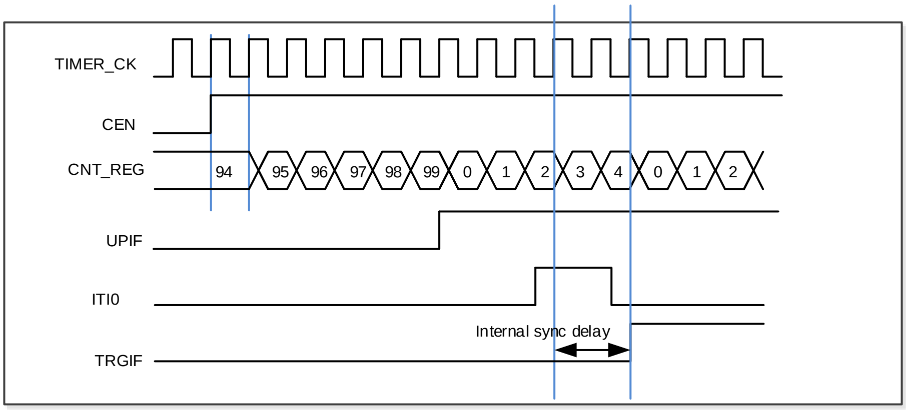
</td></tr>
</table>

Table 18-7. Slave mode example table for GD32L235

	Mode Selection	Source Selection	Polarity Selection	Filter and Prescaler
LIST	TSCFGy[3:0] y=3 (restart mode) y=4 (pause mode) y=5 (event mode)	TSCFGy[3:0] 0001: ITI0 0010: ITI1 0011: ITI2 0100: ITI3 0101: CI0F_ED 0110: CI0FE0 0111: CI1FE1 1000: 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	TSCFG3[3:0] =	For ITI0, no polarity	For the ITI0, no filter and

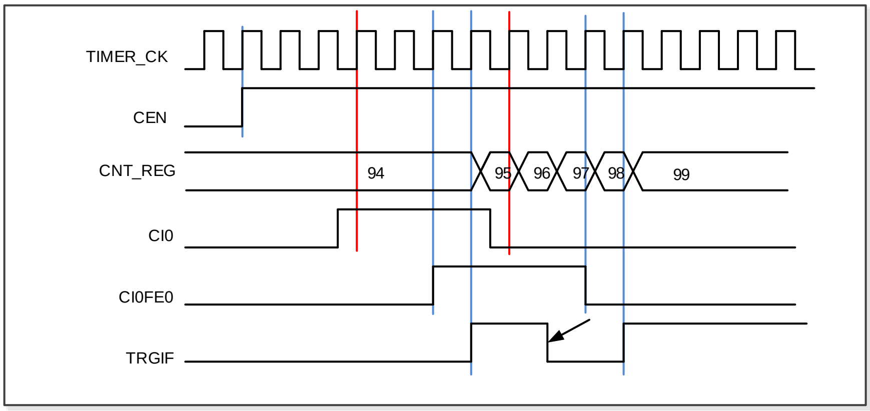
<table>
<tr><th></th><th>Mode Selection</th><th>Source Selection</th><th>Polarity Selection</th><th>Filter and Prescaler</th></tr>
<tr><td rowspan="2"></td><td>The counter can be clear and restart when a rising trigger input.</td><td>4’b0001
ITI0 is the selection.</td><td>selector can be used.</td><td>prescaler can be used.</td></tr>
<tr><td colspan="4">

Figure 18-52. 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>TSCFG4[3:0] = 4’b0110
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">

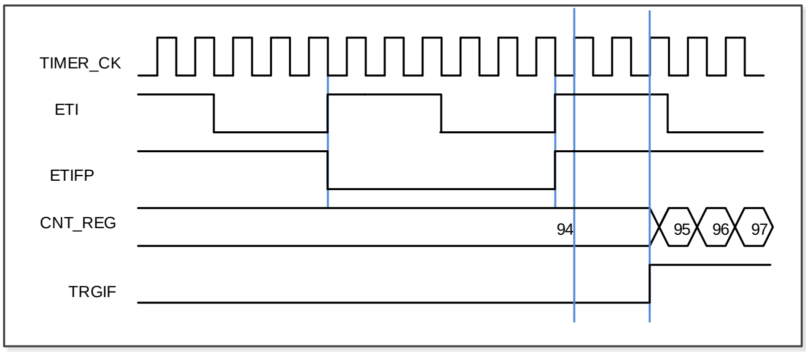
Figure 18-53. Pause mode



</td></tr>
<tr><td>Exam3</td><td>Event mode
The counter will start to count when a rising trigger input.</td><td>TSCFG5[3:0] = 4’b1000
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>

<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 18-54. Event mode



</td></tr>
</table>

Single pulse mode

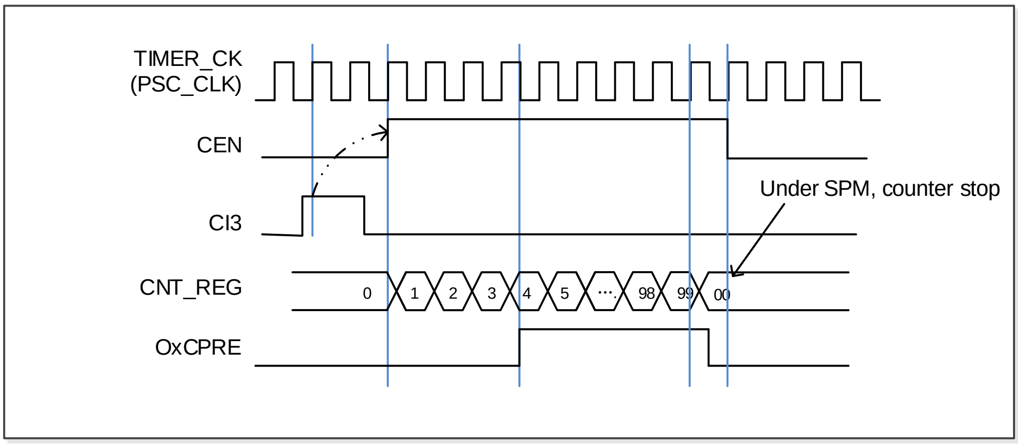
Single pulse mode is opposite to the repetitive mode, which can be enabled by setting SPM in TIMERx_CTL0. When you set SPM, the counter will be clear and stop when the next update event. In order to get pulse waveform, you can set the TIMERx to PWM mode or compare by CHxCOMCTL.

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 18-55. Single pulse mode TIMERx_CHxCV = 4 TIMERx_CAR=99 shows an example.

Figure 18-55. Single pulse mode TIMERx_CHxCV = 4 TIMERx_CAR=99



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 TIMER1

We configure TIMER2 as a prescaler for TIMER1. Do as follow:

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. Select the TIMER1 input trigger source from TIMER2 and configure TIMER1 in external clock mode 1.
   - For GD32L233, setting TRGS=000 in the TIMER1_SMCFG register and setting SMC=111 in TIMER1_SMCFG register.
   - For GD32L235, setting TSCFG6[3:0] = 0001 in the_SYSCFG_TIMERxCFG register.
4. Start TIMER1 by writing ‘1 in the CEN bit (TIMER1_CTL0 register).
5. Start TIMER2 by writing ‘1 in the CEN bit (TIMER2_CTL0 register).

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.

18.2.5. TIMERx registers(x=1, 2)

TIMER1 base address: 0x4000 0000

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)

For GD32L233xx devices

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	TRGS[2:0]			Reserved	SMC[2:0]		
rw	rw	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:

<table>
<tr><th>EXTFC[3:0]</th><th>Times</th><th>f_{SAMP}</th></tr>
<tr><td>4’b0000</td><td colspan="2">Filter disabled.</td></tr>
<tr><td>4’b0001</td><td>2</td><td rowspan="3">f_{TIMER_CK}</td></tr>
<tr><td>4’b0010</td><td>4</td></tr>
<tr><td>4’b0011</td><td>8</td></tr>
<tr><td>4’b0100</td><td>6</td><td rowspan="2">$f_{DTS\_CK}/2$</td></tr>
<tr><td>4’b0101</td><td>8</td></tr>
<tr><td>4’b0110</td><td>6</td><td rowspan="2">$f_{DTS\_CK}/4$</td></tr>
<tr><td>4’b0111</td><td>8</td></tr>
<tr><td>4’b1000</td><td>6</td><td rowspan="2">$f_{DTS\_CK}/8$</td></tr>
<tr><td>4’b1001</td><td>8</td></tr>
<tr><td>4’b1010</td><td>5</td><td rowspan="3">$f_{DTS\_CK}/16$</td></tr>
<tr><td>4’b1011</td><td>6</td></tr>
<tr><td>4’b1100</td><td>8</td></tr>
<tr><td>4’b1101</td><td>5</td><td rowspan="3">$f_{DTS\_CK}/32$</td></tr>
<tr><td>4’b1110</td><td>6</td></tr>
<tr><td>4’b1111</td><td>8</td></tr>
</table>

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
6:4	TRGS[2:0]	Trigger selection. This bit-field specifies which signal is selected as the trigger input, which is used to synchronize the counter. 000: ITI0 001: ITI1 010: ITI2 011: ITI3 100: CI0F_ED 101: CI0FE0 110: CI1FE1 111: ETIFP These bits must not be changed when slave mode is enabled.
3	Reserved	Must be kept at reset value.

2:0	SMC[2:0]	Slave mode control. 000: Disable mode. The slave mode is disabled; The prescaler is clocked directly by the internal clock (TIMER_CK) when CEN bit is set high. 001: Quadrature decoder mode 0. The counter counts on CI0FE0 edge, while the direction depends on CI1FE1 level. 010: Quadrature decoder mode 1. The counter counts on CI1FE1 edge, while the direction depends on CI0FE0 level. 011: Quadrature decoder mode 2. The counter counts on both CI0FE0 and CI1FE1 edge, while the direction depends on each other. 100: Restart Mode. The counter is reinitialized and an update event is generated on the rising edge of the selected trigger input. 101: Pause Mode. The trigger input enables the counter clock when it is high and disables the counter clock when it is low. 110: Event Mode. A rising edge of the trigger input enables the counter. 111: External Clock Mode 0. The counter counts on the rising edges of the selected trigger. Because CI0F_ED outputs 1 pulse for each transition on CI0F, and the pause mode checks the level of the trigger signal, when CI0F_ED is selected as the trigger input, the pause mode must not be used.

For GD32L235xx devices

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 TSCFGy[3:0](y=3,4,5) must not be 4'b1000 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 SYSCFG_TIMERxCFG register.
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:

<table>
<tr><th>EXTFC[3:0]</th><th>Times</th><th>f_{SAMP}</th></tr>
<tr><td>4'b0000</td><td colspan="2">Filter disabled.</td></tr>
<tr><td>4'b0001</td><td>2</td><td rowspan="3">f_{TIMER_CK}</td></tr>
<tr><td>4'b0010</td><td>4</td></tr>
<tr><td>4'b0011</td><td>8</td></tr>
<tr><td>4'b0100</td><td>6</td><td rowspan="2">$f_{DTS\_CK}/2$</td></tr>
<tr><td>4'b0101</td><td>8</td></tr>
<tr><td>4'b0110</td><td>6</td><td rowspan="2">$f_{DTS\_CK}/4$</td></tr>
<tr><td>4'b0111</td><td>8</td></tr>
<tr><td>4'b1000</td><td>6</td><td rowspan="2">$f_{DTS\_CK}/8$</td></tr>
<tr><td>4'b1001</td><td>8</td></tr>
<tr><td>4'b1010</td><td>5</td><td rowspan="3">$f_{DTS\_CK}/16$</td></tr>
<tr><td>4'b1011</td><td>6</td></tr>
<tr><td>4'b1100</td><td>8</td></tr>
<tr><td>4'b1101</td><td>5</td><td rowspan="3">$f_{DTS\_CK}/32$</td></tr>
<tr><td>4'b1110</td><td>6</td></tr>
<tr><td>4'b1111</td><td>8</td></tr>
</table>

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															

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
31:16	Reserved	Must be kept at reset value
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. For GD32L233 through TRGS bits in TIMERx_SMCFG register; For GD32L235 through TSCFG7[3:0] bit-field in SYSCFG_TIMERxCFG (x=1,2) 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 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: 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. For GD32L233 through TRGS bits in TIMERx_SMCFG register; For GD32L235 through TSCFG7[3:0] bit-field in SYSCFG_TIMERxCFG (x=1,2) 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="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>CH3COMCEN</td><td colspan="3">CH3COMCTL[2:0]</td><td>CH3COMSEN</td><td>CH3COMFEN</td><td colspan="2" rowspan="2">CH3MS[1:0]</td><td>CH2COMCEN</td><td colspan="3">CH2COMCTL[2:0]</td><td>CH2COMSEN</td><td>CH2COMFEN</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
31:16	Reserved	Must be kept at reset value
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. For GD32L233 through TRGS bits in TIMERx_SMCFG register; For GD32L235 through TSCFG7[3:0] bit-field in SYSCFG_TIMERxCFG (x=1,2) 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 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 O2CPRE signal keeps stable, independent of the comparison between the output compare register TIMERx_CH2CV and the counter TIMERx_CNT. 001: Set the channel output. O2CPRE signal is forced high when the counter is equals to the output compare register TIMERx_CH2CV. 010: Clear the channel output. O2CPRE signal is forced low when the counter is equals to the output compare register TIMERx_CH2CV. 011: Toggle on match. O2CPRE toggles when the counter is equals to the output compare register TIMERx_CH2CV. 100: Force low. O2CPRE is forced to low level. 101: Force high. O2CPRE is forced to high level. 110: PWM mode 0. When counting up, O2CPRE is high when the counter is smaller than TIMERx_CH2CV, and low otherwise. When counting down, O2CPRE is low when the counter is larger than TIMERx_CH2CV, and high otherwise. 111: PWM mode 1. When counting up, O2CPRE is low when the counter is smaller than TIMERx_CH2CV, and high otherwise. When counting down, O2CPRE is high when the counter is larger than TIMERx_CH2CV, and low otherwise. If configured in PWM mode, the O2CPRE level changes only when the output compare mode is adjusted from “Timing” mode to “PWM” mode or the comparison result changes.
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

Bits	Fields	Descriptions
		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. For GD32L233 through TRGS bits in TIMERx_SMCFG register; For GD32L235 through TSCFG7[3:0] bit-field in SYSCFG_TIMERxCFG (x=1,2) 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	

Bits	Fields	Descriptions
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)

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.

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

18.3. General level1 timer (TIMERx, x=8, 11)

18.3.1. Overview

The general level1 timer module (Timer8, 11) is a two-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 level1 time reference is a 16-bit counter that can be used as an unsigned counter.

In addition, the general level1 timers can be programmed and be used to count or time external events that drive other Timers.

Timer and timer are completely independent, but there may be synchronized to provide a larger timer with their counters incrementing in unison.

18.3.2. Characteristics

- Total channel num: 2.
- Counter width: 16bit.
- Source of count clock is selectable:
  internal clock, internal trigger, external input, external trigger.
- 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 PWM mode, Single pulse mode
- Auto-reload function.
- Interrupt output on: update, trigger event, and compare/capture event.
- Daisy chaining of timer modules to allow a single timer to initiate multiple timing events.
- Timer synchronization allows selected timers to start counting on the same clock cycle.
- Timer master-slave management.

18.3.3. Block diagram

Figure 18-56. General Level 1 timer block diagram for GD32L233 provides details on the internal configuration of the general level1 timer

Figure 18-56. General Level 1 timer block diagram for GD32L233

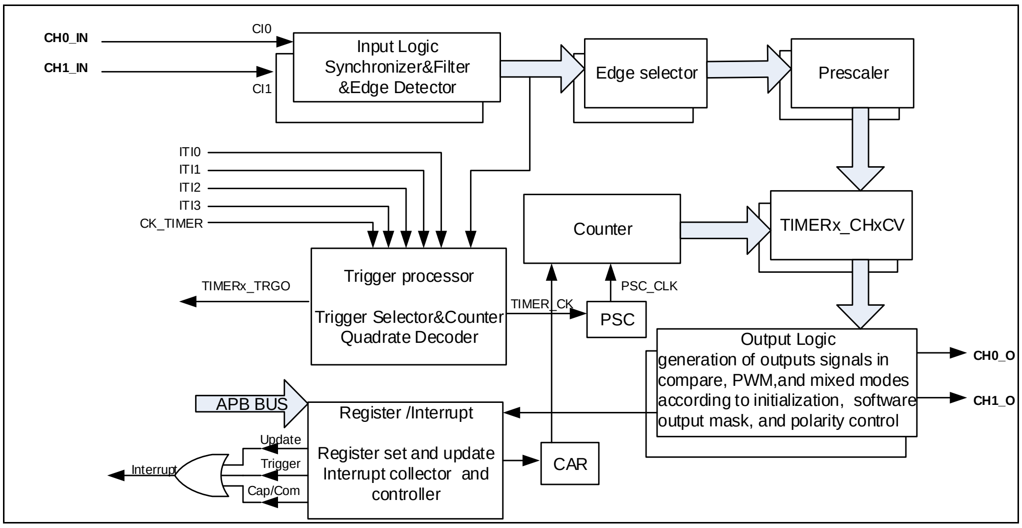
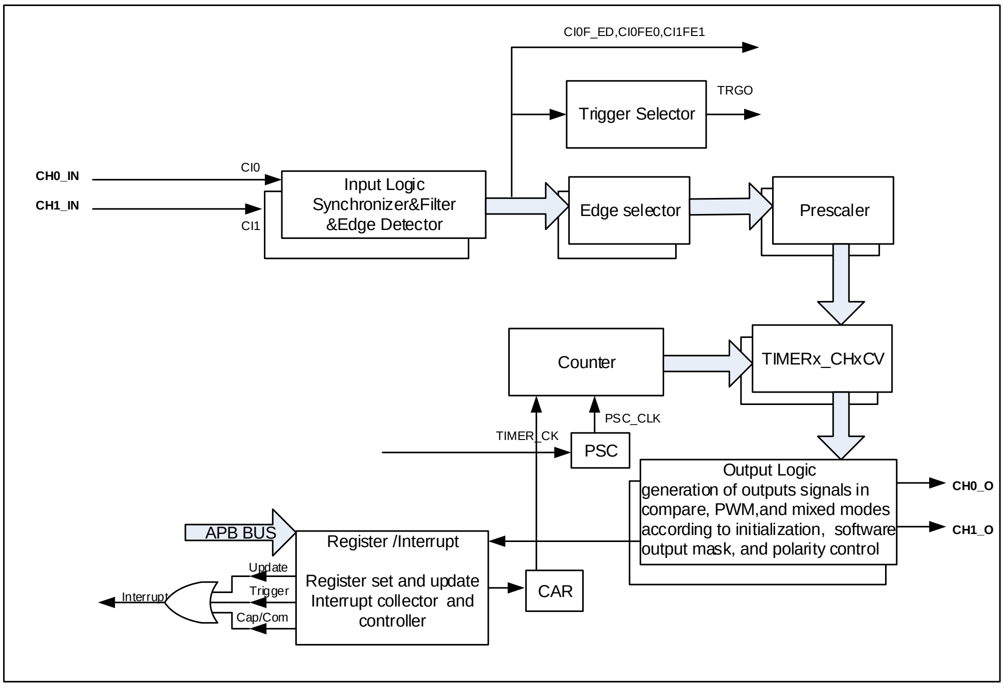


Figure 18-57. General Level 1 timer block diagram for GD32L235 provides details on the internal configuration of the general level1 timer.

Figure 18-57. General Level 1 timer block diagram for GD32L235



18.3.4. Function overview

Clock source configuration for GD32L233

The general level1 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]).

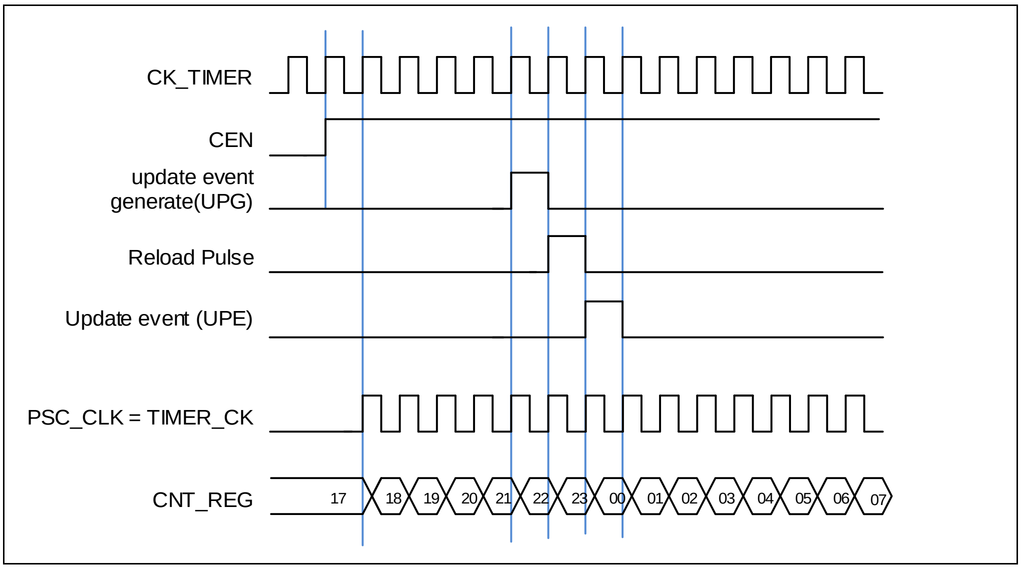
- SMC [2:0] == 3'b000. Internal timer clock CK_TIMER which is from module RCU.

The default internal clock source is the CK_TIMER used to drive 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, driven counter's prescaler to count, is equal to CK_TIMER which is from RCU.

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 and described as follows. When the SMC bits are set to 0x4, 0x5 or 0x6, the internal clock CK_TIMER is the counter prescaler driving clock source.

Figure 18-58. Timing chart of internal clock divided by 1



- SMC [2:0] == 3'b111 (external clock mode 0). External input pin source

The TIMER_CK, driven counter's prescaler to count, can be triggered by the event of rising or falling edge on the external pin TIMERx_CI0/TIMERx_CI1. 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 source configuration for GD32L235

The general level1 TIMER has the capability of being clocked by either the CK_TIMER or an alternate clock source controlled by TSCFGy[3:0] in SYSCFG_TIMER8CFG or SYSCFG_TIMER11CFG (y=0,1…7)

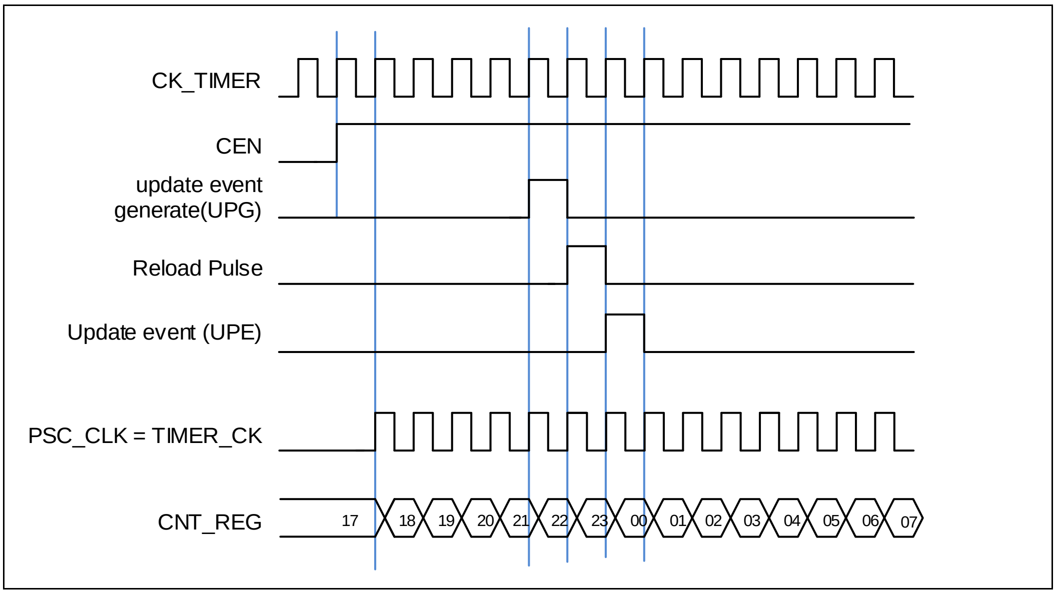
- TSCFGy[3:0] = 4'b0000 in SYSCFG_TIMER8CFG or SYSCFG_TIMER11CFG (y=0,1…7). Internal clock CK_TIMER is selected as timer clock source which is from module RCU.

The default clock source is the CK_TIMER for driving the counter prescaler when the slave mode is disabled (TSCFGy[3:0] = 4'b0000 in SYSCFG_TIMER8CFG or SYSCFG_TIMER11CFG (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 the slave mode controller is enabled by setting TSCFGy[3:0] != 4'b0000 in SYSCFG_TIMER8CFG or SYSCFG_TIMER11CFG(y=0,1,2,6), the prescaler is clocked by other clock sources selected in the TSCFGy[3:0] (y=6) register, more details will be introduced later. When the TSCFGy[3:0] != 4'b0000 (y=3,4,5), the internal clock CK_TIMER is the counter prescaler driving clock source.

Figure 18-59. Timing chart of internal clock divided by 1



- TSCFG6[3:0] != 4'b0000 (external clock mode 0). External input pin is selected as timer clock source.

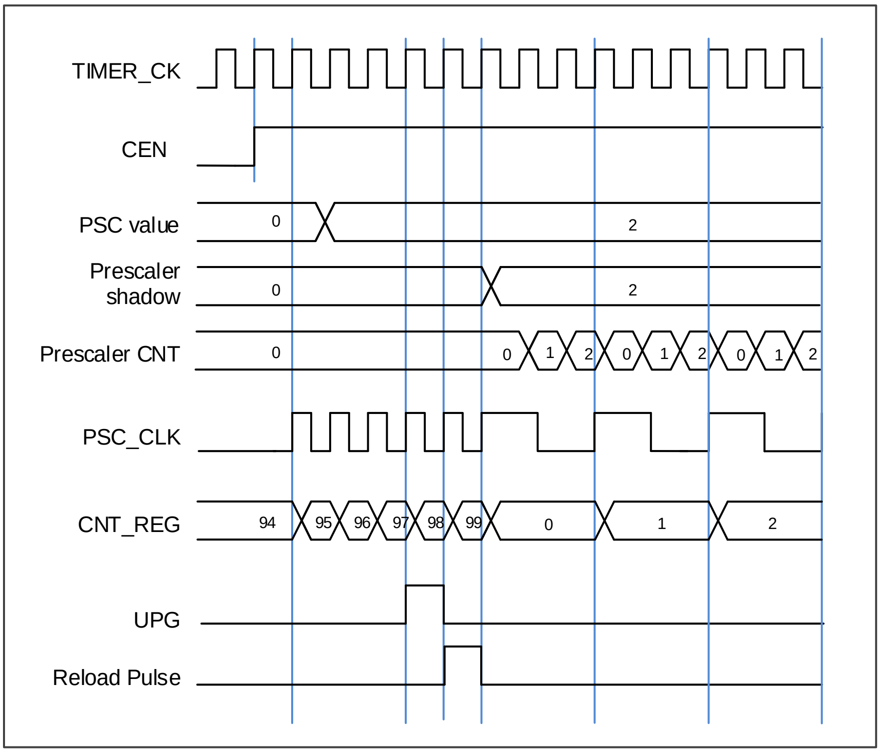
The TIMER_CK, which drives counter's prescaler to count, can be triggered by the event of rising or falling edge on the external pin TIMERx_CH0/TIMERx_CH1. This mode can be selected by setting TSCFG6[3:0] to 0x5, 0x6, 0x7.

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 TSCFG6[3:0] to 0x1,0x2,0x3,0x4.

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 18-60. 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.

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.

Figure 18-61. Up-counter timechart, PSC=0/2 and ***Figure 18-62. Up-counter timechart, change TIMERx_CAR on the go*** show some examples of the counter behavior for different clock prescaler factor when TIMERx_CAR=0x99.

Figure 18-61. Up-counter timechart, PSC=0/2

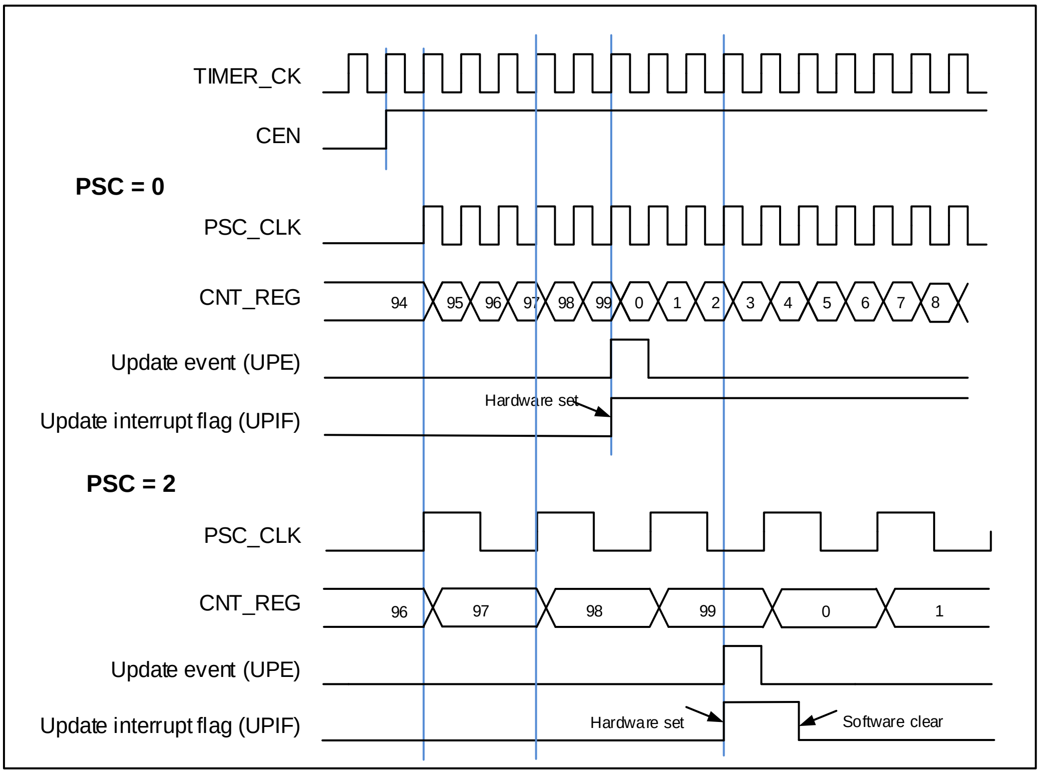
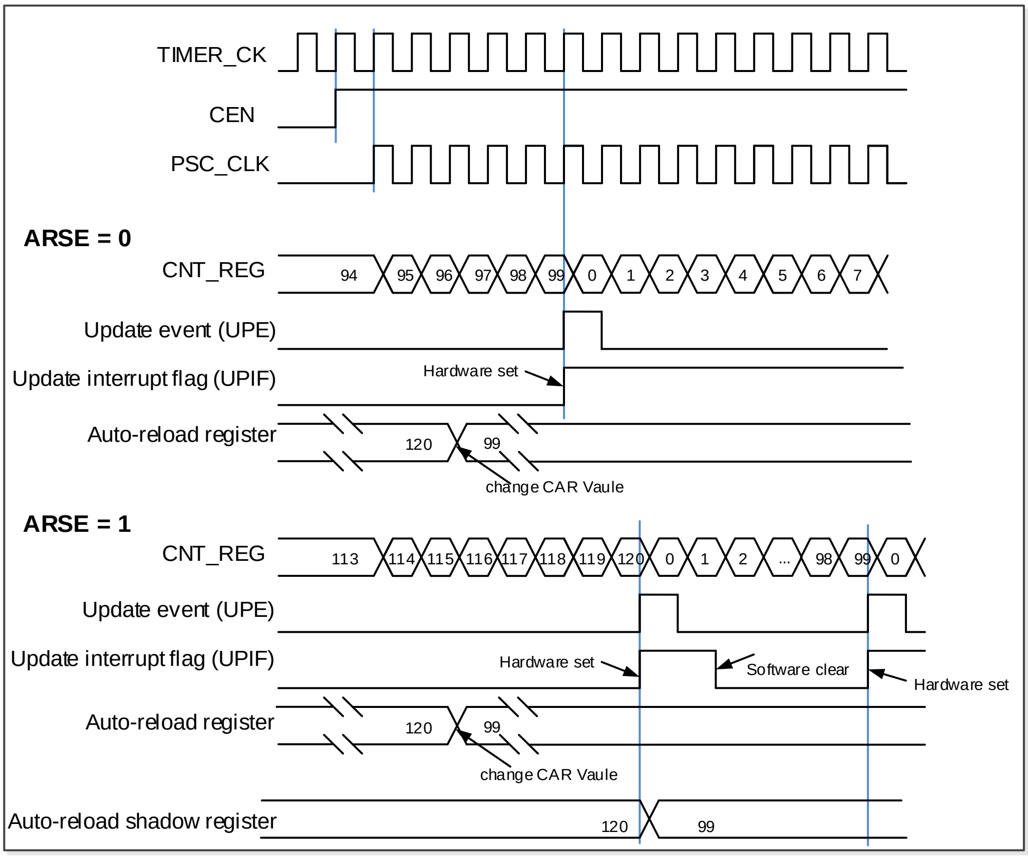


Figure 18-62. Up-counter timechart, change TIMERx_CAR on the go



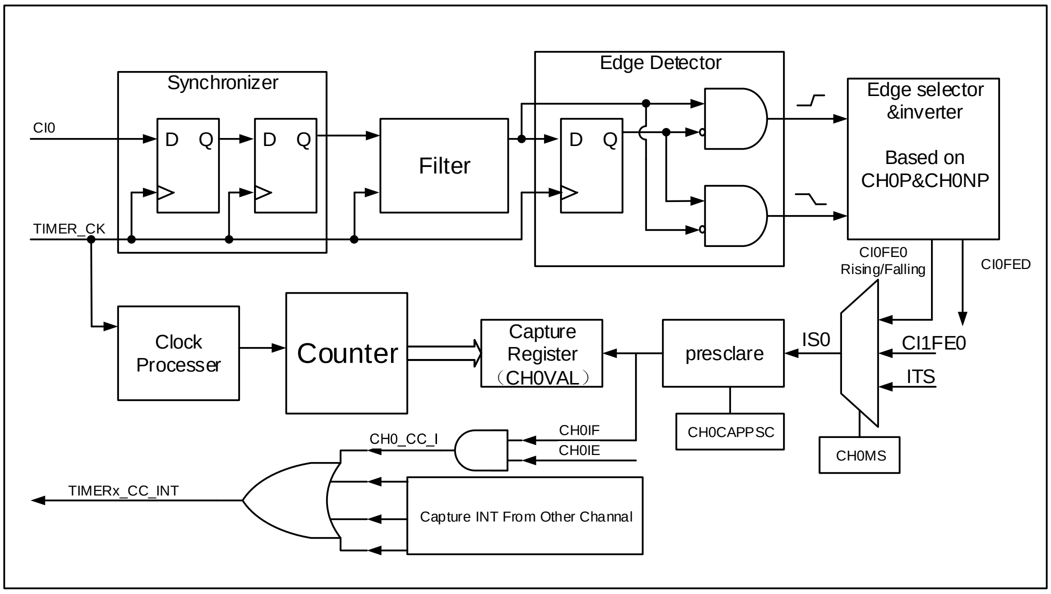
Input capture and output compare channels

The general level1 timer has two 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**

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 18-63. 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 and CHxDEN in TIMERx_DMAINTEN)
Enable the related interrupt enable; you can got 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 your 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.

The channel input capture function can be also used for pulse width measurement from signals on the TIMERx_CHx pins. For example, PWM signal connect to CI0 input. Select channel 0 capture signals to CI0 by setting CH0MS to 2'b01 in the channel control register (TIMERx_CHCTL0) and set capture on rising edge. Select channel 1 capture signal to CI0 by setting CH1MS to 2'b10 in the channel control register (TIMERx_CHCTL0) and set capture on falling edge. The counter set to restart mode and restart on channel 0 rising edge. Then the TIMERX_CH0CV can measure the PWM period and the TIMERx_CH1CV can measure the PWM duty.

- **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 CxCDE=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

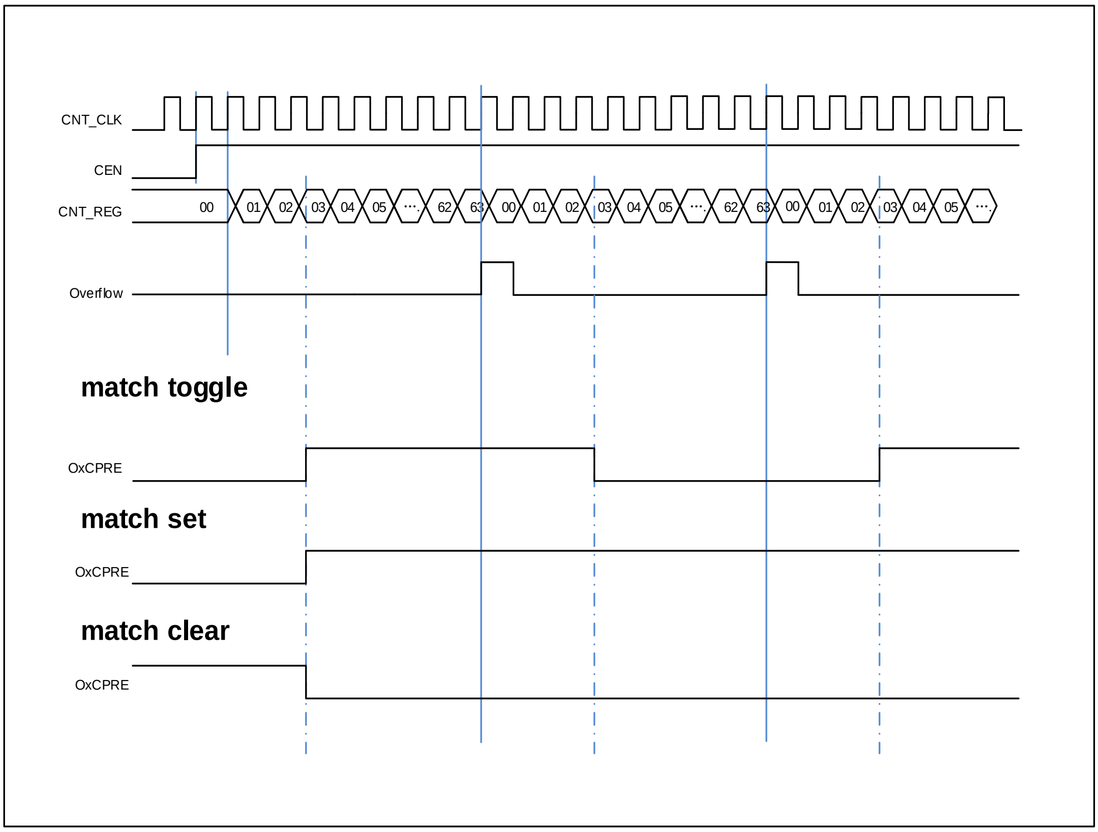
Step3: Interrupt/DMA-request enables configuration by ChxIE/CxCDE

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.

Figure 18-64. Output-compare under three modes below show the three compare modes toggle/set/clear. CAR=0x63, CHxVAL=0x3

Figure 18-64. Output-compare under three modes



Output PWM function

Iin the output PWM function (by setting the CHxCOMCTL bits to 3'b110 (PWM mode0) or to 3'b 111(PWM mode1), the channel can outputs PWM waveform according to the TIMERx_CAR registers and TIMERx_CHxCV registers.

Based on the counter mode, we have can also divide PWM into EAPWM (Edge aligned PWM) and CAPWM (Centre aligned PWM).

The EAPWM period is determined by TIMERx_CAR and duty cycle is by TIMERx_CHxCV. ***Figure 18-65. EAPWM timechart*** 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 18-66. CAPWM timechart*** shows the CAPWM output and interrupt 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 18-65. EAPWM timechart

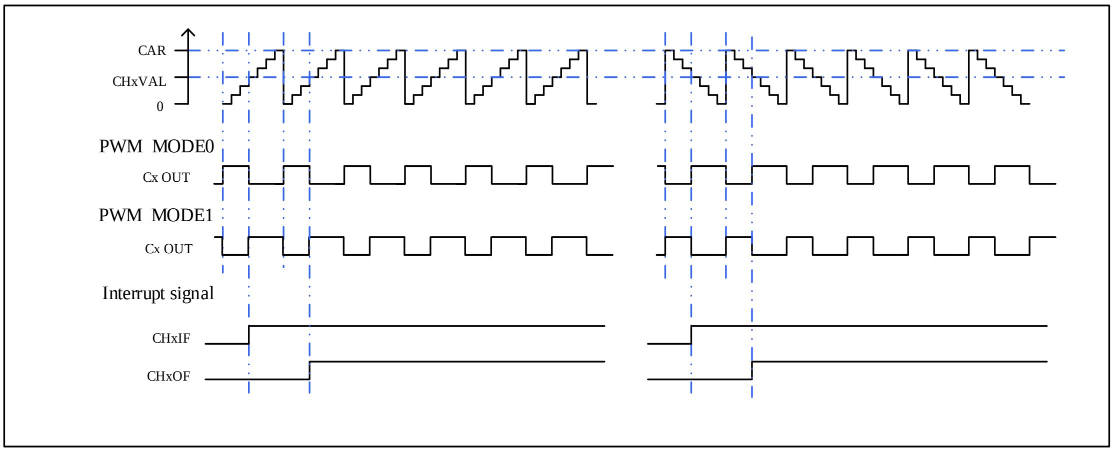
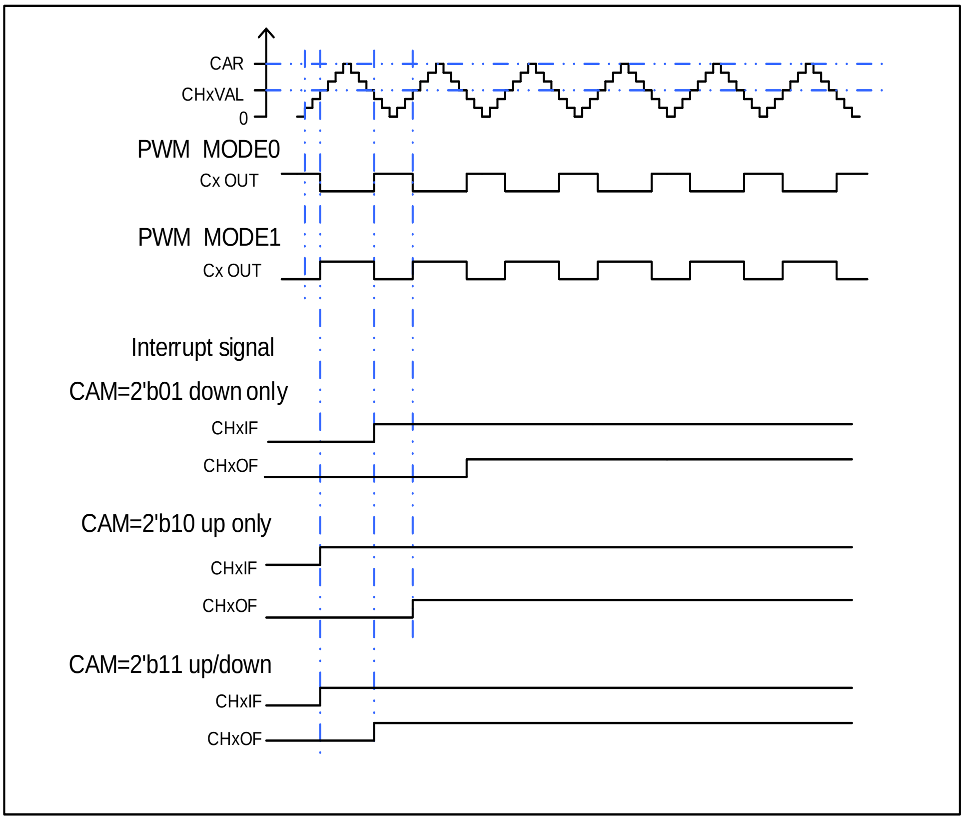


Figure 18-66. CAPWM timechart



Channel output reference 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.

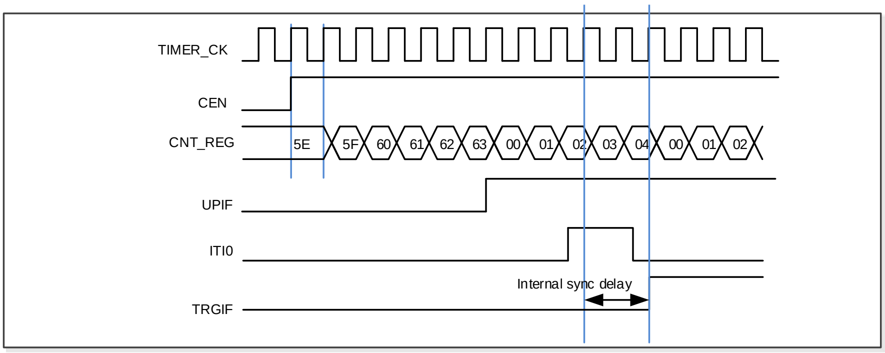
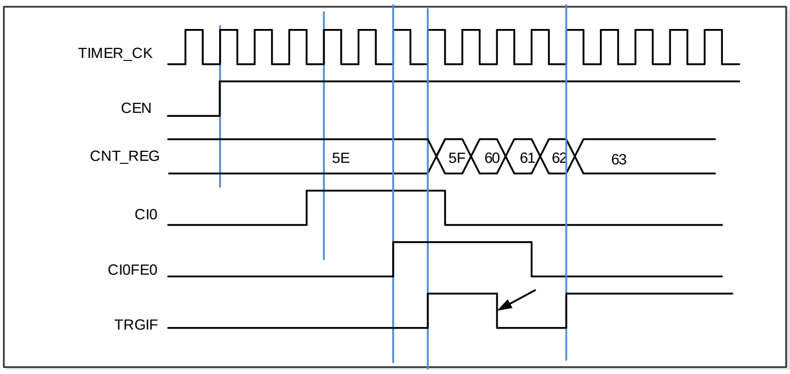
Master-slave management

The TIMERx can be synchronized with a trigger in several modes including the restart mode, the pause mode and the event mode.

- For GD32L233, these modes are 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.
- For GD32L235, these modes are selected by the TSCFGy[3:0] != 4b'0000 in SYSCFG_TIMER8CFG or SYSCFG_TIMER11CFG (y=3,4,5), The trigger input of these modes can be selected by the TSCFGy[3:0] in SYSCFG_TIMER8CFG or SYSCFG_TIMER11CFG, different TSCFGy[3:0] is correspond to different trigger input.

Table 18-8. Slave mode example table for GD32L233

	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: Reserved	If you choose the CI0FE0 or CI1FE1, configure the CHxP and CHxNP for the 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.
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 18-67. Restart mode

</td></tr>
<tr><td>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></td><td colspan="4">Figure 18-68. 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’b101
CI0FE0 is the selection.</td><td>TI0S=0(Non-xor)
[CH0NP==0, CH0P==0]
no inverted.</td><td>Filter is bypass in this example.</td></tr>
</table>

	Mode Selection	Source Selection	Polarity Selection	Filter and Prescaler

Figure 18-69. Event mode

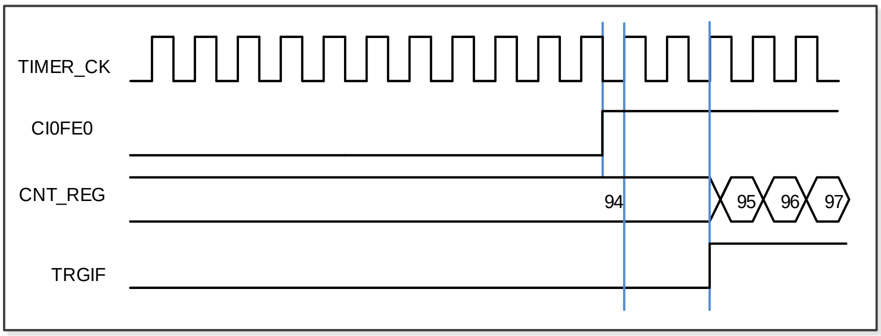
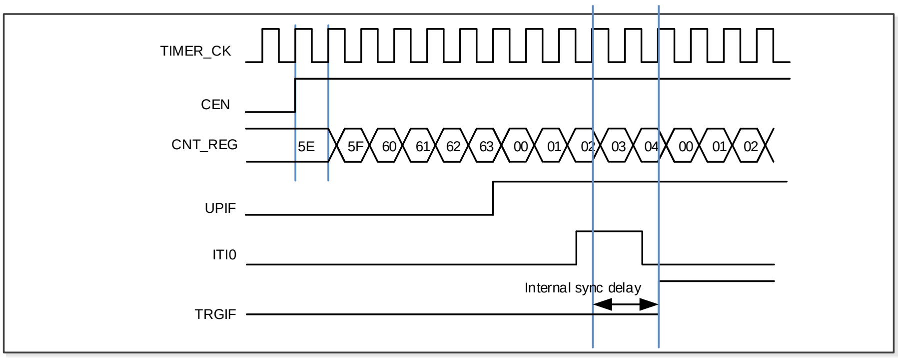


Table 18-9. Slave mode example table for GD32L235

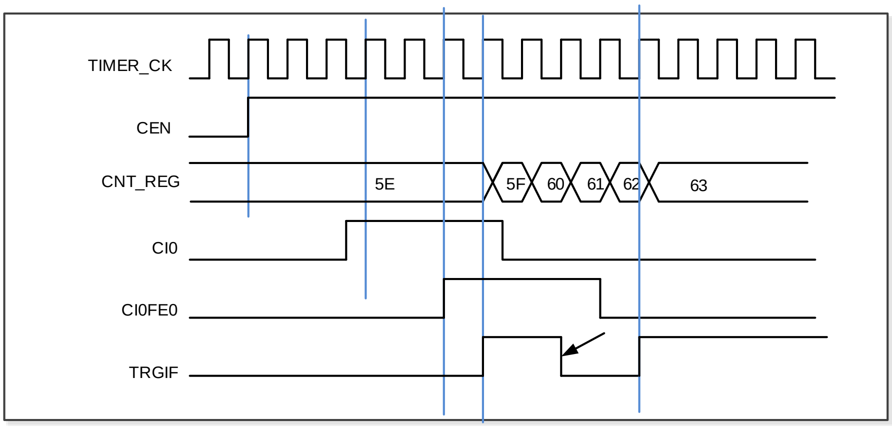
	Mode Selection	Source Selection	Polarity Selection	Filter and Prescaler
LIST	TSCFGy[3:0] y=3 (restart mode) y=4 (pause mode) y=5 (event mode)	TSCFGy[3:0] 0001: ITI0 0010: ITI1 0011: ITI2 0100: ITI3 0101: CI0F_ED 0110: CI0FE0 0111: CI1FE1 1000: Reserved	If you choose the CI0FE0 or CI1FE1, configure the CHxP and CHxNP for the 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.
Exam1	Restart mode The counter can be clear and restart when a rising trigger input.	TSCFG3[3:0] = 4'b 0001 ITI0 is the selection.	For ITI0, no polarity selector can be used.	For the ITI0, no filter and prescaler can be used.

Figure 18-70. Restart mode



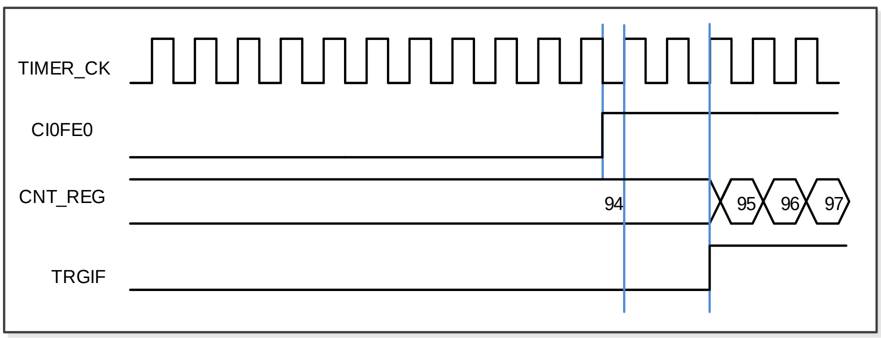
<table>
<tr><th></th><th>Mode Selection</th><th>Source Selection</th><th>Polarity Selection</th><th>Filter and Prescaler</th></tr>
<tr><td rowspan="2">Exam2</td><td>Pause mode
The counter can be paused when the trigger input is low.</td><td>TSCFG4[3:0] = 4’b 0110
CI0FE0 is the selection.</td><td>TI0S=0.（Non-xor）[CH0NP==0, 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 18-71. Pause mode



</td></tr>
<tr><td rowspan="2">Exam3</td><td>Event mode
The counter will start to count when a rising trigger input.</td><td>TSCFG5[3:0] = 4’b 0110
CI0FE0 is the selection.</td><td>TI0S=0(Non-xor)
[CH0NP==0, CH0P==0]
no inverted.</td><td>Filter is bypass in this example.</td></tr>
<tr><td colspan="4">

Figure 18-72. Event mode



</td></tr>
</table>

Single pulse mode

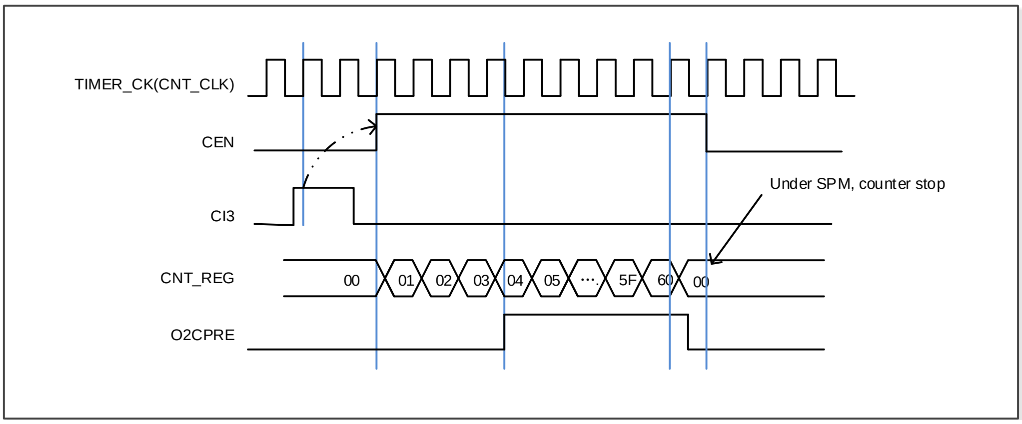
Single pulse mode is opposite to the repetitive mode, which can be enabled by setting SPM in TIMERx_CTL0. When you set SPM, the counter will be clear and stop when the next update event. In order to get pulse waveform, you can set the TIMERx to PWM mode or compare by CHxCOMCTL.

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 18-73. Single pulse mode TIMERx_CHxCV = 0x04 TIMERx_CAR=0x60 shows an example.

Figure 18-73. Single pulse mode TIMERx_CHxCV = 0x04 TIMERx_CAR=0x60



Timers interconnection

Refer to*General level0 timer (TIMERx, x=1, 2)*.

Timer debug mode

When the Cortex®-M23 halted, and the TIMERx_HOLD configuration bit in DBG_CTL2 register set to 1, the TIMERx counter stops.

18.3.5. TIMERx registers(x=8, 11)

TIMER8 base address: 0x4001 4000

TIMER11 base address: 0x4000 1800

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.

Slave mode configuration register (TIMERx_SMCFG)

For GD32L233xx devices

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								MSM	TRGS[2:0]			Reserved	SMC[2:0]		
								rw	rw				rw		

Bits	Fields	Descriptions
31:8	Reserved	Must be kept at reset value.
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
6:4	TRGS[2:0]	Trigger selection This bit-field specifies which signal is selected as the trigger input, which is used to

		synchronize the counter. 000: ITI0 001: ITI1 010: ITI2 011: ITI3 100: CI0F_ED 101: CI0FE0 110: CI1FE1 111: Reserved. These bits must not be changed when slave mode is enabled.
3	Reserved	Must be kept at reset value.
2:0	SMC[2:0]	Slave mode control 000: Disable mode. The slave mode is disabled; The prescaler is clocked directly by the internal clock (TIMER_CK) when CEN bit is set high. 001: Reserved. 010: Reserved. 011: Reserved. 100: Restart mode. The counter is reinitialized and an update event is generated on the rising edge of the selected trigger input. 101: Pause mode. The trigger input enables the counter clock when it is high and disables the counter clock when it is low. 110: Event mode. A rising edge of the trigger input enables the counter. 111: External clock mode0. The counter counts on the rising edges of the selected trigger.

For GD32L235xx devices

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								MSM	Reserved						
								rw							

Bits	Fields	Descriptions
31:8	Reserved	Must be kept at reset value.
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.

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									TRGIE	Reserved			CH1IE	CH0IE	UPIE
									rw				rw	rw	rw

Bits	Fields	Descriptions
31:7	Reserved	Must be kept at reset value.
6	TRGIE	Trigger interrupt enable 0: disabled 1: enabled
5:3	Reserved	Must be kept at reset value.
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					CH1OF	CH0OF	Reserved		TRGIF	Reserved			CH1IF	CH0IF	UPIF
					rc_w0	rc_w0			rc_w0				rc_w0	rc_w0	rc_w0

Bits	Fields	Descriptions
31:11	Reserved	Must be kept at reset value.
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.
5:3	Reserved	Must be kept at reset value.
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.			CH1G	CH0G	UPG
									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
5:3	Reserved	Must be kept at reset value.
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. 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															

<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>Reserved</td><td colspan="3">CH1COMCTL[2:0]</td><td>CH1COMSEN</td><td>CH1COMFEN</td><td colspan="2" rowspan="2">CH1MS[1:0]</td><td>Reserved</td><td colspan="3">CH0COMCTL[2:0]</td><td>CH0COMSEN</td><td>CH0COMFEN</td><td colspan="2" rowspan="2">CH0MS[1:0]</td></tr>
<tr><td colspan="4">CH1CAPFLT[3:0]</td><td colspan="2">CH1CAPPSC[1:0]</td><td colspan="4">CH0CAPFLT[3:0]</td><td colspan="2">CH0CAPPSC[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
31:15	Reserved	Must be kept at reset value.
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. For GD32L233 through TRGS bits in TIMERx_SMCFG register; For GD32L235 through TSCFG7[3:0] bit-field in SYSCFG_TIMERxCFG (x=8,11) register.
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: 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. For GD32L233 through TRGS bits in TIMERx_SMCFG register; For GD32L235 through TSCFG7[3:0] bit-field in SYSCFG_TIMERxCFG (x=8,11) 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 2 (TIMERx_CHCTL2)

Address offset: 0x20

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								CH1NP	Reserved	CH1P	CH1EN	CH0NP	Reserved	CH0P	CH0EN
								rw		rw	rw	rw		rw	rw

Bits	Fields	Descriptions
31:8	Reserved	Must be kept at reset value.
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 CH1EN 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.
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 input remap register(TIMERx_IRMP, x=8)

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
Reserved														CI0_RMP[1:0]	
														rw	

Bits	Fields	Descriptions
31:2	Reserved	Must be kept at reset value
1:0	CI0_RMP[1:0]	Channel 0 input remap 00: Channel 0 input is connected to GPIO(TIMER8_CH0) 01: Channel 0 input is connected to the LXTAL 10: Channel 0 input is connected to HXTAL/32 clock 11: Channel 0 input is connected to CKOUTSEL

Channel input remap register(TIMERx_IRMP, x=11)

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
Reserved														CI0_RMP[1:0]	
														rw	

Bits	Fields	Descriptions
31:2	Reserved	Must be kept at reset value
1:0	CI0_RMP[1:0]	Channel 0 input remap 00: Channel 0 input is connected to GPIO(TIMER11_CH0) 01: Channel 0 input is connected to the IRC32K 10: Channel 0 input is connected to LXTAL 11: Channel 0 input is connected to RTC_OUT

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.

18.4. General level3 timer (TIMERx, x=14,40)

18.4.1. Overview

The general level3 timer module (Timer14, 40) is a two-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 level3 timer has a 16-bit counter that can be used as an unsigned counter.

In addition, the general level3 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 counters incrementing in unison.

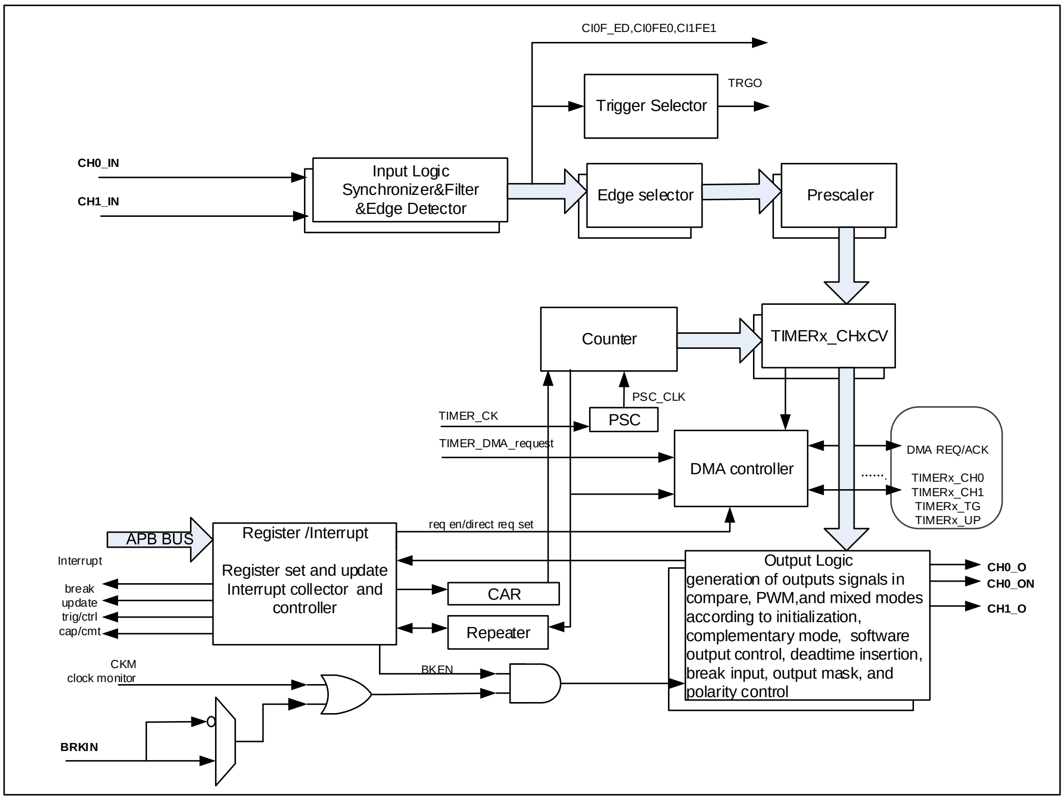
18.4.2. Characteristics

- Total channel num: 2.
- Counter width: 16 bits.
- Clock source of timer is selectable:
  internal clock, internal trigger, external input.
- Counter modes: count up only.
- Programmable prescaler: 16 bits.The factor can be changed ongoing.
- 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 on: update event, trigger event, compare/capture event, and break input.
- Daisy chaining of timer modules allows a single timer to start multiple timers.
- Timer synchronization allows selected timers to start counting on the same clock cycle.
- Timer master-slave management.

18.4.3. Block diagram

Figure 18-74. General level3 timer block **diagram** provides details of the internal configuration of the general level3 timer.

Figure 18-74. General level3 timer block diagram



18.4.4. 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[3:0] in SYSCFG_TIMER14CFG or SYSCFG_TIMER40CFG (y=0,1…7).

