User Guide User Guide

MG32F04P032

Arm®Cortex® M0 Core

User Guide

Version 1.00

Date 2025/03/03

Contents

Figures

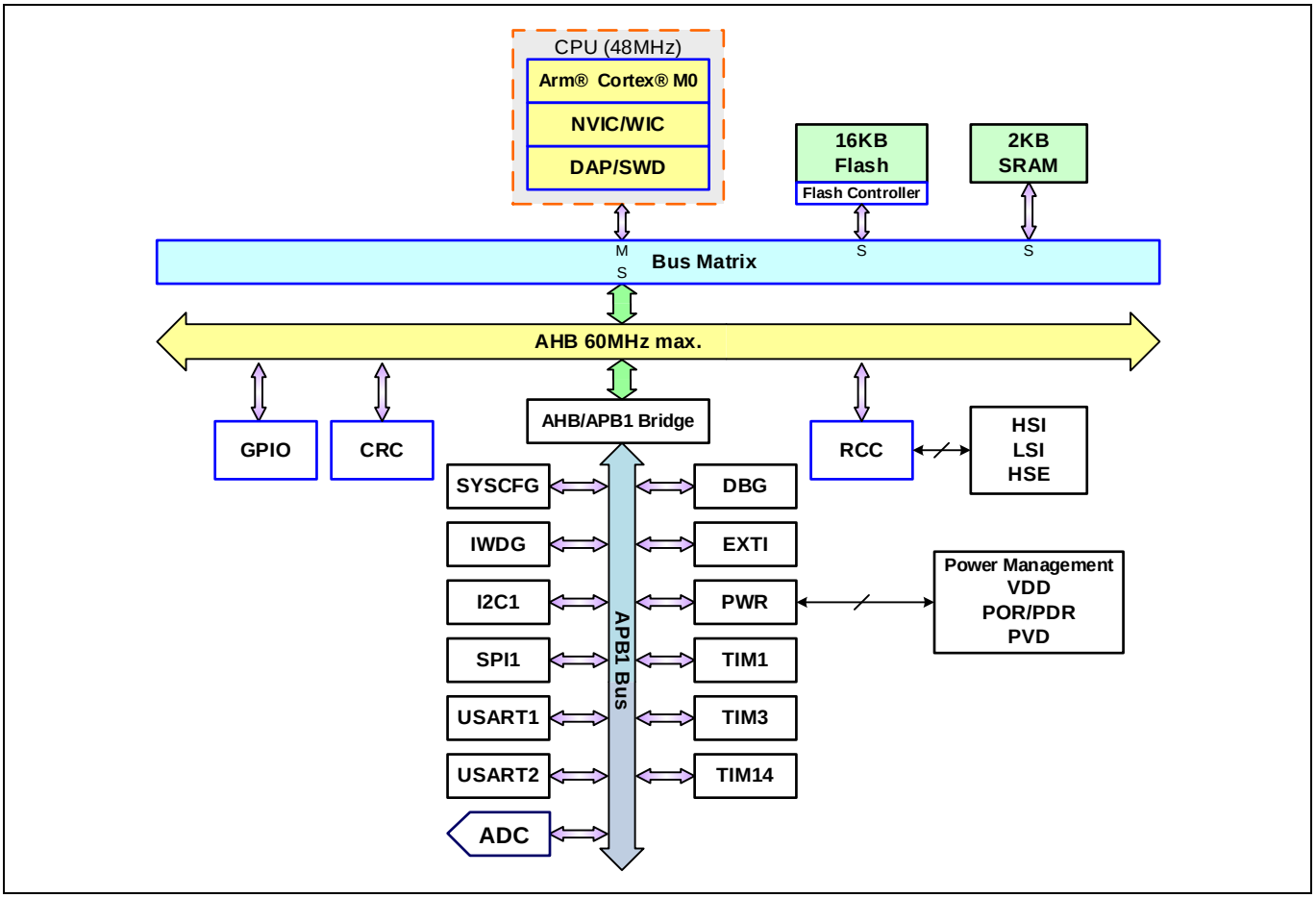
Tables

1 System and Memory Architecture

1.1 System architecture overview

MG32F04P032 product is a 32-bit microcontroller product based on Arm®Cortex®-M0 processor. It features high performance and low power consumption. It adopts the matrix bus structure, which includes 2 AHB masters: CPU and DMA as well as the following slaves: SRAM, Flash memory, AHB bus (including bus bridge from AHB to APB), and a variety of devices connected to APB bus.

Figure 1-1 System architecture block diagram



1.1.1 System bus

The system bus works to achieve data transfer by connecting the CPU core and bus matrix. CPU and DMA work as the host drive bus. The bus matrix will coordinate access between CPU core and DMA.

1.1.2 DMA bus

DMA bus works to achieve data transfer by connecting the DMA and bus matrix. The bus matrix coordinates the access control from master DMA to slave SRAM, Flash memory and a variety of peripherals connected to the APB line.

1.1.3 BUS matrix

The bus matrix consists of an AHB interconnected matrix, an AHB bus and two bridged APB buses. When CPU bus and DMA bus are requested simultaneously, it owns arbitration function. Peripherals (RCC, GPIO and DIV) of the AHB bus are connected to the system bus via the AHB interconnected matrix. The connections between the APB and AHB buses exchange data via the AHB2APB bridge. When the APB registers are accessed by 8-bit, 16-bit, the APB is automatically widened to 32-bit, and similarly, the AHB2APB bridge has an automatic widening function.

1.2 Memory introduction

Program memory, data memory, registers, and I/O interfaces are located in memory address space (linear 4GB address space) with different address ranges. 4GB address space is divided into 8 blocks, with 512MB per block.

1.2.1 Memory map and register addressing

For the memory map, please refer to the memory map diagram in the corresponding chapter of each peripheral.

Table 1-1 Memory mapping

Bus	Address range	Size	Peripheral
Flash	0x0000 0000 - 0x0000 7FFF	32 KB	Main Flash Memory, Data Memory, System Memory or SRAM depends on the configuration of BOOT
	0x0000 8000 - 0x07FF FFFF	~127 MB	Reserved
	0x0800 0000 - 0x0800 7FFF	32 KB	Main Flash Memory
	0x0800 8000 - 0x1FFD FFFF	~383 MB	Reserved
	0x1FE0 0000 - 0x1FE0 0BFF	3KB	Data Memory
	0x1FE0 0C00 - 0x1FFF F3FF	~2MB	Reserved
	0x1FFF F400 - 0x1FFF F7FF	1 KB	System memory
	0x1FFF F800 - 0x1FFF F9FF	0.5KB	Option Bytes
	0x1FFF FA00 - 0x1FFF FFFF	1.5KB	Reserved
SRAM	0x2000 0000 - 0x2000 0FFF	4 KB	SRAM
	0x2000 1000 - 0x2FFF FFFF	~255 MB	Reserved
APB1	0x4000 0000 – 0x4000 03FF	1KB	TIM2
	0x4000 0400 – 0x4000 0FFF	3KB	Reserved
	0x4000 1000 – 0x4000 13FF	1KB	TIM6
	0x4000 1400 – 0x4000 17FF	1KB	Reserved
	0x4000 1800 – 0x4000 1BFF	1KB	TIM13
	0x4000 1C00 – 0x4000 2FFF	5KB	Reserved
	0x4000 3000 – 0x4000 33FF	1KB	IWDG
	0x4000 3400 – 0x4000 6FFF	15KB	Reserved
	0x4000 7000 – 0x4000 73FF	1KB	PWR
	0x4000 7400 – 0x4000 8FFF	7KB	Reserved
	0x4000 9000 – 0x4000 93FF	1KB	OPA1
	0x4000 9400 – 0x4000 97FF	1KB	OPA2
	0x4000 9800 – 0x4000 FFFF	26KB	Reserved
	0x4001 0000 – 0x4001 03FF	1KB	SYSCFG
	0x4001 0400 – 0x4001 07FF	1KB	EXTI
	0x4001 0800 – 0x4001 0BFF	1KB	Reserved

	0x4001 0C00 – 0x4001 0FFF	1KB	USART1
	0x4001 1000 – 0x4001 23FF	5KB	Reserved
	0x4001 2400 – 0x4001 27FF	1KB	ADC1
	0x4001 2800 – 0x4001 2BFF	1KB	Reserved
	0x4001 2C00 – 0x4001 2FFF	1KB	TIM1
	0x4001 3000 – 0x4001 33FF	1KB	Reserved
	0x4001 3400 – 0x4001 37FF	1KB	DBG
	0x4001 3800 – 0x4001 3BFF	1KB	Reserved
	0x4001 3C00 – 0x4001 3FFF	1KB	COMP
	0x4001 4000 – 0x4001 43FF	1KB	TIM14
	0x4001 4400 – 0x4001 FFFF	47KB	Reserved
AHB	0x4002 0000 – 0x4002 03FF	1KB	DMA
	0x4002 0400 – 0x4002 0FFF	3KB	Reserved
	0x4002 1000 – 0x4002 13FF	1KB	RCC
	0x4002 1400 – 0x4002 1FFF	3KB	Reserved
	0x4002 2000 – 0x4002 23FF	1KB	Flash Interface
	0x4002 2400 – 0x4002 FFFF	55KB	Reserved
	0x4003 0000 – 0x4003 03FF	1KB	DIV
	0x4003 0400 – 0x47FF FFFF	~128MB	Reserved
	0x4800 0000 – 0x4800 03FF	1KB	PORT A
	0x4800 0400 – 0x4800 07FF	1KB	PORT B

1.2.2 Embedded SRAM

It features up to 4K bytes of a built-in static SRAM. It can be accessed in byte (8 bits), half-word (16 bits) or word (32 bits). The starting address of SRAM is 0x2000 0000.

SRAM can be accessed by CPU with the fastest system clock and without any waiting.

1.2.3 FLASH memory overview

The Flash memory is composed of two storage areas:

- The main Flash memory block includes the application data and user data area.
- The information block includes the data memory, option bytes and system memory:
  - Data memory - Used to store non-volatile data, or additional user programs
  - Option bytes - Containing hardware and user storage protection configuration options.
  - System memory - Containing boot loader code.

The Flash memory interface based on the AHB protocol, executes commands and data access. The prefetch buffer function of the Flash interface can speed up the code execution of CPU.

1.2.4 Boot configuration

In the chip, four different boot modes can be selected by the level state of BOOT0 pin level state and the configuration of BOOT0SEL, nBOOT0n and nBOOT1 bit, as shown in the following table:

Table 1-2 Boot Mode

BOOT Mode	Selection of BOOT Mode			
	nBOOT1	BOOT0 pin	nBOOT0	BOOT0SEL
Main Flash memory	x	0	x	1
System memory	1	1	x	1
Built-in SRAM	0	1	x	1
Main Flash memory	x	x	1	0
Data memory	x	x	0	0

The user selects BOOT modes by setting the pin value of BOOT0, BOOT0SEL, nBOOT0 and nBOOT1 bit. After the device is reset, the chip determines the BOOT mode based on different boot mode configuration.

After the device is reset, the CPU will begin to obtain the stack top value from the address 0x0000 0000, then obtain the base address of the boot code from address 0x0000 0004, and execute the program from the base address.

There are four main BOOT modes: main Flash memory, system memory, data memory and built-in SRAM.

BOOT from the main Flash memory (BOOT0=0): The starting address of the main Flash memory is 0x0800 0000. When it is chosen as the BOOT mode, it is mapped to the BOOT storage space (0x0000 0000) , but the Flash memory content can still be accessed from the starting address (0x0800 0000). When the main Flash memory is chosen as the BOOT mode, both the BOOT address and starting address can access the Flash memory.

BOOT from the system memory: The starting address of the system memory is 0x1FFF F400. When it is chosen as the BOOT mode, it is mapped to the BOOT storage space (0x0000 0000), but the system memory content can still be accessed from the starting address (0x1FFF F400). When the system memory is chosen as the BOOT mode, both the BOOT address and starting address can access the system memory.

BOOT from the data memory: The starting address of the data memory is 0x1FE0 0000. When it is chosen as the BOOT mode, it is mapped to the BOOT storage space (0x0000 0000), but the system memory content can still be accessed from the starting address (0x1FE0 0000). When the system memory is chosen as the BOOT mode, both the BOOT address and starting address can access the system memory.

BOOT from the built-in SRAM: The starting address of the built-in SRAM memory is 0x2000 0000. When it is chosen as the BOOT mode, it is mapped to the BOOT storage space (0x0000 0000), but the built-in SRAM memory content can still be accessed from the starting address (0x2000 0000). When the built-in SRAM is chosen as the BOOT mode, both the BOOT address and starting address can access the built-in SRAM.

1.2.5 Boot loader

The boot loader after leaving the factory is stored in the system memory, and ISP programming can be performed by the serial port (USART1).

2 Embedded FLASH

2.1 Overview

The embedded Flash supports the on-chip Main Flash up to 32KBytes. It also provides an option byte block and a system BOOT block (support chip Boot) as well as reserved IAP/Data space. The Flash memory controller supports read operations, page earse and whole-chip erase. These can be written into the Flash memory by 16-bit (half word)/32-bit (whole word) programming. When reading data, the Flash memory controller supports the data interface with the prefetch buffer to support the MCU running at a higher main frequency.

2.2 Flash composition and description

2.2.1 Flash composition

- The Flash memory is organized as 32-bit wide memory cells that can be used for storing both code and data.
- The main Flash memory block is divided into 32 pages (1Kbyte per page) or 8 write protection blocks (4Kbytes per block).
- The main Flash memory block can be erased by page (per 1Kbyte) and set with write protection by page.
- The write protection is set by taking the unit of 4 pages (4Kbytes) as one write protection block.
- The entire on-chip Flash is composed of two parts: Main storage block and information storage block.
- The main storage block is used to store user code and data, the user code can erase, program and read the main memory. Each 1Kbyte is called one page in the main storage block, which can perform the smallest unit of erasing. In addition, write protection is allocated in the unit of 1 write protection area (4Kbytes, 4 pages=1 write protection block), as shown in Table 2-1.

Table 2-1 Flash storage space

Module	Block name	Page name	Address	Size (byte)
Main storage block	Write protection area 0	Page 0	0x0800 0000 - 0x0800 03FF	1K
		Page 1	0x0800 0400 - 0x0800 07FF	1K
		Page 2	0x0800 0800 - 0x0800 0BFF	1K
		Page 3	0x0800 0C00 - 0x0800 0FFF	1K
	…	…	…	…
	Write protection area 3	Page 12	0x0800 3000 - 0x0800 33FF	1K
		Page 13	0x0800 3400 - 0x0800 37FF	1K
		Page 14	0x0800 3800 - 0x0800 3BFF	1K
		Page 15	0x0800 3C00 - 0x0800 3FFF	1K
	…	…	…	…
	Write protection area 7	Page 28	0x0800 7000 - 0x0800 73FF	1K
		Page 29	0x0800 7400 - 0x0800 77FF	1K

		Page 30	0x0800 7800 - 0x0800 7BFF	1K
		Page 31	0x0800 7C00 - 0x0800 7FFF	1K

In the information storage block, except for the “System Memory ISP” area that is factory locked and cannot be written by the user, other areas can be read and written by the user under certain conditions. The information memory can be divided into IAP/Data space, system memory ISP and option byte. IAP/Data space is used to store non-volatile data (EEPROM), or as IAP Flash. The system memory ISP is used to store the default ISP Bootloader. The first 12 bytes of the option byte is the write and read protection information of the main memory and the remaining bytes can be used to store user-specific data. The user can erase, program and read the information storage block through a specified procedure. As ISP is used to cure ISP upgrade code, it doesn't support the user to erase and program.

Table 2-2 Information block

Module	Name	Address	Size (byte)
Information block	IAP/Data space	0x1FE0 0000- 0x1FE0 0BFF	3K
	System storage ISP	0x1FFF F400 - 0x1FFF F7FF	1K
	Option byte	0x1FFF F800 - 0x1FFF F9FF	0.5K

2.2.2 Option byte description

In the option byte page, the content mainly includes write protection enable, hardware watchdog enables, etc. The Flash controller can disable the writing function of the main memory by setting the value in the option byte to prevent illegal write and it can also enable the hardware watchdog. Relevant information is stored in the option byte. The content in the option bytes, after modification, will only be valid after reset or power-on again. It will be written in the half word high and low byte complement mode, such as nUser, nData, etc. Each time after the system is reset, the option byte will reload the data of optional byte information block and make corresponding judgments and status changes. These sates are stored in the option byte registers (FLASH_OBR and FLASH_WRPR). Each selection bit in the information block has the corresponding inverse code bit, which is used to verify whether the selected bit is correct when loading the selected bit. If difference is detected during the loading process, an option byte error flag (OPTERR) will be produced. The interruption, if turned on, will be triggered.

The organizational structure of the option byte in the option byte block is given in the table below (the value in bit 15~8 is the inverse code of option byte in bit 7~0) ：

Table 2-3 Organizational structure of option byte

Address	[15：8]	[7：0]	Default value
0x1FFF F800	nRDP	RDP	0x5AA5
0x1FFF F802	nUSER	USER	0x00FF
0x1FFF F804	nData0	Data0	0x00FF
0x1FFF F806	nData1	Data1	0x00FF
0x1FFF F808	nWRP0	WRP0	0x00FF

Table 2-4 USER bit meaning

	Bit	Field	Type	Default	Description	FLASH_OBR
RDP	7：0	RDP	rw	0xA5	0xA5	FLASH_OBR. Bit1
nRDP	15：8	nRDP	rw	0x5A	0x5A	
User Byte	0	WDG_SW	rw	0x01	0: Hardware watchdog 1: Software watchdog	FLASH_OBR. Bit2
	1	nRST_STOP	rw	0x01	0: In Stop mode, the reset is produced 1: In Stop mode, the reset is not produced	FLASH_OBR. Bit3

	3	Reserved	rw	0x01	Reserved as 0x01	Reserved
	4	nBOOT1	rw	0x01	0: nBOOT1=0 1: nBOOT1=1	FLASH_OBR. Bit6
	5	OBR_nRST	rw	0x01	NRST MUX PA10 1：NRST function 0：GPIO function	FLASH_OBR. Bit7
	6	nBOOT0	rw	0x01	BOOT option, refer tp chip configuration chapter	FLASH_OBR. Bit8
	7	BOOT0SEL	rw	0x01	BOOT option, refer tp chip configuration chapter	FLASH_OBR. Bit9
DATA0 Byte	0	DATA0.Bit0	rw	0x01	User defined	FLASH_OBR. Bit10
	1	DATA0.Bit1	rw	0x01	User defined	FLASH_OBR. Bit11
	2	DATA0.Bit2	rw	0x01	User defined	FLASH_OBR. Bit12
	3	DATA0.Bit3	rw	0x01	User defined	FLASH_OBR. Bit13
	4	DATA0.Bit4	rw	0x01	User defined	FLASH_OBR. Bit14
	5	DATA0.Bit5	rw	0x01	User defined	FLASH_OBR. Bit15
	6	DATA0.Bit6	rw	0x01	User defined	FLASH_OBR. Bit16
	7	DATA0.Bit7	rw	0x01	User defined	FLASH_OBR. Bit17
DATA1 Byte	0	DATA1.Bit0	rw	0x01	User defined	FLASH_OBR. Bit18
	1	DATA1.Bit1	rw	0x01	User defined	FLASH_OBR. Bit19
	2	DATA1.Bit2	rw	0x01	User defined	FLASH_OBR. Bit20
	3	DATA1.Bit3	rw	0x01	User defined	FLASH_OBR. Bit21
	4	DATA1.Bit4	rw	0x01	User defined	FLASH_OBR. Bit22
	5	DATA1.Bit5	rw	0x01	User defined	FLASH_OBR. Bit23
	6	DATA1.Bit6	rw	0x01	User defined	FLASH_OBR. Bit24
	7	DATA1.Bit7	rw	0x01	User defined	FLASH_OBR. Bit25

Note: In the write protection values, one bit corresponds to four pages, namely 4096 Bytes.

2.2.3 IAP/Data space description

The IAP/Data space can be used to store non-volatile data (EEPROM) or as IAP Flash. This requires users to develop and program the IAP code by themselves. The erasing and programming operation flow of the IAP/Data space is the same as that of the option byte area. IAP/Data space has the same read/fetch behavior as the main flash block except that the address is not contiguous with the main flash block. Boot option can be configured to Boot from this area.

2.3 Flash option and procedure

2.3.1 Flash read operation

User code and data are stored in the main memory block and the Flash memory controller can read data or instructions in the 8bit/16bit/32bit. The main Flash module is uniformly addressed like ordinary peripherals. Based on the read and write protection requirements, any read or write operations on the content of the main memory block shall undergo a specific judgment process to prevent illegal read and write.

The Flash memory executes instruction fetching and date fetching via the AHB bus according to the method set in the Flash access control register (FLASH_ACR). Based on AHB clock, set the corresponding access latency and enable the prefetch value buffer. By doing so, the instruction fetch speed of CPU can be increased, thereby raising the running speed of CPU. The access latency if less than or equal to 30MHz in SYSCLK, can be set as 0. For an increment of 3024MHz, an additional latency is required.

After power-on reset, the Flash memory controller sets the prefetch buffer to be closed by default. If the prefetch instruction buffer function needs to be closed or reopened, SYSCLK must be set below 30MHz, and AHB clock can close or reopen the prefetch instruction buffer function provided that no

frequency division has been exerted (SYSCLK must be equal to HCLK).

In order to protect the correct reading of Flash, the speed ratio of the prefetch instruction controller must be specified in the LATENCY [2:0] in the Flash access control register. The value is equal to the number of waiting cycles inserted from each Flash access to the next access. After reset, this value is zero by default, that is, no waiting cycle is inserted, and the corresponding system clock is also reset as the built-in clock HSI=8MHz. If the system clock is to be modified after reset, the safe value of LATENCY [2:0] shall be configured first. When the AHB clock prescaler is greater than 1, the corresponding access latency of the prefetch instruction buffer should also be set.

Table 2-5 Latency setting relation

SYSCLK	AHB DIV	Latency
0MHz < SYSCLK <=30MHz	1	0
30MHz < SYSCLK <= 60MHz	1	1

2.3.2 Flash programming mode and operation procedure

The embedded Flash supports three programming modes below.

Table 2-6 Programming mode

Programming mode	Programming description
In circuit programming (ICP)	ICP means that the user code is burned into MCU by a specific programmer using SWD interface and changing Flash content.
In system programming (ISP)	ISP means that the user code is burned into MCU by ISP Firmware based on a specific UART interface and changing Flash content.
In application programming (IAP)	Different from ICP and ISP mode, IAP (in application programming) can use any MCU supported communication interface (UART, I2C, SPI, CAN, USB, etc.) to download program or data. IAP allows the user to rewrite the application during program operation, under the precondition that some programs must be pre-burned by ICP or ISP mode.

The burning and erasing can be completed within the working voltage range of the whole product. When writing or erasing the Flash space, the internal oscillator (HSI) must be turned on and the AHB clock must be greater than or equal to 8MHz.

As long as CPU does not access Flash space, the ongoing Flash write operation will not hamper the running of CPU (running from RAM or ISP). When writing or erasing the Flash, the read access to the Flash will always encounter a bus pause, and it will not continue until the writing or erasing operation is completed. Therefore, it is not allowed to fetch instruction and access data when the Flash is written or erased.

The programming of Flash is composed of a series of actions, mainly including:

- Unlock and protect Flash operation
- Erase Flash (page erasing and whole-chip erase)
- Program Flash (half-word or whole-word programming)
- Unlock and protect the operations of each sector in the information block (such as option byte)
- Erase each sector in the information block (such as option byte)
- Program (half-word or whole-word programming) each sector in the information block (such as option byte)

ISP, IAP programming procedure

Figure 2-1 ISP programming procedure

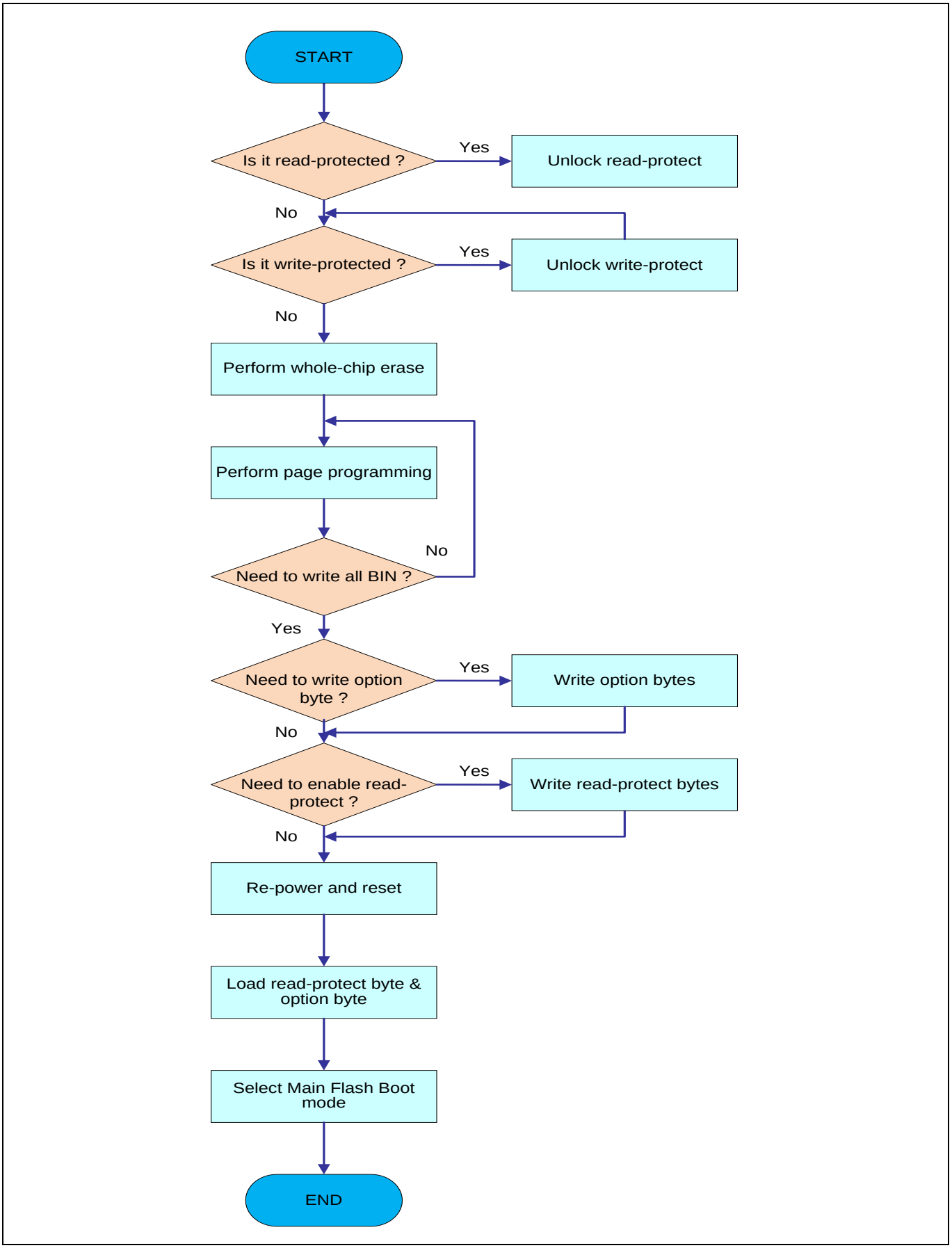
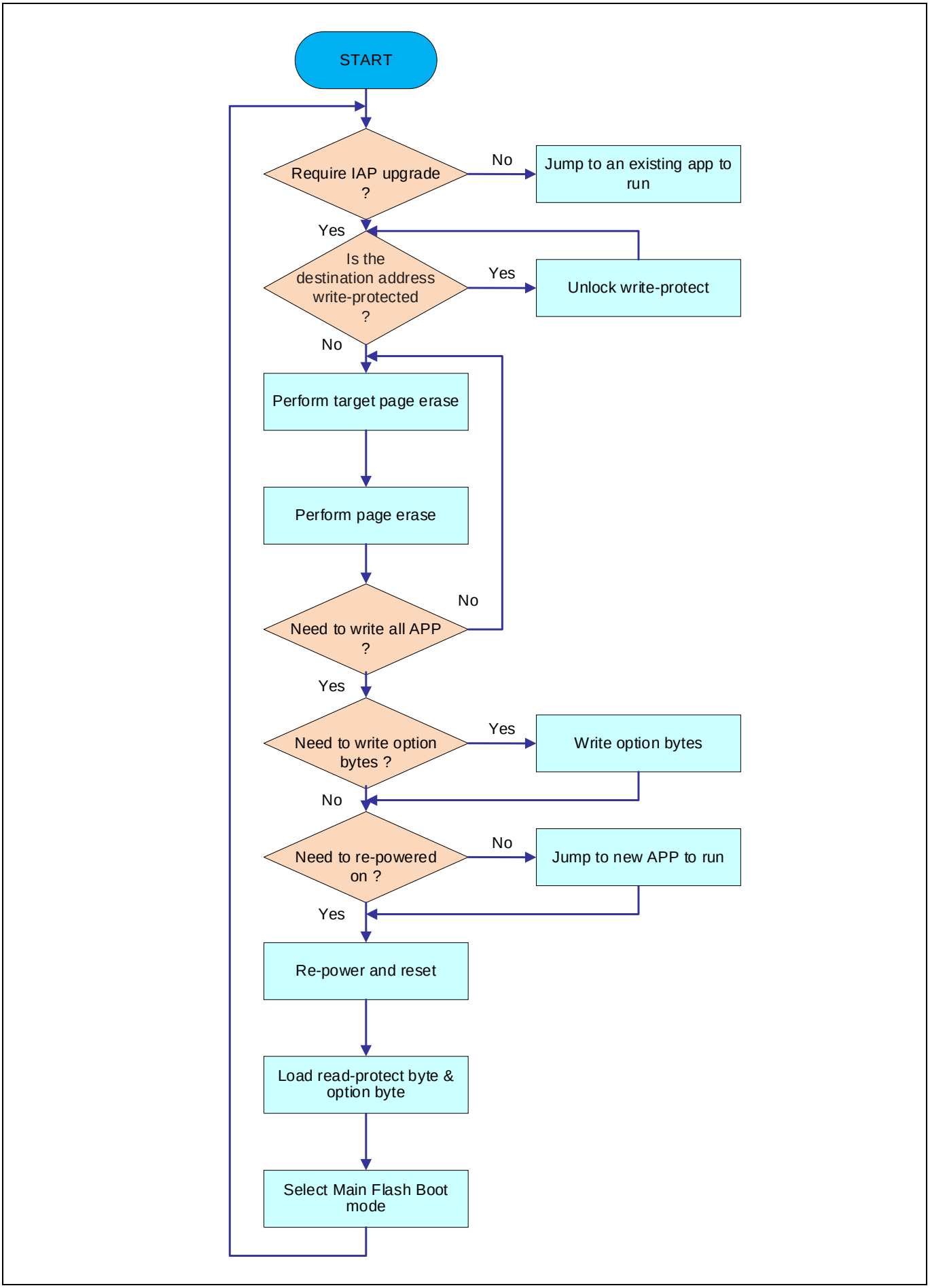


Figure 2-2 IAP programming procedure



2.3.3 Disable and enable the operation restriction on Flash block

The embedded Flash memory is in a protected status after reset, which can avoid accidental operations that damage the Flash storage space, such as page erase, whole chip erase and value writing. FLASH_CR register is locked after reset, and the LCOK bit of FLASH_CR is set as 1 by the controller module. Only by writing 0x45670123 and 0xCDEF89AB respectively to the FLASH_KEYR register for unlocking and setting the LCOK bit of FLASH_CR as 0, can FLASH_CR be accessed. Otherwise, the FLASH_CR register is not allowed to be rewritten.

Set the LOCK bit of FLASH_CR LOCK as 1 by software to lock again, making the Flash memory in the protected status.

Disable protection operation code:

```
#define FLASH_KEY1              ((unsigned int)0x45670123)
#define FLASH_KEY2              ((unsigned int)0xCDEF89AB)
void FLASH_Unlock()
{
  FLASH->KEYR = ((unsigned int)0x45670123);
  FLASH->KEYR = ((unsigned int)0xCDEF89AB);
}
```

Operations that do not conform to the above sequence and writing the wrong value will lock FLASH_CR and trigger a bus error till the next reset.

Enable protection operation code:

```
#define FLASH_CR_LOCK_Pos       (7)
#define FLASH_CR_LOCK           (0x01U << FLASH_CR_LOCK_Pos)
void FLASH_Lock(void)
{
  FLASH->CR |= FLASH_CR_LOCK;
}
```

2.3.4 Disable and enable the operation restriction on option byte block

After the Flash memory controller is reset, its option byte block is write-protected by default, and readable at any time. In order to avoid destructive operations such as block erasing and writing value on the option byte area and read protection setting area, after reset, FLASH_CR register enters the locked status and the LOCK bit of FLASH_CR is set as 1 by the controller module, while OPTWRE bit is cleared as 0 by the controller module. Therefore, it is necessary to write 0x45670123 and 0xCDEF89AB to the FLASH_KEYR register to unlock the FLASH. After the LOCK bit of FLASH_CR is 0, the option byte block are unlocked. Writing 0x45670123 and 0xCDEF89AB to FLASH_OPT_KEYR register, the hardware will set the OPTWRE bit of FLASH_CR register as 1. By doing so, block erasing and half-word or whole-word programming can be performed on the option byte area. The OPTWRE bit of the FLASH_CR register can be set as 0, thereby disabling block erasing and half-word or whole-word programming on the option byte area.

Table 2-7 Changes of status in the protection setting

Set and status	Main Flash block	Information block	Description
Power on and reset **The status of Flash controller is** **FLASH_CR.LOCK=1** **FLASH_CR.OPTWRE=0**	Protection	Protection	Enable operation protection of main Flash block Enable operation protection of option byte block
Set **FLASH_KEYR=0x45670123** **FLASH_KEYR=0xCDEF89AB** **The status of Flash controller is** **FLASH_CR.LOCK=0** **FLASH_CR.OPTWRE=0**	Disable protection	Protection	Disable operation protection of main Flash block, execute whole chip erasing, and page erasing, half-word or whole-word programming of main Flash block. Keep enabling operation protection of option byte block. It disables block erasing and half-word or whole-word programming of option byte area.
FLASH_KEYR=0x45670123 **FLASH_KEYR=0xCDEF89AB** **FLASH_OTPKEYR=0x45670123** **FLASH_OTPKEYR=0xCDEF89AB**	Disable protection	Disable protection	Disable operation protection of main Flash block, execute whole chip erasing, and page erasing, half-word or whole-word programming of main Flash block. Disable

The status of Flash controller turns FLASH_CR.LOCK=0 FLASH_CR.OPTWRE=1			operation protection of option byte block. It enables block erasing and half-word or whole-word programming of option byte area.
Set FLASH_CR.OPTWRE=0 Keep FLASH_CR.LOCK=0	Disable protection	Enable protection	It disables operation protection of main Flash block, and enables operation protection on option byte block
Set FLASH_CR.OPTWRE=0 Set FLASH_CR.LOCK=1	Enable protection	Enable protection	Enable operation protection of main Flash block as well as option byte block

Disable protection operation code：

```
#define FLASH_KEY1              ((unsigned int)0x45670123)
#define FLASH_KEY2              ((unsigned int)0xCDEF89AB)
void FLASH_Unlock(void)
{
  FLASH->KEYR = ((unsigned int)0x45670123);
  FLASH->KEYR = ((unsigned int)0xCDEF89AB);
}
```

Enable protection operation code:

```
#define FLASH_CR_LOCK_Pos           (7)
#define FLASH_CR_LOCK               (0x01U << FLASH_CR_LOCK_Pos)
void FLASH_Lock(void)
{
  FLASH->CR |= FLASH_CR_LOCK;
}
```

Disable operation code of option byte as well as read protection setting area:

```
#define FLASH_KEY1              ((unsigned int)0x45670123)
#define FLASH_KEY2              ((unsigned int)0xCDEF89AB)
void FLASH_OPT_Unlock (void)
{
  FLASH->OPTKEYR = FLASH_KEY1;
  FLASH->OPTKEYR = FLASH_KEY2;
}
```

Enable operation code of option byte as well as read protection setting area:

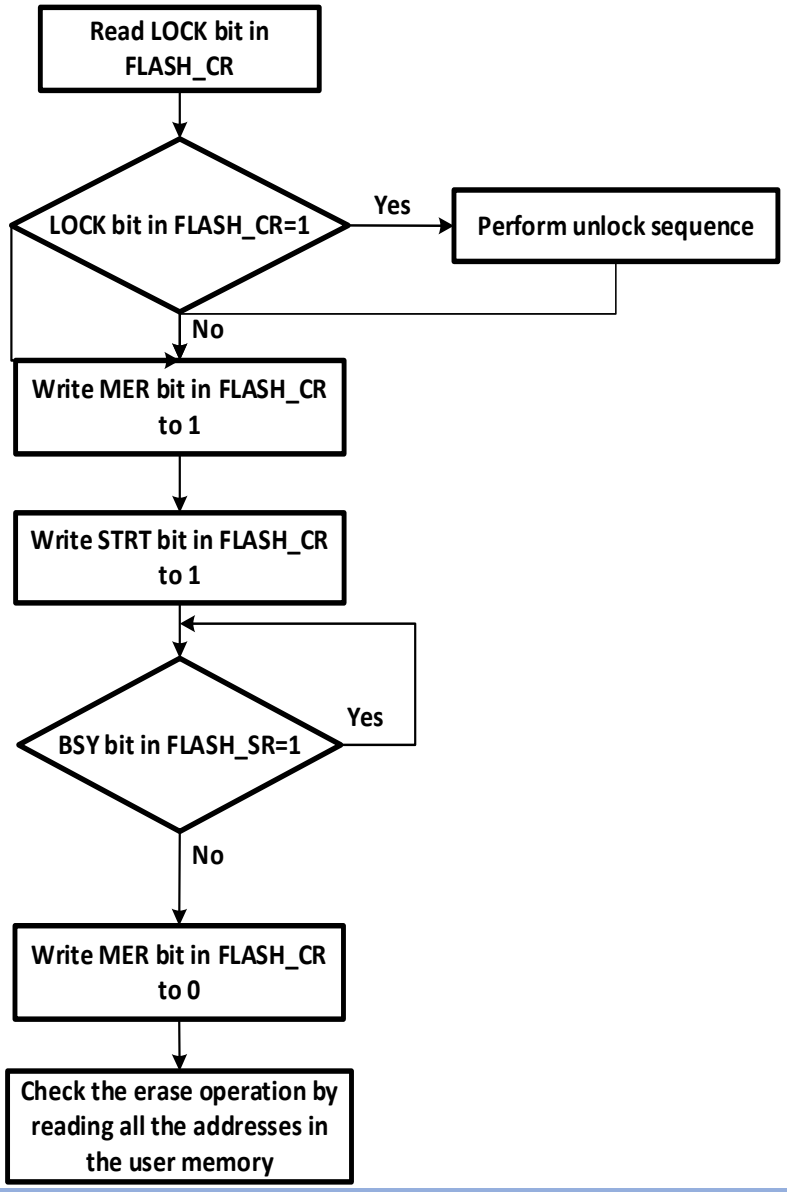
```
#define FLASH_CR_OPTWRE_Pos    (9)
#define FLASH_CR_OPTWRE        (0x01U << FLASH_CR_OPTWRE_Pos)
void FLASH_OPT_Lock(void)
{
  FLASH->CR &= ~FLASH_CR_OPTWRE;
}
```

2.3.5 Main Flash block erasing

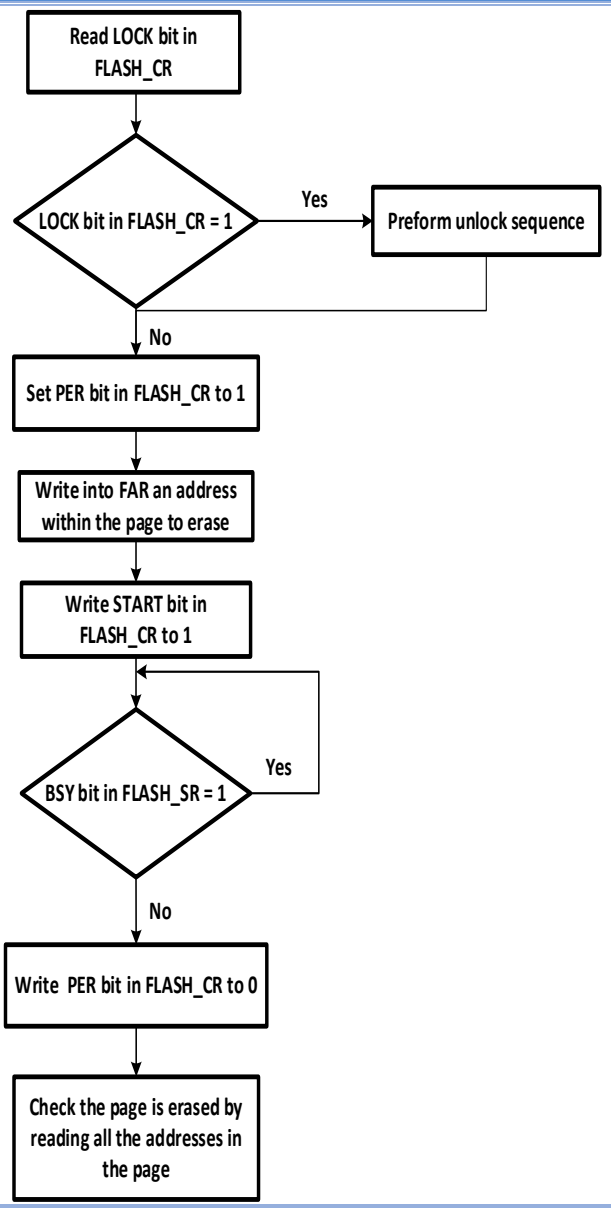
The Flash memory controller supports the whole chip erasing of main Flash block and the erasing of pages inside the main Flash memory in the unit of page.

The whole chip erasing function will initialize all the contents of the main Flash memory block and set all values as 0xFFFF. However, the information block will not be affected by this instruction.

Main Flash memory Erase procedure:

Steps	Simple flowchart
(1)Read the LOCK bit of FLASH_CR to ensure that Flash protection is disabled, (2) Otherwise execute Flash Unlock (3) Set MER bit in the FLASH_CR register as 1 and enable the whole chip erasing of main Flash (4) Set STRT bit in the FLASH_CR register as 1 (5) Start the whole chip erase of the main Flash (6) Check BUSY bit in FLASH_SR (7) Wait for completion of whole erase of main Flash (8) Clear the MER bit of FLASH_CR register, and make this bit as 0 (9) Optionally conduct Blank check to ensure erasing success	

Flash memory Page Erase procedure:

Steps	Simple flowchart
(1) Read the LOCK bit of FLASH_CR to ensure that Flash protection is disabled, otherwise, execute Flash Unlock (2) Set PER bit in FLASH_CR register as 1 and enable page erasing (3) Write the base address of the to-be-erased page to FLASH_AR register (4) Set STRT bit in FLASH_CR register as 1 and start page erasing (5) Check BUSY bit in FLASH_SR, (6) Wait for page erasing to finish (7) Clear the PER bit of FLASH_CR register, and make this bit as 0 (8) Select content of this page to check and ensure the success of page erasing (9) For continuous erasing of multiple pages, repeat 2~7 operations till the completion of erasing all pages	

2.3.6 Main Flash block programming

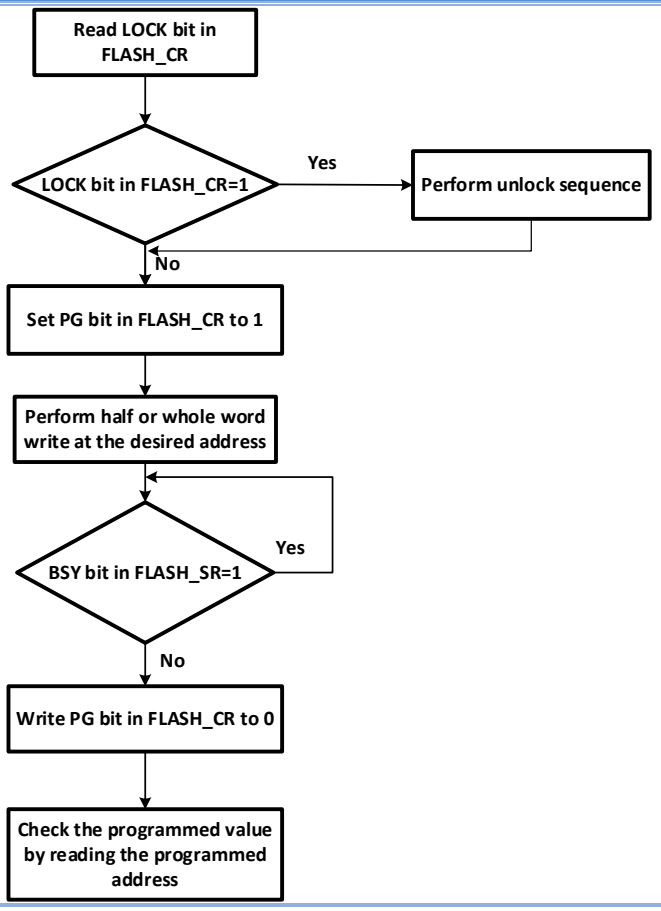
The main Flash memory only supports 16-bit half-word or 32-bit word programming to change the content of main storage Flash memory block. If it is programmed with a length of 8-bit byte, a hardware error interrupt is caused. When the PG bit of FLASH_CR is 1, a half-word (16 bits) or whole-word (32 bits) can be directly written to the corresponding address, namely one programming operation.

The main Flash memory controller will pre-read whether the half word or whole word to be programmed is whole 1 (whether it is 0xFFFF or 0xFFFFFFFF); if not, the programming will be automatically canceled and a programming error warning will be displayed on the PGERR bit of FLASH_SR register.

If the write protection bit in FLASH_WRRP of the corresponding write protection block of the address to be programmed is valid, there will be no programming action; but a programming error warning is generated. It will also produce programming error alarm. After programming, EOP bit in FLASH_SR register will produce the prompt.

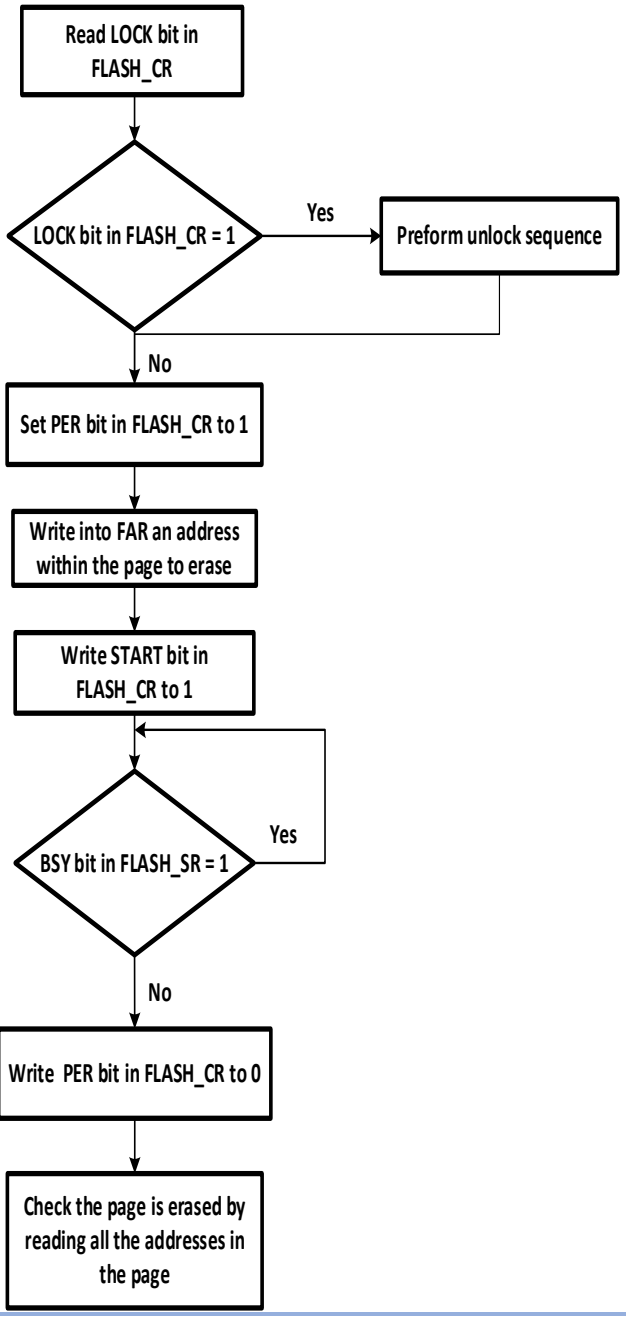
Note: When the CPU enters the power saving mode, an error will be produced to the Flash memory operation via SWD interface. It is necessary to avoid running interrupt programs while perform erasing or programming operations.

Specific steps for the register setting of main Flash block programming:

Steps	Simple flowchart
(1)Read the LOCK bit of FLASH_CR to ensure that Flash protection is disabled, otherwise, execute Flash Unlock (2) Set PG bit in FLASH_CR register as 1 and enable half-word or whole-word programming (3) Write data to the target address in a unit of half word or whole-word, and align the target address with half word or whole-word (4) Check BUSY bit in FLASH_SR (5) Wait for half-word or whole-word programming to finish (6) Clear the PG bit of FLASH_CR register, and make this bit as 0 (7) Select to read target address data to ensure the success of half-word or whole-word programming (8) For more continuous half-word or whole-word programming, repeat 2~6 operations, till the completion of half-word or whole-word programming of all target addresses	

2.3.7 Option byte block erasing

Specific steps for the register setting of option byte block erasing:

Steps	Simple flowchart
(1) Read the LOCK bit of FLASH_CR to ensure that Flash protection is disabled, otherwise, execute Flash Unlock (2) Read the OPTWRE bit of FLASH_CR to ensure that the option block protection is disabled. Otherwise, execute option block Unlock (3) Write block base address to be erased to FLASH_AR register (4) Set OPTER bit in FLASH_CR register as 1 and enable option byte block erasing (5) Set STRT bit in FLASH_CR register as 1 (6) Check BUSY bit in FLASH_SR (7) Wait for block erasing to finish (8) Clear the OPTER bit of FLASH_CR register, and set the value as 0 (9) Select to check the block content to ensure the success of the block erasing (10) For continuous erasing of multiple blocks, repeat 3~8 operations, till the completion of erasing all option blocks	

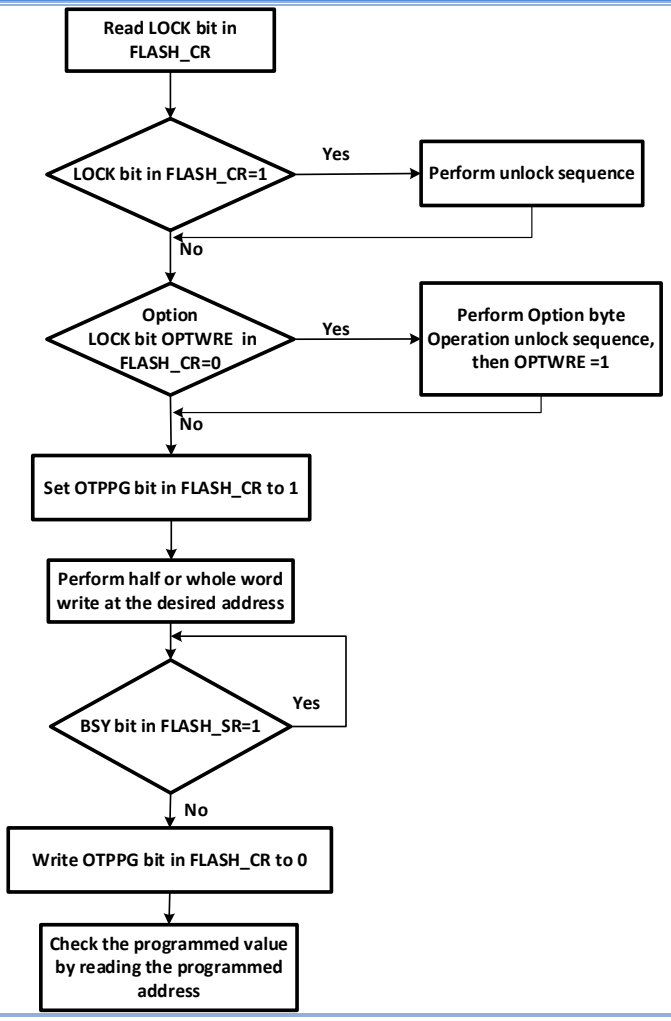
2.3.8 Option byte block programming

The programming of the option byte block is different from that of the main Flash memory block address, because the written value after reset is loaded to the configuration option, which requires more stringent protection. After the access restriction on the Flash memory controller is removed, the access restriction on the option byte block and read protection area should also be disabled. After this operation is completed, the OPTWRE bit in the FLASH_CR register will be set as 1 before the follow-up programming is allowed.

The valid data of option byte is the low 8 bits, and the high 8 bits are the inverse code of low 8 bits, thus forming the 16-bit data. During programming, the software sets the high 8 bits as the inverse code of low 8 bits to ensure that the written value of the option byte is always correct, and then write 16-bit data in sequence.

The change of optional byte is only valid after the system is powered-on and reset.

Specific steps for the register setting of option byte block half-word or whole-word programming:

Steps	Simple flowchart
(1) Read the LOCK bit of FLASH_CR to ensure that the Flash protection is disabled. Otherwise, execute Flash Unlock (2) Unlock Read the OPTWRE bit of FLASH_CR to ensure that the option block protection is disabled. Otherwise, execute option block Unlock (3) Set OPTPG bit of FLASH_CR register as 1 and enable option byte block half-word or whole-word programming (4) Write data (half word or whole-word) to target address and align target address by half word or whole-word (5) Check BUSY bit of FLASH_SR (6) Wait for the completion of half word or whole-word programming (7) Clear the OPTPG bit of FLASH_CR register, and set the value as 0 (8) Select to read target address data and ensure the success of half-word or whole-word programming (10) For continuous programming of multiple half words, repeat 3~7 operations, till the completion of half word or whole-word programming of all target addresses	

2.3.9 Flash memory protection

The main Flash block enables read and write protection can prevent the codes of the main Flash block from being read by unreliable codes and the accidental erasing and programming of main Flash memory block during program runaway. The read protection enables operation works for the whole main memory block while the minimum unit of enabling write protection is a write protection block (namely 4 pages).

2.3.9.1 Main Flash block read protection

The main Flash block enables or disables read protection is to set RDP half word by built-in SRAM or ICP, ISP, and works when loading new RDPs after the system is re-powered on and reset. It works by executing one power-on and reset rather than system reset, if read protection is set.

Enable read protection

According to the operation mode of half word or whole word programming in the option byte area, write in sequence the half word of RDP to the corresponding address

- Set the address value of FLASHAR as 0x1FFFF800, and execute erasing of the option block.
- According to the operation mode of half word or whole word programming in the option byte area, write in sequence half words of 0x807F to the corresponding address.
- Conduct a power-on reset to reload the option byte and the read protection is then disabled.

When RDP word includes the following values, the main Flash block is placed in the protected status after being re-powered on and reset.

Table 2-8 Flash read protection status

Enable read protection	Read protection status
Erase 0x1FFFF800 option block Write 0x807F half word to the corresponding address 0x1FFFF800 Disable read protection after being re-powered on and reset	Protection

When read protection half word is written within the corresponding value:

1. Only read operations to the main Flash memory from user code (start from main Flash memory in non-debug mode) are allowed.
2. After read protection, operations to the Flash are disabled in the debug mode (SRAM boot and debug mode).
3. MCU can be programmed by the code executed in the main Flash memory (enable functions such as IAP or data storage), but write or erasing operations in debug mode or booting from the internal SRAM (except for whole chip erasing) are not allowed.
4. All functions of loading and executing code to the built-in SRAM through SWD remain valid or it can also boot from the built-in SRAM. This function can be used to disable read protection.
5. Access to the main Flash memory by executing code from the built-in SRAM or access to the Flash via DMA, SWV (Serial Wire Viewer) and SWD (Serial Wire Debug) will all be disabled.

Disable read protection

Disable read protection from built-in SRAM or ICP mode:

1. Set FLASH_AR address value as 0x1FFFF800 and execute the operation block erase.
2. According to the operation mode of half word or whole word programming in the option byte area, write in sequence half words of 0x5AA5 to the corresponding address.
3. Set FLASHAR address value as 0x08000000 and execute main FLASH whole chip erasing.
4. Conduct a power-on reset to reload the option byte, and disable read protection.

Table 2-9 Flash disable read protection status

Disable read protection status	Read protection status
Erase 0x1FFFF800 option block Write 0x5AA5 half word to the corresponding address 0x1FFFF800 Erase 0x08000000 main Flash whole chip Power on and reset, disable read protection	Disable read protection

Note: 1. If the corresponding address value of the option byte block is not 0xFFFF, execute the erasing of option byte block first, which will not lead to an automatic erasing of the whole chip or change the read protection status. 2. Whole chip erasing must be performed on the main Flash of 0x08000000.

2.3.9.2 Main Flash block write protection

Enable write protection

The write protection is enabled by setting WRP bit of WRP0~WRP15 in the option byte block as 0. After system reset, load the new option byte to enable write protection. An attempt to write or erase a write protection area page will assert WRPRTERR flag bit in the FLASH_SR.

Table 2-10 Write protection area

Address	[15：8]	[7：0]	Default	Note
0x1FFF F808	nWRP0	WRP0	0xFFFF	

Disable write protection

There are two scenarios for disabling the write protection:

1. Scenario 1: Disable write and read protection:
   a. Use the OPTER bit in the Flash control register (FLASH_CR) to erase the whole option byte block; write a half word of 0x5AA5 to the corresponding address 0x1FFFF800;
   b. Perform a whole-chip erasing on the main Flash block of 0x08000000;
   c. Reset the system and reload the option byte (including the new WRP byte); write protection is disabled. Use of this method will disable the writing protection of the whole chip main Flash module and erase the whole chip of main Flash block.
2. Scenario 2: Disable write protection and keep the read protection valid. This scenario is commonly seen in the user's startup program:
   a. Use OPTER bit of Flash control register (FLASH_CR) to only erase the whole option byte block.
   b. Reset the system and reload option byte (including the new WRP byte); write protection is disabled.

Use of this method will disable the write protection of the whole main Flash module and keep the read protection valid.

2.4 Register

2.4.1 Register overview

Table 2-11 FLASH register overview

Offset	Acronym	Register Name	Reset
0x00	FLASH_ACR	Flash access control register	0x00000038
0x04	FLASH_KEYR	FPEC key register	0x00000000
0x08	FLASH_OPTKEYR	Flash OPTKEY register	0x00000000
0x0C	FLASH_SR	Flash status register	0x00000000
0x10	FLASH_CR	Flash control register	0x00000080
0x14	FLASH_AR	Flash address register	0x00000000
0x1C	FLASH_OBR	Option byte register	0x03FFFC1C
0x20	FLASH_WRPR	Write protection register	0xFFFFFFFF

Note: Flash register only supports 32-bit access

2.4.2 FLASH_ACR Flash Access Control Register

Offset address:0x00
Reset value:0x0000 0008

31	30	29	28	27	26	25	24	23	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										PRFTBS	PRFTBE	Res.	LATENCY		
										r	rw		rw		

Bit	Field	Description
31：6	Reserved	Reserved, must retain the reset value
5	PRFTBS	Prefetch buffer status 0: Prefetch buffer close 1: Prefetch buffer open
4	PRFTBE	Prefetch buffer enable 0: Close prefetch buffer. 1: Open prefetch buffer.
3	Reserved	Reserved, must retain the reset value

Bit	Field	Description
2：0	LATENCY	Latency These bits indicate the ratio of SYSCLK (system clock) period to Flash memory access time. 000: zero wait state, 0 < SYSCLK ≤ 30MHz 001: one wait state, 30MHz < SYSCLK ≤ 60MHz

2.4.3 FLASH_KEYR FPEC Key Register

Offset address:0x04
Reset value:0x0000 0000

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

Bit	Field	Description
31：0	FKEYR	FPEC Flash key These bits are used to input FPEC unlock key.

Note: All these bits are write-only and return to 0 when read.

2.4.4 FLASH_OPTKEYR Flash OPTKEY Register

Offset address:0x08
Reset value:0x0000 0000

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

Bit	Field	Description
31：0	OPTKEYR	Option byte key These bits are used as the input option byte key to disable OPTWRE.

Note: All these bits are write-only and return to 0 when read.

2.4.5 FLASH_SR Flash Status Register

Offset address:0x0C
Reset value:0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved										EOP	WRPRT ERR	Res.	PGERR	Res.	BSY
										rw	rw		rw		r

Bit	Field	Description
31:6	Reserved	Reserved, must retain the reset value
5	EOP	End of operation When the Flash memory operation (programming I erasing) is completed, the hardware sets this bit as “1” and write “1” to clear the bit status.

Bit	Field	Description
4	WRPRTERR	Write protection error When an attempt is made to program the write-protected Flash memory address, the hardware sets this bit as “1” and write “1” to clear the bit status.
3	Reserved	Reserved, must retain the reset value
2	PGERR	Programming error When an attempt is made to program an address whose content is not “0x FFFF”, the hardware sets this bit as “1” and write “1” to clear the bit status
1	Reserved	Reserved, must retain the reset value
0	BSY	Busy This bit indicates that the operation of Flash memory is in progress. Before Flash operation starts, this bit is set as “1”. After the operation or in case of an error, this bit is cleared as “0”.

2.4.6 FLASH_CR Flash Control Register

Offset address:0x10
Reset value:0x0000 0080

31	30	29	28	27	26	25	24	23	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						OPTWRE	Res.	LOCK	STRT	OPTER	OPTPG	Res.	MER	PER	PG
						rw		rw	rw	rw	rw		rw	rw	rw

Bit	Field	Description
31:10	Reserved	Reserved, must retain the reset value
9	OPTWRE	Option byte write enable When this bit is “1”, the option byte is allowed to be programmed. When the correct key sequence is written in the FLASH_OPTKEYR register, the bit is set as “1”. This bit can be cleared by writing 0.
8	Reserved	Reserved, must retain the reset value
7	LOCK	Lock Only write “1”. When the bit is “1”, it indicates that FPEC and FLASH_CR are locked. When the correct unlocking sequence is detected, the hardware automatically clears this bit as “0”. After an unsuccessful unlocking operation, this bit cannot be changed before the next system reset.
6	STRT	Start An erasing is triggered when this bit is “1”. The bit can only be set to “1” by software and automatically be cleared to “0” when BSY turns into “1”.
5	OPTER	Option byte erase Erase option byte.
4	OPTPG	Option byte programming Program option byte.
3	Reserved	Reserved, must retain the reset value
2	MER	Mass erase Choose to erase all user pages.
1	PER	Page erase Choose to erase page.
0	PG	Programming Choose to conduct programming.

2.4.7 FLASH_AR Flash Address Register

Offset address:0x14
Reset value:0x0000 0000

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

Bit	Field	Description
31：0	FAR	Flash Address Select the page to be erased when performing page erasing. Note: When the BSY bit in FLASH_SR is “1”, the register cannot be written.

The address is modified by the hardware to the currently used one. During the operation of page erasing, the register is modified to specify the page to be erased.

2.4.8 FLASH_OBR Option Byte Register

Offset address:0x1C
Reset value:0x03FF FC1C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved						Data1								Data0	
						r								r	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Data0						BOOT0SEL	nBOOT0	OBR_nRST	nBOOT1	Res.		nRST_STOP	WDG_SW	RDPRT	OPTERR
r						r	r	r	r	r		r	r	r	r

Bit	Field	Description
31：26	Reserved	Reserved, must retain the reset value
25：18	Data1	Data1
17：10	Data0	Data0
9	BOOT0SEL	BOOT0SEL
8	nBOOT0	nBOOT0
7	OBR_nRST	NRST multiplexing PA10 1：NRST function 0：GPIO function
6	nBOOT1	nBOOT1
5	Reserved	Reserved, must retain the reset value
4	Reserved	Reserved, must retain the reset value
3	nRST_STOP	Reset event in stop mode 0: Reset in stop mode 1: No reset in stop mode
2	WDG_SW	Select watchdog event 0: Hardware watchdog 1: Software watchdog
1	RDPRT	Read protection level status When setting to “1”, it indicates that the Flash memory is read protected. Note: The bit is read-only.
0	OPTERR	Option byte error When this bit is “1”, it indicates that the option byte does not match its inverse code. Note: The bit is read-only.

2.4.9 FLASH_WRPR Write Protection Register

Offset address: 0x20
Reset value: 0xFFFF FFFF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.								WRP							
								r							

Bit	Field	Description
31:8	Reserved	Reserved, must retain the reset value
7:0	WRP	Write protection The register includes the write protection option byte loaded by OBL. 0:Write protection enabled 1:Write protection disabled

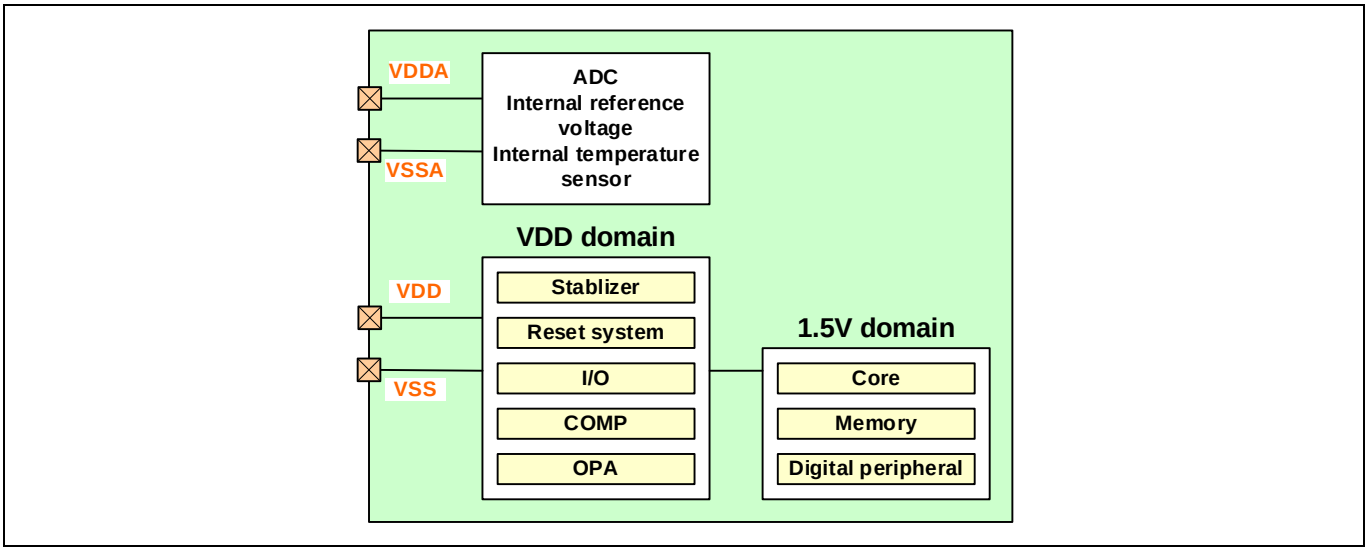
3 PWR Power Controller

Power controller (PWR) mainly involves features such as chip's power supply, power manager and low-power mode.

3.1 Power supply system

The chip is powered by four types of power supply below:

Figure 3-1 Power control block diagram



- The analog power supply provided by VDDA and VSSA provides voltage for the analog module of the chip and is used for ADC module, internal reference voltage and internal temperature sensor.
- The digital power supplied by VDD and VSS provides power for digital part and I/O pins.
- In the power supply system, it's required to connect the 10uF and 100nF capacitor in the corresponding power pin and try to get close to the chip power pin.

Note:

- Among packages with VDDA and VSSA, VDDA and VSSA cannot be pending, and the voltage difference of VDD and VDDA needs to be lower than 50mV.
- Among packages without VDDA and VSSA, VDD and VDDA, VSSA and VSS have been connected within the package.

3.1.1 Analog module power supply

The analog module mainly supplies power for the internal analog circuit of the MCU. It is mainly composed of ADC modules, reset system, etc. Therefore, the stability of the power supply affects the performance of the analog module.

ADC accuracy partially depends on the power stability of ADC module. For ADC application of high precision, in order to filter and mask ADC sampling interference by the burr on printed circuit board and improve ADC conversion accuracy, ADC uses an independent external power for stable external power supply.

- ADC power pin is VDDA
- Independent power ground of ADC is VSSA

3.1.2 Digital module power supply

VDD and VSS are digital module power ports of chip. They mainly provide power for IO, as well as core, built-in digital peripherals and memory via the voltage stabilizer.

3.1.3 VDD domain

VDD domain mainly provides power for the analog part of PMU (LDO and power supply detection) and ADC/COMP/OPA and keeps open after power-on.

3.1.4 1.5V domain

1.5V domain mainly provides power for the core, memory and peripherals of the chip, and keeps open after power-on by default. When entering the low-power mode, the chip hardware will choose to enter the low-power mode of the stabilizer, and it automatically switches to the normal power mode of the stabilizer upon wake-up. It has the following several working states:

Run Mode: 1.5V domain runs in a normal power consumption mode, and memory and peripherals normally work.

Sleep Mode: 1.5V power runs in a normal power consumption mode. CPU enters the sleep mode, while memory and peripherals work in normal power consumption mode.

Stop Mode: 1.5V domain works in a low power consumption mode and only keeps the content of register and RAM.

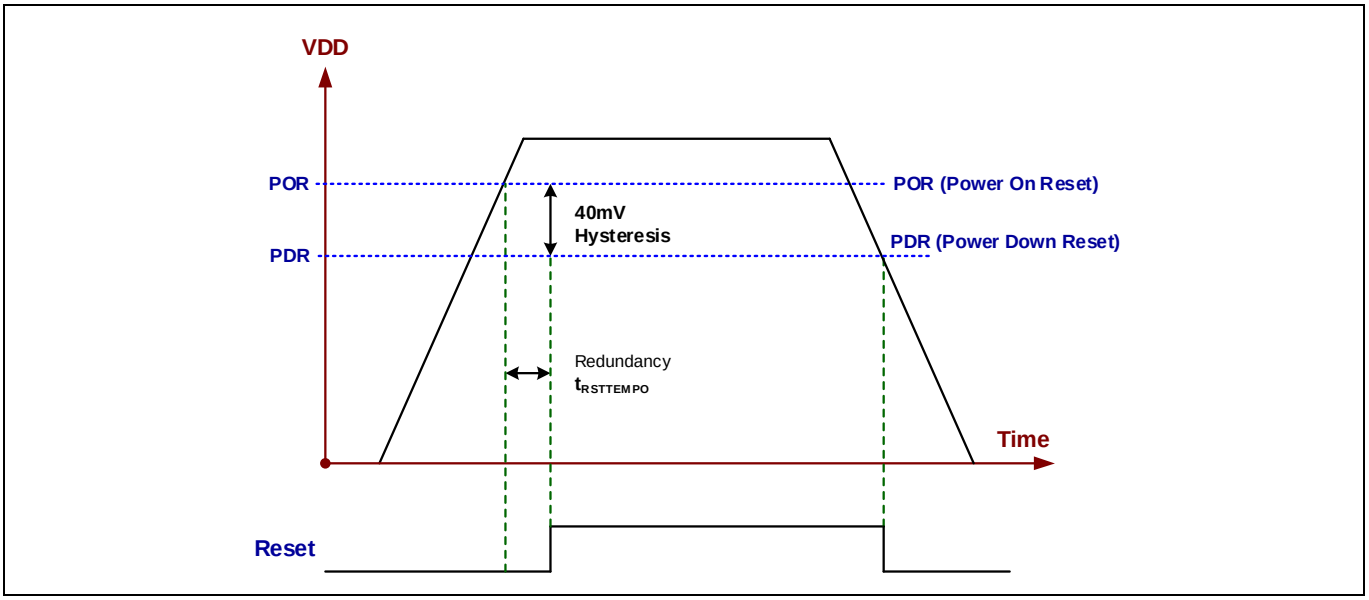
3.2 Power manager

3.2.1 Power on reset and power down reset

The chip has a complete power-on reset (POR) and power down reset (PDR) circuit. When the power voltage reaches the minimum working voltage, the system works normally. When the power voltage is lower than the minimum working voltage, the system doesn't work. When the chip is powered on or powered down, the power voltage reaches the minimum working voltage of the chip during the power-on operation and the chip will produce a power-on reset. Similarly, during the chip power-down operation, the chip will produce a power-down reset when the voltage is lower than the minimum working voltage.

When VDD is lower than the specified limit voltage POR/PDR, the system remains reset, and NRST reset pin is at a low level. Please refer to the electric characteristics section of the Data Sheet for details on the reset duration ($t_{RSTTEMPO}$) power-on reset and power-down reset.

Figure 3-2 Power-on reset and power-down reset waveform



3.2.2 Programmable voltage detector

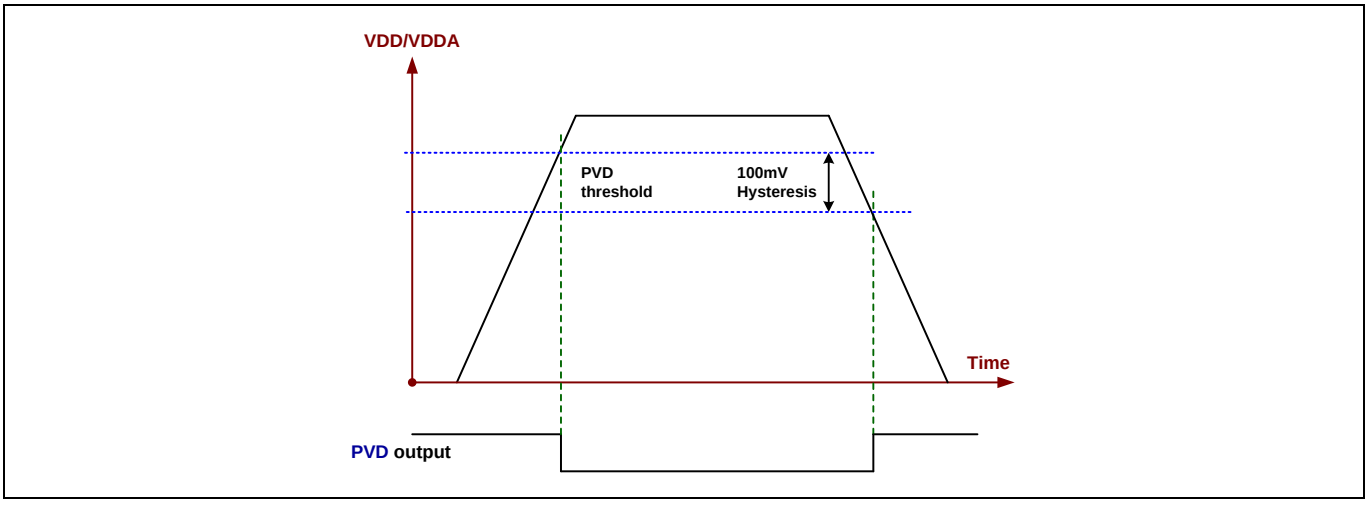
The Programmable Voltage Detector (PVD) can be used to monitor the power voltage of the chip. When the power voltage drops below a specified threshold, it also produces an interrupt, and the software can conduct an emergency treatment. When the power voltage returns to a specified threshold higher than the original one, it also produces an interrupt. The power supply is resumed after the software processing. There is a fixed difference between the power-down threshold and power-rise threshold, and this is PVD lag voltage. This difference can be detected through the listed PVD threshold data. The purpose of introducing the difference is to prevent the frequent interrupt caused by voltage fluctuation around the threshold.

The user can set the threshold voltage of the PLS bit in the power control register (PWR_CR) through the software and compare this value with the power supply voltage of the chip for power monitoring.

PVD can be enabled by setting the PVDE bit in the power control register (PWR_CR). PVDO flag in the power control/status register (PWR_CSR) is to indicate whether VDD is higher or lower than PLS bit threshold voltage in the PWR_CR.

PVD interrupt corresponds to the external interrupt EXTI16. If the user has the EXTI 16 of external interrupt, the event will produce interrupt and enter interrupt service function. When VDD drops below the PVD threshold or VDD rises above the PVD threshold, PVD interrupt will produce (PVD reset also produces via software configuration) according to rising/falling edge of the EXTI 16 of external interrupt. The user may perform some corresponding operations during interrupt. For example, when the condition triggers and power down speed is slower than the execution time of interrupt processing procedure, the emergency close may be executed if the system needs to enter the special protection status. Save some important system data, and conduct corresponding protection operation on the peripherals.

Figure 3-3 PVD threshold waveform



3.3 Power control

3.3.1 Power control overview

In order to extend the service life of battery related products, when MCU doesn't work, MCU may enable low power mode to save power. When MCU needs to work, MCU may be woken by external wake-up source or low power timer so that it works upon need and saves current consumption.

The chip has two low power modes with different power consumption, wake-up time and wake-up sources. Upon need, the user may choose the optimal low power mode.