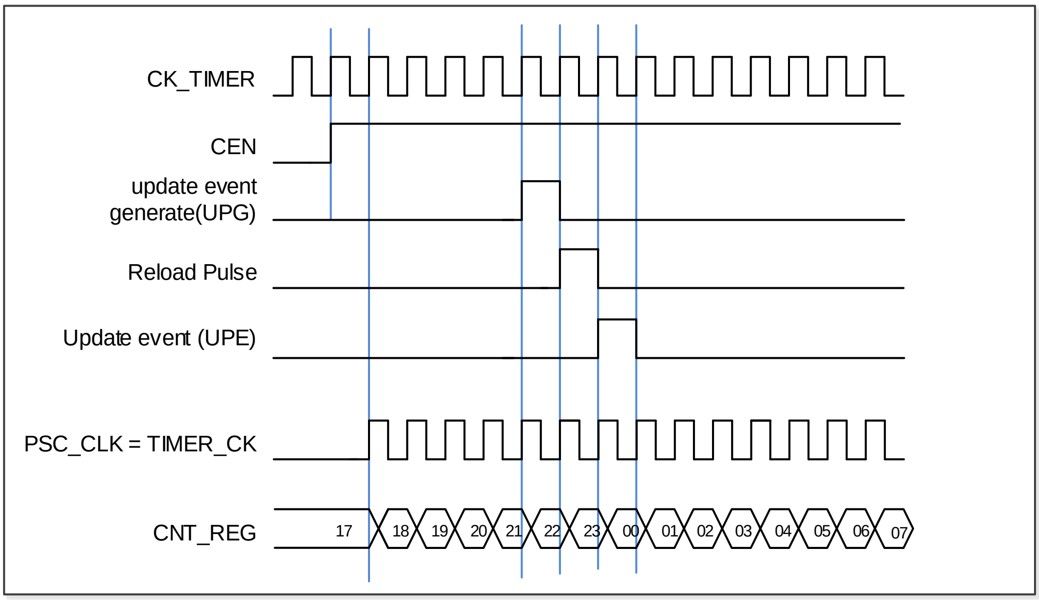
- TSCFGy[3:0] = 4'b0000 in SYSCFG_TIMER14CFG or SYSCFG_TIMER40CFG (y=0,1…7). Internal clock CK_TIMER is selected as timer clock source which is from module RCU.

The default clock source is the CK_TIMER for driving the counter prescaler when the TSCFGy[3:0] = 4'b0000 in SYSCFG_TIMER14CFG or SYSCFG_TIMER40CFG (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.

If TSCFGy[3:0] != 4'b0000 in SYSCFG_TIMER14CFG or SYSCFG_TIMER40CFG(y=0,1,2,6), the prescaler is clocked by other clock sources selected in the TSCFGy[3:0] (y=6) register, more details will be introduced later. When the TSCFGy[3:0] != 4'b0000 (y=3,4,5), the internal clock CK_TIMER is the counter prescaler driving clock source.

Figure 18-75. Normal mode, internal clock divided by 1



- TSCFG6[3:0] != 4'b0000 (external clock mode 0). External input pin is selected as timer clock source

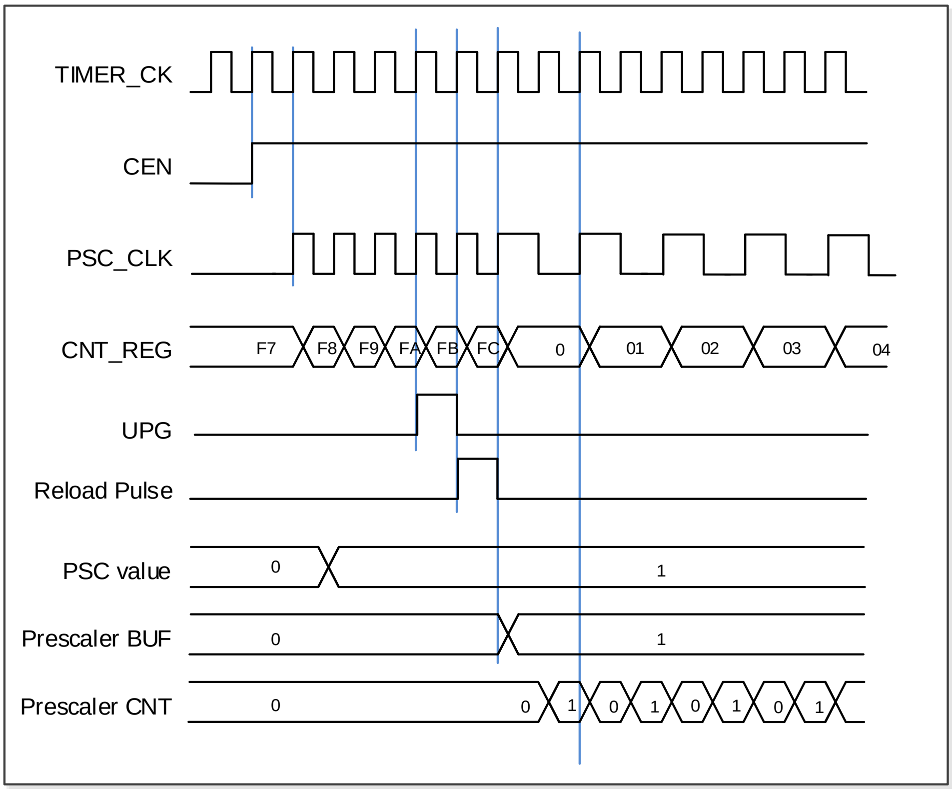
The TIMER_CK, which drives counter's prescaler to count, can be triggered by the event of rising or falling edge on the external pin TIMERx_CH0/TIMERx_CH1. This mode can be selected by setting TSCFG6[3:0] to 0x5, 0x6, 0x7.

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 TSCFG6[3:0] to 0x1, 0x2, 0x3, 0x4.

Prescaler

The prescaler can divide the timer clock (TIMER_CK) to a counter clock (PSC_CLK) by any factor between 1 and 65536. It is controlled by prescaler register (TIMERx_PSC) which can be changed ongoing but is taken into account at the next update event.

Figure 18-76. Counter timing diagram with prescaler division change from 1 to 2



Up counting mode

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 events will be generated after (TIMERx_CREP+1) times of overflow. Otherwise the update event is generated each time when 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 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 registers (repetition counter, auto reload register, prescaler register) are updated.

Figure 18-77. Timing chart of up counting mode, PSC=0/1

_show some examples of the counter behavior for different clock prescaler factor when TIMERx_CAR=0x63.

Figure 18-77. Timing chart of up counting mode, PSC=0/1

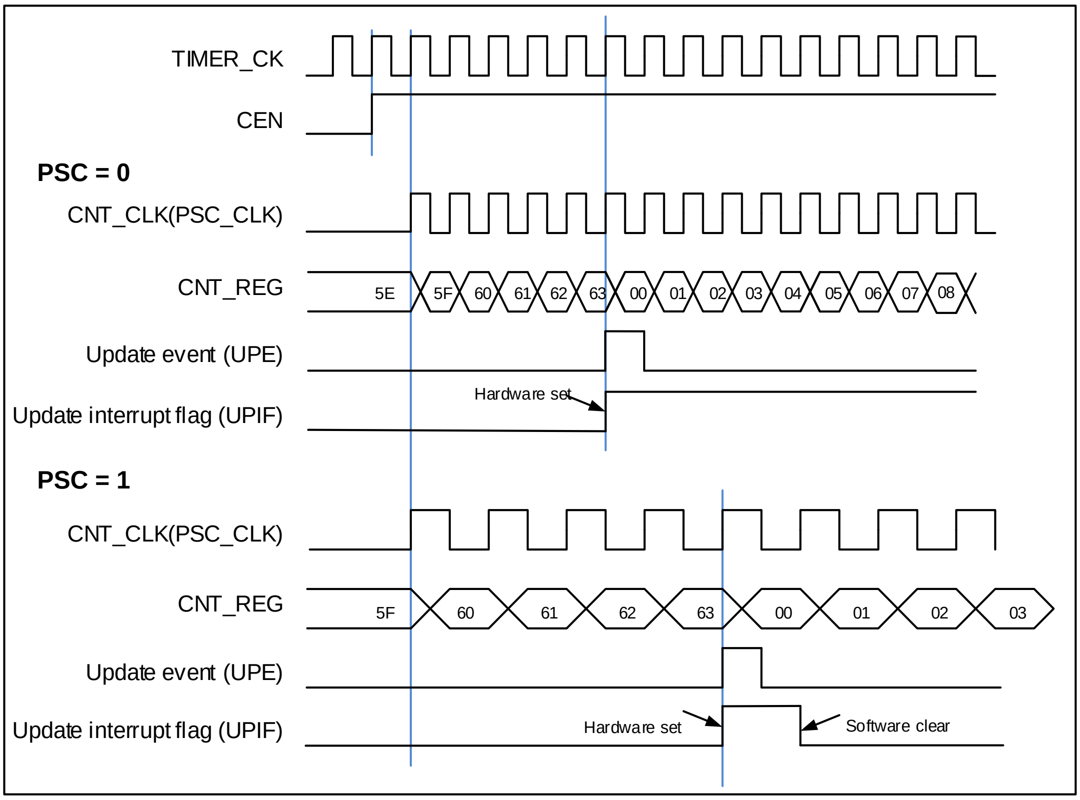
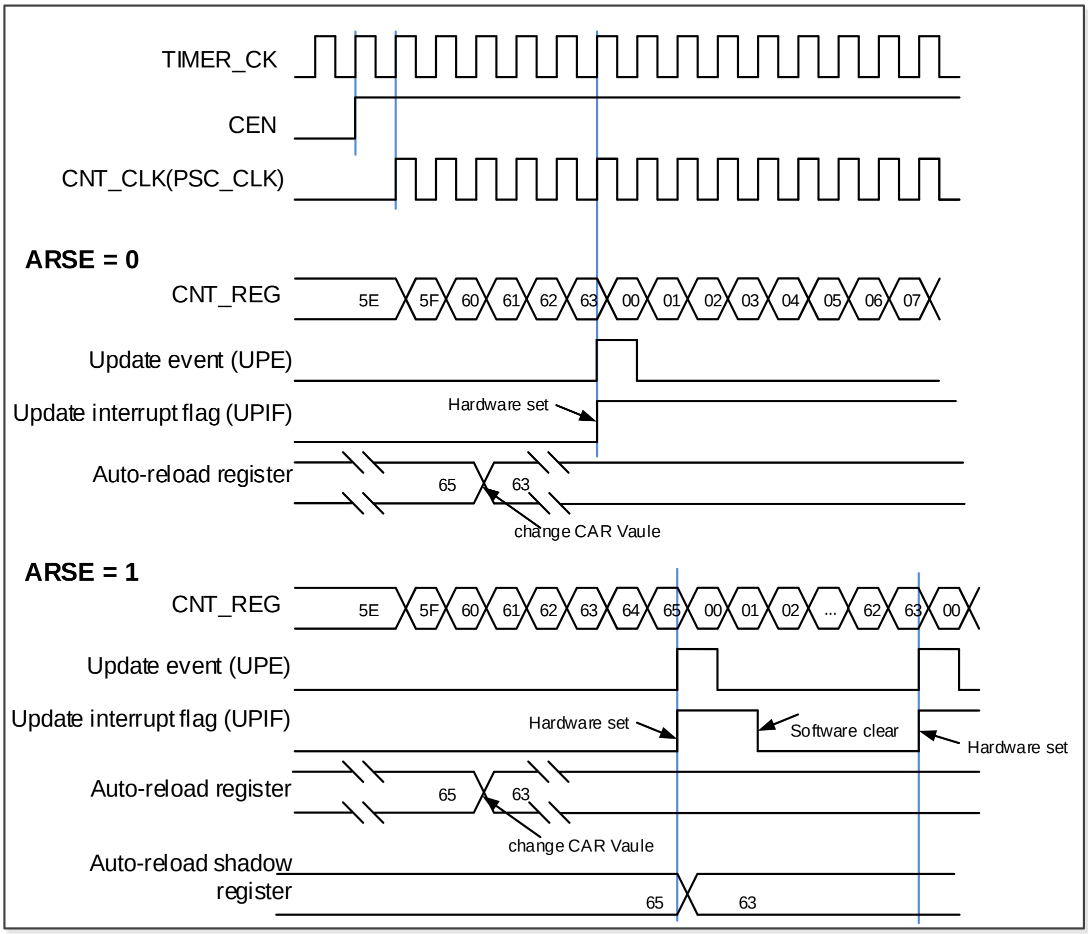


Figure 18-78. Up-counter timechart, change TIMERx_CAR ongoing

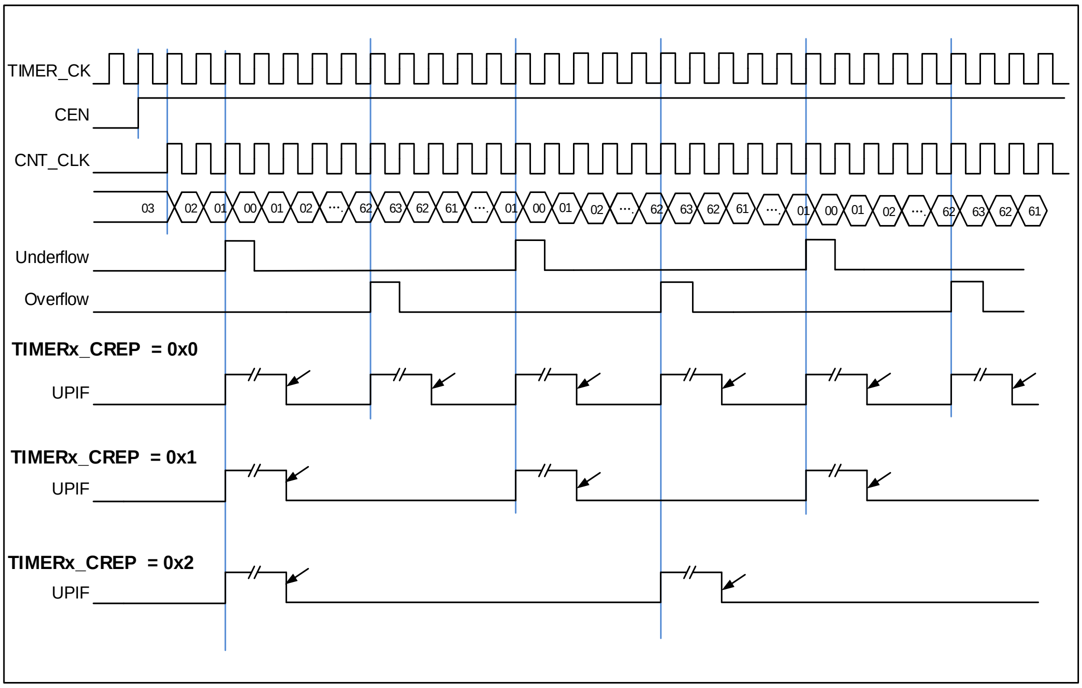


Repetition counter

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 18-79. Repetition counter timing chart of up counting mode



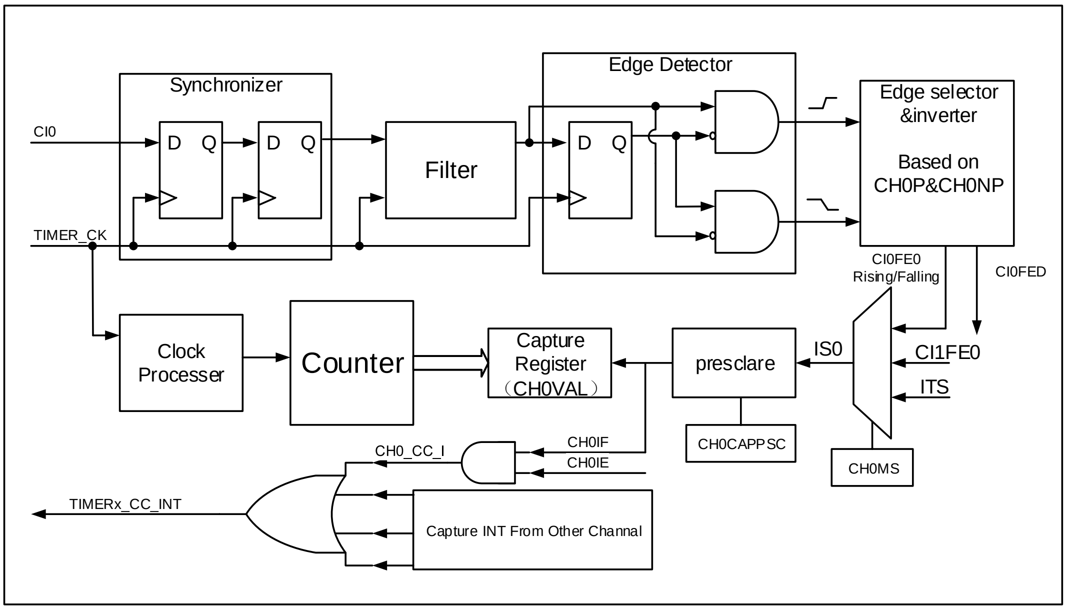
Input capture and output compare channels

The general level3 timer has two 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.

Input capture mode

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 18-80. Input capture logic



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 enable; you can got 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.

The input capture mode can be also used for pulse width measurement from signals on the TIMERx_CHx pins. For example, PWM signal connect to CI0 input. Select channel 0 capture signals to CI0 by setting CH0MS to 2'b01 in the channel control register (TIMERx_CHCTL0) and set capture on rising edge. Select channel 1 capture signal to CI0 by setting CH1MS to 2'b10 in the channel control register (TIMERx_CHCTL0) and set capture on falling edge. The counter set to restart mode and restart on channel 0 rising edge. Then the TIMERX_CH0CV can measure the PWM period and the TIMERx_CH1CV can measure the PWM duty.

Output compare mode

Figure 18-81. Output compare logic (with complementary output, x=0)

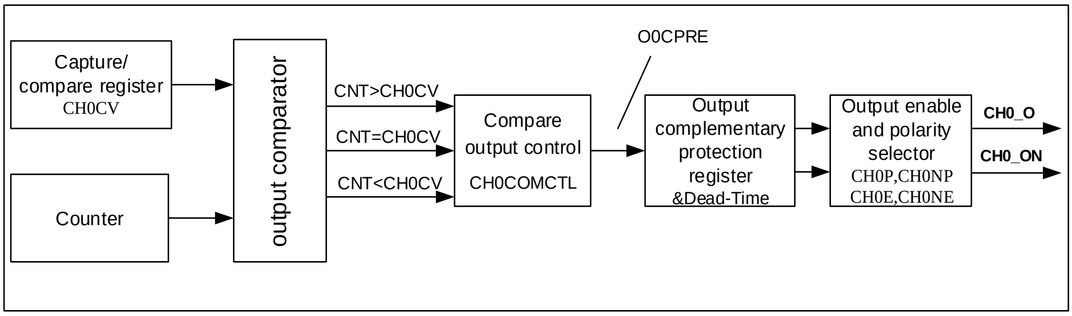


Figure 18-82. Output compare logic (CH1_O)

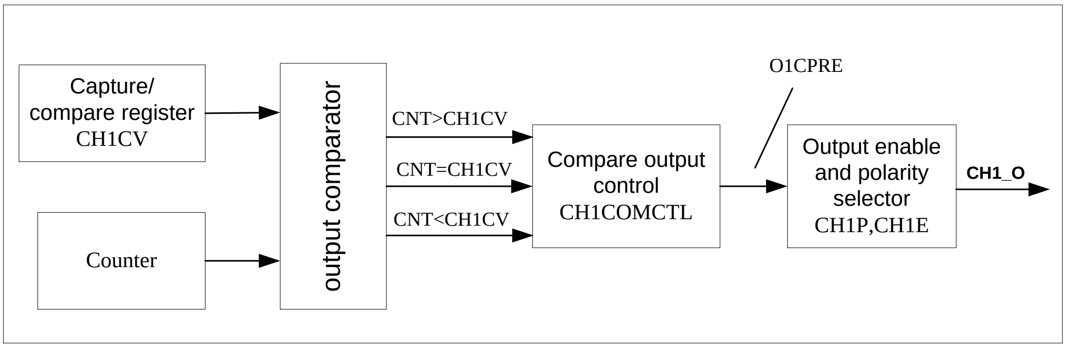


Figure 18-81. Output compare logic (with complementary output, x=0) and *Figure 18-82. Output compare logic (CH1_O)* show the logic circuit of output compare mode. The relationship between the channel output signal CHx_O/CHx_ON and the OxCPRE signal (more details refer to *Complementary outputs*) is 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). For examples, 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.

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 ***Complementary outputs*** 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 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

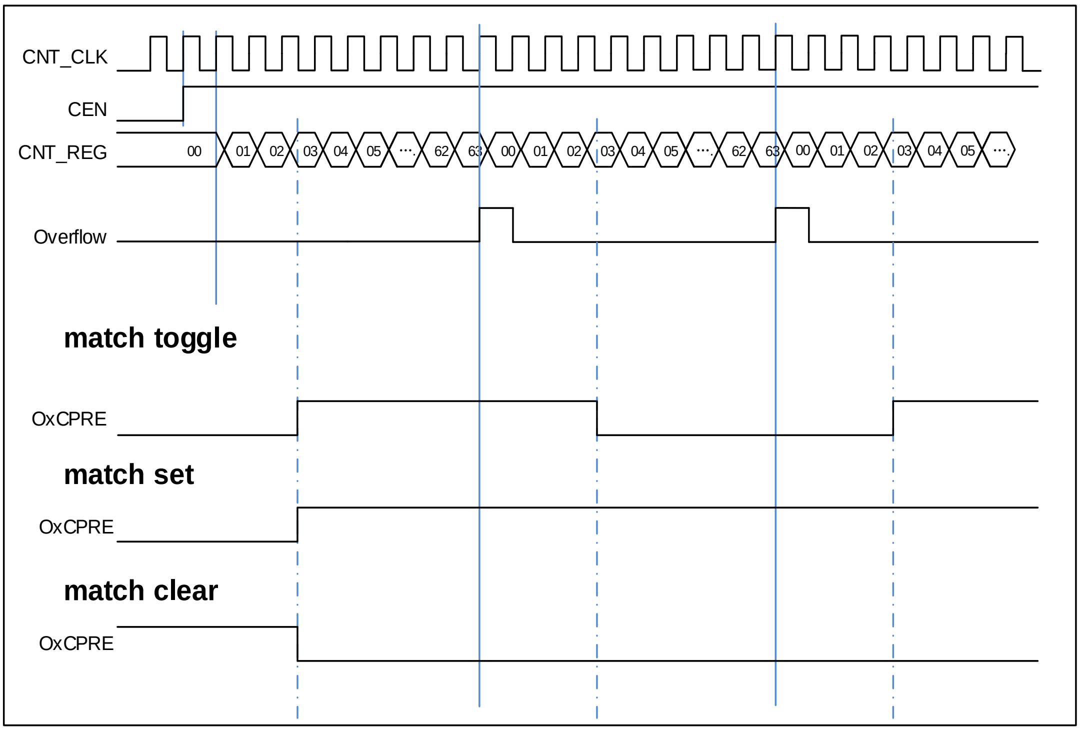
About the CHxVAL; you can change it ongoing to meet the waveform you expected.

Step5: Start the counter by CEN.

Figure 18-83. Output-compare in three modes

show the three compare modes toggle/set/clear. CAR=0x63, CHxVAL=0x3

Figure 18-83. Output-compare in three modes



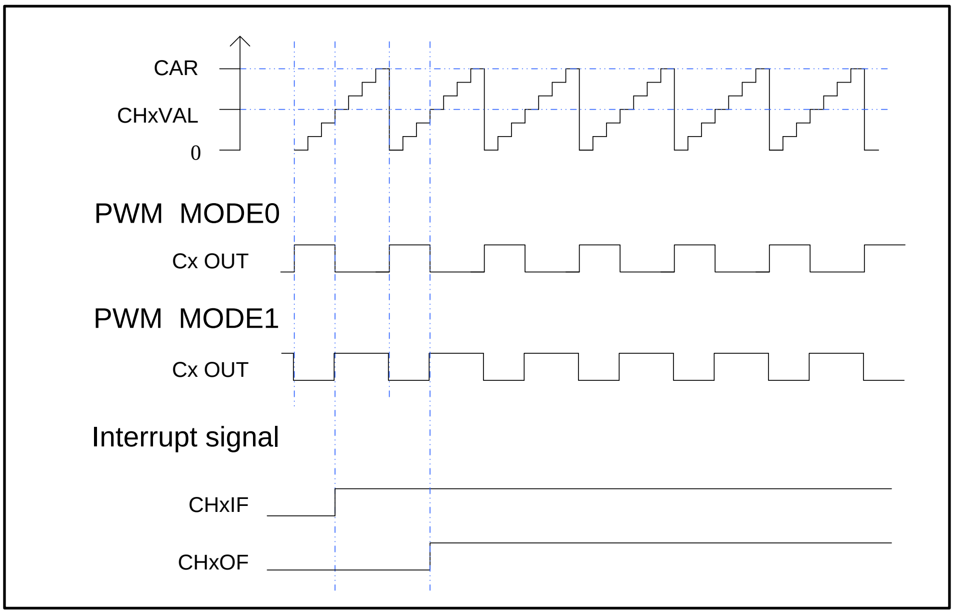
PWM mode

In the output PWM mode (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 18-84. 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 18-84. PWM mode timechart



Channel output reference 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.

Complementary outputs

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 2 channels, but only the first 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 18-10. Complementary outputs controlled by parameters

<table>
<tr><th colspan="5">Complementary Parameters</th><th colspan="2">Output Status</th></tr>
<tr><th>POEN</th><th>ROS</th><th>IOS</th><th>ChxEN</th><th>CHxNEN</th><th>CHx_O</th><th>CHx_ON</th></tr>
<tr><td rowspan="5">0</td><td rowspan="5">0/1</td><td rowspan="4">0</td><td rowspan="2">0</td><td>0</td><td colspan="2">CHx_O / CHx_ON = LOW
CHx_O / CHx_ON output disable[1].</td></tr>
<tr><td>1</td><td colspan="2" rowspan="3">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]</td></tr>
<tr><td rowspan="2">1</td><td>0</td></tr>
<tr><td>1</td></tr>
<tr><td>1</td><td>x</td><td>x</td><td colspan="2">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.</td></tr>
<tr><td rowspan="8">1</td><td rowspan="4">0</td><td rowspan="8">0/1</td><td rowspan="2">0</td><td>0</td><td colspan="2">CHx_O/CHx_ON = LOW
CHx_O/CHx_ON output disable.</td></tr>
<tr><td>1</td><td>CHx_O = LOW
CHx_O output disable.</td><td>CHx_ON =OxCPRE⊕[4]CHxNP
CHx_ON output enable.</td></tr>
<tr><td rowspan="2">1</td><td>0</td><td>CHx_O=OxCPRE⊕CHxP
CHx_O output enable.</td><td>CHx_ON = LOW
CHx_ON output disable.</td></tr>
<tr><td>1</td><td>CHx_O=OxCPRE⊕CHxP
CHx_O output enable.</td><td>CHx_ON =(!OxCPRE)[5]⊕ CHxNP.
CHx_ON output enable.</td></tr>
<tr><td rowspan="4">1</td><td rowspan="2">0</td><td>0</td><td>CHx_O = CHxP
CHx_O output “off-state”.</td><td>CHx_ON = CHxNP
CHx_ON output “off-state”.</td></tr>
<tr><td>1</td><td>CHx_O = CHxP
CHx_O output “off-state”</td><td>CHx_ON =OxCPRE⊕CHxNP
CHx_ON output enable</td></tr>
<tr><td rowspan="2">1</td><td>0</td><td>CHx_O=OxCPRE⊕CHxP
CHx_O output enable</td><td>CHx_ON = CHxNP
CHx_ON output “off-state”.</td></tr>
<tr><td>1</td><td>CHx_O=OxCPRE⊕CHxP
CHx_O output enable</td><td>CHx_ON =(!OxCPRE)⊕ CHxNP
CHx_ON output enable.</td></tr>
</table>

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.

Dead time insertion

The dead time insertion is enabled when both ChxEN and CHxNEN are 1’b1, and set POEN is also necessary. The field named DTCFG defines the dead time delay that can be used for channel 0. 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.

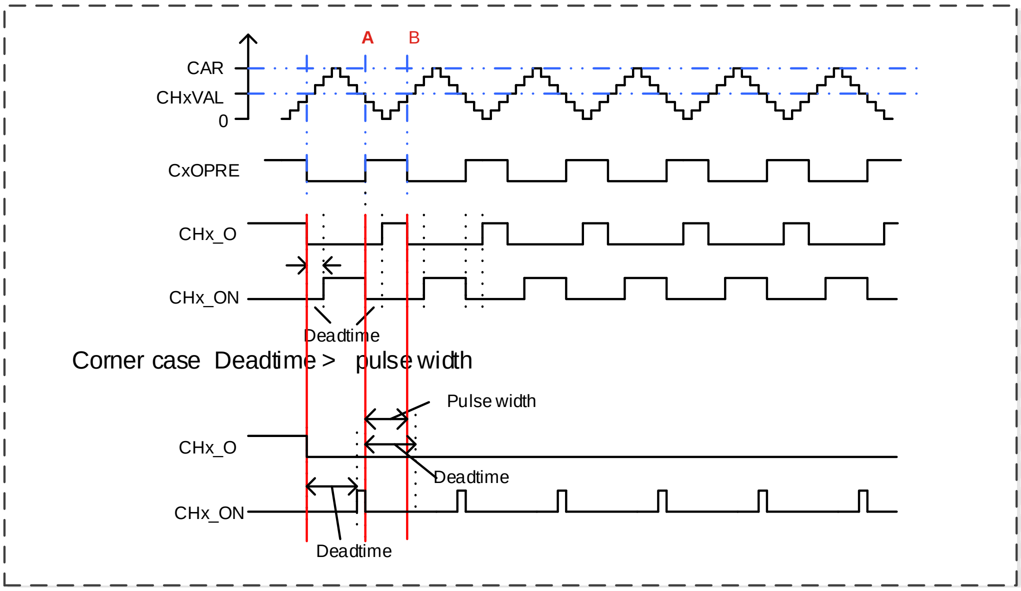
When the channel (x) match (TIMERx counter = CHxVAL) occurs, OxCPRE will be toggled because under PWM0 mode. At point A in the ***Figure 18-85. Complementary output with dead-time insertion***. CHx_O signal remains at the low value until the end of the deadtime delay, while CHx_ON will be cleared at 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 18-85. 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 18-85. Complementary output with dead-time insertion.



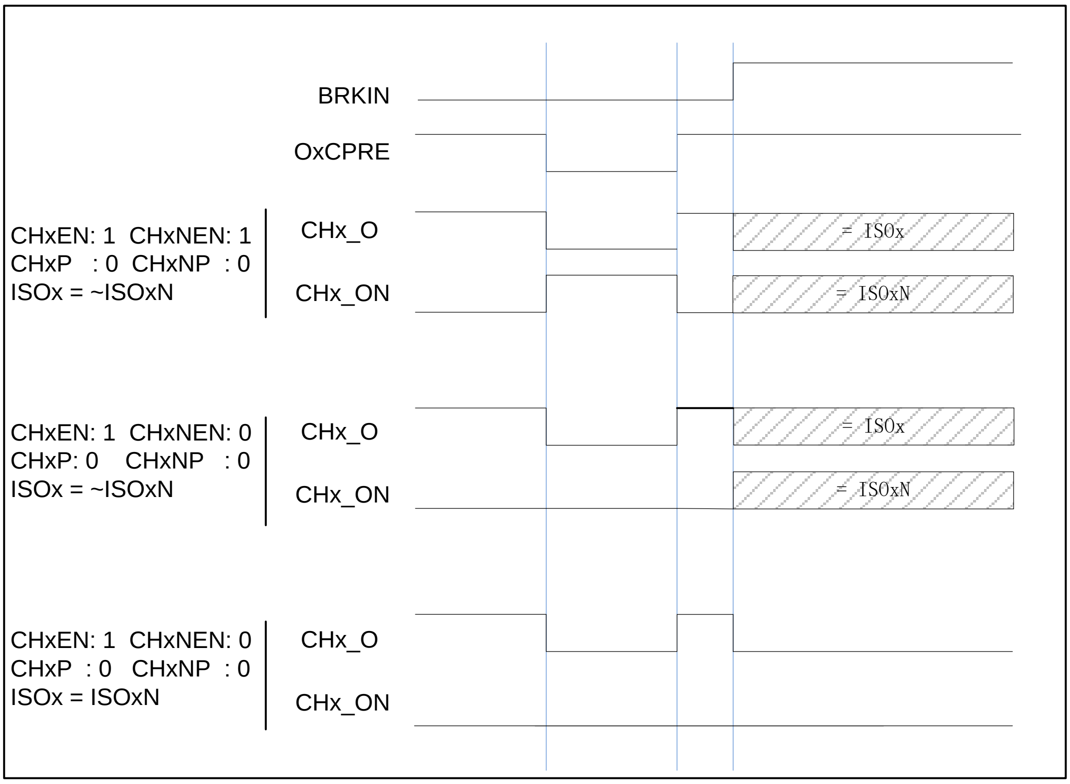
Break function

In this function, 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.

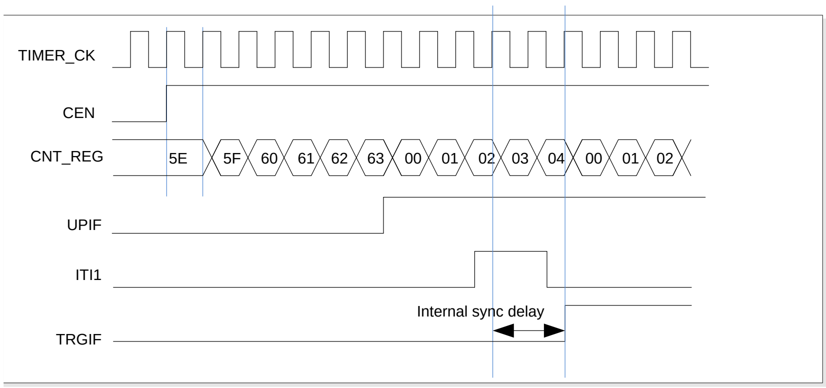
Figure 18-86. Output behavior in response to a break(The break high active)

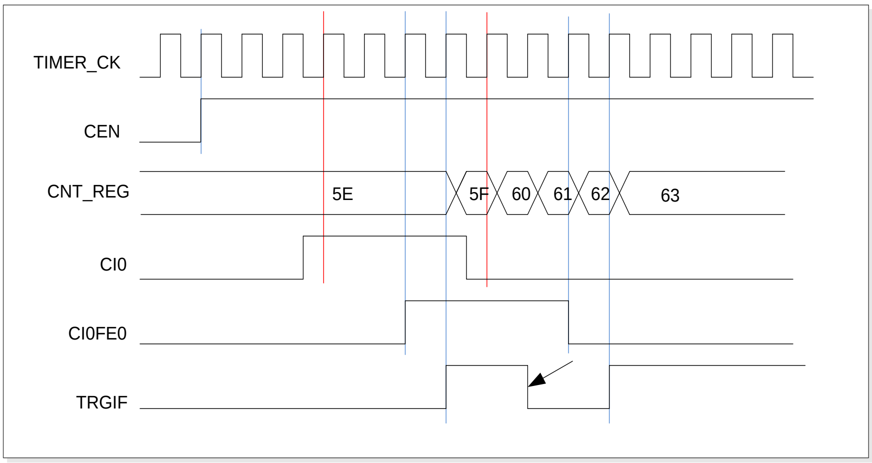


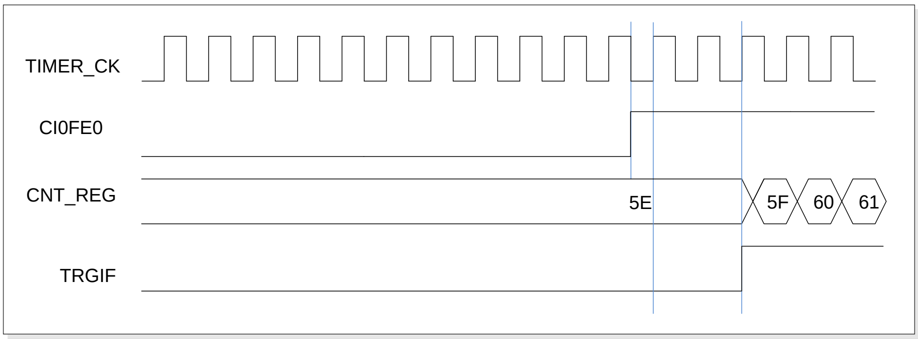
Master-slave management

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 TSCFGy[3:0] in SYSCFG_TIMER14CFG or SYSCFG_TIMER40CFG (y=3,4,5).

Table 18-11. Slave mode example table

<table>
<tr><th></th><th>Mode Selection</th><th>Source Selection</th><th>Polarity Selection</th><th>Filter and Prescaler</th></tr>
<tr><td>LIST</td><td>TSCFGy[3:0]
y=3 (restart mode)
y=4 (pause mode)
y=5 (event mode)</td><td>TSCFGy[3:0]
0001: ITI0
0010: ITI1
0011: ITI2
0100: ITI3
0101: CI0F_ED
0110: CI0FE0
0111: CI1FE1
1000: Reserved</td><td>If you choose the CI0FE0 or CI1FE1, configure the CHxP and CHxNP for the polarity selection and inversion.</td><td>For the ITIx no filter and prescaler can be used.
For the Cix, configure Filter by CHxCAPFLT, no prescaler can be used.</td></tr>
<tr><td rowspan="2">Exam1</td><td>Restart mode
The counter can be clear and restart when a rising trigger input.</td><td>TSCFG3[3:0] = 4’b 0001
ITI0 is the selection.</td><td>For ITI0, no polarity selector can be used.</td><td>For the ITI1, no filter and prescaler can be used.</td></tr>
<tr><td colspan="4">Figure 18-87. 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>TSCFG4[3:0] = 4’b0110
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 18-88. Pause mode</td></tr>
</table>

<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">


</td></tr>
<tr><td rowspan="2">Exam3</td><td>Event mode
The counter will start to count when a rising trigger input.</td><td>TSCFG5[3:0] = 4’b0110
CI0FE0 is the selection.</td><td>TI0S=0.(Non-xor)
[CH0NP==0, CH0P==0]
no inverted.</td><td>Filter is bypass in this example.</td></tr>
<tr><td colspan="4">
Figure 18-89. Event mode


</td></tr>
</table>

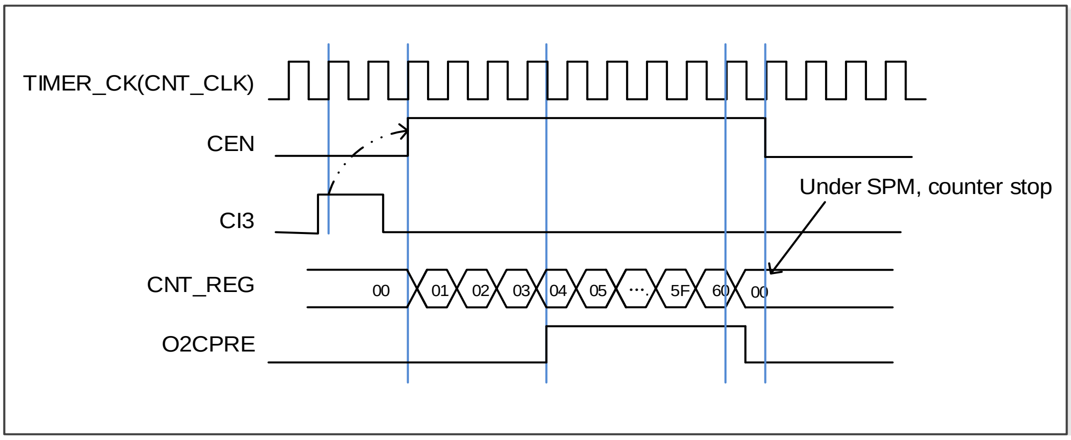
Single pulse mode

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 18-90. Single pulse mode TIMERx_CHxCV = 0x04 TIMERx_CAR=0x60



Timers interconnection

Refer to ***General level0 timer (TIMERx, x=1, 2)***.

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 one 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™-M3 halted, and the TIMERx_HOLD configuration bit in DBG_CTL1 register set to 1, the TIMERx counter stops.

18.4.5. TIMERx registers(x=14,40)

TIMER14 base address: 0x4001 4000

TIMER40 base address: 0x4001 D000

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 ratio between the timer clock (TIMER_CK) and the dead-time and sampling clock (DTS), which is used by the dead-time generators and the digital filters. 00: $f_{DTS}=f_{TIMER\_CK}$ 01: $f_{DTS}= f_{TIMER\_CK} /2$ 10: $f_{DTS}= f_{TIMER\_CK} /4$ 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
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															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved					ISO1	ISO0N	ISO0	Reserved	MMC[2:0]			DMAS	CCUC	Reserved	CCSE
					rw	rw	rw		rw			rw	rw		rw

Bits	Fields	Descriptions
31:11	Reserved	Must be kept at reset value
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	Reserved	Must be kept at reset value
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
Reserved								MSM	Reserved						
								rw		rw				rw	

Bits	Fields	Descriptions
31:8	Reserved	Must be kept at reset value
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			CH1DEN	CH0DEN	UPDEN	BRKIE	TRGIE	CMTIE	Reserved		CH1IE	CH0IE	UPIE
	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
13:11	Reserved	Must be kept at reset value
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
4:3	Reserved	Must be kept at reset value
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					CH1OF	CH0OF	Reserved	BRKIF	TRGIF	CMTIF	Reserved.		CH1IF	CH0IF	UPIF
					rc_w0	rc_w0		rc_w0	rc_w0	rc_w0			rc_w0	rc_w0	rc_w0

Bits	Fields	Descriptions
31:11	Reserved	Must be kept at reset value
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	Reserved	Must be kept at reset value.
7	BRKIF	Break interrupt flag This flag is set by hardware when the break input goes active, and cleared by software if the break input is not active. 0: No active level break has been detected. 1: An active level has been detected.
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	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
4:3	Reserved	Must be kept at reset value
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								BRKG	TRGG	CMTG	Reserved		CH1G	CH0G	UPG
								w	w	w			w	w	w

Bits	Fields	Descriptions
31:8	Reserved	Must be kept at reset value
7	BRKG	Break event generation This bit is set by software and cleared when the next CK_TIMER clock arrives. 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 when the next CK_TIMER clock arrives. 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
4:3	Reserved	Must be kept at reset value
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															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	CH1COMCTL[2:0]			CH1COMSEN	CH1COMFEN	CH1MS[1:0]		Reserved	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
31:15	Reserved	Must be kept at reset value
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 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 TSCFG7[3:0] bit-field in SYSCFG_TIMERxCFG (x=14,40) register.
7	Reserved	Must be kept at reset value
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. 000: Frozen. 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 matches the output compare register TIMERx_CH0CV. 010: Clear the channel output. O0CPRE signal is forced low when the counter matches the output compare register TIMERx_CH0CV. 011: Toggle on match. O0CPRE toggles when the counter matches the output compare register TIMERx_CH0CV. 100: Force low. O0CPRE is forced low level. 101: Force high. O0CPRE is forced high level. 110: 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. 111: 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. 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 CI0FE1 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 TSCFG7[3:0] bit-field in SYSCFG_TIMERxCFG (x=14,40) 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 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								CH1NP	Reserved	CH1P	CH1EN	CH0NP	CH0NEN	CH0P	CH0EN
								rw		rw	rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31:8	Reserved	Must be kept at reset value
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 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 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.

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.

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.

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															

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
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	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 CCS 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_CHCTL0/1 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 t_{DT}, t_{DT}=t_{DTS}. DTCFG [7:5] =3'b 10x: Dtvalue = (64+DTCFG [5:0])xt_{DT}, t_{DT} =t_{DTS}*2. DTCFG [7:5] =3'b 110: Dtvalue = (32+DTCFG [4:0])xt_{DT}, t_{DT}=t_{DTS}*8.

DTCFG [7:5] =3'b 111: Dtvalue = (32+DTCFG [4:0])xt_{DT}, t_{DT} =t_{DTS}*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.

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

18.5. Basic timer (TIMERx, x=5, 6)

18.5.1. Overview

The basic timer module (Timer5, 6) reference is a 16-bit counter that can be used as an unsigned counter. The basic timer can be configured to generate DMA request and TRGO to DAC.

18.5.2. Characteristics

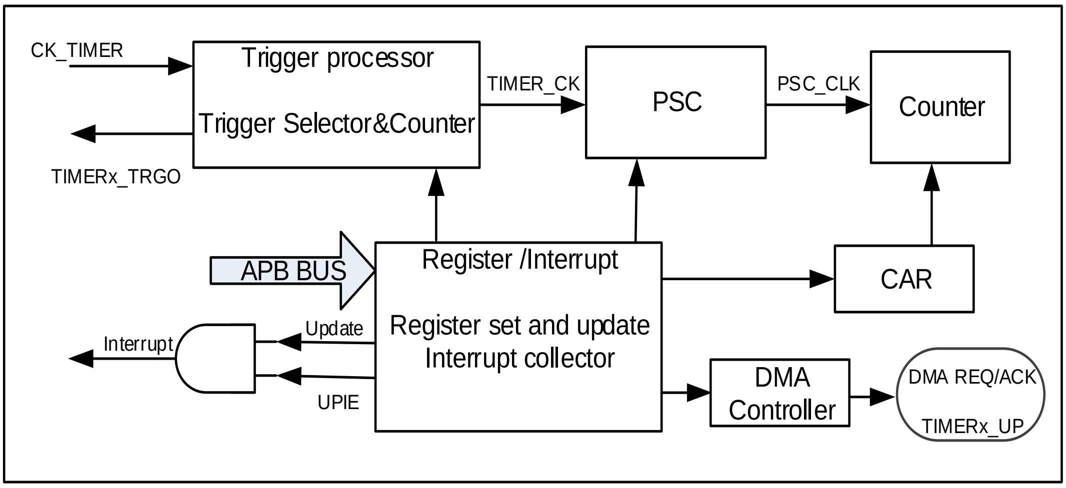
- Counter width: 16-bit.
- Source of count clock is internal clock only.
- Multiple counter modes: count up.
- Programmable prescaler: 16-bit. The factor can be changed ongoing.
- Single pulse mode is supported.
- Auto-reload function.
- Interrupt output or DMA request on update event.

18.5.3. Block diagram

Figure 18-91. Basic timer block diagram

provides details on the internal configuration of the basic timer.

Figure 18-91. Basic timer block diagram



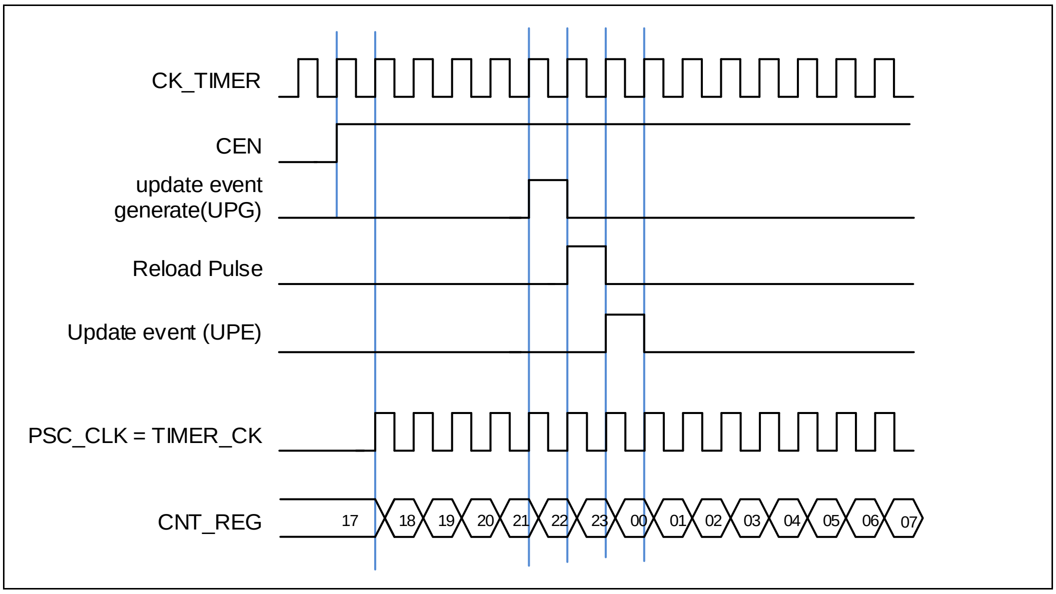
18.5.4. Function overview

Clock source configuration

The basic TIMER can only being clocked by the internal timer clock CK_TIMER, which is from the source named CK_TIMER in RCU

The TIMER_CK, driven counter's prescaler to count, is equal to 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.

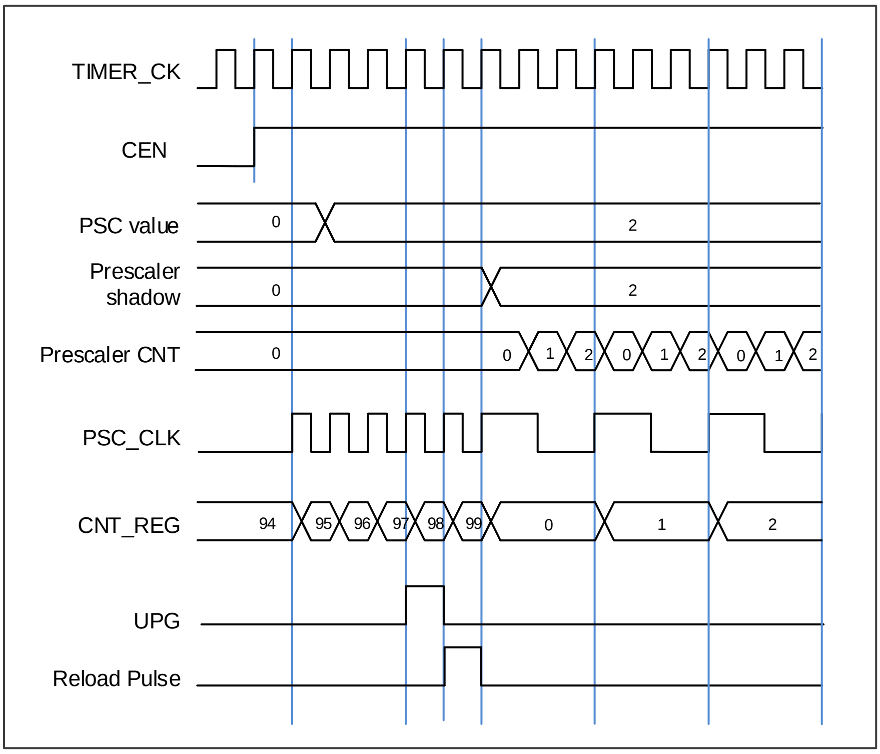
Figure 18-92. Timing chart of internal clock divided by 1



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 18-93. 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.

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 set the UPDIS bit in TIMERx_CTL0 register, the update event is disabled.

When an update event occurs, all the shadow registers (counter auto reload register, prescaler register) are updated.

Figure 18-94. Timing chart of up counting mode, PSC=0/2 and *Figure 18-95. Timing chart of up counting mode, change TIMERx_CAR ongoing* show some examples of the counter behavior for different clock prescaler factor when TIMERx_CAR=0x99.

Figure 18-94. Timing chart of up counting mode, PSC=0/2

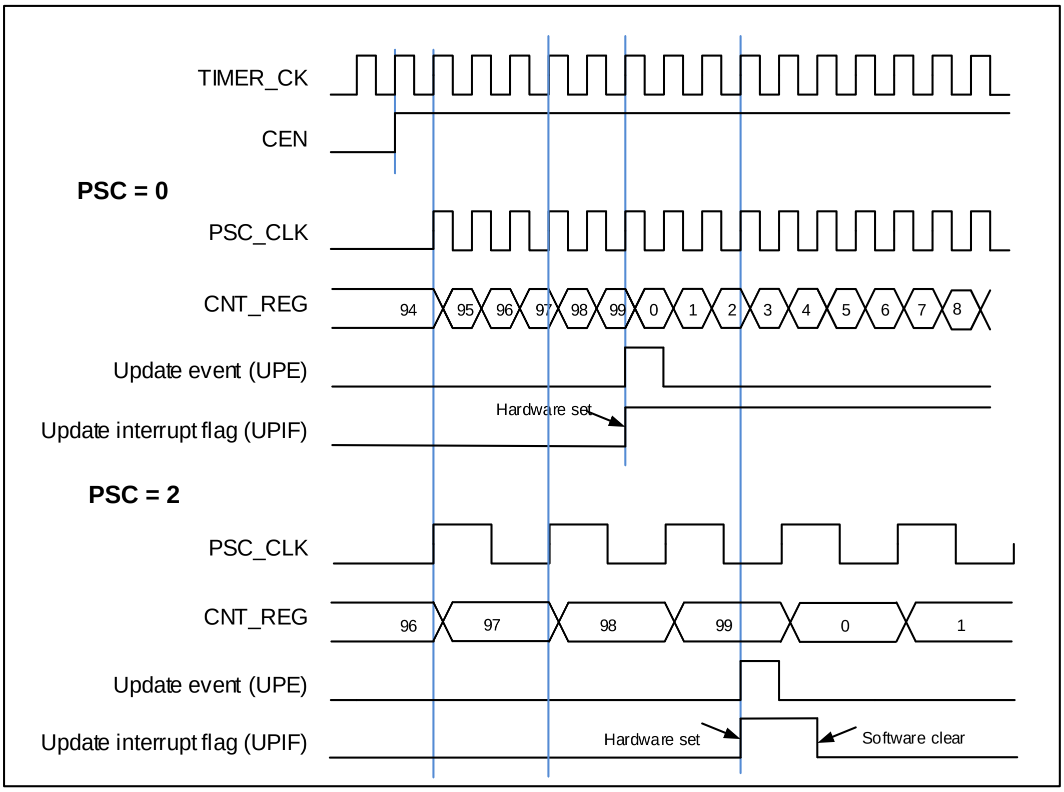
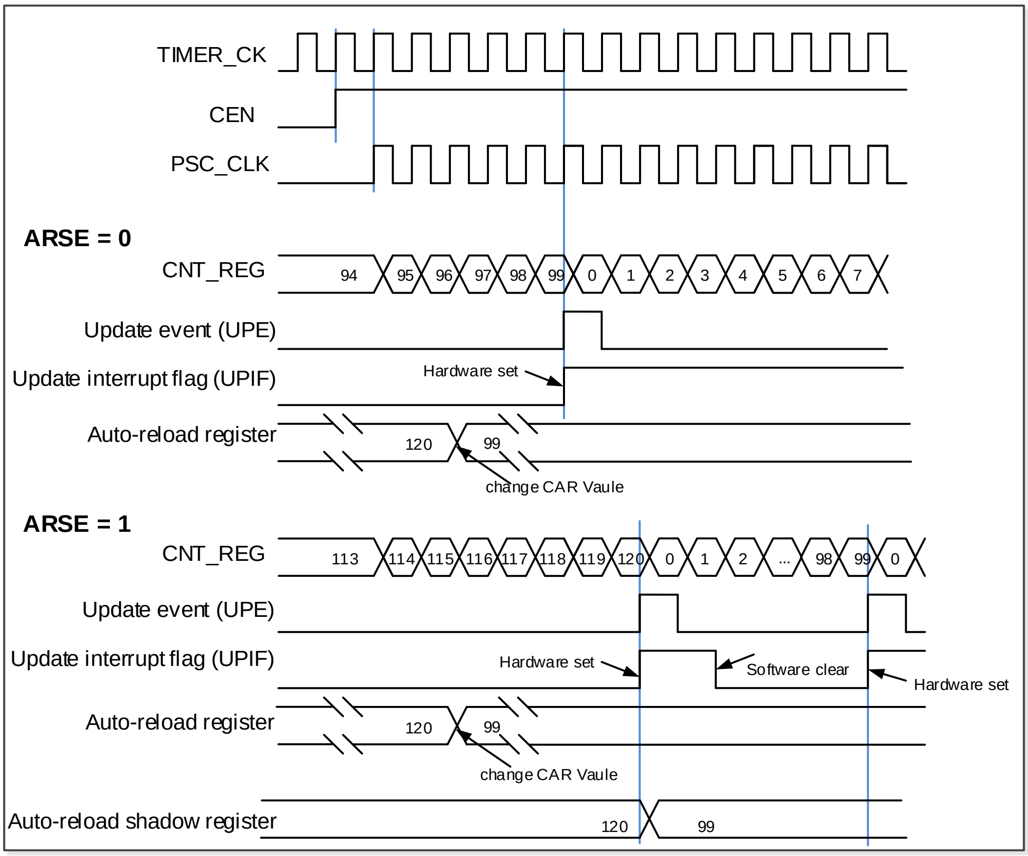


Figure 18-95. Timing chart of up counting mode, change TIMERx_CAR ongoing



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.

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, then the CEN bit keeps 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.

Timer debug mode

When the Cortex®-M23 halted, and the TIMERx_HOLD configuration bit in DBG_CTL register set to 1, the TIMERx counter stops.

18.5.5. TIMERx registers(x=5,6)

TIMER5 base address: 0x4000 1000

TIMER6 base address: 0x4000 1400

Control register 0 (TIMERx_CTL0)

Address offset: 0x00
Reset value: 0x0000

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								ARSE	Reserved			SPM	UPS	UPDIS	CEN
								rw				rw	rw	rw	rw

Bits	Fields	Descriptions
31:8	Reserved	Must be kept at reset value
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. 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: When enabled, 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 generates an update event. 1: When enabled, 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 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

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									MMC[2:0]			Reserved			
									rw						

Bits	Fields	Descriptions
31:7	Reserved	Must be kept at reset value
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. 011~111: Reserved.

3:0	Reserved	Must be kept at reset value.

Interrupt enable register (TIMERx_DMAINTEN)

Address offset: 0x0C

Reset value: 0x0000

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							UPDEN	Reserved							UPIE
							rw								rw

Bits	Fields	Descriptions
31:9	Reserved	Must be kept at reset value.
8	UPDEN	Update DMA request enable 0: disabled 1: enabled
7:1	Reserved	Must be kept at reset value.
0	UPIE	Update interrupt enable 0: disabled 1: enabled

Interrupt flag register (TIMERx_INTF)

Address offset: 0x10

Reset value: 0x0000

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															UPIF
															rc_w0

Bits	Fields	Descriptions
31:1	Reserved	Must be kept at reset value.
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

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															UPG
															w

Bits	Fields	Descriptions
31:1	Reserved	Must be kept at reset value.
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. The prescaler counter is cleared at the same time. 0: No generate an update event 1: Generate an update event

Counter register (TIMERx_CNT)

Address offset: 0x24

Reset value: 0x0000

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

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

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.

19. Low power timer (LPTIMER)

19.1. Overview

The LPTIMER is a 32-bit (GD32L233) or 16-bit (GD32L235) timer and it is able to keep running in all power modes except for standby mode with its diversity of clock sources. The LPTIMER provides a flexible mechanism of the clock, which reduces the power consumption to a minimum while also achieving the required functions and performance.

The LPTIMER can be used as a pulse counter with no internal clock source. The LPTIMER has the ability to wake up the system from the low-power modes, and it is suitable for realizing timeout mode with very low power consumption.

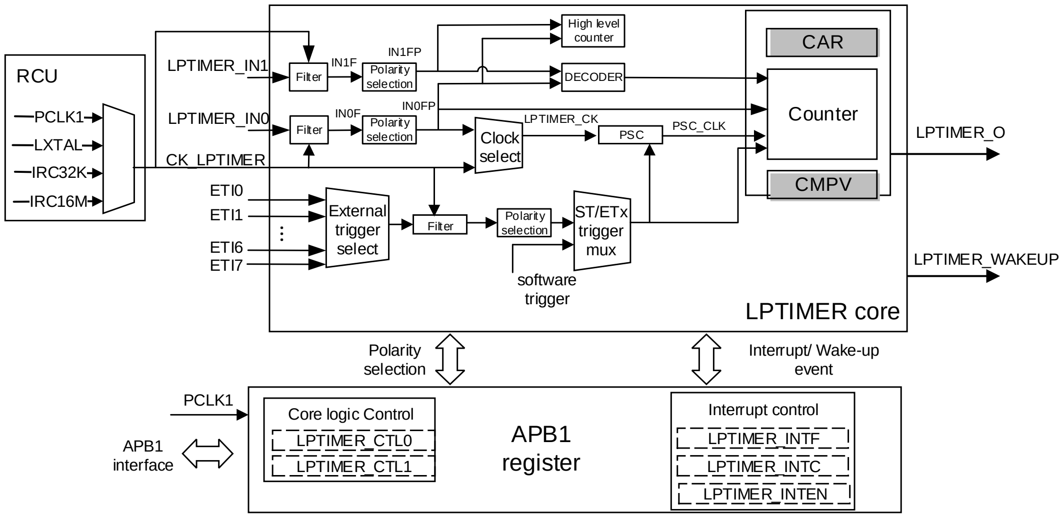
19.2. Characteristics

- Counter width: 32-bit (GD32L233) or 16-bit (GD32L235).
- Source of counter clock is selectable:
  - Internal clock: an Internal 16 MHz RC oscillator (IRC16M), an Internal 32 KHz RC oscillator (IRC32K), a 32.768 KHz Low Speed crystal oscillator (LXTAL), or an APB1 clock (PCLK1).
  - External clock: the sources through LPTIMER external input 0 (used as a pulse counter).
- Counter modes: count up.
- Operating mode : continuous counting mode or single counting mode
- Programmable prescaler: 3-bit.
- Channel output is user-configurable:
  Programmable PWM mode, single pulse mode, set mode
- Auto reload function.
- Interrupt output
- Selectable trigger: software trigger or hardware input trigger
- Decoder mode: decoder mode 0 and decoder mode 1

19.3. Block diagram

Figure 19-1. LPTIMER block diagram provides details of the internal configuration of the low power timer.

Figure 19-1. LPTIMER block diagram



19.4. Function overview

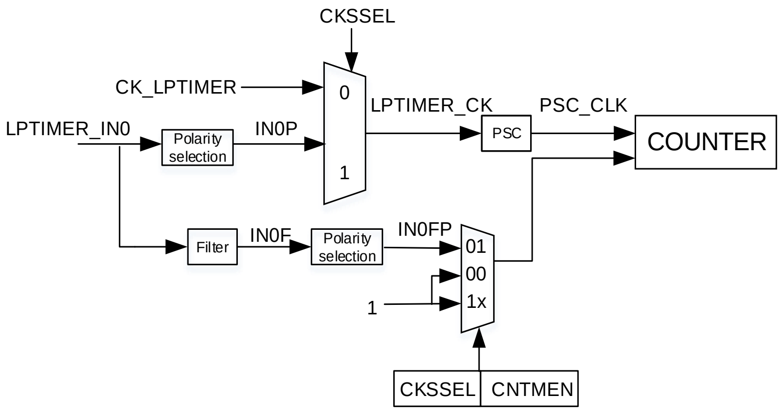
19.4.1. Clock selection

The LPTIMER can be clocked by several clock sources. It can be clocked using an internal clock signal: internal 16 MHz RC oscillator (IRC16M), internal 32 KHz RC oscillator (IRC32K), 32.768 KHz Low Speed crystal oscillator (LXTAL), APB1 clock (PCLK1) sources through the Reset and clock unit (RCU).

LPTIMER can also use an external clock signal on its external input 0 (LPTIMER_IN0) for clock control. When using an external clock source as the clock source, LPTIMER has the following two possible configurations:

- **Case 0:** When LPTIMER is clocked by an external signal, meanwhile APB1 or any other oscillator (including IRC16M, IRC32K and LXTAL) provides an internal clock signal to LPTIMER.
- **Case 1:** LPTIMER is only clocked by an external clock source on LPTIMER_IN0. When all oscillators are turned off after entering low power mode, this configuration is a configuration used to implement the timeout mode or pulse counter function.