Two low power modes of the chip:

Sleep Mode: CPU stops, and all chip peripherals, including CPU peripherals, such as NVIC, system clock (SysTick) still run.

Stop Mode: 1.5V domain works at lower power mode. It only keeps content of the register and RAM and CPU and all peripheral clocks stop.

Besides, under Run Mode, reduce power consumption by any mode below:

- Reduce system clock frequency: While meeting system demand, select low speed time frequency or adopt the cyclic changeover of high-speed clock and low speed clock to save power.
- Close unused peripheral clock of APB and AHB bus: The user only enables the required application clock signal. Other excessive clocks are closed.
- Select lower voltage power: The higher the power voltage, the higher the power consumption of the chip. Select the proper power voltage within the chip's safety voltage range.

Table 3-1 Low power modes

Mode	Entry mode	Wake-up mode	Influence on 1.5V domain clock	Influence on VDD domain clock	Voltage regulator	Influence on data and register	Precaution
Sleep Mode	WFI (Wait for Interrupt)	Any interrupt	CPU clock off, no influence on other clock and ADC clock	N/A	On		External clock remains, and the content of register and SRAM remains
	WFE (Wait for Event)	Wake-up event					
Stop Mode	Clear LPDS bit; set SLEEPDEEP bit; WFI or WFE;	Any external interrupt (setup in external interrupt register) or event, IWDG interrupt (no reset) wake-up	All 1.5V domain clocks close	PLL, HSI and HSE oscillators close	On	Content of register and SRAM remains	Unused GPIO before entering low power should be equipped with analog input status

3.3.2 Reducing system clock under run mode

While meeting use demand, select low speed time frequency or adopt the cyclic changeover of high speed clock and low speed clock to save power.

The system clock of the chip may be flexibly configured. The user may select the system clock of different sources or reduce the frequency of any system clock (SYSCLK, HCLK and PCLK1) by configuring different clock dividers.

Prior to entering the sleep mode, reducing peripheral clock may effectively save power under the sleep mode.

3.3.3 External clock control

While the chip executes the program, close the peripheral clock to reduce power consumption.

In sleep mode, close the peripheral clock before executing WFI or WFE instruction and effectively reduce current consumption of peripheral under sleep mode.

The peripheral clock is mainly on the buses of the AHB peripheral clock enable register

(RCC_AHBENR) and APB1 peripheral clock enable register. Users can turn off the peripheral clock by independently configuring the register peripheral control bits.

3.3.4 Sleep mode

3.3.4.1 Enter sleep mode

Execute WFI (Wait For Interrupt)/WFE (Wait for Event) instruction, and request MCU to enter the Sleep Mode. According to the value of SLEEPONEXIT bit inside the CPU system control register (SCB->SCR), there are two options available for selecting the Sleep Mode entry mechanism:

SLEEPNOW: If SLEEPONEXIT bit is cleared, when WFI or WFE is executed, the MCU immediately enters the Sleep Mode.

SLEEPONEXIT: If SLEEPONEXIT bit is asserted and the system exits from the interrupt processing program with the lowest priority, the MCU immediately enters the Sleep Mode.

Under the Sleep Mode, all I/O pins keep the status under the Run Mode.

3.3.4.2 Exit Sleep Mode

The Sleep Mode is woken up immediately after interrupt or event.

Table 3-2 SLEEPNOW mode

SLEEP NOW mode	Description
Entry	Execute WFI (Wait for Interrupt) or WFE (Wait for Event) instructions under conditions below: SLEEPDEEP = 0 SLEEPONEXIT = 0
Exit	If executing WFI, enter Sleep Mode: interrupt (refer to interrupt vector table) If executing WFE, enter Sleep Mode: wake-up event (refer to wake-up event management)
Wake-up latency	Immediate wake-up

Table 3-3 SLEEPONEXIT mode

SLEEP ON EXIT mode	Description
Entry	Execute WFI (Wait for Interrupt) instructions under conditions below: SLEEPDEEP = 0 SLEEPONEXIT = 1
Exit	Interrupt (refer to interrupt vector table)
Wake-up latency	Immediate wake-up

3.3.5 Stop Mode

Stop mode is composed of CPU Deep Sleep Mode and peripheral clock control. In Stop Mode, CPU enters the Deep Sleep Mode, and all clocks in the 1.5V domain are stopped. HSI is disabled, and contents of SRAM and register are saved.

In stop mode, all I/O pins remain in the same state as they were in Run Mode.

If the system clock is switched to LSI, RCC_CFGR.HPRE needs to be configured to 4'b1010 before entering the Stop Mode, and the AHB clock is 8 divisions of the system clock.

3.3.5.1 Enter Stop Mode

The stop mode has two entry methods based on different wake-up methods by programming separate control bits:

- Stop Mode to wait for external interrupt line WFI mode: SLEEPDEEP=1 of CPU system control register (SCR). When WFI is executed, the MCU immediately enters Stop Mode.

- Stop Mode to wait for external event WFE: SLEEPDEEP=1 of CPU system control register (SCR). When WFE is executed, the MCU immediately enters Stop Mode.

Functions below can be selected when entering the stop mode:

- Independent watchdog (IWDG): Initiate the independent watchdog by the key register or hardware written into the independent watchdog. The independent watchdog may select interrupt or reset to wake up chips. When the interrupt wakes up the chip, MCU continues executing the procedure before entering low power. MCU executes reset when the reset is woken. The user may select to close LSI clock source, and thus close independent watchdog.
- Internal low speed oscillator (LSI oscillator): Set by the LSION bit of RCC control/status register (RCC_CSR).

If ADC is not closed before entering the stop mode, ADC continues to consume current. Close the peripheral by setting ADEN bit of the register ADC_ADCFG as 0. The analog input mode should be set for the unused GPIO. Or else, there is still current consumption.

3.3.5.2 Exit Stop Mode

When one interrupt or wake-up event leads to exiting stop mode, the system clock source hardware is auto selected as the HSI oscillator. The user needs to make reconfiguration when other clock source is selected as the clock source.

When the voltage regulator is in run mode and the system exits from the stop mode, there will be an extra startup delay.

Table 3-4 Stop mode

Stop mode	Description
Entry	Execute WFI (Wait for Interrupt) or WFE (Wait for Event) instructions under conditions below: Assert SLEEPDEEP bit in CPU system control register. System clock switches to LSI or HSI; Note: In order to enter the Stop Mode, all external interrupt request bit (interrupt event pending register EXTI_PR) flags must be cleared. Or else, the entry procedure of stop mode will be jumped, and the procedure continues to run.
Exit	Execute WFI (Wait for Interrupt) instructions under conditions below: Any external interrupt line is set as interrupt mode (corresponding external interrupt vector must be enabled in NVIC), refer to interrupt vector table Wait for Event; Execute WFE (Wait for Event) instructions under conditions below: Any external interrupt line is set as event mode, such as RTC clock event wake-up, watchdog interrupt;
Wake-up latency	LSI or HSI wake-up time and extra time produced by voltage regulator wake-up
Precaution	When entering the stop mode, the unused GPIO is set as analog input mode

3.4 Register

Table 3-5 Register overview

Offset	Acronym	Register Name	Reset
0x00	PWR_CR	Power control register	0x00000600
0x04	PWR_CSR	Power control status register	0x00000000
0x24	PWR_CFGR	Power configuration register	0x00000160
0x30	PWR_MEMCR	Power memory control register	0x00000020

3.4.1 PWR_CR Power Control Register

Address offset: 0x00
Reset value: 0x00000600

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

Res

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res			PLS				Res				PVDE	Res.			
			rw								rw				

Bit	Field	Description
31:13	Reserved	Reserved, always read as 0
12:9	PLS	PVD level selection These bits are used to select the voltage threshold for the supply voltage monitor. 0000: 1.8V 0100: 3.0V 1000: 4.2V 0001: 2.1V 0101: 3.3V 1001: 4.5V 0010: 2.4V 0110: 3.6V 1010: 4.8V 0011: 2.7V 0111: 3.9V Others: Reserved Note: Refer to the Electrical Characteristics section of the Datasheet for detailed descriptions.
8:5	Reserved	Reserved, always read as 0
4	PVDE	Power voltage detector enable 1: Turn on PVD 0: Turn off PVD
3:0	Reserved	Reserved, always read as 0

3.4.2 PWR_CSR Power Control/Status Register

Address offset: 0x04
Reset value: 0x00000000 (not cleared when wake up from the Standby Mode)

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

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

Bit	Field	Description
31:3	Reserved	Reserved, always read as 0
2	PVDO	PVD output This bit is valid when PVD is enabled by the PVDE bit. 1: VDD/VDDA is lower than the PVD threshold selected by PLS[3:0] 0: VDD/VDDA is higher than the PVD threshold selected by PLS[3:0]
1:0	Reserved	Reserved, always read as 0

3.4.3 PWR_CFGR Power Configuration Register

Address offset: 0x24
Reset value: 0x00000160

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res						LSICAL					LSICALSEL				
						rw					w				

Bit	Field	Description
31:10	Reserved	Reserved, always read as 0

Bit	Field	Description
9:5	LSICAL	Internal low-speed clock calibration bit During system startup, these bits are initialized as the default calibration value. The user may write other calibration values, and the read out value is always the default calibration value. In case of LSICALSEL=0x1F, the written value may recalibrate LSI frequency. Otherwise, the written value doesn't work.
4:0	LSICALSEL	Internal low speed clock calibration select mode The initial value is 0. When writing a different value: Write 1F: Select the value of register LSICAL. It's still read as 0 after writing 1F. Write others: Select the default calibration value

3.4.4 PWR_MEMCR Power Memory Control Register

Address offset: 0x30
Reset value: 0x00000020

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res									FDP_EN	FWUP_EN	MEM_LOCK				
									rw	rw	w				

Bit	Field	Description
31:7	Reserved	Reserved, always read as 0
6	FDP_EN	Flash DeepStandby enabled in STOP Mode 0: Flash remains in Standby Mode when the chip enters STOP Mode 1: Flash enters DeepStandby Mode when the chip enters STOP Mode
5	FWUP_EN	Flash fast wakeup enable 0: No Flash fast wakeup 1: Use Flash fast wakeup
4:0	MEM_LOCK	PWR_MEMCR configuration lock It must be unlocked before configuring other bits of this register. When MEM_LOCK=0x15, other control bits can be configured. Otherwise, the other bits cannot be changed.

4 RCC Clocks and Reset

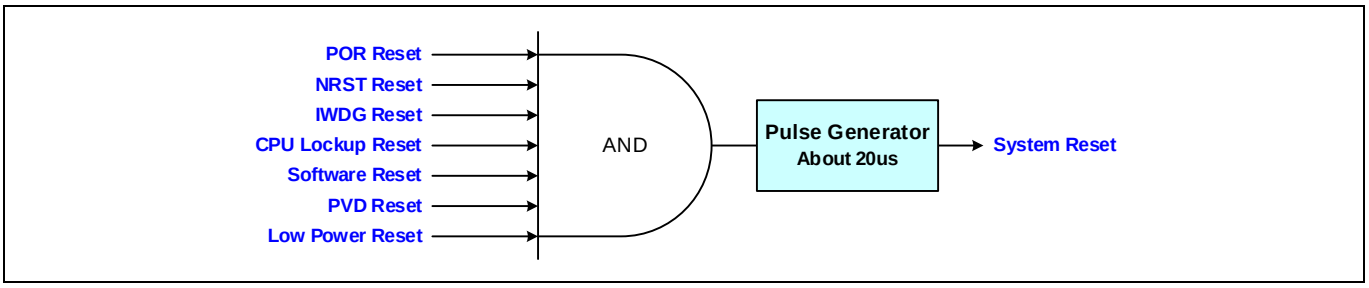
4.1 Reset unit

4.1.1 Introduction

There are two types of reset, defined as power reset and system reset.

4.1.2 Functional block diagram

Figure 4-1 Reset functional block diagram



4.1.3 Main features

- The reset event can be identified by checking the reset flags in the control status register (RCC_CSR).
- Power reset: It sets all registers to their reset values.
- System reset: It sets all registers to their reset values except the reset flags and the LSI oscillator enable flags in the clock control register (RCC_CSR), the wakeup flags in the power control register (PWR_CSR), and the DBG control register (DBG_CR).

4.1.4 Functional description

4.1.4.1 Power Reset (POR Reset)

A power reset is generated in the following ways:

- Power-on Reset
- Power-down Reset

4.1.4.2 System Reset

A system reset is generated in the following ways:

- External reset (NRST reset)
- Independent watchdog reset (IWDG reset)
- Software reset
- CPU lockup reset
- PVD reset
- Low power reset

External reset (NRST reset):

- An external reset is generated when a low level is input on the NRST Pin.

Independent watchdog reset (IWDG reset):

- The counter starts counting down from the reset value of 0xFFF. An independent watchdog reset is generated when the counter reaches 0x0000.
- An independent watchdog reset is generated if the compare/output program is abnormal and the dog cannot be fed properly.
- For details, please refer to the section of Independent Watchdog.

Software reset:

- A software reset is generated if the SCB_AIRCR[SYSRESETREQ] is set to 1.

CPU lockup reset:

- The CPU lockup reset is enabled when the LOCKUPEN bit of the control status register (RCC_CSR) is set to 1;
- The CPU lockup reset will be generated when the CPU enters the lock state.

PVD reset:

- Set the PVDRSTEN bit of the control status register (RCC_CSR) to enable the PVD reset;
- Set the PVDE bit of the power control register (PWR_CR) to enable the PVD;
- Configure the PLS bit of the power control register (PWR_CR) to select the PVD threshold;
- A PVD reset will be generated when the VDD power is below the selected threshold voltage after detection.

Low power reset:

- To prevent the application from entering low power mode by mistake, the option byte space nRST_STOP bit can be set to 0. Then a system reset will be generated before the application enters low power mode by mistake.
- By configuring the option byte space nRST_STOP bit to 0, the system will be reset instead of entering the Stop mode.
- For details, please refer to the section of Embedded Flash Memory.

4.2 Clock unit

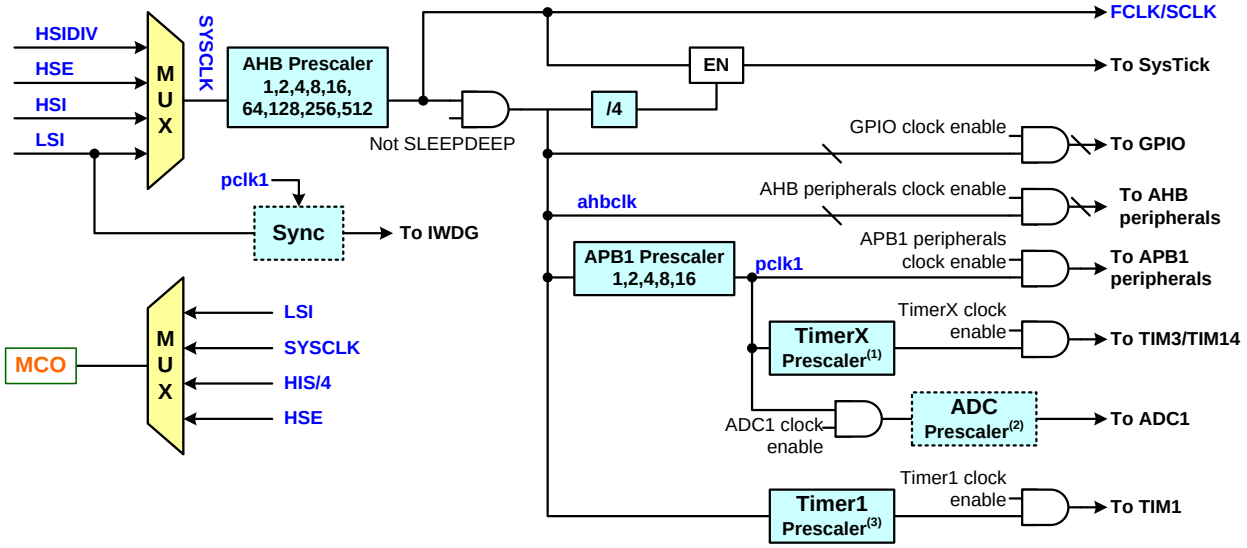
4.2.1 Introduction

Two configurable independent system clock sources:

- High-speed internal clock (HSI)
- Low-speed internal clock (LSI)

4.2.2 Functional block diagram

Figure 4-2 Clock tree



4.2.3 Main features

The clock frequencies of the AHB and APB1 buses are configured individually through the prescale control bits of the clock configuration register (RCC_CFGR). The maximum frequency for both the AHB and the APB1 bus clocks is 60MHz.

4.2.4 Functional description

4.2.4.1 High-speed internal clock (HSI)

The HSI clock signal is generated from an internal 60MHz oscillator. The HSI clock source is enabled by default after the chip is powered on.

Configuration procedure of enabling HSI:

- Set the HSION bit of the clock control register (RCC_CR) to enable the HSI;
- Wait for the HSIRDY bit in the clock control register (RCC_CR) to be set to 1, indicating that the HSI is stable and can output a valid clock; at this time, it can be selected as a system clock or a peripheral clock source.

Notes:

- Once HSI is enabled, the HSI-related configuration cannot be changed. To change the configuration, HSI must be disabled first.

4.2.4.2 Low-speed internal clock (LSI)

As a low-power clock source, the LSI provides a clock source for the independent watchdog. The center frequency of the clock is around 40 kHz. For details, refer to the electrical characteristics section of the datasheets.

Configuration procedure of enabling LSI:

- Set the LSION bit of the control status register (RCC_CSR) to enable the LSI;
- Wait for the LSIRDY bit in the control status register (RCC_CSR) to be set to 1, indicating that the LSI is stable and can output a valid clock.

Notes:

- Once LSI is enabled, the LSI-related configuration cannot be changed. To change the configuration, LSI must be disabled first.

4.2.4.3 Interrupts

Table 4-1 RCC global interrupts

Interrupt Event	Event Flag Bit	Enable Control Bit	Flag Clear Bit
RCC_HSIRDY	HSIRDYF	HSIRDYIE	HSIRDYC
RCC_LSIRDY	LSIRDYF	LSIRDYIE	LSIRDYC

Note: All these flag/control/clear bits can be configured through the clock interrupt register (RCC_CIR).

4.2.4.4 System clock selection (SWS)

Two system clock sources:

- High-speed internal clock (default clock after HSI is powered on)
- Low-speed internal clock (LSI)

Configuration procedure of system clock:

- Enable the desired system clock source (HSI, LSI). Each clock is enabled in a different way; see the HSI and LSI sections for details.
- Wait for the selected clock source RDY signal to be set to 1, indicating that the system clock source is stable (the system clock can only be switched until the target clock source is stable);
- Select the system clock by configuring the SW bit of the clock configuration register (RCC_CFGR);
- Identify the clock source of the current system clock by reading the SWS bit of the clock configuration register (RCC_CFGR).

Notes:

- When LSI is selected as the system clock, you need to configure RCC_CFGR.HPRE to be divided by 8 at first before entering STOP mode. The system clock will automatically switch to HSI after the system is woken up.

4.2.4.5 Peripheral reset

The APB1 peripheral reset register (RCC_APB1RSTR) and AHB peripheral reset register (RCC_AHBRSTR) can be used to implement the software reset of the corresponding peripherals.

4.2.4.6 Microcontroller clock output (MCO)

The microcontroller clock output (MCO) capability allows the clock to be output onto the external MCO pin. The configuration registers of the corresponding GPIO port must be configured as Alternate Function Output. One of the following five clock signals can be selected as the MCO output clock:

Table 4-2 Correspondence of MCO to clock sources

MCO bit of the clock configuration register (RCC_CFGR)	Clock source
00x	No clock output
010	LSI
100	SYSCLK/8
110	HSID8

4.2.4.7 Independent watchdog clock

If the independent watchdog is started by hardware option, the LSI oscillator is automatically turned on and cannot be disabled;

If the independent watchdog is started by software access, the LSI oscillator should be enabled by software. After the LSI oscillator temporization, the clock is provided to the IWDG and the LSI can be turned off by software.

4.3 Register

4.3.1 Overview of registers

Table 4-3 Overview of RCC registers

Offset	Acronym	Register Name	Reset
0x00	RCC_CR	RCC control register	0x0000_0001
0x04	RCC_CFGR	RCC configuration register	0x0000_00A0
0x08	RCC_CIR	RCC clock interrupt register	0x0000_0000
0x10	RCC_AHBRSTR	RCC AHB peripheral reset register	0x0000_0000
0x18	RCC_APB1RSTR	RCC APB1 peripheral reset register	0x0000_0000
0x20	RCC_AHBENR	RCC AHB peripheral clock enable register	0x0000_0014
0x28	RCC_APB1ENR	RCC APB1 peripheral clock enable register	0x8800_0000
0x38	RCC_CSR	RCC control status register	0x0800_0000

4.3.2 RCC_CR clock control register

Address offset: 0x00
Reset value: 0x0000 0001
Access: no wait state, word, half-word and byte access

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.														HSIRDY	HSION
														r	rw

Bit	Field	Description
31:2	Reserved	Reserved, must be kept at reset value
1	HSIRDY	Internal High-speed Clock Ready Flag Set by hardware to indicate that internal clock is stable. After the HSION bit is cleared, HSIRDY goes low after 3 AHB clock cycles. 0: high-speed internal clock not ready 1: high-speed internal clock ready
0	HSION	Internal High-speed Clock Enable Set and cleared by software. Set by hardware to force enable the internal oscillator when exiting the Stop mode. This bit cannot be reset when the system clock has used or will use HSI as the clock source. 0: high-speed internal clock OFF 1: high-speed internal clock ON

4.3.3 RCC_CFGR clock configuration register

Address offset: 0x04
Reset value: 0x0000 00A0
Access: no wait state, word, half-word and byte access
1 or 2 wait states inserted only if the access occurs during clock source switch.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.					MCO			Res.							
					rw										
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.					PPRE1			HPRE				SWS		SW	
					rw			rw				r		rw	

Bit	Field	Description
31:27	Reserved	Reserved, must be kept at reset value
26:24	MCO	Micro Controller Clock Output Set and cleared by software 010: LSI clock output 100: SYSCLK clock output is divided by 8 110: HSI clock output is divided by 8 Others: no clock output Note: This clock output may be disabled at startup or during MCO clock source switching. When the system clock is output to the MCO pin, make sure that this clock does not exceed 50MHz.
23:11	Reserved	Reserved, must be kept at reset value
10:8	PPRE1	PPRE1: APB1 prescale coefficient Set and cleared by software to control the prescale coefficient of the APB1 clock (PCLK1). 0xx: HCLK not divided 100: HCLK is divided by 2 101: HCLK is divided by 4 110: HCLK is divided by 8 111: HCLK is divided by 16
7:4	HPRE	AHB prescale coefficient Set and cleared by software to control the prescale coefficient of the AHB clock. 0xxx: SYSCLK not divided 1000: SYSCLK is divided by 2 1001: SYSCLK is divided by 4 1010: SYSCLK is divided by 8 1011: SYSCLK is divided by 16 Note: The prefetch buffer must be kept on when using a prescale coefficient greater than 1 on the AHB clock. For details, see the section of Flash access control register.
3:2	SWS	System Clock Switch Status 00: HSI used as system clock 11: LSI used as system clock

Bit	Field	Description
1:0	SW	System Clock Switch Set and cleared by software to switch the system clock source. Set by hardware to force HSI to be the system clock when leaving Stop mode. 00: HSI used as system clock 11: LSI used as system clock

4.3.4 RCC_CIR clock Interrupt register

Address offset: 0x08
Reset value: 0x0000 0000
Access: no wait cycle, word, half-word and byte access

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.													HSIRDYC w1c	Res.	LSIRDYC w1c
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.					HSIRDYIE rw	Res.	LSIRDYIE rw	Res.					HSIRDYF r	Res.	LSIRDYF r

Bit	Field	Description
31:19	Reserved	Reserved, must be kept at reset value
18	HSIRDYC	HSI Ready Interrupt Clear Set by software to clear the HSI ready interrupt flag HSIRDYF. 0: no effect 1: clear HSI ready interrupt flag HSIRDYF
17	Reserved	Reserved, must be kept at reset value
16	LSIRDYC	LSI Ready Interrupt Clear Set by software to clear the LSI ready interrupt flag LSIRDYF. 0: no effect 1: clear LSI ready interrupt flag LSIRDYF
15:11	Reserved	Reserved, must be kept at reset value
10	HSIRDYIE	HSI Ready Interrupt Enable Set and cleared by software to enable/disable interrupt caused by the internal oscillator stabilization. 0: HSI ready interrupt disabled 1: HSI ready interrupt enabled
9	Reserved	Reserved, must be kept at reset value
8	LSIRDYIE	LSI Ready Interrupt Enable Set and cleared by software to enable/disable interrupt caused by the internal 40KHz oscillator stabilization. 0: LSI ready interrupt disabled 1: LSI ready interrupt enabled
7:3	Reserved	Reserved, must be kept at reset value
2	HSIRDYF	HSI Ready Interrupt Flag Set to “1” by hardware when the high-speed internal clock is ready. It is cleared by software setting the HSIRDYC bit to “1”. 0: no clock ready interrupt caused by the internal HSI oscillator 1: clock ready interrupt caused by the internal HSI oscillator
1	Reserved	Reserved, must be kept at reset value
0	LSIRDYF	LSI Ready Interrupt Flag Set to “1” by hardware when the low-speed internal clock is ready. It is cleared by software setting the LSIRDYC bit to “1”. 0: no clock ready interrupt caused by the internal 40KHz oscillator 1: clock ready interrupt caused by the internal 40KHz oscillator

4.3.5 RCC_AHBRSTR AHB peripheral reset register

Address offset: 0x10
Reset value: 0x0000 0000
Access: no wait cycle, word, half-word and byte access

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.					HWDIV	Res.							GPIOB	GPIOA	Res.
					rw								rw	rw	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.															DMA
															rw

Bit	Field	Description
31:27	Reserved	Reserved, must be kept at reset value
26	DIVIDER	DIVIDER Reset Set and cleared by software 0: no effect 1: reset
25:19	Reserved	Reserved, must be kept at reset value
18	GPIOB	GPIOB Reset Set and cleared by software 0: no effect 1: reset
17	GPIOA	GPIOA Reset Set and cleared by software 0: no effect 1: reset
16:1	Reserved	Reserved, must be kept at reset value
0	DMA	DMA Reset Set and cleared by software. 0: no effect 1: reset

4.3.6 RCC_APB1RSTR APB1 peripheral reset register

Address offset: 0x18
Reset value: 0x0000 0000
Access: no wait cycle, word, half-word and byte access

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	SYSCFG	DBG	PWR	Res.	OPA2	OPA1	Res.	COMP.	Res.						USART1
	rw	rw	rw		rw	rw		rw							rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.		ADC1	Res.			TIM14	TIM13	Res.		TIM6	Res.			TIM2	TIM1
		rw				rw	rw			rw				rw	rw

Bit	Field	Description
31	Reserved	Reserved, must be kept at reset value
30	SYSCFG	SYSCFG Reset Set and cleared by software 0: no effect 1: reset

Bit	Field	Description
29	DBG	DBG Reset Set and cleared by software 0: no effect 1: reset
28	PWR	Power Interface Reset Set and cleared by software 0: no effect 1: reset
27	Reserved	Reserved, must be kept at reset value
26	OPA2	OPA2 Reset Set and cleared by software 0: no effect 1: reset
25	OPA1	OPA1 Reset Set and cleared by software 0: no effect 1: reset
24	Reserved	Reserved, must be kept at reset value
23	COMP	COMP Reset Set and cleared by software 0: no effect 1: reset
22:17	Reserved	Reserved, must be kept at reset value
16	USART1	USART1 Reset Set and cleared by software 0: no effect 1: reset
15:14	Reserved	Reserved, must be kept at reset value
13	ADC1	ADC1 Reset Set and cleared by software 0: no effect 1: reset
12:10	Reserved	Reserved, must be kept at reset value.
9	TIM14	TIM14 Reset Set and cleared by software 0: no effect 1: reset
8	TIM13	TIM13 Reset Set and cleared by software 0: no effect 1: reset
7:6	Reserved	Reserved, must be kept at reset value
5	TIM6	TIM6 Reset Set and cleared by software 0: no effect 1: reset
4:2	Reserved	Reserved, must be kept at reset value
1	TIM2	TIM2 Reset Set and cleared by software 0: no effect 1: reset
0	TIM1	TIM1 Reset Set and cleared by software 0: no effect 1: reset

4.3.7 RCC_AHBENR AHB peripheral clock enable register

Address offset: 0x20
Reset value: 0x0000 0014
Access: no wait cycle, word, half-word and byte access

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

Res.					HWDIV	Res.							GPIOB	GPIOA	Res.
					rw								rw	rw	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.											FLASH	Res.	SRAM	Res.	DMA
											rw		rw		rw

Bit	Field	Description
31:27	Reserved	Reserved, must be kept at reset value
26	HWDIV	HWDIV clock enable Set and cleared by software 0: clock disabled 1: clock enabled
25:19	Reserved	Reserved, must be kept at reset value
18	GPIOB	GPIOB Clock Enable Set and cleared by software 0: clock disabled 1: clock enabled
17	GPIOA	GPIOA Clock Enable Set and cleared by software 0: clock disabled 1: clock enabled
16:5	Reserved	Reserved, must be kept at reset value
4	Flash	FLASH Clock Enable Set and cleared by software 0: clock disabled 1: clock enabled
3	Reserved	Reserved, must be kept at reset value
2	SRAM	SRAM Clock Enable Set and cleared by software 0: clock disabled 1: clock enabled
1	Reserved	Reserved, must be kept at reset value
0	DMA	DMA Clock Enable Set and cleared by software. 0: clock disabled 1: clock enabled

4.3.8 RCC_APB1ENR APB1 peripheral clock enable register

Address offset: 0x28
Reset value: 0x8800 0000
Access: no wait cycle, word, half-word and byte access
Note: if the peripheral clock is OFF, the software cannot read the value of the peripheral register.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EXTI	SYSCFG	DBG	PWR	IWDG	OPA2	OPA1	Res.	COMP.	Res.						USART1
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
Res.		ADC1	Res.			TIM14	TIM13	Res.		TIM6	Res.			TIM2	TIM1
		rw				rw	rw			rw				rw	rw

Bit	Field	Description
31	EXTI	EXTI Clock Enable Set and cleared by software. 0: clock disabled 1: clock enabled
30	SYSCFG	SYSCFG Clock Enable Set and cleared by software 0: clock disabled 1: clock enabled

Bit	Field	Description
29	DBG	DBG Clock Enable Set and cleared by software 0: clock disabled 1: clock enabled
28	PWR	Power Clock Enable Set and cleared by software 0: clock disabled 1: clock enabled
27	IWDG	IWDG Clock Enable Set and cleared by software. 0: clock disabled 1: clock enabled
26	OPA2	OPA2 Clock Enable Set and cleared by software 0: clock disabled 1: clock enabled
25	OPA1	OPA1 Clock Enable Set and cleared by software 0: clock disabled 1: clock enabled
24	Reserved	Reserved, must be kept at reset value
23	COMP	COMP Clock Enable Set and cleared by software. 0: clock disabled 1: clock enabled
22:17	Reserved	Reserved, must be kept at reset value
16	USART1	USART1 Clock Enable Set and cleared by software 0: clock disabled 1: clock enabled
15:14	Reserved	Reserved, must be kept at reset value
13	ADC1	ADC1 Clock Enable Set and cleared by software 0: clock disabled 1: clock enabled
12:10	Reserved	Reserved, must be kept at reset value.
9	TIM14	TIM14 Clock Enable Set and cleared by software 0: clock disabled 1: clock enabled
8	TIM13	TIM13 Clock Enable Set and cleared by software 0: clock disabled 1: clock enabled
7:6	Reserved	Reserved, must be kept at reset value
5	TIM6	TIM6 Clock Enable Set and cleared by software 0: clock disabled 1: clock enabled
4:2	Reserved	Reserved, must be kept at reset value
1	TIM2	TIM2 Clock Enable Set and cleared by software 0: clock disabled 1: clock enabled
0	TIM1	TIM1 Clock Enable Set and cleared by software 0: clock disabled 1: clock enabled

4.3.9 RCC_CSR control status register

Address offset: 0x38
Reset value: 0x0800 0000
Access: 0 to 3 wait cycles, word, half-word and byte access
Wait states are inserted in the case of successive accesses to this register.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LPWRRSTF	Res.	IWDGRSTF	SFTRSTF	PORRSTF	PINRSTF	Res.	RMVF	LOCKUPF	PVDRSTF	Res.					
r		r	r	r	r		w1c	r	r						
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.								LOCKUPEN	PVDRSTEN	Res.				LSIRDY	LSION
								rw	rw					r	rw

Bit	Field	Description
31	LPWRRSTF	Low Power Reset Flag Set to “1” by hardware when a low power reset occurs. It is only cleared by resetting the power or writing to the RMVF bit via software. 0: no low power reset occurred 1: low power reset occurred
30	Reserved	Reserved, must be kept at reset value
29	IWDGRSTF	Independent Watchdog Reset Flag Set by hardware when an independent watchdog reset occurs. It is only cleared by resetting the power or writing to the RMVF bit via software. 0: no independent watchdog reset occurred 1: independent watchdog reset occurred
28	SFTRSTF	Software Reset Flag Set by hardware when a software reset occurs. It is only cleared by resetting the power or writing to the RMVF bit via software. 0: no software reset occurred 1: software reset occurred
27	PORRSTF	POR/PDR Reset Flag Set by hardware when a POR/PDR reset occurs. It is only cleared by resetting the power or writing to the RMVF bit via software. 0: no POR/PDR reset occurred 1: POR/PDR reset occurred
26	PINRSTF	NRST pin reset flag (PIN Reset Flag) Set by hardware when a NRST pin reset occurs. It is only cleared by resetting the power or writing to the RMVF bit via software. 0: no NRST pin reset occurred 1: NRST pin reset occurred
25	Reserved	Reserved, must be kept at reset value
24	RMVF	Remove Reset Flag It is cleared by software setting this bit to “1”. 0: no effect 1: remove the reset flag
23	LOCKUPF	CPU Lockup Reset Flag Set by hardware when a reset from the CPU lockup occurs. It is only cleared by resetting the power or writing to the RMVF bit via software. 0: no reset from the CPU lockup occurred 1: reset from the CPU lockup occurred
22	PVDRSTF	PVD Reset Flag Set by hardware when a PVD reset occurs. It is only cleared by resetting the power or writing to the RMVF bit via software. 0: no PVD reset occurred 1: PVD reset occurred
21:8	Reserved	Reserved, must be kept at reset value
7	LOCKUPEN	CPU Lockup Reset Enable 0: CPU lockup reset disabled 1: CPU lockup reset enabled
6	PVDRSTEN	PVD Reset Enable 0: PVD reset disabled 1: PVD reset enabled
5:2	Reserved	Reserved, must be kept at reset value

Bit	Field	Description
1	LSIRDY	Internal Low-speed Oscillator Ready Set and cleared by hardware to indicate whether the internal 40KHz oscillator is ready. After the LSION bit is cleared, LSIRDY goes low after 3 AHB clock cycles. 0: internal 40KHz oscillator clock not ready 1: internal 40KHz oscillator clock ready
0	LSION	Internal Low-speed Oscillator Enable Set and cleared by software, or cleared by resetting the power. 0: internal 40KHz oscillator OFF 1: internal 40KHz oscillator ON

5 SYSCFG System Controller

5.1 Introduction

The chip is composed of a set of system configuration registers. Main functions of these registers:

- Manage external interrupt connected to GPIO port (pin configuration)
- Remap memory to code initial area
- System level configuration of some peripherals

5.2 Register

5.2.1 Register overview

Table 5-1 SYSCFG register overview

Offset	Acronym	Register Name	Reset
0x00	SYSCFG_CFGR	SYSCFG configuration register	0x0000000X
0x08	SYSCFG_EXTICR1	SYSCFG external interrupt configuration register 1	0x00000000
0x0C	SYSCFG_EXTICR2	SYSCFG external interrupt configuration register 2	0x00000000
0x10	SYSCFG_EXTICR3	SYSCFG external interrupt configuration register 3	0x00000000
0x14	SYSCFG_EXTICR4	SYSCFG external interrupt configuration register 4	0x00000000

5.2.2 SYSCFG_CFGR Configuration Register

This register has three control bits MEM_MODE, and is used to configure the mapping of different memories to the initial address 0x00000000. The software configures these three control bits to mask BOOT selection. After reset, this control bit is the actual BOOT mode configuration.

Offset address: 0x00

Reset value: 0x0000 000X

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.					USART1_RX_DMA_RMP	USART1_TX_DMA_RMP	ADC1_DMA_RMP	Res.					MEM_MODE		
					rw	rw	rw						rw		

Bit	Field	Description
31:11	Reserved	Reserved, must retain the reset value
10	USART1_RX_DMA_RMP	USART1 RX DMA Remap 0:UART1 RX DMA functionality maps to DMA channel 2 1:UART1 RX DMA functionality maps to DMA channel 1

Bit	Field	Description
9	USART1_TX_DMA_RMP	USART1 TX DMA Remap 0:UART1 TX DMA functionality maps to DMA channel 1 1:UART1 TX DMA functionality maps to DMA channel 2
8	ADC1_DMA_RMP	ADC1 DMA Remap 0: ADC1 DMA functionality maps to DMA channel 1 1: ADC1 DMA functionality maps to DMA channel 2
7:3	Reserved	Reserved, must retain the reset value
2:0	MEM_MODE	Memory selection Bit Control memory internal mapping to address 0x0000 0000. Set and clear these bits by software. After reset, these bit values are determined by BOOT0 pin configuration value and BOOTSEL/nBOOT1/nBOOT0 bit value. 0x0：Main Flash memory mapping to 0x0000 0000 001：System Flash mapping to 0x0000 0000 011：Built in RAM mapping to 0x0000 0000 1x1：IAP/Data memory mapping to 0x0000 0000 Others：Reserved，Main Flash memory mapping to 0x0000 0000

5.2.3 SYSCFG_EXTICR1 External Interrupt Configuration Register 1

Offset address: 0x08
Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXTI3				EXTI2				EXTI1				EXTI0			
rw				rw				rw				rw			

Bit	Field	Description
31:16	Reserved	Reserved, must retain the reset value
15:0	EXTIx	EXTIx configuration（x=0 …3） Select the input source of EXTIx external interrupt. 0000: PA[x] pin 0001: PB[x] pin

5.2.4 SYSCFG_EXTICR2 External Interrupt Configuration Register 2

Offset address: 0x0C
Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXTI7				EXTI6				EXTI5				EXTI4			
rw				rw				rw				rw			

Bit	Field	Description
31:16	Reserved	Reserved, must retain the reset value
15:0	EXTIx	EXTIx configuration（x=4…7） Select the input source of EXTIx external interrupt. 0000: PA[x] pin 0001: PB[x] pin

5.2.5 SYSCFG_EXTICR3 External Interrupt Configuration Register 3

Offset address: 0x10
Reset value: 0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXTI11				EXTI10				EXTI9				EXTI8			
rw				rw				rw				rw			

Bit	Field	Description
31:16	Reserved	Reserved, must retain the reset value
15:0	EXTIx	EXTIx configuration（x=8…11） Select the input source of EXTIx external interrupt. 0000: PA[x] pin 0001: PB[x] pin

5.2.6 SYSCFG_EXTICR4 External Interrupt Configuration Register 4

Offset address: 0x014
Reset value: 0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXTI15				EXTI14				EXTI13				EXTI12			
rw				rw				rw				rw			

Bit	Field	Description
31:16	Reserved	Reserved, must retain the reset value
15:0	EXTIx	EXTIx configuration（x=12…15） Select the input source of EXTIx external interrupt. 0000: PA[x] pin 0001: PB[x] pin

6 HWDIV Hardware Divider

6.1 Overview

Hardware divider is capable of automatic signed or unsigned division of 32-bit integers.

6.2 Main characteristics

- Outputs 32-bit quotient and remainder from 32-bit dividend and divisor
- Completes one division in 8 HCLK cycles
- If the divisor is zero, an overflow interrupt flag will be generated
- Writing the divisor register will trigger the division operation automatically
- Wait automatically via hardware until the calculation is completed when reading the quotient and remainder registers
- Signed or unsigned division of integers

6.3 Functional description

The hardware division unit consists of four 32-bit data registers which are dividend, divisor, quotient and remainder, and the unit is capable of signed or unsigned 32-bit division. The USIGN bit in Hardware Division Control Register HWDIV_CR can be used to select whether the division is signed or unsigned.

Each time the divisor register is written, the division operation is automatically triggered. After the operation is completed, the result is written to the quotient and remainder registers. If the quotient register, remainder register, or status register is read before the calculation is completed, the read operation is held. The result will not return until the end of the calculation. If the divisor is zero, an overflow interrupt flag will be generated.

The divisor and dividend registers must be initialized each time before the values of the quotient, remainder, and status registers can be read.

6.4 Register

6.4.1 Overview of registers

Table 6-1 Overview of HWDIV registers

Offset	Acronym	Register Name	Reset
0x00	HWDIV_DVDR	Dividend register	0x00000000
0x04	HWDIV_DVSR	Divisor register	0x00000001
0x08	HWDIV_QUOTR	Quotient register	0x00000000
0x0C	HWDIV_RMDR	Remainder register	0x00000000
0x10	HWDIV_SR	Status register	0x00000000
0x14	HWDIV_CR	Control register	0x00000001

6.4.2 HWDIV_DVDR Dividend Register

Address offset: 0x00
Reset value: 0x0000 0000

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

Bit	Field	Description
31:0	DIVIDEND	Dividend data

6.4.3 HWDIV_DVSR Divisor Register

Address offset: 0x04
Reset value: 0x0000 0001

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

Bit	Field	Description
31:0	DIVISOR	Each time the divisor register (Divisor data) is written, the division operation is automatically triggered.

6.4.4 HWDIV_QUOTR Quotient Register

Address offset: 0x08
Reset value: 0x0000 0000

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

Bit	Field	Description
31:0	QUOTIENT	Quotient data

6.4.5 HWDIV_RMDR Remainder Register

Address offset: 0x0C
Reset value: 0x0000 0000

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

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

Bit	Field	Description
31:0	REMAINDER	Remainder data

6.4.6 HWDIV_SR Status Register

Address offset: 0x10
Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.															OVF
															r

Bit	Field	Description
31:0	Reserved	Reserved, always read as 0
0	OVF	Overflow flag bit is cleared by writing 1 in the software during the next division operation 1: Current divisor is zero. 0: Current divisor is not zero.

6.4.7 HWDIV_CR Control Register

Address offset: 0x14
Reset value: 0x0000 0001

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

Bit	Field	Description
31:2	Reserved	Reserved and always read as 0
1	OVFE	Overflow interrupt enable 1: Overflow interrupt enabled 0: Overflow interrupt enabled
0	USIGN	Unsigned enable 1: Unsigned division 0: Signed division

7 EXTI Interrupt and Event

7.1 Introduction

Nestled vector interrupt controller (NVIC) connects the processor core and manages the anomaly and interrupt processing with low latency. NVIC includes 2-bit interrupt priority levels, which can thus provide 4 interrupt priority levels. For other more anomalies and NVIC programming details, please refer to the Cortex-Mx Technology Reference Manual.

The EXTI module includes the edge detection circuit, which can produce the interrupt request or wake-up event, and the edge detection supports the rising edge, falling edge or any edge configuration. Each edge detection circuit supports the independent enable and mask.

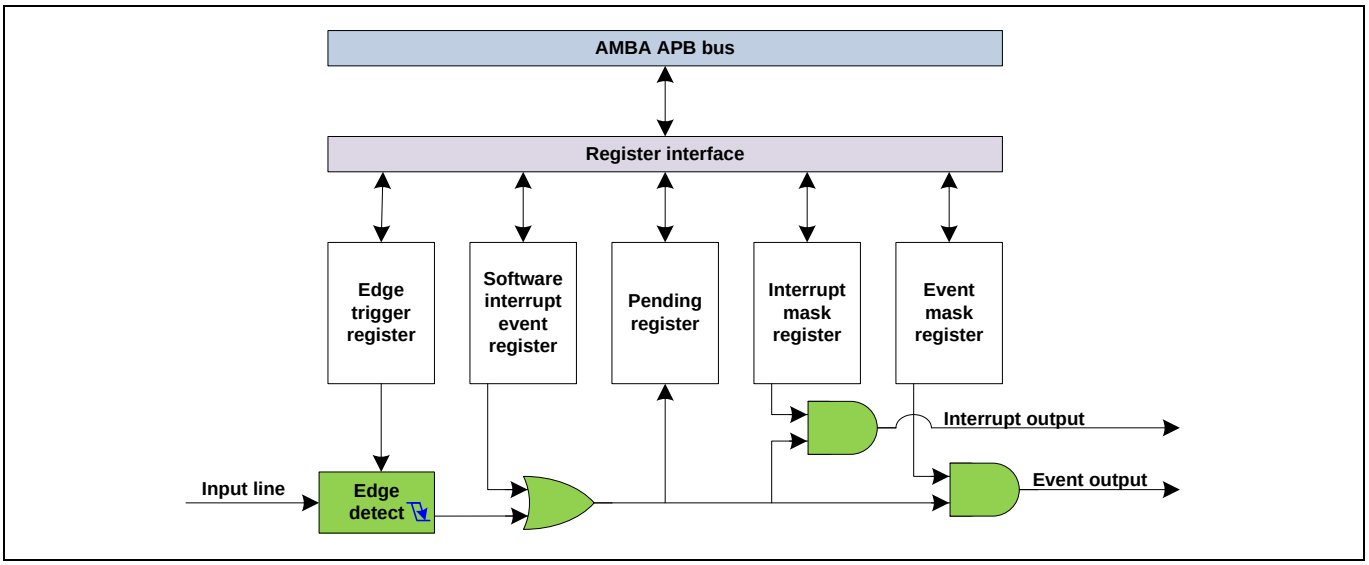
7.2 Main characteristics

- Independently trigger and mask each interrupt
- Configure interrupt/event output by software
- Produce the wake-up event to wake up the low power mode
- The pending register saves the status of each corresponding interrupt line
- All GPIOs supports the trigger source configured as EXTI
- Support the rising edge trigger, falling edge trigger and any edge trigger

7.3 Function description

7.3.1 Function block diagram

Figure 7-1 EXTI structure block diagram



7.3.2 Interrupt and anomaly vector

Under Handler mode, Cortex-M0 processor and nested interrupt vector control (NVIC) can process all anomalies based on priority. In case of anomaly, the system will pop the current processing work stack after interrupt service. Simultaneous processing of vector and current work status improves the interrupt efficiency. The table below presents the anomaly type and interrupt vector.

Table 7-1 Abnormal vector

Position	Priority	Priority type	Name	Description	Address
				Reserved	0x0000 0000
	-3	Fixed	Reset	Reset	0x0000 0004
	-2	Fixed	NMI	Unmaskable interrupt	0x0000 0008
	-1	Fixed	Hardware fault	All types of faults	0x0000 000C

Table 7-2 Interrupt vector

Position	Priority	Priority type	Name	Description	Address
	3	Settable	SVCall	System service call via SWI instruction	0x0000 002C
				Reserved	0x0000 0030
				Reserved	0x0000 0034
	6	Settable	PendSV	Pending system service	0x0000 0038
	7	Settable	SysTick	System tick timer	0x0000 003C
0	8	Settable	IWDG	independent watchdog interrupt connected to EXTI 24	0x0000_0040
1	9	Settable	PVD	Power voltage detection (PVD) connected to EXTI 16	0x0000_0044
2	10	Settable	Reserved	Reserved	0x0000_0048
3	11	Settable	FLASH	Flash global interrupt	0x0000_004C
4	12	Settable	RCC	RCC global interrupt	0x0000_0050
5	13	Settable	EXTI0_1	EXTI line [1:0] interrupt	0x0000_0054
6	14	Settable	EXTI2_3	EXTI line [3:2] interrupt	0x0000_0058
7	15	Settable	EXTI4_15	EXTI line [15:4] interrupt	0x0000_005C
8	16	Settable	HWDIV	Hard Divider interrupt	0x0000_0060
9	17	Settable	DMA channel1	DMA channel 1 global interrupt	0x0000_0064
10	18	Settable	DMA channel2	DMA channel 2 global interrupt	0x0000_0068
11	19	Settable	Reserved	Reserved	0x0000_006C
12	20	Settable	ADC	ADC global interrupt	0x0000_0070
13	21	Settable	TIM1_BRK_UP_TRG_COM	TIM1 break refresh trigger COM interrupt	0x0000_0074
14	22	Settable	TIM1_CC	TIM1 capture compare interrupt	0x0000_0078
15	23	Settable	TIM2	TIM2 global interrupt	0x0000_007C
16	24	Settable	TIM6	TIM6 global interrupt	0x0000_0080
17	25	Settable	Reserved	Reserved	0x0000_0084
18	26	Settable	Reserved	Reserved	0x0000_0088
19	27	Settable	Reserved	Reserved	0x0000_008C
20	28	Settable	TIM13	TIM13 global interrupt	0x0000_0090
21	29	Settable	TIM14	TIM14 global interrupt	0x0000_0094
22	30	Settable	Reserved	Reserved	0x0000_0098
23	31	Settable	Reserved	Reserved	0x0000_009C
24	32	Settable	COMP1/2	Comparator 1/2 output connected to EXTI19 and EXTI20	0x0000_00A0
25	33	Settable	Reserved	Reserved	0x0000_00A4
26	34	Settable	Reserved	Reserved	0x0000_00A8

Position	Priority	Priority type	Name	Description	Address
27	35	Settable	USART1	USART1 global interrupt	0x0000_00AC
28	36	Settable	Reserved	Reserved	0x0000_00B0
29	37	Settable	Reserved	Reserved	0x0000_00B4
30	38	Settable	Reserved	Reserved	0x0000_00B8
31	39	Settable	Reserved	Reserved	0x0000_00BC

7.3.3 Wake-up event management

All EXTI lines support the generation of interrupt or the event is used to wake the system from the low power mode. The user executes the WFE instruction to enter the corresponding low power mode. Configure the EXTI line event output to wake up the system. The user executes the WFI to enter the low power mode, and configures EXTI line interrupt output to wake up the system. For the detailed configuration, please refer to the section of power control.

7.3.4 Interrupt function description

To enable interrupt function and produce interrupt, configure the edge detection trigger register as the required trigger type, and open the corresponding bit of interrupt mask register allowing interrupt request. When the corresponding external interrupt line detects the configured trigger condition, it produces an interrupt request and the corresponding pending position of register is 1. Write the corresponding bit of pending register as 1 and clear the interrupt.

To configure the generation event, configure the edge detection trigger register as the required trigger type, and open the corresponding bit of interrupt mask register allowing interrupt request. When the corresponding external interrupt line detects the configured trigger condition, it produces an interrupt request.

7.3.5 Hardware interrupt output

Specific steps to configure hardware interrupt source:

- Open the mask bit of the corresponding interrupt line (EXTI_IMR), enable interrupt.
- Configure the trigger register bit (EXTI_RTSR/EXTI_FTSR) of the corresponding interrupt line.
- Open the NVIC connected interrupt channel so that the interrupt request can be transmitted to CPU and correctly responded.

When configuring EXTIx (x=31~0) line and producing interrupt output, the corresponding bit of EXTI_PR register will be set 1. It's required to clear the corresponding pending bit of EXTI_PR register before again detecting the rolling of EXTIx (x=31~0) line and producing the interrupt.

There are three ways to clear the pending bit of EXTI_PR register:

- Write the pending bit of EXTI_PR register as 1.
- When configuring the rising edge trigger selection register (EXTI_RTSR), writing 0 to the corresponding bit will clear the pending bit. When configuring the falling edge trigger selection register (EXTI_FTSR), writing 0 in the corresponding bit will clear the pending bit.
- Clear by changing the edge detection polarity of EXTI line.

7.3.6 Hardware event output

Specific steps to configure hardware event source:

- Open the mask bit (EXTI_EMR) of the corresponding event line;

- Configure the trigger register bit (EXTI_RTSR/EXTI_FTSR) of the corresponding event line.

7.3.7 Software interrupt and event output

Specific steps to support configuration of interrupt and event by software:

- Enable event or interrupt enable bit (EXTI_IMR, EXTI_EMR).
- Configure the corresponding bit of the software interrupt event register as 1 (EXTI_SWIER).

7.3.8 External interrupt mapping

All GPIOs are used as the EXTI trigger source for producing the interrupt or event request. Configure SYSCFG_EXTICRx register, and meanwhile support the internal module (including PVD, RTC, USB, comparator, IWDG) trigger.

The specific connection relationships are shown below:

Table 7-3 EXTI trigger source

External interrupt line	IO mapping	Control bit
EXTI0	PA0;PB0	SYSCFG_EXTICR1 register EXTI0
EXTI1	PA1;PB1	SYSCFG_EXTICR1 register EXTI1
EXTI2	PA2;PB2	SYSCFG_EXTICR1 register EXTI2
EXTI3	PA3;PB3	SYSCFG_EXTICR1 register EXTI3
EXTI4	PA4;PB4	SYSCFG_EXTICR2 register EXTI4
EXTI5	PA5;PB5	SYSCFG_EXTICR2 register EXTI5
EXTI6	PA6;PB6	SYSCFG_EXTICR2 register EXTI6
EXTI7	PA7;PB7	SYSCFG_EXTICR2 register EXTI7
EXTI8	PA8;PB8	SYSCFG_EXTICR3 register EXTI8
EXTI9	PA9;PB9	SYSCFG_EXTICR3 register EXTI9
EXTI10	PA10	SYSCFG_EXTICR3 register EXTI10
EXTI11	PA11	SYSCFG_EXTICR3 register EXTI11
EXTI12	PA12	SYSCFG_EXTICR4 register EXTI12
EXTI13	PA13	SYSCFG_EXTICR4 register EXTI13
EXTI14	PA14	SYSCFG_EXTICR4 register EXTI14
EXTI15	PA15	SYSCFG_EXTICR4 register EXTI15

Other external interrupt/event controller connection:

- EXTI line 16 connected to PVD output
- EXTI line 19 connected to comparator1 output
- EXTI line 20 connected to comparator2 output
- EXTI line 24 connected to IWDG interrupt

7.4 Register

7.4.1 Register overview

Table 7-4 EXTI register overview

Offset	Acronym	Register Name	Reset
0x00	EXTI_IMR	Interrupt mask register	0x00000000

Offset	Acronym	Register Name	Reset
0x04	EXTI_EMR	Event mask register	0x00000000
0x08	EXTI_RTSR	Rising edge trigger selection register	0x00000000
0x0C	EXTI_FTSR	Falling edge trigger selection register	0x00000000
0x10	EXTI_SWIER	Software interrupt event register	0x00000000
0x14	EXTI_PR	Pending register	0x00000000

7.4.2 EXTI_IMR Interrupt Mask Register

Address offset: 0x0
Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.							IMRx(x=24~16)								
							rw								
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IMRx(x=15~0)															
rw															

Bit	Field	Description
31:25	Reserved	Reserved, must be kept at reset value.
24:0	IMRx	Line x interrupt enable bit 1: Configure the bit as 1, enable line x corresponding interrupt 0: Configure the bit as 0, disable line x corresponding interrupt

7.4.3 EXTI_EMR Event Mask Register

Address offset: 0x04
Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.							EMRx(x=24~16)								
							rw								
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EMRx(x=15~0)															
rw															

Bit	Field	Description
31:25	Reserved	Reserved, must be kept at reset value.
24:0	EMRx	Line x event enable bit 1: Configure the bit as 1, enable line x corresponding event 0: Configure the bit as 0, disable line x corresponding event

7.4.4 EXTI_RTSR Rising Edge Trigger Selection Register

Address offset: 0x08
Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.							TRx(x=24~16)								
							rw								
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TRx(x=15~0)															

rw

Bit	Field	Description
31:25	Reserved	Reserved, must be kept at reset value.
24:0	TRx	Line x corresponding interrupt or event trigger polarity 1: Configure the bit as 1, enable line x corresponding rising edge trigger interrupt or event 0: Configure the bit as 0, disable line x corresponding rising edge trigger interrupt or event

7.4.5 EXTI_FTSR Falling Edge Trigger Selection Register

Address offset: 0x0C
Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.							TRx(x=24~16)								
							rw								
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TRx(x=15~0)															
rw															

Bit	Field	Description
31:25	Reserved	Reserved, must be kept at reset value.
24:0	TRx	Line x corresponding interrupt or event trigger polarity 1: Configure the bit as 1, enable line x corresponding falling edge trigger interrupt or event 0: Configure the bit as 0, disable line x corresponding falling edge trigger interrupt or event

7.4.6 EXTI_SWIER Software Interrupt Event Register

Address offset: 0x10
Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.							SWIERx(x=24~16)								
							rw								
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SWIERx(x=15~0)															
rw															

Bit	Field	Description
31:25	Reserved	Reserved, must be kept at reset value.
24:0	SWIERx	Line x software interrupt or event enable Writing 1 will set the corresponding pending bit of EXTI_PR register. Meanwhile, configuring the corresponding bit of EXTI_IMR or EXTI_EMR as 1 can produce interrupt or event.
		Note: Writing 1 to the corresponding bit of EXTI_PR register may clear the bit

7.4.7 EXTI_PR Software Interrupt Event Pending Register

Address offset: 0x14
Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.							PRx(x=24~16)								
							rc_w1								

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PRx(x=15~0)															
rc_w1															

Bit	Field	Description
31:25	Reserved	Reserved, must be kept at reset value.
24:0	PRx	Line x trigger pending bit 1: Selected trigger request 0: No trigger request In case of the selected edge event on the external interrupt line, the bit is set as 1. Writing 1 clears the bit, or clear by changing the edge detection polarity.

8 DMA Direct Memory Access Controller

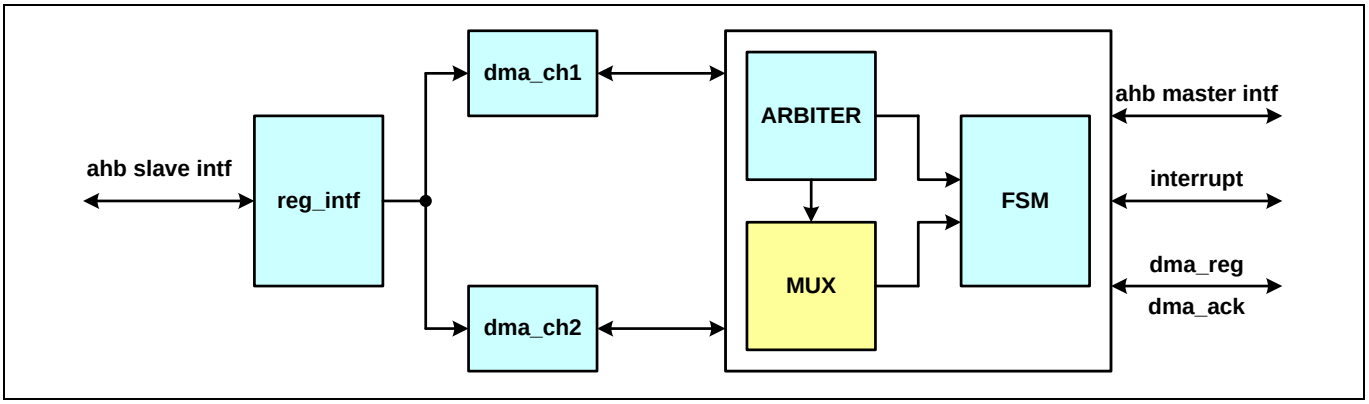
8.1 Overview

The DMA controller provides high-speed and automatic data transfer without any CPU action through a shared system bus.

The DMA controller has 2 channels, and multiple peripheral DMA requests are sent to the corresponding channels for processing.

8.2 DMA functional block diagram

Figure 8-1 DMA functional block diagram



8.3 Main characteristics

- 2 independent channels with relevant functions configurable via registers
- Hardware DMA requests are directly connected to their dedicated DMA channels. DMA requests can also be triggered on each channel if the registers are configured via software.
- Priorities between two requests on the channels are decided by software configuring registers (4 levels consisting of very high, high, medium, low) or by hardware automatically in case of equality (the channel with the lower number gets priority).
- Independent source and destination data zone transfer size, which can be configured as byte, half word, and word.
- The DMA packs transfers according to the data width of the data source and unpacks transfers according to the data widths of the destination in the destination data zone. Source/destination addresses must be aligned on their respective data transfer size.
- Support for circular buffer control.

- 3 event flags (DMA Half Transfer, DMA Transfer Complete and DMA Transfer Error) are supported on each channel. These three event flags are logically ORed together in a single interrupt request for each channel.
- Support for Memory-to-memory transfer.
- Supported data transfer directions: peripheral-to-memory and memory-to-peripheral transfers.
- Access to peripherals on SRAM, APB1 and AHB bus as data source and destination.
- The number of data to be transferred is up to 65535, which can be set via software by programming the corresponding registers.

8.4 Interrupt

Three event flags (DMA Half Transfer, DMA Transfer Complete and DMA Transfer Error) can be generated on each DMA channel. These three event flags are logically ORed together in a single interrupt request for each channel.

Separate interrupt enable bits are configurable in the registers to meet various program demands.

Table 8-1 DMA interrupt request

Interrupt Event	Event Flag	Enable Control Bit
Half-transfer	HTIF	HTIE
Transfer complete	TCIF	TCIE
Transfer error	TEIF	TEIE

8.5 DMA

8.5.1 DMA request mapping

Multiple transfer requests from the peripherals enter the DMA controller through DMAMAP. To avoid conflicts, only one peripheral DMA request must be enabled at a time on one channel. Refer to the following figure DMA request mapping for more details.

The peripheral transfer requests should be independently controlled by programming the DMA enable bit in the control registers of the corresponding peripheral.

Table 8-2 DMA request list for DMA channels

Peripheral	DMA Channel 1	DMA Channel 2
ADC	ADC(1)	ADC(2)
USART	USART_TX(1) USART_RX(2)	USART_RX(1) USART_TX(2)
TIM1	TIM1_CH2 TIM1_CH4 TIM1_TRIG TIM1_COM	TIM1_CH1 TIM1_CH3 TIM1_UP TIM1_CH5
TIM2	TIM2_CH3 TIM2_UP	TIM2_CH1 TIM2_CH2 TIM2_CH4
TIM6	-	TIM6_UP
TIM13	-	TIM13_CH1 TIM13_UP
TIM14	TIM14_CH1 TIM14_UP	-

- If the corresponding mapping bit of SYSCFG_CFGR register is reset, the DMA request is mapped on this DMA channel.
- If the corresponding mapping bit of the SYSCFG_CFGR register is set, the DMA request is mapped on this DMA channel.

8.6 Functional description

DMA and CPU both access the memory or peripheral data through the system bus. When the CPU and DMA accesses conflict, the DMA request may occupy the system bus, and then the CPU can only wait until the DMA transfer completes to release the bus. If DMA keeps occupying the bus, the CPU may fail to work. To prevent this situation, the bus arbiter implements round-robin scheduling, thus ensuring at least half of the bus control system for the CPU.

8.6.1 DMA transactions

After generating a related event, the peripheral sends a DMA request signal to the corresponding channel of the DMA controller. The DMA controller serves the request depending on the DMA channel priorities set by software, or default hardware rules. The DMA will acknowledge the peripheral request. As soon as the DMA controller accesses the peripheral through the bus, an Acknowledge is sent to the peripheral by the DMA controller, informing the peripheral that this request has been acknowledged. The peripheral releases its request as soon as it gets the Acknowledge from the DMA controller. Once the DMA detects that the peripheral request has disappeared, the corresponding Acknowledge signal is released. This DMA transfer is then finished.

In summary, each DMA transfer consists of three operations:

- The loading of source address which is configured by software.
- The storage of the data is loaded to a designed location which is configured by software.
- One execution of DMA transfer. The DMA_CNDTRx counter decrements from the programmed number of transfers, indicating the number of DMA transfers that have still to be performed.

The operation sequence for each transfer is as follows:

- Turn on DMA enable
- Turn on the peripheral DMA request
- Turn off the peripheral DMA request when the transfer is completed
- Repeat steps 1-3

8.6.2 Arbiter

The arbiter decides which DMA request gets priority to be handled by the DMA controller. The priorities are managed in two logics, software and hardware:

- Software: There are four priority levels. Each channel priority can be configured in the DMA_CCRx register:
  - Very high priority
  - High priority
  - Medium priority
  - Low priority
- Hardware: Requests with higher software priority level are handled first. If requests have the same

software priority level, the channel with the lower number will get priority versus the channel with the higher number by default.

8.6.3 DMA channels

DMA transfer is available between the fixed addresses of a peripheral register and a memory through the DMA channel. The number of data to be transferred is up to 65535, which can be set via software by programming the corresponding registers. After each transfer, the DMA_CNDTRx counter decrements from the programmed number of transfers, indicating the number of DMA transfers that have still to be performed.

8.6.3.1 Programmable data width

Transfer data sizes of the peripheral and memory are fully programmable through the PSIZE and MSIZE bits in the DMA_CCRx register.

8.6.3.2 Pointer increment

The access addresses of the peripheral and memory can be incremented by step size, depending on the PINC and MINC flag bits in the DMA_CCRx register. It is not necessary to configure the access address for each transaction.

If the incremented mode register is cleared, a fixed address will be accessed for each DMA transfer.

If incremented mode is enabled, the address of the next transfer will be the address of the previous one incremented by step size. The step size depends on the chosen data size, which is 1 for 8-bit data size, 2 for 16-bit data size, and 4 for 32-bit data size. The first transfer address is stored in the DMA_CPARx/DMA_CMARx registers.

The channel is configured in non-circular mode, and no DMA transfer is performed once the DMA_CNDTRx decrements down to 0.

8.6.3.3 Channel configuration

The following sequence should be followed to configure a DMA channel x (where x is the channel number):

- Set the peripheral register address in the DMA_CPARx register. The DMA transfer direction decides whether the peripheral address is the source or destination.
- Set the memory address in the DMA_CMARx register. The DMA transfer direction decides whether the data will be loaded from or stored to this memory.
- Configure the number of DMA transfers in the DMA_CNDTRx register. After each DMA transfer is completed, this value will be decremented by 1. Writing to this register via software is forbidden during the DMA transfer.
- Configure the channel priority using the PL[1:0] bits in the DMA_CCRx register.
- Configure data transfer direction, circular mode, peripheral & memory incremented mode, peripheral & memory data size, and interrupt types in the DMA_CCRx register.
- Activate the channel by setting the ENABLE bit in the DMA_CCRx register. As soon as the DMA channel is enabled, it can operate normally. It will serve any request from the peripheral to perform DMA transfer.

If the half-transfer flag (HTIF) is set by hardware, it means the current number of DMA transfers is half of the programmed value. To generate an interrupt, the Half-Transfer Interrupt Enable bit (HTIE)

should be enabled.

If the Transfer Complete Flag (TCIF) is set by hardware, it means the programmed number of DMA transfers is completed currently. To generate an interrupt, the Transfer Complete Interrupt Enable bit (TCIE) should be enabled.

8.6.3.4 Circular mode

To handle circular read/write buffers or continuous data flows (such as ADC scan mode), the circular mode can be used. This mode can be enabled by setting the CIRC bit in the DMA_CCRx register. When DMA_CNDTRx decreases to 0 in the circular mode, it will be automatically reloaded with the value programmed previously and then decrement again. The DMA transfer will continue.

8.6.3.5 Memory-to-memory mode

The DMA supports memory-to-memory access, without the participation of a peripheral. If the MEM2MEM bit in the DMA_CCRx register is set, then the DMA channel initiates transfers as soon as the channel enable bit is set in the DMA_CCRx register. The DMA transfer stops once the DMA_CNDTRx register decrements down to zero.

Memory-to-memory access does not support the circular mode.

8.6.4 Programmable data width, data alignment and endian

When PSIZE and MSIZE are not equal, the DMA module performs some data alignments as described in the following table.

The bits of source are loaded to the destination address in alignment.

If the data width of the destination data is greater than that of the source data, the extra bits in the destination data is filled with zero.

If the data width of the destination data is shorter than that of the source data, the extra bits in the source data is truncated.

Table 8-3 Programmable data width & endian behavior (PINC = MINC = 1), with the number of transfers as 4

Composite Type	Transfer Size		Transfer Operations	
	Source	Destination	Source (address I data)	Destination (address I data)
The data widths of the source and destination data are equal The address step is consistent with the data size	8	8	READ B0[7:0] @0x0 READ B1[7:0] @0x1 READ B2[7:0] @0x2 READ B3[7:0] @0x3	WRITE B0[7:0] @0x0 WRITE B1[7:0] @0x1 WRITE B2[7:0] @0x2 WRITE B3[7:0] @0x3
	16	16	READ B1B0[15:0] @0x0 READ B3B2[15:0] @0x2 READ B5B4[15:0] @0x4 READ B7B6[15:0] @0x6	WRITE B1B0[15:0] @0x0 WRITE B3B2[15:0] @0x2 WRITE B5B4[15:0] @0x4 WRITE B7B6[15:0] @0x6
	32	32	READ B3B2B1B0[31:0] @0x0 READ B7B6B5B4[31:0] @0x4 READ BBBAB9B8[31:0] @0x8 READ BFBEBDBC[31:0] @0xC	WRITE B3B2B1B0[31:0] @0x0 WRITE B7B6B5B4[31:0] @0x4 WRITE BBBAB9B8[31:0] @0x8 Write BFBEBDBC[31:0] @0xC
The data width of the source is shorter than the data width of the destination The address step is the data size divided by 8 Extra bits in the destination data is filled with 0	8	16	READ B0[7:0] @0x0 READ B1[7:0] @0x1 READ B2[7:0] @0x2 READ B3[7:0] @0x3	WRITE 00B0[15:0] @0x0 WRITE 00B1[15:0] @0x2 WRITE 00B2[15:0] @0x4 WRITE 00B3[15:0] @0x6
	8	32	READ B0[7:0] @0x0 READ B1[7:0] @0x1 READ B2[7:0] @0x2 READ B3[7:0] @0x3	WRITE 000000B0[31:0] @0x0 WRITE 000000B1[31:0] @0x4 WRITE 000000B2[31:0] @0x8 WRITE 000000B3[31:0] @0xC
	16	32	READ B1B0[15:0] @0x0 READ B3B2[15:0] @0x2 READ B5B4[15:0] @0x4 READ B7B6[15:0] @0x6	WRITE 0000B1B0[31:0] @0x0 WRITE 0000B3B2[31:0] @0x4 WRITE 0000B5B4[31:0] @0x8 WRITE 0000B7B6[31:0] @0xC
	16	8	READ B1B0[15:0] @0x0	WRITE B0[7:0] @0x0

Composite Type	Transfer Size		Transfer Operations	
	Source	Destination	Source (address I data)	Destination (address I data)
The data width of the source is greater than the data width of the destination The address step is the data size divided by 8 The destination data is insufficient, extra bits are truncated			READ B3B2[15:0] @0x2 READ B5B4[15:0] @0x4 READ B7B6[15:0] @0x6	WRITE B2[7:0] @0x1 WRITE B4[7:0] @0x2 WRITE B6[7:0] @0x3
	32	8	READ B3B2B1B0[31:0] @0x0 READ B7B6B5B4[31:0] @0x4 READ BBBAB9B8[31:0] @0x8 READ BFBEBDBC[31:0] @0xC	WRITE B0[7:0] @0x0 WRITE B4[7:0] @0x1 WRITE B8[7:0] @0x2 WRITE BC[7:0] @0x3
	32	16	READ B3B2B1B0[31:0] @0x0 READ B7B6B5B4[31:0] @0x4 READ BBBAB9B8[31:0] @0x8 READ BFBEBDBC[31:0] @0xC	WRITE B1B0[15:0] @0x0 WRITE B5B4[15:0] @0x2 WRITE B9B8[15:0] @0x4 WRITE BDBC[15:0] @0x6

8.6.4.1 Addressing an AHB peripheral that does not support byte or half word write operations

HSIZE represents the data width in the AHB bus transmission. If the destination device does not support byte or half word operations, it means the corresponding slave is not configured to handle HSIZE. The destination devices agree that the data is transferred by word (32-bit).

AHB devices usually support word (32-bit) operations. When some AHB devices do not support byte (8-bit) or half word (16-bit) write operations, DMA will expand them to word (32-bit) operations. At the same time, the corresponding destination address step should be changed to 4 corresponding to 32 bits, and the destination data width should be configured to 32 bits.

For example: The memory is configured as a data source. The data width is 8 bits, and the data to be transferred is 0xDA. We want to transfer it to the address 0x2 of the destination device. Since the corresponding slave device does not support 8-bit/16-bit operation, it will consider it is receiving the 32-bit 0x0000_00DA, and write the 32-bit 0x0000_00DA to the address 0x0 (for a 32-bit operation, the address unit is 0x4). That is to say, the address 0x2 corresponds to data 0x00.

Therefore, it is necessary to copy and extend the data. 0xDA will be copied and extended into four 8-bit data which are combined to a 32-bit datum 0xDADA_DADA. Considering the destination address 0x2, the slave will receive the 32-bit 0xDADA_DADA and then write it to the address 0x0. In this way, the data at the destination address 0x2 is the desired 0xDA.

Indeed, redundant data is written to the destination register with this method. However, it is assured that the data written to the destination address is what was programmed, and no errors occur.

The 8-bit source data is copied and extended (multiplied by 4) into 32-bit data, e.g. 0x12 will be expanded to 0x1212_1212.

The 16-bit source data is copied and extended (multiplied by 2) into 32-bit data, e.g. 0x1234 will be expanded to 0x1234_1234.

Only 32-bit transfer is supported. Data width of slave devices that do not support 8-bit/16-bit should be configured to 8-bit/16-bit.

8.6.5 Error management

There are reserved areas that are not allowed to be accessed in the address space. These reserved areas may be accessed when the DMA transfer address is automatically incremented or when an address is specified. The DMA transfer error interrupt flag (TEIF) is set to '1' when the DMA operates on a reserved address space, and the corresponding enable bit for that DMA channel is also cleared by hardware to stop the erroneous transfer on that channel. At this time, the channel's transfer error interrupt flag (TEIF) in the DMA_IFT register is set. To generate an interrupt, the corresponding transfer error interrupt enable bit in the DMA_CCRx register should be configured.

8.7 DMA register description

Table 8-4 Overview of DMA registers

Offset	Acronym	Register Name	Reset
0x00	DMA_ISR	DMA interrupt status register	0x00000000
0x04	DMA_IFCR	DMA interrupt flag clear register	0x00000000
0x08+20×(n-1)	DMA_CCRx	DMA channel x configuration register	0x00000000
0x0C+20×(n-1)	DMA_CNDTRx	DMA channel x number of data register	0x00000000
0x10+20×(n - 1)	DMA_CPARx	DMA channel x peripheral address register	0x00000000
0x14+20×(n - 1)	DMA_CMARx	DMA channel x memory address register	0x00000000

8.7.1 DMA_ISR DMA Interrupt Status Register

Address offset: 0x00
Reset value: 0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.								TEIF2	HTIF2	TCIF2	GIF2	TEIF1	HTIF1	TCIF1	GIF1
								r	r	r	r	r	r	r	r

Bit	Field	Description
31:8	Reserved	Reserved and always read as 0
7,3	TEIFx	Channel x transfer error flag (x=1~2) This bit is read-only by software. It is set or cleared by hardware. This flag bit is cleared by writing 1 to the corresponding bit in the DMA_IFCR register. 0: No DMA transfer error (TE) on channel x 1: A transfer error (TE) occurred on channel x due to the DMA access to the reserved address
6,2	HTIFx	Channel x half transfer flag (x=1~2) This bit is read-only by software. It is set or cleared by hardware. This flag bit is cleared by writing 1 to the corresponding bit in the DMA_IFCR register. 0: No DMA half transfer (HT) event on channel x 1: A DMA half transfer (HT) event occurred on channel x
5,1	TCIFx	Channel x transfers complete flag (x=1~2) This bit is read-only by software. It is set or cleared by hardware. This flag bit is cleared by writing 1 to the corresponding bit in the DMA_IFCR register. 0: No DMA transfer complete (TC) event on channel x 1: A DMA transfer complete (TC) event occurred on channel x
4,0	GIFx	Channel x global interrupt flag (x=1~2) This bit is read-only by software. It is set or cleared by hardware. This flag bit is cleared by writing 1 to the corresponding bit in the DMA_IFCR register. 0: No TE, HT or TC event on channel x 1: A TE, HT or TC event occurred on channel x

8.7.2 DMA_IFCR DMA Interrupt Flag Clear Register

Address offset: 0x04
Reset value: 0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								CTEIF2	CHTIF2	CTCIF2	CGIF2	CTEIF1	CHTIF1	CTCIF1	CGIF1

	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c

Bit	Field	Description
31:8	Reserved	Reserved and always read as 0
7,3	CTEIFx	Channel x transfer error clear (x=1~2) This bit is set and cleared by software. 0: No effect 1: Clears the corresponding TEIF flag in the DMA_ISR register
6,2	CHTIFx	Channel x half transfer clear (x=1~2) This bit is set and cleared by software. 0: No effect 1: Clears the corresponding HTIF flag in the DMA_ISR register
5,1	CTCIFx	Channel x transfers completely clear (x=1~2) This bit is set and cleared by software. 0: No effect 1: Clears the corresponding TCIF flag in the DMA_ISR register
4,0	CGIFx	Channel x global interrupt clear (x=1~2) This bit is set and cleared by software. 0: No effect 1: Clears the GIF, TEIF, HTIF and TCIF flags in the DMA_ISR register

8.7.3 DMA_CCRx DMA Channel x Configuration Register (x=1~2)

Address offset: 0x08 + 20x(channel number–1)
Reset value: 0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ARE	MEM2MEM	PL		MSIZE		PSIZE		MINC	PINC	CIRC	DIR	TEIE	HTIE	TCIE	EN
rw	rw	rw		rw		rw		rw	rw	rw	rw	rw	rw	rw	rw

Bit	Field	Description
31:16	Reserved	Reserved and always read as 0
15	ARE	Auto reload This bit is set and cleared by software. 1: Auto reload number of data，peripheral address，memory address to transfer enabled 0: Auto reload number of data，peripheral address，memory address to transfer disabled Note: When ARE is configured to 1, automatic reload is triggered after the closing of the DMA channel enable
14	MEM2MEM	Memory to memory mode This bit is set and cleared by software. 1: Memory to memory mode enabled 0: Memory to memory mode disabled
13:12	PL	Channel priority level This bit is set and cleared by software. 00: Low 01: Medium 10: High 11: Very high
11:10	MSIZE	Memory size This bit is set and cleared by software. 00: 8 bits 01: 16 bits 10: 32 bits 11: Reserved, undefined

Bit	Field	Description
9:8	PSIZE	Peripheral size This bit is set and cleared by software. 00: 8 bits 01: 16 bits 10: 32 bits 11: Reserved, undefined
7	MINC	Memory increment mode This bit is set and cleared by software. 0: Memory increment mode disabled 1: Memory increment mode enabled
6	PINC	Peripheral increment mode This bit is set and cleared by software. 0: Peripheral increment mode disabled 1: Peripheral increment mode enabled
5	CIRC	Circular mode This bit is set and cleared by software. 0: Circular mode disabled 1: Circular mode enabled
4	DIR	Data transfer direction This bit is set and cleared by software. 0: Read from peripheral 1: Read from memory
3	TEIE	Transfer error interrupt enable This bit is set and cleared by software. 0: TE interrupt disabled 1: TE interrupt enabled
2	HTIE	Half transfer interrupt enable This bit is set and cleared by software. 0: HT interrupt disabled 1: HT interrupt enabled
1	TCIE	Transfer complete interrupt enable This bit is set and cleared by software. 0: TC interrupt disabled 1: TC interrupt enabled
0	EN	Channel enable This bit is set and cleared by software. 0: Channel disabled 1: Channel enabled Note: EN bit clearing does not stop ongoing DMA transfers on this channel. To stop the transfers, follow the specific software operation procedure.