Figure 19-2. LPTIMER clock source selection



LPTIMER has the capability of being clocked by either the internal clock signal or external clock signal controlled by bits CNTMEN and CKSSEL in LPTIMER_CTL0 register. The CKSSEL bit is used to select which clock drivies the counter prescaler and the default clock source is the PCLK1. The CNTMEN bit is used to select which clock drivies LPTIMER counter.

When LPTIMER use an external clock signal, the CKPSEL bits are used to configure the active edge used by the counter. The counter can be updated with a rising/ falling edge or both edges of an external clock signal, which depends on the value of the CKPSEL [1:0] bits.

Note that when external clock source signal is derived from the external input 0 (LPTIMER_IN0) pin, if both edges are configured to be active ones(CKPSEL=2'b10) or the pin sampled by a digital filter(ECKFLT≠2'b00), an internal clock signal should also be provided (Case 0). In this case, the internal clock signal frequency should be at least four times the frequency of the external clock signal.

The following clock modes can be selected according to configuration of the CKSSEL bit and CNTMEN bit:

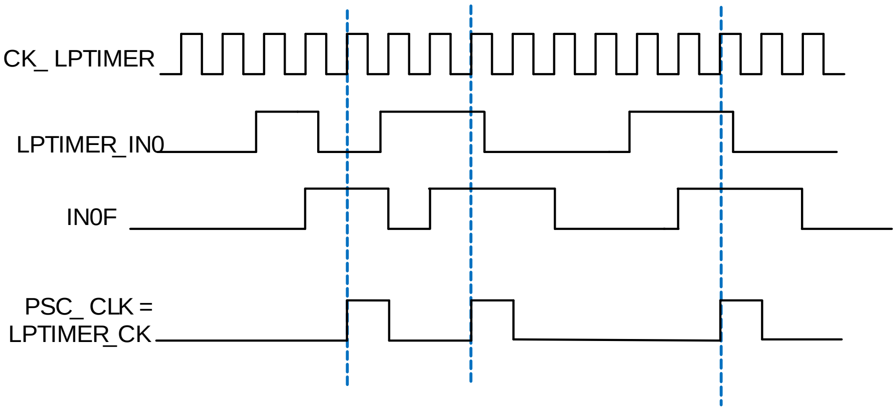
- CKSSEL = 0: the LPTIMER clock is provided by an internal clock signal
  - Internal clock mode 0 (CNTMEN = 0)
    The LPTIMER is clocked by an internal clock signal and the counter is count with every internal clock pulse.
  - Internal clock mode 1 (CNTMEN = 1)
    The external input 0 (LPTIMER_IN0) is sampled with the internal clock. Therefore, without losing any events, the change frequency of the external input signal should never exceed the frequency of the internal clock. And, the LPTIMER's internal clock of cannot be prescaled (PSC [2:0] = 000).

- CKSSEL = 1: the LPTIMER clock is provided by an external clock signal.
  In this case, the CNTMEN bit can be set or reset. The LPTIMER does not require an internal clock source (unless the input filter is enabled or both edges are configured to be active ones). The external signal on the LPTIMER_IN0 is used as LPTIMER's system clock, this is suitable for operation modes without an embedded oscillator.

For this case, the LPTIMER counter can be clocked either on rising or falling edges of the external input clock signal, but not on both edges.

Since the external signal added to the LPTIMER_IN0 pin is also used to clock the LPTIMER core logic, there is some initial delay (after the LPTIMER is enabled) before the counter is counting. Therefore, the first five active edges on theLPTIMER_IN0 are lost after the LPTIMER has enabled.

Figure 19-3. Internal clock mode1 (CKSSEL = 0 and CNTMEN = 1 and PSC[2:0] = 000)



19.4.2. LPTIMER enable

The LPTEN bit in the LPTIMER_CTL1 register is used to enable the LPTIMER core logic. After the LPTEN bit is set to 1, it is necessary to delay two LPTIMER_CK clocks before the LPTIMER is actually enabled.

The LPTIMER_CTL0 and LPTIMER_INTEN registers (except for INHLCOIE and HLCMVUPIF bit) must be modified only when the LPTIMER is disabled.

19.4.3. Prescaler

The prescaler can divide the timer clock (LPTIMER_CK) to the counter clock (PSC_CLK) by a configurable power of 2 prescaler. The prescaler is select by the PSC[2:0] bit-field in LPTIMER_CTL0 must only be modified when the LPTIMER is disabled (LPTEN bit is reset to '0'). The table below lists all the possible division ratios:

Table 19-1. Prescaler division factor

Prescaler divider	PSC[2:0] bit-filed
1/1	000
1/2	001
1/4	010
1/8	011
1/16	100
1/32	101
1/64	110

Prescaler divider	PSC[2:0] bit-filed
1/128	111

19.4.4. Input filter

The external (mapped to GPIOs) or internal (mapped on-chip peripherals, such as comparators) signals on the LPTIMER_Inx needs to be filtered by a digital filter to prevent the glitches and noise interference from spreading in LPTIMER. This can be used to prevent false counts and triggers.

Before using the digital filters, It is necessary to provide an internal clock source to the LPTIMER to ensure the proper operation of the filters.

There are two types of digital filters:

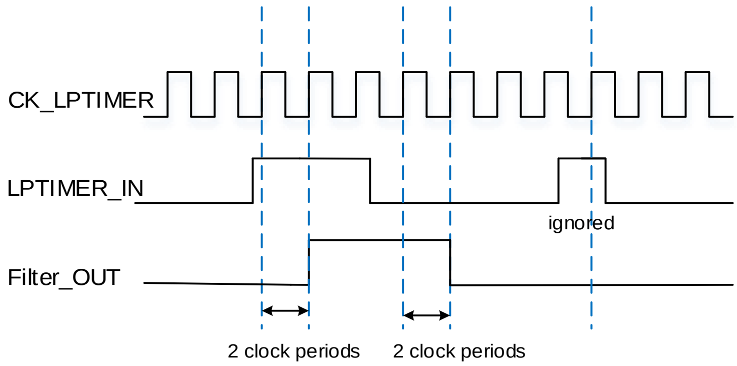
- The first type: protects the LPTIMER external inputs (LPTIMER_IN0/ LPTIMER_IN1). The digital filters is controlled by the ECKFLT[1:0] bits.
- The second type: protects the LPTIMER trigger inputs (ETIx). The digital filters is controlled by the TFLT[1:0] bits.

Note: The same type of digital filters should be keep the same configuration.

The sensitivity of the digital filters depends on the number of consecutive identical samples that should be detected on the LPTIMER inputs and treats the signal level changes as valid.

Figure 19-4. Input filter timing diagram (ECKFLT=2'b01) shows an example of 2 consecutive samples of the input filter.

Figure 19-4. Input filter timing diagram (ECKFLT=2'b01)



Note: If there is no internal clock signal, the ECKFLT and TFLT bits must be set to 0 to disable the digital filter. In this case, an external analog filter can be used to protect the LPTIMER external inputs against the disturbance.

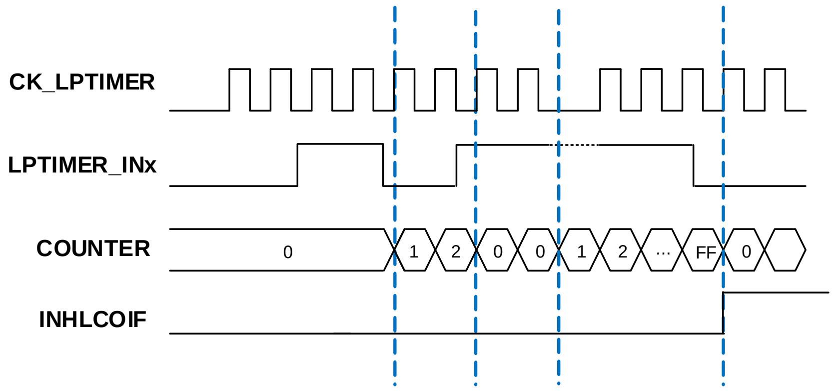
19.4.5. External inputs high level counter

The INHLCEN bit is set to 1 to enable the high level count function of external inputs (LPTIMER_Inx). In this mode, the high level counter is clocked by CK_LPTIMER (internal clock). The counter starts counting up when the hign level occurs, and once a low level occurs, the counter is cleared to 0.

The INHLCOIF flag (in LPTIMER_INTF register) is set by hardware when the value of LPTIMER_Inx high level counter equal to the value of INHLCMVAL bits (in LPTIMER_INHLCMV register). An interrupr will generated if the INHLCOIE bit is enabled (in LPTIMER_INTEN register). The INHLCOIF flag can be cleared by writing 1 to the INHLCOIC bit in the INTC register.

Figure 19-5. External inputs high level counter shows an example of the external inputs high level counter

Figure 19-5. External inputs high level counter



The APB bus and the CK_LPTIMER use different clocks, so there is a delay between the APB write and the time when these values are actually used in the LPTIMER_INHLCMV register. Within this delay time period, any additional write into this register must be avoided.

The HLCMVUPIF flag in the LPTIMER_INTF register is used to indicate when the write operation to the LPTIMER_INHLCMV register is completed.

19.4.6. Start counting mode

The LPTIMER counter may be triggered by software or by detecting a valid edge on one of the 8 trigger inputs. ETMEN [1:0] is used to configure the trigger mode of LPTIMER:

- ETMEN[1:0] = 2'b 00: The LPTIMER counter is started as soon as the CTNMST bit or SMST bit is set by software.
- ETMEN[1:0] ≠ 2'b00: ETSEL [2:0] is used to select one of the 8 trigger inputs is used to start the counter. The remaining three non-zero values of the ETMEN [1:0] bits are used to configure the valid edge used by the trigger inputs. The LPTIMER counter starts as soon as a valid edge is detected.

The external triggers can be regarded as asynchronous signals of the LPTIMER. Therefore, once a trigger is detected, for synchronization, a delay of two counter clock period is required before the timer starts running. If a new trigger event occurs after the LPTIMER starts, the trigger event will be ignored (unless timeout mode is enabled).

Note: The LPTEN bits must be enabled before the SMST/CTNMST bits are set. When the

LPTEN bit equals “0”, any write on these bits will be discarded by hardware.

19.4.7. External trigger mapping

The LPTIMER external trigger mapping is shown in ***Table 19-2. External trigger mapping***.

Table 19-2. External trigger mapping

ETSEL[2:0]	External trigger mapping
ETI0	GPIO
ETI1	RTC Alarm 0
ETI2	RTC Alarm 1
ETI3	RTC_TAMP0
ETI4	RTC_TAMP1
ETI5	RTC_TAMP2
ETI6	CMP0_OUT
ETI7	CMP1_OUT

19.4.8. Counter operating mode

The LPTIMER counter works in two operating modes:

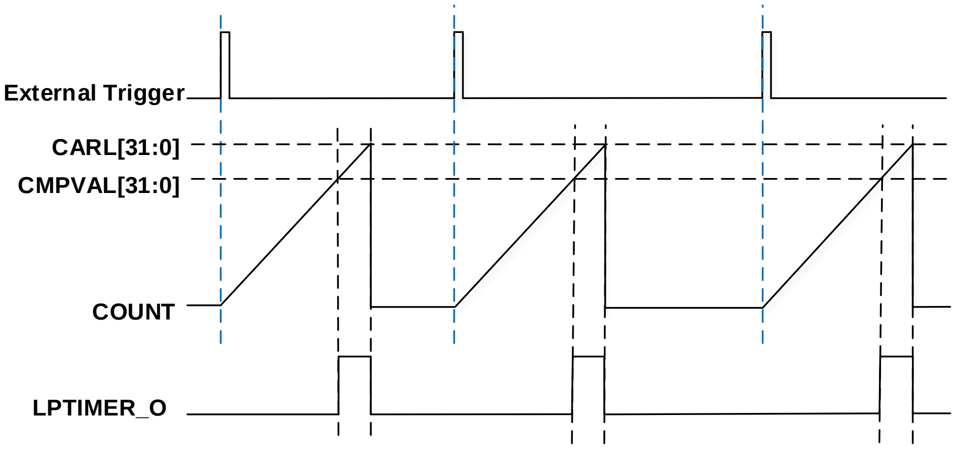
- Continuous counting mode: the LPTIMER counter is running continuously, the counter is started from a trigger event (software trigger or external trigger) and will not stop until the timer is disabled.
- Single counting mode: the LPTIMER counter is started from a trigger event (software trigger or external trigger) and stops after counting the value of CARL bits in LPTIMER_CAR register.

Single counting mode

The SMST bit is set to 1 to enable the the single counting mode. In this mode, a new trigger event will restart the LPTIMER counter. Any trigger event that occurs after the counter is started and before the counter reaches the value of CARL bits will be ignored.

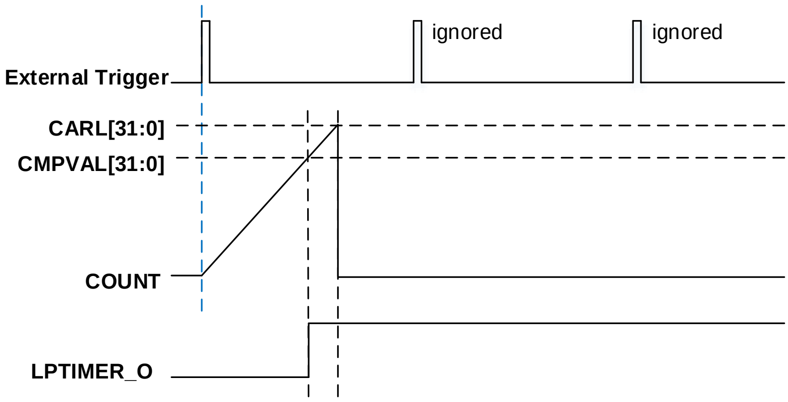
If an external trigger is selected to start LPTIMER counter, when the SMST bit is set, each external trigger event that arrives after the counter stops counting (the value of the CNT bits is zero), will start the counter in a new single counting cycle as shown in ***Figure 19-6. LPTIMER output with SMST = 1(32-bit)***.

Figure 19-6. LPTIMER output with SMST = 1(32-bit)



When the OMSEL bit in the LPTIMER_CTL0 register is set, the set mode is enable. In this case, the counter is only started once after the first trigger, and all subsequent trigger events is ignored, as shown in ***Figure 19-7. LPTIMER output with OMSEL = 1(32-bit)***.

Figure 19-7. LPTIMER output with OMSEL = 1(32-bit)



If ETMEN [1:0] = 2'b 00, the software trigger is enabled, setting the SMST bit will start the counter in single counting mode.

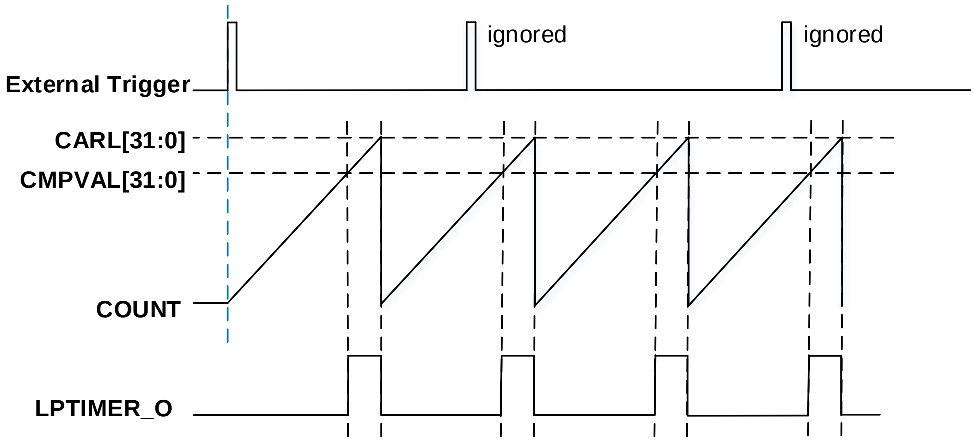
Continuous counting mode

The CTNMST bit is set to 1 to enable the the continuous counting mode.

If an external trigger is selected to start LPTIMER counter, an external trigger event arriving after the CTNMST bit is set will start the counter in continuous counting mode. Any trigger event that occurs after the counter is started will be ignored as shown in ***Figure 19-8. LPTIMER output with CTNMST = 1(32-bit)***.

If ETMEN [1:0] = 2'b 00, the software trigger is enabled, setting the CTNMST bit will start the counter in continuous counting mode.

Figure 19-8. LPTIMER output with CTNMST = 1(32-bit)



The SMST and CTNMST bits can be modified only when the timer is enabled (the LPTEN bit is set). And the single counting mode and continuous counting mode can be modified on the fly.

If the LPTIMER previously working in the continuous counting mode, setting SMST will switch the LPTIMER to the single counting mode. The counter will stop counting after counting the value of CARL bit-filed.

If the LPTIMER previously working in the single counting mode, setting CTNMST will switch the LPTIMER to the continuous counting mode. The counter will restart after counting to the value of CARL bit-filed.

19.4.9. Output Mode

By configuring the LPTIMER_CARL register and LPTIMER_CMPV register, the LPTIMER can output several different waveforms.

The LPTIMER can generate the following waveforms:

- PWM mode: the LPTIMER output is set when a match occurs between the value of LPTIMER_CMPV and the LPTIMER_CNT registers. The LPTIMER output is reset when a match occurs between the value of LPTIMER_CAR and the LPTIMER_CNT registers.
- Single pulse mode: the output waveform is same as the first pulse in PWM mode, and then the output will always be reset.
- Set mode: the output waveform is similar to the single pulse mode, except that the output remains at the last signal level (depends on the value of the OPSEL bit in the LPTIMER_CTL0 register).

There modes require that the value of LPTIMER_CAR register is greater than the value of the LPTIMER_CMPV register.

The OMSEL bit in the LPTIMER_CTL0 register is used to controls the output mode.

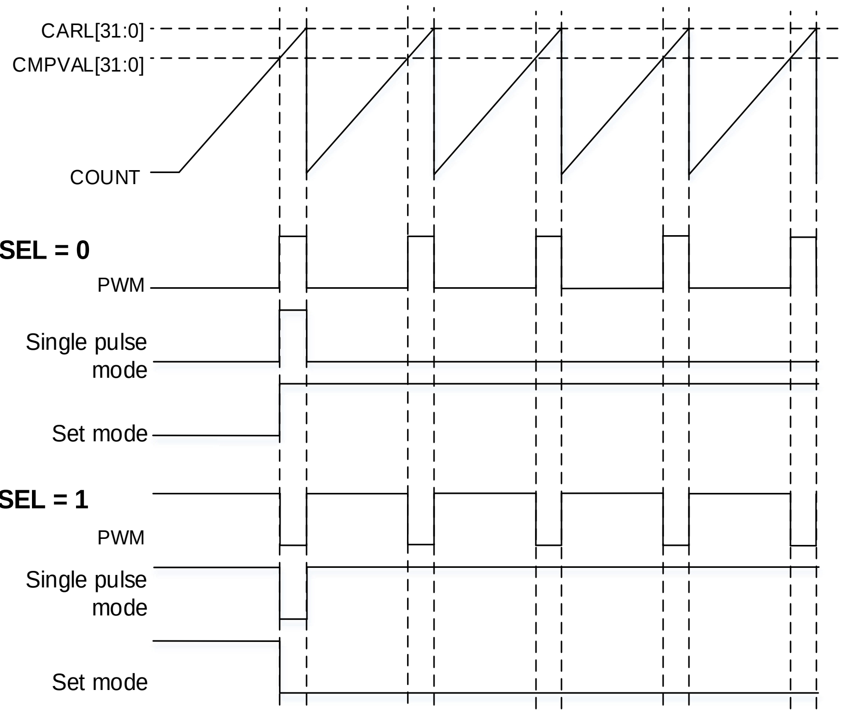
- OMSEL = 0: the LPTIMER to generate either a PWM mode waveform or a single pulse mode waveform (depending on the CTNMST bit or SMST bit is set).

- OMSEL = 1: the LPTIMER to generate a set mode waveform.

The OPSEL bit is used to configure the LPTIMER output polarity, the modification of this bit will take effect immediately. Therefore, any modification to the polarity configuration bit before enabling the LPTIMER will immediately change the output default value.

The maximum frequency of the generated signal is the LPTIMER clock frequency divided by 2. *Figure 19-9. LPTIMER_O output mode with OPSEL bit(32-bit)* below shows three waveforms output by the LPTIMER. Also, it shows the effect of the polarity change with the OPSEL bit.

Figure 19-9. LPTIMER_O output mode with OPSEL bit(32-bit)



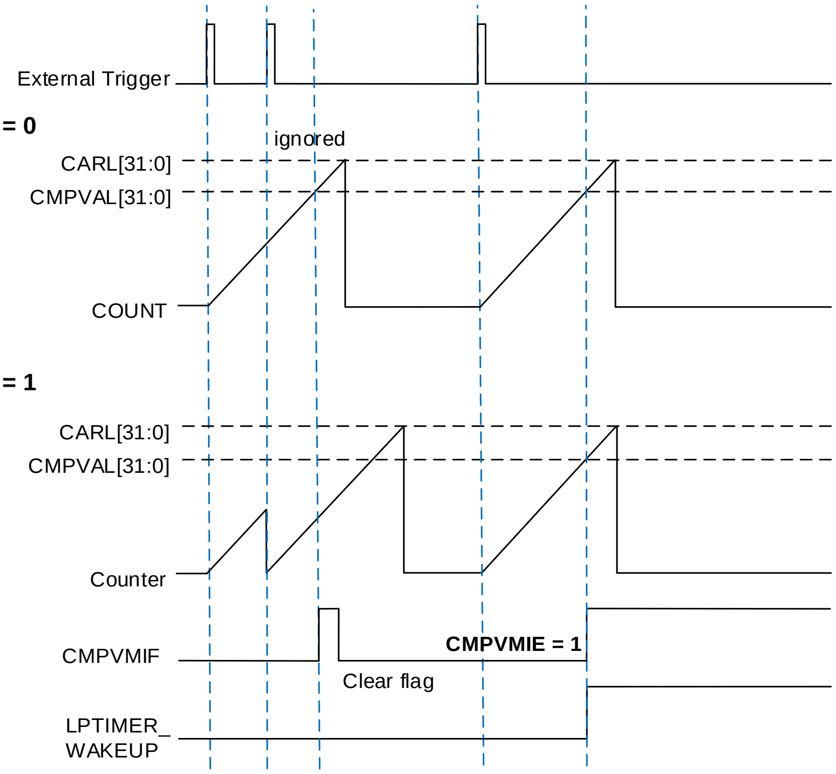
19.4.10. Timeout mode

By setting the TIMEOUT bit, a valied edge detected on a selected trigger input can be used to reset the LPTIMER counter. The first trigger event will start the timer, and subsequent trigger events will reset and restart the counter.

The LPTIMER can realize the low power consumption timeout mode, and the timeout value can be determined by the value of the LPTIMER_CMPV register.

If no trigger occurs within the comparison value range, when the counter counts to the comparison value, a comparison match interrupt will be generated to wake up the MCU.

Figure 19-10. LPTIMER timeout mode(32-bit)



19.4.11. Decoder mode

The LPTIMER has two decoder modes:

- Decoder mode 0: the inputs of LPTIMER_IN0 and LPTIMER_IN1 are quadrature signals, this mode is enabled when DECMEN = 1 and DECMSEL = 0.
- Decoder mode 1: the inputs of LPTIMER_IN0 and LPTIMER_IN1 are non-quadrature signals, this mode is enabled when DECMEN = 1 and DECMSEL = 1.

Decoder mode 0

The decoder mode 0 function uses two quadrature inputs derived from the LPTIMER_IN0 and LPTIMER_IN1 pins respectively to interact with each other to generate the counter value.

At first, the CTNMST bit is set to 1 to enable the the continuous counting mode and the DECMEN bit is set to 1 to enable the decoder mode. Then setting DECMSEL = 0 to select that the decoder mode 0, and setting the CKPSEL [1:0] = 2b’00, 2b’01, or 2b’10 to select that the timer counting is determined only by the rising edge, only by falling edge, or both edges.

The counting direction is modified by hardware automatically during the voltage level change of IN0F (LPTIMER_IN0 after input filter) and IN1F (LPTIMER_IN1 after input filter) signals. The mechanism of changing the counter direction is shown in ***Table 19-3. Counting direction versus decoder signals***. The 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 LPTIMER_CAR register before the counter starts to count.

When the counter direction changes, the corresponding flag is set. When the counter direction moves from up to down, the DOWNIF bit is set. When the counter direction moves from down to up, the UPIF bit is set. The corresponding interrupt is generated if enabled by DOWNIE = 1 or UPIE = 1 in LPTIMER_INTEN register.

Table 19-3. Counting direction versus decoder signals

Counting Mode (CKPSEL [1:0])	Level	IN0F		IN1F	
		Rising	Falling	Rising	Falling
Decoder rising-edge-mode	IN0F = High	x	x	Up	-
	IN0F = Low	x	x	Down	-
	IN1F = High	Down	-	x	x
	IN1F = Low	Up	-	x	x
Decoder falling-edge-mode	IN0F = High	x	x	-	Down
	IN0F= Low	x	x	-	Up
	IN1F = High	-	Up	x	x
	IN1F = Low	-	Down	x	x
Decoder both-edge-mode	IN0F = High	x	x	Up	Down
	IN0F = Low	x	x	Down	Up
	IN1F = High	Down	Up	x	x
	IN1F = Low	Up	Down	x	x

Note:"-" means "no counting"; "x" means impossible.

Figure 19-11. Counter operation in decoder mode 0 with rising-edge-mode and ***Figure 19-12. Counter operation in decoder mode 0 with falling-edge-mode*** show the counting situations with rising-edge and falling-edge modes.

Figure 19-11. Counter operation in decoder mode 0 with rising-edge-mode

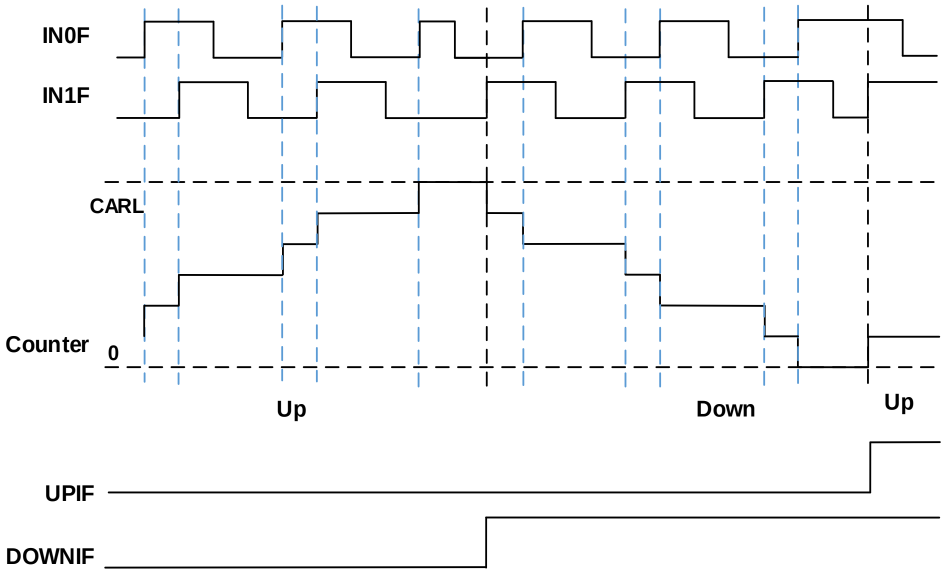
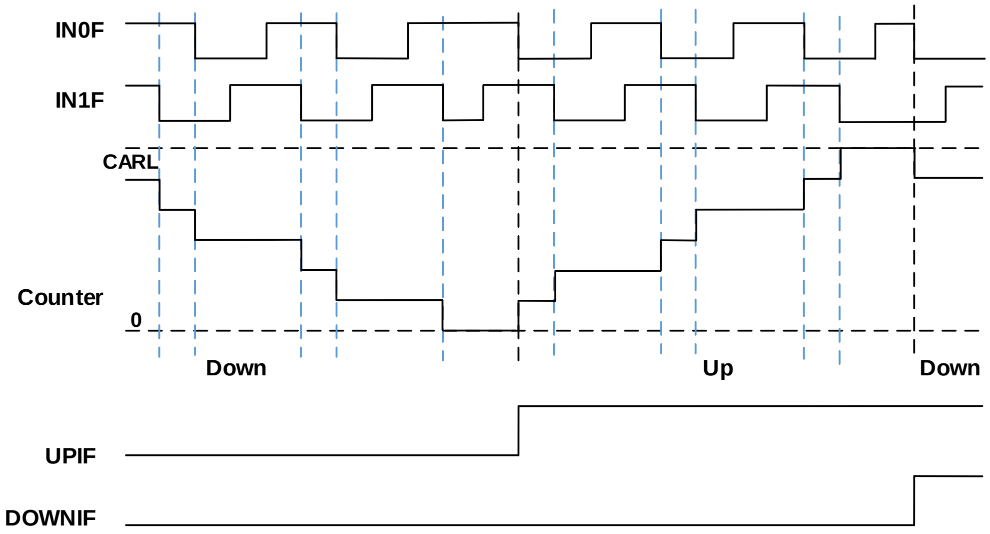


Figure 19-12. Counter operation in decoder mode 0 with falling-edge-mode



Decoder mode 1

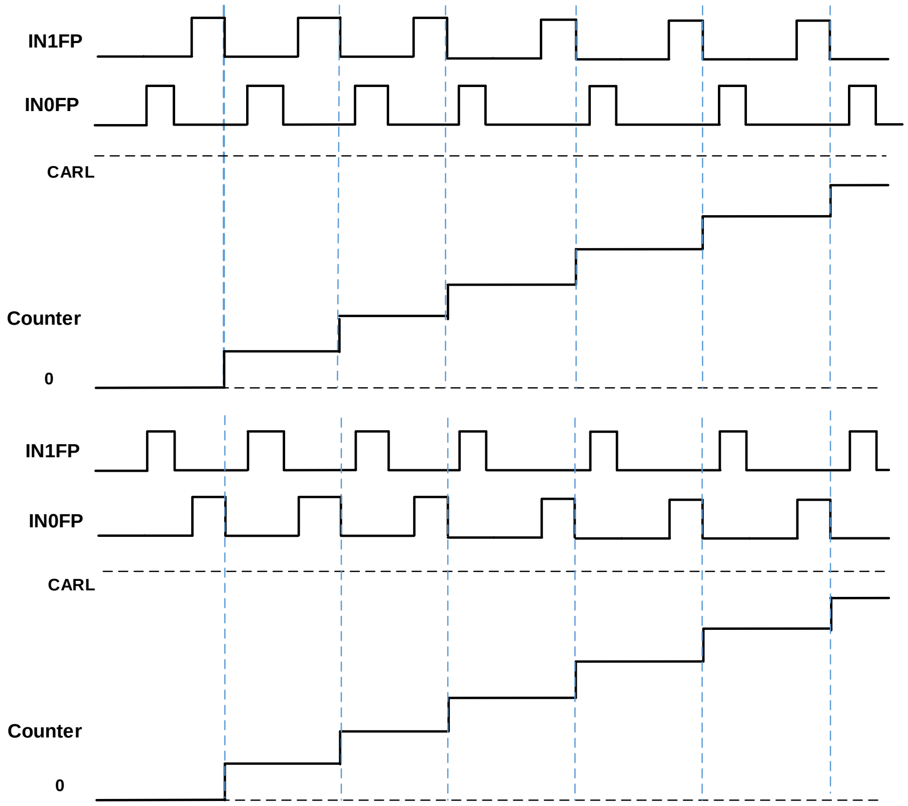
The decoder mode 1 function uses two non-quadrature inputs derived from the LPTIMER_IN0 and LPTIMER_IN1 pins respectively to generate the counter value.

At first, the CTNMST bit is set to 1 to enable the the continuous counting mode and the DECMEN bit is set to 1 to enable the decoder mode. Then setting DECMSEL = 1 to select that the decoder mode 1, and setting the CKPSEL [1:0] = 2b'00, 2b'01 to select that the inputs of LPTIMER_IN0 and LPTIMER_IN1 are non-inverted or inverted.

When two non-overlap pulses appear in IN0FP and IN1FP in sequence, the counter will increment once. ***Figure 19-13. Counter operation in decoder mode 1 with non-inverted*** shows two waveform timing diagrams for the counter can count correctly in decoder mode 1. The high level of IN0FP and IN1FP are not overlap.

The decoder can be regarded as an external clock. This means that the counter counts continuously from 0 to the counter-reload value. Therefore, users must configure the LPTIMER_CAR register before the counter starts to count.

Figure 19-13. Counter operation in decoder mode 1 with non-inverted



When the inputs of LPTIMER_IN0 and LPTIMER_IN1 do not meet the timing relationship in *Figure 19-13. Counter operation in decoder mode 1 with non-inverted*, the counter cannot count. Depending on the input wavefroms the corresponding interrupt flags will be set (IN1EIF，IN0EIF，INRFOEIF，INHLOEIF), and the interrupts will generated if enabled by IN1EIE, IN0EIE ,INRFOEIE or INHLOEIE in LPTIMER_INTEN register.

Figure 19-14. Counter operation in decoder mode 1 with non-inverted(IN1EIF)

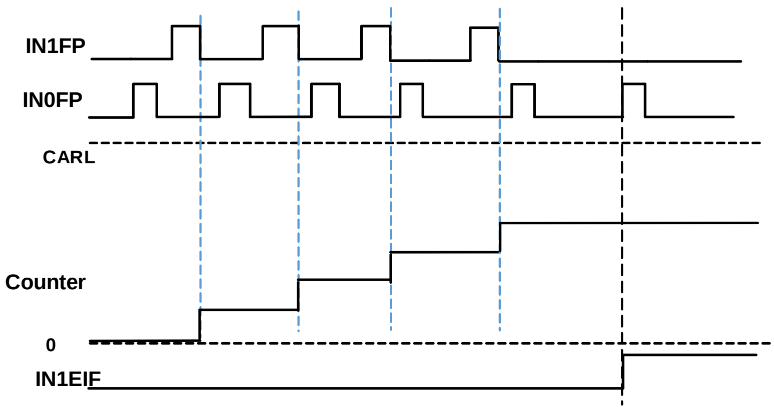


Figure 19-15. Counter operation in decoder mode 1 with non-inverted(IN0EIF)

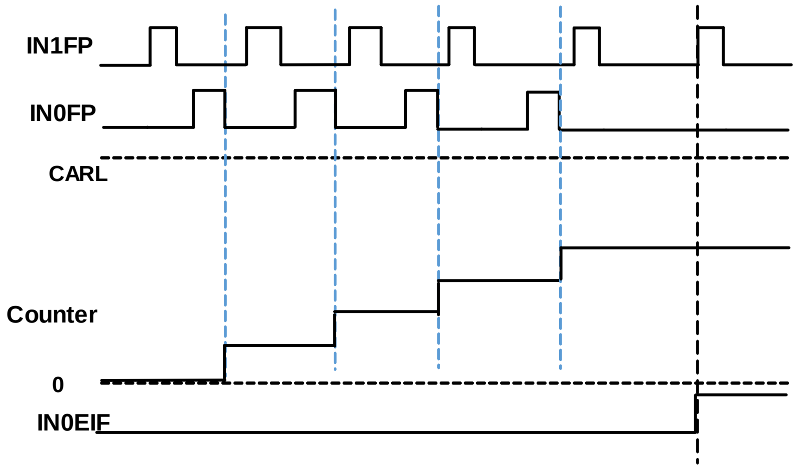


Figure 19-16. Counter operation in decoder mode 1 with non-inverted(INRFOEIF)

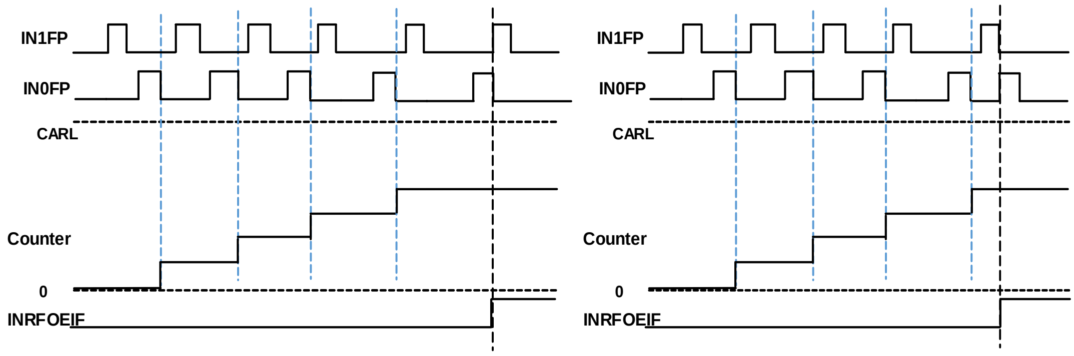
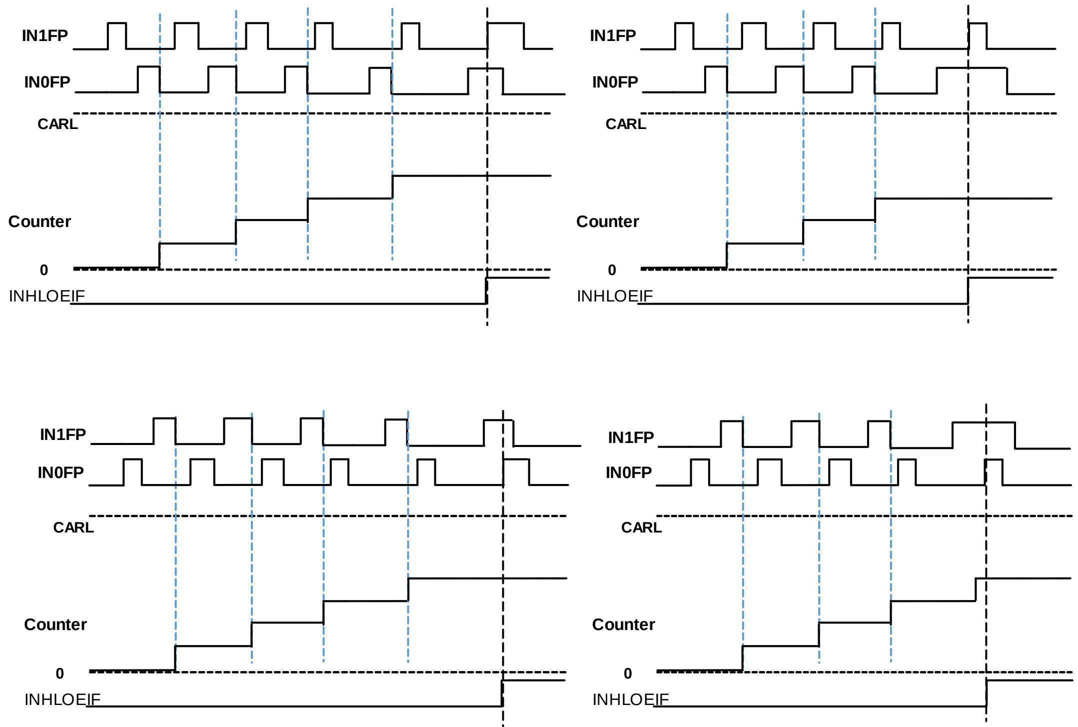


Figure 19-17. Counter operation in decoder mode 1 with non-inverted(INHLOEIF)



Note that when the LPTIMER used in decoder modes, an internal clock signal should also be provided (CKSSEL = 0) and the internal clock of LPTIMER cannot be prescaled (PSC [2:0] = 000). In this case, the internal clock signal frequency should be at least four times the frequency of the external clock signal.

19.4.12. Register update operation

The LPTIMER_CAR register and LPTIMER_CMPV register are updated immediately after the APB bus completes the write operation, or updated at the end of the current period, when the LPTIMER has already started. The SHWEN bit is used to configure the update of the LPTIMER_CAR and the LPTIMER_CMPV registers:

- SHWEN = 0: after any write access, the LPTIMER_CAR and the LPTIMER_CMPV registers are updated immediately.
- SHWEN = 1: after the LPTIMER is started,the LPTIMER_CAR and the LPTIMER_CMPV registers are updated at the end of the current period.

The APB bus and the LPTIMER core use different clocks, so there is some delay between the APB write and when the LPTIMER_CAR register and LPTIMER_CMPV register actually use these values. During this delay time, any additional writes to these registers must be avoided.

The CARUPIF flag and the CMPVUPIF flag in the LPTIMER_INTF register are respectively used to indicate when the write operation to the LPTIMER_CAR register and the LPTIMER_CMPV register is completed.

After the LPTIMER_CAR register or the LPTIMER_CMPV register is written, only the previous write operation is completed, a new write operation to the same register can be performed.

Any continuous write operations performed before the CMPVUPIF flag or the CARUPIF flag are set respectively will cause unpredictable results.

19.4.13. Low-power modes

The LPTIMER is able to keep running in all power modes except for Standby mode with its diversity of clock sources. The LPTIMER has the ability to wake up the system from the low-power modes, and it is suitable for realizing timeout with very low power consumption.

Table 19-4. LPTIMER works in low-power modes

Mode	Description
Sleep mode	Operating normally. LPTIMER interrupts cause the device to exit Sleep mode.
Run2 mode	Operating normally.
Sleep2 mode	Operating normally. LPTIMER interrupts cause the device to exit the LPSleep mode.
Deep-sleep0 /1 mode	When LPTIMER is clocked by LXTAL or Internal low speed RC oscillator, LPTIMER interrupts cause the device to exit Deep-sleep

Mode	Description
	0 /1 mode.
Deep-sleep2 mode	LPTIMER interrupts cause the device to exit the Deep-sleep 2 mode.

19.4.14. Interrupts

The following events can generate interrupts or wake-up events, if they are enabled through the LPTIMER_INTEN register:

- LPTIMER_IN1 error
- LPTIMER_IN0 error
- The falling and rising edges of LPTIMER_IN0 and LPTIMER_IN1 overlap error
- The high level of LPTIMER_IN0 and LPTIMER_IN1 overlap error
- LPTIMER_Inx(x=0,1) high level counter overflow
- Input high level counter max value register update interrupt
- LPTIMER counter direction change up to down
- LPTIMER counter direction change down to up
- Counter auto reload register update
- Compare value register update
- External trigger edge event
- Counter auto reload register match
- Compare value register match

If the interrupt flag in the LPTIMER_INTF register is set before its corresponding interrupt enable bit in the LPTIMER_INTEN register is set, the interrupt is invalid.

Table 19-5. LPTIMER interrupt events

Interrupt event	Description
LPTIMER_IN1 error	Interrupt flag is set when the signal of LPTIMER_IN1 does not jump between the two consecutive rising edges of LPTIMER_IN0 (just used in decoder mode 1).
LPTIMER_IN0 error	Interrupt flag is set when the signal of LPTIMER_IN0 does not jump between the two consecutive rising edges of LPTIMER_IN1 (just used in decoder mode 1).
The falling and rising edges of LPTIMER_IN0 and LPTIMER_IN1 overlap	Interrupt flag is set when the falling edge of LPTIMER_IN0 and the rising edge of LPTIMER_IN1 occur simultaneously or the falling edge of LPTIMER_IN1 and the rising edge of LPTIMER_IN0 occur simultaneously (just used in decoder mode 1).
The high level of LPTIMER_IN0 and LPTIMER_IN1 overlap	Interrupt flag is set when the high level of LPTIMER_IN0 and LPTIMER_IN1 overlap (just used in decoder mode 1).
LPTIMER_Inx(x=0,1) high level counter overflow	Interrupt flag is set when LPTIMER_Inx high level counter equal to external input high level counter max value register (LPTIMER_INHLCMV).

Interrupt event	Description
Input high level counter max value register update	Interrupt flag is set when the APB bus write operation to the LPTIMER_INHLCMV register has been successfully completed.
LPTIMER counter direction change up to down	Interrupt flag is set when the counter direction moves from up to down.
LPTIMER counter direction change down to up	Interrupt flag is set when the counter direction moves from down to up.
Counter auto reload register update	Interrupt flag is set when the APB bus write operation to the LPTIMER_CAR register has been successfully completed.
Compare value register update	Interrupt flag is set when the APB bus write operation to the LPTIMER_CMPV register has been successfully completed.
External trigger edge event	Interrupt flag is set when an external trigger active edge event is detected.
Counter auto reload register match	Interrupt flag is set when the value of the Counter register (LPTIMER_CNT) matches the value of the Counter auto-reload register (LPTIMER_CAR).
Compare value register match	Interrupt flag is set when the value of the Counter register (LPTIMER_CNT) matches the value of the compare value register (LPTIMER_CMPV).

19.4.15. LPTIMER debug mode

When the Cortex®-M23 halted, and the corresponding LPTIMER_HOLD configuration bit in DBG_CTL1 register is set to 1, the LPTIMER counter stops.

19.5. LPTIMER registers

For GD32L233:

LPTIMER base address: 0x4000 9400

For GD32L235:

LPTIMER0 base address: 0x4000 9400

LPTIMER1 base address: 0x4000 7C00

19.5.1. Interrupt flag register (LPTIMER_INTF)

Address offset: 0x00

Reset value: 0x0000

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
IN1EIF	IN0EIF	INRFOEIF	INHLOEIF	INHLCOIF	HLCMVUPIF	Reserved									
r	r	r	r	r	r										

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved									DOWNIF	UPIF	CARUPIF	CMPVUPIF	ETEDEVIF	CARMIF	CMPVMIF
									r	r	r	r	r	r	r

Bits	Fields	Descriptions
31	IN1EIF	LPTIMER_IN1 error interrupt flag This flag is set by hardware when the signal of LPTIMER_IN1 does not jump between the two consecutive rising edges of LPTIMER_IN0. IN1EIF flag can be cleared by writing 1 to the IN1EIC bit in the INTC register. **Note:** This flag just used in decoder mode 1.
30	IN0EIF	LPTIMER_IN0 error interrupt flag This flag is set by hardware when the signal of LPTIMER_IN0 does not jump between the two consecutive rising edges of LPTIMER_IN1. IN0EIF flag can be cleared by writing 1 to the IN0EIC bit in the INTC register. **Note:** This flag just used in decoder mode 1.
29	INRFOEIF	The falling and rising edges of LPTIMER_IN0 and LPTIMER_IN1 overlap error interrupt flag. This flag is set by hardware when the falling edge of LPTIMER_IN0 and the rising edge of LPTIMER_IN1 occur simultaneously or the falling edge of LPTIMER_IN1 and the rising edge of LPTIMER_IN0 occur simultaneously. INRFOEIF flag can be cleared by writing 1 to the INRFOEIC bit in the INTC register.

		Note: This flag just used in decoder mode 1.
28	INHLOEIF	The high level of LPTIMER_IN0 and LPTIMER_IN1 overlap error interrupt flag. This flag is set by hardware when the high level of LPTIMER_IN0 and LPTIMER_IN1 overlap. INHLOEIF flag can be cleared by writing 1 to the INHLOEIC bit in the INTC register. **Note:** This flag just used in decoder mode 1.
27	INHLCOIF	LPTIMER_Inx(x=0,1) high level counter overflow interrupt flag This flag is set by hardware when LPTIMER_Inx high level counter equal to external input high level counter max value register (LPTIMER_INHLCMV). INHLCOIF flag can be cleared by writing 1 to the INHLCOIC bit in the INTC register.
26	HLCMVUPIF	Input high level counter max value register update interrupt flag This flag is set by hardware when the APB bus write operation to the LPTIMER_INHLCMV register has been successfully completed. HLCMVUPIF flag can be cleared by writing 1 to the HLCMVUPIC bit in the INTC register.
25:7	Reserved	Must be kept at reset value.
6	DOWNIF	LPTIMER counter direction change up to down interrupt flag In decoder mode 0, the DOWNIF bit is set by hardware when the counter direction moves from up to down. The DOWNIF flag can be cleared by writing 1 to the DOWNIC bit of the INTC register.
5	UPIF	LPTIMER counter direction change down to up interrupt flag In decoder mode 0, the UPIF bit is set by hardware when the counter direction moves from down to up. The UPIF flag can be cleared by writing 1 to the UPIC bit in the INTC register.
4	CARUPIF	Counter auto reload register update interrupt flag This flag is set by hardware when the APB bus write operation to the LPTIMER_CAR register has been successfully completed. The CARUPIF flag can be cleared by writing 1 to the CARUPIC bit in the INTC register.
3	CMPVUPIF	Compare value register update interrupt flag This flag is set by hardware when the APB bus write operation to the LPTIMER_CMPV register has been successfully completed. The CMPVUPIF flag can be cleared by writing 1 to the CMPVUPIC bit in the INTC register.
2	ETEDEVIF	External trigger edge event interrupt flag This flag is set by hardware when the active edge of the external trigger occurs. The ETEDEVIF flag can be cleared by writing 1 to the ETEDEVIC bit in the INTC register. **Note:** This flag will not be set when the active edge of the external trigger happened after LPTIMER started.
1	CARMIF	Counter auto reload register match interrupt flag This flag is set by hardware when the LPTIMER_CNT value matches the value of the LPTIMER_CAR register. The CARMIF flag can be cleared by writing 1 to the

		CARMIC bit in the INTC register.
0	CMPVMIF	Compare value register match interrupt flag This flag is set by hardware when the LPTIMER_CNT value matches the value of the LPTIMER_CMPV register. The CMPVMIF flag can be cleared by writing 1 to the CMPVMIC bit in the INTC register.

19.5.2. Interrupt flag clear register (LPTIMER_INTC)

Address offset: 0x04

Reset value: 0x0000

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
IN1EIC	IN0EIC	INRFOEIC	INHLOEIC	INHLCOIC	HLCMVUPIC	Reserved									
w	w	w	w	w	w										

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved									DOWNIC	UPIC	CARUPIC	CMPVUPIC	ETEDEVIC	CARMIC	CMPVMIC
									w	w	w	w	w	w	w

Bits	Fields	Descriptions
31	IN1EIC	LPTIMER_IN1 error interrupt flag clear bit. Write 1 to this bit to clear the IN1EIF flag, and write 0 has no effect.
30	IN0EIC	LPTIMER_IN0 error interrupt flag clear bit. Write 1 to this bit to clear the IN0EIF flag, and write 0 has no effect.
29	INRFOEIC	The falling and rising edges of LPTIMER_IN0 and LPTIMER_IN1 overlap error interrupt flag clear bit. Write 1 to this bit to clear the INRFOEIF flag, and write 0 has no effect.
28	INHLOEIC	The high level of LPTIMER_IN0 and LPTIMER_IN1 overlap error interrupt flag clear bit. Write 1 to this bit to clear the INHLOEIF flag, and write 0 has no effect.
27	INHLCOIC	LPTIMER_Inx(x=0, 1) high level counter overflow interrupt flag clear bit. Write 1 to this bit to clear the INHLCOIF flag, and write 0 has no effect.
26	HLCMVUPIC	Input high level counter max value register update interrupt flag clear bit. Write 1 to this bit to clear the HLCMVUPIF flag, and write 0 has no effect.
25:7	Reserved	Must be kept at reset value.
6	DOWNIC	LPTIMER counter direction change up to down interrupt flag clear bit. Write 1 to this bit to clear the DOWNIF flag, and write 0 has no effect.

5	UPIC	LPTIMER counter direction change down to up interrupt flag clear bit. Write 1 to this bit to clear the UPIF flag, and write 0 has no effect.
4	CARUPIC	Counter auto reload register update interrupt flag clear bit. Write 1 to this bit to clear the CARUPIF flag, and write 0 has no effect.
3	CMPVUPIC	Compare value register update interrupt flag clear bit. Write 1 to this bit to clear the CMPVUPIF flag, and write 0 has no effect.
2	ETEDEVIC	External trigger edge event interrupt flag clear bit. Write 1 to this bit to clear the ETEDEVIF flag, and write 0 has no effect.
1	CARMIC	Counter auto reload register match interrupt flag clear bit. Write 1 to this bit to clear the CARMIF flag, and write 0 has no effect.
0	CMPVMIC	Compare value register match interrupt flag clear bit. Write 1 to this bit to clear the CMPVMIF flag, and write 0 has no effect.

19.5.3. Interrupt enable register (LPTIMER_INTEN)

Address offset: 0x08

Reset value: 0x0000

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
IN1EIE	IN0EIE	INRFOEIE	INHLOEIE	INHLCOIE	HLCMVUPIE	Reserved									
rw	rw	rw	rw	rw	rw										

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved									DOWNIE	UPIE	CARUPIE	CMPVUPIE	ETEDEVIE	CARMIE	CMPVMIE
									rw	rw	rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31	IN1EIE	LPTIMER_IN1 error interrupt enable bit 0: disabled 1: enabled This bit can be modified only when the LPTIMER is disabled (The LPTEN bit in LPTIMER_CTL1 register is 0).
30	IN0EIE	LPTIMER_IN0 error interrupt enable bit 0: disabled 1: enabled This bit can be modified only when the LPTIMER is disabled (The LPTEN bit in LPTIMER_CTL1 register is 0).
29	INRFOEIE	The falling and rising edges of LPTIMER_IN0 and LPTIMER_IN1 overlap error

		interrupt enable bit 0: disabled 1: enabled This bit can be modified only when the LPTIMER is disabled (The LPTEN bit in LPTIMER_CTL1 register is 0).
28	INHLOEIE	The high level of LPTIMER_IN0 and LPTIMER_IN1 overlap error interrupt enable bit. 0: disabled 1: enabled This bit can be modified only when the LPTIMER is disabled (The LPTEN bit in LPTIMER_CTL1 register is 0).
27	INHLCOIE	LPTIMER_Inx(x=0,1) high level counter overflow interrupt enable bit 0: disabled 1: enabled This bit can be modified only when the LPTIMER is external input high level counter disabled (The INHLCEN bit in LPTIMER_CTL1 register is 0).
26	HLCMVUPIE	Input high level counter max value register update interrupt enable bit 0: disabled 1: enabled This bit can be modified only when the LPTIMER is external input high level counter disabled (The INHLCEN bit in LPTIMER_CTL1 register is 0).
25:7	Reserved	Must be kept at reset value.
6	DOWNIE	LPTIMER counter direction change up to down interrupt enable bit 0: disabled 1: enabled This bit can be modified only when the LPTIMER is disabled (The LPTEN bit in LPTIMER_CTL1 register is 0).
5	UPIE	LPTIMER counter direction change down to up interrupt enable bit 0: disabled 1: enabled This bit can be modified only when the LPTIMER is disabled (The LPTEN bit in LPTIMER_CTL1 register is 0).
4	CARUPIE	Counter auto reload register update interrupt enable bit 0: disabled 1: enabled This bit can be modified only when the LPTIMER is disabled (The LPTEN bit in LPTIMER_CTL1 register is 0).
3	CMPVUPIE	Compare value register update interrupt enable bit 0: disabled 1: enabled

		This bit can be modified only when the LPTIMER is disabled (The LPTEN bit in LPTIMER_CTL1 register is 0).
2	ETEDEVIE	External trigger edge event interrupt enable bit 0: disabled 1: enabled This bit can be modified only when the LPTIMER is disabled (The LPTEN bit in LPTIMER_CTL1 register is 0).
1	CARMIE	Counter auto reload register match interrupt enable bit 0: disabled 1: enabled This bit can be modified only when the LPTIMER is disabled (The LPTEN bit in LPTIMER_CTL1 register is 0).
0	CMPVMIE	Compare value register match interrupt enable bit 0: disabled 1: enabled This bit can be modified only when the LPTIMER is disabled (The LPTEN bit in LPTIMER_CTL1 register is 0).

19.5.4. Control register 0 (LPTIMER_CTL0)

Address offset: 0x0C

Reset value: 0x0000

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						DECMSEL	DECMEN	CNTMEN	SHWEN	OPSEL	OMSEL	TIMEOUT	ETMEN[1:0]		Reserved
						rw	rw	rw	rw	rw	rw	rw	rw		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ETSEL[2:0]			Reserved	PSC[2:0]			Reserved	TFLT [1:0]		Reserved	ECKFLT[1:0]		CKPSEL[1:0]		CKSSEL
rw				rw				rw			rw		rw		rw

Bits	Fields	Descriptions
31:26	Reserved	Must be kept at reset value.
25	DECMSEL	Decoder mode select 0: Decoder mode 0 1: Decoder mode 1 This bit can be modified only when the LPTIMER is disabled (The LPTEN bit in LPTIMER_CTL1 register is 0).
24	DECMEN	Decoder mode enabled 0: Decoder mode disabled 1: Decoder mode enabled

		This bit can be modified only when the LPTIMER is disabled (The LPTEN bit in LPTIMER_CTL1 register is 0).
23	CNTMEN	Counter mode select This bit is used to select the clock source of the LPTIMER counter. 0: The counter is count with each internal clock pulse 1: The counter is count with each active clock pulse on the LPTIMER_IN0. This bit can be modified only when the LPTIMER is disabled (The LPTEN bit in LPTIMER_CTL1 register is 0).
22	SHWEN	LPTIMER_CAR and LPTIMER_CMPV shadow registers enable 0: The shadow registers are disable. The registers are updated immediately after every APB bus write access. 1: The shadow registers are enable. The registers are updated at the end of the LPTIMER period. This bit can be modified only when the LPTIMER is disabled (The LPTEN bit in LPTIMER_CTL1 register is 0).
21	OPSEL	Output polarity select This bit is used to controls the output polarity. 0: The output is non-inverted. When counting up, the output is set as long as the counter is match the value of LPTIMER_CMPV; The output is reset as long as the counter is match the value of LPTIMER_CAR. 1: The output is inverted. When counting up, the output is reset as long as the counter is match the value of LPTIMER_CMPV; The output is set as long as the counter is match the value of LPTIMER_CAR. This bit can be modified only when the LPTIMER is disabled (The LPTEN bit in LPTIMER_CTL1 register is 0).
20	OMSEL	Output Mode select This bit is used to controls the output mode. 0: PWM mode or single pulse mode (CTNMST bit for PWM mode and SMST for single pulse mode). 1: Set mode. This bit can be modified only when the LPTIMER is disabled (The LPTEN bit in LPTIMER_CTL1 register is 0).
19	TIMEOUT	Timeout mode enable This bit is used to controls the timeout mode. 0: A new trigger event will be ignored after LPTIMER started 1: A new trigger event will reset and restart the count after LPTIMER started This bit can be modified only when the LPTIMER is disabled (The LPTEN bit in LPTIMER_CTL1 register is 0).
18:17	ETMEN[1:0]	External Trigger mode enable The ETMEN bits are used to configure the trigger mode for LPTIMER. 00: External trigger disable (Software trigger)

		01: Rising edge of external trigger enable 01: Falling edge of external trigger enable 11: Rising and falling edges of external trigger enable These bits can be modified only when the LPTIMER is disabled (The LPTEN bit in LPTIMER_CTL1 register is 0).
16	Reserved	Must be kept at reset value.
15:13	ETSEL[2:0]	External trigger select The ETSEL bits are used to select the external trigger source for LPTIMER. 000: ETI0 (GPIO) 001: ETI1 (RTC Alarm 0) 010: ETI2 (RTC Alarm 1) 011: ETI3 (RTC_TAMP0) 100: ETI4 (RTC_TAMP1) 101: ETI5 (RTC_TAMP2) 110: ETI6 (CMP0_OUT) 111: ETI7 (CMP1_OUT) These bits can be modified only when the LPTIMER is disabled (The LPTEN bit in LPTIMER_CTL1 register is 0).
12	Reserved	Must be kept at reset value.
11:9	PSC[2:0]	Clock prescaler selection The PSC bits are used to configure the prescaler to divide the timer clock (LPTIMER_CK) to a counter clock (PSC_CLK). 000: $f_{PSC\_CLK}=f_{LPTIMER\_CK}$ 001: $f_{PSC\_CLK}=f_{LPTIMER\_CK}$ /2 010: $f_{PSC\_CLK}=f_{LPTIMER\_CK}$ /4 011: $f_{PSC\_CLK}=f_{LPTIMER\_CK}$ / 8 100: $f_{PSC\_CLK}=f_{LPTIMER\_CK}$ / 16 101: $f_{PSC\_CLK}=f_{LPTIMER\_CK}$ / 32 110: $f_{PSC\_CLK}=f_{LPTIMER\_CK}$ / 64 111: $f_{PSC\_CLK}=f_{LPTIMER\_CK}$ / 128 These bits can be modified only when the LPTIMER is disabled (The LPTEN bit in LPTIMER_CTL1 register is 0).
8	Reserved	Must be kept at reset value.
7:6	TFLT[1:0]	Trigger filter The TFLT bits are used to configure the digital filter for triggers. An internal clock source must be used in this function. 00: Filter disabled, any active level of the trigger is valid. 01: The active level change of the trigger need to be maintained at least 2 clock periods. 10: The active level change of the trigger need to be maintained at least 4 clock periods.

		11: The active level change of the trigger need to be maintained at least 8 clock periods. These bits can be modified only when the LPTIMER is disabled (The LPTEN bit in LPTIMER_CTL1 register is 0).
5	Reserved	Must be kept at reset value.
4:3	ECKFLT[1:0]	External clock filter The ECKFLT bits are used to configure the digital filter for external clock. An internal clock source must be used in this function. 00: Filter disabled, any active level change of external clock is valid. 01: The active level change of the external clock need to be maintained at least 2 clock periods. 10: The active level change of the external clock need to be maintained at least 4 clock periods. 11: The active level change of the external clock need to be maintained at least 8 clock periods. These bits can be modified only when the LPTIMER is disabled (The LPTEN bit in LPTIMER_CTL1 register is 0).
2:1	CKPSEL[1:0]	Clock polarity select If LPTIMER is clocked by an external clock source, CKPSEL bits is used to configure the active edge or edges used for the counter counting: 00: the rising edge is the active edge used for counting. If the LPTIMER is configured in decoder mode 0 (DECMEN = 1, DECMSEL = 0), the decoder rising-edge-mode is active. If the LPTIMER is configured in decoder mode 1 (DECMEN = 1, DECMSEL = 1), the inputs of LPTIMER_IN0 and LPTIMER_IN1 are non-inverted. If the LPTIMER external input high level counter enable（INHLCEN=1），the inputs of LPTIMER_IN0 and LPTIMER_IN1 are non-inverted. 01: the falling edge is the active edge used for counting If the LPTIMER is configured in decoder mode 0, the decoder falling-edge-mode is active. If the LPTIMER is configured in decoder mode 1, the inputs of LPTIMER_IN0 and LPTIMER_IN1 are inverted. If the LPTIMER external input high level counter enable（INHLCEN=1），the inputs of LPTIMER_IN0 and LPTIMER_IN1 are inverted. 10: both edges are the active edge used for counting. When the both edges of two external clock signals are considered as valid edges, the LPTIMER must be clocked by an internal clock source, and the internal clock frequency is at least equal to four times the external clock frequency. If the LPTIMER is configured in decoder mode 0, the decoder both-edge-mode is active. This is not allowed if the LPTIMER is configured in decoder mode 1.

		If the LPTIMER external input high level counter enable（INHLCEN=1）， the inputs of LPTIMER_IN0 and LPTIMER_IN1 are non-inverted. 11: not allowed These bits can be modified only when the LPTIMER is disabled (The LPTEN bit in LPTIMER_CTL1 register is 0).
0	CKSSEL	Clock source select This bit is used to select the clock source for LPTIMER. 0: LPTIMER is clocked by internal clock source. 1: LPTIMER is clocked by external clock source on the LPTIMER_IN0. This bit can be modified only when the LPTIMER is disabled (The LPTEN bit in LPTIMER_CTL1 register is 0). Note：when the decoder mode 0 enabled (DECMEN=1), the CKSSEL bit will cleared by hardware.

19.5.5. Control register 1 (LPTIMER_CTL1)

Address offset: 0x10

Reset value: 0x0000

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
INHLCEN	LPTENF	Reserved													
rw	r														

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

Bits	Fields	Descriptions
31	INHLCEN	LPTIMER external input high level counter enable. 0: disabled 1: enabled
30	LPTENF	LPTIMER enabled from LPTIMER core flag. This bit is set and reset by hardware. 0: LPTIMER is disabled 1: LPTIMER is enabled
29:3	Reserved	Must be kept at reset value.
2	CTNMST	LPTIMER start for continuous counting mode. This bit is set by software and reset by hardware. This bit can be modified only when the LPTIMER is enabled (The LPTEN bit in LPTIMER_CTL1 register is 1).
1	SMST	LPTIMER start for single counting mode.