8.7.4 DMA_CNDTRx DMA Channel x Number of Data Register (x=1~2)

Address offset: 0x0C + 20x(channel number–1)
Reset value: 0x0000 0000

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

Bit	Field	Description
31:16	Reserved	Reserved and always read as 0

Bit	Field	Description
15:0	NDT	Number of data to transfer Number of data to be transferred (0 up to 65535). This register can only be written when the channel is disabled (EN bit in the DMA_CCRx is 0). Once the channel is enabled, this register is read-only by software, indicating the number of DMA transfers that have still to be performed. This register decrements after each DMA transfer. When the value decrements down to 0, all the data is transmitted. At this time, the register will be reloaded automatically by the value previously programmed if the channel is configured in Auto Reload mode. If this register is zero, no DMA transaction can be served whether the channel is enabled or not.

8.7.5 DMA_CPARx DMA Channel x Peripheral Address Register (x=1~2)

Address offset: 0x10 + 20x(channel number–1)
Reset value: 0x0000 0000
This register must not be written when the channel is enabled (EN bit in the DMA_CCRx is 1).

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

Bit	Field	Description
31:0	PA	Peripheral address Base address of the peripheral data register from/to which the data will be read/written. When PSIZE is ‘01’ (16-bit), the basic unit of address is 0x2. The least significant bit PA[0] is ignored. Access is automatically aligned to a half word address. When PSIZE is ‘10’ (32-bit), the basic unit of address is 0x4. The last two bits PA[1:0] are ignored. Access is automatically aligned to a word address.

8.7.6 DMA_CMARx DMA Channel x Memory Address Register (x=1~2)

Address offset: 0x14 + 20x(channel number–1)
Reset value: 0x0000 0000
This register must not be written when the channel is enabled (EN bit in the DMA_CCRx is 1).

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

Bit	Field	Description
31:0	MA	Memory address Base address of the memory area from/to which the data will be read/written. When MSIZE is ‘01’ (16-bit), the basic unit of address is 0x2. The least significant bit MA[0] is ignored. Access is automatically aligned to a half word address. When MSIZE is ‘10’ (32-bit), the basic unit of address is 0x4. The last two bits MA[1:0] are ignored. Access is automatically aligned to a word address.

9 GPIO General-Purpose I/Os

9.1 Overview

Each general-purpose IO (GPIO) port can be individually configured by two 32-bit control registers (GPIOx_CRL/GPIOx_CRH) and two 32-bit multiplexed control registers (GPIOx_AFRL, GPIOx_AFRH) in eight modes: analog input, input floating, input pull-up, input pull-down, output push-pull, output open-drain, alternate function push-pull and alternate function open-drain.

Each I/O port bit is freely programmable. All registers can be accessed as 32-bit (words), 16-bit (half-words) or 8-bit (bytes). The GPIOx_BSRR and GPIOx_BRR control registers in the GPIO register group are able to modify the GPIOx_ODR bit to output 0 or 1 independently through write accesses.

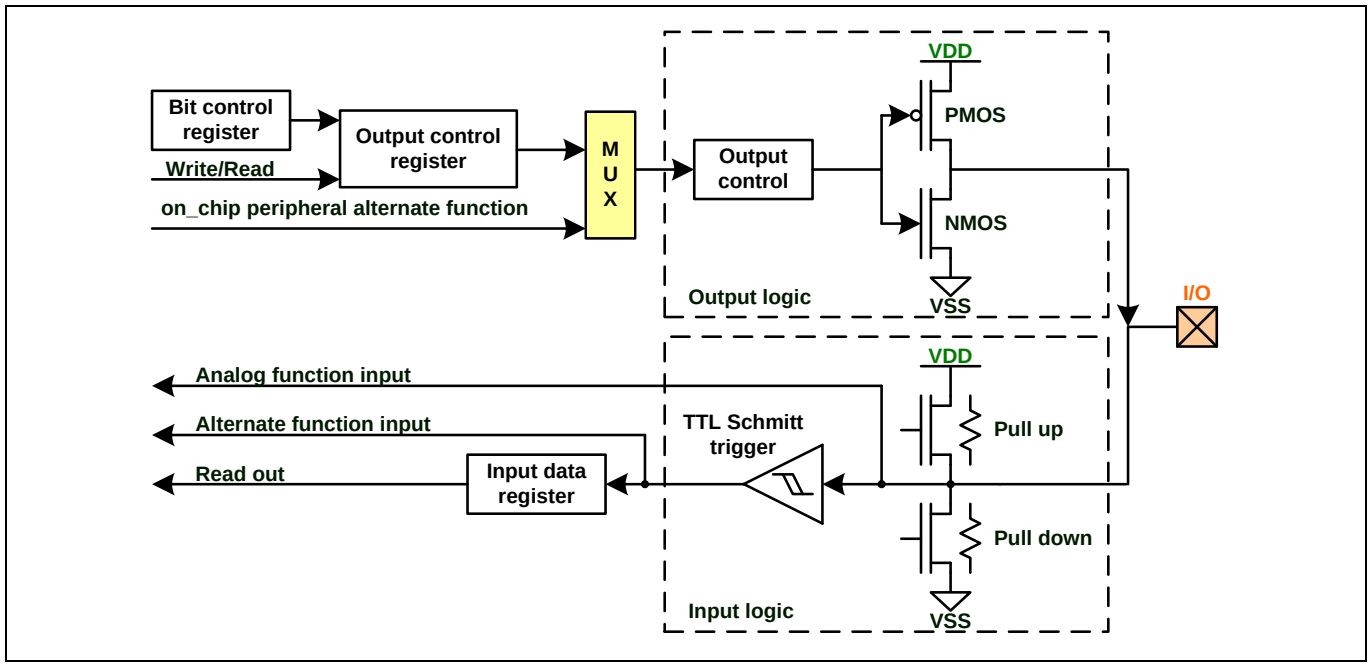
9.2 Main characteristics

- A single AHB write access changes one or multiple bits for GPIOx_ODR
- All I/Os support programmable EXTI configuration register to output externally triggered interrupts
- Support GPIO locking mechanism
- Supported input states: floating, pull-up/down, analog
- Supported output states: push-pull or open drain with pull-up/down
- Input floating as default, configurable input/output direction

9.3 Functional description

9.3.1 Functional block diagram

Figure 9-1 Standard I/O port



9.3.2 GPIO port configuration

Table 9-1 Port bit configuration table（take port0 as an example）

Pin mode		Pull up/down	DCR[1:0]		CNF0		MODE0	ODRx
Analog input		x	x	x	0	0	00	x
General input Alternate input		Floating	x	x	0	1		x
		Pull-up	x	x	1	0		1
		Pull-down	x	x	1	0		0
General output	Push-pull	x	x	x	0	0	01 10 11	0 or 1
	Open-drain	Floating	x	0	0	1		0 or 1
		Pull-up	1	1	0	1		0 or 1
		Pull-down	0	1	0	1		0 or 1
Alternate output	Push-pull	x	x	x	1	0		x

Note: x represents “don’t care” for I/Os in the corresponding mode. ODR0 represents bit 0 of the output data register.

Input and output follow these configurations:

- General-purpose input:

  The user should configure the CNF0 in the GPIOx_CRL to choose the Input mode.

- General-purpose output:

  Push-Pull output: The user configures MODE0 and sets CNF0=00;

  Open-Drain output: The user configures MODE0 and sets CNF0=01. To activate the pull-up/pull-down feature for pins, the GPIOx_DCR register should be configured individually. This feature is invalid if it is not an open-drain mode output.

- Alternate function:

  Configure the AFRLx[3:0] and AFRHx[3:0] registers to choose the alternate function:
  Alternate push-pull output: The user configures MODE0 and sets CNF0=10;
  Alternate open-drain output: The user configures MODE0 and sets CNF0=11.
  To activate the pull-up/pull-down feature for IO in the output mode, the GPIOx_DCR register should be configured individually. This feature is invalid if it is not an open-drain mode output.

During and just after reset, the GPIO ports are configured in Input Floating mode. The Serial-Wired Debug pins default to input pull-up (PU)/pull-down (PD).

When configured in general purpose output mode, the value of the output data register (GPIOx_ODR) is output on the corresponding I/O pin. The Input Data register (GPIOX_IDR) captures the data present on the I/O pin at every AHB clock cycle.

Note: Not all chips contain the JTAG and SWD ports. For specific chip configuration, refer to the chip datasheet.

- PA14: SWCLK in PD
- PA13: SWDIO in PU

9.3.3 Alternate function

The corresponding IO alternate functions are enabled by setting the alternate function register.

- When an IO is configured as Alternate Function Input, the port should choose pull-up, pull-down or floating input.
- When an IO is configured as Alternate Function Output, the port should choose output push-pull or open-drain mode.
- When an IO is configured with bidirectional alternate function, the port should choose output push-pull

or open-drain mode. In this case the input is configured in input floating mode. In the open-drain mode, the GPIOx_DCR register is configured to choose weak pull-up or weak pull-down resistor.

If a port bit is configured as Alternate Function Output, this connects the port to the output signal of an on-chip peripheral. If a GPIO pin is configured as Alternate Function Output solely via software, but peripheral is not activated, its output is not specified.

9.3.4 GPIO locking mechanism

It is possible to freeze the IO configuration by applying the GPIO locking mechanism. When the LOCK mechanism has been applied on a port, it is no longer possible to modify the port configuration until the next reset. The write sequence of the LOCK key:

- GPIOx_LCKR[16]='1'+LCKR[15:0].
- GPIOx_LCKR[16]='0'+LCKR[15:0].
- GPIOx_LCKR[16]='1'+LCKR[15:0].

To lock the PA[0] port of GPIOA, follow these configurations:

- GPIOA->GPIOA_LCKR=0x10001.
- GPIOA->GPIOA_LCKR=0x00001.
- GPIOA->GPIOA_LCKR=0x10001.

After three steps above are finished, the bit 16 of the GPIOA_LCKR register is set. The GPIOA_LCKR register can no longer be written until the next soft reset. The bit 16 of the GPIOA_LCKR register remains 1 and PA[0] keeps its configuration before the locking.

When the port is locked, the value of the port bit can no longer be modified until the soft reset. Each lock bit in the GPIOx_LCKR register freezes 4 bits in the port configuration registers (GPIOx_CRL and GPIOx_CRH).

Notes:

Only the value of PA[0] is locked due to the proceeding configuration. It is still possible to configure the values of PA[15:1] and other GPIO control registers.

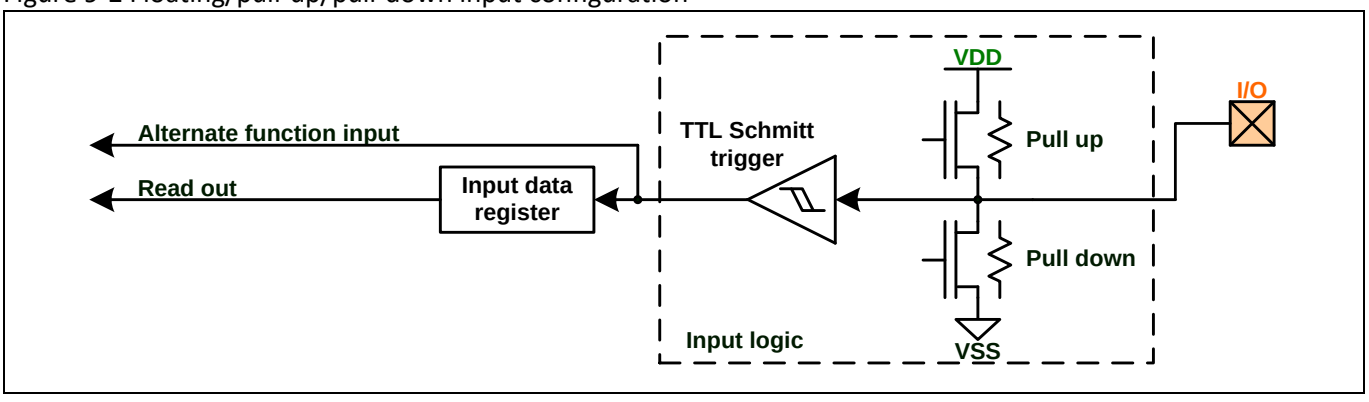
9.3.5 Input configuration

When the I/O port is programmed as Input:

- The Schmitt Trigger Input is activated.
- The Output Buffer is disabled.
- Floating, pull-up or pull-down input mode can be used.
- The data present on the I/O pin is sampled into the Input Data register every AHB clock cycle.
- A read access to the Input Data register gets the I/O state.

The figure below shows the input configuration of the I/O port:

Figure 9-2 Floating/pull-up/pull-down input configuration



To configure the PA[0] as pull-up input in the GPIOA, refer to the following:

- GPIOA->GPIOA_ODR=0x0001.
- GPIOA->GPIOA_CRL=0x00000008.

To configure the PA[0] as pull-down input in the GPIOA, refer to the following:

- GPIOA->GPIOA_ODR=0x0000.
- GPIOA->GPIOA_CRL=0x00000008.

Notes:

To configure the port as pull-up input, the corresponding bit in the GPIO_ODR register should output 1 first.

To configure the port as pull-down input, the corresponding bit in the GPIO_ODR register should output 0 first.

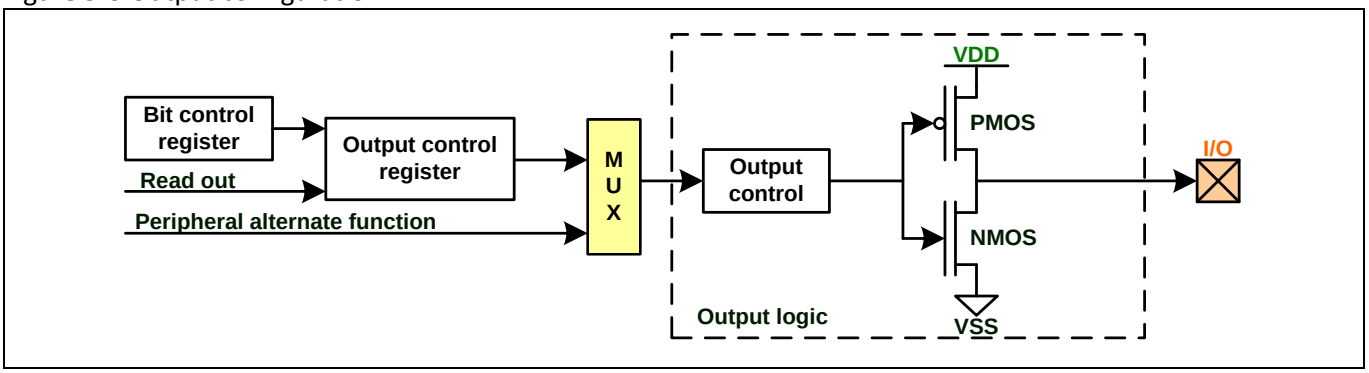
9.3.6 Output configuration

When the GPIO pin is configured as output:

- The Schmitt Trigger Input is activated.
- The Output Buffer is enabled.
- In general purpose output mode, the weak pull-up and pull-down resistors are disabled.
- Open Drain Mode: When the Port Output Data Register is set to 0, the corresponding pin outputs low level; when the Port Output Data Register is set to 1, the corresponding pin is in Hi-Z.
- Push-Pull Mode: When the output register is set to 0, the corresponding pin outputs low level; when the output register is set to 1, the corresponding pin outputs high level.
- A read access to the Port Output Data register gets the last written value.
- A read access to the Port Input Data register gets the current I/O state.

The figure below shows the output configuration of the I/O port:

Figure 9-3 Output configuration



9.3.7 Alternate function configuration

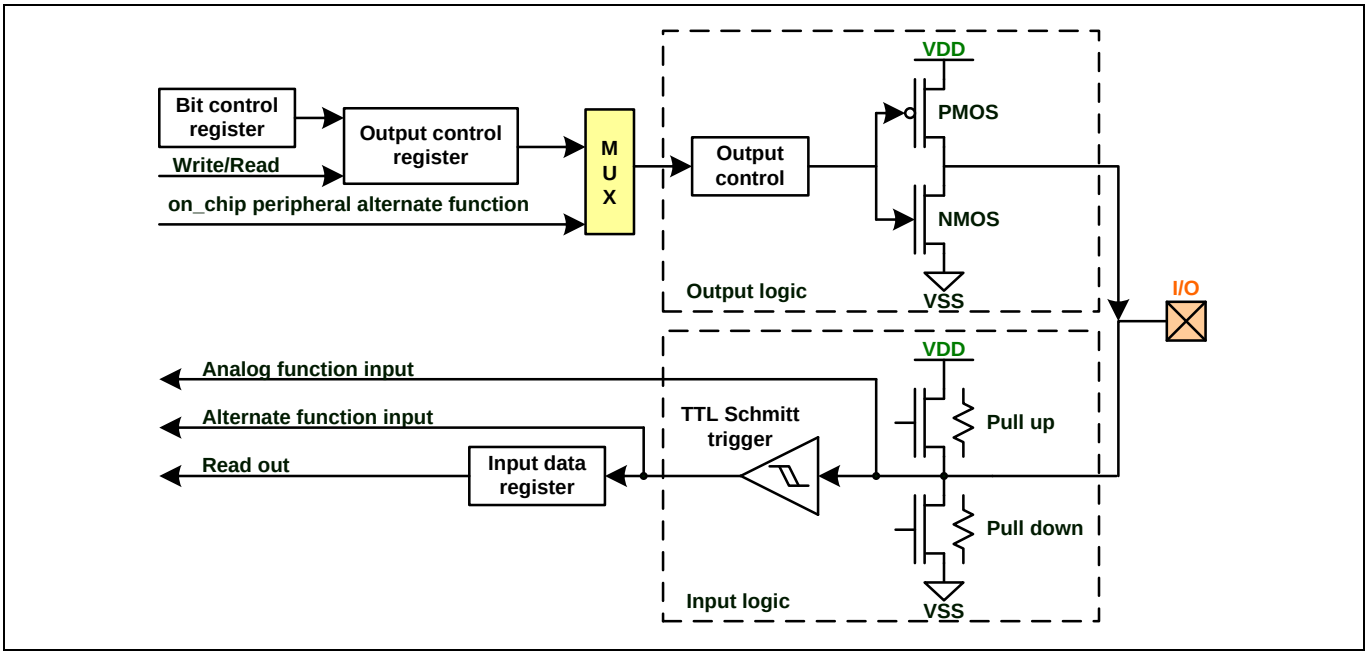
When the pin is configured as an alternate function:

- The Schmitt Trigger Input is activated.
- The Output Buffer can be configured as Open Drain or Push-Pull.
- In the output open-drain mode, the GPIOx_DCR register is configured to choose weak pull-up or weak pull-down resistor.

- The weak pull-up or weak pull-down resistor can be chosen when the pin is configured as Input.
- The data present on the I/O pin is sampled into the Input Data register every AHB clock cycle.

The figure below shows the alternate function configuration of the I/O port. Refer to AFRL and AFRH registers as well as the datasheet for further information.

Figure 9-4 Alternate function configuration



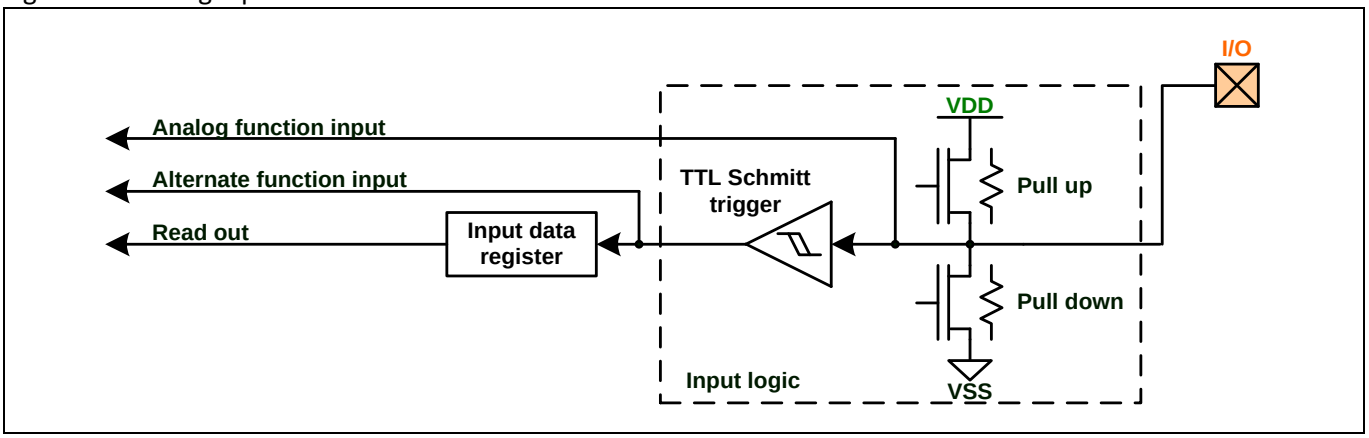
9.3.8 Analog input configuration

When the I/O port is configured as analog input configuration:

- The Output Buffer is disabled.
- The Schmitt Trigger Input is de-activated.
- The weak pull-up and pull-down resistors are disabled.
- The Port Input Data Register remains 0.

The figure below shows the analog input configuration of the I/O port:

Figure 9-5 Analog input



9.3.9 External clock used as alternate GPIO port

The external HSE/LSE clock can be used as an alternate GPIO. When the corresponding clock PAD is used as the GPIO, the user should turn off the external clock first and then set the GPIO function as normal. For specific mapping, please refer to the chip datasheet.

9.3.10 SWD alternate function remapping

The SWD interface signals are mapped on the GPIO ports as shown in the table below:

Table 9-2 SWD alternate function remapping

Alternate function	GPIO port
SWDIO	PA13
SWCLK	PA14

9.4 Register

9.4.1 Overview of registers

Table 9-3 Overview of GPIO registers

Offset	Acronym	Register Name	Reset
0x00	GPIOx_CRL	Port configuration register low	0x44444444
0x04	GPIOx_CRH	Port configuration register high	0x44444444
0x08	GPIOx_IDR	Port input data register	0x0000XXXX
0x0C	GPIOx_ODR	Port output data register	0x00000000
0x10	GPIOx_BSRR	Port bit set/reset register	0x00000000
0x14	GPIOx_BRR	Port bit reset register	0x00000000
0x18	GPIOx_LCKR	Port configuration lock register	0x00000000
0x1C	GPIOx_DCR	Port output open drain control register	0x00000000
0x20	GPIOx_AFRL	Port alternate function register low	0x00000000
0x24	GPIOx_AFRH	Port alternate function register high	0x00000000

Note: The possible range of "X" in GPIOx is A to I, but not all chips include all GPIOA to GPIOI groups. The specific configuration of each chip can be referred to the data manual of each chip.

9.4.2 GPIOx_CRL Port Configuration Register Low

Address offset: 0x00
Reset value: 0x4444 4444

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CNF7		MODE7		CNF6		MODE6		CNF5		MODE5		CNF4		MODE4	
rw		rw		rw		rw		rw		rw		rw		rw	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNF3		MODE3		CNF2		MODE2		CNF1		MODE1		CNF0		MODE0	
rw		rw		rw		rw		rw		rw		rw		rw	

Bit	Field	Description
31:30	CNF7	Port configuration bits (y=7..0)
27:26	CNF6	

Bit	Field	Description
23:22	CNF5	Set MODEy to be 0 and the port as input mode while configuring the CNFy bit to choose the input mode： 00: Analog input mode 01: Input floating mode 10: Input pull-up/pull-down mode 11: Reserved Set MODEy not to be 0 and the port as output mode while configuring the CNFy bit to choose the output mode： 00: General purpose output push-pull 01: General purpose output open-drain 10: Alternate function output push-pull 11: Alternate function output open-drain Port input/output configuration (MODEy)(y = 0..7) The corresponding I/O port is configured via software; refer to Port bit configuration table. Set MODEy is not equal to 0, different configurations result in same rates： 00: input mode。
19:18	CNF4	
15:14	CNF3	
11:10	CNF2	
7:6	CNF1	
3:2	CNF0	
29:28	MODE7	
25:24	MODE6	
21:20	MODE5	
17:16	MODE4	
13:12	MODE3	
9:8	MODE2	
5:4	MODE1	
1:0	MODE0	

9.4.3 GPIOx_CRH Port Configuration Register High

Address offset: 0x04
Reset value: 0x4444 4444

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CNF15		MODE15		CNF14		MODE14		CNF13		MODE13		CNF12		MODE12	
rw		rw		rw		rw		rw		rw		rw		rw	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNF11		MODE11		CNF10		MODE10		CNF9		MODE9		CNF8		MODE8	
rw		rw		rw		rw		rw		rw		rw		rw	

Bit	Field	Description
31:30	CNF15	Port configuration bits (y=15..8) Set MODEy to be 0 and the port as input mode while configuring the CNFy bit to choose the input mode： 00: Analog input mode 01: Input floating mode 10: Input pull-up/pull-down mode 11: Reserved Set MODEy not to be 0 and the port as output mode while configuring the CNFy bit to choose the output mode： 00: General purpose output push-pull 01: General purpose output open-drain 10: Alternate function output push-pull 11: Alternate function output open-drain Port input/output configuration (MODEy)(y = 15..8) The corresponding I/O port is configured via software; refer to Port bit configuration table. Set MODEy is not equal to 0, different configurations result in same rates： 00: input mode。
27:26	CNF14	
23:22	CNF13	
19:18	CNF12	
15:14	CNF11	
11:10	CNF10	
7:6	CNF9	
3:2	CNF8	
29:28	MODE15	
25:24	MODE14	
21:20	MODE13	
17:16	MODE12	
13:12	MODE11	
9:8	MODE10	
5:4	MODE9	
1:0	MODE8	

9.4.4 GPIOx_IDR Port Input Data Register

Address offset: 0x08
Reset value: 0x0000 XXXX

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

Res															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IDRy(y=15~0)															
r															

Bit	Field	Description
31:16	Reserved	Always read as 0
15:0	IDRy	Port input data (y=15..0) The reads represent the corresponding I/O states.

9.4.5 GPIOx_ODR Port Output Data Register

Address offset: 0xC
Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ODRy(y=15~0)															
rw															

Bit	Field	Description
31:16	Reserved	Always read as 0
15:0	ODRy	Port output data (y=15..0) When configured in general purpose output mode, the written value is output to the corresponding I/O. Note: The ODR bits can be individually set and cleared by setting the GPIOx_BSRR (x=A..H) Register.

9.4.6 GPIOx_BSRR Port Bit Set/Reset Register

Address offset: 0x10
Reset value: 0x0000 000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BRy(y=15~0)															
w															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BSy(y=15~0)															
w															

Bit	Field	Description
31:16	BRy	Port Reset bit y (y=15..0) Writing it to 0 will make the corresponding ODRY bit unchanged Writing it to 1 will clear the corresponding ODRY bit
15:0	BSy	Port Set bit y (y=15..0) Writing it to 0 will make the corresponding ODRY bit unchanged Writing it to 1 will set the corresponding ODRY bit to 1 Note: When both BSy and BRy bits are written to 1, BSy has priority over BRy.

9.4.7 GPIOx_BRR Port Bit Reset Register

Address offset: 0x14
Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BRy(y=15~0)															
w															

Bit	Field	Description
31:16	Reserved	Always read as 0
15:0	BRy	Port Reset bit y (y=15..0) Writing it to 0 will make the corresponding ODRY bit unchanged Writing it to 1 will clear the corresponding ODRY bit

9.4.8 GPIOx_LCKR Port Configuration Lock Register

Address offset: 0x18
Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res															LCKK
															rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LCKy(y=15~0)															
w															

Bit	Field	Description
31:17	Reserved	Always read as 0
16	LCKK	Lock key This bit can be read anytime. It can only be modified using the Lock Key Writing Sequence. 0: port configuration lock key not active 1: port configuration lock key active. GPIOx_LCKR register is locked until the next soft reset. LOCK key sequence: write 1 -> write 0 -> write 1
15:0	LCKy	Port x Lock bit y (y = 15..0) These bits are read write but can only be written when the LCKK bit is 0. 0: port configuration not locked 1: port configuration locked

9.4.9 GPIOx_DCR Port Output Open Drain Control Register

Address offset: 0x1C
Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PX15		PX14		PX13		PX12		PX11		PX10		PX9		PX8	
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PX7		PX6		PX5		PX4		PX3		PX2		PX1		PX0	
rw															

Bit	Field	Description
31:2	PX15-PX1	See PX0
1:0	PX0	PX0[1:0]: 11: output open-drain mode with port pull-up 01: output open-drain mode with port pull-down x0: output open-drain mode without port pull-up/pull-down

9.4.10 GPIOx_AFRL Port Alternate Function Register Low

Address offset: 0x20
Reset value: GPIOx_AFRL（x = A..B）：0xFFFF FFFF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
AFR7				AFR6				AFR5				AFR4			
rw				rw				rw				rw			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AFR3				AFR2				AFR1				AFR0			
rw				rw				rw				rw			

Bit	Field	Description
31:0	AFRy	Port x alternate function bit y (y = 0..7), which can be accessed by software to configure the function. 0000: AF0 0001: AF1 0010: AF2 0011: AF3 0100: AF4 0101: AF5 0110: AF6 0111: AF7 1000: AF8 1001: AF9 1010: AF10 1011: AF11 1100: AF12 1101: AF13 1110: AF14 1111: AF15

9.4.11 GPIOx_AFRH Port Alternate Function Register High

Address offset: 0x24
Reset value: GPIOB_AFRH：0x0000 00FF，GPIOA_AFRH：0xF00F FFFF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
AFR15				AFR14				AFR13				AFR12			
rw				rw				rw				rw			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AFR11				AFR10				AFR9				AFR8			
rw				rw				rw				rw			

Bit	Field	Description
31:0	AFRy	Port x alternate function bit y (y = 8..15), which can be accessed by software to configure the function. 0000: AF0 0001: AF1 0010: AF2 0011: AF3 0100: AF4 0101: AF5 0110: AF6 0111: AF7 1000: AF8 1001: AF9 1010: AF10 1011: AF11 1100: AF12 1101: AF13 1110: AF14 1111: AF15

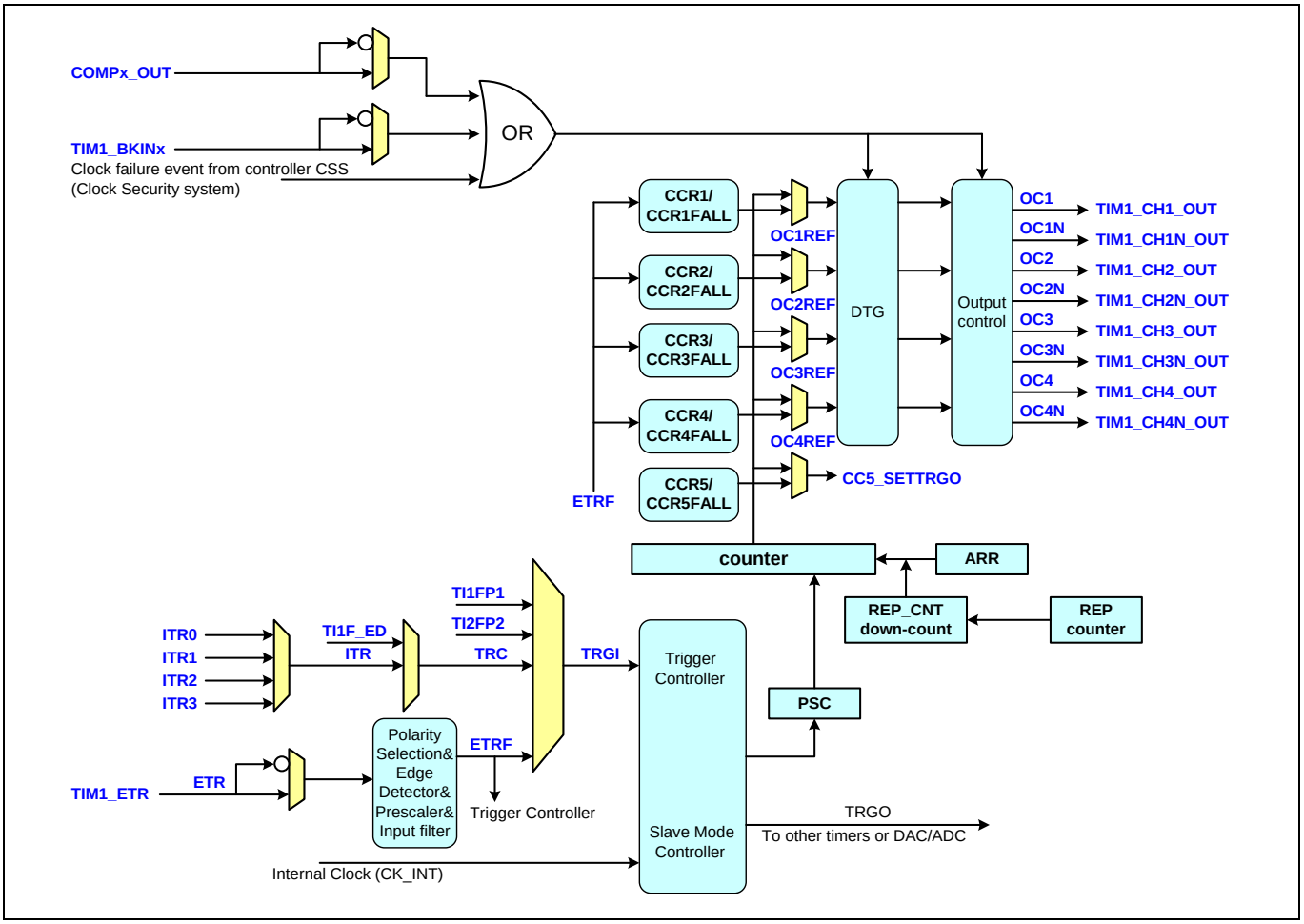
10 TIM1 Advanced Timer

10.1 Overview

TIM1 is composed of a 16-bit real-time programmable prescaler and a 16-bit auto reload counter with the programmable counting direction, and can conveniently provide with the counting and timing function. The counter is driven by a programmable prescaler. The advanced timer can be used for a variety of purposes, such as PWM output, programmable dead-time inserted between the complementary outputs, one-pulse mode, etc.

10.2 Function block diagram

Figure 10-1 TIM1 block diagram



TIM1 structure block diagram is mainly composed of output compare unit, counter unit and break unit.

10.3 Main characteristics

- 16-bit real-time programmable prescaler, division factor: 1-65536.
- Clock source: Internal clock source, external clock input (ETRx), internal trigger input (ITRx).
- 16 bit auto reload counter (counting direction: up, down, up/down).
- 8-bit programmable repeat counting function, repeat counter can be auto reloaded (auto reload at update event)
- Synchronization circuit to control the timer with external signals and to interconnect several timers to each other.
- Trigger input can serve as external clock or be managed cycle-by-cycle.
- 4 complementary output channels.
- Output compare (control output waveform or indicator counter already finishes timing).
- PWM output (dead-time programmable) (edge alignment or central alignment mode)
- Break input will put the timer output signal in a safe state (reset state or preset state, choice by user).
- One-pulse mode.
- Interrupt/DMA request event: update event, trigger event, output compare or break input.

10.4 Function description

10.4.1 Clock

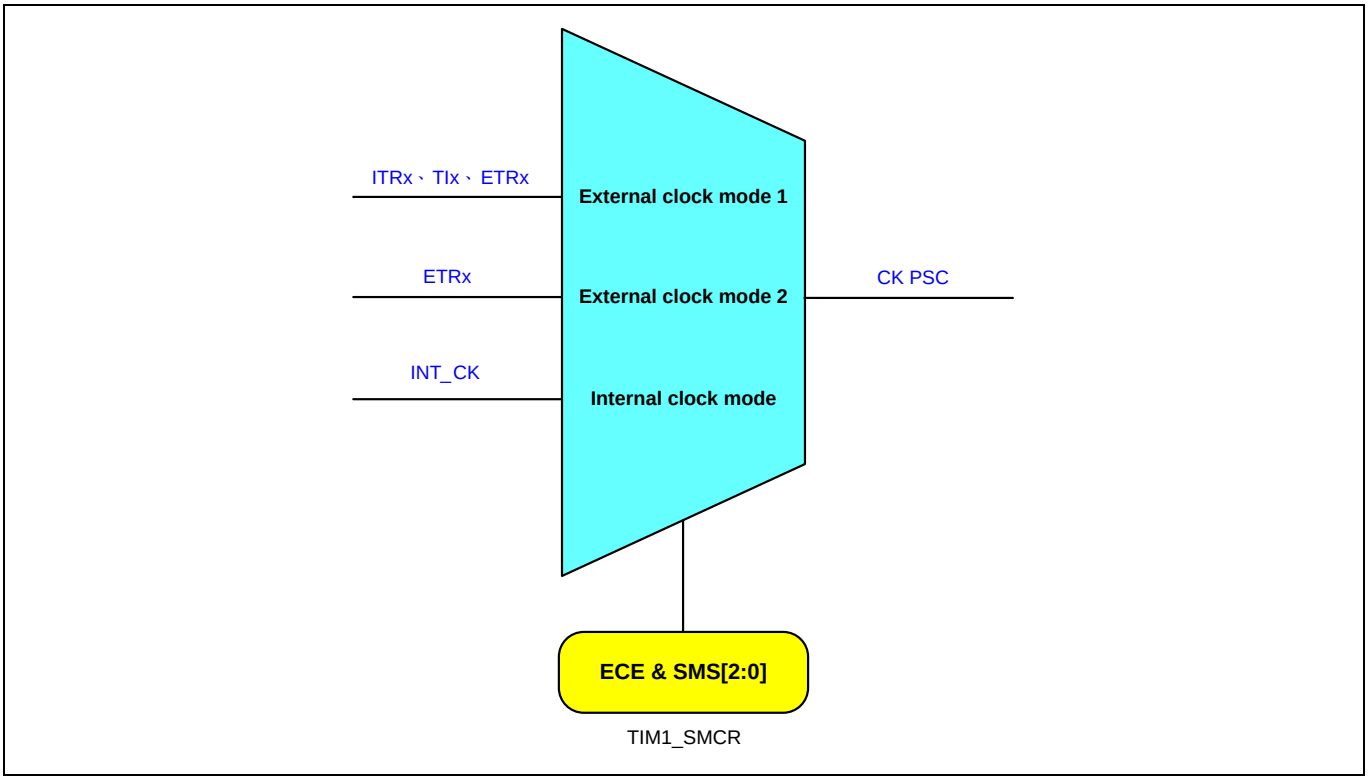
10.4.1.1 Clock selection

Several types of clock sources of the counter below:

- Internal clock (INT_CK)
- External clock mode 1: External trigger input TRGI (including ITRx, ETRx)
- External clock mode 2: External trigger input ETR (including ETRx)

The diagram of several types of clock choice above:

Figure 10-2 Clock selection



10.4.1.1.1 Internal clock source (INT _CK)

In case of SMS =000 of TIM1_SMCR register, disable the slave mode. Enable the counter, and the prescaler clock is driven by internal clock.

10.4.1.1.2 External clock mode 1(external trigger input TRGI, including ITRx, ETRx)

In case of SMS = 111 of TIM1_SMCR register, select the external clock mode 1 (TRGI). The counter is driven by the rising edge or falling edge of the selected input signal.

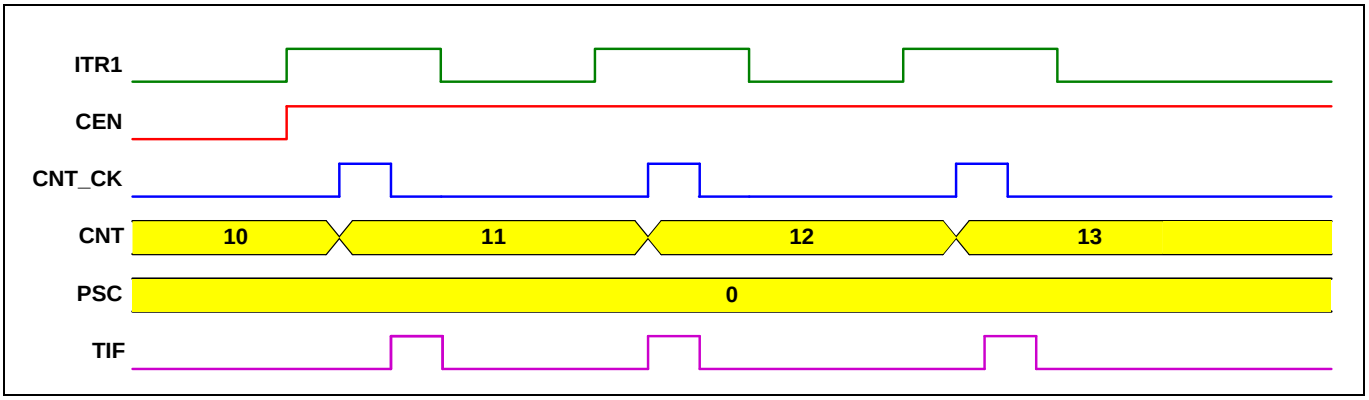
For example: count up at the rising edge on ITR1 input. Specific configuration:

1. Configure TIM1_SMCR register TS=001, select ITR1 as the trigger input source; configure TIM1_SMCR register SMS=111, select the external clock mode 1.

2. Configure TIM1_CR1 register DIR=0, select up counting mode; configure TIM1_CR1 register CEN=1 and start the counter.

ITR1 input the active edge, count up once and TIF flag bit is set by hardware. The delay between the active edge of ITR1 and actual clock of counter depends on the IRT1 input pin sync circuit.

Figure 10-3 Control circuit in external clock mode 1



Note：TIM1_PSC and TIM1_RCR must be kept at reset value when select the external clock mode.

10.4.1.1.3 External clock mode 2(external trigger input ETR, including ETRx)

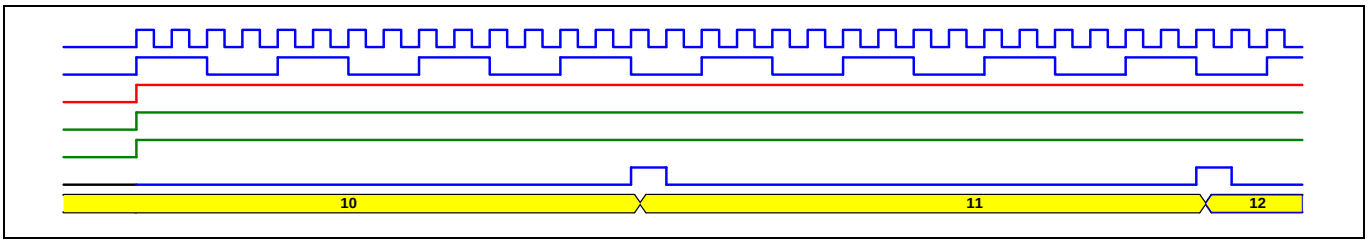
In case of TIM1_SMCR register ECE=1, enable external clock mode 2, and the counter is driven by the active edge of ETR signal.

For example: count up once of each 4 falling edge on ETR Specific configuration:

1. Configure TIM1_SMCR register ETF[3:0] =0010, count up once of each 4 active edges on ETR; configure TIM1_SMCR register ETP=1, select falling edge as active edge; configure TIM1_SMCR register ECE=1, select external clock mode 2.
2. Configure TIM1_CR1 register DIR=0, select up counting mode; Configure TIM1_CR1 register CEN=1, start the counter.

The delay between the falling edge of ETR and actual clock of counter depends on the sync circuit of ETR signal terminal.

Figure 10-4 Control circuit in external clock mode 2



Note：TIM1_PSC and TIM1_RCR must be kept at reset value when select the external clock mode.

10.4.1.2 Time base unit

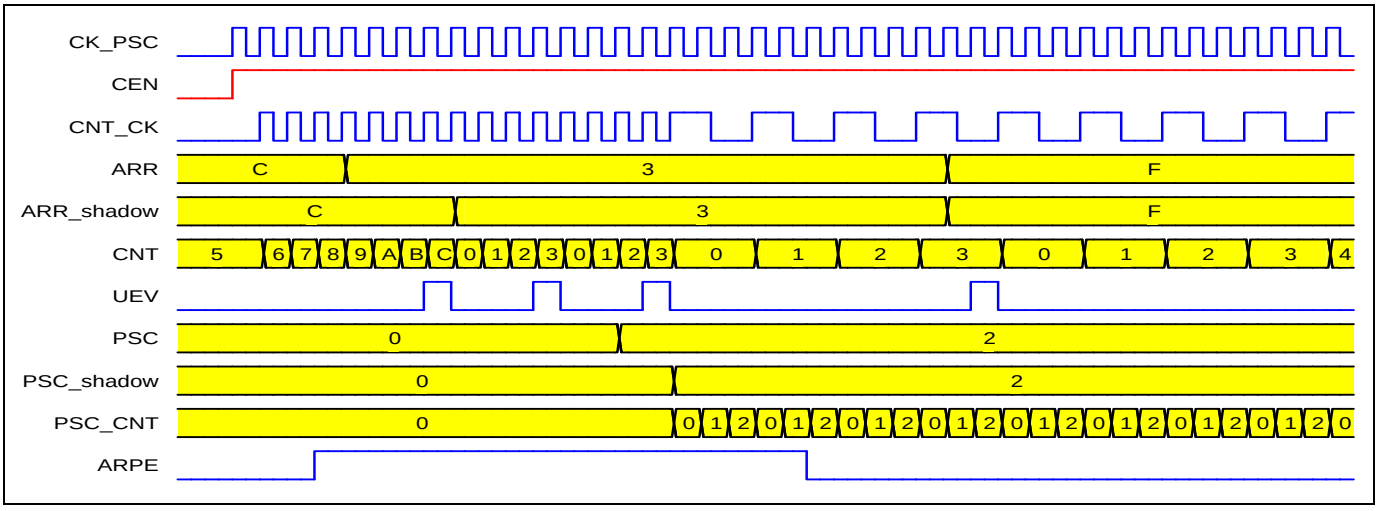
TIM1 time base unit mainly includes: Counter register (TIM1_CNT), prescaler register (TIM1_PSC), auto reload register (TIM1_ARR) and repetition counter register (TIM1_RCR).

The counter unit is composed of a 16-bit counter with its related auto reload register, The counter can count up, down or both up and down.

The counter clock is provided by prescaler. The prescaler is composed of the prescaler counter with its related register. The division factor is 1-65536. It can be written at any time, and gets active at the next update.

The auto preload register has a 16-bit shadow register with preload function. Set TIM1_CR1 register ARPE bit to select the content of ARR register are transferred into the shadow register permanently or at each update event.

Figure 10-5 Auto preload



10.4.1.3 Counter modes

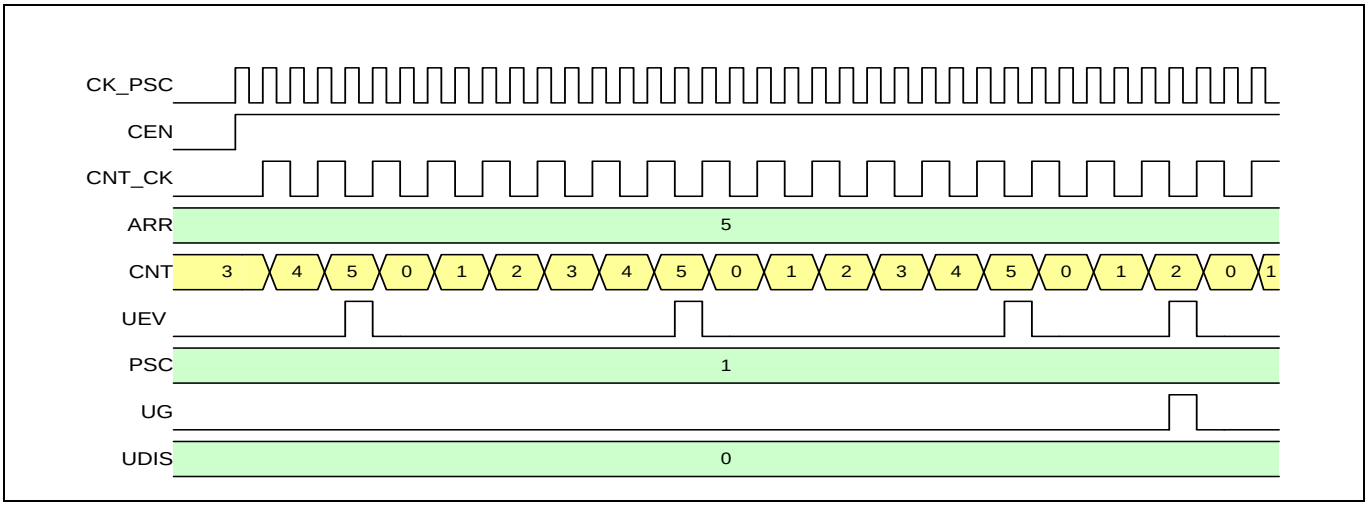
Configure DIR bit and CMS bit of TIM1_CR1 register to select the count mode. There are three count modes, up counting mode, down counting mode and central alignment mode (up/ down counting mode). Detailed descriptions are provided for each count mode.

10.4.1.3.1 Up counting mode

Configure TIM1_CR1 register CMS=0, DIR＝0, select up counting mode.

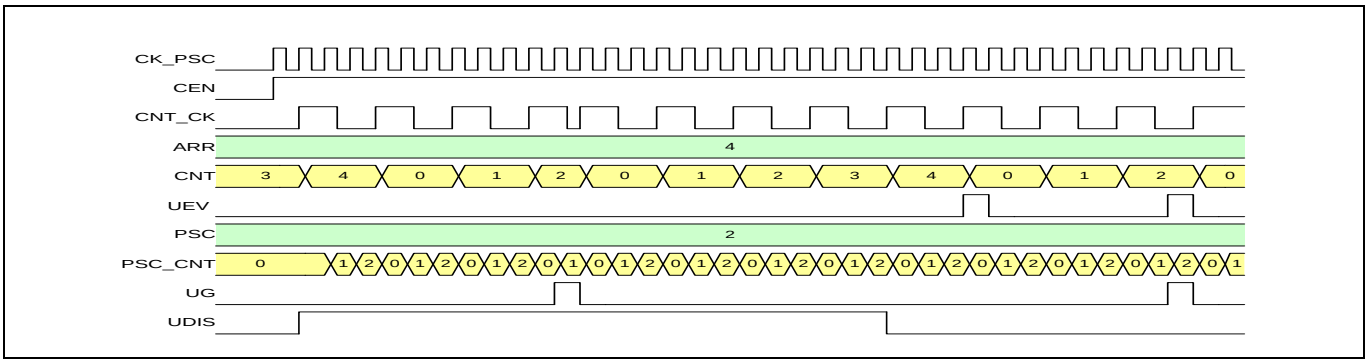
In up counting mode, enable TIM1_CR1 register CEN, and the counter begins up count from 0 to TIM1_ARR value and generates a counter overflow event (update). The counter begins up count again from 0. When the user enables the repeat count function, the repeat counter conducts down count at each overflow event. When the repeat counter down to 0 from the set value, generate an update event. Set TIM1_EGR register UG =1, also generate an update event.

Figure 10-6 Up counting mode（UDIS=0）



Configuring TIM1_CR1 register UDIS=1, disable update event. In case of counter overflow event, the update event is not generated. Configuring UG=1, the update event is not generated. However, the counter and prescaler counter will initialize, and begin up count from 0.

Figure 10-7 Up counting mode (UDIS =1, disable update event)



Note: In case of update event:

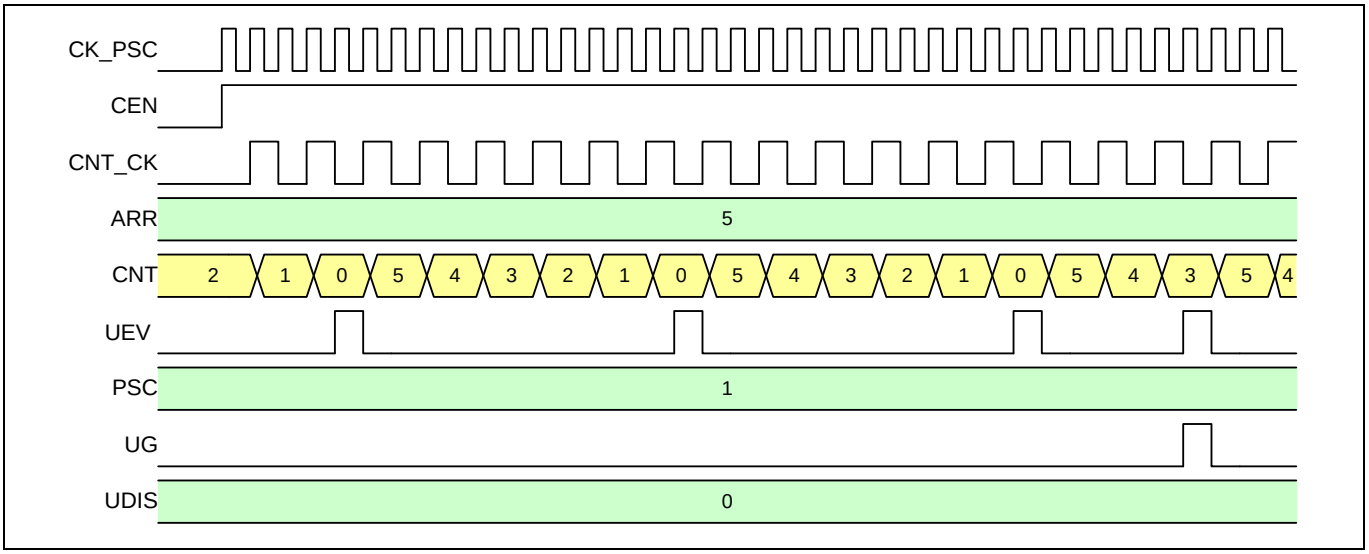
- Repeat counter is loaded with RCR register value, and down count again.
- ARR register value is loaded into ARR shadow register.
- Prescaler preload value becomes active.

10.4.1.3.2 Down counting mode

Configure TIM1_CR1 register CMS=0, DIR＝1, select down counting mode.

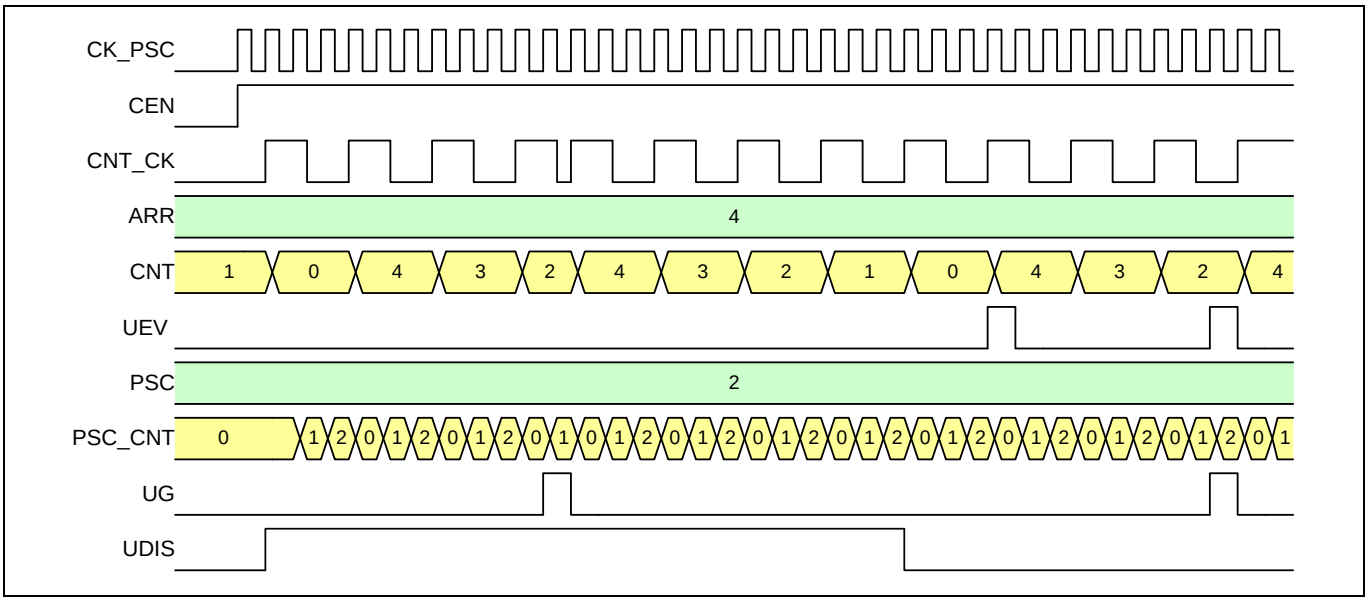
In down counting mode, the counter begins down count from the auto preload value TIM1_ARR. When counting to 0, generate an underflow event (update event). When the user enables the repeat count function, the repeat counter conducts down count at each underflow event. When the repeat counter down to 0 from the set value, generate the update event. Set TIM1_EGR register UG =1, also generate an update event. After the update event, begin down count (TIM1_CR1 register UDIS=0) from the auto load value TIM1_ARR.

Figure 10-8 Down counting mode（UDIS＝0）



Configuring TIM1_CR1 register UDIS=1, disable update event. When the counter generates the underflow event, it doesn't generate update event. Configuring UG=1 also doesn't generate update event. However, the counter and prescaler counter will initialize, and count begins from TIM1_ARR.

Figure 10-9 Down counting mode (UDIS=1 disables update event)



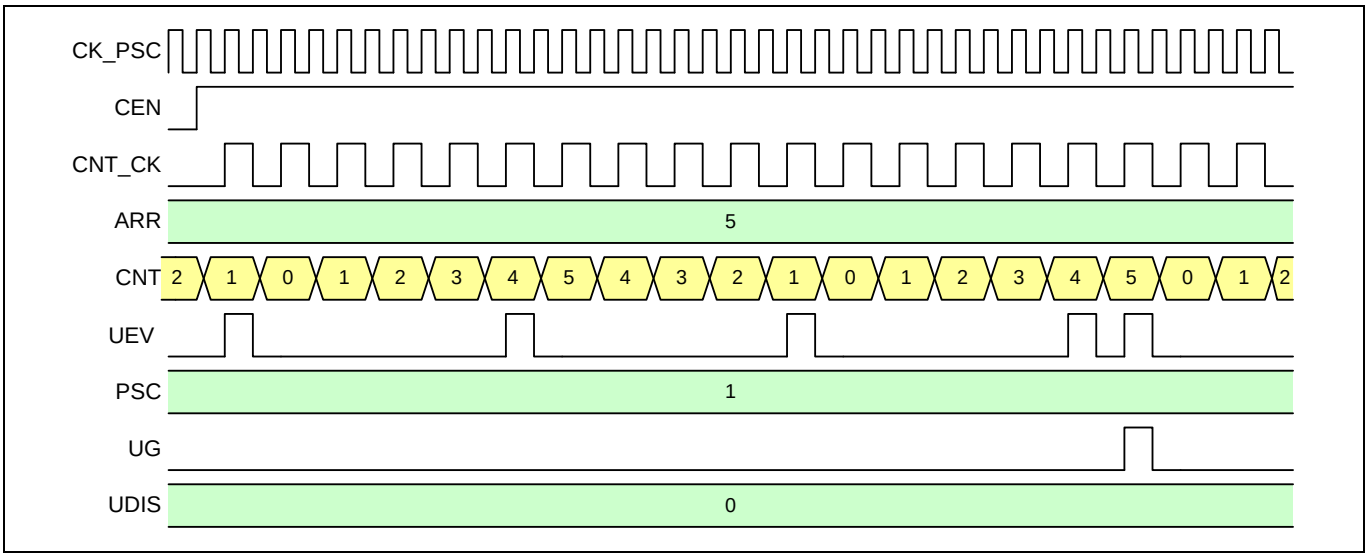
10.4.1.3.3 Central count mode (up/down count mode)

Configure TIM1_CR1 register CMS ≠ 0 (write DIR failure), and select central alignment count mode.

Central alignment counts mode, up count and down count run alternatively. In case of up count to ARR-1, generate an overflow event. Begin down count to 1 from ARR, generate an underflow event, and begin up count from 0.

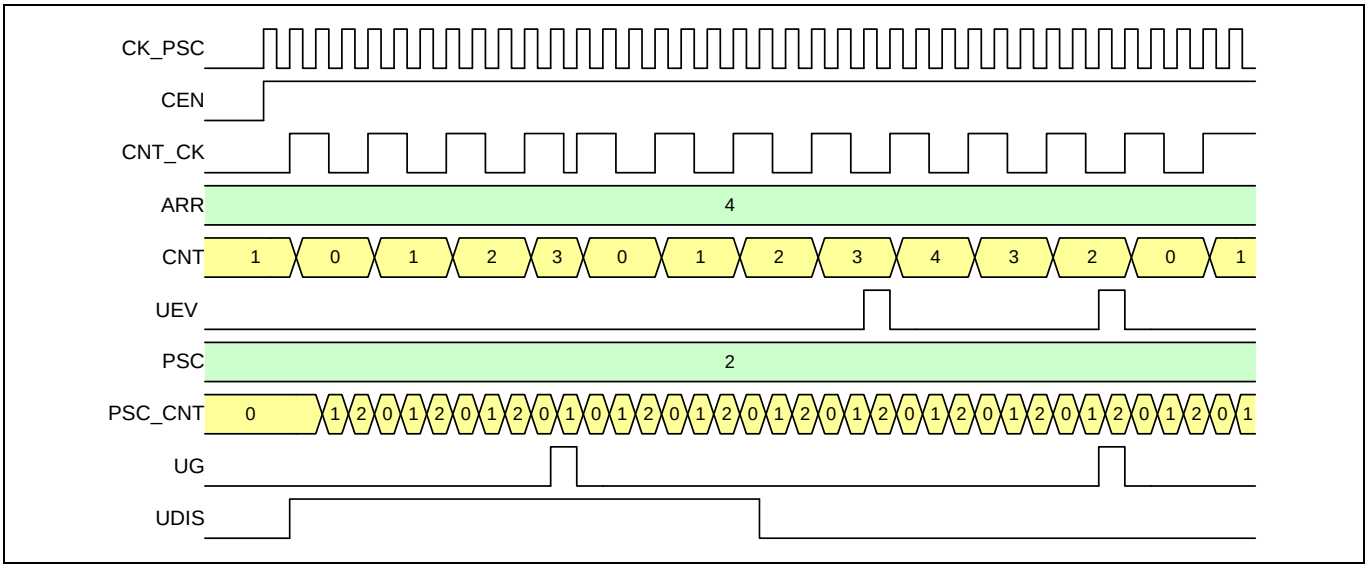
When the user enables repeat count function, the repeat counter counts down once at each overflow or underflow event. When the repeat counter down to 0 from the set value, generate an update event; setting TIM1_EGR register UG =1 also generate an update event. After the update event, the counter begins up count again from 0 (TIM1_CR1 register UDIS=0).

Figure 10-10 Central count mode（UDIS=0）



When configuring TIM1_CR1 register UDIS=1, disable update event. When the counter generates the overflow or underflow event, it doesn't generate update event. Configuring UG=1 also doesn't generate update event. However, the counter and prescaler counter will initialize, and the count begins from 0.

Figure 10-11 Central count mode (UDIS=1 disables update event)



10.4.2 Repeat counter

The repeat counter may adjust the frequency of the update event. During up count in the edge alignment mode, the repeat counter counting down at each overflow of the counter; during down count, the repeat counter counting down at each underflow of the counter. In the central alignment mode, the repeat counter counting down at the counter overflow and underflow. Configuring REP of the TIM1_RCR register to adjust the frequency of the update event. The repeat counter generates the update event after REP+1 count cycle. In the central alignment mode, the writing REP value decides whether the update event generates an overflow or underflow.

In case of the update event, REP value will be updated to the real-time repeat counter REP_CNT. To enable the real-time writing of REP_CNT achieves the flexible adjustment of the occurrence time of the update event.

Adjust DMA request frequency by configuring TIM1_PDER register CCDREPE. In case of CCDREPE=0, DMA request is generated by the value of the repeat count register. In case of CCDREPE=1, DMA request will happen at each overflow or underflow.

Figure 10-12 Repeat count time sequence in the central alignment mode

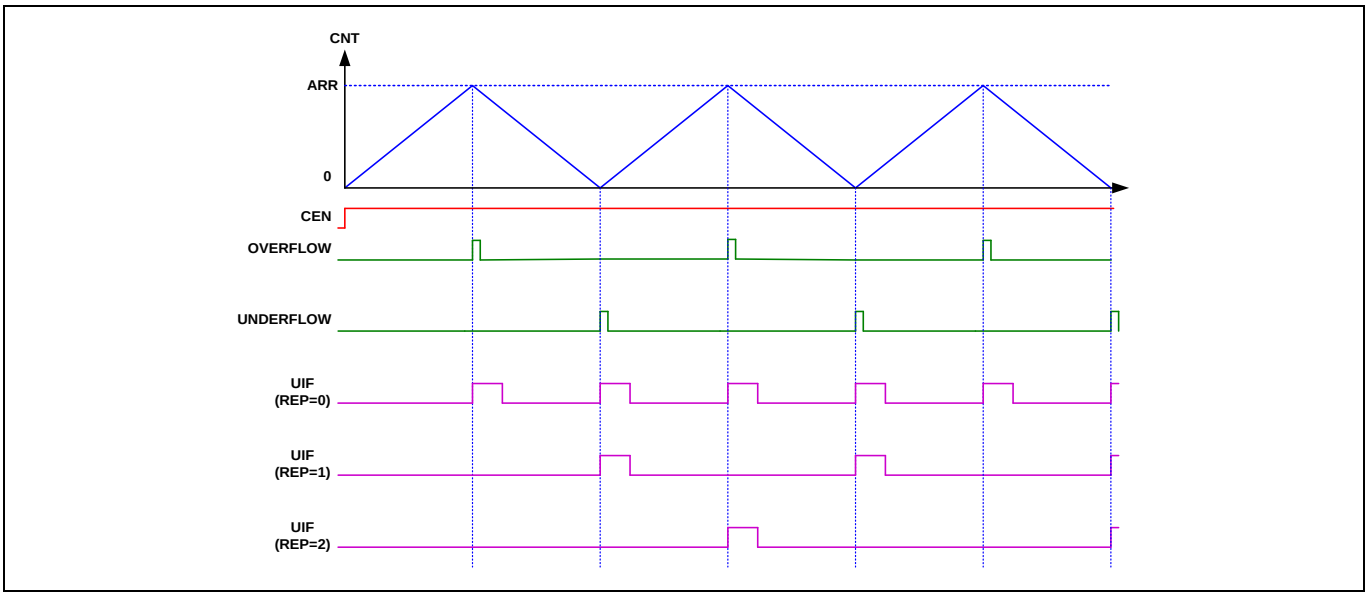


Figure 10-13 Up count time sequence in the edge alignment mode

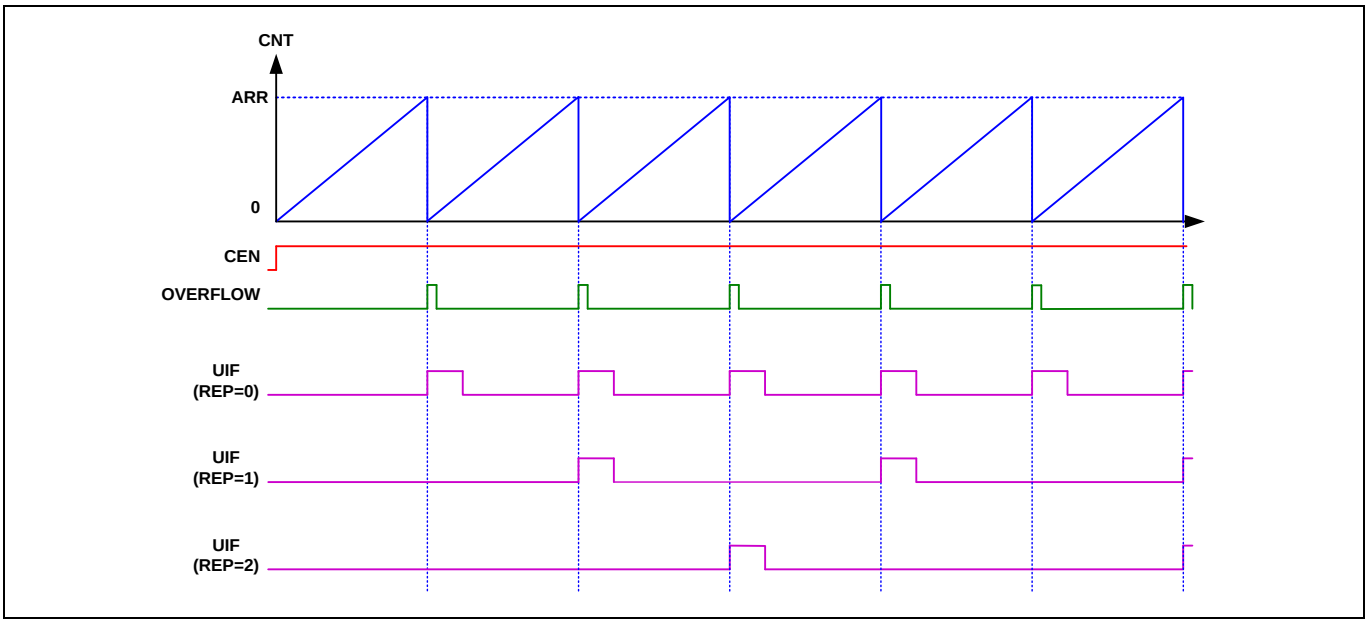
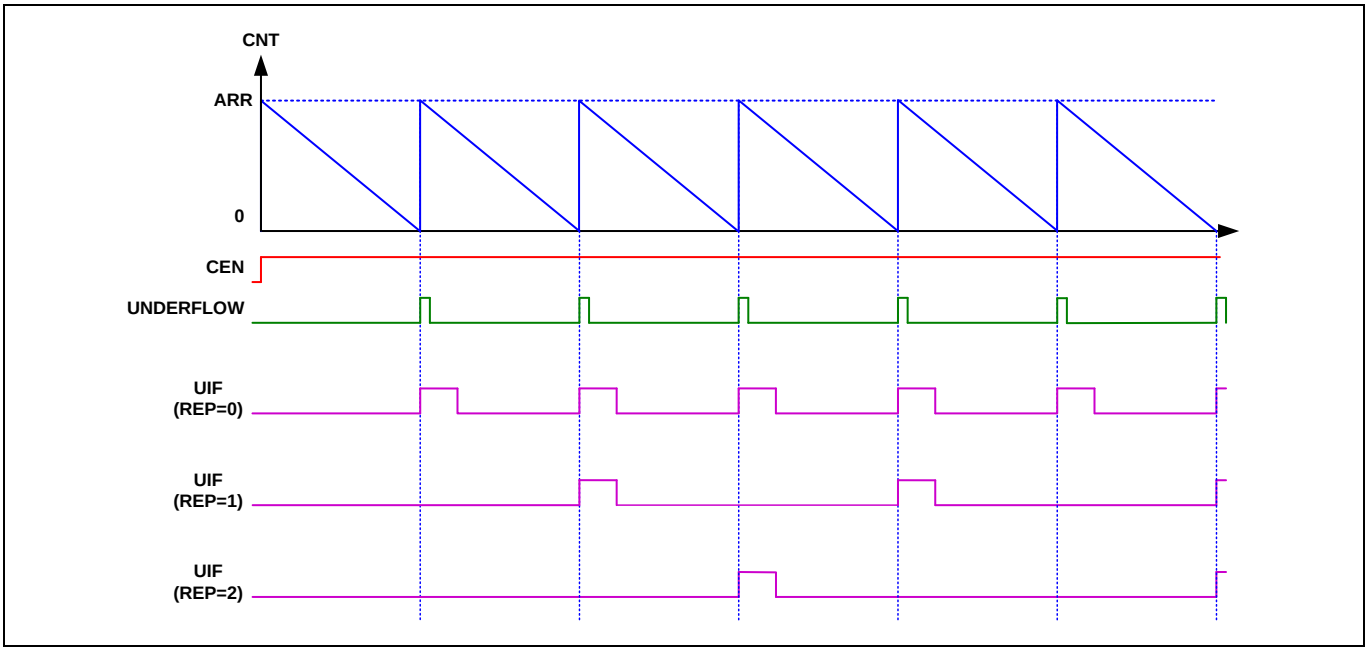


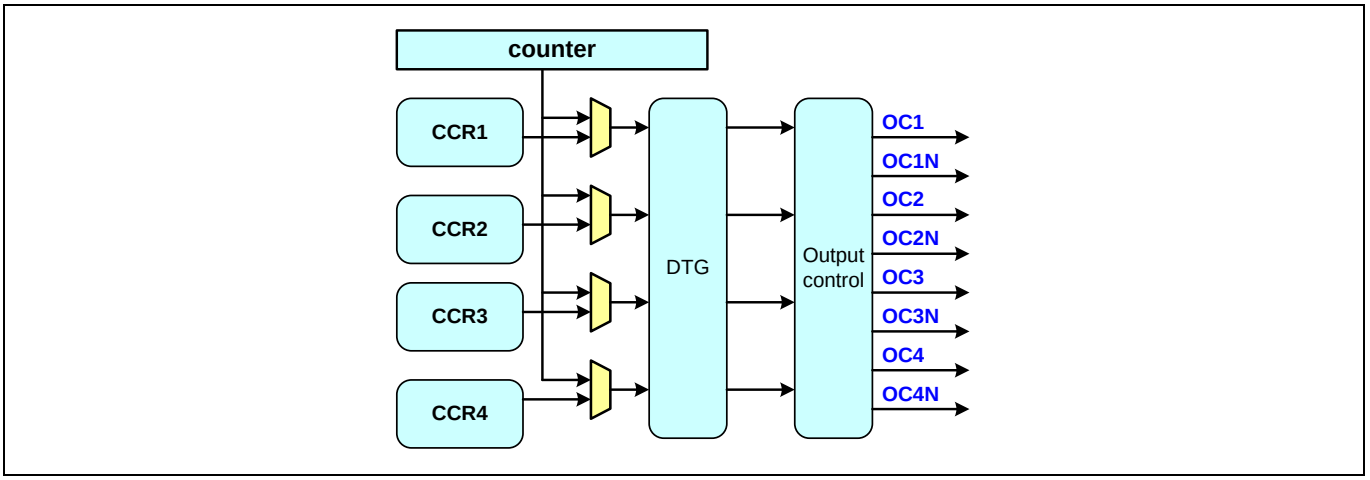
Figure 10-14 Down count time sequence in the edge alignment mode



10.4.3 Output compare

The compare output of the compare channel is composed of the comparator, output control circuit and compare register. Its structure is shown below:

Figure 10-15 Output compare block diagram



In output compare mode, the content of compare register is loaded into the shadow register, and then compare the content of shadow register with the current value of counter. The compare module includes a compare register (preload register) and a shadow register. When read and write, only programmed the compare register.