Bits	Fields	Descriptions
		This bit is set by software and reset by hardware. This bit can be modified only when the LPTIMER is enabled (The LPTEN bit in LPTIMER_CTL1 register is 1).
0	LPTEN	LPTIMER enable This bit is set and reset by software. 0: LPTIMER is disabled 1: LPTIMER is enabled

19.5.6. Compare value register (LPTIMER_CMPV)

For GD32L233xx devices

Address offset: 0x14

Reset value: 0x0000

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
CMPVAL[31:16]															
rw															

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

Bits	Fields	Descriptions
31:0	CMPVAL[31:0]	Compare value This bit-filed specifies the compare value of the counter. This bit-filed can be modified only when the LPTIMER is enabled (The LPTEN bit in LPTIMER_CTL1 register is 1).

For GD32L235xx devices

Address offset: 0x14

Reset value: 0x0000

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
CMPVAL[15:0]															
rw															

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.

15:0	CMPVAL[15:0]	Compare value This bit-filed specifies the compare value of the counter. This bit-filed can be modified only when the LPTIMER is enabled (The LPTEN bit in LPTIMER_CTL1 register is 1).

19.5.7. Counter auto reload register (LPTIMER_CAR)

For GD32L233xx devices

Address offset: 0x18

Reset value: 0x0001

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
CARL[31:16]															
rw															

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

Bits	Fields	Descriptions
31:0	CARL[31:0]	Counter auto reload value This bit-filed specifies the auto reload value of the counter. This bit-filed can be modified only when the LPTIMER is enabled (The LPTEN bit in LPTIMER_CTL1 register is 1).

For GD32L235xx devices

Address offset: 0x18

Reset value: 0x0001

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. This bit-filed can be modified only when the LPTIMER is enabled (The LPTEN bit in

LPTIMER_CTL1 register is 1).

19.5.8. Counter register (LPTIMER_CNT)

For GD32L233xx devices

Address offset: 0x1C

Reset value: 0x0000

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
CNT[31:16]															
r															

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

Bits	Fields	Descriptions
31:0	CNT[31:0]	Counter value **Note:** When the LPTIMER uses an asynchronous clock, reads the LPTIMER_CNT register may return unreliable values. So it is necessary to perform two consecutive read operations and confirm whether the two read values are the same.

For GD32L235xx devices

Address offset: 0x1C

Reset value: 0x0000

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]															
r															

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	CNT[15:0]	Counter value **Note:** When the LPTIMER uses an asynchronous clock, reads the LPTIMER_CNT register may return unreliable values. So it is necessary to perform two consecutive read operations and confirm whether the two read values are the same.

19.5.9. External input remap register (LPTIMER_EIRMP)

Address offset: 0x20

Reset value: 0x0000

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														IN1_RMP	IN0_RMP
														rw	rw

Bits	Fields	Descriptions
31:2	Reserved	Must be kept at reset value.
1	IN1_RMP	External input1 remap 0: External input is remaped to GPIO. 1: External input is remaped to CMP1_OUT.
0	IN0_RMP	External input0 remap 0: External input is remaped to GPIO. 1: External input is remaped to CMP0_OUT.

19.5.10. Input high level counter max value register (LPTIMER_INHLCMV)

Address offset: 0X24

Reset value: 0x0000

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						INHLCMVAL [25:16]									
						rw									

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

Bits	Fields	Descriptions
31:26	Reserved	Must be kept at reset value.
25:0	INHLCMVAL	Input high level counter max value This bit can be modified only when the LPTIMER is external input high level counter enabled (The INHLCEN bit in LPTIMER_CTL1 register is 1).

20. Universal synchronous/asynchronous receiver / transmitter (USART)

20.1. 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 IRC16M) 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.

20.2. 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 8 Mbits/s when the clock frequency is 64 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 and USART1 is fully implemented, UART3 / UART4 is only partially implemented with the following features not supported.

- Smartcard mode.
- IrDA SIR ENDEC block.
- LIN mode.
- Dual clock domain and wakeup from deep-sleep mode.
- Receiver timeout interrupt.
- ModBus communication.
- Synchronous mode.
- Hardware flow control.

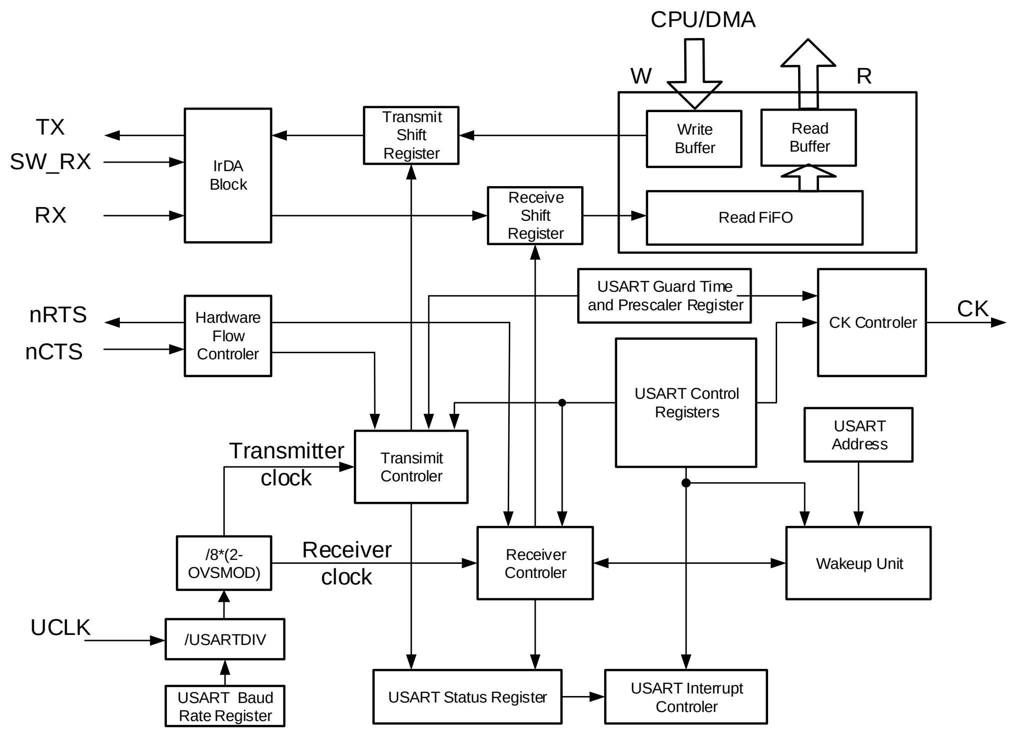
20.3. Function overview

The interface is externally connected to another device by the main pins listed in *Table 20-1. Description of USART important pins*.

Table 20-1. 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
CK	Output	Serial clock for synchronous communication
nCTS	Input	Clear to send in Hardware flow control mode
nRTS	Output	Request to send in Hardware flow control mode

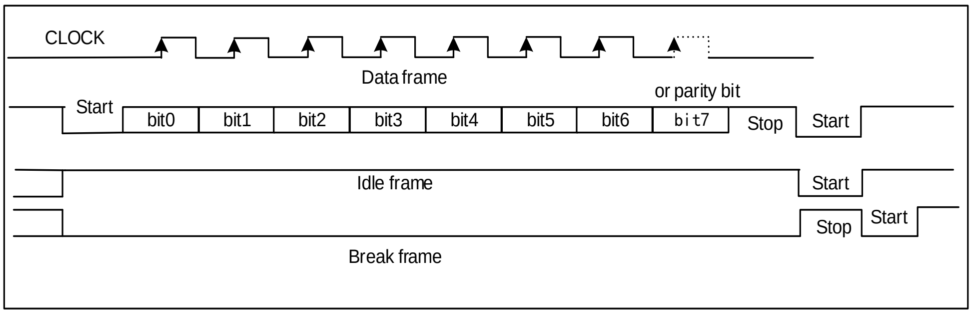
Figure 20-1. USART module block diagram



20.3.1. 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 7^{th} bit. When the WL bit is set, the parity bit is the 8^{th} bit. The method of calculating the parity bit is selected by the PM bit in USART_CTL0 register.

Figure 20-2. 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 20-2. 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.

20.3.2. 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 peripheral clock:

In case of oversampling by 16, the equation is:

$$USARTDIV=\frac{UCLK}{16\times Baud\ Rate} \tag{19-1}$$

In case of oversampling by 8, the equation is:

$$USARTDIV=\frac{UCLK}{8\times Baud\ Rate} \tag{19-2}$$

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.

20.3.3. 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.

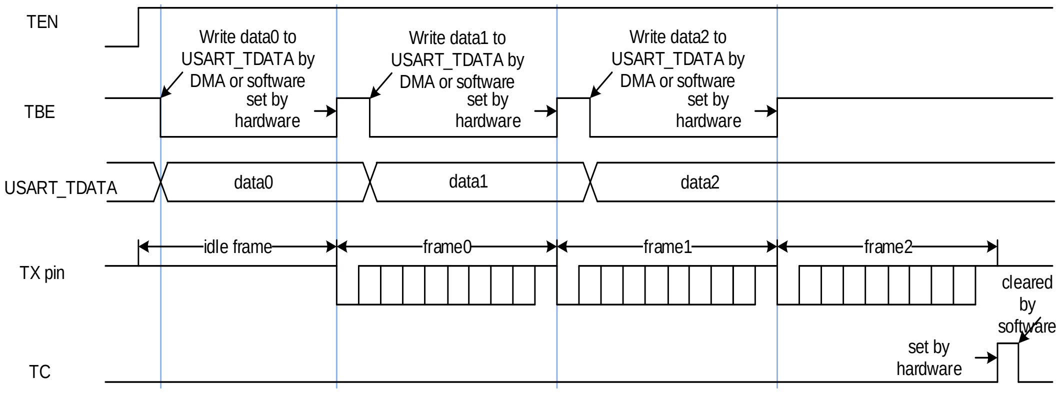
After power on, the TBE bit is high by default. Data can be written to the USART_TDATA when the TBE bit in the USART_STAT register is asserted. The TBE bit is cleared by writing USART_TDATA register and it is set by hardware after the data is put into the transmit shift register. If a data is written 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 20-3. 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 20-3. 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.

20.3.4. 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.

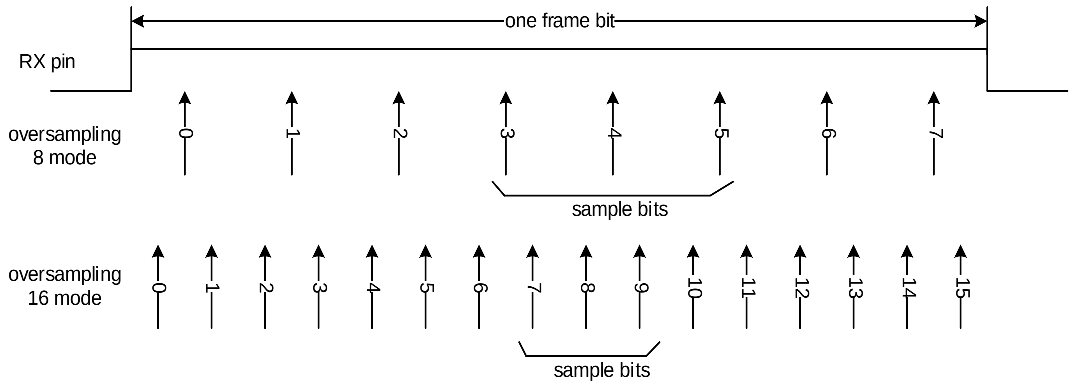
When a frame is received, the RBNE bit in USART_STAT is asserted, an interrupt is generated if the corresponding interrupt enable bit (RBNEIE) is set in the USART_CTL0 register. The status of the reception are stored in the USART_STAT register.

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 3^{rd}, 4^{th} and 5^{th} samples are used, while in the oversampling 16 mode, the 7^{th}, 8^{th}, and 9^{th} 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 20-4. 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.

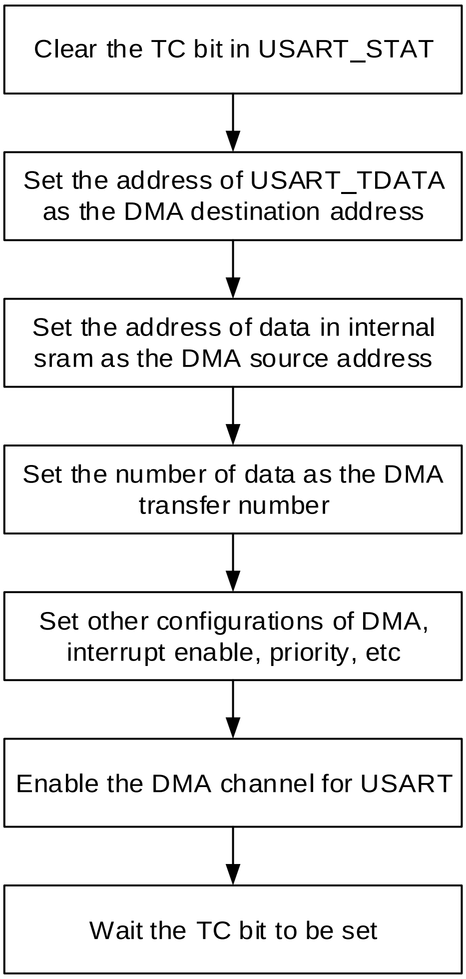
The RBNE, NERR, PERR, FERR and ORERR flags are always set at the same time in a reception. If the receive DMA is not enabled, software can check NERR, PERR, FERR and ORERR flags when serving the RBNE interrupt.

20.3.5. 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 20-5. Configuration step when using DMA for USART transmission***.

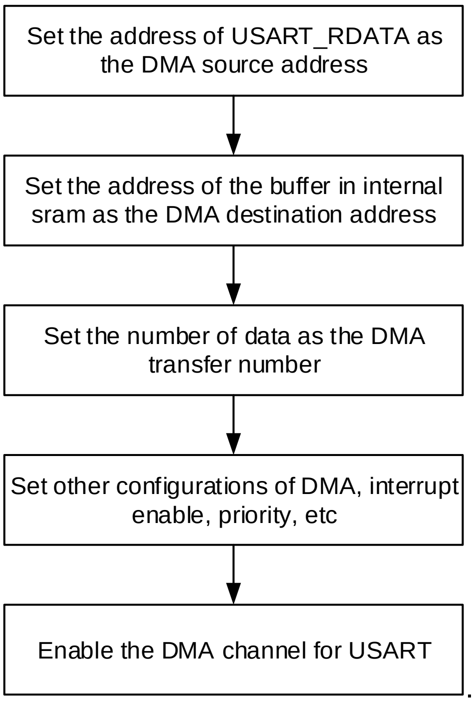
Figure 20-5. Configuration step when using DMA for USART transmission



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 20-6. 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 20-6. Configuration step when using DMA for USART reception

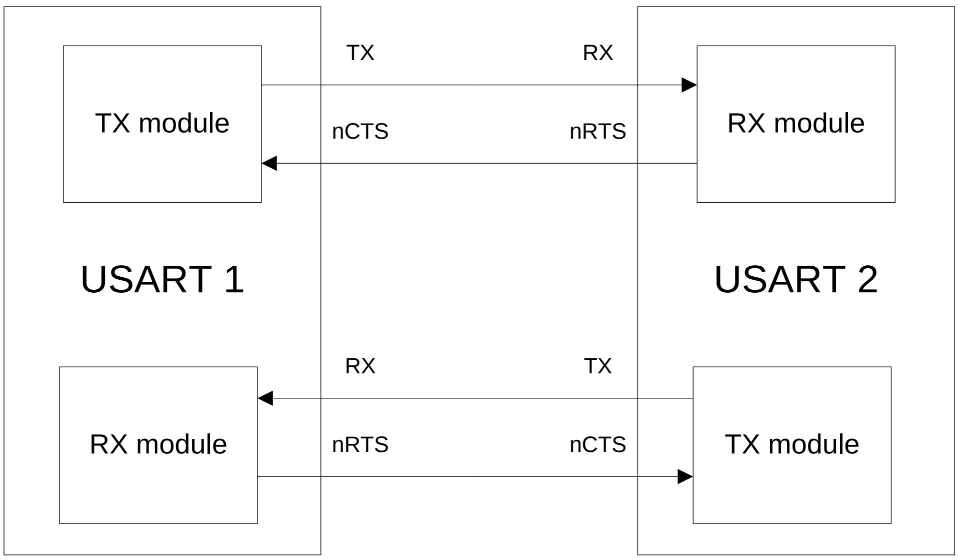


When the number of the data received by USART reaches the DMA transfer number, an end of transfer interrupt can be generated in the DMA module.

20.3.6. 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 20-7. Hardware flow control between two USARTs



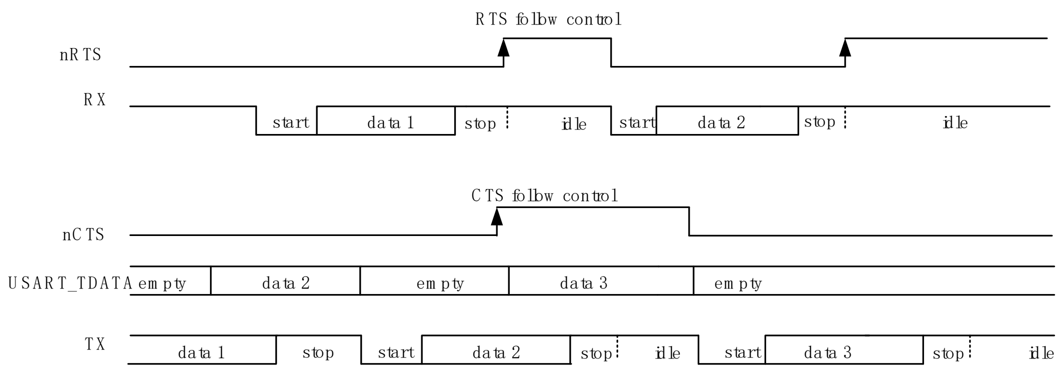
RTS flow control

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 20-8. 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.

20.3.7. 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.

20.3.8. LIN mode

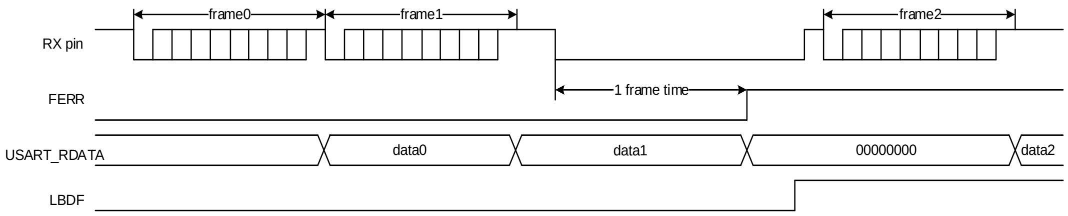
The local interconnection network mode is enabled by setting the LMEN bit in USART_CTL1. The CKEN, STB[1:0] bit in USART_CTL1 and the SCEN, HDEN, IREN bits in USART_CTL2 should be cleared in LIN mode.

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.

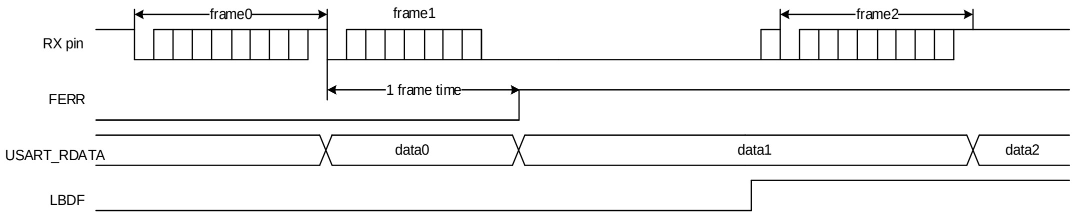
As shown in ***Figure 20-9. Break frame occurs during idle state***, if a break frame occurs during the idle state on the RX pin, the USART receiver will receive an all ‘0’ frame, with an asserted FERR status.

Figure 20-9. Break frame occurs during idle state



As shown in ***Figure 20-10. 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 20-10. Break frame occurs during a frame



20.3.9. 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 20-11. Example of USART in synchronous mode

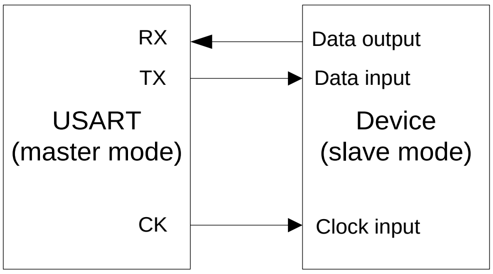
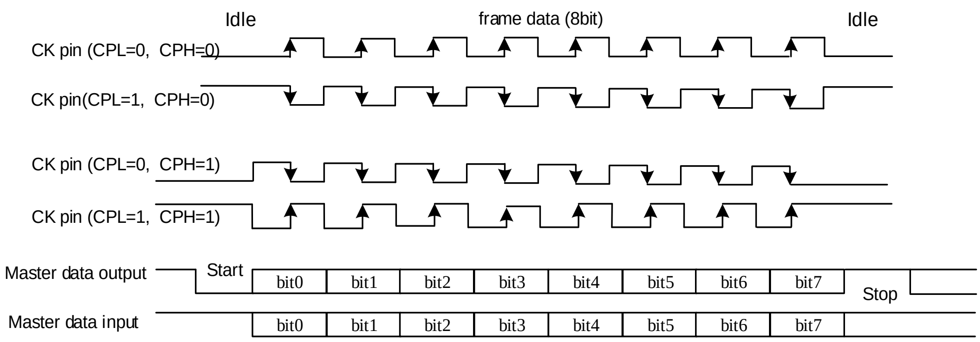


Figure 20-12. 8-bit format USART synchronous waveform (CLEN=1)

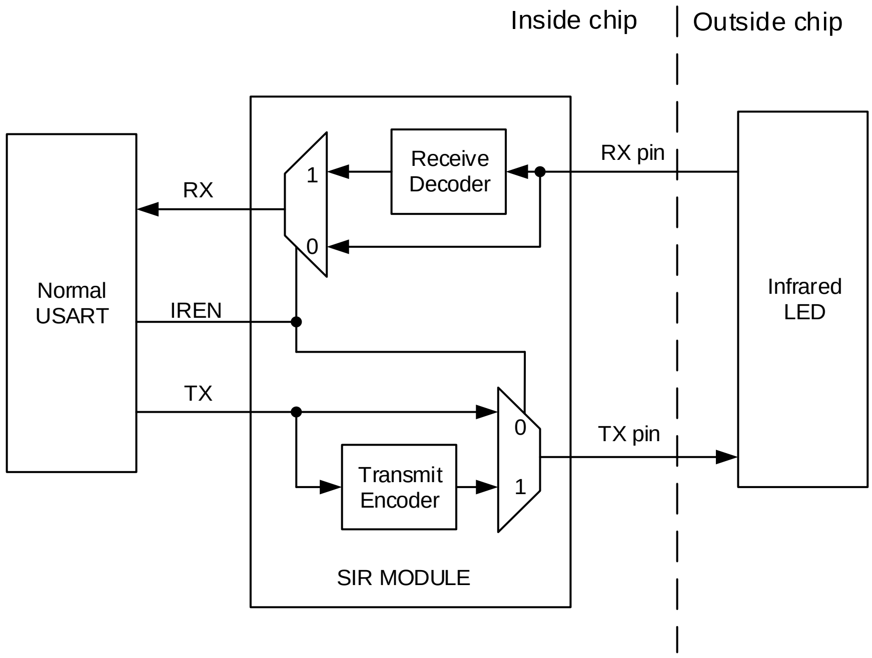


20.3.10. IrDA SIR ENDEC mode

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.

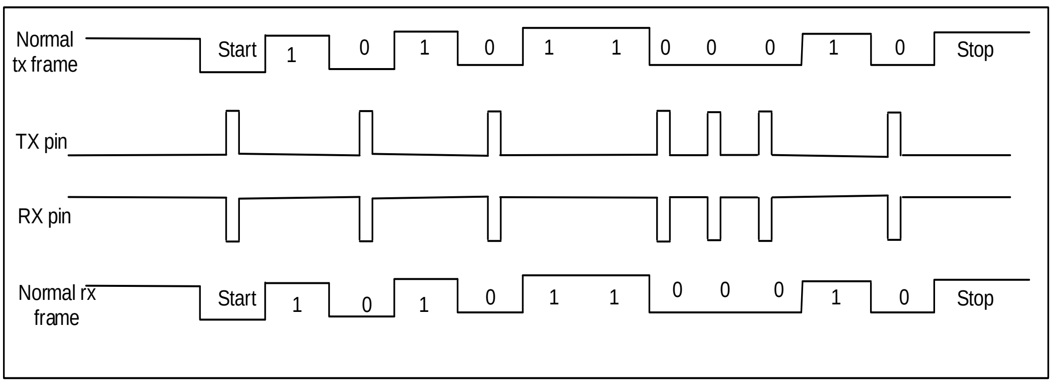
Figure 20-13. IrDA SIR ENDEC module



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 20-14. 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.

20.3.11. 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. The TX pin should be configured as IO pin. The conflicts should be controlled by the software. When the TEN bit is set, the data in the data register will be sent.

20.3.12. Smartcard (ISO7816-3) mode

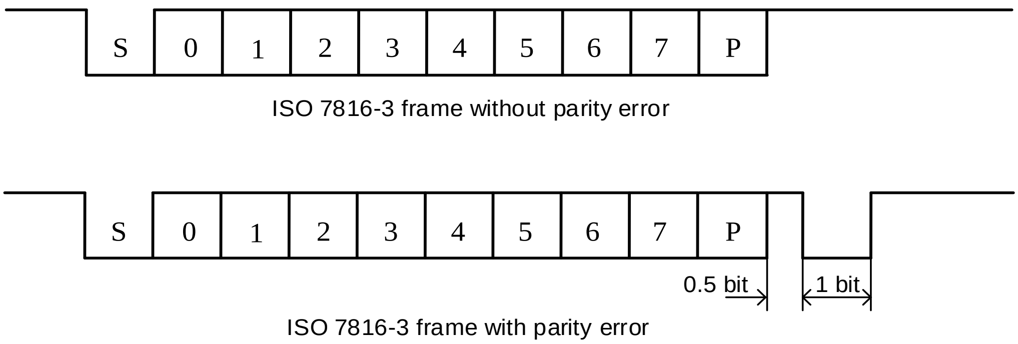
The smartcard mode is an asynchronous mode, which is designed to support the ISO7816-3 protocol. Both the character (T=0) mode and the block (T=1) mode are supported. The smartcard mode is enabled by setting the SCEN bit in USART_CTL2. The LMEN bit in USART_CTL1 and HDEN, IREN bits in USART_CTL2 should 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 20-15. ISO7816-3 frame format



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 4^{th} 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.

20.3.13. 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.

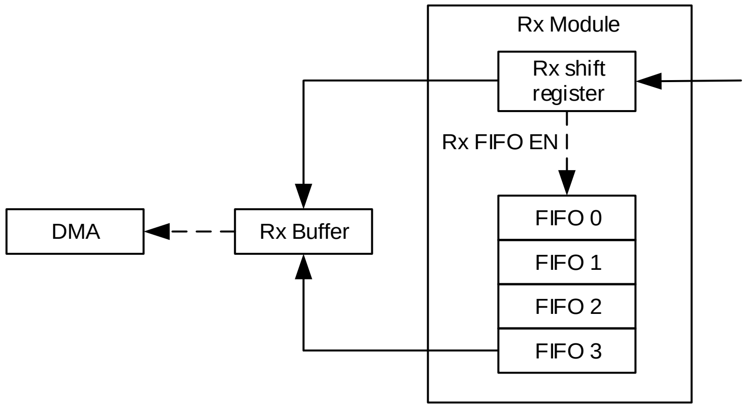
To detect the idle line, the RTEN bit in the USART_CTL1 register and the RTIE in the USART_CTL0 register must be set. The USART_RT register must be set to the value corresponding to a timeout of 2 characters time. After the last stop bit is received, when the receive line is idle for this duration, an 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.

20.3.14. 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 20-16. 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.

20.3.15. 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 IRC16M 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.

20.3.16. USART interrupts

The USART interrupt events and flags are listed in ***Table 20-3. USART interrupt requests***.

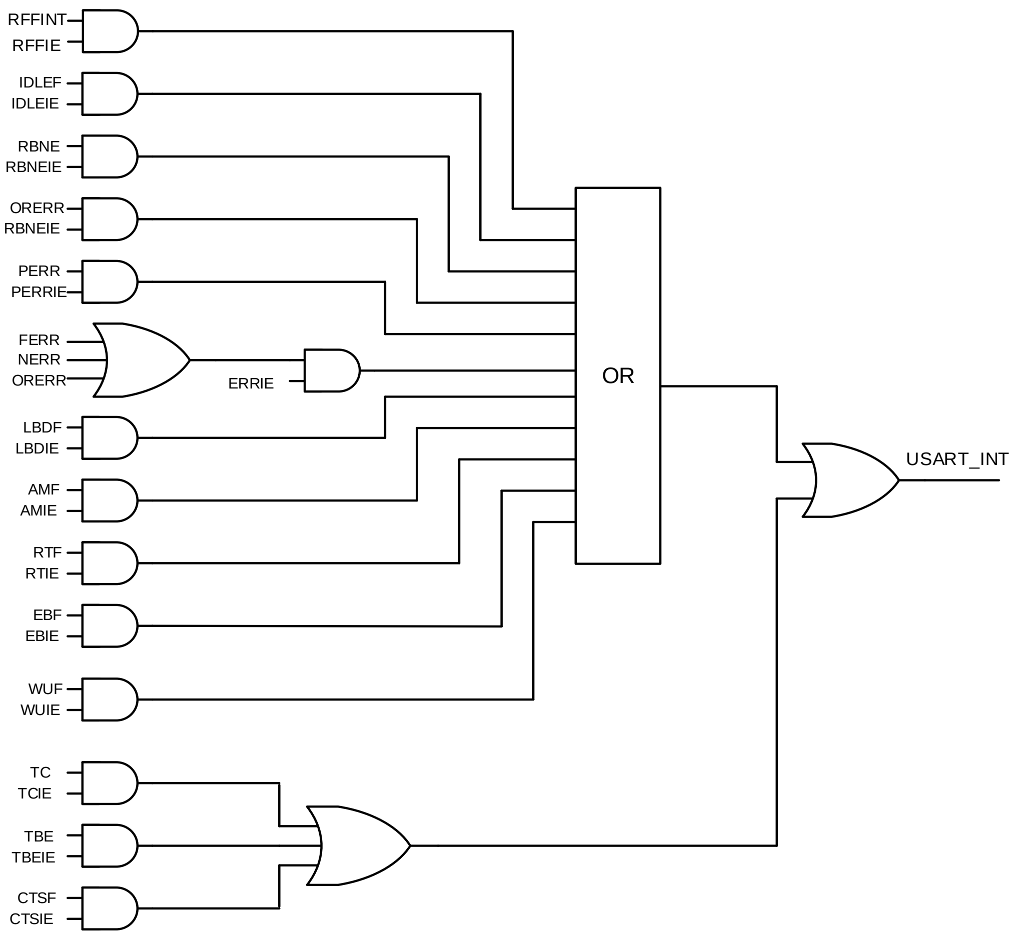
Table 20-3. USART interrupt requests

Interrupt event	Event flag	Enable Control bit
Transmit data register empty	TBE	TBEIE
CTS flag	CTSF	CTSIE

Interrupt event	Event flag	Enable Control bit
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 20-17. USART interrupt mapping diagram



20.4. Register definition

USART0 base address: 0x4001 3800

USART1 base address: 0x4000 4400

UART3 base address: 0x4000 4C00

UART4 base address: 0x4000 5000

20.4.1. 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 UART3 and UART4.
26	RTIE	Receiver timeout interrupt enable 0: Receiver timeout interrupt is disabled 1: Receiver timeout interrupt is enabled This bit is reserved in UART3 and UART4.
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 UART3 and UART4.
0	UEN	USART enable 0: USART prescaler and outputs disabled 1: USART prescaler and outputs enabled

20.4.2. 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 UART3 and UART4.
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 UART3 and UART4.
13:12	STB[1:0]	STOP bits length 00: 1 Stop bit 01: 0.5 Stop bit 10: 2 Stop bits 11: 1.5 Stop bit

		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). This bit is reserved in UART3 and UART4.
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 UART3 and UART4.
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 UART3 and UART4.
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.

20.4.3. 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 UART3 and UART4.
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 UART3 and UART4.
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 UART3 and UART4.
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 UART3 and UART4.
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 UART3 and UART4.
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 UART3 and UART4.
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.

20.4.4. 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.

20.4.5. 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 UART3, UART4.

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).

20.4.6. 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 UART3 and UART4.

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.

20.4.7. 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 UART3 and UART4.
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.

20.4.8. 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 UART3 and UART4.
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 UART3 and UART4.
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 UART3 and UART4
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 UART3 and UART4.
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 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.

20.4.9. 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 UART3 and UART4.
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 UART3 and UART4.
11	RTC	Receiver timeout clear Writing 1 to this bit clears the RTF flag in the USART_STAT register. This bit is reserved in UART3 and UART4.
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 UART3 and UART4.
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.

20.4.10. 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.
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).

20.4.11. 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.

20.4.12. 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.

20.4.13. 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 UART3 and UART4.

21. Low-power universal asynchronous receiver /transmitter (LPUART)

21.1. Overview

The Low-power universal Asynchronous Receiver/Transmitter (LPUART) provides a flexible serial data exchange interface with a limited power consumption. LPUART can perform asynchronous serial communication even with low power consumption. Data frames can be transferred in full duplex or half duplex mode, asynchronously through this interface. A programmable baud rate generator divides the LPUCLK (PCLK1, CK_SYS, LXTAL and IRC16M) to produces a dedicated wide range baudrate clock for the LPUART transmitter and receiver.

Besides the standard asynchronous receiver and transmitter mode, the LPUART also implements a half-duplex serial data exchange 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.

LPUART support DMA function for high-speed data communication.

21.2. Characteristics

- NRZ standard format.
- Asynchronous, full duplex communication.
- Half duplex single wire communications.
- Dual clock domain:
  - Asynchronous pclk and LPUART clock.
  - Baud rate programming independent from the LPUCLK reprogramming.
- Programmable baud-rate from 300 baud to 9600 baud using LXTAL clock.
- Programmable baud-rate generator allowing speed up to 10 Mbits/s when the clockfrequency is 32 MHz.
- Fully programmable serial interface characteristics:
  - A data word (7 or 8 or 9 bits) LSB or MSB first.
  - Even, odd or no-parity bit generation/detection.
  - 1 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.
- Multiprocessor communication:
  - Enter into mute mode if address match does not occur.
  - Wake up from mute mode by idle line or address match detection.
- 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), 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 multiprocessor communication: IDLE frame detected (IDLEF).
  - Wakeup from Deep-sleep mode flag (WUF).
  - Interrupt occurs at these events when the corresponding interrupt enable bits are set.

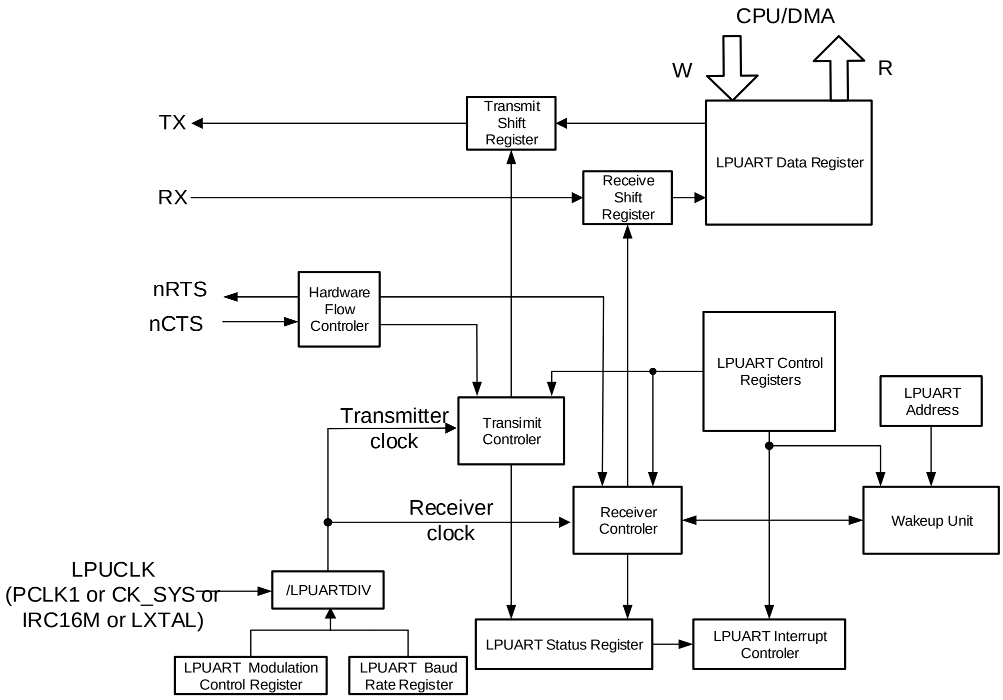
21.3. Function overview

The interface is externally connected to another device by the main pins listed in ***Table 21-1. Description of LPUART important pins***.

Table 21-1. Description of LPUART important pins

Pin	Type	Description
RX	Input	Receive Data
TX	Output I/O (single-wire)	Transmit Data. High level When enabled but nothing to be transmitted
nCTS	Input	Clear to send in Hardware flow control mode
nRTS	Output	Request to send in Hardware flow control mode

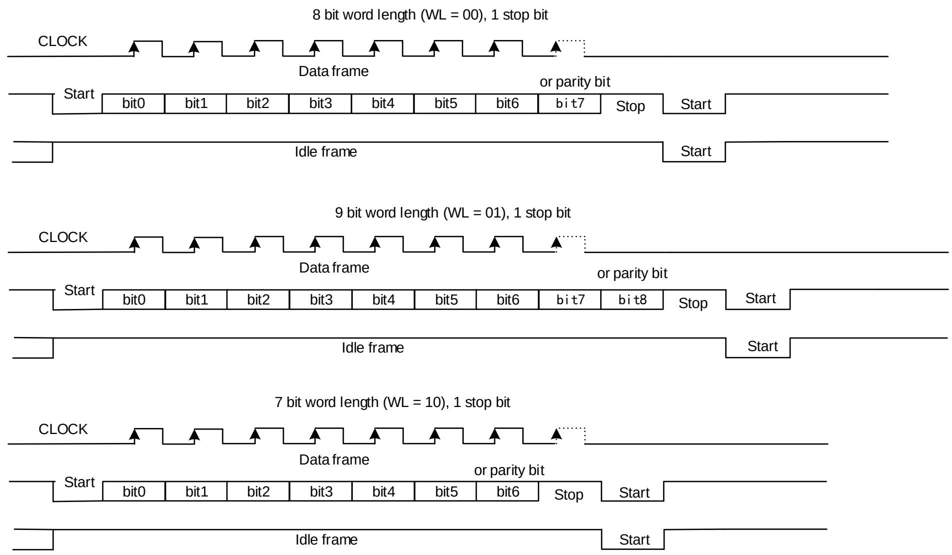
Figure 21-1. LPUART module block diagram



21.3.1. LPUART frame format

The LPUART 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[1:0] bit in the LPUART_CTL0 register, refer to ***Figure 21-2. LPUART character frame***. The method of calculating the parity bit is selected by the PM bit in LPUART_CTL0 register.

Figure 21-2. LPUART character frame



In transmission and reception, the number of stop bits can be configured by the STB[1:0] bits in the LPUART_CTL1 register:

- STB[1:0] = 00: 1 stop bit length
- STB[1:0] = 10: 2 stop bit length

In an idle frame, all the frame bits are logic 1. The frame length is equal to the normal LPUART frame.

21.3.2. Baud rate generation

The baudrate divider is a 20-bits number. The number is used by the baudrate generator to determine the bit period. The baudrate divider (LPUARTDIV) has the following relationship with the LPUCLK:

$$\text{LPUARTDIV}=\frac{256\times \text{LPUCLK}}{\text{Baud Rate}} \tag{21-1}$$

Where:

- LPUARTDIV: The baudrate divider, which is defined in LPUART_BAUD register.

Note:

1. The value of LPUART_BAUD[19:0] must be greater than 0x300.
2. (3 x baudrate) ≤ LPUCLK ≤ (4096 x baudrate).
3. The value of the LPUART_BAUD register should not be changed during communication.

21.3.3. LPUART transmitter

If the transmit enable bit (TEN) in LPUART_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 LPUART_CTL1 register.

After the TEN bit is set, an idle frame will be sent. The TEN bit should not be cleared while the transmission is ongoing.

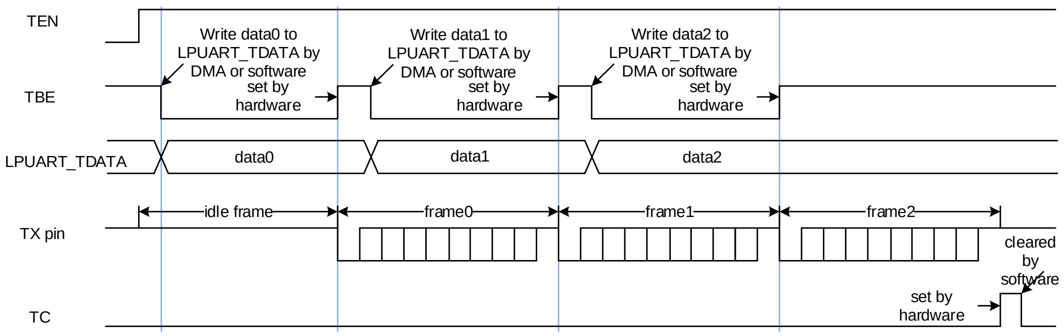
After power on, the TBE bit is high by default. Data can be written to the LPUART_TDATA when the TBE bit in the LPUART_STAT register is set. The TBE bit is cleared by writing LPUART_TDATA register and it is set by hardware after the data is put into the transmit shift register. If a data is written to the LPUART_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 LPUART_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 LPUART_STAT register will be set. An interrupt will be generated if the corresponding interrupt enable bit (TCIE) is set in the LPUART_CTL0 register.

The LPUART transmit procedure is shown in *Table 21-3. LPUART interrupt requests*. The software operating process is as follows:

1. Write the WL[1:0] bits in LPUART_CTL0 to set the data bits length.
2. Set the STB[1:0] bits in LPUART_CTL1 to configure the number of stop bits.
3. Enable DMA (DENT bit) in LPUART_CTL2 if multibuffer communication is selected.
4. Set the baud rate in LPUART_BAUD.
5. Set the UEN bit in LPUART_CTL0 to enable the LPUART.
6. Set the TEN bit in LPUART_CTL0.
7. Wait for the TBE being asserted.
8. Write the data to the LPUART_TDATA register.
9. Repeat step7-8 for each data, if DMA is not enabled.
10. Wait until TC=1 to finish.

Figure 21-3. LPUART transmit procedure



It is necessary to wait for the TC bit to be asserted before disabling the LPUART or entering the power saving mode. This bit can be cleared by set TCC bit in LPUART_INTC register.

21.3.4. LPUART receiver

After power on, the LPUART receiver can be enabled by the following procedure:

1. Write the WL[1:0] bits in LPUART_CTL0 to set the data bits length.
2. Set the STB[1:0] bits in LPUART_CTL1.
3. Enable DMA (DENR bit) in LPUART_CTL2 if multibuffer communication is selected.
4. Set the baud rate in LPUART_BAUD.
5. Set the UEN bit in LPUART_CTL0 to enable the LPUART.
6. Set the REN bit in LPUART_CTL0.

After being enabled, the start bit is detected when a falling edge occurs on the RX line, and then samples are taken in the middle of the start bit to confirm whether the level is still “0”. If the start sample is “1”, the noise error flag (NERR) is set, the start bit will be discarded, and the receiver will wait for a new start bit, an interrupt is generated, if the ERRIE bit in LPUART_CTL2 register is set. The receiver receives a bit stream after a valid start pulse has been detected. Detection on parity error, frame error and overrun error is performed during the reception of a frame. The receiver gets a sample in the middle of the bit to evaluate its value, there is no noise detection for data.

When a frame is received, the RBNE bit in LPUART_STAT is asserted, an interrupt is generated if the corresponding interrupt enable bit (RBNEIE) is set in the LPUART_CTL0 register. The status of the reception are stored in the LPUART_STAT register.

The software can get the received data by reading the LPUART_RDATA register directly, or through DMA. The RBNE bit is cleared by a read operation on the LPUART_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.

If the parity check function is enabled by setting the PCEN bit in the LPUART_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 LPUART_STAT register will be set. An interrupt is generated, if the PERRIE bit in LPUART_CTL0 register is set.

If the RX pin is evaluated as 0 during a stop bit, the frame error (FERR) bit in LPUART_STAT register will be set. An interrupt is generated, If the ERRIE bit in LPUART_CTL2 register is set. When the number of stop bits is configured as 1 bit, sampling in the middle of the stop bit. When the number of stop bits is configured as 2 bits, sampling in the middle of the second stop bit and the first stop bit is not checked 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 LPUART_STAT register will be set. An interrupt is generated, if the ERRIE bit in LPUART_CTL2 register is set, or if the RBNEIE is set.

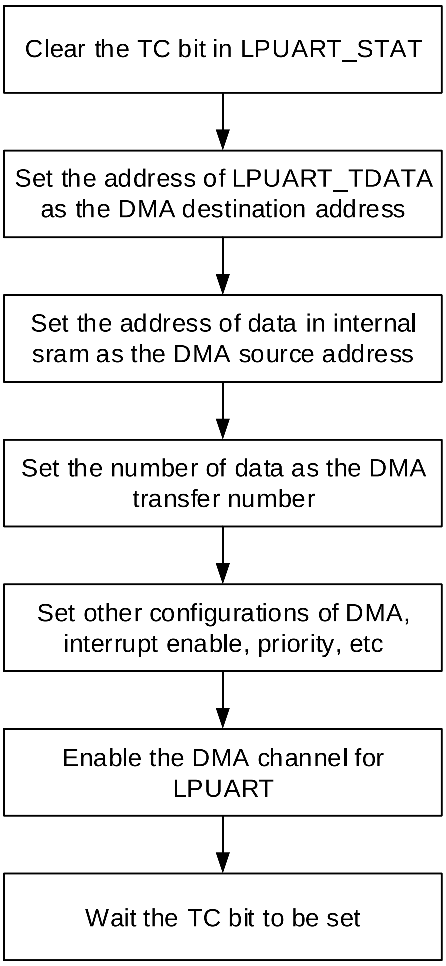
The NERR, PERR, FERR and ORERR flags are always set at the same time with the RBNE in a reception. If the receive DMA is not enabled, software can check NERR, PERR, FERR and ORERR flags when serving the RBNE interrupt.

21.3.5. 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 LPUART_CTL2 is used to enable the DMA transmission, and the DENR bit in LPUART_CTL2 is used to enable the DMA reception.

When DMA is used for LPUART transmission, DMA transfers data from internal SRAM to the transmit data buffer of the LPUART. The configuration step are shown in ***Figure 21-4. Configuration step when using DMA for LPUART transmission***.

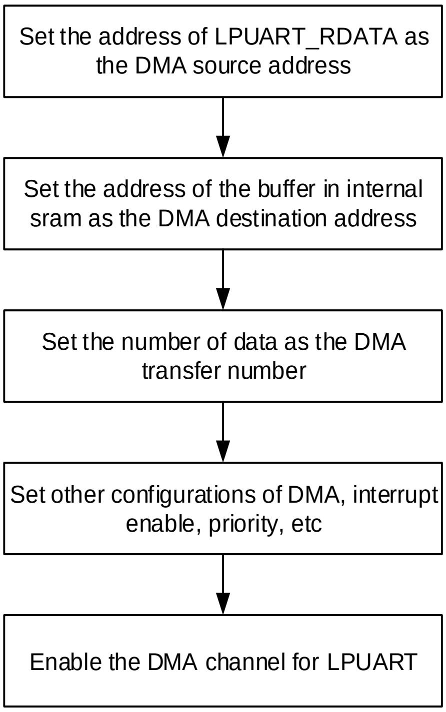
Figure 21-4. Configuration step when using DMA for LPUART transmission



After all of the data frames are transmitted, the TC bit in LPUART_STAT is set. An interrupt occurs if the TCIE bit in LPUART_CTL0 is set.

When DMA is used for LPUART reception, DMA transfers data from the receive data buffer of the LPUART to the internal SRAM. The configuration steps are shown in ***Figure 21-5. Configuration step when using DMA for LPUART reception***. If the ERRIE bit in LPUART_CTL2 is set, interrupts can be generated by the Error status bits (FERR, ORERR and NERR) in LPUART_STAT.

Figure 21-5. Configuration step when using DMA for LPUART reception

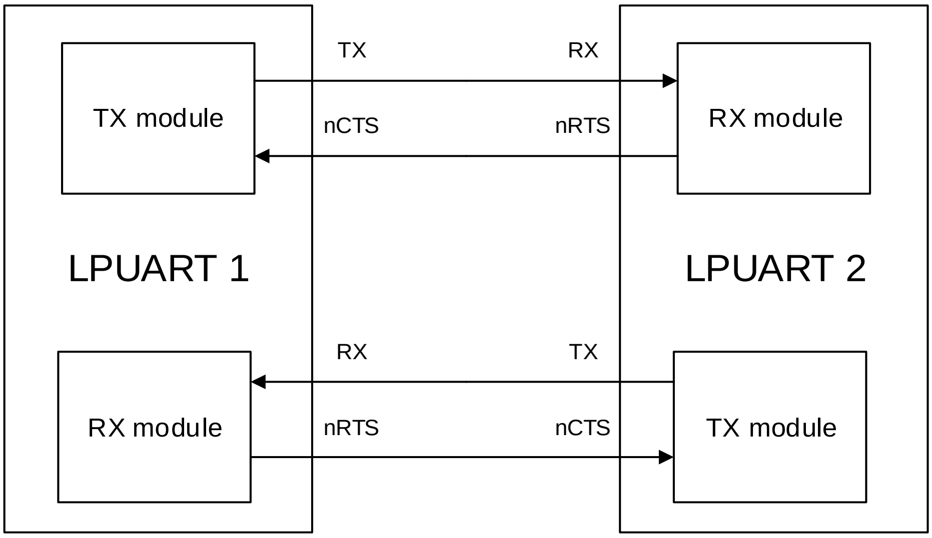


When the number of the data received by LPUART reaches the DMA transfer number, an end of transfer interrupt can be generated in the DMA module.

21.3.6. 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 LPUART_CTL2 and the CTS flow control is enabled by writing ‘1’ to the CTSEN bit in LPUART_CTL2.

Figure 21-6. Hardware flow control between two LPUARTs



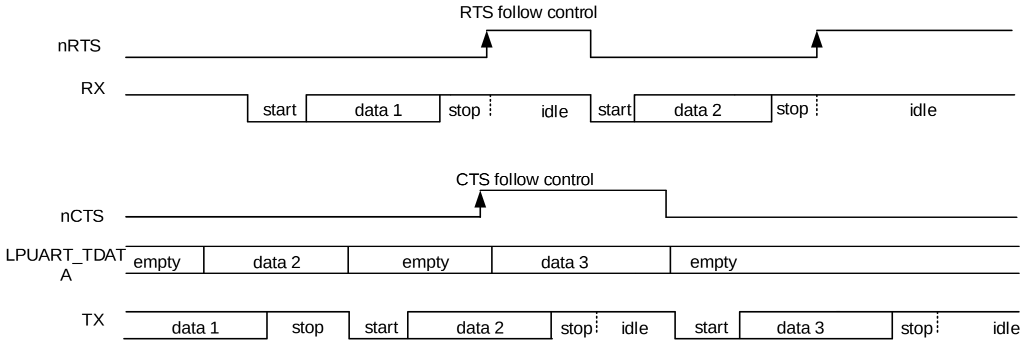
RTS flow control

The LPUART 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 LPUART transmitter monitors the nCTS input pin to decide whether a data frame can be transmitted. If the TBE bit in LPUART_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 21-7. Hardware flow control



RS485 Driver Enable

The driver enable feature, which is enabled by setting bit DEM in the LPUART_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 LPUART_CTL0 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 LPUART_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 LPUART_CTL2 control register.

In LPUART, the DEA and DED are expressed in LPUCLK (f_{ck}), as shown in ***Table 21-2. The driver enable assertion time and de-assertion time***.

Table 21-2. The driver enable assertion time and de-assertion time

BRR[14:11]	The driver enable assertion time	The Driver enable de-assertion time
BRR[14:11] = 0	$(1+DEA)\times f_{ck}$	$(1+DED)\times f_{ck}$
BRR[14:11] ≠ 0	$(1+(DEA\times BRR[14:11]))\times f_{ck}$	$(1+(DED\times BRR[14:11]))\times f_{ck}$

21.3.7. Multi-processor communication

In multiprocessor communication, several LPUARTs 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, user can enable the mute mode using the MEN bit in the LPUART_CTL0 register, software can put an LPUART module into a mute mode by writing 1 to the MMCMD bit in LPUART_CMD register.

If a LPUART is in mute mode, all of the receive status bits cannot be set. The LPUART 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, he IDLEF bit in LPUART_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 LPUART_STAT will not be set.

When the WM bit of in LPUART_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 LPUART_CTL1 register, of an address frame is the same as the ADDR bits in the LPUART_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 LPUART. The status bits are available in the LPUART_STAT register. If the LSB 4/7 bits of an address frame defers from the ADDR bits in the LPUART_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 LPUART_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.

21.3.8. Half-duplex communication mode

The half-duplex communication mode is enabled by setting the HDEN bit in LPUART_CTL2.

Only one wire is used in half-duplex mode. The TX and RX pins are connected together internally. The TX pin should be configured as IO pin. The conflicts should be controlled by the software. When the TEN bit is set, the data in the data register will be sent.

21.3.9. Wakeup from Deep-sleep mode

The LPUART 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 LPUART clock must be set to IRC16M or LXTAL (refer to ***Configuration register 2 (RCU_CFG2)***). When the LPUART clock source is configured to be IRC16M or LXTAL, it is possible to keep enabled this clock during Deep-sleep mode by setting the UCESM bit in LPUART_CTL2 register.

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 can 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 BSY flag in the LPUART_STAT register to guarantee the LPUART is not performing a transfer. The REA bit must be checked to ensure the LPUART 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 Deep-sleep or active mode.

21.3.10. LPUART interrupts

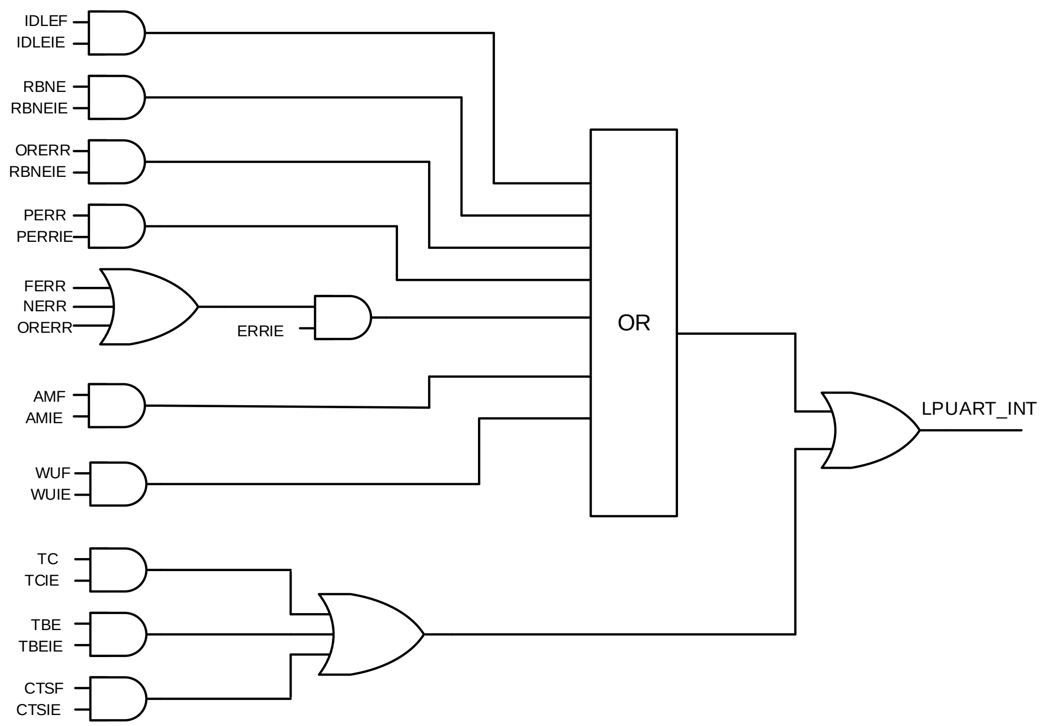
The LPUART interrupt events and flags are listed in ***Table 21-3. LPUART interrupt requests***.

Table 21-3. LPUART 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	
Idle line detected	IDLEF	IDLEIE
Parity error flag	PERR	PERRIE
Reception Errors (Noise flag, overrun error, framing error)	NERR or ORERR or FERR	ERRIE
Character match	AMF	AMIE
Wakeup from Deep-sleep mode	WUF	WUIE

All of the interrupt events are Ored together before being sent to the interrupt controller, so the LPUART 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 21-8. LPUART interrupt mapping diagram



21.4. Register definition

For GD32L233xx devices

LPUART0 base address: 0x4000 8000

For GD32L235xx devices

LPUART0 base address: 0x4000 8000

LPUART1 base address: 0x4000 4800

21.4.1. Control register 0 (LPUART_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			WL1	Reserved		DEA[4:0]					DED[4:0]				
			rw			rw					rw				

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	AMIE	MEN	WL0	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

Bits	Fields	Descriptions
31:29	Reserved	Must be kept at reset value.
28	WL1	Word length This bit, with WL0 bit determines the word length WL[1:0] = 00, 8 data bits WL[1:0] = 01, 9 data bits WL[1:0] = 10, 7 data bits WL[1:0] = 11, 7 data bits This bit field cannot be written when the LPUART is enabled (UEN=1).
27:26	Reserved	Must be kept at reset value.
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 LPUART CLK cycles. This bit field cannot be written when the LPUART 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 LPUART CLK cycles. This bit field cannot be written when the LPUART is enabled (UEN=1).
15	Reserved	Must be kept at reset value.
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	WL0	Word length This bit, with WL1 bit determines the word length WL[1:0] = 00, 8 data bits WL[1:0] = 01, 9 data bits WL[1:0] = 10, 7 data bits WL[1:0] = 11, 7 data bits This bit field cannot be written when the LPUART 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 LPUART 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 LPUART is enabled (UEN=1).
9	PM	Parity mode 0: Even parity 1: Odd parity This bit field cannot be written when the LPUART 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 LPUART_STAT.
7	TBEIE	Transmitter register empty interrupt enable 0: Interrupt is inhibited 1: An interrupt will occur whenever the TBE bit is set in LPUART_STAT
6	TCIE	Transmission complete interrupt enable If this bit is set, an interrupt occurs when the TC bit in LPUART_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 LPUART_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 LPUART_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	LPUART enable in Deep-sleep mode 0: LPUART not able to wake up the MCU from Deep-sleep mode. 1: LPUART able to wake up the MCU from Deep-sleep mode. Providing that the clock source for the LPUART must be IRC16M or LXTAL.
0	UEN	LPUART enable 0: LPUART prescaler and outputs disabled 1: LPUART prescaler and outputs enabled

21.4.2. Control register 1 (LPUART_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]								Reserved				MSBF	DINV	TINV	RINV
rw												rw	rw	rw	rw

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

Bits	Fields	Descriptions
31:24	ADDR[7:0]	Address of the LPUART terminal These bits give the address of the LPUART 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 LPUART (UEN=1) are enabled.
23:20	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 LPUART 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 LPUART 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 LPUART 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 LPUART 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 LPUART is enabled (UEN=1).
14	Reserved	Must be kept at reset value.
13:12	STB[1:0]	STOP bits length 00 ~ 01: 1 Stop bit 01 ~ 11: 2 Stop bits This bit field cannot be written when the LPUART is enabled (UEN=1).
11:5	Reserved	Must be kept at reset value.
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 LPUART is enabled (UEN=1).
3:0	Reserved	Must be kept at reset value.

21.4.3. Control register 2 (LPUART_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								UCESM	WUIE	WUM[1:0]		Reserved			
								rw	rw	rw					

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DEP	DEM	DDRE	OVRD	Reserved	CTSIE	CTSEN	RTSEN	DENT	DENR	Reserved		HDEN	Reserved		ERRIE
rw	rw	rw	rw		rw	rw	rw	rw	rw			rw			rw

Bits	Fields	Descriptions
31:24	Reserved	Must be kept at reset value.
23	UCESM	LPUART clock enable in Deep-sleep mode 0: LPUART clock is disable in Deep-sleep mode 1: LPUART clock is enable in Deep-sleep mode
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
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 LPUART_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 LPUART is enabled (UEN=1).
19:16	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 LPUART 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 LPUART 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. 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 LPUART 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 LPUART_RDATA register. This bit field cannot be written when the LPUART is enabled (UEN=1).
11	Reserved	Must be kept at reset value
10	CTSIE	CTS interrupt enable 0: CTS interrupt is disabled 1: An interrupt will occur whenever the CTS bit is set in LPUART_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 LPUART 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 LPUART 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
5:4	Reserved	Must be kept at reset value.
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 LPUART is enabled (UEN=1).
2:1	Reserved	Must be kept at reset value.
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 LPUART_STAT in multibuffer communication

21.4.4. Baud rate generator register (LPUART_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 LPUART is enabled (UEN=1).

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

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

Bits	Fields	Descriptions
31:20	Reserved	Must be kept at reset value.
19:0	BRR[19:0]	The value of LPUARTDIV **Note:** BRR[19:0] ≥ 0x300 and (3 x baudrate) ≤ LPUCLK ≤ (4096 x baudrate)

21.4.5. Command register (LPUART_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	RXFCMD	MMCMD	Reserved
	w	w	

Bits	Fields	Descriptions
31:4	Reserved	Must be kept at reset value.
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 LPUART into mute mode and sets the RWU flag.
1:0	Reserved	Must be kept at reset value.

21.4.6. Status register (LPUART_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	Reserved	AMF	BSY
									r	r	r	r		r	r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved					CTS	CTSF	Reserved	TBE	TC	RBNE	IDLEF	ORERR	NERR	FERR	PERR
					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 LPUART core logic. 0: The LPUART core receiving logic has not been enabled 1: The LPUART 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 LPUART core logic. 0: The LPUART core transmitting logic has not been enabled 1: The LPUART 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

		LPUART_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 LPUART_INTC register. This bit can also be cleared when UESM is cleared.
19	RWU	Receiver wakeup from mute mode This bit is used to indicate if the LPUART 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 WM bit in the LPUART_CTL0 register. This bit can only be set by writing 1 to the MMCMD bit in the LPUART_CMD register when wakeup on IDLEIE mode is selected.
18	Reserved	Must be kept at reset value.
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 LPUART_CTL0 register. Set by hardware, when the character defined by ADDR [7:0] is received. Cleared by writing 1 to the AMC in the LPUART_INTC register.
16	BSY	Busy flag 0: LPUART reception path is idle 1: LPUART reception path is working
15:11	Reserved	Must be kept at reset value
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 LPUART_CTL2 Set by hardware when the nCTS input toggles. Cleared by writing 1 to CTSC bit in LPUART_INTC register.
8	Reserved	Must be kept at reset value
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 LPUART_CTL0

		Set by hardware when the content of the LPUART_TDATA register has been transferred into the transmit shift register. Cleared by a write to the LPUART_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 LPUART_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 LPUART_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 LPUART_CTL0. Set by hardware when the content of the receive shift register has been transferred to the LPUART_RDATA. Cleared by reading the LPUART_RDATA or writing 1 to RXFCMD bit of the LPUART_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 LPUART_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 LPUART_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 LPUART_CTL0. In multibuffer communication, an interrupt will occur if the ERRIE bit is set in LPUART_CTL2. Set by hardware when the word in the receive shift register is ready to be transferred into the LPUART_RDATA register while the RBNE bit is set. Cleared by writing 1 to OREC bit in LPUART_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 LPUART_CTL2. Set by hardware when noise error is detected on a received frame. Cleared by writing 1 to NEC bit in LPUART_INTC register.
1	FERR	Frame error flag 0: No framing error is detected

		1: Frame error flag is detected. An interrupt will occur if the ERRIE bit is set in LPUART_CTL2. Set by hardware when a de-synchronization, excessive noise is detected. This bit will be set. Cleared by writing 1 to FEC bit in LPUART_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 LPUART_CTL0. Set by hardware when a parity error occurs in receiver mode. Cleared by writing 1 to PEC bit in LPUART_INTC register.

21.4.7. Interrupt status clear register (LPUART_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						CTSC	Reserved		TCC	Reserved	IDLEC	OREC	NEC	FEC	PEC
						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 LPUART_STAT register.
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 LPUART_STAT register.
16:10	Reserved	Must be kept at reset value
9	CTSC	CTS change clear Writing 1 to this bit clears the CTSF bit in the LPUART_STAT register.
8:7	Reserved	Must be kept at reset value
6	TCC	Transmission complete clear Writing 1 to this bit clears the TC bit in the LPUART_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 LPUART_STAT register.
3	OREC	Overrun error clear Writing 1 to this bit clears the ORERR bit in the LPUART_STAT register.
2	NEC	Noise detected clear Writing 1 to this bit clears the NERR bit in the LPUART_STAT register.
1	FEC	Frame error flag clear Writing 1 to this bit clears the FERR bit in the LPUART_STAT register
0	PEC	Parity error clear Writing 1 to this bit clears the PERR bit in the LPUART_STAT register.

21.4.8. Receive data register (LPUART_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.
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 LPUART_CTL0 register).

21.4.9. Transmit data register (LPUART_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 LPUART_CTL0 register). This register must be written only when TBE bit in LPUART_STAT register is set.

21.4.10. Coherence control register (LPUART_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.

22. Inter-integrated circuit interface (I2C)

22.1. 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 implements standard I2C protocol with standard mode, fast mode and fast mode plus as well as CRC calculation and checking, 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.

22.2. 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) and fast mode plus (up to 1MHz, high current capability I/O must be enabled in SYSCFG_CFG0).
- Configurable SCL stretching in slave mode.
- Supports DMA mode.
- SMBus 3.0 and PMBus 1.3 compatible.
- Optional PEC (packet error checking) generation and check.
- Programmable analog and digital noise filters.
- Wakeup from Deep-sleep mode, Deep-sleep 1 mode and Deep-sleep 2 mode on I2C address match.
- Independent clock from PCLK.

22.3. Function overview

Figure 22-1. I2C module block diagram below provides details on the internal configuration of the I2C interface.

Figure 22-1. I2C module block diagram

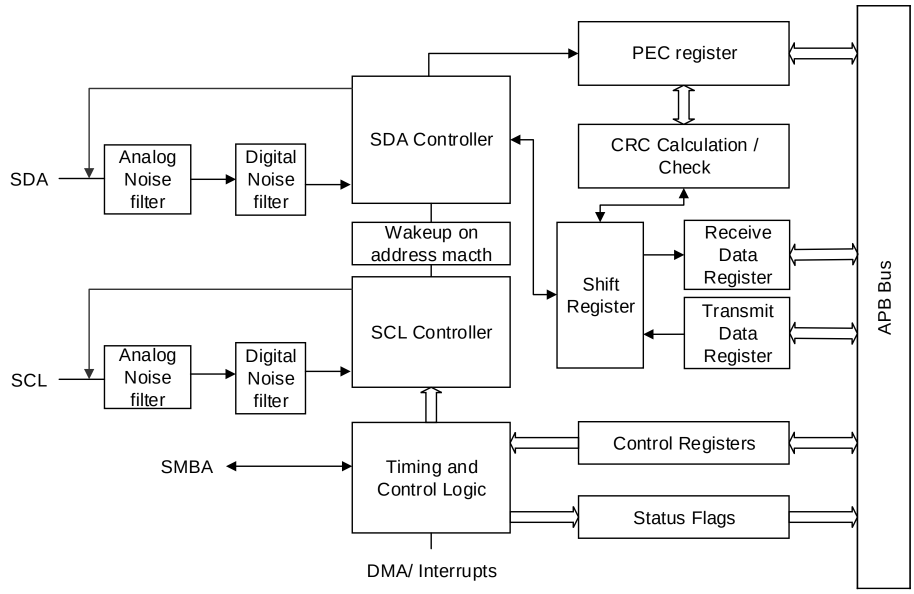


Table 22-1. Definition of I2C-bus terminology (refer to the I2C specification of Philips semiconductors)

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

22.3.1. 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:

- PCLK1: APB1 clock (default value)
- IRC16M: internal 16 MHz RC
- SYSCLK: 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 200ns for GD32L233xx devices and 125ns for GD32L235xx devices. 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}.

22.3.2. I2C communication flow

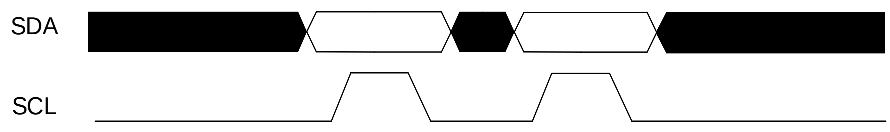
An I2C device is able to transmit or receive data whether it's a master or a slave, thus, there're 4 operation modes for an I2C device:

- 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 22-2. Data validation***). One clock pulse is generated for each data bit transferred.

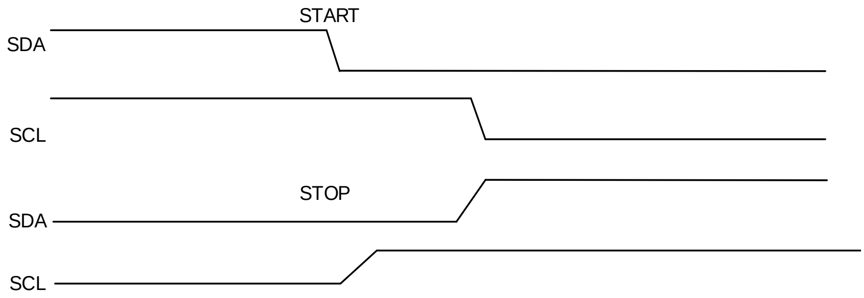
Figure 22-2. Data validation



START and STOP signal

All transactions begin with a START and are terminated by a STOP (see ***Figure 22-3. 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 22-3. 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 22-4. I2C communication flow with 10-bit address (Master Transmit)

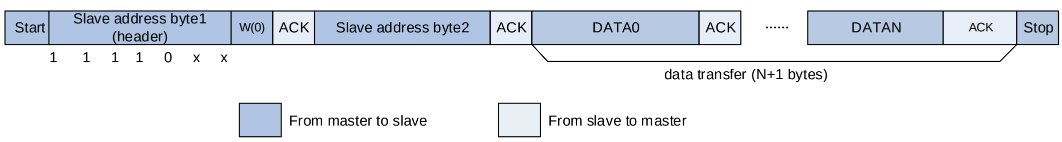


Figure 22-5. I2C communication flow with 7-bit address (Master Transmit)

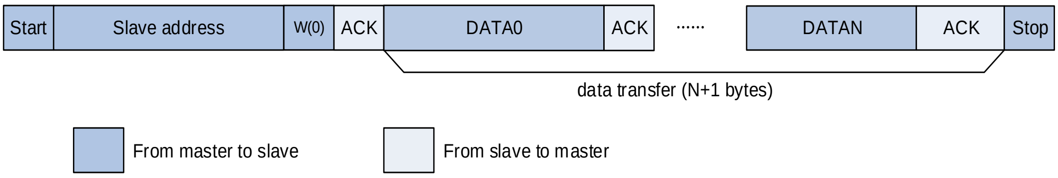
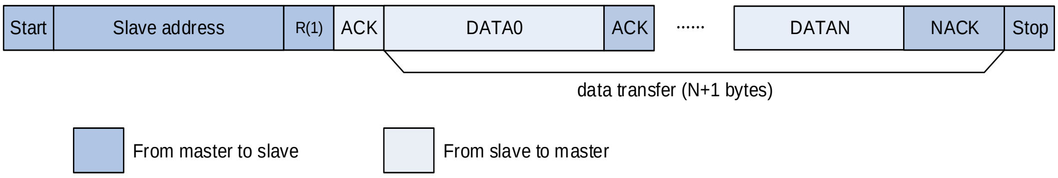


Figure 22-6. I2C communication flow with 7-bit address (Master Receive)



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 22-7. 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 22-8. I2C communication flow with 10-bit address (Master Receive when HEAD10R=1)***.

Figure 22-7. I2C communication flow with 10-bit address (Master Receive when HEAD10R=0)

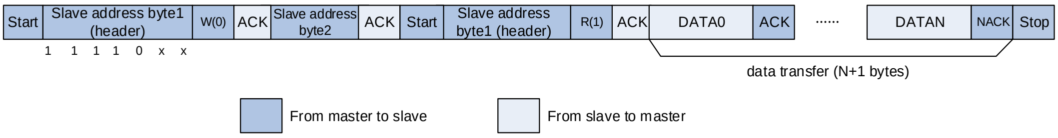
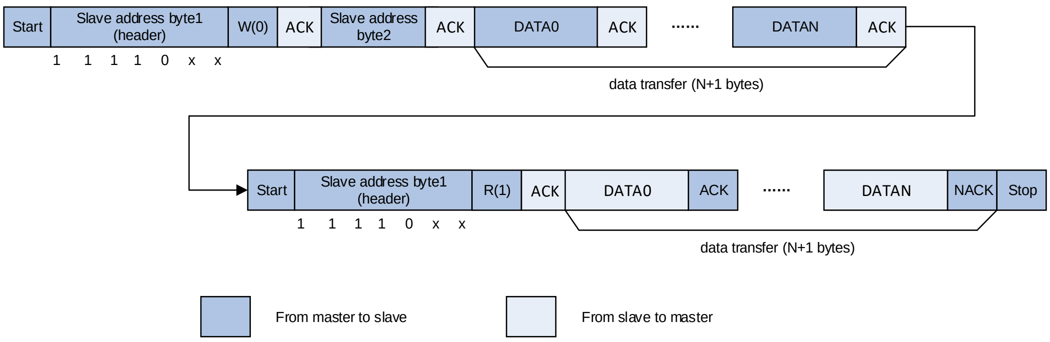


Figure 22-8. I2C communication flow with 10-bit address (Master Receive when HEAD10R=1)



22.3.3. 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.

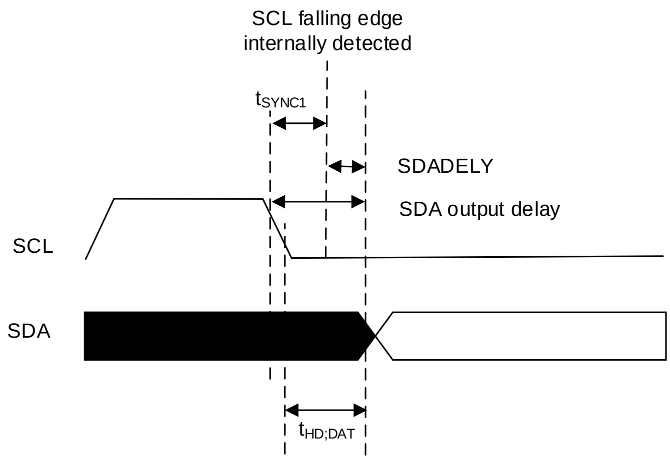
The digital noise filter can be used by configuring the DNF[3:0] bit in I2C_CTL0 register. The level of the SCL or the SDA will be changed if the level is stable for more than DNF[3:0]×t_{I2CCLK}. The length of spikes to be suppressed is configured by DNF[3:0].

22.3.4. 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.

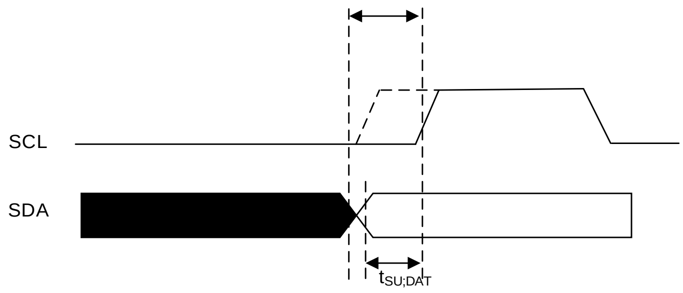
If the data is already available in I2C_TDATA register, the data will be sent on SDA after the SDADELY delay. As is shown in ***Figure 22-9. Data hold time***.

Figure 22-9. Data hold time



The SCLDELY counter starts when the data is sent on SDA output. As is shown in ***Figure 22-10. Data setup time***.

Figure 22-10. 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 22-2. 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	

22.3.5. 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:

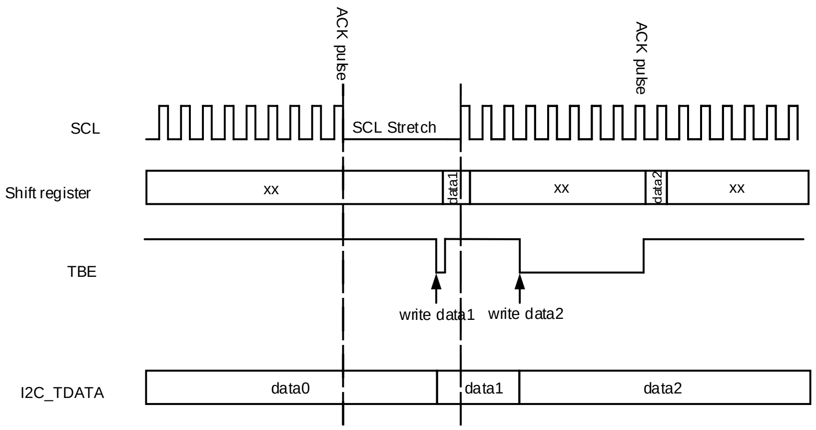
- Write I2CEN = 0
- Check I2CEN = 0
- Write I2CEN = 1

22.3.6. 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.

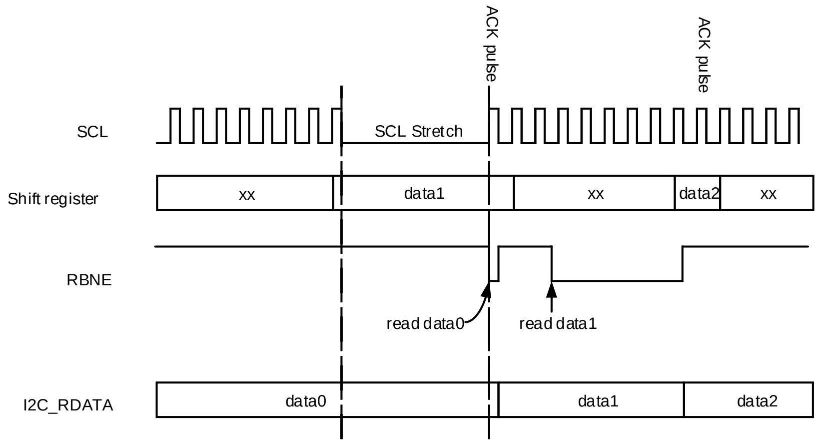
Figure 22-11. Data transmission



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 22-12. 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 22-3. Communication modes to be shut down***, the I2C embedded a byte counter in the hardware.

Table 22-3. 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.

22.3.7. 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:

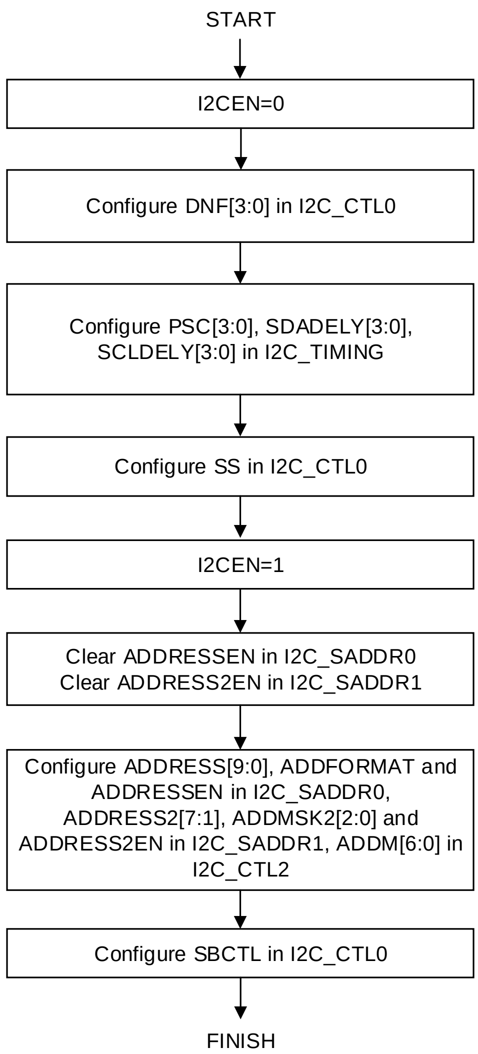
1. I2CEN=0.

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 22-13. I2C initialization in slave mode



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 22-14. Programming model for slave transmitting when SS=0

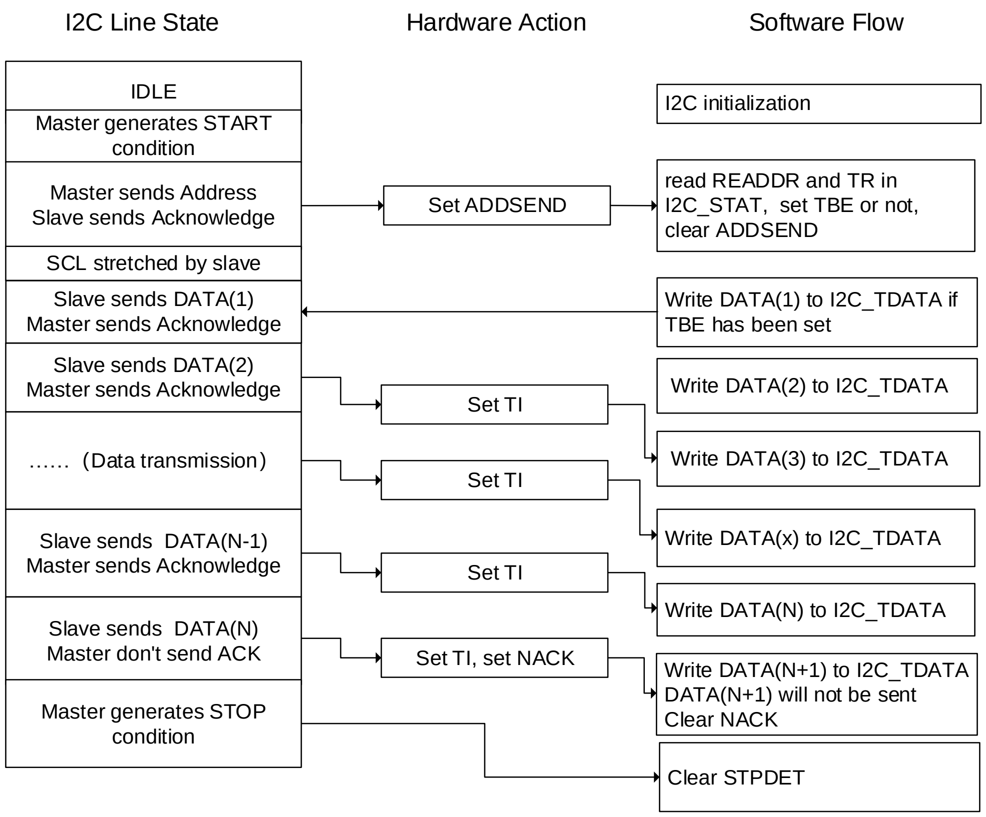
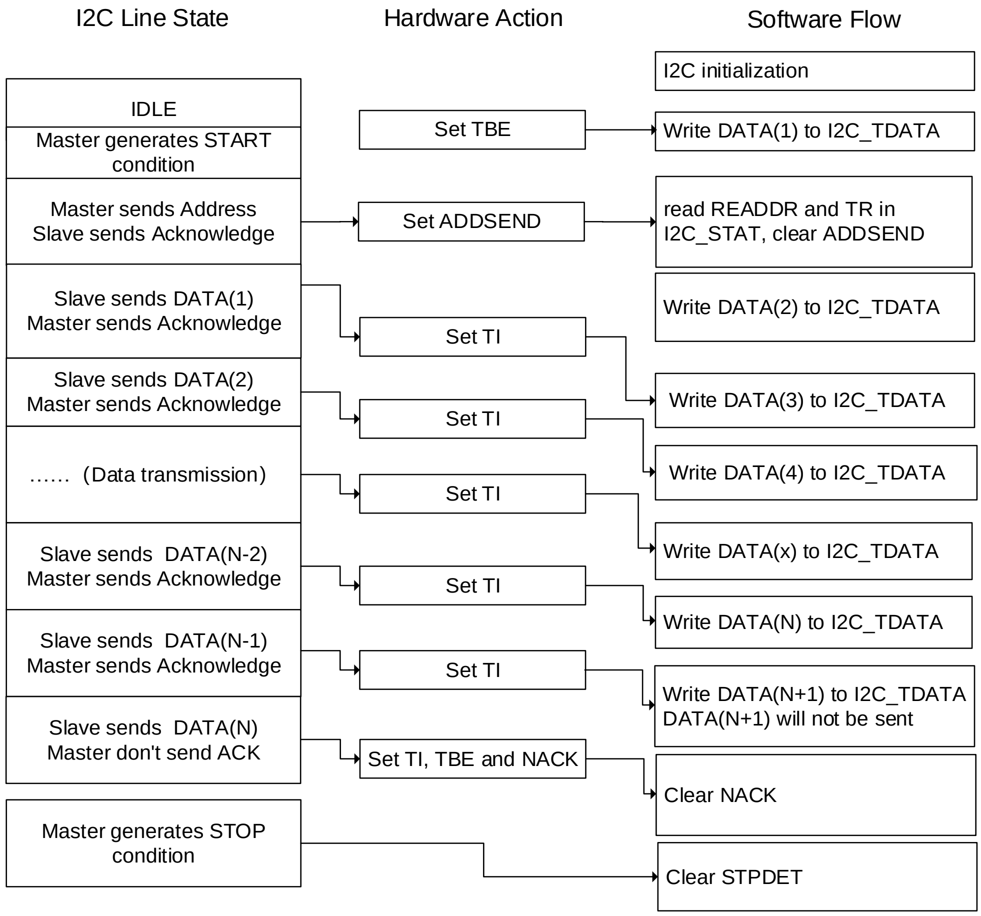


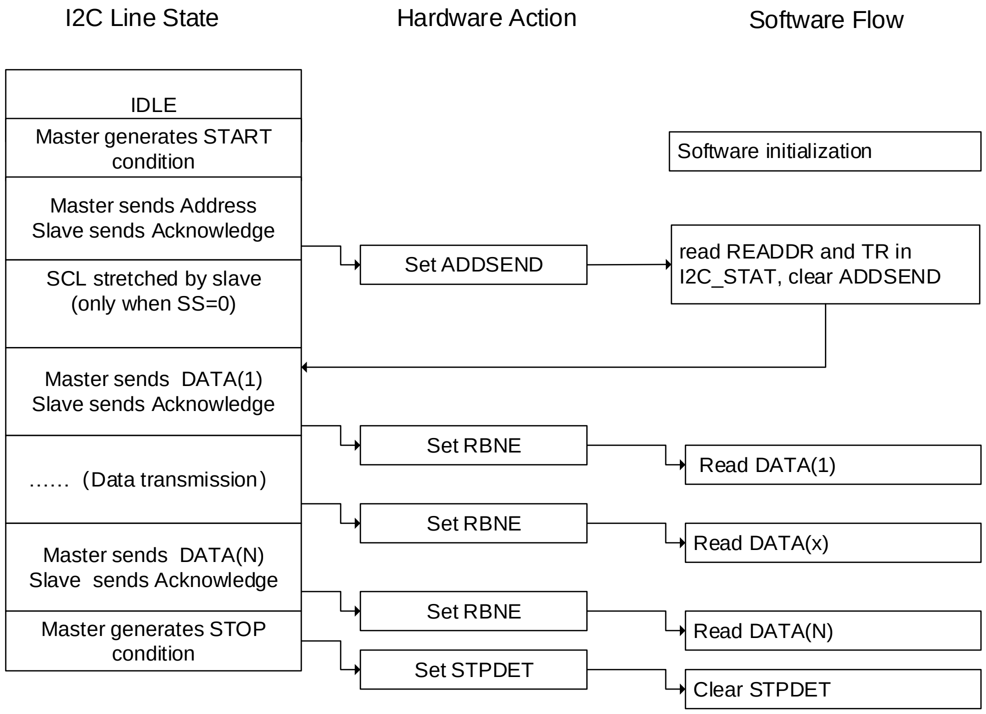
Figure 22-15. Programming model for slave transmitting when SS=1



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 22-16. Programming model for slave receiving



22.3.8. 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:

$t_{SCL}=t_{SYNC1}+t_{SYNC2}+\{[(SCLH[7:0]+1)+(SLLL[7:0]+1)]*(PSC+1)*t_{I2CCLK}\}$.

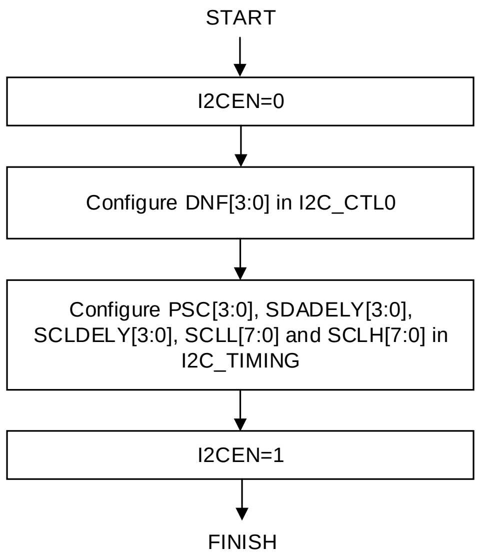
The t_{SYNC1} depends on the SCL falling slope, delay by input analog and digital noise filter and SCL synchronization with I2CCLK clock, which generally 2 to 3 I2CCLK periods. The t_{SYNC2} depends on the SCL rising slope, delay by input analog and digital noise filter and SCL synchronization with I2CCLK 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 22-17. I2C initialization in master mode



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 22-18. Programming model for master transmitting (N<=255)

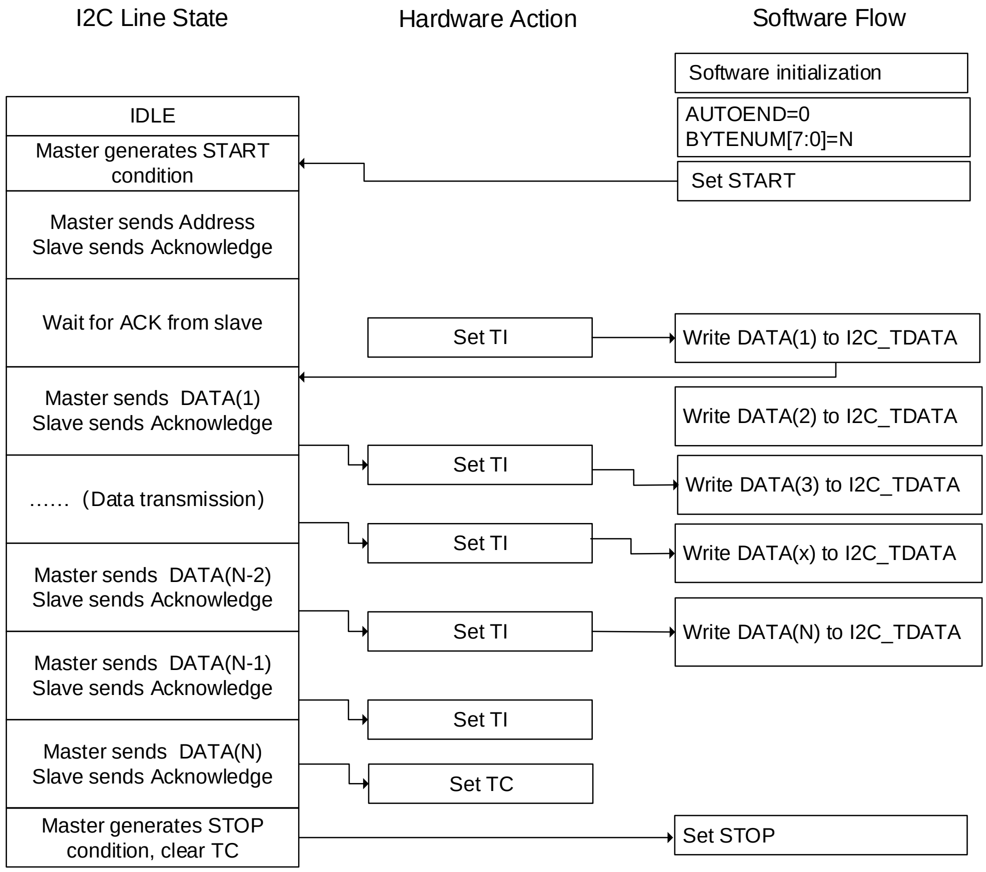
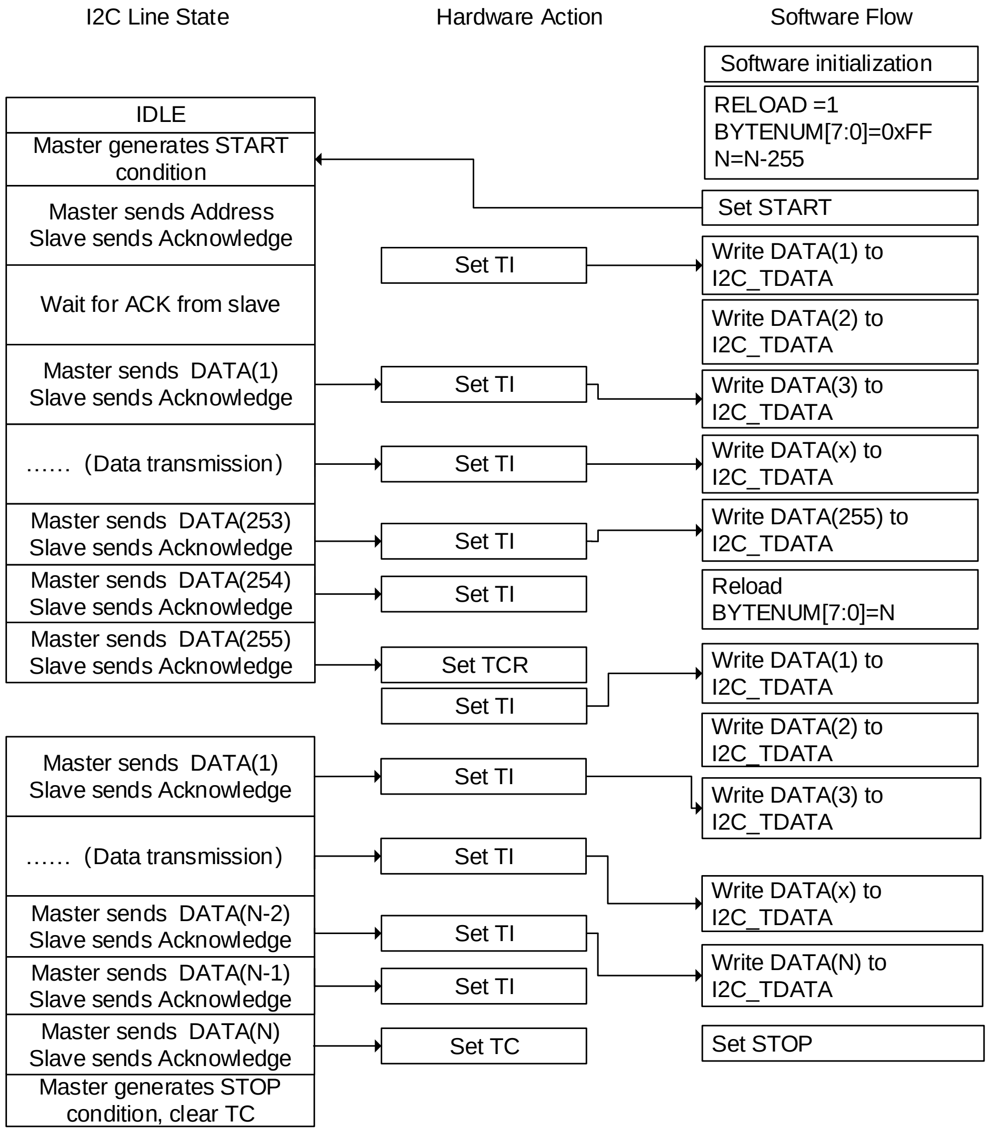


Figure 22-19. Programming model for master transmitting (N>255)



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 22-20. Programming model for master receiving (N<=255)

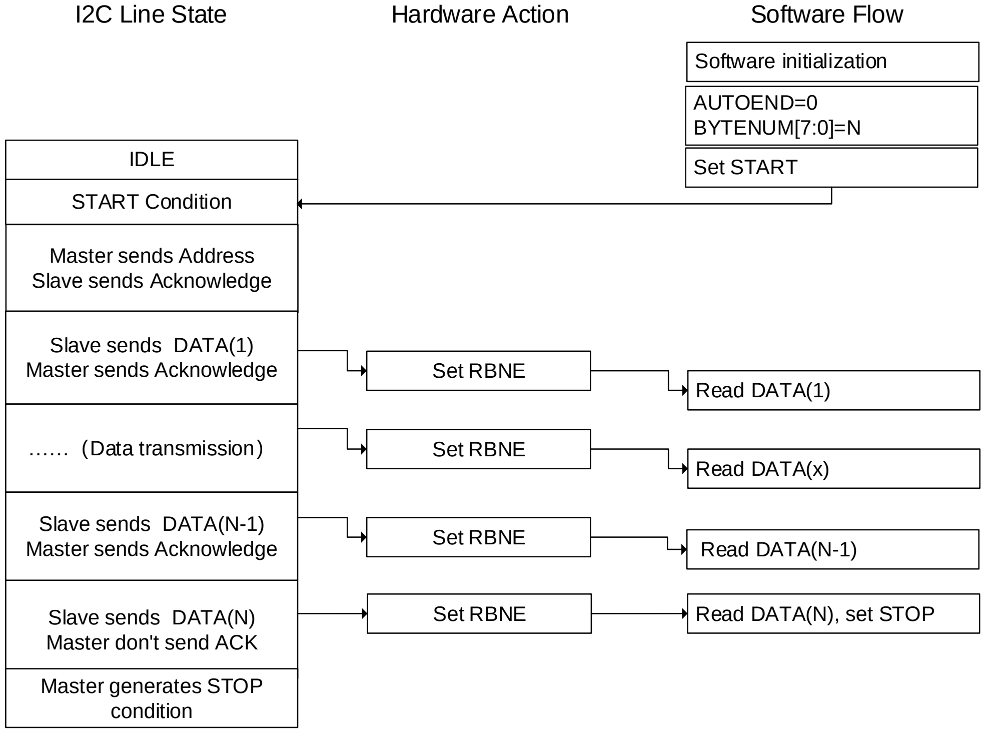
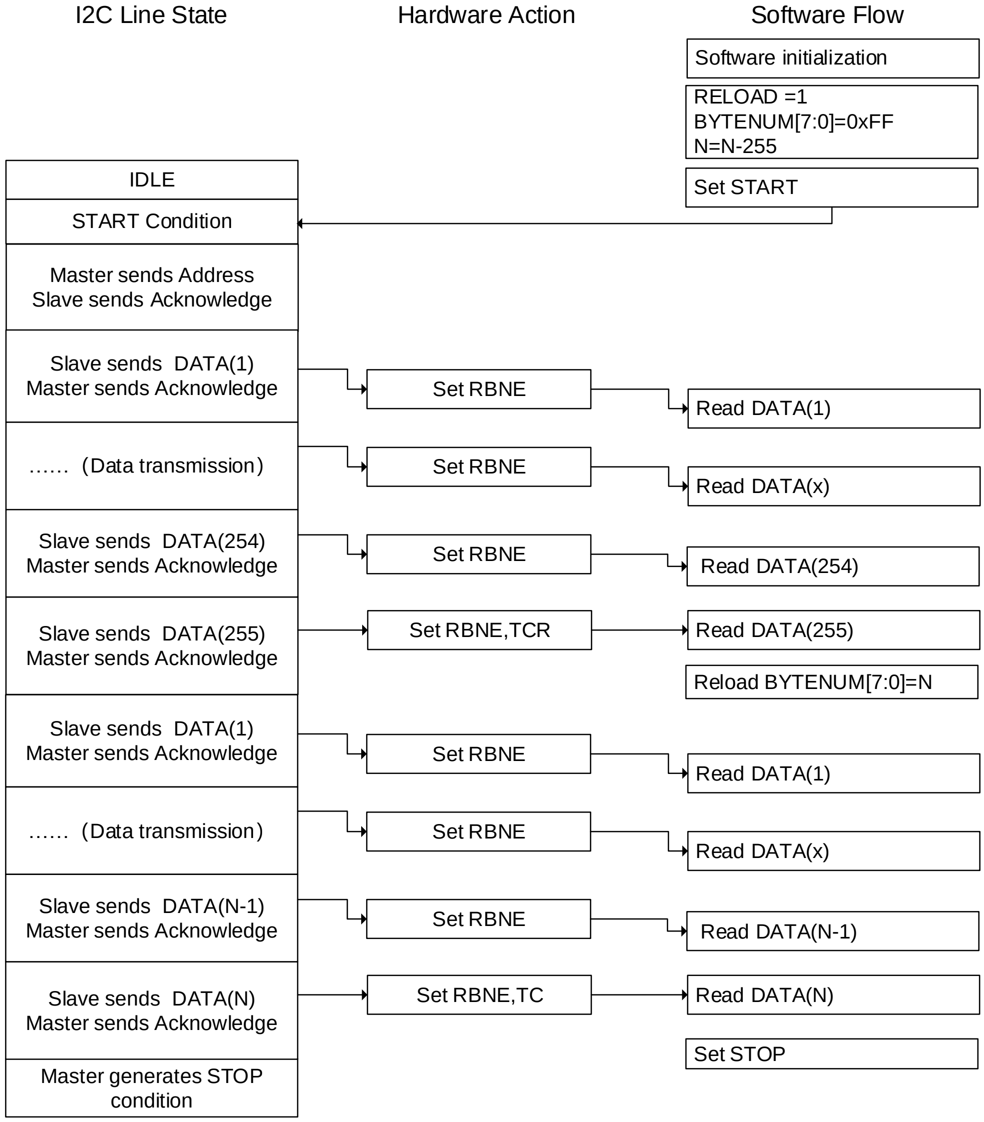


Figure 22-21. Programming model for master receiving (N>255)



22.3.9. 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.

22.3.10. SMBus mode

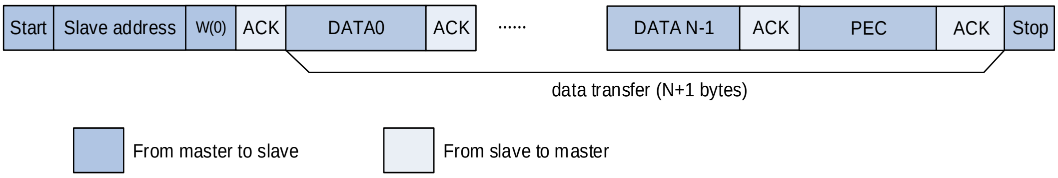
SMBus master transmitter and slave receiver

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 22-22. SMBus master transmitter and slave receiver communication flow



SMBus master receiver and slave transmitter

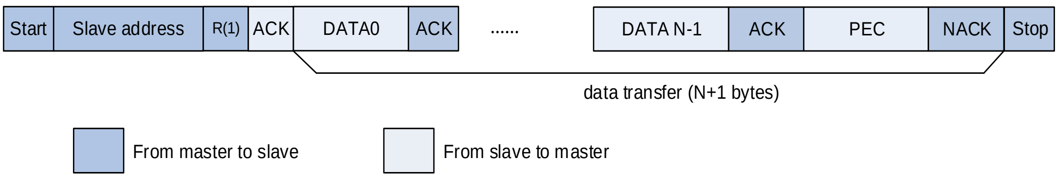
If the SMBus master is required to receive PEC at the end of bytes transfer, automatic end mode can be enabled. Before sending a START signal on the bus, PECTRANS bit must be set and slave addresses must be programmed. After receiving BYTENUM-1 data, the next received byte will be compared with the data 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 22-23. SMBus master receiver and slave transmitter communication flow



22.3.11. Wakeup from power saving modes

When the address of I2C matches correctly, it can wake up MCU from Deep-sleep mode, Deep-sleep 1 mode and Deep-sleep 2 mode. In order to wake up from these power saving modes, WUEN bit must be set in the I2C_CTL0 register and the IRC16M must be selected as the clock source for I2CCLK. During Deep-sleep mode, Deep-sleep 1 mode and Deep-sleep 2 mode, the IRC16M is switched off. The I2C interface switches the IRC16M on, and stretches SCL low until IRC16M is woken up when a START is detected. Then the IRC16M 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, IRC16M 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, IRC16M 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).

22.3.12. 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.

22.3.13. I2C error and interrupts

The I2C error flags are listed in ***Table 22-4. I2C error flags***.

Table 22-4. 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 22-5. I2C interrupt events***.

Table 22-5. I2C interrupt events

Interrupt event	Event flag	Enable control bit
I2C_RDATA is not empty during receiving	RBNE	RBNEIE
Transmit interrupt	TI	TIE
STOP signal detected in slave mode	STPDET	STPDETIE

Interrupt event	Event flag	Enable control bit
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	

22.3.14. 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.

22.4. Register definition

I2C0 base address: 0x4000 5400

I2C1 base address: 0x4000 5800

I2C2 base address: 0x4000 C000

22.4.1. 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
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)
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.
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.
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, Deep-sleep 1 mode and Deep-sleep 2 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. For I2C2, Deep-sleep / Deep-sleep1 / Deep-sleep2 wakeup are supported. For I2C0 and I2C1, Deep-sleep / Deep-sleep1 wakeup are supported.
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} 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

22.4.2. 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.
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. 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.

22.4.3. 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.

22.4.4. 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.

22.4.5. 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.

22.4.6. Timeout register (I2C_TIMEOUT)

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
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.

22.4.7. 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
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
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.
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

22.4.8. 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.
12	TIMEOUTC	TIMEOUT flag clear. Software can clear the TIMEOUT bit of I2C_STAT by writing 1 to this bit.
11	PECERRC	PEC error flag clear. Software can clear the PECERR bit of I2C_STAT by writing 1 to this bit.
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.

22.4.9. PEC register (I2C_PEC)

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								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.

22.4.10. 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

22.4.11. 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

22.4.12. Control register 2 (I2C_CTL2)

Address offset: 0x90

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
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.

23. Serial peripheral interface/Inter-IC sound (SPI/I2S)

23.1. 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 SPI0.

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.

23.2. Characteristics

23.2.1. SPI characteristics

- Master or slave operation with full-duplex or simplex mode.
- Separate transmit and receive buffer, 16 bits wide (only in SPI1).
- Separate transmission and reception 32-bit FIFO (only in SPI0).
- Data frame size can be 8 or 16 bits (only in SPI1).
- Data frame size can be 4 to 16 bits (only in SPI0).
- 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 SPI0).

23.2.2. I2S characteristics

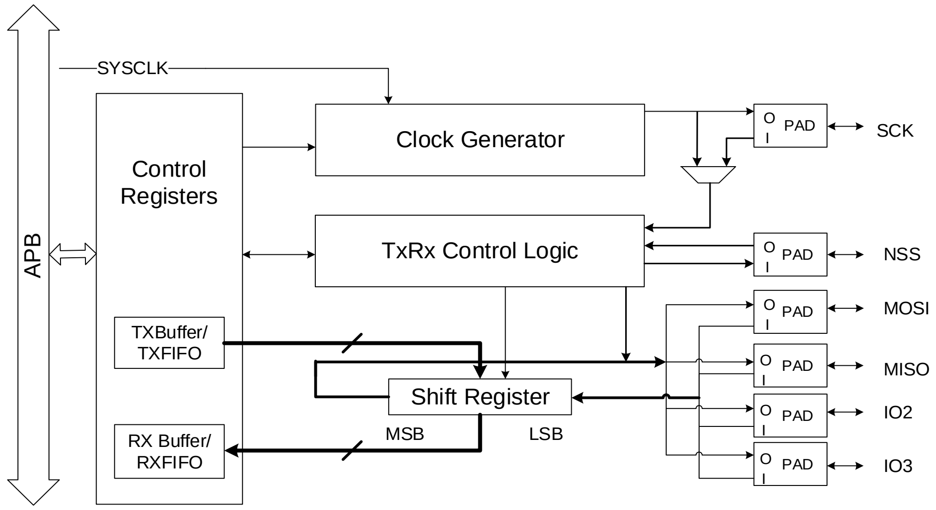
- Master or slave operation for transmission/reception.
- Four I2S standards supported: Phillips, MSB justified, LSB justified and PCM standard.
- Data length can be 16 bits, 24 bits or 32 bits.
- Channel length can be 16 bits or 32 bits.
- Transmission and reception using a 16 bits wide buffer.
- Audio sample frequency can be 8 kHz to 192 kHz using I2S clock divider.
- Programmable idle state clock polarity.
- Master clock (MCK) can be output.

- Transmission and reception using DMA.

23.3. SPI function overview

23.3.1. SPI block diagram

Figure 23-1. Block diagram of SPI



23.3.2. SPI signal description

Normal configuration (Not Quad-SPI Mode)

Table 23-1. 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

Pin name	Direction	Description
		output, suitable for single master application; when NSSDRV=0, it is NSS input, suitable for multi-master application. 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 SPI0). 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 23-2. 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

23.3.3. 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 SPI0 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 23-2. SPI0 timing diagram in normal mode

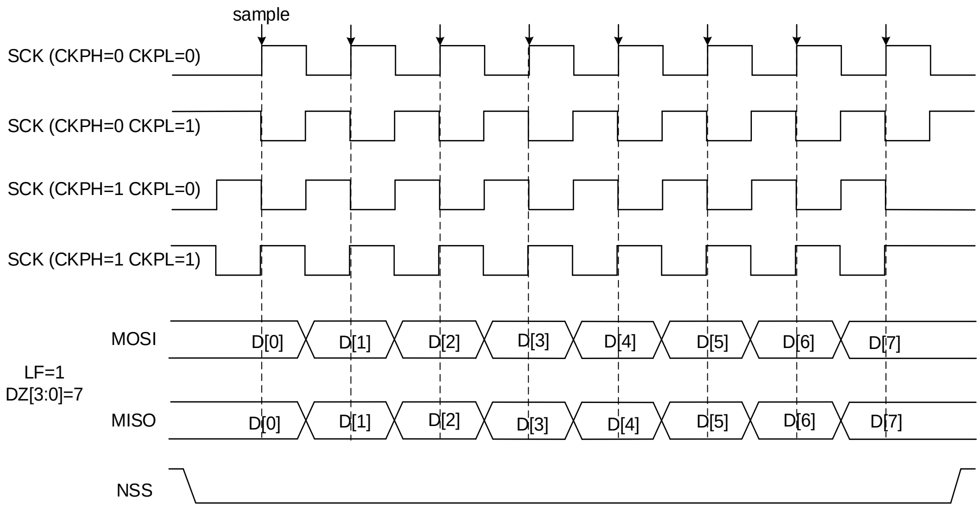
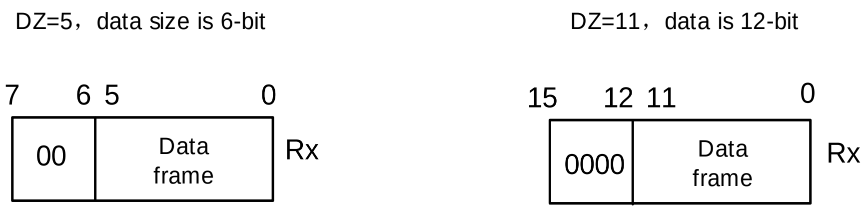


Figure 23-3. SPI0 data frame right-aligned diagram



In SPI1 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 SPI1 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 23-4. SPI1 timing diagram in normal mode

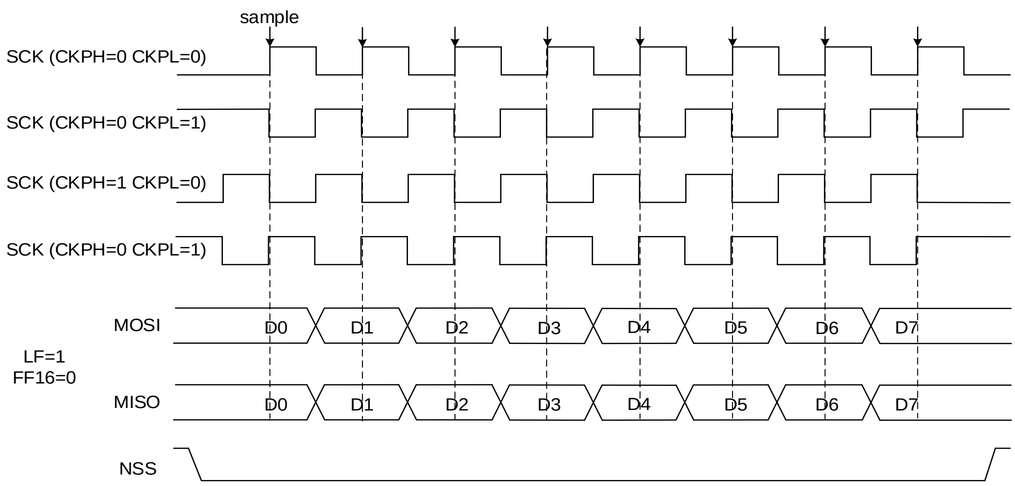
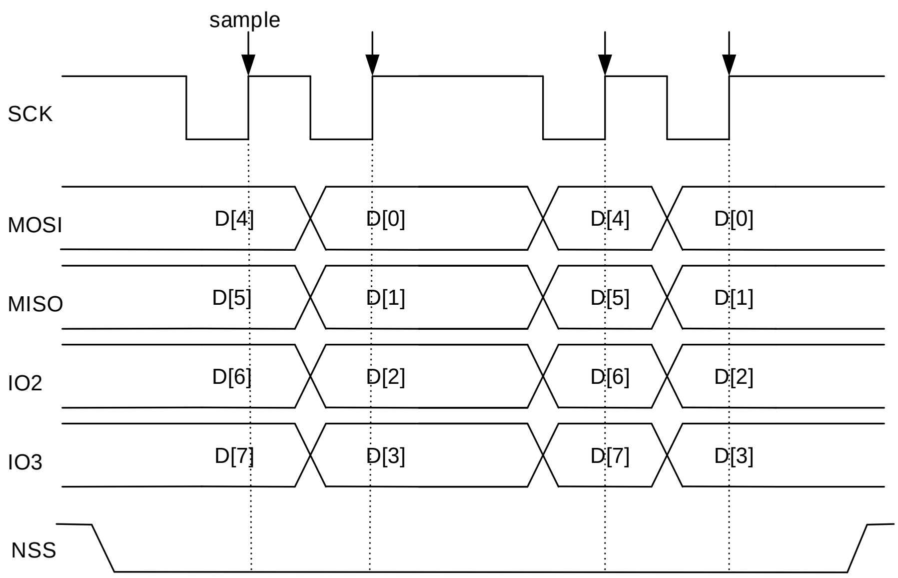


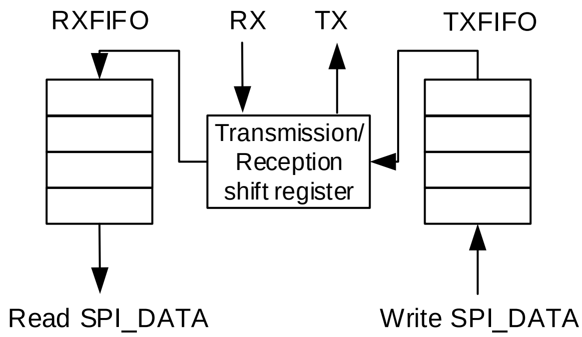
Figure 23-5. SPI0 timing diagram in Quad-SPI mode (CKPL=1, CKPH=1, LF=0)



23.3.4. 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 SPI0).

Figure 23-6. 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 SPI0, 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 SPI0, 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 SPI0)

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.

23.3.5. 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 23-3. 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.

Mode	Register configuration	Description
		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 23-4. 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.

23.3.6. SPI operation modes

Table 23-5. SPI operation modes

Mode	Description	Register configuration	Data pin usage
MFD	Master full-duplex	MSTMOD = 1 RO = 0 BDEN = 0 BDOEN: Don't care	MOSI: Transmission 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	MOSI: Not used MISO: Reception

Mode	Description	Register configuration	Data pin usage
		BDEN = 1 BDOEN = 0	

Figure 23-7. A typical full-duplex connection

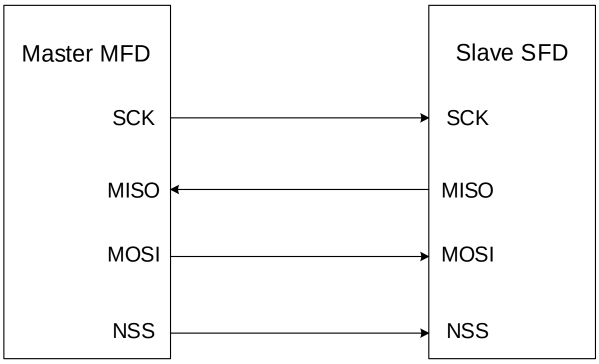


Figure 23-8. A typical simplex connection (Master: Receive, Slave: Transmit)

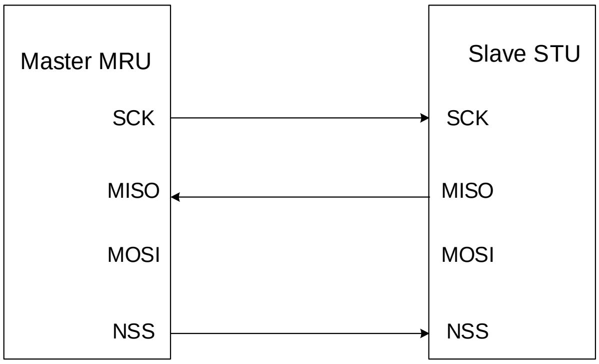


Figure 23-9. A typical simplex connection (Master: Transmit only, Slave: Receive)

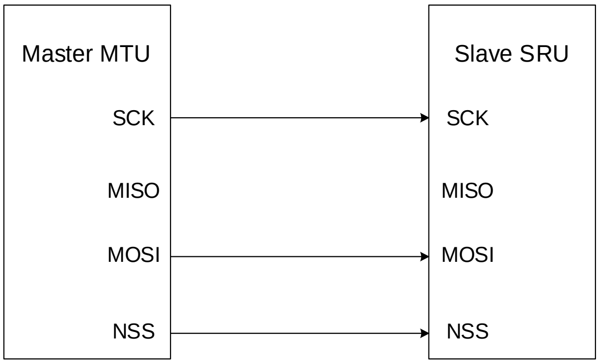
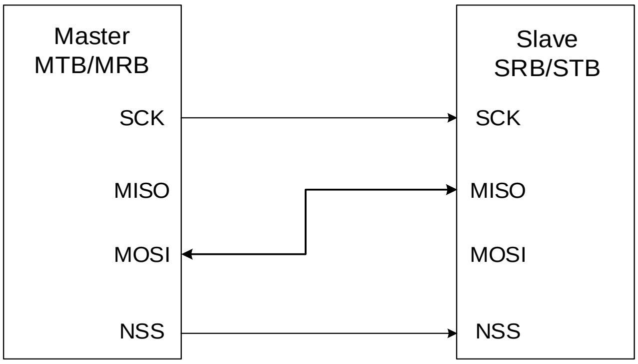


Figure 23-10. A typical bidirectional connection



Initialization sequence

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 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.

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 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 23-11. Timing diagram of TI master mode with discontinuous transfer

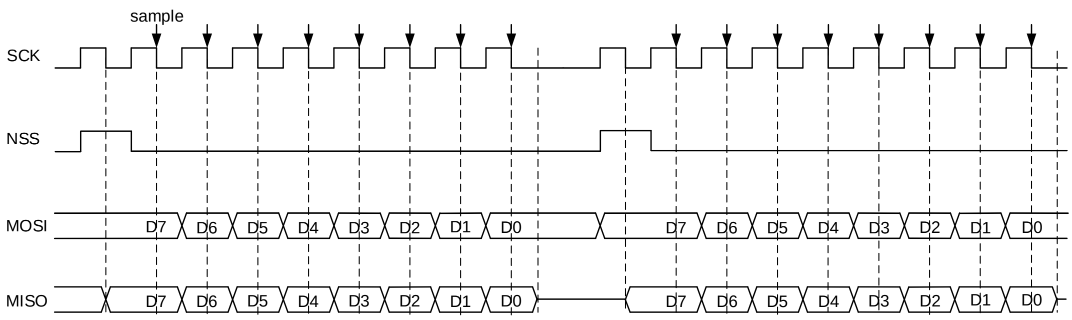
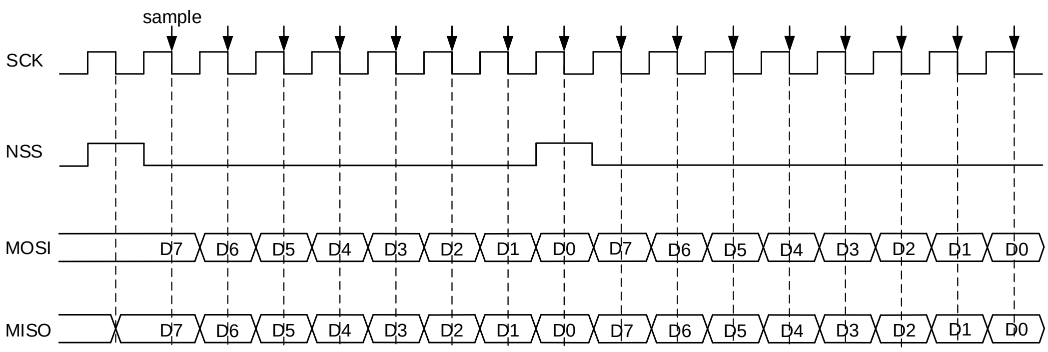
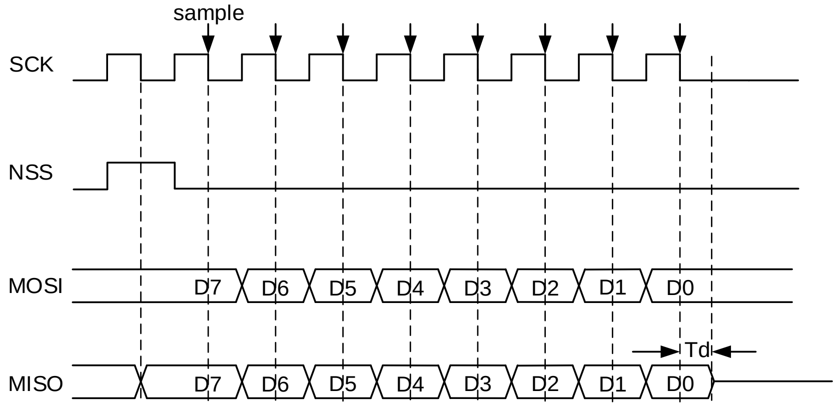


Figure 23-12. Timing diagram of TI master mode with continuous transfer



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 23-13. 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.

$$T_d=\frac{T_{bit}}{2}+5*T_{pclk} \tag{23-1}$$

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.

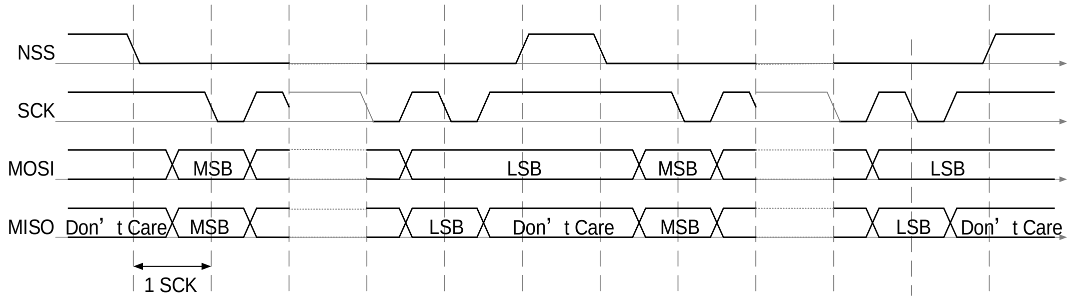
NSS pulse mode operation sequence

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 23-14. 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.

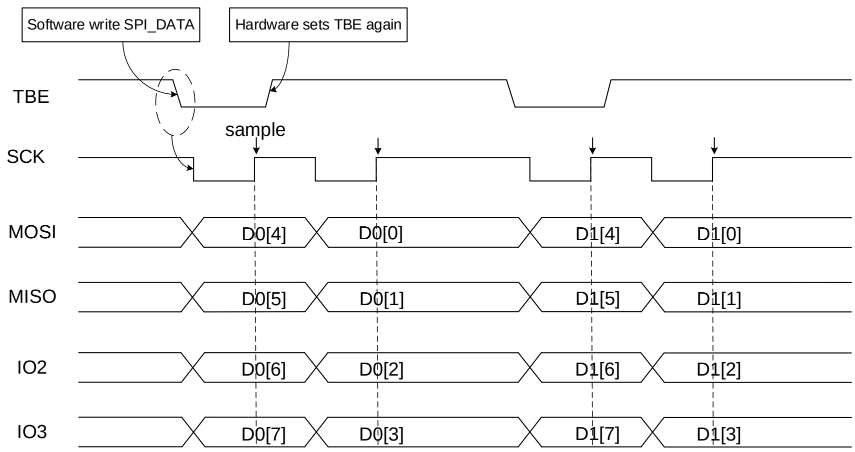
Quad write operation

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 23-15. Timing diagram of quad write operation in Quad-SPI mode



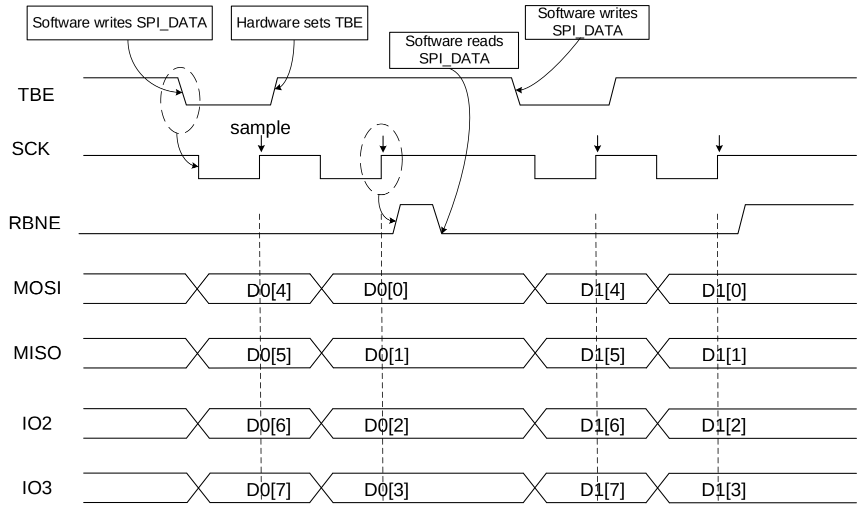
Quad read operation

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 23-16. 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 SPI0, 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 SPI1, 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 SPI0, wait until TXLVL[1:0]=00 and confirm TRANS=0. Then disable the SPI by clearing SPIEN bit.