10.4.3.1 Force output

Configure TIM1_CCMRx register OCxM bit, and directly enforce the output compare signal as active or inactive level, independently of the output compare result. Configure TIM1_CCMRx register OCxM = 100, and enforce output compare signal as inactive level. And OCxREF is enforced as low level. Configure TIM1_CCMRx register OCxM = 101, and enforce output compare signal as active level. And the OCxREF is enforced as high level (OCxREF always active as high level).

Note: In force output mode, TIM1_CCRx shadow register and counter output compare is still performed. The corresponding flag bit of compare result will also be changed. When enable the

corresponding interrupt and DMA request, it will also generate the corresponding interrupt and DMA request.

10.4.3.2 Output compare

In the output compare mode, when the values of counter and compare register matches, different waveforms can be exported according to the OCxM bit of TIM1_CCMRx register.

For example, when the value of counter and compare register matches, the functions in the output compare mode:

1. When compare match, according to OCxM value, output channel x signal OCx run differently:
   - OCxM = 000: OCx signal keeps its level.
   - OCxM = 001: OCx signal is set as active level.
   - OCxM = 010: OCx signal is set as inactive level.
   - OCxM = 011: OCx signal toggle.
2. When match, set the flag bit of status register (CCxIF bit of TIM1_SR register).
3. Configure TIM1_DIER register CCxIE =1, generate an interrupt when match.
4. Configure TIM1_DIER register CCxDE =1, generate a DMA request when match (only suitable for the built in DMA product).

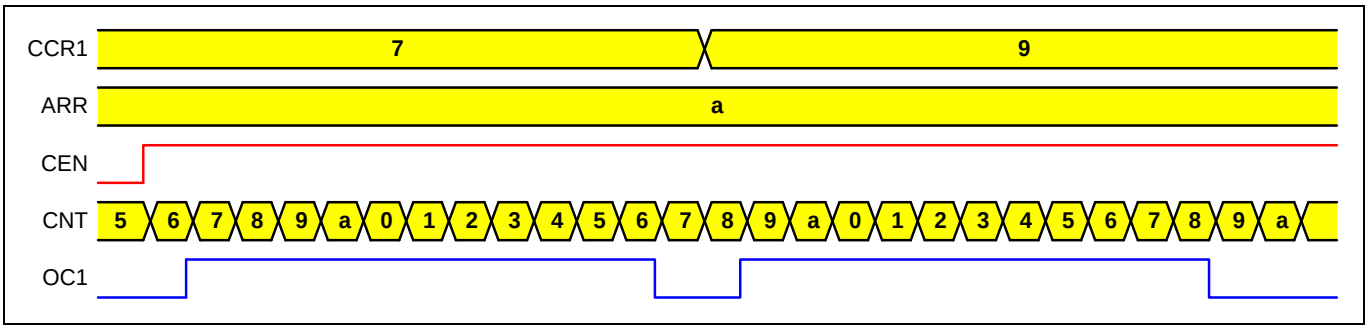
The output compare mode can also be used to export a single pulse (one-pulse mode).

Configuration steps of channel 1 output compare mode:

1. Configure counter clock (select clock source, configure prescaler factor).
2. Configure TIM1_ARR and TIM1_CCR1 register.
3. Configure TIM1_DIER register CC1IE =1, enable compare 1 interrupt.
4. Configure output mode:
   - Configure TIM1_CCMR1 register OC1M = 011, toggle when OC1 compare match.
   - Configure TIM1_CCMR1 register OC1PE = 0, disable preload function of TIM1_CCR1 register.
   - Configure TIM1_CCMR1 register CC1P = 1, OC1 is active low polarity.
   - Configure TIM1_CCER register CC1E = 1, enable channel 1 output, OC1 signal is exported to the corresponding output pin.
5. Configure TIM1_CR1 register CEN =1, enable the counter.

Configure TIM1_CCMRx register OCxPE=0, disable TIM1_CCRx register preload function, write TIM1_CCRx register any time, and the write value updated immediately. Configure TIM1_CCMRx register OCxPE=1, enable TIM1_CCRx register preload function, Updated only at the next update event. The figure below is an example.

Figure 10-16 Output compare mode, OC1 signals toggle when match



Note: In the output compare mode, the update event has no effect on output result. In the force output mode, the compare output between TIM1_CCRx shadow register and counter still performed. The corresponding flag bit of compare result is also be changed. When enabling the corresponding interrupt and DMA request, it still generates the corresponding interrupt and DMA request.

10.4.3.3 PWM output

In the PWM mode, according to the value of TIM1_ARR register and TIM1_CCRx register, generate a frequency and duty cycle controllable PWM waveform.

Configure OCxM=110 or OCxM=111 of TIM1_CCMRx register corresponding to the channel x, select channel x to PWM mode 1 or PWM mode 2. In the PWM mode, the counter will always compare with CCRx. According to configuration and compare results, the channel x exports different signals. Therefore, TIM1 can generate 4 PWM output signals with independent duty under the same frequency. In PWM mode, enable the preload function of TIM1_CCRx and TIM1_ARR. The written value of TIM1_CCRx preload register and TIM1_ARR preload register will become effective at the next update event, and load the corresponding shadow register. Therefore, in the PWM mode, before enabling counter, it's required to set TIM1_EGR UG=1, and generate the update event to initialize all registers.

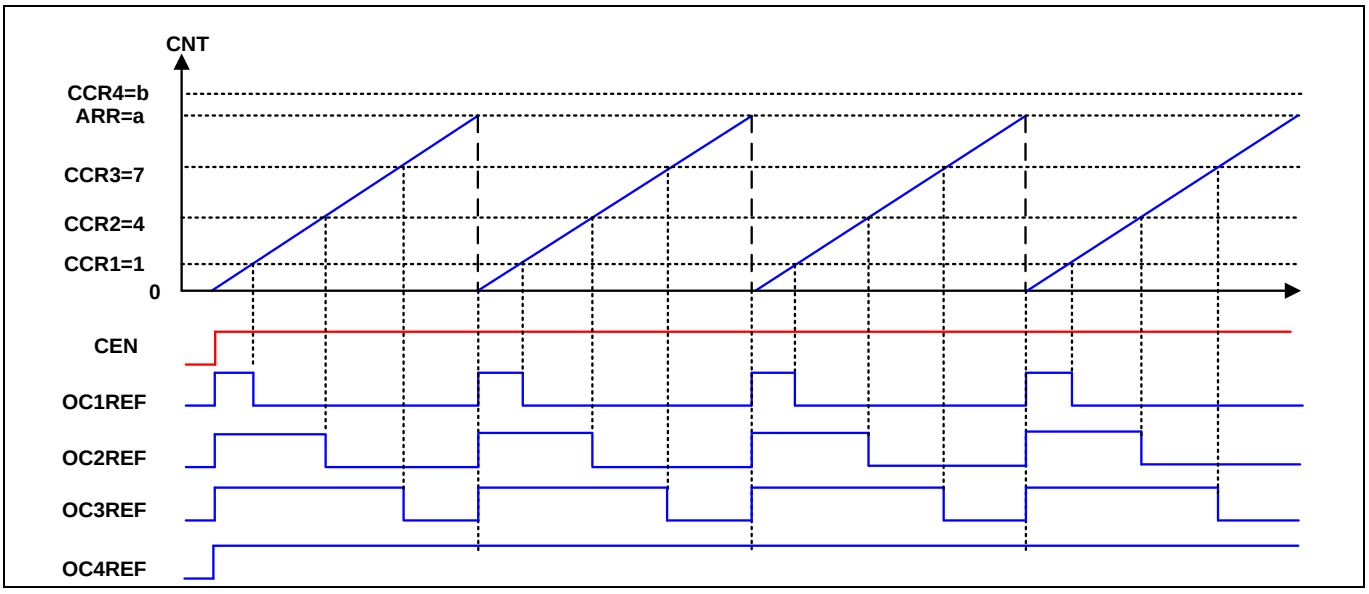
Configure TIM1_CCER register CCxP, select OCx's active polarity. Configure TIM1_CCER register CCxE, CCxNE bit and TIM1_BDTR register MOE, OSSI, OSSR bit and control OCx output enable. Configure TIM1_CR1 register CMS bit, and select edge alignment or central alignment PWM signal.

- CMS=00, edge alignment mode, configure DIR, select count mode (up or down).
- CMS=01, central alignment mode 1.
- CMS=10, central alignment mode 2.
- CMS=11, central alignment mode 3.

10.4.3.3.1 PWM edge alignment mode - up counting mode

Based on up counting mode configuration, configure OCxM =110, select PWM mode 1. In case of TIM1_CNT ＜TIM1_CCRx, channel x (OCxREF) is in active level. Otherwise, it's inactive level. If TIM1_CCRx compare value is greater than the auto reloads value (TIM1_ARR), OCxREF stays at an active level. When the value of the comparison is 0, OCxREF is at an inactive level. The figure below is the example of waveform in PWM mode 1 during edge alignment up count in case of CCR1 =0x1, CCR2 =0x4, CCR3 =0x7, CCR4 =0xb and ARR =0xa.

Figure 10-17 Waveform in PWM mode 1 during edge alignment up count

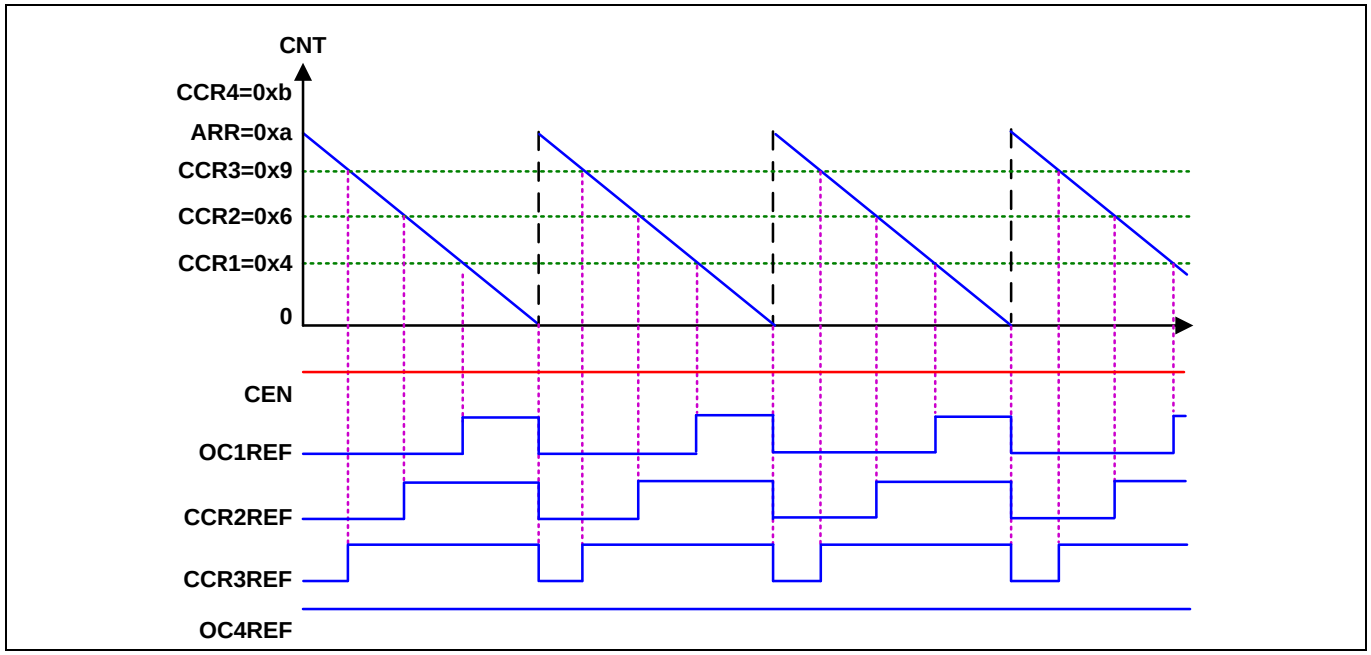


10.4.3.3.2 PWM edge alignment mode -down counting mode

Based on the down counting mode configuration, configureOCxM=110, select PWM mode 1. In

case of TIM1_CNT > TIM1_CCRx, channel x (OCxREF) is at inactive level. Or else, it's at an active level. The figure is an example of a waveform in PWM mode 1 during edge alignment down count in case of CCR1=0x4, CCR2=0x6, CCR3=0x9, CCR4=0xb and ARR=0xa.

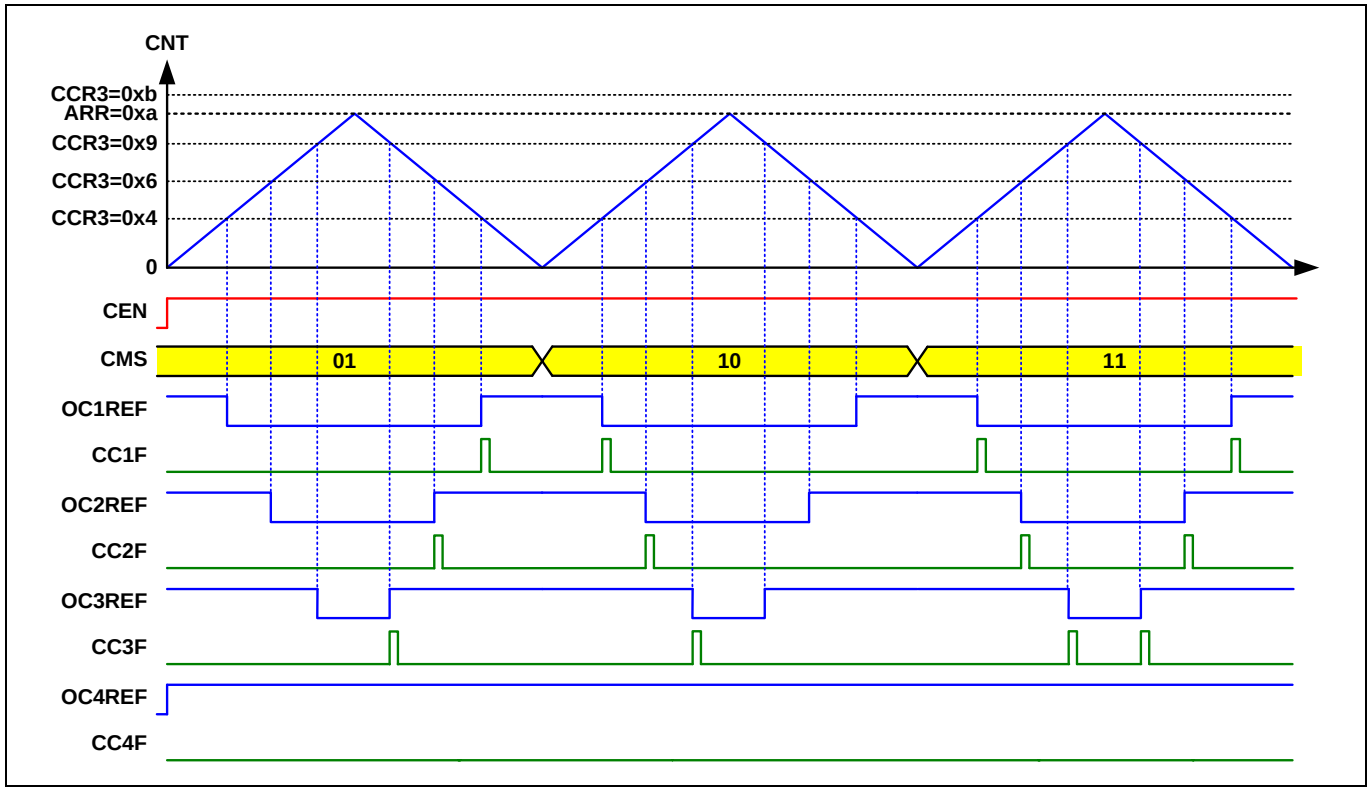
Figure 10-18 Waveform in PWM mode 1 during edge alignment down count



10.4.3.3.3 PWM central alignment mode

Configure TIM1 counter as central alignment count mode. According to the CMS with different configurations, the output compare interrupt flag bit is set during up count (CMS =01), set during down count (CMS =10) or set during up or down count (CMS =11). The figure below is an example of waveform in central alignment PWM mode 1 (CMS =1) during CCR1=0x4, CCR2=0x6, CCR3=0x9, CCR4=0xb, ARR=0xa.

Figure 10-19 Waveform in central alignment PWM mode 1



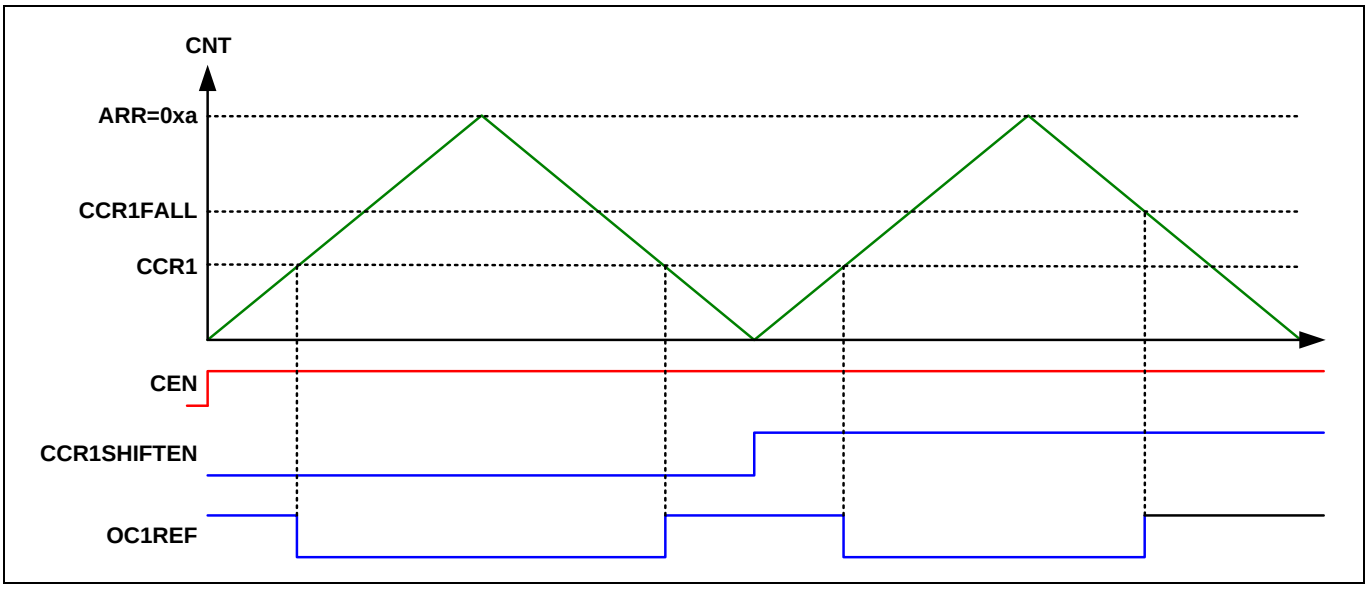
Note：

- In central alignment mode, use the current up/down count configuration, and the count direction depends on the current DIR value.
- In the central alignment mode, it's better not to change the counter value or it may generate unexpected result. When the counter is in up counting, write the counter value > TIM1_ARR, and the counter will continue up count. Writing 0 or ARR will immediately update the count direction. However, it won't generate the update event.
- In the central alignment mode, before enabling counter, configure TIM1_EGR register UG =1, generate a software update. Update all registers, don't change the counter value after enabling the counter.

10.4.3.3.4 Phase shift function in PWM central alignment mode

Configure PDER (channel x output PWM phase shift enable bit) and CCRxFALL register (channel x compare value in PWM central alignment down count mode), set the channel output PWM phase shift function. Configure CCRxFALL and CCRx, PWM output programmable phase shift waveform, and it can shift left or right.

Figure 10-20 Move phase function



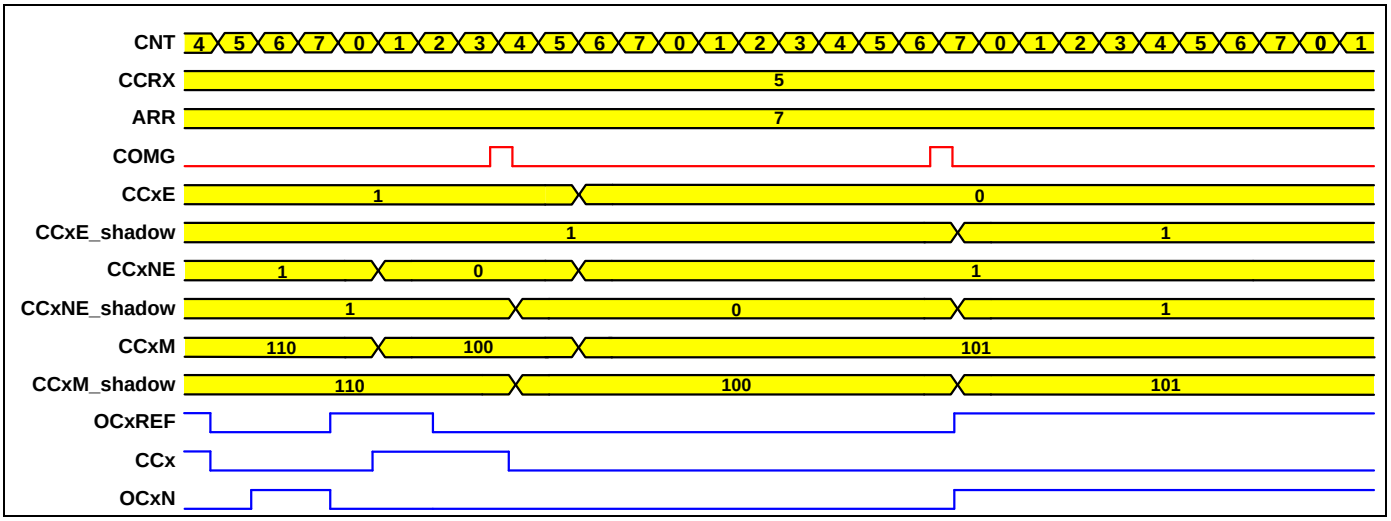
10.4.3.3.5 6-step PWM output

Configure OCxM and select output mode. CCxE=1 and CCxNE=1, Open channel x and complementary channel output enable. In channel x, it generates complementary output. These functions are in preload bit. In case of COM commutation event, these preload bits will be loaded in the corresponding shadow register. Thus, writing these bits won't influence the current output, and can change the configuration of all the channels at the same time. Configure TIM1_EGR register COMG =1 or generate COM event at TRGI rising edge.

In case of COM event, COM interrupt flag will be set by hardware. Configure TIM1_DIER register COMIE=1 and COMDE = 1, generate COM event will generate one COM interrupt and one DMA request at the same time (generate DMA request is only suitable for the built in DMA product).

OCx and OCxN output under different configurations in case of COM event.

Figure 10-21 Example of six step PWM with COM (OSSR = 1)

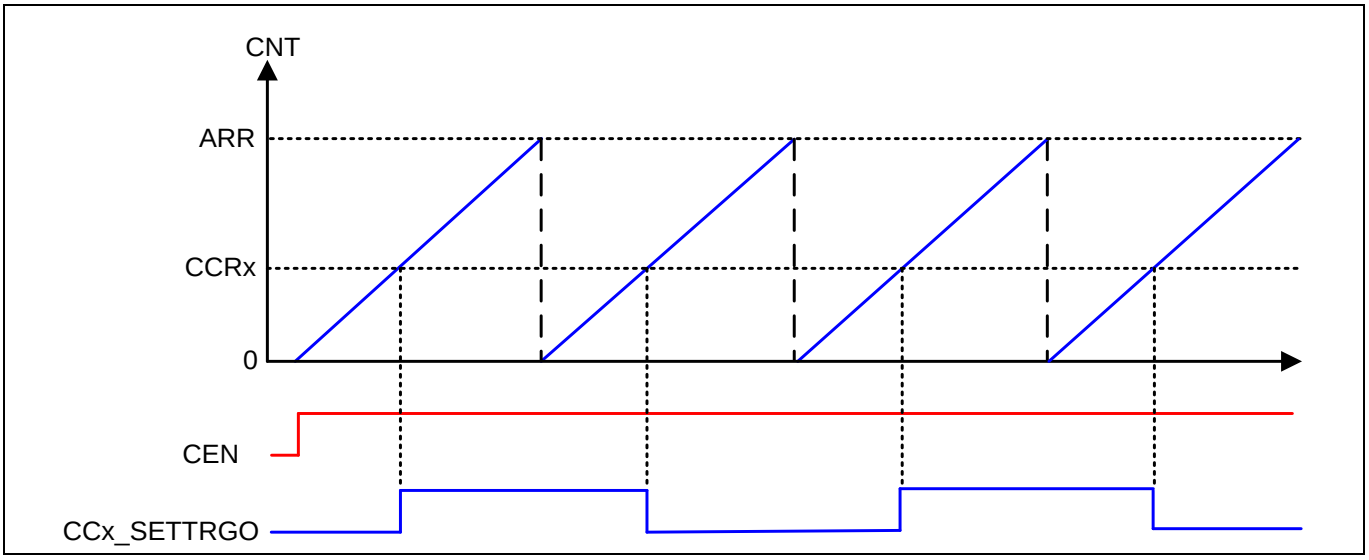


10.4.3.3.6 Trigger Source Output

In PWM mode, CCx_SETTRGO signal can be used to trigger modules such as ADC. This chapter only introduces CCx_SETTRGO. Please refer to the ADC chapter for the trigger logic of this signal, detailed trigger source selection, trigger edge selection, and other information.

In edge alignment mode, every time a match is compared (the current count value of TIMx_CNT is equal to TIMx_CCRx), CCx_ SETTRGO undergoes a flip. The following figure shows CCx_SETTRGO in edge aligned incremental counting mode example of output.

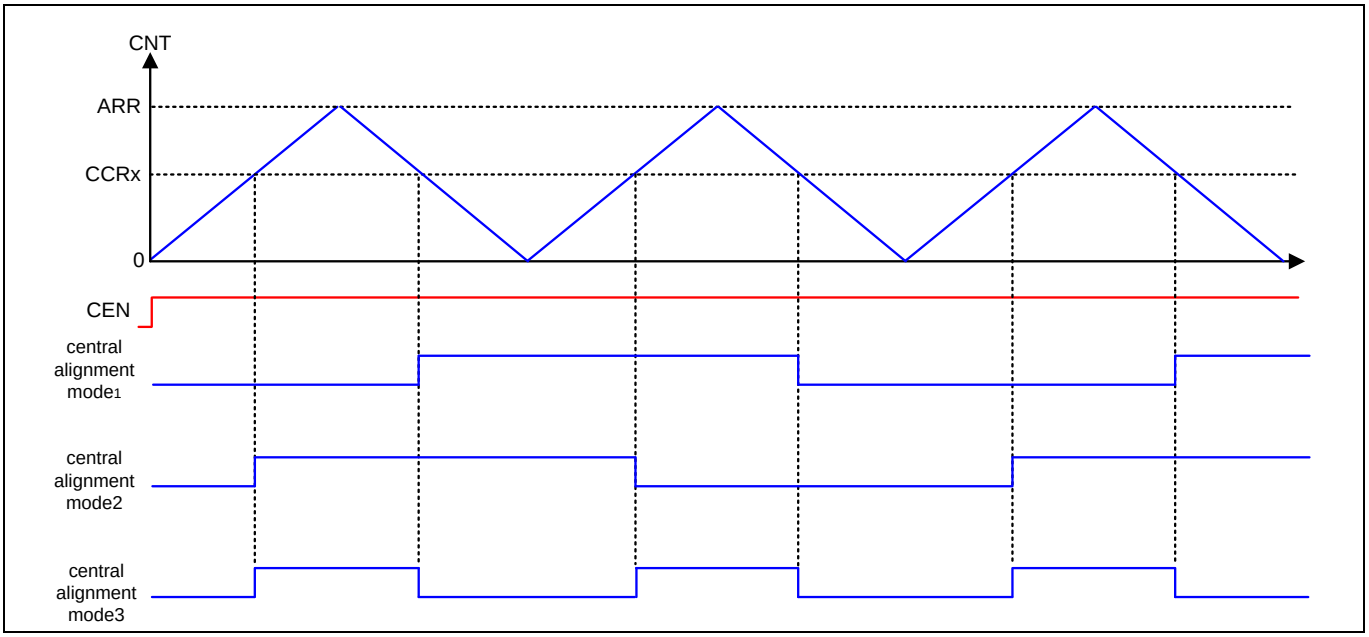
Figure 10-22 CCx_SETTRGO in edge aligned incremental counting mode output example



In central alignment mode 1, when comparing and matching the decreasing counting cycles, CCx_SETTRGO undergoes a flip. In central alignment mode 2, when comparing and matching the increasing counting cycles, CCx_SETTRGO undergoes a flip. In central alignment mode 3, CCx_SETTRGO is used when the increasing or decreasing counting cycles match undergoes a flip. The following figure shows CCx_SETTRGO in central alignment mode example of output.

Note: When using phase-shifting mode, the matching target for decreasing count comparison is CCRxFALL.

Figure 10-23 CCx_SETTRGO in central aligned incremental counting mode output example



10.4.3.4 Complementary output and dead-time insertion

OCx and OCxN are a pair of complementary output channels. TIMx all channels can export complementary signals with switching-off and the switching-on instants of the outputs. It also has a programmable dead-time area. The user adjusts the dead time according to the connected output devices and their characteristics (intrinsic delays of level-shifters, delays due to power switches).

TIM1_BDTR register DTG [7:0] bit defines the dead-time when inserting the complementary output. The specific calculation mode is given in the table below:

Table 10-1 Dead-time calculation

DTG[7：5]	DT
0xx	DT= DTG[7：0] x T_{dtg}（T_{dtg}=T_{DTS}）
10x	DT=（64+DTG[5：0]）x T_{dtg}（T_{dtg} =2xT_{DTS}）
110	DT=（32+DTG[4：0]）x T_{dtg}（T_{dtg} =8xT_{DTS}）
111	DT=（32+DTG[4：0]）x T_{dtg}（T_{dtg} =16xT_{DTS}）

For example, in case of T_{DTS}=125ns, possible dead time:

- In case of step time 125ns, dead time 0-15875ns.
- In case of step time 250ns, dead time 16μs -31750ns.
- In case of step time 1μs, dead time 32μs -63μs.
- In case of step time 2us, dead time 64μs -126μs.

In case of no break circuit, configure CCxE=1 and CCxNE=1, enable dead time insert. Otherwise, configure MOE =1.

Configure TIM1_CCER register CCxP and CCxNP, and independently select polarity for each output (main output OCx or complementary output OCxN).

Configure TIM1_CCER register CCxE and CCxNE bit. Different combinations of MOE, OISx, OISxN, OSSI and OSSR in TIM1_BDTR and TIM1_CR2 register can control the complementary signal OCx and OCxN output. The specific combination control configuration is given in the complementary output channel OCx and OCxN control bit in table 2, table 3, table 4 and table 5 in this chapter.

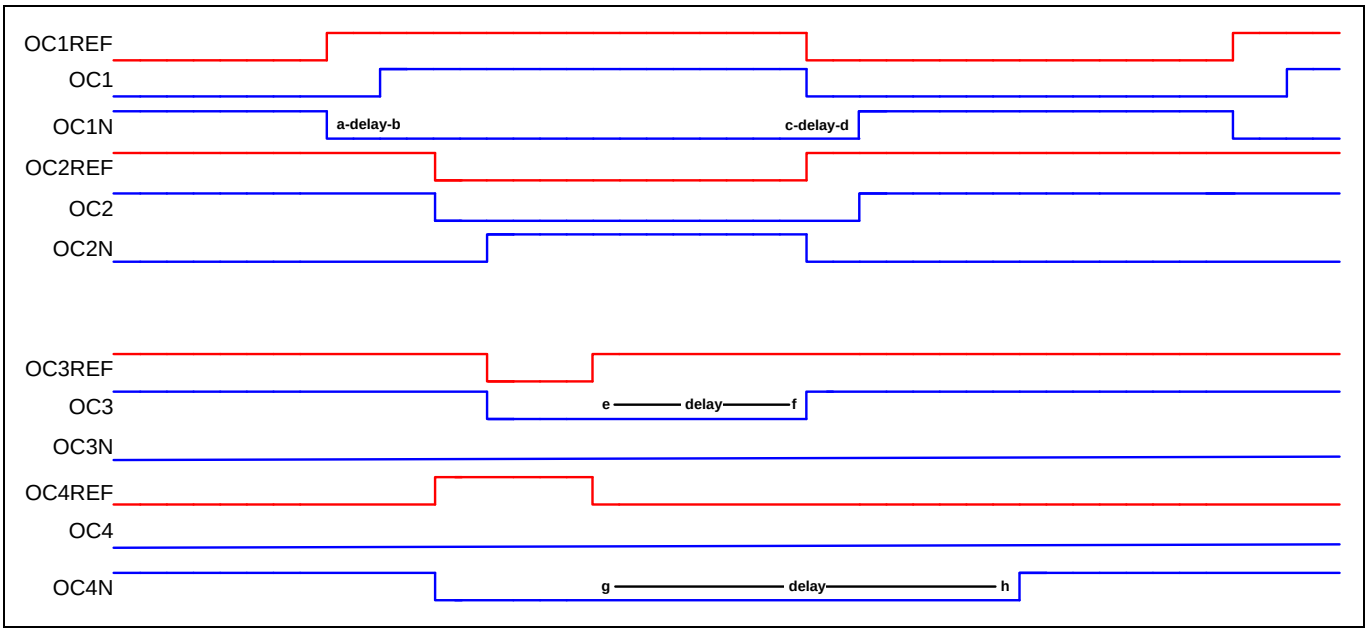
For example: OCx and OCxN are active high. In PWM mode, in case of match, export reference signal OCxREF signal toggle and output signal OCx and reference signal are same. However, OCx signal's rising edge has a latency with response to the reference signal; complementary output signal

OCxN and reference signal are opposite. OCxN signal's rising edge has a latency with response to the reference signal's falling edge.

Note: Dead time cannot be more than or equal to the duty cycle of OCx or OCxN signal (active output). Or else, OCx or OCxN signal are always inactive values.

The figures below show the relation of dead-time generator output signal and current reference signal OCxREF.

Figure 10-24 Dead-time insertion



10.4.3.5 Break function

Break input pin, clock failure event and comparator output may serve as TIM1 break source. The clock failure event is generated by the clock safety system of the reset clock counter.

During the use of break function, OCx and OCxN output signal levels are controlled by the function level combination: TIM1_BDTR register MOE, OSSI and OSSR bit, and TIM1_CR2 register OISx and OISxN bit. In case of break event, OCx and OCxN output cannot be active at the same time. The output state of complementary output channel OCx and OCxN with the break function is shown in the table below.

Table 10-2 In case of MOE=1, OSSI=0/1, OSSR=0:

CCxE	CCxNE	OCx	OCxN
0	0	OCx=0，OCx_EN=0	OCxN=0，OCxN_EN=0
0	1	OCx=0，OCx_EN=0	OCxN=OCxREF+Polarity，OCxN_EN=1
1	0	OCx=OCxREF+Polarity，OCx_EN=1	OCxN=0，OCxN_EN=0
1	1	OCx=OCxREF+Polarity+dead-time, OCx_EN=1	OCxN=OCxREF+reverse phase +Polarity+dead-time, OCxN_EN=1

Table 10-3 In case of MOE=1, OSSI=0/1, OSSR=1

CCxE	CCxNE	OCx	OCxN
0	0	OCx=0，OCx_EN=0	OCxN=0，OCxN_EN=0
0	1	OCx=CCxP，OCx_EN=1	OCxN=OCxREF+Polarity，OCxN_EN=1
1	0	OCx=OCxREF+Polarity, OCx_EN=1	OCxN=CCxNP，OCxN_EN=1
1	1	OCx=OCxREF+Polarity+dead-time, OCx_EN=1	OCxN=OCxREF reverse phase+Polarity+dead-time, OCxN_EN=1

Table 10-4 In case of MOE=0, OSSI=0, OSSR=0/1

CCxE	CCxNE	OCx	OCxN
0	0	OCx_EN=0，OCxN_EN=0 Insync：OCx=CCxP，OCxN=CCxNP In case the clock is available: After a dead-time, OCx=OISx，OCxN=OISxN OISx and OISxN don't correspond to the active level of OCx and OCxN	
0	1	OCx_EN=0, OCxN_EN=0 Insync：OCx=CCxP，OCxN=CCxNP In case the clock is available: After a dead-time, OCx=OISx，OCxN=OISxN OISx and OISxN don't correspond to the active level of OCx and OCxN	
1	0	OCx_EN=0，OCxN_EN=0 Insync：OCx=CCxP，OCxN=CCxNP In case the clock is available: After a dead-time, OCx=OISx，OCxN=OISxN OISx and OISxN don't correspond to the active level of OCx and OCxN	
1	1	OCx_EN=0，OCxN_EN=0 Insync：OCx=CCxP，OCxN=CCxNP In case the clock is available: After a dead-time, OCx=OISx，OCxN=OISxN OISx and OISxN don't correspond to the active level of OCx and OCxN	

Table 10-5 In case of MOE=0, OSSI=1, OSSR=0/1

CCxE	CCxNE	OCx	OCxN
0	0	OCx_EN=1，OCxN_EN=1 Insync：OCx=CCxP，OCxN=CCxNP In case the clock is available: After a dead-time, OCx=OISx，OCxN=OISxN OISx and OISxN don't correspond to the active level of OCx and OCxN	
0	1	OCx_EN=1, OCxN_EN=1 Insync：OCx=CCxP, OCxN=CCxNP In case the clock is available: After a dead-time, OCx=OISx，OCxN=OISxN OISx and OISxN don't correspond to the active level of OCx and OCxN	
1	0	OCx_EN=1，OCxN_EN=1 Insync：OCx=CCxP，OCxN=CCxNP In case the clock is available: After a dead-time, OCx=OISx，OCxN=OISxN OISx and OISxN don't correspond to the active level of OCx and OCxN	
1	1	OCx_EN=1，OCxN_EN=1 Insync：OCx=CCxP，OCxN=CCxNP In case the clock is available: After a dead-time, OCx=OISx，OCxN=OISxN OISx and OISxN don't correspond to the active level of OCx and OCxN	

Note: When the channel output and complementary output are closed, OISx, OISxN, CCxP and CCxNP must be configured as 0.