For SPI1, 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 SPI0, 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 SPI1, 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 SPI0, 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 SPI1, 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.

23.3.7. 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 SPI0)

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

23.3.8. 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 SPI1, if DMA function is enabled, application doesn't need to operate CRCNT bit and hardware will automatically process the CRC transmitting and checking.

For SPI0, 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.

23.3.9. 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 SPI0, 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 SPI1, 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 SPI1, that means the last data has not been read out and the newly incoming data is received. For SPI0, 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 23-6. SPI interrupt requests

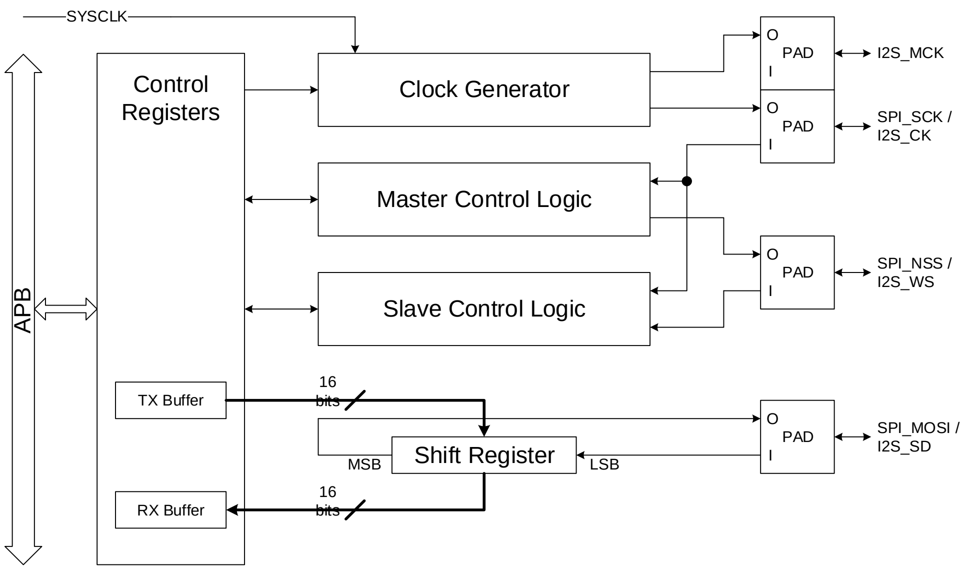
Flag	Description	Clear method	Interrupt enable bit
TBE	Transmit buffer/TXFIFO empty	Write SPI_DATA register.	TBEIE

Flag	Description	Clear method	Interrupt enable bit
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	

23.4. I2S function overview

23.4.1. I2S block diagram

Figure 23-17. 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.

23.4.2. I2S signal description

There are four pins on the I2S interface, including I2S_CK, I2S_WS, I2S_SD and I2S_MCK.

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. I2S_MCK is the master clock signal. It produces a frequency rate equal to 256 x Fs, and Fs is the audio sampling frequency.

23.4.3. 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 23-18. I2S Phillips standard timing diagram (DTLEN=00, CHLEN=0, CKPL=0)

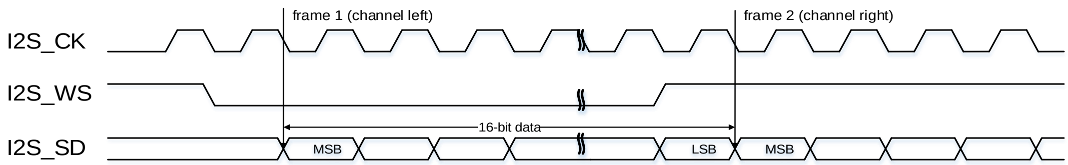
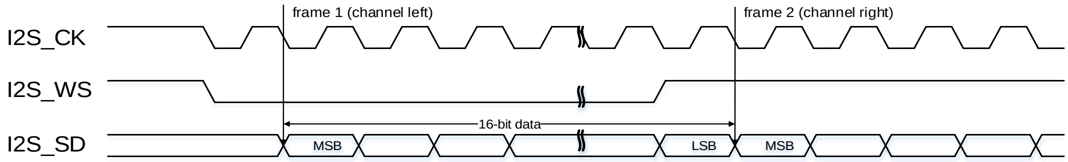


Figure 23-19. I2S Phillips standard timing diagram (DTLEN=00, CHLEN=0, CKPL=1)



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 23-20. I2S Phillips standard timing diagram (DTLEN=10, CHLEN=1, CKPL=0)

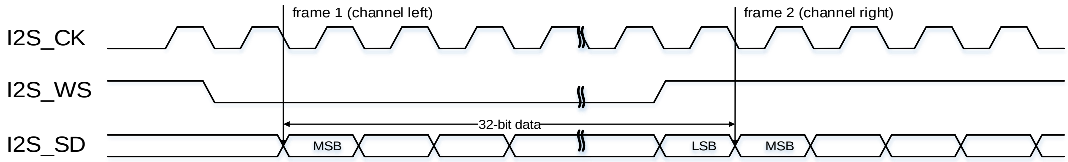
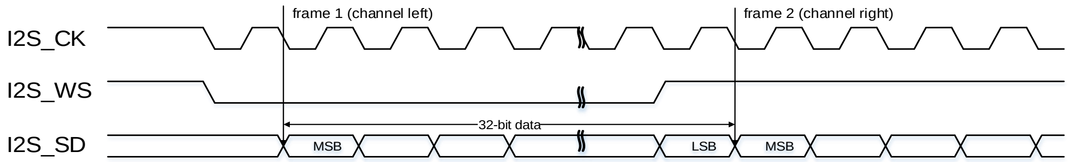


Figure 23-21. I2S Phillips standard timing diagram (DTLEN=10, CHLEN=1, CKPL=1)



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 23-22. I2S Phillips standard timing diagram (DTLEN=01, CHLEN=1, CKPL=0)

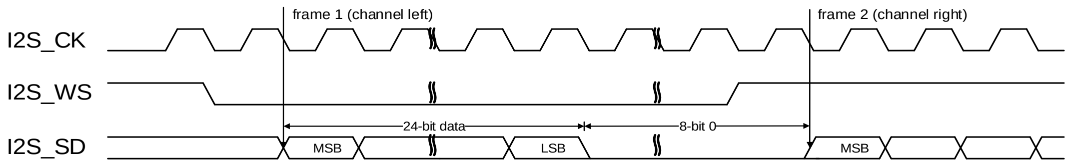
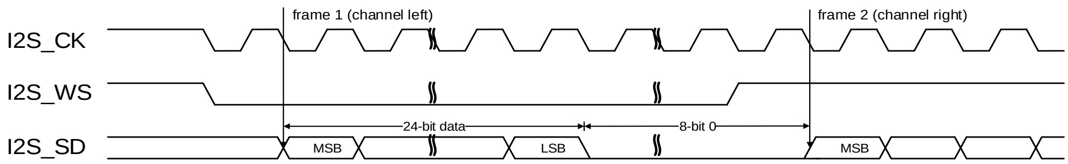


Figure 23-23. I2S Phillips standard timing diagram (DTLEN=01, CHLEN=1, CKPL=1)



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 23-24. I2S Phillips standard timing diagram (DTLEN=00, CHLEN=1, CKPL=0)

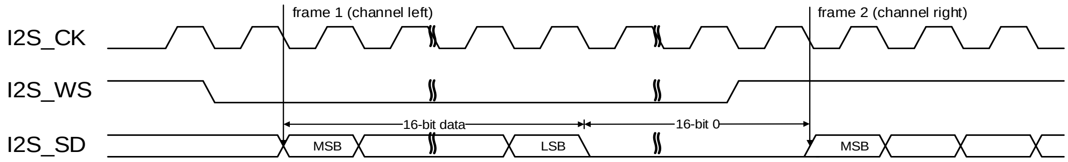
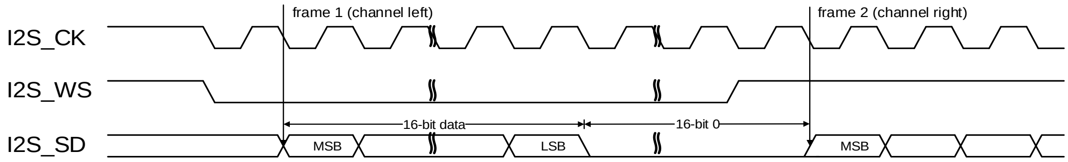


Figure 23-25. I2S Phillips standard timing diagram (DTLEN=00, CHLEN=1, CKPL=1)



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 23-26. MSB justified standard timing diagram (DTLEN=00, CHLEN=0, CKPL=0)

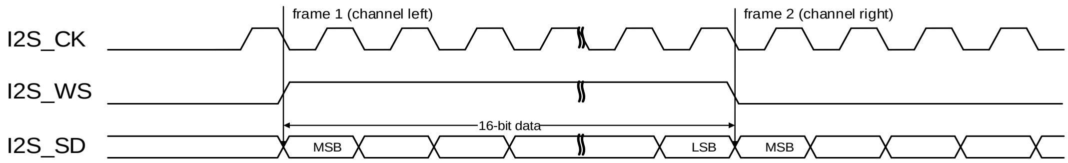


Figure 23-27. MSB justified standard timing diagram (DTLEN=00, CHLEN=0, CKPL=1)

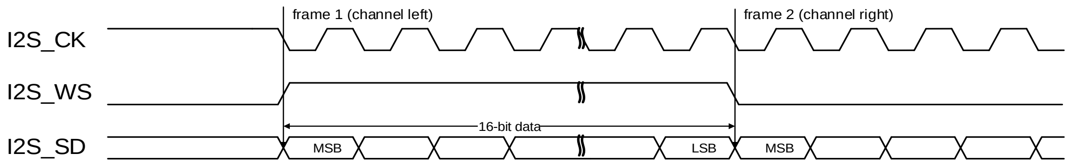


Figure 23-28. MSB justified standard timing diagram (DTLEN=10, CHLEN=1, CKPL=0)

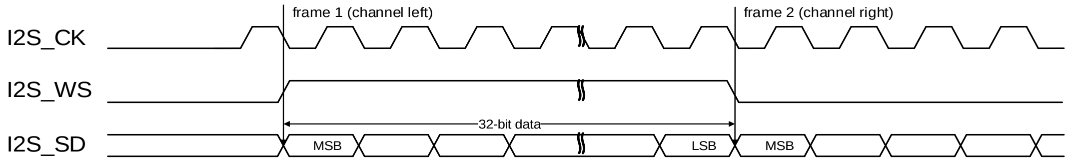


Figure 23-29. MSB justified standard timing diagram (DTLEN=10, CHLEN=1, CKPL=1)

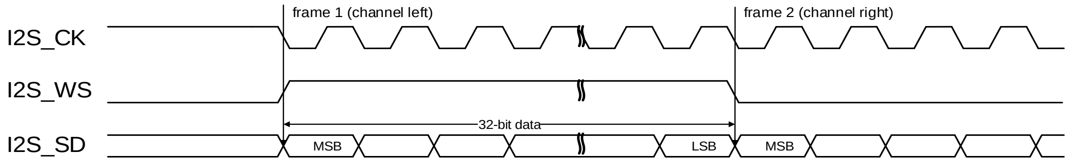


Figure 23-30. MSB justified standard timing diagram (DTLEN=01, CHLEN=1, CKPL=0)

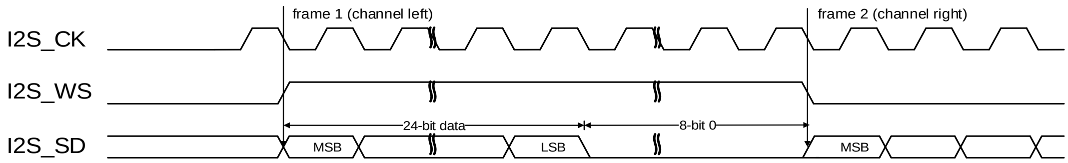


Figure 23-31. MSB justified standard timing diagram (DTLEN=01, CHLEN=1, CKPL=1)

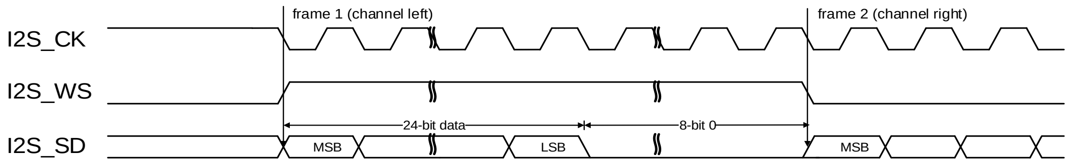


Figure 23-32. MSB justified standard timing diagram (DTLEN=00, CHLEN=1, CKPL=0)

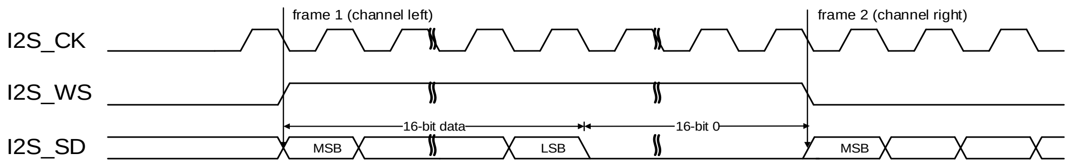
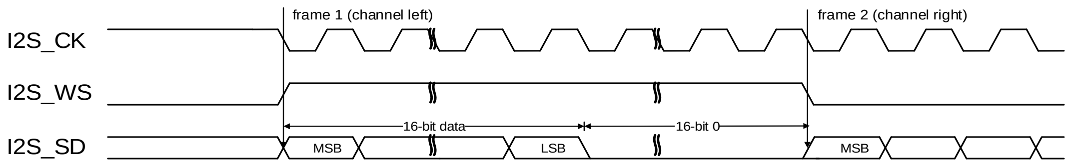


Figure 23-33. MSB justified standard timing diagram (DTLEN=00, CHLEN=1, CKPL=1)



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 23-34. LSB justified standard timing diagram (DTLEN=01, CHLEN=1, CKPL=0)

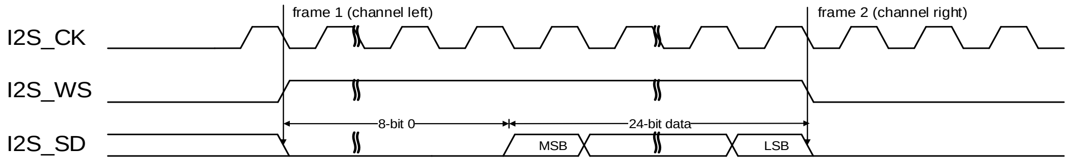
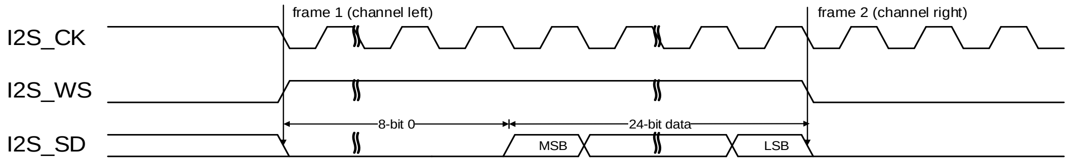


Figure 23-35. LSB justified standard timing diagram (DTLEN=01, CHLEN=1, CKPL=1)



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 23-36. LSB justified standard timing diagram (DTLEN=00, CHLEN=1, CKPL=0)

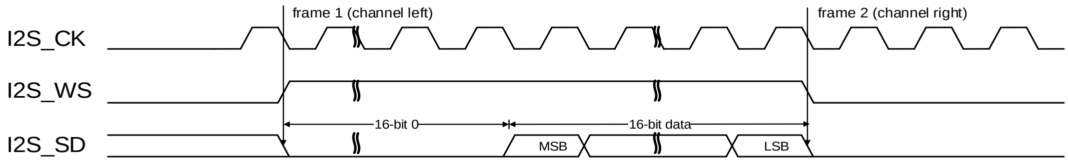
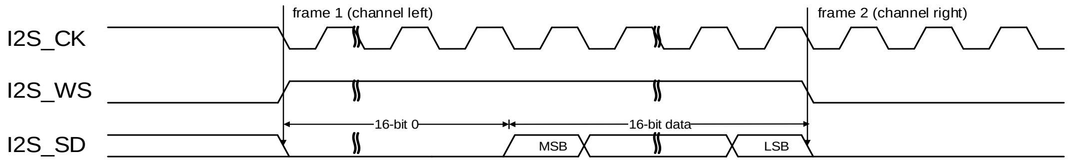


Figure 23-37. LSB justified standard timing diagram (DTLEN=00, CHLEN=1, CKPL=1)



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 23-38. PCM standard short frame synchronization mode timing diagram (DTLEN=00, CHLEN=0, CKPL=0)

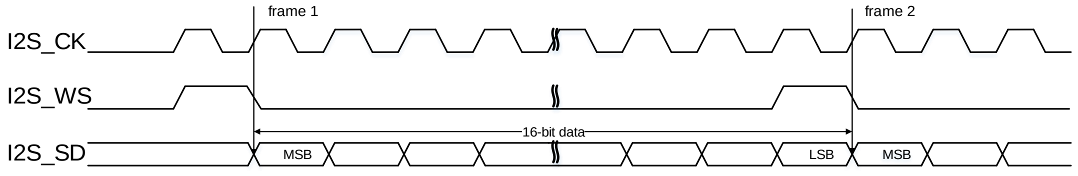


Figure 23-39. PCM standard short frame synchronization mode timing diagram (DTLEN=00, CHLEN=0, CKPL=1)

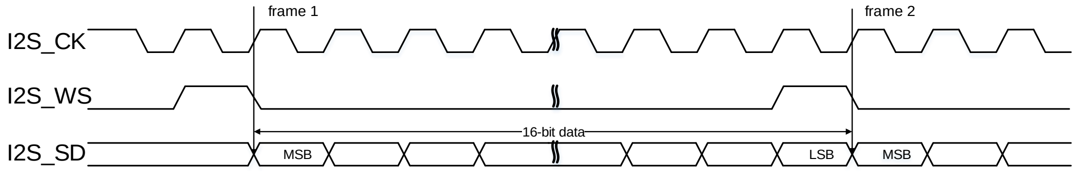


Figure 23-40. PCM standard short frame synchronization mode timing diagram (DTLEN=10, CHLEN=1, CKPL=0)

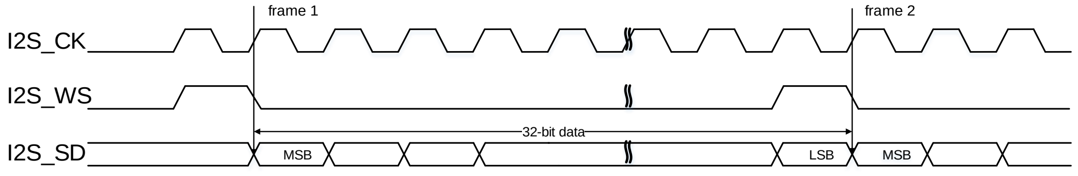


Figure 23-41. PCM standard short frame synchronization mode timing diagram (DTLEN=10, CHLEN=1, CKPL=1)

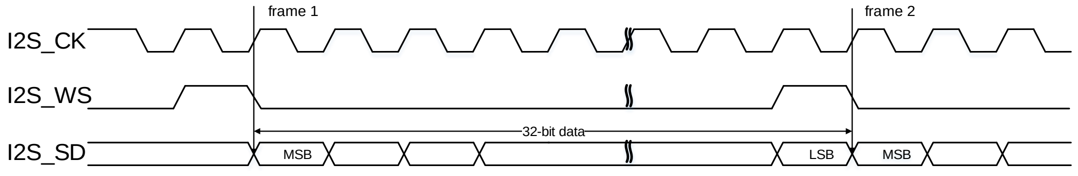


Figure 23-42. PCM standard short frame synchronization mode timing diagram

(DTLEN=01, CHLEN=1, CKPL=0)

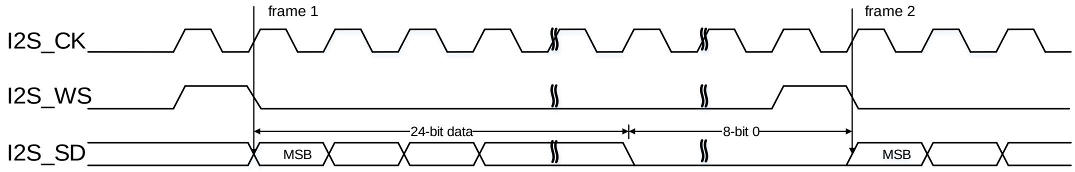


Figure 23-43. PCM standard short frame synchronization mode timing diagram (DTLEN=01, CHLEN=1, CKPL=1)

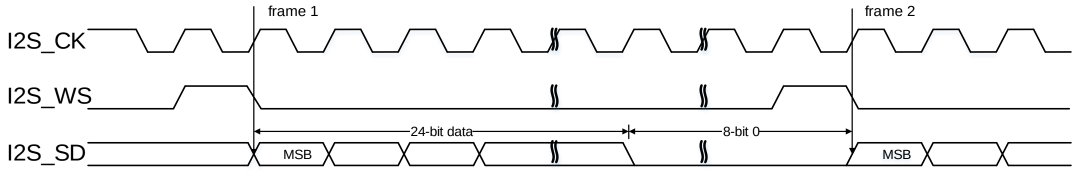


Figure 23-44. PCM standard short frame synchronization mode timing diagram (DTLEN=00, CHLEN=1, CKPL=0)

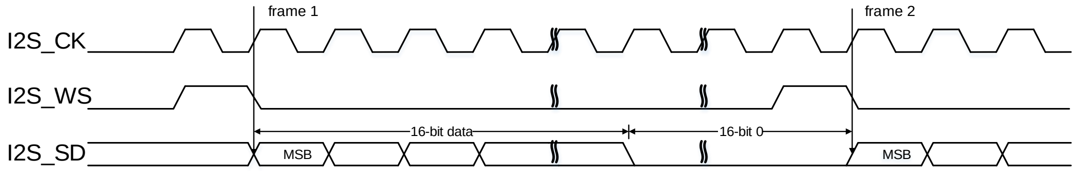
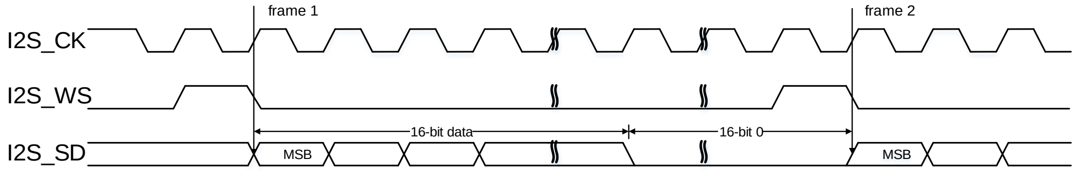


Figure 23-45. PCM standard short frame synchronization mode timing diagram (DTLEN=00, CHLEN=1, CKPL=1)



The timing diagrams for each configuration of the long frame synchronization mode are shown below.

Figure 23-46. PCM standard long frame synchronization mode timing diagram (DTLEN=00, CHLEN=0, CKPL=0)

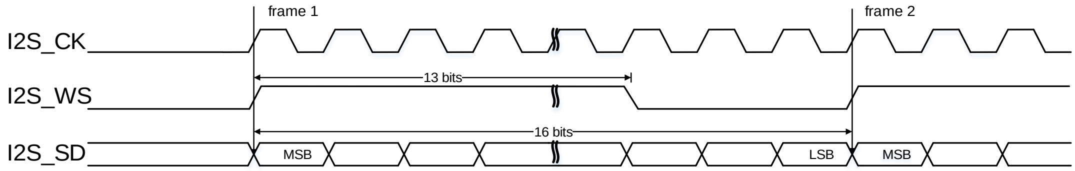


Figure 23-47. PCM standard long frame synchronization mode timing diagram

(DTLEN=00, CHLEN=0, CKPL=1)

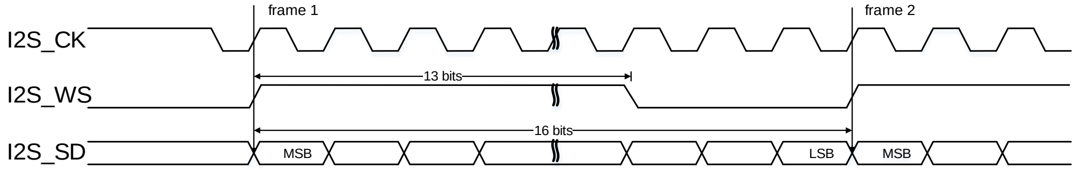


Figure 23-48. PCM standard long frame synchronization mode timing diagram (DTLEN=10, CHLEN=1, CKPL=0)

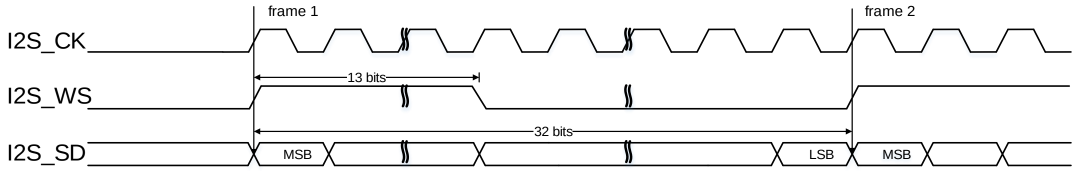


Figure 23-49. PCM standard long frame synchronization mode timing diagram (DTLEN=10, CHLEN=1, CKPL=1)

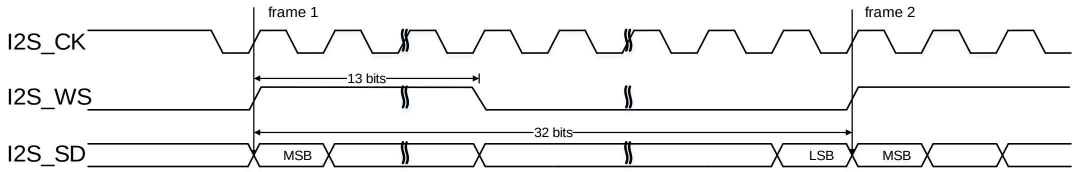


Figure 23-50. PCM standard long frame synchronization mode timing diagram (DTLEN=01, CHLEN=1, CKPL=0)

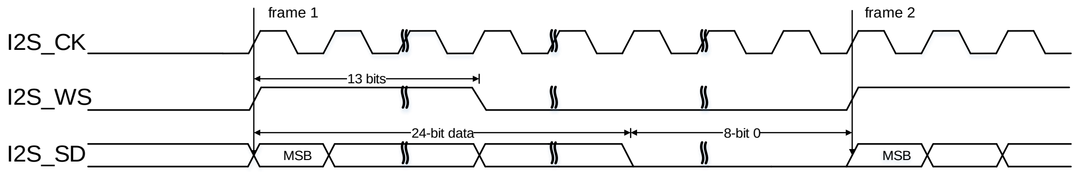


Figure 23-51. PCM standard long frame synchronization mode timing diagram (DTLEN=01, CHLEN=1, CKPL=1)

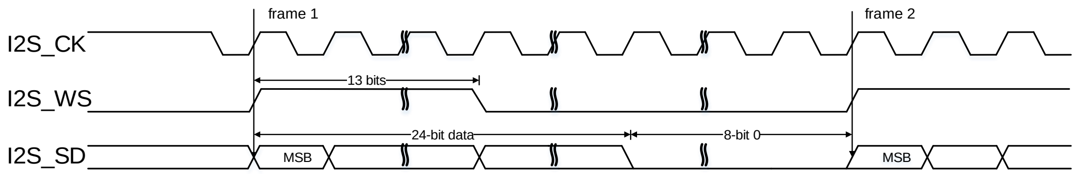


Figure 23-52. PCM standard long frame synchronization mode timing diagram (DTLEN=00, CHLEN=1, CKPL=0)

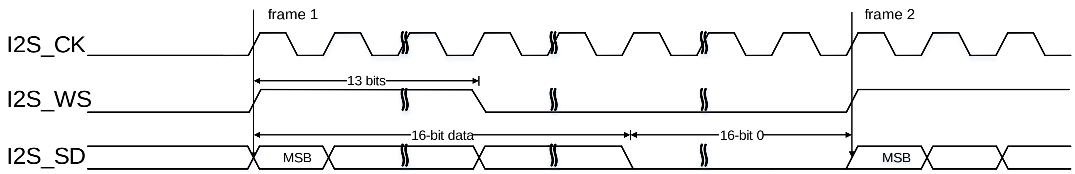
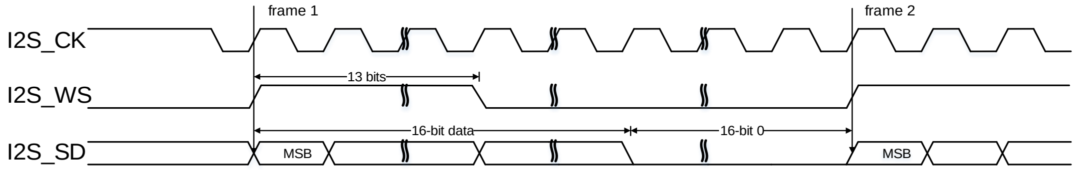
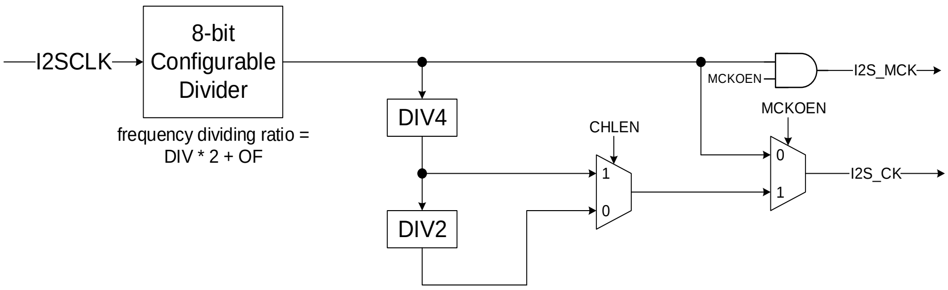


Figure 23-53. PCM standard long frame synchronization mode timing diagram (DTLEN=00, CHLEN=1, CKPL=1)



23.4.4. I2S clock

Figure 23-54. Block diagram of I2S clock generator



The block diagram of I2S clock generator is shown as ***Figure 23-54. 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 source clock is the system clock(CK_SYS). The I2S bitrate can be calculated by the formulas shown in ***Table 23-7. I2S bitrate calculation formulas***.

Table 23-7. 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 23-8. Audio sampling frequency calculation formulas***.

Table 23-8. 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))

23.4.5. 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 23-9. Direction of I2S interface signals for each operation mode*.

Table 23-9. Direction of I2S interface signals for each operation mode

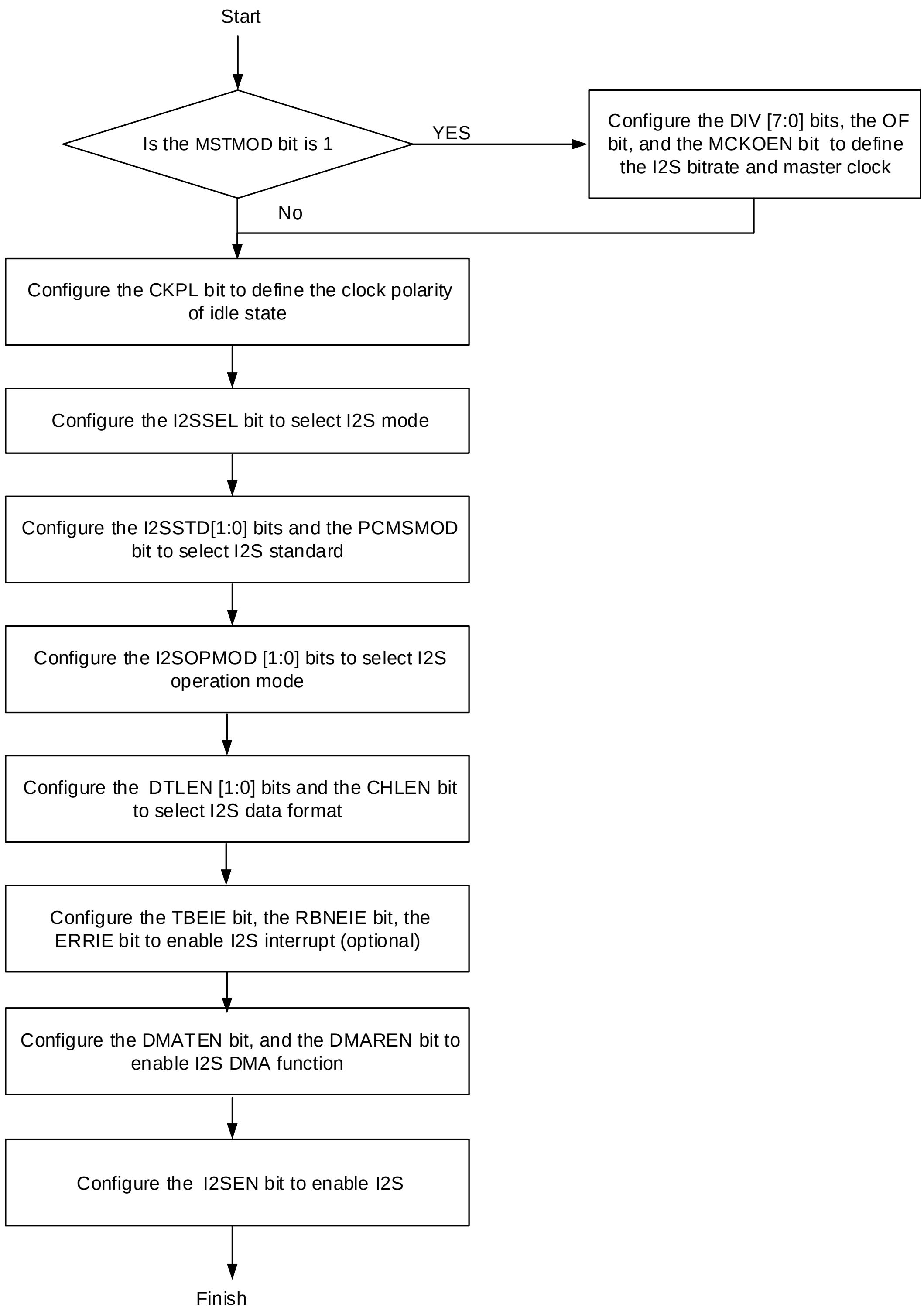
Operation mode	I2S_MCK	I2S_CK	I2S_WS	I2S_SD
Master transmission	Output or NU(1)	Output	Output	Output
Master reception	Output or NU(1)	Output	Output	Input
Slave transmission	Input or NU(1)	Input	Input	Output
Slave reception	Input or NU(1)	Input	Input	Input

(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 23-55. I2S initialization sequence*.

Figure 23-55. I2S initialization sequence



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.

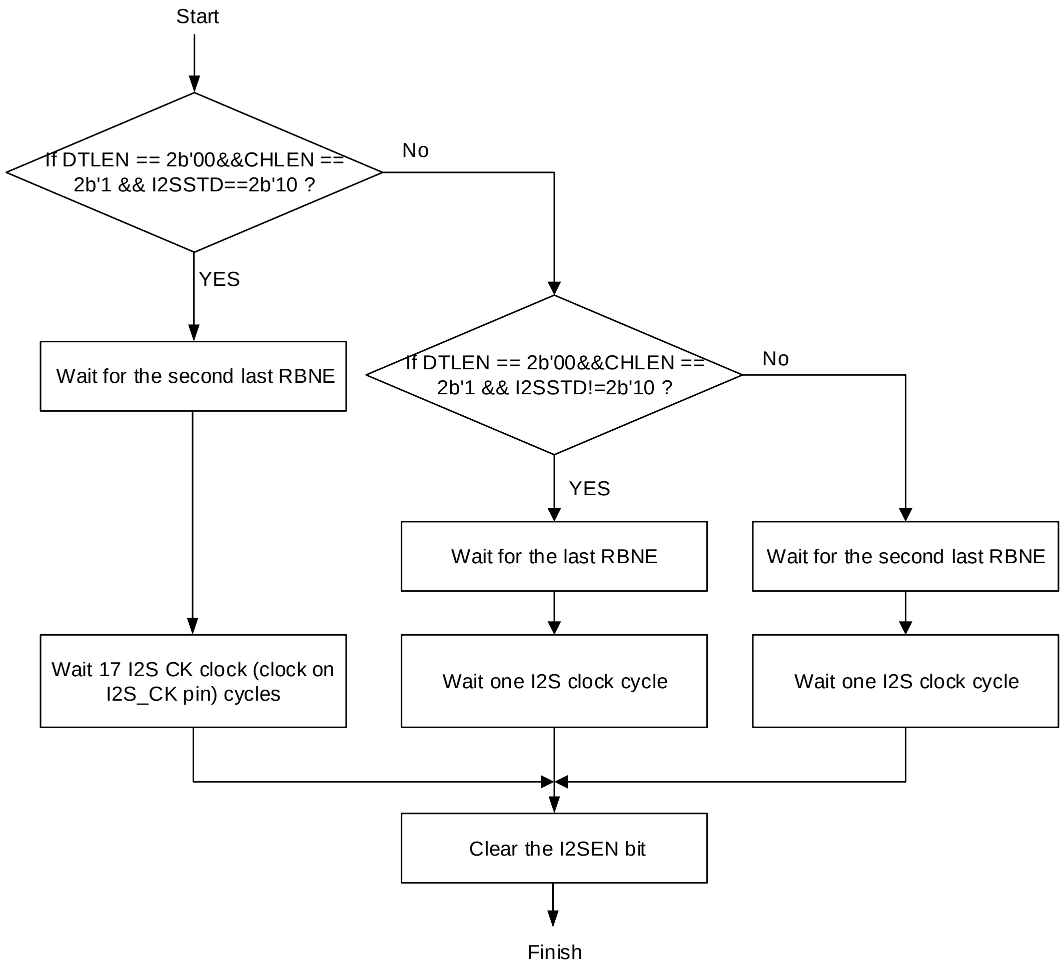
I2S master reception sequence

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 23-56. I2S master reception disabling sequence***.

Figure 23-56. I2S master reception disabling sequence



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.

23.4.6. DMA function

DMA function is the same as SPI mode. The only difference is that the CRC function is not available in I2S mode.

23.4.7. 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 23-10. I2S interrupt.***

Table 23-10. 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	

23.5. Register definition

SPI0 base address: 0x4001 3000

SPI1/ I2S1 base address: 0x4000 3800

23.5.1. 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	SWNSSEN	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 SPI1) 0: 8-bit data frame format 1: 16-bit data frame format
	CRCL	CRC length (only for SPI0) 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 PCLK means PCLK2 when using SPI0. PCLK means PCLK1 when using SPI1

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.

23.5.2. Control register 1 (SPI_CTL1)

Address offset: 0x04

Reset value: 0x0000 0700 for SPI0, 0x0000 0000 for SPI1

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 SPI0) 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 SPI0) 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 SPI0)

		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 SPI0) 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.

23.5.3. 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 SPI0) 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 SPI0) 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.

23.5.4. Data register (SPI_DATA)

Address offset: 0x0C

Reset value: 0x0000 0000

For SPI0, this register can be accessed by byte (8-bit) or half-word (16-bit).For SPI1, 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 SPI0, 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 SPI1, 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, SPI0 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.

23.5.5. 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

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.

23.5.6. 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).

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
RCRC[15:0]															
r															

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 SPI1, 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 SPI0, 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.

23.5.7. 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).

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
TCRC[15:0]															
r															

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 SPI1, 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 SPI0, 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.

23.5.8. 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.

23.5.9. 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.
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.

23.5.10. Quad-SPI mode control register (SPI_QCTL) of SPI0

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 SPI0.
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 SPI0.
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 SPI0.

24. VREF

24.1. Overview

A precision internal reference circuit is inside. The precision internal reference is used to provide reference voltage for ADC/DAC, or used by off-chip circuit connecting to V_{REF} pin.

24.2. Characteristics

The precision internal reference features are described as follows:

- Stable voltage, and product trimmed
- Connects to V_{REF} pin to source off-chip circuits
- Provides 2.5V/2.048V reference voltage(only 2.5V for GD32L233XX)

24.3. Function overview

The precision reference is enabled by set the VREFEN bit in VREF_CS register (the SYSCFGEN bit in RCU_APB2EN register needs to be set to 1 before that), producing 2.5V reference voltage and connecting to V_{REF} pin. When VREFEN is disabled, off-chip reference voltage could be injected to V_{REF} pin to source ADC/DAC. If there is no V_{REF} pin (refer to datasheet), the V_{REF} is connected to V_{DDA} and the VREFEN bit must keep 0.

When using precision internal reference voltage, and a bypass capacitor about 1uF (or 1uF and 10nF connected in parallel) which is recommended to ground is required.

Figure 24-1. Precision Reference Connection for GD32L233xx

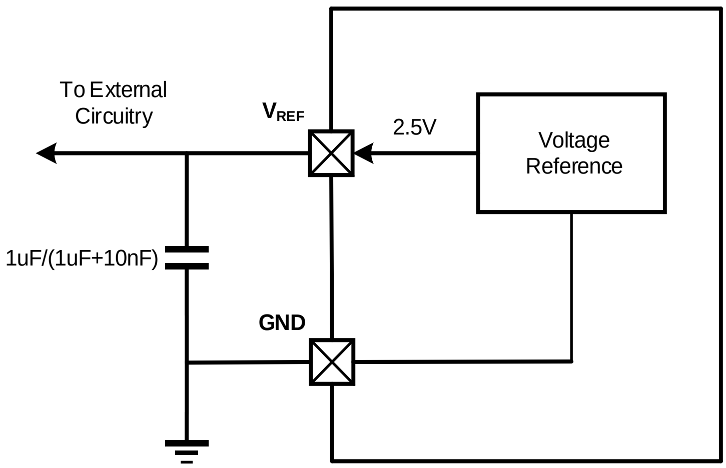
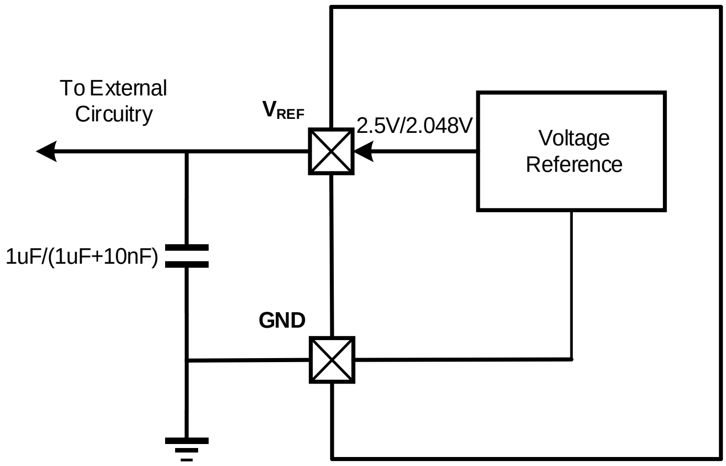


Figure 24-2. Precision Reference Connection for GD32L235xx



The internal reference voltage can be configured in four different modes depending on VREFEN and HIPM bits configuration. These modes are provided in the table below:

Table 24-1 VREF MODES

VREFEN	HIPM	Mode
0	0	VREF disabled – V_{REF} pin pulled-down to V_{SSA}
0	1	External voltage reference mode: – VREF disabled – V_{REF} pin floating
1	0	Internal voltage reference mode: – VREF enabled – V_{REF} pin connected to VREF output
1	1	Hold mode: – VREF disabled – V_{REF} pin floating. The voltage is held with the external capacitor – VREFRDY detection disabled and VREFRDY bit keeps last state

After enabling the VREF by setting VREFEN bit and reset HIPM bit in the VREF_CS register, the user must wait VREFRDY bit is set, indicating that the voltage reference output has reached its expected value.

24.4. Register definition

VREF base address: 0x4001 0030

24.4.1. Control and status register (VREF_CS)

For GD32L233xx devices

Address offset: 0x00

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												VREFRDY	Reserved	HIPM	VREFEN
												r		rw	rw

Bits	Fields	Descriptions
31:4	Reserved	Must be kept at reset value.
3	VREFRDY	VREF ready 0: The VREF output is not ready. 1: The VREF output is ready.
2	Reserved	Must be kept at reset value.
1	HIPM	High impedance mode 0: V_{REF+} is internally connected to the VREF output. 1: V_{REF+} pin is high impedance.
0	VREFEN	VREF enable 0: VREF is disabled 1: VREF is enabled

For GD32L235xx devices

Address offset: 0x00

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												VREFRDY	VREFS	HIPM	VREFEN
												r	rw	rw	rw

Bits	Fields	Descriptions

31:4	Reserved	Must be kept at reset value.
3	VREFRDY	VREF ready 0: The VREF output is not ready. 1: The VREF output is ready.
2	VREFS	Voltage reference select This bit sets the value of voltage reference output by the VREF 0: The voltage reference is around 2.048 V. 1: The voltage reference is around 2.5 V.
1	HIPM	High impedance mode 0: V_{REF+} is internally connected to the VREF output. 1: V_{REF+} pin is high impedance.
0	VREFEN	VREF enable 0: VREF is disabled 1: VREF is enabled

24.4.2. Calibration register (VREF_CALIB)

Address offset: 0x04

Reset value: 0x0000 00xx

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										VREFCAL[5:0]					
										rw					

Bits	Fields	Descriptions
31:6	Reserved	Must be kept at reset value.
5:0	VREFCAL	VREF calibration These bits are automatically initialized after reset with the trimming value stored in the Flash during production test. Writing into these bits to adjust the internal VREF voltage.

25. Segment LCD controller (SLCD)

25.1. Overview

The SLCD controller directly drives LCD displays by creating the AC segment and common voltage signals automatically. It can drive the monochrome passive liquid crystal display (LCD) which composed of a plurality of segments (pixels or complete symbols) that can be converted to visible or invisible. The SLCD controller can support up to 32 segments and 8 commons.

25.2. Characteristics

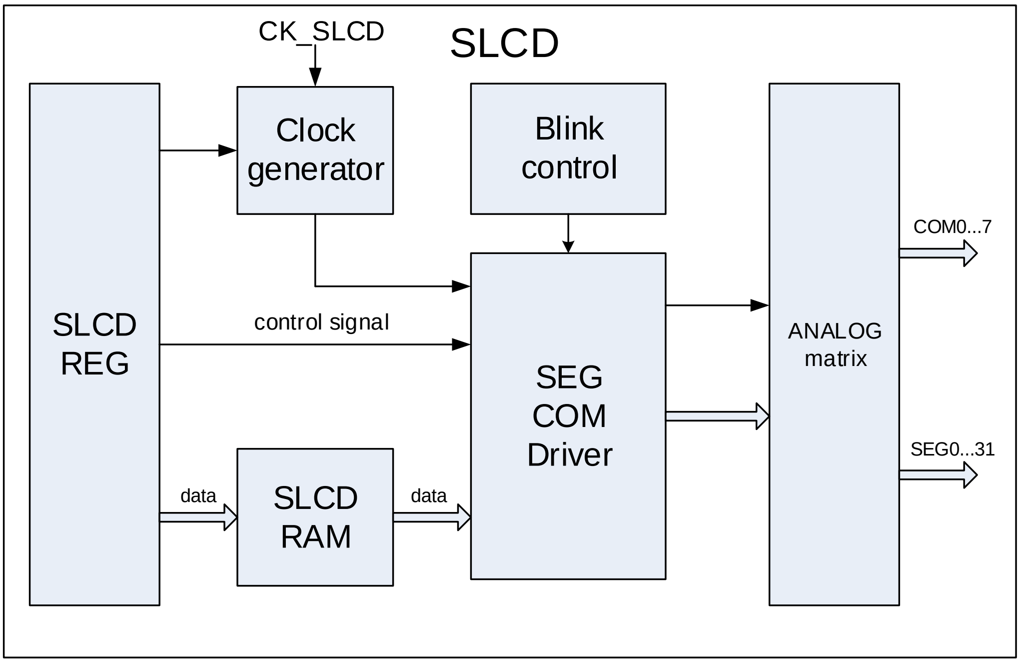
- Configurable frame frequency
- Blinking of individual segments or all segments
- Supports Static, 1/2, 1/3, 1/4, 1/6 and 1/8 duty
- Supports 1/2, 1/3 and 1/4 bias
- Double buffer up to 8x32 bits registers to store SLCD_DATAx
- The contrast can also be adjusted by configuring dead time
- Optional voltage output driver for enhance SLCD driving capability (Only for GD32L233)

25.3. Function overview

25.3.1. SLCD Architecture

The block diagram of the SLCD controller is shown as follows.

Figure 25-1. SLCD Block Diagram



The SLCD REG is the register of SLCD controller, which configured by APB bus, and generate interrupt to CPU. It includes SLCD_CTL, SLCD_CFG, SLCD_STAT, SLCD_STATC, SLCD_DATAx registers.

The Clock generator generates SLCD clock from input clock. The SLCD clock drivers the blink control and SEG/COM driver. The Blink control generates blink frequency and blink pixels. The SEG/COM driver generates segment and common signals to ANALOG matrix. The ANALOG matrix implements segment and common voltages.

25.3.2. Clock generator

SLCD input clock is the same as RTCCLK, 3 different clock sources: LXTAL, IRC32K and HXTAL divided by 32 can be selected by RTCSRC bits in RCU_BDCTL register. The input clock frequency varies from 32KHz to 1MHz.

The SLCD controller uses the input clock signal from the integrated clock divider to generate the timing for common and segment lines. The SLCD clock frequency is selected with the PSC and DIV bits in SLCD_CFG registers. The resulting SLCD clock frequency is calculated by:

$$f_{SLCD} = \frac{f_{CK\_SLCD}}{2^{PSC} \times (DIV+16)} \tag{25-1}$$

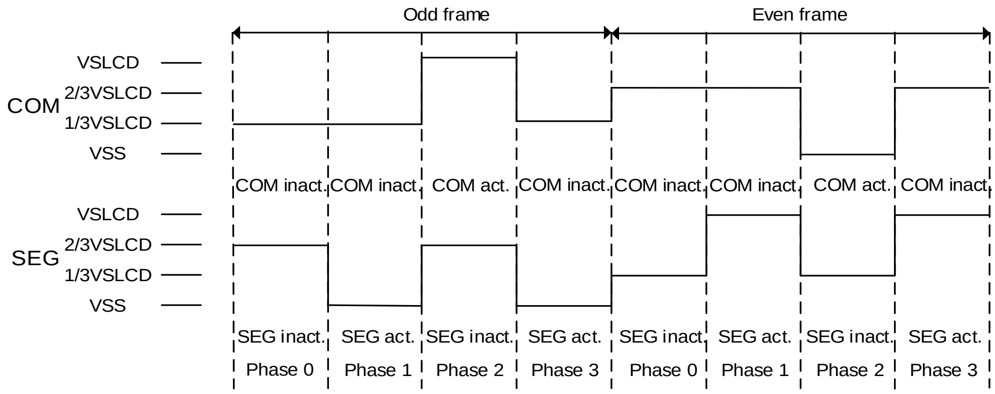
The SLCD clock is the time base for the SLCD controller. The frequency of SLCD clock is equivalent to the phase frequency. One SLCD frame is one odd frame or one even frame, both of them have several phases as many as active common terminals. So the frame frequency is calculated by:

$$f_{frame} = f_{SLCD} \times DUTY \tag{25-2}$$

Note: The DUTY is the number defined as 1/ (the number of common terminals on a given SLCD display).

The SOF bit in SLCD_STAT register is set by the hardware at the start of the frame, and the SLCD interrupt is executed if the SOFIE bit in SLCD_CFG is set. SOF is cleared by writing 1 to the SOFC bit in SLCD_STATC register.

Figure 25-2. 1/3 Bias, 1/4 Duty



25.3.3. Blink control

The SLCD controller also supports blinking. The blinking mode controlled by BLKMOD bits in SLCD_CFG register. BLKMOD = 01 allows to blink individual segment on SEG0 with COM0, with BLKMOD = 10 all commons on SEG0 are blinking, with BLKMOD = 11 all segments with all commons are blinking, and with BLKMOD = 00 blinking is disabled.

The blink frequency is generated from SLCD clock and selected with BLKDIV bit in SLCD_CFG registers. The resulting BLINK frequency is calculated by:

$$f_{BLINK} = \frac{f_{SLCD}}{2^{(BLKDIV+3)}} \tag{25-3}$$

After a blinking mode (BLKMOD = 01, 10 or 11) is selected, the enabled segments or all segments go blank at the next frame boundary and stay off for half a BLKCLK period. Then they go active at the next frame boundary and stay on for another half BLKCLK period before they go blank again at a frame boundary.

25.3.4. SEG/COM Driver

SEG/COM Driver generates segments and commons signals.

BIAS generator:

The bias is selected by setting BIAS bits in SLCD_CTL registers. It has its max amplitude VSLCD or VSS only in the corresponding phase of a frame cycle. The odd frame voltage and even frame voltage are shown as follows:

Table 25-1. The odd frame voltage

BIAS	Static	1/2 bias	1/3 bias	1/4 bias
COM active	VSLCD	VSLCD	VSLCD	VSLCD
COM inactive	/	1/2 VSLCD	1/3 VSLCD	1/4 VSLCD
SEG active	VSS	VSS	VSS	VSS
SEG inactive	VSLCD	VSLCD	2/3 VSLCD	1/2 VSLCD

Table 25-2. The even frame voltage

BIAS	Static	1/2 bias	1/3 bias	1/4 bias
COM active	VSS	VSS	VSS	VSS
COM inactive	/	1/2 VSLCD	2/3 VSLCD	3/4 VSLCD
SEG active	VSLCD	VSLCD	VSLCD	VSLCD
SEG inactive	VSS	VSS	1/3 VSLCD	1/2 VSLCD

COM signal:

The common signal is selected by DUTY bits in SLCD_CTL register. When DUTY is 000, static duty selected. Only COM[0] used and only one phase in odd frame or even frame, so COM[0] driver active signal always. When DUTY is 001, only COM[1:0] and 2 phases used. When DUTY is 010, only COM[2:0] and 3 phases used. When DUTY is 011, only COM[3:0] and 4 phases used. When DUTY is 100, COM[7:0] and 8 phases used. When DUTY is 101, COM[5:0] and 6 phases used. The all common signal driver is shown as follows:

Table 25-3. The all common signal driver

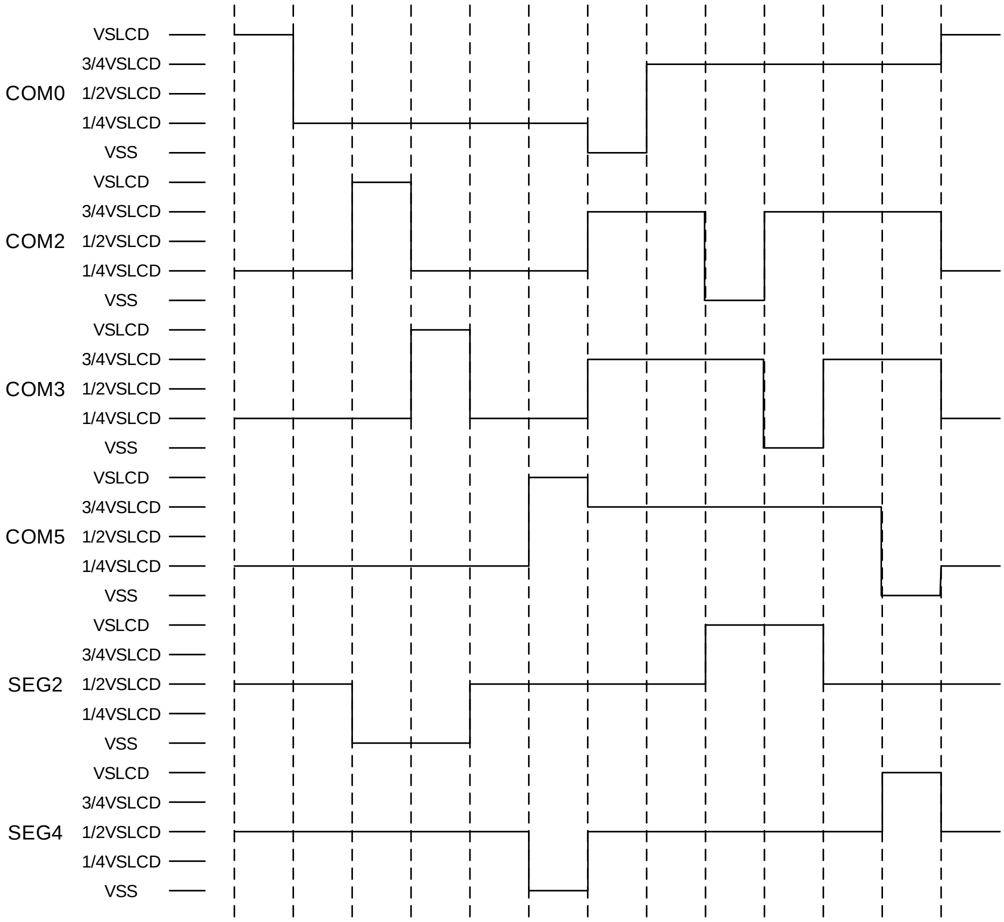
phase	1	2	3	4	5	6	7	8
COM0	active	inactive	inactive	inactive	inactive	inactive	inactive	inactive
COM1	inactive	active	inactive	inactive	inactive	inactive	inactive	inactive
COM2	inactive	inactive	active	inactive	inactive	inactive	inactive	inactive
COM3	inactive	inactive	inactive	active	inactive	inactive	inactive	inactive
COM4	inactive	inactive	inactive	inactive	active	inactive	inactive	inactive
COM5	inactive	inactive	inactive	inactive	inactive	active	inactive	inactive
COM6	inactive	inactive	inactive	inactive	inactive	inactive	active	inactive
COM7	inactive	inactive	inactive	inactive	inactive	inactive	inactive	active

SEG signal:

The segment signals are read from SLCD_DATAx registers. The segment signals data are SLCD_DATAx when phase x. When the value is 1, the corresponding segment drives active signal. When the value is 0, the corresponding segment drives inactive signal.

For example, if the application need to active the pixel COM2 SEG2, COM3 SEG2, and COM5 SEG4. It should set the bit2 in the SLCD_DATA2, the bit2 in the SLCD_DATA3, and the bit4 in the SLCD_DATA5. Then the SEG2 signal will active at the third and fourth phase of each odd and even frame, the SEG4 signal will active at the sixth phase of each odd and even frame. The active and inactive voltages are shown in ***Table 25-1. The odd frame voltage*** and ***Table 25-2. The even frame voltage***. The segment signals show in figure ***Figure 25-3. 1/4 Bias, 1/6 Duty*** is the result to the above configuration when bias is 1/4 and duty is 1/6.

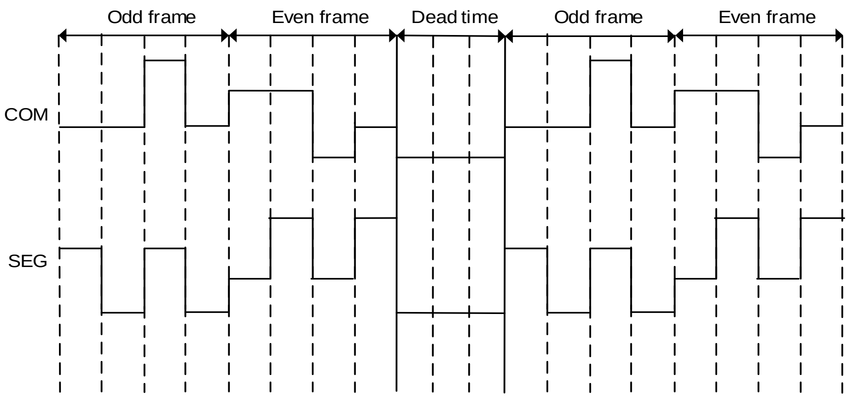
Figure 25-3. 1/4 Bias, 1/6 Duty



DEAD time:

The dead time is using DTD bits in SLCD_CFG register. It inserts VSS after each even frame. The number of phase inserted is defined by DTD bits. The application can adjust the contrast according to the configuration of dead time.

Figure 25-4. SLCD dead time (1/3 Bias, 1/4 Duty)



25.3.5. Double buffer memory

The double buffer memory is used to ensure the coherency of the displayed information.

The application access the first buffer according to modify the SLCD_DATAx registers. After writing the displayed information into the SLCD_DATAx registers, the application need to set the UPRF bit in SLCD_STAT register, then the hardware will transfer the data from the first buffer to the seconed buffer, during this time, the UPRF keeps set and the SLCD_DATAx registers are write protected. When the transfer is completed, the UPRF is cleared and the UPDF is set by the hardware, an interrupt will be generated if the UPDIE is set. The segment signal is driven by the data in the second buffer, so the displayed information will not be influenced by writing SLCD_DATAx.

If the UPRF is set when the display is disabled (SLCDON = 0), the transfer will not occur until the SLCDON is set.

25.3.6. ANALOG matrix

The analog matrix supplies SLCD voltage. The SLCD voltage levels are generated by the VSLCD pin or by the internal voltage step-up converter (depending on the VSRC bit in the SLCD_CTL register).

For GD32L233 series, when using the internal voltage, the VSLCD value can be selected from VSLCD0 to VSLCD7 by the CONR[2:0] bits in the SLCD_CFG register (Refer to the product datasheet for the VSLCDx values). The application can adjust the contrast according to the change of VSLCD value.

For GD32L235 series, when using the internal voltage, when using the internal voltage, the VDD voltage as internal voltage source.

User can adjust the contrast is by using the deadtime.

The analog matrix supplies intermediate voltage levels (1/3 VSLCD, 2/3 VSLCD or 1/4 VSLCD, 2/4 VSLCD, 3/4 VSLCD) between VSS and VSLCD through an internal resistor divider network as shown in ***Figure 25-5. SLCD Resistr divider network for GD32L233 series*** and ***Figure 25-6. SLCD Resistr divider network for GD32L235 series***.

Figure 25-5. SLCD Resistr divider network for GD32L233 series

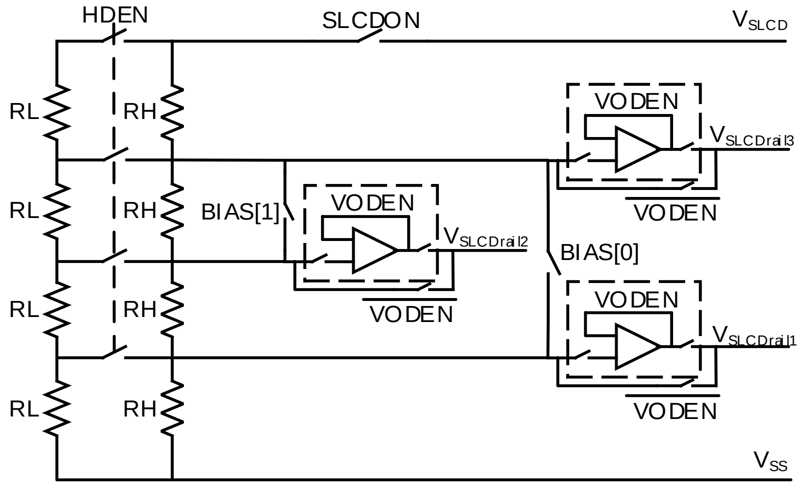
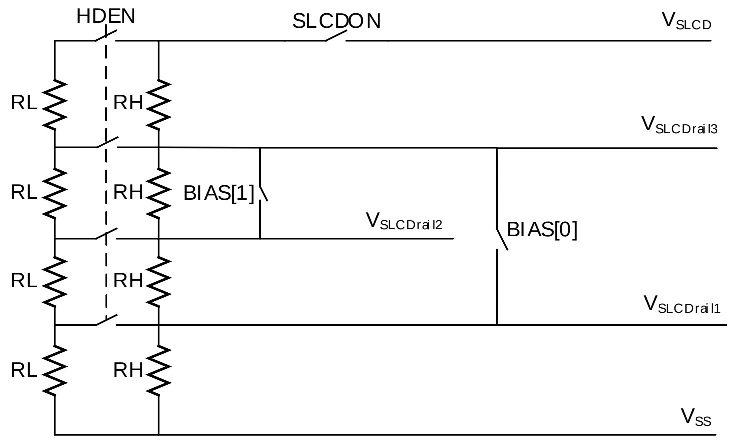


Figure 25-6. SLCD Resistr divider network for GD32L235 series



During the transitions, the low value resistors (R_L) are switched on to increase the current in order to quickly reach the static state. Then the low value resistors (R_L) are switched off, the high value resistors (R_H) are used to reduce the power. The length of the time during R_L is switched on depend on the PULSE[2:0] bits in the SLCD_CFG register. The R_L can be always switched on according to setting the HDEN bit in the SLCD_CFG register.

Enhance mode:

For GD32L233 series, the SLCD module integrates an optional voltage output driver, which can enter the enhanced mode by enabling the VODEN bit of the SLCD_CTL register. Since the voltage output driver is enabled, the voltage interference caused by the LCD capacitive load mounted on the bridge can be reduced. Thus, a stable voltage can be obtained to enhance SLCD driving capacity.

In enhanced mode, the high value resistor bridge (R_{HN}) will generate an intermediate voltage, the HDEN bit or PULSE bit configuration will be ignored, and the low value resistor bridge (R_{LN}) will be automatically disabled, thereby reducing power consumption.

The VLCD power supply is not used for the voltage output driver. The voltage output driver can only be configured when the SLCD controller is not activated.

Note: GD32L235 series does not support enhance mode.

25.3.7. V_{SLCD} voltage source

V_{SLCD} voltage monitoring:

The VSLCDEN bit in the ADC_CTL1 register is used to measure the V_{SLCD} voltage. Since the V_{SLCD} voltage may be higher than V_{DDA}, in order to ensure the normal operation of the ADC, the internal $V_{SLCDrail1}$ analog voltage is connected to the ADC_IN19 input channel.

The $V_{SLCDrail1}$ value under different BIAS[1:0] biases is different, which is determined by the internal analog circuit.

1. BIAS[1:0]=00, $V_{SLCDrail1}$=1/4V_{SLCD}, V_{SLCD} voltage monitoring function can be used, the value obtained by ADC conversion is one-fourth of the V_{SLCD} voltage.
2. BIAS[1:0]=01, $V_{SLCDrail1}$ is an invalid value, and the V_{SLCD} voltage monitoring function cannot be used normally.
3. BIAS[1:0]=10, $V_{SLCDrail1}$=1/3V_{SLCD}, V_{SLCD} voltage monitoring function can be used, and the value obtained by ADC conversion is one-third of the V_{SLCD} voltage.

To prevent the battery power from being accidentally consumed, it is recommended to enable VSLCDEN only when necessary to perform ADC conversion.

V_{SLCD} voltage source configuration:

The SLCD can selected the internal voltage source or external voltage source by the VSRC bit of the SLCD_CTL register. The precautions for using internal/external voltage sources for SLCD are as follows:

Internal voltage source:

For GD32L233 series, when the SLCD selects the internal voltage source, the VSRC should be configured 1’b0 and the PD6 pin needs to be configured in analog mode. A external capacitor should be connected to GND. Please refer to Datasheet for the capacitance value. When an SLCD selects an internal voltage source, the following procedure should be followed.

1. Configure the PD6 pin to analog mode.
2. Configure the SLCD register and select the internal voltage source.
3. Wait for the external capacitor to complete charging (when the external capacitor is 2uF, the typical charging time is about 1.5ms).
4. Enable the SLCD module.

For GD32L235 series, when the SLCD selects the internal voltage source, the VDD voltage as internal voltage.

External voltage source:

For GD32L233 and GD32L235 series, when the SLCD selects an external voltage source, th the VSRC should be configured 1’b1 and the PD6 pin needs to be configured in analog mode and connected to an external voltage source. When an SLCD selects an external voltage source, the following procedure should be followed.

1. Configure the PD6 pin to analog mode.
2. Configure the SLCD register and select the external voltage source.
3. Enable the SLCD module.

25.4. Register definition

SLCD base address: 0x4000 2400

25.4.1. Control register (SLCD_CTL)

For GD32L233xx devices

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							VODEN	COMS	BIAS[1:0]		DUTY[2:0]			VSRC	SLCDON
							rw	rw	rw		rw			rw	rw

Bits	Fields	Descriptions
31:9	Reserved	Must be kept at reset value.
8	VODEN	Voltage output driver enable 0: Voltage output driver disable 1: Voltage output driver enable When VBUFEN=1, the SLCD voltage driving capability is improved.
7	COMS	Common/segment pad select This bit is used to common/segment pad selection. When duty selects 1/8 or 1/6, SLCD_COM[7:4] pad is always select SLCD_COM[7:4] function whatever this bit is set or reset. 0: SLCD_COM[7:4] pad select SLCD_COM[7:4] 1: SLCD_COM[7:4] pad select SLCD_SEG[31:28]
6:5	BIAS[1:0]	Bias select Bias is the number of voltage levels used when driving a SLCD. It is defined as 1/(number of voltage levels used to drive an SLCD display – 1). 00: 1/4 Bias (5 voltage levels: VSS, 1/4VSLCD, 1/2VSLCD, 3/4VSLCD, VSLCD) 01: 1/2 Bias (3 voltage levels: VSS, 1/2VSLCD, VSLCD) 10: 1/3 Bias (4 voltage levels: VSS, 1/3VSLCD, 2/3VSLCD, VSLCD) 11: Reserved
4:2	DUTY[2:0]	Duty select These bits determine the duty cycle. Duty is the number defined as 1/(number of common terminals on a given SLCD display).

		000: Static duty 001: 1/2 duty 010: 1/3 duty 011: 1/4 duty 100: 1/8 duty 101: 1/6 duty 110: Reserved 111: Reserved
1	VSRC	SLCD Voltage source Set this bit determines which is the SLCD voltage source. 0: Internal source 1: External source (VSLCD pin)
0	SLCDON	SLCD controller start Set this bit by software to start SLCD controller. Clear this bit by software to stop SLCD controller and the SLCD controller stop at the beginning of the next frame. 0: SLCD Controller stop 1: SLCD Controller start

For GD32L235xx devices

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								COMS	BIAS[1:0]		DUTY[2:0]			VSRC	SLCDON
								rw	rw		rw			rw	rw

Bits	Fields	Descriptions
31:8	Reserved	Must be kept at reset value.
7	COMS	Common/segment pad select This bit is used to common/segment pad selection. When duty selects 1/8 or 1/6, SLCD_COM[7:4] pad is always select SLCD_COM[7:4] function whatever this bit is set or reset. 0: SLCD_COM[7:4] pad select SLCD_COM[7:4] 1: SLCD_COM[7:4] pad select SLCD_SEG[31:28]
6:5	BIAS[1:0]	Bias select Bias is the number of voltage levels used when driving a SLCD. It is defined as 1/(number of voltage levels used to drive an SLCD display – 1).

		00: 1/4 Bias (5 voltage levels: VSS, 1/4VSLCD, 1/2VSLCD, 3/4VSLCD, VSLCD) 01: 1/2 Bias (3 voltage levels: VSS, 1/2VSLCD, VSLCD) 10: 1/3 Bias (4 voltage levels: VSS, 1/3VSLCD, 2/3VSLCD, VSLCD) 11: Reserved
4:2	DUTY[2:0]	Duty select These bits determine the duty cycle. Duty is the number defined as 1/(number of common terminals on a given SLCD display). 000: Static duty 001: 1/2 duty 010: 1/3 duty 011: 1/4 duty 100: 1/8 duty 101: 1/6 duty 110: Reserved 111: Reserved
1	VSRC	SLCD Voltage source Set this bit determines which is the SLCD voltage source. 0: Internal source (VDD voltage) 1: External source (VSLCD pin)
0	SLCDON	SLCD controller start Set this bit by software to start SLCD controller. Clear this bit by software to stop SLCD controller and the SLCD controller stop at the beginning of the next frame. 0: SLCD Controller stop 1: SLCD Controller start

25.4.2. Configuration register (SLCD_CFG)

For GD32L233xx devices

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						PSC[3:0]				DIV[3:0]				BLKMOD[1:0]	
						rw				rw				rw	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BLKDIV[2:0]			CONR[2:0]			DTD[2:0]			PULSE[2:0]			UPDIE	Reserved	SOFIE	HDEN
rw			rw			rw			rw			rw		rw	rw

Bits	Fields	Descriptions

31:26	Reserved	Must be kept at reset value.
25:22	PSC[3:0]	SLCD clock prescaler Set these bits define the prescaler of SLCD clock. 0000: $f_{PSC} = f_{in\_clk}$ 0001: $f_{PSC} = f_{in\_clk}/2$ 0010: $f_{PSC} = f_{in\_clk}/4$ ... 1111: $f_{PSC} = f_{in\_clk}/32768$
21:18	DIV[3:0]	SLCD clock divider Set these bits define the division factor of the DIV divider. 0000: $f_{SLCD} = f_{PSC}/16$ 0001: $f_{SLCD} = f_{PSC}/17$ 0010: $f_{SLCD} = f_{PSC}/18$ ... 1111: $f_{SLCD} = f_{PSC}/31$
17:16	BLKMOD[1:0]	Blink mode 00: No Blink 01: Blink on SEG[0], COM[0] (1 pixel) 10: Blink on SEG[0], all COMs (up to 8 pixels depending on the programmed duty) 11: Blink on all SEGs and all COMs (all pixels)
15:13	BLKDIV[2:0]	Blink frequency divider 000: $f_{BLINK} = f_{SLCD}/8$ 001: $f_{BLINK} = f_{SLCD}/16$ 010: $f_{BLINK} = f_{SLCD}$ /32 011: $f_{BLINK} = f_{SLCD}$ /64 100: $f_{BLINK} = f_{SLCD}/128$ 101: $f_{BLINK} = f_{SLCD}/256$ 110: $f_{BLINK} = f_{SLCD}/512$ 111: $f_{BLINK} = f_{SLCD}/1024$
12:10	CONR[2:0]	Contrast ratio When chosing the internal voltage source (VSRC=0), these bits specify the VSLCD voltage. It ranges from VSLCD0 to VSLCD7(typical 2.65 V to 3.67V), Refer to the product datasheet for the VSLCDx values. When chosing the external voltage source (VSRC=1), these bits is invalid. 000: VSLCD0 001: VSLCD1 010: VSLCD2 011: VSLCD3 100: VSLCD4 101: VSLCD5 110: VSLCD6

		111: VSLCD7
9:7	DTD[2:0]	Dead time duration Set these bits configure the length of the dead time between frames. 000: No dead time 001: 1 phase dead time 010: 2 phase dead time ... 111: 7 phase dead time
6:4	PULSE[2:0]	Pulse ON duration Set these bits define the pulse duration in terms of PSC pulses. 000: 0 001: $1/f_{PSC}$ 010: $2/f_{PSC}$ 011: $3/f_{PSC}$ 100: $4/f_{PSC}$ 101: $5/f_{PSC}$ 110: $6/f_{PSC}$ 111: $7/f_{PSC}$
3	UPDIE	SLCD update done interrupt enable This bit is set and cleared by software. 0: SLCD Update Done interrupt disabled 1: SLCD Update Done interrupt enabled
2	Reserved	Must be kept at reset value.
1	SOFIE	Start of frame interrupt enable This bit is set and cleared by software. 0: SLCD Start of Frame interrupt disabled 1: SLCD Start of Frame interrupt enabled
0	HDEN	High drive enable This bit is set and cleared by software. 0: Permanent high drive disabled. The time during which R_L is enabled is configured by the PULSE[2:0]. 1: Permanent high drive enabled. RL is always switched on, and the PULSE[2:0] is invalid.

For GD32L235xx devices

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				RSEL		PSC[3:0]				DIV[3:0]				BLKMOD[1:0]	
				rw		rw				rw				rw	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BLKDIV[2:0]			Reserved			DTD[2:0]			PULSE[2:0]			UPDIE	Reserved	SOFIE	HDEN
rw						rw			rw			rw		rw	rw

Bits	Fields	Descriptions
31:28	Reserved	Must be kept at reset value.
27:26	RSEL[1:0]	Weak driving resistance select 00: 6M 01: 4M 10: 2M 11: 1M
25:22	PSC[3:0]	SLCD clock prescaler Set these bits define the prescaler of SLCD clock. 0000: $f_{PSC} = f_{in\_clk}$ 0001: $f_{PSC} = f_{in\_clk}/2$ 0010: $f_{PSC} = f_{in\_clk}/4$ ... 1111: $f_{PSC} = f_{in\_clk}/32768$
21:18	DIV[3:0]	SLCD clock divider Set these bits define the division factor of the DIV divider. 0000: $f_{SLCD} = f_{PSC}/16$ 0001: $f_{SLCD} = f_{PSC}/17$ 0010: $f_{SLCD} = f_{PSC}/18$ ... 1111: $f_{SLCD} = f_{PSC}/31$
17:16	BLKMOD[1:0]	Blink mode 00: No Blink 01: Blink on SEG[0], COM[0] (1 pixel) 10: Blink on SEG[0], all COMs (up to 8 pixels depending on the programmed duty) 11: Blink on all SEGs and all COMs (all pixels)
15:13	BLKDIV[2:0]	Blink frequency divider 000: $f_{BLINK} = f_{SLCD}/8$ 001: $f_{BLINK} = f_{SLCD}/16$ 010: $f_{BLINK} = f_{SLCD}/32$ 011: $f_{BLINK} = f_{SLCD}/64$ 100: $f_{BLINK} = f_{SLCD}/128$ 101: $f_{BLINK} = f_{SLCD}/256$ 110: $f_{BLINK} = f_{SLCD}/512$

		111: $f_{BLINK} = f_{SLCD}/1024$
12:10	Reserved	Must be kept at reset value.
9:7	DTD[2:0]	Dead time duration Set these bits configure the length of the dead time between frames. 000: No dead time 001: 1 phase dead time 010: 2 phase dead time ... 111: 7 phase dead time
6:4	PULSE[2:0]	Pulse ON duration Set these bits define the pulse duration in terms of PSC pulses. 000: 0 001: $1/f_{PSC}$ 010: $2/f_{PSC}$ 011: $3/f_{PSC}$ 100: $4/f_{PSC}$ 101: $5/f_{PSC}$ 110: $6/f_{PSC}$ 111: $7/f_{PSC}$
3	UPDIE	SLCD update done interrupt enable This bit is set and cleared by software. 0: SLCD Update Done interrupt disabled 1: SLCD Update Done interrupt enabled
2	Reserved	Must be kept at reset value.
1	SOFIE	Start of frame interrupt enable This bit is set and cleared by software. 0: SLCD Start of Frame interrupt disabled 1: SLCD Start of Frame interrupt enabled
0	HDEN	High drive enable This bit is set and cleared by software. 0: Permanent high drive disabled. The time during which R_L is enabled is configured by the PULSE[2:0]. 1: Permanent high drive enabled. RL is always switched on, and the PULSE[2:0] is invalid.

25.4.3. Status flag register (SLCD_STAT)

For GD32L233xx devices

Address offset: 0x08

Reset value: 0x0000 0020

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										SYNF	VRDYF	UPDF	UPRF	SOF	ONF
										r	r	r	rs	r	r

Bits	Fields	Descriptions
31:6	Reserved	Must be kept at reset value.
5	SYNF	SLCD_CFG register synchronization flag This bit is set when SLCD_CFG register update to SLCD clock domain, and It is cleared by hardware when writing to the SLCD_CFG register. 0: SLCD_CFG Register not yet synchronized 1: SLCD_CFG Register synchronized to SLCD clock domain
4	VRDYF	SLCD voltage ready flag This bit is set and cleared by the hardware according to the SLCD voltage. 0: SLCD voltage Is not ready 1: Step-up converter is enabled and ready to provide the correct voltage
3	UPDF	Update SLCD data done flag This bit is set by hardware when update SLCD data done. It is cleared by writing 1 to the UPDC bit in the SLCD_STATC register. 0: No effect 1: SLCD data update done
2	UPRF	Update SLCD data request flag After modifying the the first buffer by the SLCD_DATAx registers, the application should set this bit to transfer the data to the second buffer. This bit stays set until the transfer is complete, the SLCD_DATAx register is write protected during this time. 0: No effect 1: Request SLCD data update
1	SOF	Start of frame flag This bit is set by hardware at the beginning of a new frame. It is cleared by writing 1 to the SOFC bit in the SLCD_STATC register. 0: No effect 1: Start of Frame flag
0	ONF	SLCD controller on flag This bit is set by hardware when SLCDON is set to 1, it is cleard by hardware after the SLCDON is cleared and the last frame is displayed.

0: SLCD Controller disabled.

1: SLCD Controller enabled

For GD32L235xx devices

Address offset: 0x08

Reset value: 0x0000 0020

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										SYNF	Reserved	UPDF	UPRF	SOF	ONF
										r		r	rs	r	r

Bits	Fields	Descriptions
31:6	Reserved	Must be kept at reset value.
5	SYNF	SLCD_CFG register synchronization flag This bit is set when SLCD_CFG register update to SLCD clock domain, and It is cleared by hardware when writing to the SLCD_CFG register. 0: SLCD_CFG Register not yet synchronized 1: SLCD_CFG Register synchronized to SLCD clock domain
4	Reserved	Must be kept at reset value.
3	UPDF	Update SLCD data done flag This bit is set by hardware when update SLCD data done. It is cleared by writing 1 to the UPDC bit in the SLCD_STATC register. 0: No effect 1: SLCD data update done
2	UPRF	Update SLCD data request flag After modifying the the first buffer by the SLCD_DATAx registers, the application should set this bit to transfer the data to the second buffer. This bit stays set until the transfer is complete, the SLCD_DATAx register is write protected during this time. 0: No effect 1: Request SLCD data update
1	SOF	Start of frame flag This bit is set by hardware at the beginning of a new frame. It is cleared by writing 1 to the SOFC bit in the SLCD_STATC register. 0: No effect 1: Start of Frame flag

0	ONF	SLCD controller on flag This bit is set by hardware when SLCDON is set to 1, it is cleard by hardware after the SLCDON is cleared and the last frame is displayed. 0: SLCD Controller disabled. 1: SLCD Controller enabled

25.4.4. Status flag clear register (SLCD_STATC)

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												UPDC	Reserved	SOFC	Reserved
												rc_w1		rc_w1	

Bits	Fields	Descriptions
31:4	Reserved	Must be kept at reset value.
3	UPDC	SLCD data update done clear bit Set this bit to clear the UPDF flag in SLCD_STAT register. 0: No effect 1: Clear UPDF flag
2	Reserved	Must be kept at reset value.
1	SOFC	Start of frame flag clear Set this bit to clear the SOF flag in the SLCD_STAT register. 0: No effect 1: Clear SOF flag
0	Reserved	Must be kept at reset value.

25.4.5. Display data registers (SLCD_DATAx) (x=0…7)

Address offset: 0x14 + 0x08 * 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
DATAx[31:16]															
rw															

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

Bits	Fields	Descriptions
31:0	SEG_DATAx[31:0]	Each bit corresponds to one pixel to display. 0: Pixel inactive 1: Pixel active

26. Comparator (CMP)

26.1. 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.

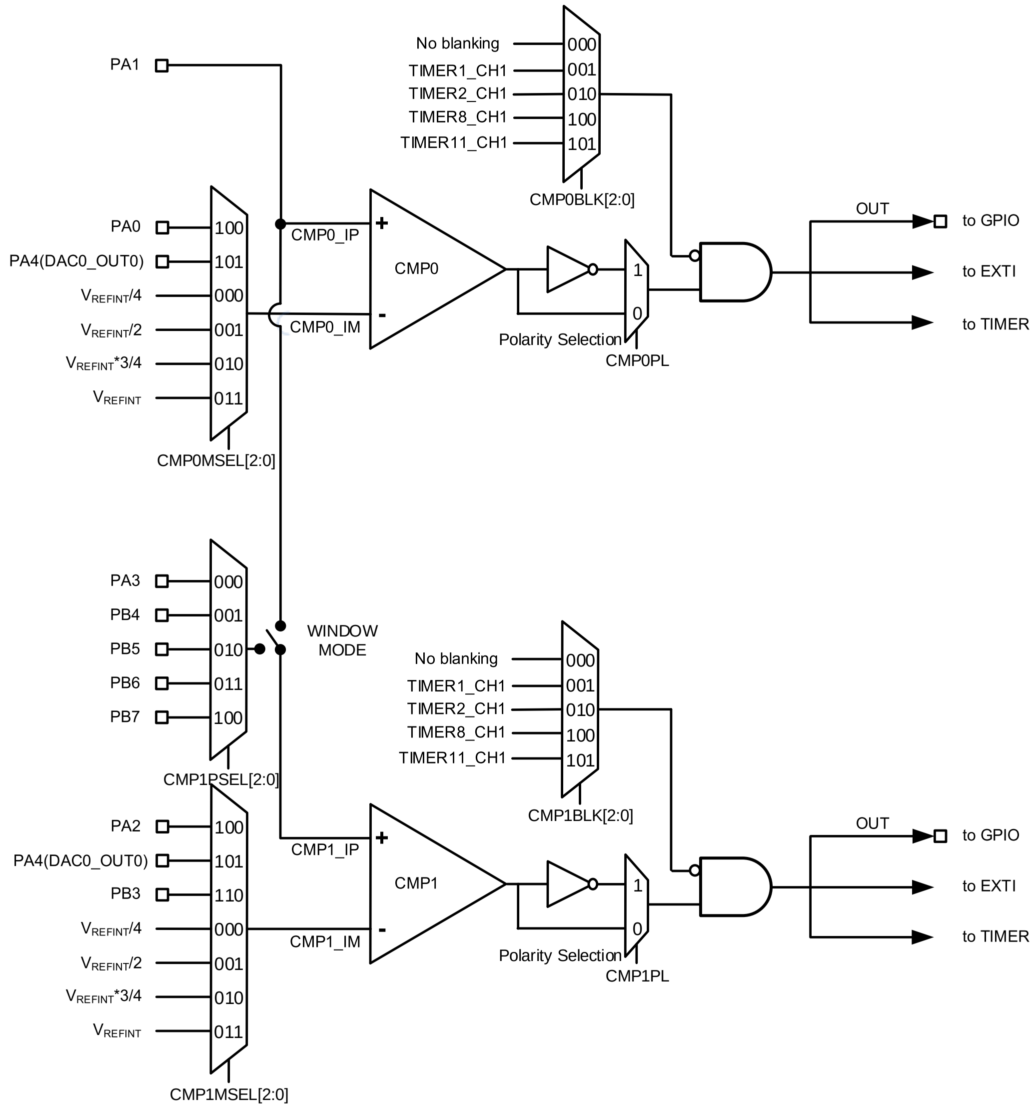
26.2. Characteristics

- 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.
- Window comparator.
- Outputs to I / O.
- Outputs to timers for triggering.
- Outputs to EXTI.

26.3. Function overview

The block diagram of CMP is shown below:

Figure 26-1. CMP block diagram



Note: V_{REFINT} is 1.2V.

26.3.1. CMP clock

The clock of the CMP which is connected to APB bus, is synchronous with PCLK.

26.3.2. CMP I / O configuration

These I / Os must be configured in analog mode in the GPIOs registers before they are selected as CMP inputs.

Considering pin definitions in datasheet, and the CMP output must be connected to corresponding alternate I / Os.

The CMP output can be redirected internally and externally simultaneously.

CMP output internally connect to the TIMER and the connections between them are as follows:

- CMP output to the TIMER input channel.

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 26-1 CMP inputs and outputs summary details the inputs and outputs of the CMP.

Table 26-1 CMP inputs and outputs summary

	CMP0	CMP1
CMP non inverting inputs connected to I / Os	PA1	PA3 PB4 PB5 PB6 PB7
CMP inverting inputs connected to I / Os	PA0	PA2 PB3
CMP inverting inputs connected to internal signals	V_{REFINT} / 4 V_{REFINT} / 2 V_{REFINT} * 3 / 4 V_{REFINT} DAC0_OUT0	V_{REFINT} / 4 V_{REFINT} / 2 V_{REFINT} * 3 / 4 V_{REFINT} DAC0_OUT0
CMP outputs connected to I/Os	PA0 PA6 PA11 PB0 PB8 PB10	PA2 PA7 PA12 PB5 PB9 PB11
CMP outputs connected to EXTI	•	
CMP outputs connected to internal signals	TIMER1_CH3, TIMER2_CH0	TIMER1_CH3, TIMER2_CH0

26.3.3. 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.

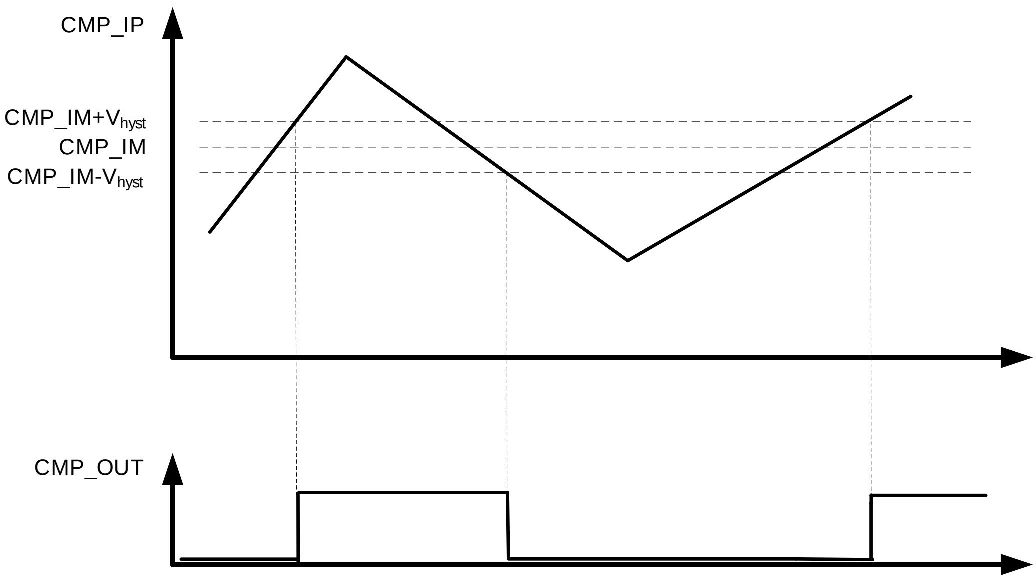
26.3.4. CMP windows mode

If the WNDEN bit in CMPx_CS register is set, comparator windows mode is enabled, input plus of comparator 1 is connected with input plus of comparator 0. If the minus input of CMP0 and CMP1 is connected to different voltage, the voltage range from lower threshold to upper threshold, is monitored by analyzing the comparator 0 and comparator 1 output.

26.3.5. 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 26-2. CMP hysteresis



26.3.6. CMP register write protection

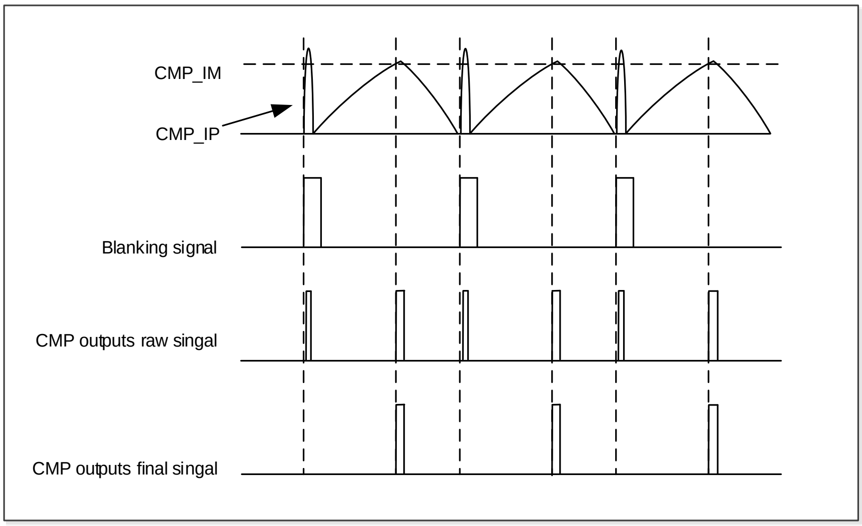
The CMP control and status register (CMPx_CS) can be protected from writing by setting CMPxLK bit to 1. The CMPx_CS register, including the CMPxLK bit will be read-only, and can only be reset by the MCU reset.

26.3.7. 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 26-3 The CMP outputs signal blanking shows the comparator output blank function.

Figure 26-3 The CMP outputs signal blanking



26.3.8. 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.

26.3.9. 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.

26.4. Register definition

CMP base address: 0x4001 7C00

26.4.1. 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	CMP0OSEL[1:0]		Reserved						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 TIMER1_CH1 output compare signal as blanking source 010: Select TIMER2_CH1 output compare signal as blanking source 011: Select TIMER8_CH1 output compare signal as blanking source 100: Select TIMER11_CH1 output compare signal as blanking source 101~111: 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
14:13	CMP0OSEL[1:0]	CMP0 output selection These bits are used to select the destination of the CMP0 output. 00: No selection 01: TIMER1 CH3 input capture 10: TIMER2 CH0 input capture 11: Reserved **Note:** It is recommended to enable CMP first, and then configure the timer channel, when using TIMER to capture the output signal of the comparator.
12:7	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: PA0 101: PA4 (DAC0_OUT0) 110~111: Reserved
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 / 10: Medium speed / medium power

		11: Low speed / low power
1	Reserved	Must be kept at reset value.
0	CMP0EN	CMP0 enable 0: CMP0 disabled 1: CMP0 enabled

26.4.2. 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	CMP1OSEL[1:0]		Reserved		CMP1PSEL[2:0]			Reserved	CMP1MSEL[2:0]			CMP1M[1:0]		WNDEN	CMP1EN
rw	rw				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 TIMER1_CH1 output compare signal as blanking source 010: Select TIMER2_CH1 output compare signal as blanking source 011: Select TIMER8_CH1 output compare signal as blanking source 100: Select TIMER11_CH1 output compare signal as blanking source All other values: 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
14:13	CMP1OSEL[1:0]	CMP1 output selection These bits are used to select the destination of the CMP1 output. 00: no selection 01: TIMER1 CH3 input capture 10: TIMER2 CH0 input capture 11: Reserved **Note:** It is recommended to enable CMP first, and then configure the timer channel, when using TIMER to capture the output signal of the comparator.
12:11	Reserved	Must be kept at reset value.
10:8	CMP1PSEL[2:0]	CMP1_IP input selection These bits are used to select the source connected to the CMP1_IP input of the CMP1. 000: PA3 001: PB4 010: PB5 011: PB6 100: PB7 All other values: reserved
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: PA2 101: PA4 (DAC0_OUT0) 110: PB3 111: Reserved
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 / 10: Medium speed / medium power 11: Low speed / low power
1	WNDEN	Windows mode enable This bit is used to select CMP1_IP source. 0: CMP1_IP is connected to CMP1 non_inverting input 1: CMP1_IP is connected to CMP0_IP
0	CMP1EN	CMP1 enable. 0: CMP1 disabled 1: CMP1 enabled

27. Controller area network (CAN)

27.1. Overview

CAN bus (Controller Area Network) is a bus standard designed to allow microcontrollers and devices to communicate with each other without a host computer.

As CAN network interface, basic extended CAN supports the CAN protocols version 2.0A and B. The CAN interface automatically handles the transmission and the reception of CAN frames. The CAN provides 28 scalable/configurable identifier filter banks in GD32L235xx. The filters are used for selecting the input message as software requirement and otherwise discarding the message. Three transmit mailboxes are provided to the software for transfer messages. The transmission scheduler decides which mailbox will be transmitted firstly. Three complete messages can be stored in every FIFO. The FIFOs are managed completely by hardware. Two receiving FIFOs are used by hardware to store the incoming messages. In addition, the CAN controller provides all hardware functions, which supports the time-triggered communication option, in safety-critical applications.

27.2. Characteristics

- Supports CAN protocols version 2.0A, B.
- Baud rates up to 1 Mbit/s.
- Supports the time-triggered communication.
- Interrupt enable and clear.

Transmission

- Supports 3 transmit mailboxes.
- Supports priority of transmission message.
- Supports time stamp at SOF transmission.

Reception

- Supports 2 Rx FIFOs and each has 3 messages depth.
- 28 scalable/configurable identifier filter banks in GD32L235xx.
- FIFO lock.

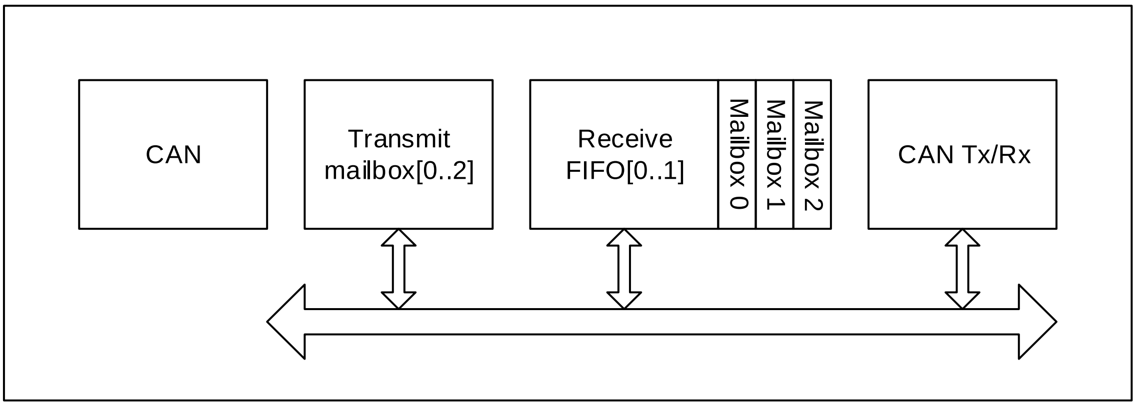
Time-triggered communication

- Disable retransmission automatically in time-triggered communication mode.
- 16-bit free timer.
- Time stamp on SOF reception.
- Time stamp sent in last two data bytes.

27.3. Function overview

Figure 27-1. CAN module block diagram shows the CAN block diagram.

Figure 27-1. CAN module block diagram



27.3.1. Working mode

The CAN interface has three working modes:

- Sleep working mode.
- Initial working mode.
- Normal working mode.

Sleep working mode

Sleep working mode is the default mode after reset. In sleep working mode, the CAN is in the low-power status and the CAN clock is stopped.

When SLPWMOD bit in CAN_CTL register is set, the CAN enters the sleep working mode. Then the SLPWS bit in CAN_STAT register is set by hardware.

To leave sleep working mode automatically: the AWU bit in CAN_CTL register is set and the CAN bus activity is detected. To leave sleep working mode by software: clear the SLPWMOD bit in CAN_CTL register.

Sleep working mode to initial working mode: set IWMOD bit and clear SLPWMOD bit in CAN_CTL register.

Sleep working mode to normal working mode: clear IWMOD and SLPWMOD bit in CAN_CTL register.

Initial working mode

When the configuration of CAN bus communication is needed to be changed, the CAN must enter initial working mode.

When IWMOD bit in CAN_CTL register is set, the CAN enters the initial working mode. Then the IWS bit in CAN_STAT register is set.

Initial working mode to sleep working mode: set SLPWMOD bit and clear IWMOD bit in CAN_CTL register.

Initial working mode to normal working mode: clear IWMOD bit and clear SLPWMOD bit in CAN_CTL register.

Normal working mode

The CAN could communicate with other CAN communication nodes in normal working mode.

To enter normal working mode: clear IWMOD and SLPWMOD bit in CAN_ CTL register.

Normal working mode to sleep working mode: set SLPWMOD bit in CAN_CTL register and wait the current transmission or reception completed.

Normal working mode to initial working mode: set IWMOD bit in CAN_CTL register, and wait the current transmission or reception completed.

27.3.2. Communication modes

The CAN interface has four communication modes:

- Silent communication mode.
- Loopback communication mode.
- Loopback and silent communication mode.
- Normal communication mode.

Silent communication mode

Silent communication mode means reception available and transmission disable.

The RX pin of the CAN could detect the signal from the network and the TX pin always holds logical one.

When the SCMOD bit in CAN_BT register is set, the CAN enters the silent communication mode. When it is cleared, the CAN leaves silent communication mode.

Silent communication mode is useful for monitoring the network messages.

Loopback communication mode

Loopback communication mode means the transmitted messages are transferred into the Rx FIFOs, the RX pin is disconnected from the CAN network and the TX pin can still send messages to the CAN network.

Setting LCMOD bit in CAN_BT register to enter loopback communication mode, while clearing it to leave. Loopback communication mode is useful for self-test.

Loopback and silent communication mode

Loopback and silent communication mode means the RX and TX pins are disconnected from the CAN network while the transmitted messages are transferred into the Rx FIFOs.

Setting LCMOD and SCMOD bit in CAN_BT register to enter loopback and silent communication mode, while clearing them to leave.

Loopback and silent communication mode is used for self-test. The TX pin holds in recessive state. The RX pin holds in high impedance state.

Normal communication mode

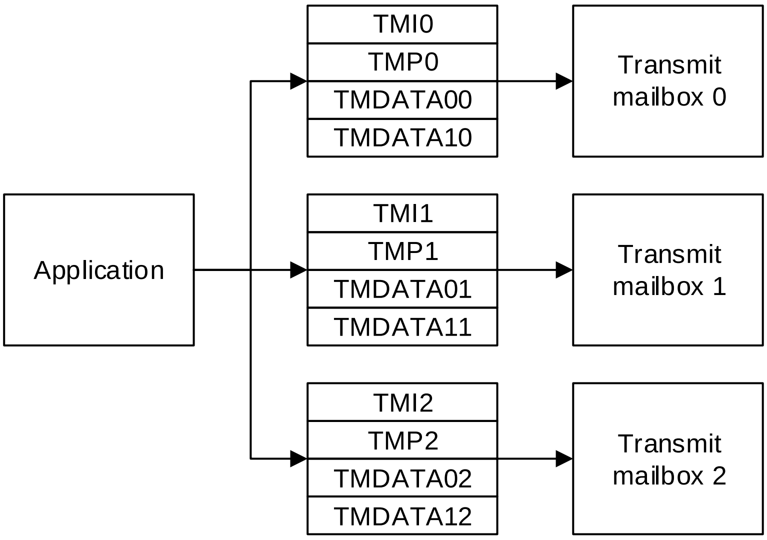
Normal communication mode is the default communication mode when the LCMOD and SCMOD bits in CAN_BT register are cleared.

27.3.3. Data transmission

Transmission register

Three transmit mailboxes are used for the application. Transmit mailboxes are used by configuring four transmission registers: CAN_TMIx, CAN_TMPx, CAN_TMDATA0x and CAN_TMDATA1x. As is shown in ***Figure 27-2. Transmission register***.

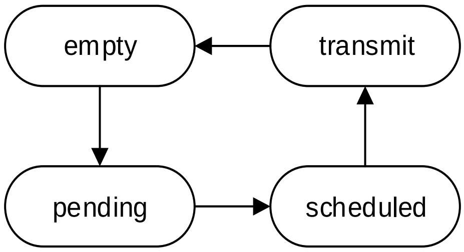
Figure 27-2. Transmission register



Transmit mailbox state

A transmit mailbox can be used when it is free (**empty state**). If the mailbox is filled with data, set TEN bit in CAN_TMIx register to prepare for starting the transmission **(pending state**). If more than one mailbox is in the pending state, they need scheduling the transmission (**scheduled state**). A mailbox with highest priority enters into transmit state and starts transmitting the message (**transmit state**). After the message has been sent, the mailbox is free (**empty state**). As is shown in ***Figure 27-3. State of transmit mailbox***.

Figure 27-3. State of transmit mailbox



Transmit status and error

The CAN_TSTAT register includes the transmit status and error bits: MTF, MTFNERR, MAL, MTE.

- MTF: mailbox transmit finished. Typically, MTF is set when the frame in the transmit mailbox has been sent.
- MTFNERR: mailbox transmit finished with no error. MTFNERR is set when the frame in the transmit mailbox has been sent without any error.
- MAL: mailbox arbitration lost. MAL is set when the frame transmission is failed due to the arbitration lost.
- MTE: mailbox transmit error. MTE is set when the frame transmission is failed due to the error detected on the CAN bus.

Steps of sending a frame

To send a frame through the CAN:
Step 1: Select one free transmit mailbox.
Step 2: Configure four transmission registers with the application's acquirement.
Step 3: Set TEN bit in CAN_TMIx register.
Step 4: Check the transmit status. Typically, MTF and MTFNERR are set if transmission is successful.

Transmission options

Abort

MST bit in CAN_TSTAT register can abort the transmission.

If the transmit mailbox's status is **pending** or **scheduled**, the abort of transmission can be done immediately.

In the **transmit** state, the abort of transmission does not take effect immediately until the transmission is finished. In case that the transmission is successful, the MTFNERR and MTF in CAN_TSTAT are set and state changes to be **empty**. In case that the transmission is failed, the state changes to be **scheduled** and then the abort of transmission can be done immediately.

Priority

When more than one transmit mailbox is pending, the transmission order is given by the TFO bit in CAN_CTL register.

In case that TFO is 1, the three transmit mailboxes work first-in first-out (FIFO).

In case that TFO is 0, the transmit mailbox with lowest identifier has the highest priority of transmission. If the identifiers are equal, the lower mailbox number will be scheduled firstly.

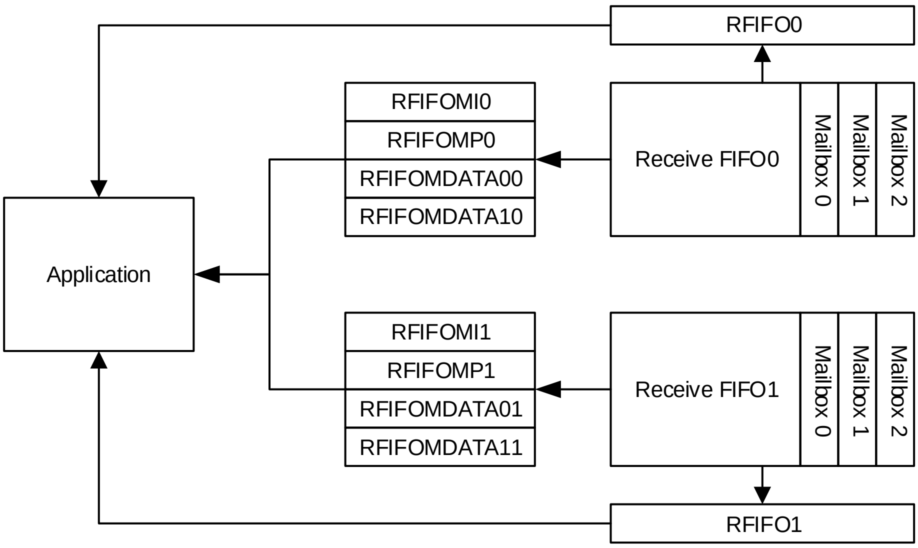
27.3.4. Data reception

Reception register

Two Rx FIFOs are used for the application. Rx FIFOs are managed by five registers: CAN_RFIFOx, CAN_RFIFOMIx, CAN_RFIFOMPx, CAN_RFIFOMDATA0x and CAN_RFIFOMDATA1x. FIFO's status and operation can be handled by CAN_RFIFOx register. Reception frame data can be achieved through the registers: CAN_RFIFOMIx, CAN_RFIFOMPx, CAN_RFIFOMDATA0x and CAN_RFIFOMDATA1x.

Each FIFO consists of three receive mailboxes. As is shown in *Figure 27-4. Reception register*.

Figure 27-4. Reception register



Rx FIFO

Rx FIFO has three mailboxes. The reception frames are stored in the mailbox according to the arriving sequence. First arrived frame can be accessed by application firstly.

The number of frames in the Rx FIFO and the status can be accessed by the register CAN_RFIFO0 and CAN_RFIFO1.

If at least one frame has been stored in the Rx FIFO0, the frame data is stored in the

CAN_RFIFOMI0, CAN_RFIFOMP0, CAN_RFIFOMDATA00 and CAN_RFIFOMDATA10 registers. After reading the current frame, set RFD bit in CAN_RFIFO0 to release a frame in the Rx FIFO and the software can read the next frame.

Rx FIFO status

RFL (Rx FIFO length) bits in CAN_RFIFOx register is 0 when no frame is stored in the Rx FIFO and it is 3 when FIFOx is full.

When RFF bit in CAN_RFIFOx register is set, it indicates FIFOx is full, at this time, RFL is 3.

When a new frame arrives after the FIFO has held three frames, the RFO bit in CAN_RFIFOx register will be set, and it indicates FIFOx is overrun. If the RFOD bit in CAN_CTL register is set, the new frame is discarded. If the RFOD bit in CAN_CTL register is reset, the new frame is stored into the Rx FIFO and the last frame in the Rx FIFO is discarded.

Steps of receiving a message

Step 1: Check the number of frames in the Rx FIFO.

Step 2: Read CAN_RFIFOMIx, CAN_RFIFOMPx, CAN_RFIFOMDATA0x and CAN_RFIFOMDATA1x.

Step 3: Set the RFD bit in CAN_RFIFOx register.

27.3.5. Filtering function

The CAN receives frames from the CAN bus. If the frame passes the filter, it is stored in the Rx FIFOs. Otherwise, the frame will be discarded without intervention by the software.

The identifier of frame is used for the matching of the filter.

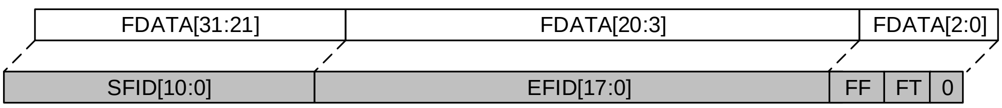
Scale

In GD32L235xx, the filter consists of 28 banks: bank0 to bank27. Each bank has two 32-bit registers: CAN_FxDATA0 and CAN_FxDATA1.

Each filter bank can be configured to 32-bit or 16-bit.

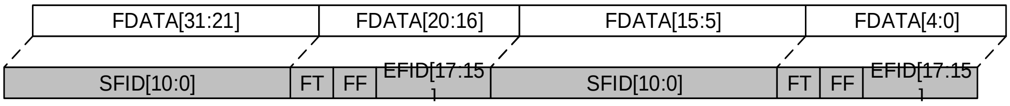
32-bit: SFID[10:0], EFID[17:0], FF and FT bits. As is shown in ***Figure 27-5. 32-bit filter***.

Figure 27-5. 32-bit filter



16-bit: SFID [10:0], FT, FF and EFID[17:15] bits. As is shown in ***Figure 27-6. 16-bit filter***.

Figure 27-6. 16-bit filter



Mask mode

For the Identifier of a data frame to be filtered, the mask mode is used to specify which bits must be the same as the preset Identifier and which bits need not be judged. 32-bit mask mode example is shown in ***Figure 27-7. 32-bit mask mode filter***.

Figure 27-7. 32-bit mask mode filter

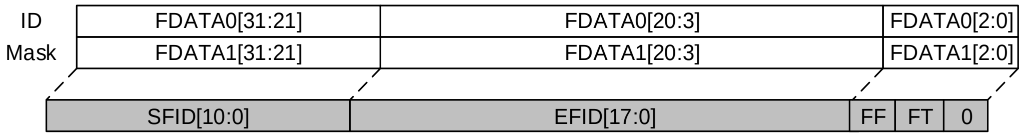
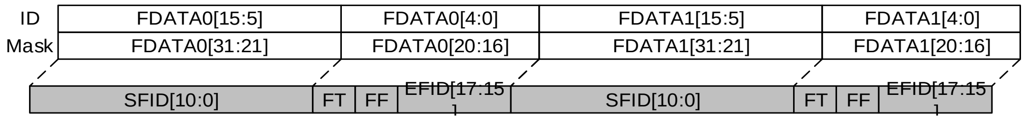


Figure 27-8. 16-bit mask mode filter



List mode

The filter consists of frame identifiers. The filter can determine whether a frame will be discarded or not. When one frame arrived, the filter will check which member can match the identifier of the frame.

32-bit list mode example is shown in ***Figure 27-9. 32-bit list mode filter***.

Figure 27-9. 32-bit list mode filter

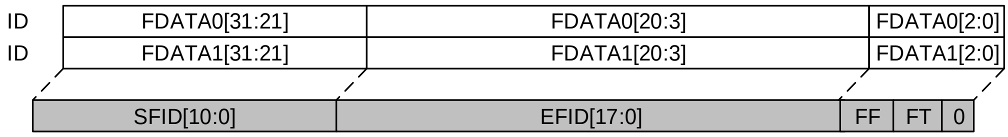
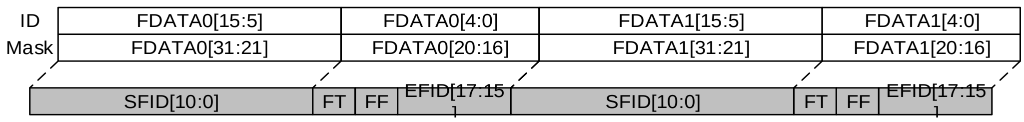


Figure 27-10. 16-bit list mode filter



Filter number

Filter consists of some filter bank. According to the mode and the scale of each of the filter banks, filter has different effects.

For example, there are two filter banks. Bank0 is configured as 32-bit mask mode. Bank1 is configured as 32-bit list mode. The filter number is shown in ***Table 27-1. 32-bit filter number***.

Table 27-1. 32-bit filter number

Filter bank	Filter data register	Filter number
0	F0DATA0-32bit-ID	0
	F0DATA1-32bit-Mask	
1	F1DATA0-32bit-ID	1
	F1DATA1-32bit-ID	2

Associated FIFO

28 banks can be associated with FIFO0 or FIFO1. If the bank is associated with FIFO0, the frames passed the bank will be stored in the FIFO0.

Active

The filter bank needs to be activated if the bank is to be used, otherwise, the filter bank should be left deactivated.

Filtering index

Each filter number corresponds to a filtering rule. When the frame which is associated with a filter number N passes the filters, the filter index is N. It stores in the FI bits in CAN_RFIFOMPx.

Filter bank has filter index once it is associated with the FIFO no matter whether the bank is active or not.

The example about filtering index is shown in ***Table 27-2. Filtering index***.

Table 27-2. Filtering index

<table>
<tr><th>Filter bank</th><th>FIFO0</th><th>Active</th><th>Filter nunber</th><th>Filter bank</th><th>FIFO1</th><th>Active</th><th>Filter nunber</th></tr>
<tr><td rowspan="2">0</td><td>F0DATA0-32bits-ID</td><td rowspan="2">Yes</td><td rowspan="2">0</td><td rowspan="4">2</td><td>F2DATA0[15:0]-16bits-ID</td><td rowspan="4">Yes</td><td rowspan="2">0</td></tr>
<tr><td>F0DATA1-32bits-Mask</td><td>F2DATA0[31:16]-16bits-Mask</td></tr>
<tr><td rowspan="2">1</td><td>F1DATA0-32bits-ID</td><td rowspan="2">Yes</td><td>1</td><td>F2DATA1[15:0]-16bits-ID</td><td rowspan="2">1</td></tr>
<tr><td>F1DATA1-32bits-ID</td><td>2</td><td>F2DATA1[31:16]-16bits-Mask</td></tr>
<tr><td rowspan="4">3</td><td>F3DATA0[15:0]-16bits-ID</td><td rowspan="4">No</td><td rowspan="2">3</td><td rowspan="2">4</td><td>F4DATA0-32bits-ID</td><td rowspan="2">No</td><td rowspan="2">2</td></tr>
<tr><td>F3DATA0[31:16]-16bits-Mask</td><td>F4DATA1-32bits-Mask</td></tr>
<tr><td>F3DATA1[15:0]-16bits-ID</td><td rowspan="2">4</td><td rowspan="2">5</td><td>F5DATA0-32bits-ID</td><td rowspan="2">No</td><td>3</td></tr>
<tr><td>F3DATA1[31:16]-16bits-Mask</td><td>F5DATA1-32bits-ID</td><td>4</td></tr>
<tr><td rowspan="4">7</td><td>F7DATA0[15:0]-16bits-ID</td><td rowspan="4">No</td><td>5</td><td rowspan="4">6</td><td>F6DATA0[15:0]-16bits-ID</td><td rowspan="4">Yes</td><td>5</td></tr>
<tr><td>F7DATA0[31:16]-16bits-ID</td><td>6</td><td>F6DATA0[31:16]-16bits- ID</td><td>6</td></tr>
<tr><td>F7DATA1[15:0]-16bits-ID</td><td>7</td><td>F6DATA1[15:0]-16bits-ID</td><td>7</td></tr>
<tr><td>F7DATA1[31:16]-16bits-ID</td><td>8</td><td>F6DATA1[31:16]-16bits- ID</td><td>8</td></tr>
<tr><td rowspan="4">8</td><td>F8DATA0[15:0]-16bits-ID</td><td rowspan="4">Yes</td><td>9</td><td rowspan="4">10</td><td>F10DATA0[15:0]-16bits-ID</td><td rowspan="4">No</td><td rowspan="2">9</td></tr>
<tr><td>F8DATA0[31:16]-16bits-ID</td><td>10</td><td>F10DATA0[31:16]-16bits-Mask</td></tr>
<tr><td>F8DATA1[15:0]-16bits-ID</td><td>11</td><td>F10DATA1[15:0]-16bits-ID</td><td rowspan="2">10</td></tr>
<tr><td>F8DATA1[31:16]-16bits-ID</td><td>12</td><td>F10DATA1[31:16]-16bits-Mask</td></tr>
<tr><td rowspan="4">9</td><td>F9DATA0[15:0]-16bits-ID</td><td rowspan="4">Yes</td><td rowspan="2">13</td><td rowspan="4">11</td><td>F11DATA0[15:0]-16bits-ID</td><td rowspan="4">No</td><td>11</td></tr>
<tr><td>F9DATA0[31:16]-16bits-Mask</td><td>F11DATA0[31:16]-16bits- ID</td><td>12</td></tr>
<tr><td>F9DATA1[15:0]-16bits-ID</td><td rowspan="2">14</td><td>F11DATA1[15:0]-16bits-ID</td><td>13</td></tr>
<tr><td>F9DATA1[31:16]-16bits-Mask</td><td>F11DATA1[31:16]-16bits- ID</td><td>14</td></tr>
<tr><td rowspan="2">12</td><td>F12DATA0-32bits-ID</td><td rowspan="2">Yes</td><td rowspan="2">15</td><td rowspan="2">13</td><td>F13DATA0-32bits-ID</td><td rowspan="2">Yes</td><td>15</td></tr>
<tr><td>F12DATA1-32bits-Mask</td><td>F13DATA1-32bits- ID</td><td>16</td></tr>
</table>

Priority

The filters have the priority rules:

1. 32-bits mode is higher than 16-bits mode.
2. List mode is higher than mask mode.
3. Smaller filter number has the higher priority.

27.3.6. Time-triggered communication

The time-triggered CAN protocol is a higher layer protocol on top of the CAN data link layer. Time-triggered communication means that activities are triggered by the elapsing of time segments. In a time-triggered communication system, all time points of message transmission are pre-defined.

In this mode, an internal 16-bit counter starts working, incrementing by 1 at each CAN bit time. This internal counter provides time stamps for sending and receiving data, stored in registers CAN _RFIFOMPx and CAN_TMPx.

The automatic retransmission is disabled in the time-triggered CAN communication.

27.3.7. Communication parameters

Automatic retransmission forbid mode

In time-triggered communication mode, the requirement for automatic retransmission must be disabled and can be met by setting ARD position 1 of the CAN_CTL register.

In this mode, the data is sent only once, and if the transmission fails due to arbitration failure or bus error, the CAN bus controller does not automatically resend the data as usual.

At the end of sending, the MTF bit of register CAN_TSTAT is hardware set to 1, and the sending status information can be obtained via MTFNERR, MAL, and MTE.

Bit time

On the bit-level, the CAN protocol uses synchronous bit transmission. This not only enhances the transmitting capacity but also requires a sophisticated method of bit synchronization. While bit synchronization in a character-oriented transmission (asynchronous) is performed upon the reception which the start bit is available with each character, the synchronous transmission protocol just need one start bit available at the beginning of a frame. To ensure that the receiver correctly reads the messages, resynchronization is required. Phase buffer segments' sample point of the front-end and back-end should be inserted a bit interval.

The CAN protocol regulates bus access by bit-wise arbitration. The signal propagated from sender to receiver and back to the sender must be completed within one bit-time. For synchronization, in addition to the phase buffer segments, a propagation delay segment is needed. The propagation delay segment is regarded as signal delays caused by transmitting and receiving nodes in the process of the signal propagation on the bus.

The normal bit time from the CAN protocol has three segments as follows:

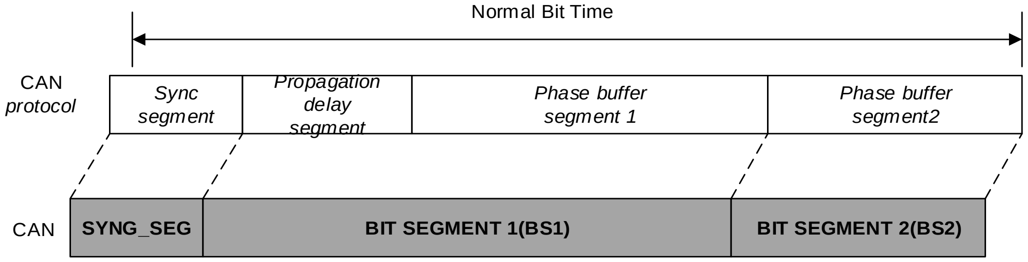
Synchronization segment (SYNC_SEG): a bit change is expected to occur within this time segment. It has a fixed length of one time quantum ($1 \times t_q$).

Bit segment 1 (BS1): It defines the location of the sample point. It includes the Propagation delay segment and Phase buffer segment 1 in the CAN standard. Its duration is programmable from 1 to 16 time quanta but it may be automatically lengthened to compensate for positive phase drifts due to different frequency of the various nodes of the network.

Bit segment 2 (BS2): It defines the location of the transmit point. It represents the Phase buffer segment 2 in the CAN standard. Its duration is programmable from 1 to 8 time quanta but it may also be automatically shortened to compensate for negative phase drifts.

The bit time is shown as in the ***Figure 27-11. The bit time***.

Figure 27-11. The bit time



The resynchronization Jump Width (SJW): it can be lengthened or shortened to compensate for the Synchronization error of the CAN network node. It is programmable from 1 to 4 time quanta.

A valid edge is defined as the first toggle in a bit time from dominant to recessive bus level before the controller sends a recessive bit.

If a valid edge is detected in BS1, not in SYNC_SEG, BS1 is added up to SJW maximumly, so that the sample point is delayed.

Conversely, if a valid edge is detected in BS2, not in SYNC_SEG, BS2 is cut down to SJW at most, so that the transmit point is moved earlier.

Baud rate

The clock of the CAN derives from the APB1 bus. The CAN calculates its baud rate as follow:

$$\mathrm{BaudRate}=\frac{1}{\mathrm{Normal\ Bit\ Time}} \qquad (22\text{-}1)$$

$$\mathrm{Normal\ Bit\ Time}=t_{SYNC\_SEG}+t_{BS1}+t_{BS2} \qquad (22\text{-}2)$$

with:

$$t_{SYNC\_SEG}=1\times t_q \qquad (22\text{-}3)$$

$$t_{BS1}=(1+\mathrm{BT.BS1})\times t_q \qquad (22\text{-}4)$$

$$t_{BS2}=(1+\mathrm{BT.BS2})\times t_q \qquad (22\text{-}5)$$

$$t_q=(1+\mathrm{BT.BAUDPSC})\times t_{PCLK1} \qquad (22\text{-}6)$$

27.3.8. Error flags

The state of CAN bus can be reflected by Transmit Error Counter (TECNT) and Receive Error Counter (RECNT) of CAN_ERR register. The value can be increased or decreased by the hardware according to the error, and the software can judge the stability of the CAN network by these values. For details on incorrect counting, refer to the CAN protocol section.

By using the CAN_INTEN register (ERRIE bit, etc.), the software can control the interrupt generation when error is detected.

Bus-Off recovery

The CAN controller is in Bus-Off state when TECNT is over than 255. In This state, BOERR bit is set in CAN_ERR register, and no longer able to transmit and receive messages.

According to the ABOR configuration in register CAN_CTL, there are two ways to recover from Bus-Off (to an error active state). Both of these methods require the CAN bus controller in the Bus-Off state to detect the Bus-Off recovery sequence defined by CAN protocol (when CAN_RX detects 128 consecutive 11-bit recessive bits) before automatic recovery.

If ABOR is set, it will be automatically recovered when a Bus-Off recovery sequence is detected.

If ABOR is cleared, CAN controller must be configured to enter initialization mode by setting IWMOD bit in CAN_CTL register, then exit and enter nomal mode. After this operation, it will recover when the recovering sequence is detected.

27.3.9. CAN interrupts

The CAN bus controller occupies 4 interrupt vectors, which are controlled by the register CAN _INTEN.

The interrupt sources can be classified as:

- Transmit interrupt
- FIFO0 interrupt
- FIFO1 interrupt
- Error and status change interrupt

Transmit interrupt

The transmit interrupt can be generated by any of the following conditions and TMEIE bit in CAN_INTEN register will be set:

- TX mailbox 0 transmit finished: MTF0 bit in the CAN_TSTAT register is set.
- TX mailbox 1 transmit finished: MTF1 bit in the CAN_TSTAT register is set.
- TX mailbox 2 transmit finished: MTF2 bit in the CAN_TSTAT register is set.

Receive FIFO0 interrupt

The Rx FIFO0 interrupt can be generated by the following conditions:

- Rx FIFO0 not empty: RFL0 bits in the CAN_RFIFO0 register are not ‘00’ and RFNEIE0 in CAN_INTEN register is set.
- Rx FIFO0 full: RFF0 bit in the CAN_RFIFO0 register is set and RFFIE0 in CAN_INTEN register is set.
- Rx FIFO0 overrun: RFO0 bit in the CAN_RFIFO0 register is set and RFOIE0 in CAN_INTEN register is set.

Rx FIFO1 interrupt

The Rx FIFO1 interrupt can be generated by the following conditions:

- Rx FIFO1 not empty: RFL1 bits in the CAN_RFIFO1 register are not ‘00’ and RFNEIE1 in CAN_INTEN register is set.
- Rx FIFO1 full: RFF1 bit in the CAN_RFIFO1 register is set and RFFIE1 in CAN_INTEN register is set.
- Rx FIFO1 overrun: RFO1 bit in the CAN_RFIFO1 register is set and RFOIE1 in CAN_INTEN register is set.

Error and working mode change interrupt

The error and working mode change interrupt can be generated by the following conditions:

- Error: ERRIF bit in the CAN_STAT register and ERRIE bit in the CAN_INTEN register are set. Refer to ERRIF description in the CAN_STAT register.
- Wakeup: WUIF bit in the CAN_STAT register is set and WIE bit in the CAN_INTEN register is set.
- Enter sleep working mode: SLPIF bit in the CAN_STAT register is set and SLPWIE bit in the CAN_INTEN register is set.

27.4. CAN registers

CAN0 base address: 0x4000 6400

27.4.1. Control register (CAN_CTL)

Address offset: 0x00

Reset value: 0x0001 0002

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															DFZ
															rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SWRST	Reserved							TTC	ABOR	AWU	ARD	RFOD	TFO	SLPWMOD	IWMOD
rs								rw	rw	rw	rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31:17	Reserved	Must be kept at reset value.
16	DFZ	Debug freeze If the CANx_HOLD in DBG_CTL0 register is set, this bit defines the CAN controller is in debug freezing mode or normal working mode. If the CANx_HOLD in DBG_CTL0 register is cleared, this bit takes no effect. 0: CAN reception and transmission work normal even during debug 1: CAN reception and transmission stop working during debug
15	SWRST	Software reset 0: No effect 1: Reset CAN to enter sleep working mode. This bit is automatically reset to 0.
14:8	Reserved	Must be kept at reset value.
7	TTC	Time-triggered communication 0: Disable time-triggered communication 1: Enable time-triggered communication
6	ABOR	Automatic Bus-Off recovery 0: The Bus-Off state is left manually by software 1: The Bus-Off state is left automatically by hardware
5	AWU	Automatic wakeup If this bit is set, the CAN leaves sleep working mode when CAN bus activity is detected, and SLPWMOD bit in CAN_CTL register will be cleared automatically. 0: The sleeping working mode is left manually by software

		1: The sleeping working mode is left automatically by hardware
4	ARD	Automatic retransmission disable 0: Enable automatic retransmission 1: Disable automatic retransmission
3	RFOD	Rx FIFO overwrite disable 0: Enable Rx FIFO overwrite when Rx FIFO is full and overwrite the FIFO with the incoming frame 1: Disable Rx FIFO overwrite when Rx FIFO is full and discard the incoming frame
2	TFO	Tx FIFO order 0: Order with the identifier of the frame (the smaller identifier has higher priority) 1: Order with first-in and first-out
1	SLPWMOD	Sleep working mode If this bit is set by software, the CAN enters sleep working mode after current transmission or reception is completed. This bit can be cleared by software or hardware. If AWU bit in CAN_CTL register is set, this bit is cleared by hardware when CAN bus activity is detected. 0: Disable sleep working mode 1: Enable sleep working mode
0	IWMOD	Initial working mode 0: Disable initial working mode 1: Enable initial working mode

27.4.2. Status register (CAN_STAT)

Address offset: 0x04

Reset value: 0x0000 0C02

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				RXL	LASTRX	RS	TS	Reserved			SLPIF	WUIF	ERRIF	SLPWS	IWS
				r	r	r	r				rc_w1	rc_w1	rc_w1	r	r

Bits	Fields	Descriptions
31:12	Reserved	Must be kept at reset value.
11	RXL	RX level
10	LASTRX	Last sample value of RX pin

9	RS	Receiving state 0: CAN is not working in the receiving state 1: CAN is working in the receiving state
8	TS	Transmitting state 0: CAN is not working in the transmitting state 1: CAN is working in the transmitting state
7:5	Reserved	Must be kept at reset value.
4	SLPIF	Status change interrupt flag of entering sleep working mode This bit is set by hardware when entering sleep working mode, and cleared by hardware when the CAN is not in sleep working mode. This bit can also be cleared by software when writting 1 to this bit. 0: CAN is not in the sleep working mode 1: CAN is in the sleep working mode
3	WUIF	Status change interrupt flag of waking up from sleep working mode This bit is set when CAN bus activity event is detected in sleep working mode. This bit can be cleared by software when writting 1 to this bit. 0: Wakeup event is not coming 1: Wakeup event is coming
2	ERRIF	Error interrupt flag This bit is set by the following events. The BOERR bit in CAN_ERR register is set and BOIE bit in CAN_INTEN register is set. Or the PERR bit in CAN_ERR register is set and PERRIE bit in CAN_INTEN register is set. Or the WERR bit in CAN_ERR register is set and WERRIE bit in CAN_INTEN register is set. Or the ERRN bits in CAN_ERR register are set to 1 to 6 (not 0 and not 7) and ERRNIE in CAN_INTEN register is set. This bit is cleared by software when writting 1 to this bit. 0: No error interrupt event 1: Any error interrupt event has happened
1	SLPWS	Sleep working state This bit is set by hardware when the CAN enters sleep working mode after setting SLPWMOD bit in CAN_CTL register. If the CAN leaves normal working mode to sleep working mode, it must wait the current frame transmission or reception to be completed. This bit is cleared by hardware when the CAN leaves sleep working mode. Clear SLPWMOD bit in CAN_CTL register or automatically detect the CAN bus activity when AWU bit is set in CAN_CTL register. If leaving sleep working mode to normal working mode, this bit will be cleared after receiving 11 consecutive recessive bits from the CAN bus. 0: CAN is not in the state of sleep working mode 1: CAN is in the state of sleep working mode
0	IWS	Initial working state

		This bit is set by hardware when the CAN enters initial working mode after setting IWMOD bit in CAN_CTL register. If the CAN leaves normal working mode to initial working mode, it must wait the current frame transmission or reception to be completed. This bit is cleared by hardware when the CAN leaves initial working mode after clearing IWMOD bit in CAN_CTL register. If leaving initial working mode to normal working mode, this bit will be cleared after receiving 11 consecutive recessive bits from the CAN bus. 0: CAN is not in the state of initial working mode 1: CAN is in the state of initial working mode

27.4.3. Transmit status register (CAN_TSTAT)

Address offset: 0x08

Reset value: 0x1C00 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
TMLS2	TMLS1	TMLS0	TME2	TME1	TME0	NUM[1:0]		MST2	Reserved			MTE2	MAL2	MTFNERR2	MTF2
r	r	r	r	r	r	r		rs				rc_w1	rc_w1	rc_w1	rc_w1

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MST1	Reserved			MTE1	MAL1	MTFNERR1	MTF1	MST0	Reserved			MTE0	MAL0	MTFNERR0	MTF0
rs				rc_w1	rc_w1	rc_w1	rc_w1	rs				rc_w1	rc_w1	rc_w1	rc_w1

Bits	Fields	Descriptions
31	TMLS2	Transmit mailbox 2 last sending in Tx FIFO This bit is set by hardware when transmit mailbox 2 has the last sending order in the Tx FIFO with at least two frames pending.
30	TMLS1	Transmit mailbox 1 last sending in Tx FIFO This bit is set by hardware when transmit mailbox 1 has the last sending order in the Tx FIFO with at least two frames pending.
29	TMLS0	Transmit mailbox 0 last sending in Tx FIFO This bit is set by hardware when transmit mailbox 0 has the last sending order in the Tx FIFO with at least two frames pending.
28	TME2	Transmit mailbox 2 empty 0: Transmit mailbox 2 not empty 1: Transmit mailbox 2 empty
27	TME1	Transmit mailbox 1 empty 0: Transmit mailbox 1 not empty 1: Transmit mailbox 1 empty

26	TME0	Transmit mailbox 0 empty 0: Transmit mailbox 0 not empty 1: Transmit mailbox 0 empty
25:24	NUM[1:0]	These bits are the number of the Tx FIFO mailbox in which the frame will be transmitted if at least one mailbox is empty. These bits are the number of the Tx FIFO mailbox in which the frame will be transmitted at last if all mailboxes are full.
23	MST2	Mailbox 2 stop transmitting This bit is set by the software to stop mailbox 2 transmitting. This bit is reset by the hardware while the mailbox 2 is empty.
22:20	Reserved	Must be kept at reset value.
19	MTE2	Mailbox 2 transmit error This bit is set by hardware when the transmit error occurs. This bit is reset by writing 1 to this bit or MTF2 bit in CAN_TSTAT register. This bit is reset by hardware when next transmit starts.
18	MAL2	Mailbox 2 arbitration lost This bit is set when the arbitration lost occurs. This bit is reset by writting 1 to this bit or MTF2 bit in CAN_TSTAT register. This bit is reset by hardware when next transmit starts.
17	MTFNERR2	Mailbox 2 transmit finished with no error This bit is set when the transmission finishes and no error occurs. This bit is reset by writting 1 to this bit or MTF2 bit in CAN_TSTAT register. This bit is reset by hardware when the transmission finishes with error. 0: Mailbox 2 transmit finished with error 1: Mailbox 2 transmit finished with no error
16	MTF2	Mailbox 2 transmit finished This bit is set by hardware when the transmission finishes or aborts. This bit is reset by writting 1 to this bit or TEN bit in CAN_TMI2 is 1. 0: Mailbox 2 transmit is progressing 1: Mailbox 2 transmit finished
15	MST1	Mailbox 1 stop transmitting This bit is set by software to stop mailbox 1 transmitting. This bit is reset by hardware when the mailbox 1 is empty.
14:12	Reserved	Must be kept at reset value.
11	MTE1	Mailbox 1 transmit error This bit is set by hardware when the transmit error occurs. This bit is reset by writting 1 to this bit or MTF1 bit in CAN_TSTAT register. This bit is reset by hardware when next transmit starts.

10	MAL1	Mailbox 1 arbitration lost This bit is set when the arbitration lost occurs. This bit is reset by writting 1 to this bit or MTF1 bit in CAN_TSTAT register. This bit is reset by hardware when next transmit starts.
9	MTFNERR1	Mailbox 1 transmit finished with no error This bit is set when the transmission finishes and no error occurs. This bit is reset by writting 1 to this bit or MTF1 bit in CAN_TSTAT register. This bit is reset by hardware when the transmission finishes with error. 0: Mailbox 1 transmit finished with error 1: Mailbox 1 transmit finished with no error
8	MTF1	Mailbox 1 transmit finished This bit is set by hardware when the transmission finishes or aborts. This bit is reset by writting 1 to this bit or TEN bit in CAN_TMI1 is 1. 0: Mailbox 1 transmit is progressing 1: Mailbox 1 transmit finished
7	MST0	Mailbox 0 stop transmitting This bit is set by the software to stop mailbox 0 transmitting. This bit is reset by the hardware when the mailbox 0 is empty.
6:4	Reserved	Must be kept at reset value.
3	MTE0	Mailbox 0 transmit error This bit is set by hardware when the transmit error occurs. This bit is reset by writting 1 to this bit or MTF0 bit in CAN_TSTAT register. This bit is reset by hardware when next transmit starts.
2	MAL0	Mailbox 0 arbitration lost This bit is set when the arbitration lost occurs. This bit is reset by writting 1 to this bit or MTF0 bit in CAN_TSTAT register. This bit is reset by hardware when next transmit starts.
1	MTFNERR0	Mailbox 0 transmit finished with no error This bit is set when the transmission finishes and no error occurs. This bit is reset by writting 1 to this bit or MTF0 bit in CAN_TSTAT register. This bit is reset by hardware when the transmission finishes with error. 0: Mailbox 0 transmit finished with error 1: Mailbox 0 transmit finished with no error
0	MTF0	Mailbox 0 transmit finished This bit is set by hardware when the transmission finishes or aborts. This bit is reset by writting 1 to this bit or TEN bit in CAN_TMI0 is 1. 0: Mailbox 0 transmit is progressing 1: Mailbox 0 transmit finished

27.4.4. Receive message FIFO0 register (CAN_RFIFO0)

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										RFD0	RFO0	RFF0	Reserved	RFL0[1:0]	
										rs	rc_w1	rc_w1		r	

Bits	Fields	Descriptions
31:6	Reserved	Must be kept at reset value.
5	RFD0	Rx FIFO0 dequeue This bit is set by software to start dequeuing a frame from Rx FIFO0. This bit is reset by hardware when the dequeuing is done.
4	RFO0	Rx FIFO0 overfull This bit is set by hardware when Rx FIFO0 is overfull and reset by software when writting 1 to this bit. 0: The Rx FIFO0 is not overfull 1: The Rx FIFO0 is overfull
3	RFF0	Rx FIFO0 full This bit is set by hardware when Rx FIFO0 is full and reset by software when writting 1 to this bit. 0: The Rx FIFO0 is not full 1: The Rx FIFO0 is full
2	Reserved	Must be kept at reset value.
1:0	RFL0[1:0]	Rx FIFO0 length These bits are the length of the Rx FIFO0.

27.4.5. Receive message FIFO1 register (CAN_RFIFO1)

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										RFD1	RFO1	RFF1	Reserved	RFL1[1:0]	
										rs	rc_w1	rc_w1		r	

Bits	Fields	Descriptions
31:6	Reserved	Must be kept at reset value.
5	RFD1	Rx FIFO1 dequeue This bit is set by software to start dequeuing a frame from Rx FIFO1. This bit is reset by hardware when the dequeuing is done.
4	RFO1	Rx FIFO1 overfull This bit is set by hardware when Rx FIFO1 is overfull and reset by writting 1 to this bit. 0: The Rx FIFO1 is not overfull 1: The Rx FIFO1 is overfull
3	RFF1	Rx FIFO1 full This bit is set by hardware when Rx FIFO1 is full and reset by writting 1 to this bit. 0: The Rx FIFO1 is not full 1: The Rx FIFO1 is full
2	Reserved	Must be kept at reset value.
1:0	RFL1[1:0]	Rx FIFO1 length These bits are the length of the Rx FIFO1.

27.4.6. Interrupt enable register (CAN_INTEN)

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														SLPWIE	WIE
														rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ERRIE	Reserved			ERRNIE	BOIE	PERRIE	WERRIE	Reserved	RFOIE1	RFFIE1	RFNEIE1	RFOIE0	RFFIE0	RFNEIE0	TMEIE
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	SLPWIE	Sleep working interrupt enable 0: Sleep working interrupt disabled

		1: Sleep working interrupt enabled
16	WIE	Wakeup interrupt enable 0: Wakeup interrupt disabled 1: Wakeup interrupt enabled
15	ERRIE	Error interrupt enable 0: Error interrupt disabled 1: Error interrupt enabled
14:12	Reserved	Must be kept at reset value.
11	ERRNIE	Error number interrupt enable 0: Error number interrupt disabled 1: Error number interrupt enabled
10	BOIE	Bus-Off interrupt enable 0: Bus-Off interrupt disabled 1: Bus-Off interrupt enabled
9	PERRIE	Passive error interrupt enable 0: Passive error interrupt disabled 1: Passive error interrupt enabled
8	WERRIE	Warning error interrupt enable 0: Warning error interrupt disabled 1: Warning error interrupt enabled
7	Reserved	Must be kept at reset value.
6	RFOIE1	Rx FIFO1 overfull interrupt enable 0: Rx FIFO1 overfull interrupt disabled 1: Rx FIFO1 overfull interrupt enabled
5	RFFIE1	Rx FIFO1 full interrupt enable 0: Rx FIFO1 full interrupt disabled 1: Rx FIFO1 full interrupt enabled
4	RFNEIE1	Rx FIFO1 not empty interrupt enable 0: Rx FIFO1 not empty interrupt disabled 1: Rx FIFO1 not empty interrupt enabled
3	RFOIE0	Rx FIFO0 overfull interrupt enable 0: Rx FIFO0 overfull interrupt disabled 1: Rx FIFO0 overfull interrupt enabled
2	RFFIE0	Rx FIFO0 full interrupt enable 0: Rx FIFO0 full interrupt disabled 1: Rx FIFO0 full interrupt enabled
1	RFNEIE0	Rx FIFO0 not empty interrupt enable

		0: Rx FIFO0 not empty interrupt disabled 1: Rx FIFO0 not empty interrupt enabled
0	TMEIE	Transmit mailbox empty interrupt enable 0: Transmit mailbox empty interrupt disabled 1: Transmit mailbox empty interrupt enabled

27.4.7. Error register (CAN_ERR)

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
RECNT[7:0]								TECNT[7:0]							
r								r							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved									ERRN[2:0]			Reserved	BOERR	PERR	WERR
									rw				r	r	r

Bits	Fields	Descriptions
31:24	RECNT[7:0]	Receive error count defined by the CAN standard
23:16	TECNT[7:0]	Transmit error count defined by the CAN standard
15:7	Reserved	Must be kept at reset value.
6:4	ERRN[2:0]	Error number These bits indicate the error status of bit transformation. They are updated by hardware. When the bit transformation is successful, they are equal to 0. 000: No error 001: Stuff error 010: Form error 011: Acknowledgment error 100: Bit recessive error 101: Bit dominant error 110: CRC error 111: Set by software
3	Reserved	Must be kept at reset value.
2	BOERR	Bus-Off error Whenever the CAN enters Bus-Off state, the bit will be set by hardware.
1	PERR	Passive error Whenever the TECNT or RECNT is greater than 127, the bit will be set by

		hardware.
0	WERR	Warning error Whenever the TECNT or RECNT is greater than or equal to 96, the bit will be set by hardware.

27.4.8. Bit timing register (CAN_BT)

Address offset: 0x1C

Reset value: 0x0123 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
SCMOD	LCMOD	Reserved				SJW[1:0]		Reserved	BS2[2:0]			BS1[3:0]			
rw	rw					rw			rw			rw			

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

Bits	Fields	Descriptions
31	SCMOD	Silent communication mode 0: Silent communication disabled 1: Silent communication enabled
30	LCMOD	Loopback communication mode 0: Loopback communication disabled 1: Loopback communication enabled
29:26	Reserved	Must be kept at reset value.
25:24	SJW[1:0]	Resynchronization jump width Resynchronization jump width time quantum= SJW[1:0]+1
23	Reserved	Must be kept at reset value.
22:20	BS2[2:0]	Bit segment 2 Bit segment 2 time quantum = BS2[2:0]+1
19:16	BS1[3:0]	Bit segment 1 Bit segment 1 time quantum = BS1[3:0]+1
15:10	Reserved	Must be kept at reset value.
9:0	BAUDPSC[9:0]	Baud rate prescaler The CAN baud rate prescaler

27.4.9. Transmit mailbox identifier register (CAN_TMIx) (x = 0…2)

Address offset: 0x180, 0x190, 0x1A0

Reset value: 0xXXXX XXXX (bit0 = 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
SFID[10:0]/EFID[28:18]											EFID[17:13]				
rw											rw				

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

Bits	Fields	Descriptions
31:21	SFID[10:0]/EFID[28:18]	The frame identifier SFID[10:0]: Standard format frame identifier EFID[28:18]: Extended format frame identifier
20:16	EFID[17:13]	The frame identifier EFID[17:13]: Extended format frame identifier
15:3	EFID[12:0]	The frame identifier EFID[12:0]: Extended format frame identifier
2	FF	Frame format 0: Standard format frame 1: Extended format frame
1	FT	Frame type 0: Data frame 1: Remote frame
0	TEN	Transmit enable This bit is set by software when one frame will be transmitted and reset by hardware when the transmit mailbox is empty. 0: Transmit disabled 1: Transmit enabled

27.4.10. Transmit mailbox property register (CAN_TMPx) (x=0..2)

Address offset: 0x184, 0x194, 0x1A4

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

TS[15:0]
rw

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

Bits	Fields	Descriptions
31:16	TS[15:0]	Time stamp The time stamp of frame in transmit mailbox.
15:9	Reserved	Must be kept at reset value.
8	TSEN	Time stamp enable 0: Time stamp disabled 1: Time stamp enabled. The TS[15:0] will be transmitted in the DB6 and DB7 in DL. This bit is available when the TTC bit in CAN_CTL is set.
7:4	Reserved	Must be kept at reset value.
3:0	DLENC[3:0]	Data length code DLENC[3:0] is the number of bytes in a frame.

27.4.11. Transmit mailbox data0 register (CAN_TMDATA0x) (x = 0...2)

Address offset: 0x188, 0x198, 0x1A8

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
DB3[7:0]								DB2[7:0]							
rw								rw							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DB1[7:0]								DB0[7:0]							
rw								rw							

Bits	Fields	Descriptions
31:24	DB3[7:0]	Data byte 3
23:16	DB2[7:0]	Data byte 2
15:8	DB1[7:0]	Data byte 1
7:0	DB0[7:0]	Data byte 0

27.4.12. Transmit mailbox data1 register (CAN_TMDATA1x) (x=0..2)

Address offset: 0x18C, 0x19C, 0x1AC

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
DB7[7:0]								DB6[7:0]							
rw								rw							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DB5[7:0]								DB4[7:0]							
rw								rw							

Bits	Fields	Descriptions
31:24	DB7[7:0]	Data byte 7
23:16	DB6[7:0]	Data byte 6
15:8	DB5[7:0]	Data byte 5
7:0	DB4[7:0]	Data byte 4

27.4.13. Receive FIFO mailbox identifier register (CAN_RFIFOMIx) (x = 0, 1)

Address offset: 0x1B0, 0x1C0

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
SFID[10:0]/EFID[28:18]											EFID[17:13]				
r											r				

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

Bits	Fields	Descriptions
31:21	SFID[10:0]/EFID[28:18]	The frame identifier SFID[10:0]: Standard format frame identifier EFID[28:18]: Extended format frame identifier
20:16	EFID[17:13]	The frame identifier EFID[17:13]: Extended format frame identifier
15:3	EFID[12:0]	The frame identifier

		EFID[12:0]: Extended format frame identifier
2	FF	Frame format 0: Standard format frame 1: Extended format frame
1	FT	Frame type 0: Data frame 1: Remote frame
0	Reserved	Must be kept at reset value.

27.4.14. Receive FIFO mailbox property register (CAN_RFIFOMPx) (x = 0, 1)

Address offset: 0x1B4, 0x1C4

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
TS[15:0]															
r															

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

Bits	Fields	Descriptions
31:16	TS[15:0]	Time stamp The time stamp of frame in transmit mailbox.
15:8	FI[7:0]	Filtering index The index of the filter which the frame passes.
7:4	Reserved	Must be kept at reset value.
3:0	DLENC[3:0]	Data length code DLENC[3:0] is the number of bytes in a frame.

27.4.15. Receive FIFO mailbox data0 register (CAN_RFIFOMDATA0x) (x = 0, 1)

Address offset: 0x1B8, 0x1C8

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
DB3[7:0]								DB2[7:0]							
r								r							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DB1[7:0]								DB0[7:0]							
r								r							

Bits	Fields	Descriptions
31:24	DB3[7:0]	Data byte 3
23:16	DB2[7:0]	Data byte 2
15:8	DB1[7:0]	Data byte 1
7:0	DB0[7:0]	Data byte 0

27.4.16. Receive FIFO mailbox data1 register (CAN_RFIFOMDATA1x) (x = 0, 1)

Address offset: 0x1BC, 0x1CC

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
DB7[7:0]								DB6[7:0]							
r								r							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DB5[7:0]								DB4[7:0]							
r								r							

Bits	Fields	Descriptions
31:24	DB7[7:0]	Data byte 7
23:16	DB6[7:0]	Data byte 6
15:8	DB5[7:0]	Data byte 5
7:0	DB4[7:0]	Data byte 4

27.4.17. Filter control register (CAN_FCTL)

Address offset: 0x200

Reset value: 0x2A1C 1C01

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	FLD
	rw

Bits	Fields	Descriptions
31:1	Reserved	Must be kept at reset value.
0	FLD	Filter lock disable 0: Filter lock enabled 1: Filter lock disabled

27.4.18. Filter mode configuration register (CAN_FMCFG)

Address offset: 0x204

Reset value: 0x0000 0000

This register has to be accessed by word(32-bit). This register can be modified only when FLD bit in CAN_FCTL register is set.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved				FMOD27	FMOD26	FMOD25	FMOD24	FMOD23	FMOD22	FMOD21	FMOD20	FMOD19	FMOD18	FMOD17	FMOD16
				rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FMOD15	FMOD14	FMOD13	FMOD12	FMOD11	FMOD10	FMOD9	FMOD8	FMOD7	FMOD6	FMOD5	FMOD4	FMOD3	FMOD2	FMOD1	FMOD0
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.
27:0	FMODx	Filter mode 0: Filter x with mask mode 1: Filter x with list mode

27.4.19. Filter scale configuration register (CAN_FSCFG)

Address offset: 0x20C

Reset value: 0x0000 0000

This register has to be accessed by word(32-bit). This register can be modified only when FLD bit in CAN_FCTL register is set.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved				FS27	FS26	FS25	FS24	FS23	FS22	FS21	FS20	FS19	FS18	FS17	FS16
				rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FS15	FS14	FS13	FS12	FS11	FS10	FS9	FS8	FS7	FS6	FS5	FS4	FS3	FS2	FS1	FS0

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.
27:0	FSx	Filter scale 0: Filter x with 16-bit scale 1: Filter x with 32-bit scale

27.4.20. Filter associated FIFO register (CAN_FAFIFO)

Address offset: 0x214

Reset value: 0x0000 0000

This register has to be accessed by word(32-bit). This register can be modified only when FLD bit in CAN_FCTL register is set.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved				FAF27	FAF26	FAF25	FAF24	FAF23	FAF22	FAF21	FAF20	FAF19	FAF18	FAF17	FAF16
				rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FAF15	FAF14	FAF13	FAF12	FAF11	FAF10	FAF9	FAF8	FAF7	FAF6	FAF5	FAF4	FAF3	FAF2	FAF1	FAF0
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.
27:0	FAFx	Filter associated FIFO 0: Filter x associated with FIFO0 1: Filter x associated with FIFO1

27.4.21. Filter working register (CAN_FW)

Address offset: 0x21C

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				FW27	FW26	FW25	FW24	FW23	FW22	FW21	FW20	FW19	FW18	FW17	FW16
				rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FW15	FW14	FW13	FW12	FW11	FW10	FW9	FW8	FW7	FW6	FW5	FW4	FW3	FW2	FW1	FW0
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.
27:0	FWx	Filter working 0: Filter x working disabled 1: Filter x working enabled

27.4.22. Filter x data y register (CAN_FxDATAy) (x = 0…27, y = 0, 1)

Address offset: 0x240 + 8 * x + 4 * y, (x = 0…27, y = 0, 1)

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
FD31	FD30	FD29	FD28	FD27	FD26	FD25	FD24	FD23	FD22	FD21	FD20	FD19	FD18	FD17	FD16
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FD15	FD14	FD13	FD12	FD11	FD10	FD9	FD8	FD7	FD6	FD5	FD4	FD3	FD2	FD1	FD0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31:0	FDx	Filter data Mask mode 0: Mask match disable 1: Mask match enable List mode 0: List identifier bit is 0 1: List identifier bit is 1

28. Universal Serial Bus full-speed device interface (USBD)

28.1. Overview

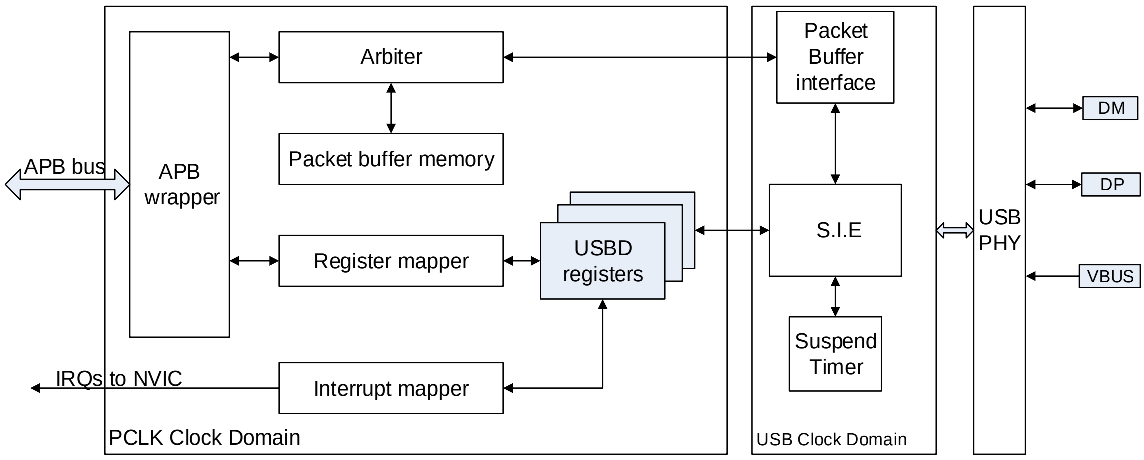
The Universal Serial Bus full-speed device interface (USBD) module provides a device solution for implementing a USB 2.0 full-speed compliant peripheral. It contains a full-speed internal USB PHY and no more external PHY chip is needed. USBD supports all the four types of transfer (control, bulk, interrupt and isochronous) defined in USB 2.0 protocol.

28.2. Main features

- USB 2.0 full-speed device controller.
- Support up to 8 configurable bidirectional endpoints.
- Support double-buffered bulk/isochronous endpoints.
- Support USB 2.0 Link Power Management.
- Each endpoint supports control, bulk, isochronous or interrupt transfer types (exclude endpoint 0, endpoint 0 only support control transfer).
- Support USB suspend/resume operations.
- Dedicated 512-byte SRAM used for data packet buffer.
- Integrated USB PHY.
- USBD connect / disconnect capability (controllable embedded pull-up resistor on DP line).

28.3. Block diagram

Figure 28-1. USBD block diagram



28.4. Signal description

Table 28-1. USBD signal description

I/O port	Type	Description
VBUS	Input	Bus power port
DM	Input / Output	Differential data line – port
DP	Input / Output	Differential data line + port

Note: As soon as the USBD is enabled, these pins are connected to the USBD internal transceiver automatically.

28.5. Clock configuration

According to the USB standard definition, the USB full-speed module adopt fixed 48MHz clock. It is necessary to configure two clock for using USBD, one is the USB controller clock, its frequency must be configured to 48MHz, and the other one is the APB1 to USB interface clock which is also APB1 bus clock.

Note: For GD32L233xx device, the frequency of the APB1 bus clock must be not less than 24MHz. For GD32L235xx device, the frequency of the APB1 bus clock must be not less than 12MHz.

48MHz clock of USB controller can be generated by dividing MCU internal or external crystal oscillator by a programmable prescaler, then multiplicating the frequency through PLL.

- Regard two frequency division of 16MHz internal oscillator as the input of the PLL, then 6 frequencies doubling the clock.
- Regard 8MHz external oscillator as the input of the PLL, then 6 frequencies doubling the clock.

Note: Regardless of using internal or external crystal oscillator to generate USB clock, the clock accuracy must reach ±500ppm. If the accuracy of the USB clock cannot meet the condition, data transfer may not conform to the requirements of the USB specification, and even it may cause USB not working directly.

28.6. Function overview

28.6.1. USB endpoints

USBD supports 8 USB endpoints that can be individually configured.

Each endpoint supports:

- Single/Double buffer (endpoint 0 can’t use double buffer).
- One endpoint buffer descriptor.

- Programmable buffer starting address and buffer length.
- Configurable response to a packet.
- Control transfer (only for endpoint 0).

Endpoint buffer

The function of the device operation is to transfer a request in the memory image to and from the Universal Serial Bus. To efficiently manage USB endpoint communications, USBD implements a dedicated data packet buffer of 512-bytes SRAM memory accessed directly by the USB peripheral. It is mapped to the APB1 peripheral memory, from 0x4000 6000 to 0x4000 6400. The total capacity is 1KB, but USBD uses actually only 512 bytes for the bus width reason.

Each endpoint can be associated with one or two data packet buffers used to store the current data payload. The bidirectional endpoint has usually two buffers, one is used for transmission and the other one is for reception. The mono-directional endpoint only has one buffer for data operation.

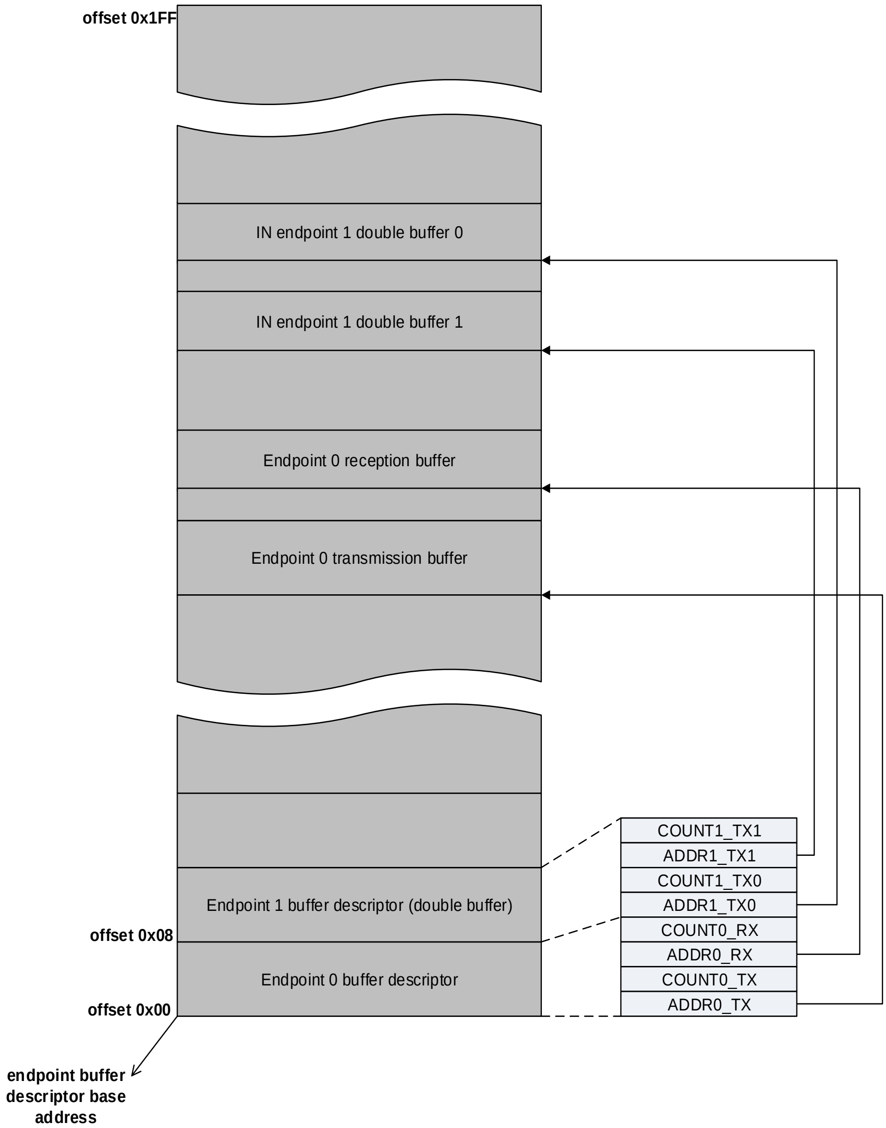
Note: For GD32L235xx device, the USBD and CAN share the dedicated 512-byte SRAM memory.

Endpoint buffer descriptor table

USBD implements an endpoint buffer descriptor table which defines the buffer address and length, and the table is also located in the endpoint data packet buffer. The endpoint buffer descriptor is used as a communication port between the application firmware and the SIE in system memory. Every endpoint direction requires two 16-bit words buffer descriptor. Therefore, each table entry includes 4 16-bit words (Tx and Rx two direction) and is aligned to 8-byte boundary. When an endpoint is double-buffered, the SIE will use the two buffers in ping-pong operation mode. The endpoint buffer descriptor table is pointed to by the USBD endpoint buffer address register.

The relationship between endpoint buffer descriptor table entries and packet buffer areas is depicted in ***Figure 28-2. An example with buffer descriptor table usage (USBD_BADDR = 0)***.

Figure 28-2. An example with buffer descriptor table usage (USBD_BADDR = 0)



Note: This figure is not drawn on the actual scale, and it is addressed through the USB bus 16-bit mode.

Double-buffered endpoints

The double-buffered feature is used to improve bulk transfer performance. To implement the new flow control scheme, the USB peripheral should know which packet buffer is currently in use by the application software, so to be aware of any conflict. Since in the USBD_EpxCS register, there are two data toggle bits (TX_DTG and RX_DTG) but only one is used by USBD for hardware data handling (due to the unidirectional constraint required by double-buffering feature), the other one can be used by the application software to show which buffer it is currently using. This new buffer flag is called software buffer bit (SW_BUF). In ***Table 28-2. Double-buffering buffer flag definition***, the correspondence between USBD_EpxCS register bits and DTG/SW_BUF definition is explained.

Table 28-2. Double-buffering buffer flag definition

Buffer flag	Tx endpoint	Rx endpoint
DTG	TX_DTG (USBD_EpxCS bit 6)	RX_DTG (USBD_EpxCS bit 14)
SW_BUF	RX_DTG (USBD_EpxCS bit 14)	TX_DTG (USBD_EpxCS bit 6)

The DTG bit and the SW_BUF bit are responsible for the flow control. When a transfer completes, the USB peripheral toggle the DTG bit; when the data have been copied, the application software need to toggle the SW_BUF bit. Except for the first time, if the value of DTG bit is equal to the SW_BUF's, the transfer will pause, and the host is NAK. When the two bits are not equal, the transfer resume.

Table 28-3. Double buffer usage

Endpoint Type	DTOG	SW_BUF	Packet buffer used by the USB peripheral	Packet buffer used by the application software
OUT	0	1	EPxRBADDR / EPxRBCNT buffer description table locations.	EPxTBADDR / EPxTBCNT buffer description table locations.
	1	0	EPxTBADDR / EPxTBCNT buffer description table locations.	EPxRBADDR / EPxRBCNT buffer description table locations.
IN	0	1	EPxTBADDR / EPxTBCNT buffer description table locations.	EPxRBADDR / EPxRBCNT buffer description table locations.
	1	0	EPxRBADDR / EPxRBCNT buffer description table locations.	EPxTBADDR / EPxTBCNT buffer description table locations.

Endpoint memory requests arbitration

As the USBD is connected to the APB1 bus through an APB1 interface, so USB APB1 interface will accept memory requests coming from the APB1 bus and from the USB interface. The arbiter will resolve the conflicts by giving priority to APB1 accesses, while always reserving half of the memory bandwidth to complete all USB transfers. This time-duplex scheme implements a virtual dual-port SRAM that allows memory access, when an USB transaction is happening. Multiword APB1 transfers of any length are also allowed by this scheme.

28.6.2. Operation procedure

USB transaction process

After the endpoint is configured and a transaction is required, the hardware will detect the token packet. When a token is recognized by the USBD, the data transfer is performed. When all the data has been transferred, the proper handshake packet over the USBD is generated or expected according to the direction of the transfer.

After the transaction process is completed, an endpoint-specific interrupt is generated. In the interrupt routine, the application can process it accordingly.

Transaction formatting is performed by the hardware, including CRC generation and checking.

Once the endpoint is enabled, endpoint control and status register, buffer address and COUNT filed should not be modified by the application software. When the data transfer operation is completed, notified by a STIF interrupt event, they can be accessed again to re-enable a new operation.

IN transaction

When a configured and valid endpoint receives an IN token packet, it will send the data packet to the host. If the endpoint is not valid, a NAK or STALL handshake is sent according to the endpoint status.

In the data packet transfer process, a configured data PID will be sent firstly, then the actual data in endpoint buffer memory is loaded into the output shift register to be transmitted. After the data are sent, the computed CRC will be sent by hardware.

When receiving the ACK sent from the host, then the USB peripheral will toggle the data PID and set the endpoint status to be NAK. At the same time, the successful transfer interrupt will be triggered. In the interrupt service routine, application fill the data packet memory with data, start next transfer by re-enable the endpoint by setting the endpoint status VALID.

OUT and SETUP transaction

USBD handle OUT and SETUP tokens in similar way, the difference details about SETUP packets would be shown in the following section about control transfer.

After the received endpoint is configured and enabled, host will send OUT/SETUP token to the device. When receiving the token, USBD will access the endpoint buffer descriptor to initialize the endpoint buffer address and length. Then the received data bytes subsequently are packed in words (LSB mode) and transferred to the endpoint buffer. When detecting the end of data packet, the computed CRC and received CRC are compared. If no errors occur, an ACK handshake packet is sent to the host.

When the transaction is completed correctly, USBD will toggle the data PID and set the endpoint status to be NAK. Then the endpoint successful transfer interrupt will be triggered by hardware. In the interrupt service routine, the application can get the transaction type and read the received data from the endpoint buffer. After the received data is processed, the application should initiate further transactions by setting the endpoint status valid.

If any error happens during reception, the USBD set the error interrupt bit and still copy data into the packet memory buffer, but will not send the ACK packet. The USBD itself can recover from reception errors and continue to handle next transfer. The USBD never override outside the data buffer, which is controlled by the internal register configured. The received 2-byte CRC is also copied to the packet memory buffer, immediately following data bytes. If the length of data is greater than actually allocated length, the excess data are not copied. This is a buffer overrun situation. A STALL handshake is sent, and this transaction fails.

If an addressed endpoint is not valid, a NAK or STALL handshake packet is sent instead of the ACK, according to the endpoint status and no data is written to the endpoint data buffers.

Control transfers

Control transfers require that a SETUP transaction be started from the host to a device to describe the type of control access that the device should perform. The SETUP transaction is followed by zero or more control DATA transactions that carry the specific information for the requested access. Finally, a STATUS transaction completes the control transfer and allows the endpoint to return the status of the control transfer to the client software. After the STATUS transaction for a control transfer is completed, the host can advance to the next control transfer for the endpoint.

USBD always use endpoint 0 in two directions as default control endpoint to handle control transfers. It is aware of the number and direction of data stages by interpreting the contents of SETUP transaction, and is required to set the unused direction endpoint 0 status to STALL except the last data stage.

At the last data stage, the application software set the opposite direction endpoint 0 status to NAK. This will keep the host waiting for the completion of the control operation. If the operation completes successfully, the software will change NAK to VALID, otherwise to STALL. If the status stage is an OUT, the STATUS_OUT bit should be set, so that a status transaction with non-zero data will be answered STALL to indicate an error happen.

According to USB specification, device isn’t allowed to abort current command and then start new command, so that device must answer a SETUP packet with an ACK handshake packet, not with a NAK or STALL handshake packet.

When the configured control endpoint 0 receives a SETUP token, the USBD accepts the data, performing the required data transfers and sends back an ACK handshake. If there is unsuccessfully handling data transfer about previously issued request, the USB discard SETUP token and regard current condition as error, and then urge the host to send the request token again.

Isochronous transfers

Isochronous transfers can guarantee constant data rate and bounded latency, but do not support data retransmission in response to errors on the bus. Consequently, the isochronous transaction does not have a handshake phase, and have no ACK packet after the data packet. Data toggling is not supported, and DATA0 PID is only used to start a data packet.

The isochronous endpoint status only can be set DISABLED and VALID, any other value is illegal. The application software can implement double-buffering to improve performance. By swapping transmission and reception data packet buffer on each transaction, the application software can copy the data into or out of a buffer, at the same time the USB peripheral handle the data transmission or reception of data in another buffer. The DTOG bit indicates which buffer that the USB peripheral is currently using.

The application software initializes the DTOG according to the first buffer to be used. At the end of each transaction, the RX_ST or TX_ST bit is set, depending on the enabled direction regardless of CRC errors or buffer-overrun conditions (if errors occur, the ERRIF bit will be set). At the same time, The USB peripheral will toggle the DTOG bit, but will not affect the STAT bit.

28.6.3. USB events and interrupts

Each USB action is always initiated by the application software, driven by one USB interrupt or event. After system reset, the application needs to wait for a succession of USB interrupts and events.

Reset events

System and power-on reset

Upon system and power-on reset, the application software should first provide all required clock to the USB module and interface, then de-assert its reset signal so to be able to access its registers, last switch on the analog part of the device related to the USB transceiver.

The USB firmware should do as follows:

- Reset CLOSE bit in USBD_CTL register.
- Wait for the internal reference voltage to be stable.
- Clear SETRST bit in USBD_CTL register.
- Clear the USBD_INTF register to remove the spurious pending interrupt and then enable other unit.

USB reset (RESET interrupt)

When this event occurs, the USB peripheral status is the same as the moment system reset.

The USB firmware should do as follows:

- Set USBEN bit in USBD_DADDR register to enable USB module in 10ms.
- Initialize the USBD_EP0CS register and its related packet buffers.

Suspend and resume events

The USB module can be forced to place in low-power mode (SUSPEND mode) by writing in the USB control register (USBD_CTL) whenever required. At this time, all static power consumption is avoided and the USB clock can be slowed down or stopped. It will be resumed when detect activity at the USB bus while in low-power mode.

The USB protocol insists on power management by the USB device. This becomes even more important if the device draws power from the bus (bus-powered device). The following constraints should be met by the bus-powered device.

- A device in the non-configured state should draw a maximum of 100mA from the USB bus.
- A configured device can draw only up to what is specified in the Max Power field of the configuration descriptor. The maximum value is 500mA.
- A suspended device should draw a maximum of 500uA.

A device will go into the suspend state if there is no activity on the USB bus for more than 3ms. A suspended device wakes up, if RESUME signaling is detected.

USBD also supports software initiated remote wakeup. To initiate remote wakeup, the application software must enable all clocks and clear the suspend bit after MCU is waked up. This will cause the hardware to generate a remote wakeup signal upstream.

Setting the SETSPS bit to 1 enables the suspend mode, and it will disable the check of SOF reception. Setting the LOWM bit to 1 will shut down the static power consumption in the analog USB transceivers, but the RESUME signal is still able to be detected.

Link Power Management (LPM) level L1

In order to optimize power consumption in SUSPEND/RESUME state, USB 2.0 has achieved Link Power Management (LPM). LPM includes 4 states from L0 to L3. LPM L1 state (sleep state) is the new power management state.

A device will go into the L1 state if the host sends a successful LPM transaction. L1 does not impose any specific power draw requirements (from VBUS) on the attached device.

For more details, please refer to USB2_LinkPowerManagement_ECN.

USB Interrupts

USBD has three interrupts: low-priority interrupt, high-priority interrupt and wakeup interrupt. Software can configure these interrupts to route the interrupt condition to these entries in the NVIC table. An interrupt will be generated when both the interrupt status bit and the corresponding interrupt enable bit are set. The interrupt status bit is set by hardware if the interrupt condition occurs (irrespective of the interrupt enable bit).

- Low-priority interrupt (Channel 20): triggered by all USB events.
- High-priority interrupt (Channel 19): triggered only by a correct transfer event for isochronous and double-buffer bulk transfer.
- Wakeup interrupt (Channel 42): triggered by the wakeup events.

28.6.4. Operation guide

This section describes the operation guide for USBD.

USBD register initialization sequence

1. Clear the CLOSE bit in USBD_CTL register, then clear the SETRST bit.

2. Clear USBD_INTF register to remove any spurious pending interrupt.
3. Program USBD_BADDR register to set endpoint buffer base address.
4. Set USBD_CTL register to enable interrupts.
5. Wait for the reset interrupt (RSTIF).
6. In the reset interrupt, initialize default control endpoint 0 to start enumeration process and program USBD_BADDR to set the device address to 0 and enable USB module function.
7. Configure endpoint 0 and prepare to receive SETUP packet.

Endpoint initialization sequence

1. Program USBD_EPxTBADDR or USBD_EPxRBADDR registers with transmission or reception data buffer address.
2. Program the EP_CTL and EP_KCTL bits in USBD_EpxCS register to set endpoint type and buffer kind according to the endpoint usage.
3. If the endpoint is a single buffer endpoint:
   1) Initialize the endpoint data toggle bit by programming the TX_DTG or RX_DTG bit in USBD_EpxCS register, but endpoint 0 needs to set them to 1 and 0 respectively for control transfer.
   2) Configure endpoint status by programming the TX_STA bit or RX_STA bit in USBD_EpxCS register, but both of them are set to ’10 (NAK) if use endpoint 0 to initialize the control transfer.

   If the endpoint is a double buffer endpoint:

   1) Both transmission and reception toggle fields need to be programmed. If the endpoint is a Tx endpoint, clear the TX_DTG and RX_DTG bit in USBD_EpxCS register, or if endpoint is a Rx endpoint, it needs to toggle TX_DTG bit.
   2) Program USBD_EPxTBCNT and USBD_EPxRBCNT register to set transfer data bit count.
   3) Endpoint transmission and reception status both need to be configured. If the endpoint is a Tx endpoint, set the TX_STA bit to be NAK and RX_STA bit to be DISABLED, or the endpoint is a Rx endpoint, set the RX_STA bit to be VALID and TX_STA bit to be DISABLED.

SETUP and OUT data transfers

1. Program USBD_EPxRBCNT register to set BLKSIZ and EPRXCNT filed, these filed defines the endpoint buffer length.
2. Configure the endpoint status to be VALID to enable the endpoint to receive data by programming USBD_EpxCS register.

3. Wait for successful transfer interrupt (STIF).
4. In the interrupt handler, application can get the transaction type by reading the STEUP bit in USBD_EpxCS register. Then application will read the data payload from the endpoint data buffer with the start address defined in USBD_EPxRBAR register. Last application will interpret the data and process the corresponding transaction.

IN data transfers

1. Program USBD_EPxTBCNT register to set EPTXCNT filed, this filed defines the endpoint buffer length.
2. Configure the endpoint status to be VALID to enable the endpoint to transmit data by programming USBD_EpxCS register.
3. Wait for successful transfer interrupt (STIF).
4. In the interrupt handler, application needs to update user buffer length and location pointer. Then application fill the endpoint buffer with user buffer data. Last application will configure the endpoint status to be VALID to start next transfer.

28.7. Registers definition

USBD base address: 0x4000 5C00

28.7.1. USBD control register (USBD_CTL)

Address offset: 0x40

Reset value: 0x0000 0003

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
STIE	PMOUIE	ERRIE	WKUPIE	SPSIE	RSTIE	SOFIE	ESOFIE	L1REQIE	Reserved	L1RSREQ	RSREQ	SETSPS	LOWM	CLOSE	SETRST
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	STIE	Successful transfer interrupt enable. 0: Successful transfer interrupt disabled. 1: Interrupt generated when STIF bit in USBD_INTF register is set.
14	PMOUIE	Packet memory overrun/underrun interrupt enable. 0: No interrupt generated when packet memory overrun / underrun. 1: Interrupt generated when PMOUIF bit in USBD_INTF register is set.
13	ERRIE	Error interrupt enable. 0: Error interrupt disabled 1: Interrupt generated when ERRIF bit in USBD_INTF register is set.
12	WKUPIE	Wakeup interrupt enable 0: Wakeup interrupt disabled 1: Interrupt generated when WKUPIF bit in USBD_INTF register is set.
11	SPSIE	Suspend state interrupt enable 0: Suspend state interrupt disabled 1: Interrupt generated when SPSIF bit in USB_IFR register is set.
10	RSTIE	USB reset interrupt enable. 0: USB reset interrupt disabled 1: Interrupt generated when RSTIF bit in USBD_INTF register is set.
9	SOFIE	Start of frame interrupt enable

		0: Start of frame interrupt disabled 1: Interrupt generated when SOFIF bit in USBD_INTF register is set.
8	ESOFIE	Expected start of frame interrupt enable 0: Expected start of frame interrupt disabled 1: Interrupt generated when ESOFIF bit in USBD_INTF register is set.
7	L1REQIE	LPM L1 state request interrupt enable 0: LPM L1 state request interrupt disabled 1: Interrupt generated when L1REQ bit in USBD_INTF register is set.
6	Reserved	Must be kept at reset value.
5	L1RSREQ	LPM L1 resume request MCU can set this bit to send a LPM L1 resume signal to the host. After the signaling ends, this bit is cleared by hardware.
4	RSREQ	Resume request The software set a resume request to the USB host, and the USB host should drive the resume sequence according the USB specifications 0: No resume request 1: Send resume request.
3	SETSPS	Set suspend The software should set suspend state when SPSIF bit in USBD_INTF register is set. 0: Not set suspend state. 1: Set suspend state.
2	LOWM	Low-power mode When set this bit, the USB goes to low-power mode at suspend state. If resume from suspend state, the hardware reset this bit. 0: No effect 1: Go to low-power mode at suspend state.
1	CLOSE	Close state When this bit is set, the USBD goes to close state, and completely close the USBD and disconnected from the host. 0: Not in close state 1: In close state.
0	SETRST	Set reset When this bit is set, the USBD peripheral should be reset. 0: No reset 1: A reset generated.

28.7.2. USBD interrupt flag register (USBD_INTF)

Address offset: 0x44

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
STIF	PMOUIF	ERRIF	WKUPIF	SPSIF	RSTIF	SOFIF	ESOFIF	L1REQ	Reserved		DIR	EPNUM[3:0]			
r	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0			r	r			

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15	STIF	Successful transfer interrupt flag This bit set by hardware when a successful transaction completes
14	PMOUIF	Packet memory overrun/underrun interrupt flag This bit set by hardware to indicate that the packet memory is inadequate to hold transfer data. The software writes 0 to clear this bit.
13	ERRIF	Error interrupt flag This bit set by hardware when an error happens during transaction. The software writes 0 to clear this bit.
12	WKUPIF	Wakeup interrupt flag This bit set by hardware in the SUSPEND state to indicate that activity is detected. The software writes 0 to clear this bit.
11	SPSIF	Suspend state interrupt flag When no traffic happen in 3ms, hardware set this bit to indicate a SUSPEND request. The software writes 0 to clear this bit.
10	RSTIF	USB reset interrupt flag Set by hardware when the USB RESET signal is detected. The software writes 0 to clear this bit.
9	SOFIF	Start of frame interrupt flag Set by hardware when a new SOF packet arrives, The software writes 0 to clear this bit.
8	ESOFIF	Expected start of frame interrupt flag Set by the hardware to indicate that a SOF packet is expected but not received. The software writes 0 to clear this bit.
7	L1REQ	Set by the hardware when LPM L1 transaction is successfully received and acknowledged. The software writes 0 to clear this bit.
6:5	Reserved	Must be kept at reset value.

4	DIR	Direction of transaction Set by the hardware to indicate the direction of the transaction 0: OUT type 1: IN type
3:0	EPNUM[3:0]	Endpoint Number Set by the hardware to identify the endpoint which the transaction is directed to

28.7.3. USBD status register (USBD_STAT)

Address offset: 0x48

Reset value: 0x0000 0XXX where X is undefined

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
RX_DP	RX_DM	LOCK	SOFLN[1:0]		FCNT[10:0]										
r	r	r	r		r										

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15	RX_DP	Receive data + line status Represent the status on the DP line
14	RX_DM	Receive data – line status Represent the status on the DM line
13	LOCK	Locked the USB Set by the hardware indicate that at the least two consecutive SOF have been received
12:11	SOFLN[1:0]	SOF lost number Increment every ESOFIF happens by hardware Cleared once the reception of SOF
10:0	FCNT[10:0]	Frame number counter The Frame number counter incremented every SOF received.

28.7.4. USBD device address register (USBD_DADDR)

Address offset: 0x4C

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								USBEN	USBDAR[6:0]						
								rw	rw						

Bits	Fields	Descriptions
31:8	Reserved	Must be kept at reset value.
7	USBEN	USB device enable Set by software to enable the USB device 0: The USB device disabled. No transactions handled. 1: The USB device enabled.
6:0	USBDAR[6:0]	USBD device address After bus reset, the address is reset to 0x00. If the enable bit is set, the device will respond on packets for function address DEV_ADDR

28.7.5. USBD buffer address register (USBD_BADDR)

Address offset: 0x50

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
BAR[12:0]													Reserved		
rw															

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:3	BAR[12:0]	Buffer address Start address of the allocation buffer(512byte on-chip SRAM), used for buffer descriptor table, packet memory
2:0	Reserved	Must be kept at reset value.

28.7.6. USBD endpoint x control and status register (USBD_EpxCS), x=[0..7]

Address offset: 0x00 to 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
RX_ST	RX_DTG	RX_STA[1:0]		SETUP	EP_CTL[1:0]		EP_KCTL	TX_ST	TX_DTG	TX_STA[1:0]		EP_ADDR[3:0]			
rc_w0	t	t		r	rw		rw	rc_w0	t	t		rw			

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15	RX_ST	Reception successful transferred Set by hardware when a successful OUT/SETUP transaction complete Cleared by software by writing 0
14	RX_DTG	Reception data PID toggle This bit represent the toggle data bit (0=DATA0,1=DATA1)for non-isochronous endpoint Used to implement the flow control for double-buffered endpoint Used to swap buffer for isochronous endpoint
13:12	RX_STA[1:0]	Reception status bits Toggle by writing 1 by software Remain unchanged by writing 0 Refer to ***Table 28-4. Reception status encoding***
11	SETUP	Setup transaction completed Set by hardware when a SETUP transaction completed.
10:9	EP_CTL[1:0]	Endpoint type control Refer to ***Table 28-5. Endpoint type encoding***
8	EP_KCTL	Endpoint kind control The exact meaning depends on the endpoint type Refer to ***Table 28-6. Endpoint kind meaning***
7	TX_ST	Transmission successful transfer Set by hardware when a successful IN transaction complete Clear by software
6	TX_DTG	Transmission data PID toggle This bit represent the toggle data bit (0=DATA0,1=DATA1)for non-isochronous endpoint Used to implement the flow control for double-buffered endpoint Used to swap buffer for isochronous endpoint

5:4	TX_STA[1:0]	Status bits, for transmission transfers Refer to *Table 28-7. Transmission status encoding*
3:0	EP_ADDR	Endpoint address Used to direct the transaction to the target endpoint

Table 28-4. Reception status encoding

RX_STA[1:0]	Meaning
00	**DISABLED:** ignore all reception requests of this endpoint
01	**STALL:** STALL handshake status
10	**NAK:** NAK handshake status
11	**VALID:** enable endpoint for reception

Table 28-5. Endpoint type encoding

EP_CTL[1:0]	Meaning
00	**BULK:** bulk endpoint
01	**CONTROL:** control endpoint
10	**ISO:** isochronous endpoint
11	**INTERRUPT:** interrupt endpoint

Table 28-6. Endpoint kind meaning

EP_CTL[1:0]		EP_KCTL Meaning
00	BULK	DBL_BUF
01	CONTROL	STATUS_OUT

Table 28-7. Transmission status encoding

TX_STA[1:0]	Meaning
00	**DISABLED:** ignore all transmission requests of this endpoint
01	**STALL:** STALL handshake status
10	**NAK:** NAK handshake status
11	**VALID:** enable endpoint for transmission

28.7.7. USBD endpoint x transmission buffer address register (USBD_EPxTBADDR), x can be in [0..7]

Address offset: [USBD_BADDR] + x * 16

USB local address: [USBD_BADDR] + x * 8

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
EPTXBAR[15:1]															EPTXBAR[0]

rw rw

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:1	EPTXBAR[15:1]	Endpoint transmission buffer address Start address of the packet buffer containing data to be sent when receive next IN token
0	EPTXBAR[0]	Must be set to 0

28.7.8. USBD endpoint x transmission buffer byte count register (USBD_EPxTBCNT), x can be in [0..7]

Address offset: [USBD_BADDR] + x * 16 + 4

USB local Address: [USBD_BADDR] + x * 8 + 2

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						EPTXCNT[9:0]									
						rw									

Bits	Fields	Descriptions
31:10	Reserved	Must be kept at reset value.
9:0	EPTXCNT[9:0]	Endpoint transmission byte count The number of bytes to be transmitted at next IN token

28.7.9. USBD endpoint x reception buffer address register (USBD_EPxRBADDR), x can be in [0..7]

Address offset: [USBD_BADDR] + x * 16 + 8

USB local Address: [USB_BADDR] + x * 8 + 4

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
EPRBAR[15:1]															EPRBAR[0]

rw rw

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:1	EPRBAR[15:1]	Endpoint reception buffer address Start address of packet buffer containing the data received by the endpoint at the next OUT/SETUP token
0	EPRBAR[0]	Must be set to 0

28.7.10. USBD endpoint x reception buffer byte count register (USBD_EPxRBCNT), x can be in [0..7]

Address offset: [USBD_BADDR] + x * 16 + 12

USB local Address: [USBD_BADDR] + x * 8 + 6

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
BLKSIZ	BLKNUM[4:0]					EPRCNT[9:0]									
rw	rw					r									

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15	BLKSIZ	Block size 0: block size is 2 bytes 1: block size is 32 bytes
14:10	BLKNUM[4:0]	Block number The number of blocks allocated to the packet buffer
9:0	EPRCNT[9:0]	Endpoint reception byte count The number of bytes to be received at next OUT/SETUP token

28.7.11. USBD LPM control and status register (USBD_LPMCS)

Address offset: 0x54

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												CHIRD			

rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				RHIRD				BLSTAT[3:0]				REMWK	Reserved	LPMACK	LPMEN
				r				r				r		rw	rw

Bits	Fields	Descriptions
31:20	Reserved	Must be kept at reset value.
19:16	CHIRD	Configured HIRD value
15:12	Reserved	Must be kept at reset value
11:8	RHIRD	Received HIRD value
7:4	BLSTAT[3:0]	bLinkState value This filed contain the bLinkState value received with last ACKed LPM token.
3	REMWK	bRemoteWake value This bit contains the bRemoteWake value received with last ACKed LPM token
2	Reserved	Must be kept at reset value.
1	LPMACK	LPM token acknowledge enable 0: the valid LPM token will be NYETed. 1: the valid LPM token will be ACKed. The NYET/ACK will be returned only on a successful LPM transaction: No errors in both the EXT token and the LPM token (else ERROR). A valid bLinkState = 0001B (L1) is received (else STALL)
0	LPMEN	LPM support enable This bit is set by the software to enable the LPM support within the USB device. If this bit is set to 0, no LPM transactions are handled.

28.7.12. USBD DP pull-up control register (USBD_DPC)

Address offset: 0x58

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
DPUEN	Reserved														
rw															

Bits	Fields	Descriptions

31:16	Reserved	Must be kept at reset value.
15	DPUEN	DP pull-up control. 0:Disable the embedded pull-up on the DP line, disconnect to host. 1:Enable the embedded pull-up on the DP line, connect to host.
14:0	Reserved	Must be kept at reset value

29. Document appendix

29.1. List of abbreviations used in registers

Table 29-1. List of abbreviations used in register

abbreviations for registers	Descriptions
read/write (rw)	Software can read and write to this bit.
Read-only ®	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)	The software can toggle this bit by writing 1. Writing 0 has no effect.

29.2. List of terms

Table 29-2. 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.
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.

29.3. Available peripherals

For availability of peripherals and their number across all MCU series types,refer to the corresponding device data datasheet.

30. Revision history

Table 30-1. Revision history

Revision No.	Description	Date
2.0	Initial Release	Jul. 2023
2.1	Delete some power saving modes supported by GD32L235 from the Power management unit (PMU)	Oct.30,2023
2.2	1. ***Section 1.3.1*** Added parity check support for GD32L235xx SRAM 2. Changed the start address 0x0803E000 on page 63 in ***Table 2.1*** to 0x0803F000 3. Revise the description of optional voltages for NPLDO and LPLDO in the ***Figure 3-2. Power supply overview of GD32L235xx devices*** 4. Modify the detailed description of LDO in Deep-sleep 1 and Deep-sleep 2 in the main characteristics section of ***3.2.*** 5. Added temperature adaptive mode (1.1V) function and usage scenario prompt to LDOVS control bit in GD32L235xx products in ***Section 3.4.1*** Control register 0 (PMU_CTL0) 6. Modify the system clock frequency range in Section ***3.3.4*** Run2 mode 7. Replace V_{REF-} and V_{REF+} with VREFN and VREFP in ***Section 14*** 8. Revise t_{US} to $t_{ST(ADC)}$ in ***Section 14.4.3*** 9. ***14.4.12. ADC Internal Channels***: Added GD32L235xx product internal temperature calculation formula 10. Modify the maximum delay generated by the analog filter in ***Section 22.3.1*** 11. Modify the VSRC bit field description in ***Section 25.4.1*** Control register (SLCD_CTL)	Mar. 2024

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 under the intellectual property laws and treaties of the People’s Republic of China and other jurisdictions worldwide. The Company reserves all rights under such laws and treaties and does not grant any license under its patents, copyrights, trademarks, or other intellectual property rights. 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.

The Company makes no warranty of any kind, express or implied, with regard to this document or any Product, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. The Company does not assume any liability arising out of the application or use of any Product described in this document. Any information provided in this document is provided only for reference purposes. It is the responsibility of the user of this document to properly design, program, and test the functionality and safety of any application made of this information and any resulting product. Except for customized products which has been expressly identified in the applicable agreement, the Products are designed, developed, and/or manufactured for ordinary business, industrial, personal, and/or household applications only. The Products are not designed, intended, or authorized for use as components in systems designed or intended for the operation of weapons, weapons systems, nuclear installations, atomic energy control instruments, combustion control instruments, airplane or spaceship instruments, transportation instruments, traffic signal instruments, life-support devices or systems, other medical devices or systems (including resuscitation equipment and surgical implants), pollution control or hazardous substances management, or other uses where the failure of the device or Product could cause personal injury, death, property or environmental damage ("Unintended Uses"). Customers shall take any and all actions to ensure using and selling the Products in accordance with the applicable laws and regulations. The Company is not liable, in whole or in part, and customers shall and hereby do release the Company as well as it’s suppliers and/or distributors from any claim, damage, or other liability arising from or related to all Unintended Uses of the Products. Customers shall indemnify and hold the Company as well as it’s 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 Products.

Information in this document is provided solely in connection with the Products. The Company reserves the right to make changes, corrections, modifications or improvements to this document and Products and services described herein at any time, without notice.