After system reset, MOE =0, disable break function. Configure TIM1_BKINF register BKIN_SEL and select break channels, it supports the selection of multiple break channels, it will enter the break state if break signal from any selected channel is active. Configure BKINFE of TIM1_BKINF register, enable the break signal filtering function. Configure BKINF of TIM1_BKINF register, select the sampling frequency of break digital filter. Before changing the sampling frequency of break digital filter, turn off the break filter function. Configure TIM1_BDTR register BKE =1 and enable break function. Configure TIM1_BDTR register BKP bit and select break input signal polarity. BKP and BKE can also be written at the same time and become active in the next clock period.

Because MOE is asynchronous cleared, insert a synchronous circuit in between the actual signal and synchronous control, and generate latency between the synchronous and asynchronous signal (write MOE =1 when the state is 0; input an dummy instruction for latency after write and before read. Otherwise, correct reading cannot be guaranteed).

In case of break events, MOE is asynchronously cleared. According to OSSI configuration OCx/OCxN output, put it in inactive state, idle state and reset state; in case of MOE=0, the output is determined by TIM1_CR2 register OISx bit. In case of OSSI =0, the timer disables the output. Otherwise, it enables the output. When using the complementary output, the output is first put in the reset state. The dead-time is generated again. After the dead-time, the output level is determined by OISx and OISxN.

Configure TIM1_DIER register BIE =1. In case of a break event, generate a break interrupt. Configure TIM1_BDTR register AOE =1, auto set MOE bit when the next update event occurred.

Note: The break input is acting on level. When the break input is active, cannot (auto or via

software) set MOE, and the status flag BIF cannot be cleared.

In the break circuit, enable write protection and safeguard application safety. Allow the user to lock the dead-time duration, OCx/OCxN polarity and disabled state, OCxM configuration, break enable and polarity. Configure TIM1_BDTR register LOCK bit, select lock levels (three levels of lock). Lock can only be written once after system reset.

The figure below shows the example of response break output:

Figure 10-25 Response break output（OISx=0 , OISxN=0）

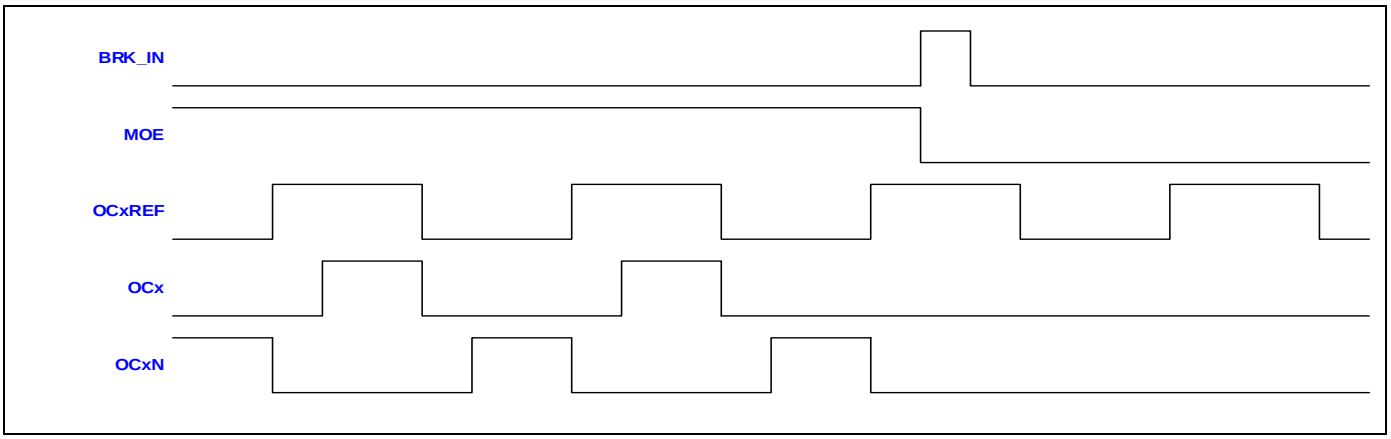


Figure 10-26 Response break output (OISx=0, OISxN=1)

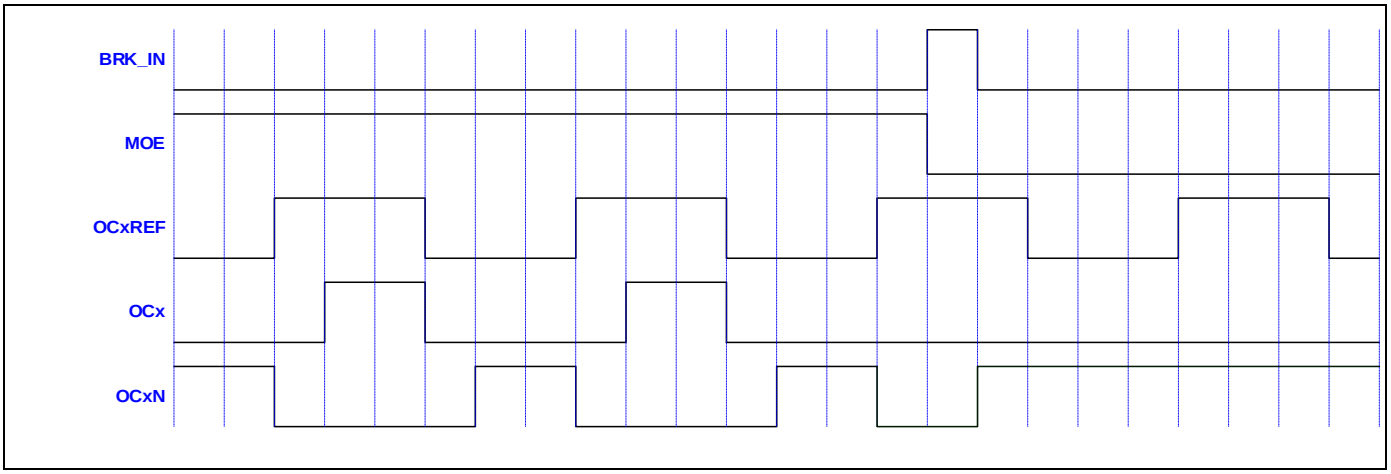


Figure 10-27 Response break output (OISx=1,OISxN=0）

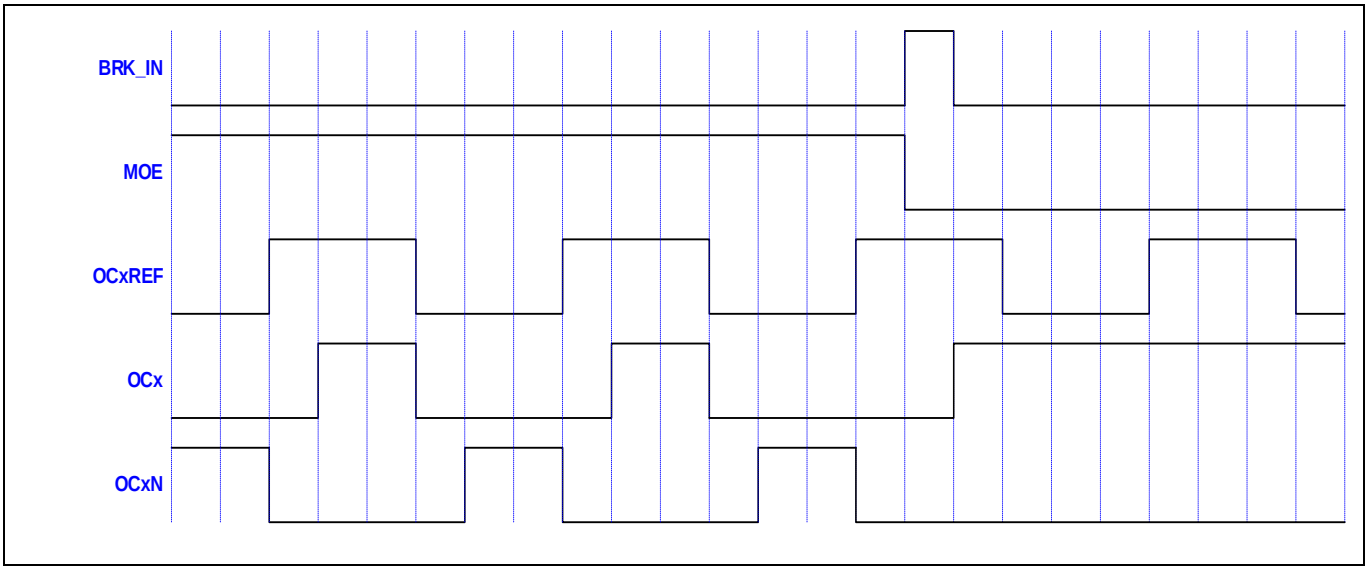
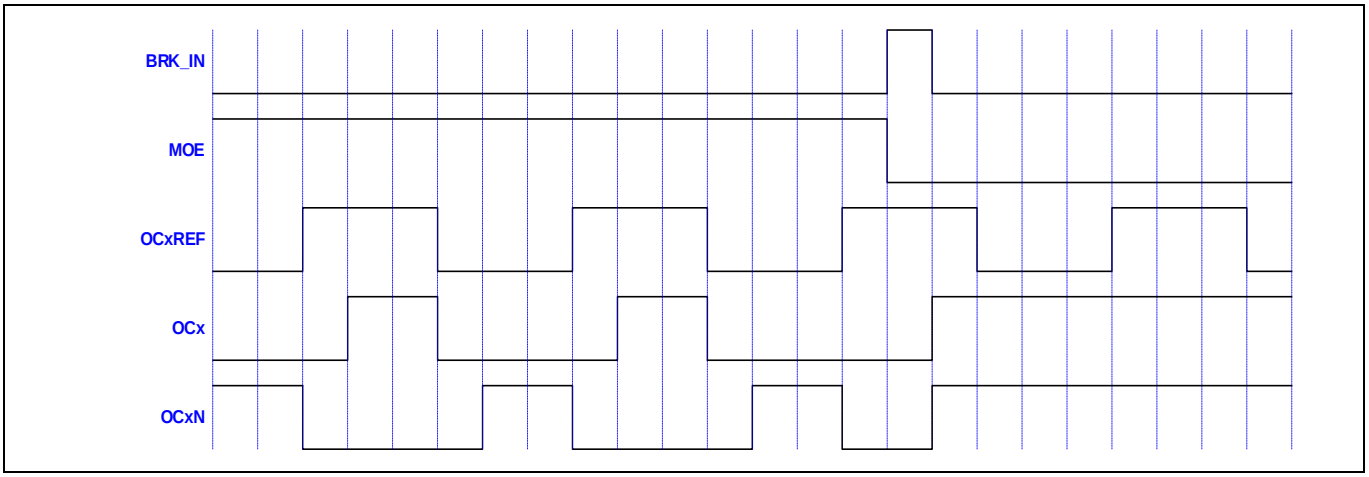


Figure 10-28 Response break output (OISx=1, OISxN=1)



10.4.3.6 External event clear OCxREF

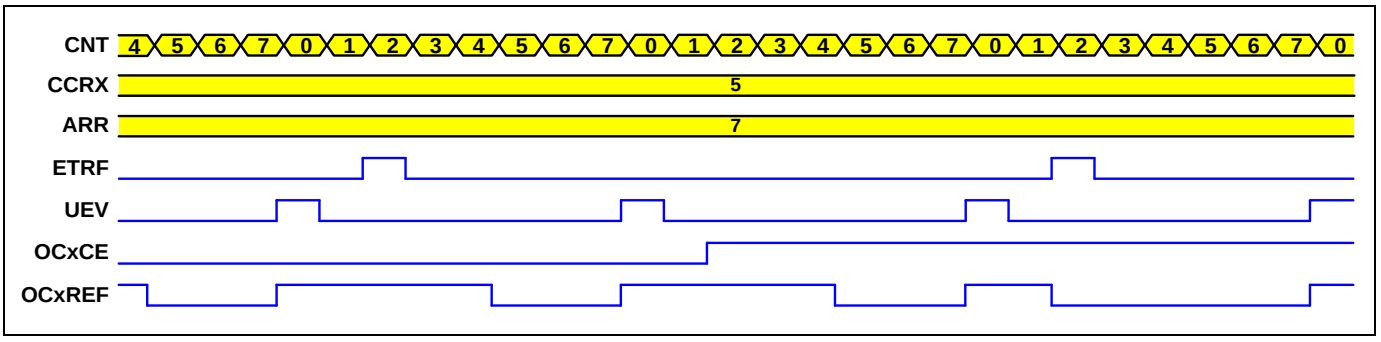
When configuring TIM1_CCMR register OCxCE=1, OCxREF can be driven low by the active level of ETRF input terminal till the next update event (UEV). The function can only be used in the output mode and PWM mode, it does not work in the force output mode.

For example, when OCxREF signal is connected to an external input, ETR configuration:

1. Configure TIM1_SMCR register ETPS [1:0] = 00, close the external trigger prescaler.
2. Configure TIM1_SMCR register ECE=0, disable external clock mode 2.
3. Configure TIM1_SMCR register ETF [3:0] and ETP, configure ETR signal trigger polarity and filter width.

In the figure below, when ETRF input change triggers ETRF as high, it corresponds to different OCxCE value, and OCxREF signal action (PWM mode).

Figure 10-29 The external event clear OCxREF



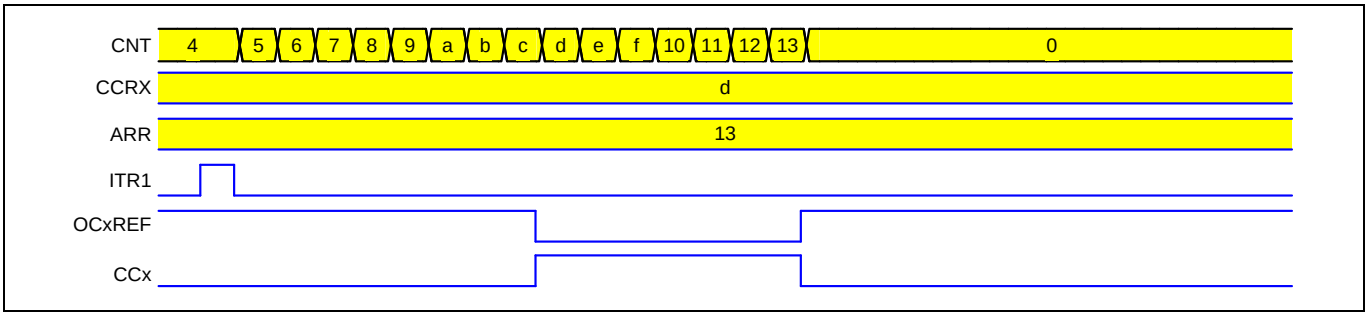
10.4.3.7 One-pulse output

In the one-pulse mode (OPM), the counter responds to a stimulus and generates a pulse with width programmable. Configure TIM1_CR1 register OPM =1. Select one-pulse mode, trigger signal's active edge or configuring CEN =1 can enable the counter. CEN =1 remains until the next update event or configure CEN =0.

The necessary condition to generate pulse is the difference of compare value and initial value of the counter. Necessary configuration prior to counter enable:

- Up counting mode: Counter CNT < CCRx ≤ ARR.
- Down counting mode: Counter CNT > CCRx.

Figure 10-30 one-pulse mode



For example, when ITR1 detects the rising edge, and after latency t_{DELAY}, OC2 generates a positive pulse with the length of t_{PULSE}.

Configure ITR1 serves as the trigger source:

1. Configure TIM1_SMCR register TS = 001, ITR1 as the slave mode counter trigger (TRGI).
2. Configure TIM1_SMCR register SMS = 110, select trigger mode, ITR1 enables the counter.

OPM waveform is determined by TIM1_ARR and TIM1_CCR1 (consider clock frequency and counter prescaler): t_{DELAY} is determined by TIM1_CCR1 register value and CNT initial value; TIM1_ARR - TIM1_CCR1 value is t_{PULSE}.

Next example is generating a negative pulse. In case of compare and match, generate a waveform from 1 to 0. When the counter preload value is achieved, generate a waveform from 0 to 1:

1. Configure TIM1_CCMR1 register OC1M = 111, select PWM mode 2.
2. Configure TIM1_CCER register CC1P = 1, export active low level.
3. Configure TIM1_CCMR1 OC1PE = 1 and TIM1_CR1 register ARPE, enable preload register.
4. Configure TIM1_CCR1 register and TIM1_ARR register.
5. Configure TIM1_EGR register UG =1, generate an update event.
6. Wait for an external trigger event on ITR1.

In this case, TIM1_CR1 register DIR=0, CMS=0, OPM= 1, stop counting in the next update event (when the counter toggles to 0 from the auto load value).

10.4.3.7.1 OCx fast enable

OCx fast enable is a special scenario of one-pulse mode. In the one-pulse mode, set TIM1_CCMR register OCxFE=1, and enforce OCxREF to directly respond to stimulus without relying on the compare result of counter and compare value. The output waveform is the same as the waveform when compare and match. Thus, it can remove the compare time and export the compare result fast. OCx rapid output enables only becomes active in the PWM mode.

10.4.4 Slave mode

10.4.4.1 Reset mode

Configure TIM1_SMCR register SMS=100, select slave reset mode. In this mode, the TRGI input event will clear and restart the counter.

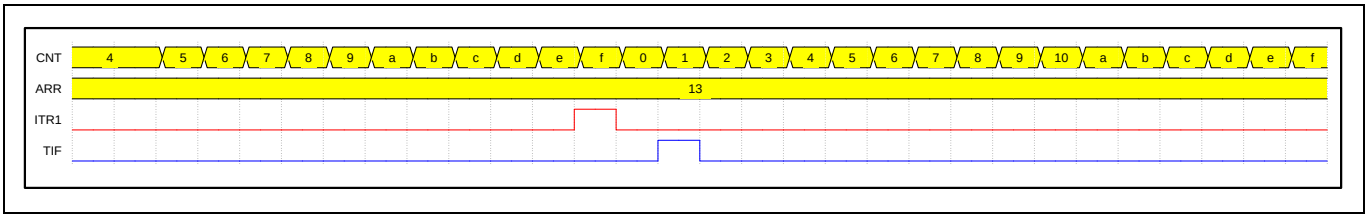
For example, ITR1 trigger counter restart:

1. Configure TIM1_SMCR register SMS = 100 and select the slave mode as reset mode. Configure TIM1_SMCR register TS =001, select ITR1 as the trigger input of sync timer.
2. Configure TIM1_CR1 register DIR=0, select the count direction as the up count; configure PSC=0, no division; configure CEN=1, enable the counter.

The counter clock source is provided by the internal clock. When detecting ITR1 rising edge, clear and restart the counter. The trigger interrupt flag is set by hardware.

TIM1_ARR = 0x13 time sequence in reset mode

Figure 10-31 Control time sequence in reset mode



10.4.4.2 Gate mode

Configure TIM1_SMCR register SMS=101, select slave gate mode. When TRGI input is active level, the counter always starts. Otherwise, the counter stops (but without reset operation), and the counter start and stop are countable.

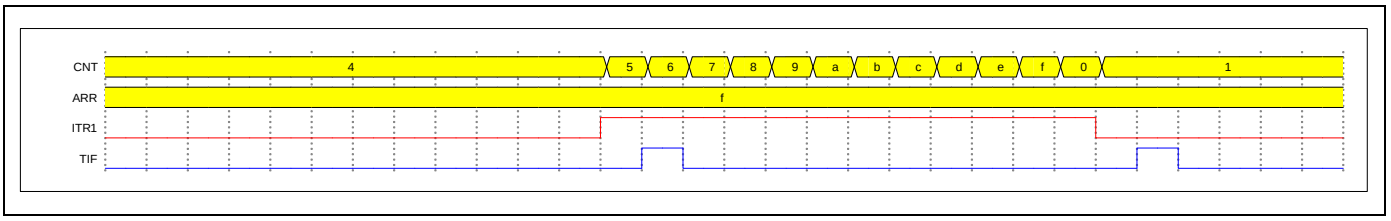
For example, the counter only counts when ITR1 is high:

1. Configure TIM1_SMCR register SMS=101, select slave gate mode; configure TIM1_SMCR register TS=001, select the ITR1 as the sync counter trigger input.
2. Configure TIM1_CR1 register DIR=0, select count direction as the up count; configure PSC=0, no division; configure CEN=1, enable the counter.

The counter clock source is provided by the internal clock. When detecting ITR1 high level, the counter count starts. When ITR1 is low level, the counter stops. When the counter starts or stops, set TIF as 1.

TIM1_ARR=0xf time sequence in gate mode.

Figure 10-32 Control time sequence in gate mode



10.4.4.3 Trigger mode

Configure TIM1_SMCR register SMS=110 and select slave trigger mode. When TRGI input is active edge, the counter starts counting. The timer startup is controlled, and the stop is not controlled.

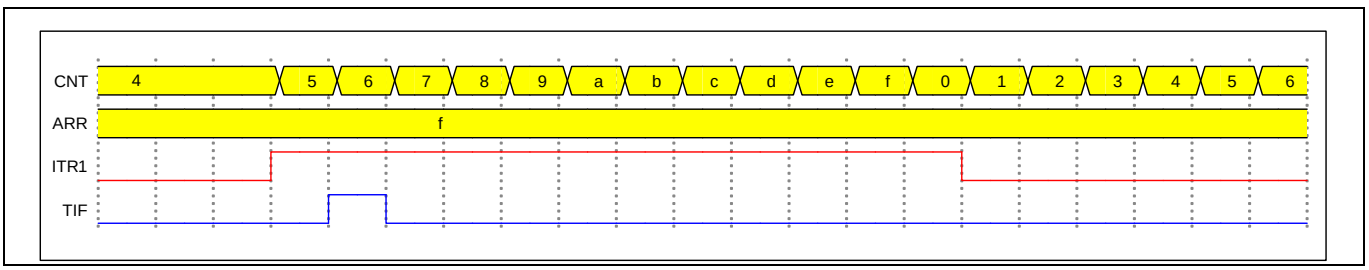
For example, the counter count starts at the rising edge of ITR1 input:

1. Configure TIM1_SMCR register SMS = 110, select slave trigger mode; configure TIM1_SMCR register TS=001, select ITR1 as the trigger input of counter.
2. Configure TIM1_CR1 register DIR =0, select count direction as the up count; configure PSC=0, no division.

The counter clock source is provided by internal clock. When detecting ITR1 rising edge, the counter count starts.

TIM1_ARR =0xf sequence in the trigger mode.

Figure 10-33 Control time sequence in trigger mode



10.4.4.4 External clock mode 2+ slave mode

When the clock source selects the external clock mode 2, and ETR signal is used as external clock input, it can be used together with the other slave mode. In this mode, the slave mode only supports the reset mode, gate mode, trigger mode, and not supports the external clock mode 1.

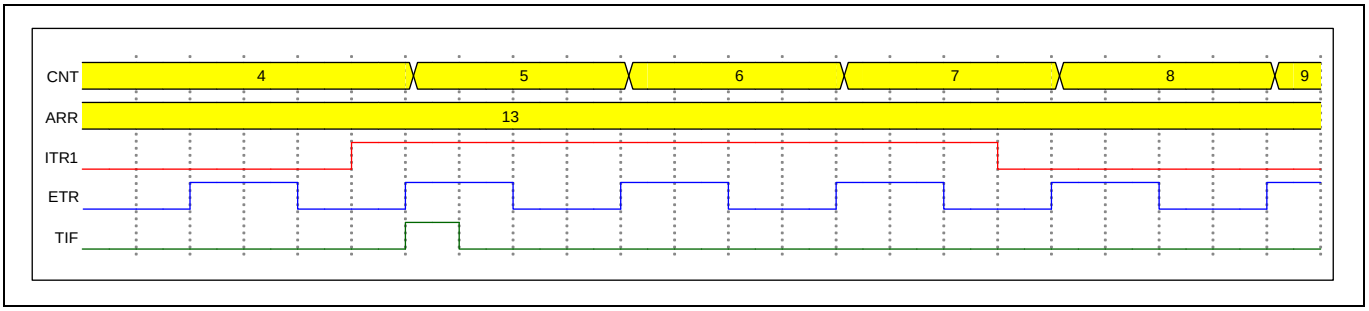
For example, in the trigger mode, the counter counts once at each rising edge of ETR:

1. Configure TIM1_SMCR register ETF = 0000, don't use the digital filter; configure TIM1_SMCR register ETPS = 00, close the prescaler; configure TIM1_SMCR register ETP =0, detect ETR rising edge; configureTIM1_SMCR register ECE = 1, enable external clock mode 2.
3. Configure TIM1_SMCR register SMS = 110, select slave trigger mode. Configure TIM1_SMCR register TS = 001, select ITR1 as input source.
4. Configure TIM1_CR1 register DIR=0, select the count direction as up count. Configure PSC=0, no division.

The counter count begins at the rising edge of ITR1 and sets TIF. The latency between the ETR signal rising edge and actual reset time of counter depends on the sync circuit of ETR input terminal.

TIM1_ARR=13 time sequence in external clock mode 2 + slave mode (trigger mode).

Figure 10-34 Control time sequence in external clock mode 2 +slave mode (trigger mode)

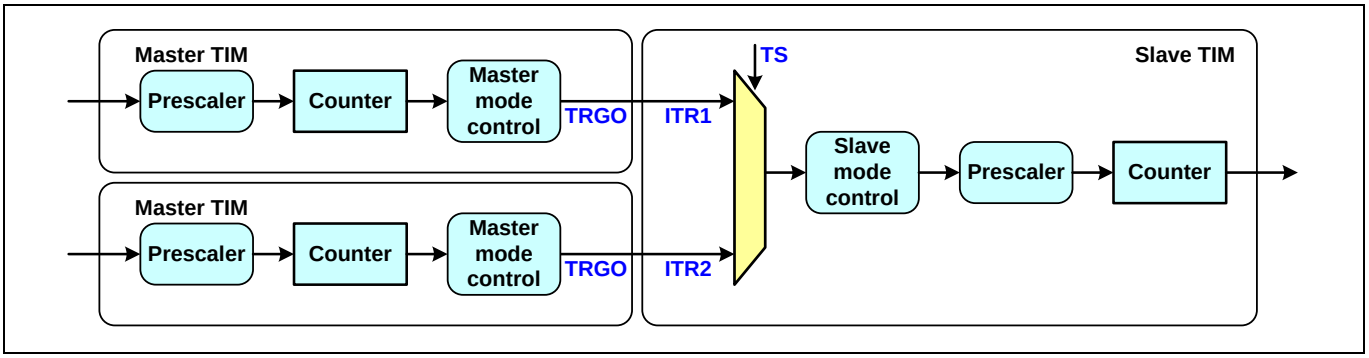


10.4.5 Timer synchronization

Different timers are internally connected, achieve timer cascading or synchronization.

Timer synchronization connectivity:

Figure 10-35 Timer connectivity



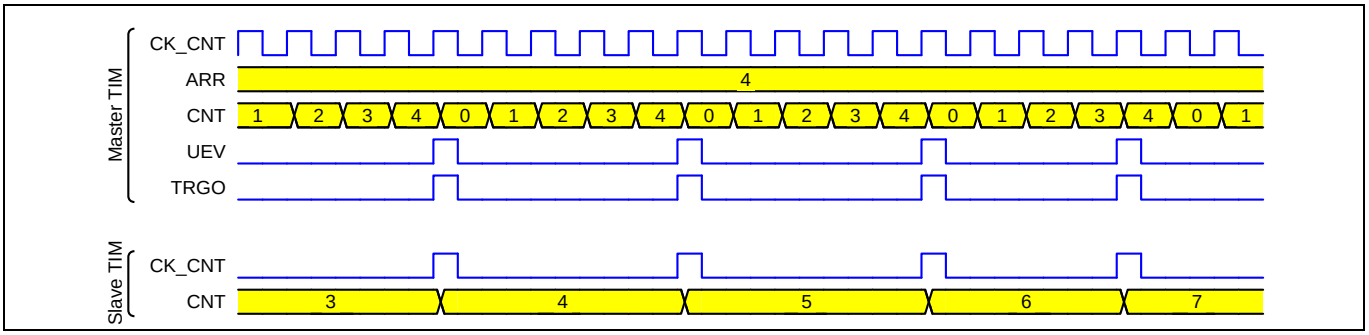
Several typical applications of connectivity.

Use one timer as the prescaler of another timer

For example, use master timer as slave timer prescaler, the time sequence diagram is shown above:

1. Configure master timer CR2 register MMS=010, master timer update event as trigger input (TRGO), master timer exports a cyclic signal at each update event.
2. Configure master timer ARR register, set master timer output period.
3. Configure slave timer SMCR register TS, select slave timer trigger source as master timer TRGO;
4. Configure slave timer SMCR register SMS=111, select slave external trigger mode 1.
5. Configure master timer CR1 register CEN=1, start master timer.
6. Configure slave timer CR1 register CEN=1, start slave timer.

Figure 10-36 Use master timer as slave timer prescaler



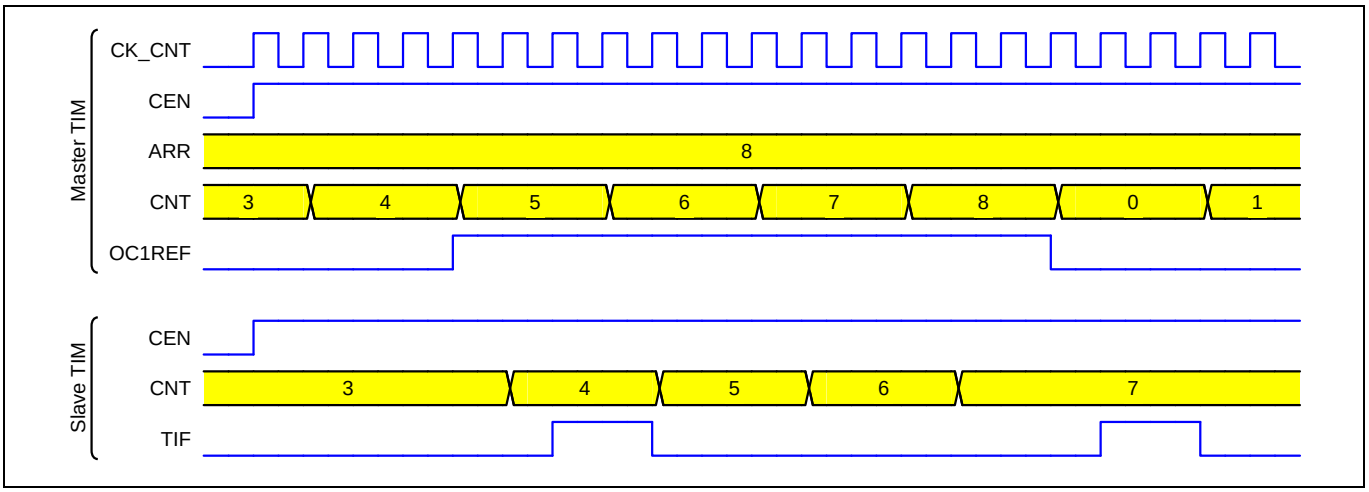
Use one timer to enable the other timer

For example: Use master timer to enable slave timer, and the time sequence diagram is shown below:

When master timer OC1REF is high, slave timer counter count begins. The clock frequency of both timers is provided by CK_CNT/3. Specific configuration:

1. Configure master timer CR2 register MMS=100, configure master timer output compare reference signal (OC1REF) as trigger output (TRGO).
2. Configure master timer CCR1 register, ARR register, CCMR1 register OC1M bit, configure the output waveform of master timer output signal TRGO.
3. Configure slave timer SMCR register TS, configure master timer OC1REF as slave timer trigger output.
4. Configure slave timer SMCR register SMS=101, configure slave timer as gate mode.
5. Configure slave timer CR1 register CEN=1, enable slave timer.
6. Configure master timer CR1 register CEN =1, enable master timer.

Figure 10-37 Use master timer to enable slave timer



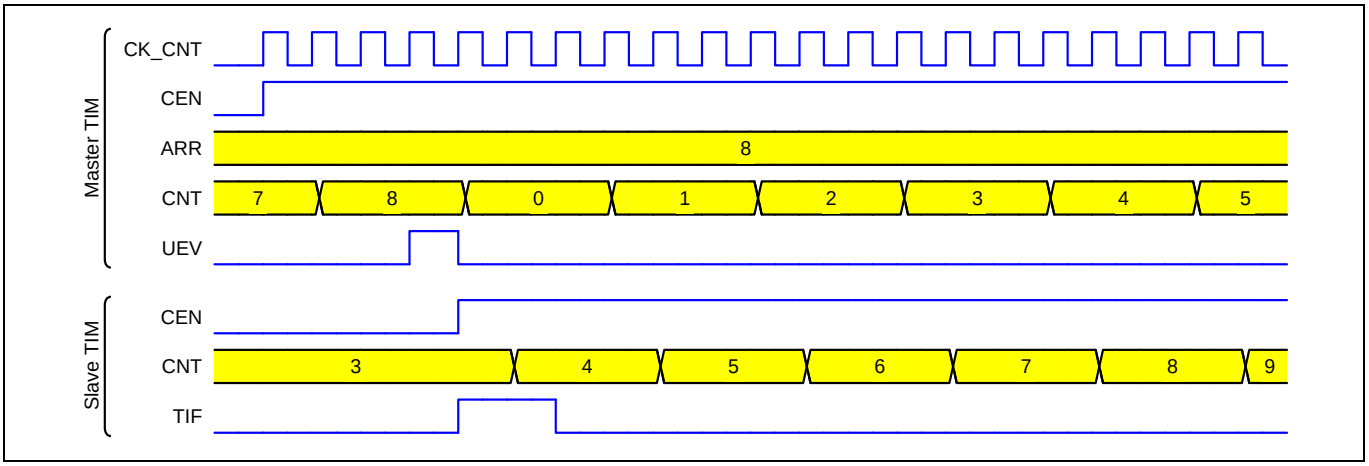
Use one timer to start another timer

For example: Use master timer update event to enable slave timer, and the time sequence is shown in the figure below:

When master timer generate the update event, slave timer receives the trigger signal, slave timer CEN is auto set by hardware. Slave timer counter count begins. The clock frequency of two timers is provided by CK_CNT/3. The specific configuration:

1. Configure master timer CR2 register MMS=010, select master timer update event as trigger output (TRGO);
2. Configure master timer ARR register, configure the period of update event;
3. Configure slave timer SMCR register TS, configure master timer TRGO as slave timer trigger input;
4. Configure slave timer SMCR register SMS=110, configure slave timer as trigger mode;
5. Configure master timer CR1 register CEN=1, enable master timer.

Figure 10-38 Use master timer update event to start slave timer



Use one external trigger to start two timers in step

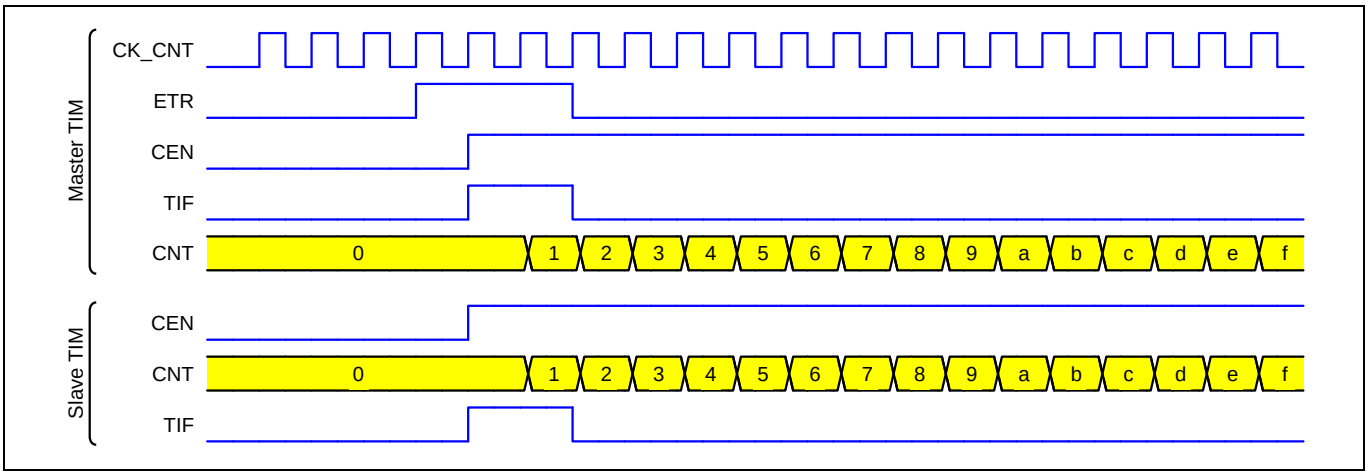
For example: Use master timer ETR rising edge start master timer and slave timer, the time sequence is shown in the figure below:

In order to ensure simultaneous start two timers, master timer must be configured in master/slave mode. Specific configuration:

1. Configure master timer CR2 register MMS=001, make master timer enable signal CEN as trigger output (TRGO).
2. Configure master timer SMCR register TS=111, make ETR as trigger input.
3. Configure master timer SMCR register SMS=110, configure master timer as trigger mode.
4. Configure slave timer SMCR register TS, configure master timer trigger output as slave timer trigger input.
5. Configure slave timer SMCR register SMS=110, configure slave timer as trigger mode.

When master timer ETR generates the rising edge, two timers start simultaneously (according to internal clock). The counter count begins, and two TIF flags are set.

Figure 10-39 master timer TI1 start master and slave timer in step



10.4.6 Debug mode

In the debug mode, configure DBG_CR register DBG_TIM1_STOP=1, TIM1 counter stops counting. (Details in the debug section)

10.4.7 Interrupt

TIM1 interrupt includes: compare 1 interrupt, compare 2 interrupt, compare 3 interrupt, compare 4 interrupt, compare 5 interrupt, update interrupt, COM interrupt, trigger interrupt and break interrupt. When the corresponding interrupt enable bit is set, generate the corresponding event and interrupt.

Table 10-6 List of interrupt events

Interrupt event	Flag bit	Enable bit
Compare 1 interrupt	CC1IF	CC1IE
Compare 2 interrupt	CC2IF	CC2IE
Compare 3 interrupt	CC3IF	CC3IE
Compare 4 interrupt	CC4IF	CC4IE
Compare 5 interrupt	CC5IF	CC5IE
Update interrupt	UIF	UIE
COM interrupt	COMIF	COMIE
Trigger interrupt	TIF	TIE
Break interrupt	BIF	BIE

10.4.8 DMA

TIM1 can generate multiple requests in case of a single event. The main objective is to read multiple registers based on the period by reprogramming some registers of TIM1 without software.

TIM1_DCR and TIM1_DMAR registers are relevant to DMA mode. The objective of DMA counter is unique and must point to the TIM1_DMAR register. After enabling DMA and at the specified TIM1 event, TIM1 will send requests to DMA. Each write to TIM1_DMAR register will be redirected to one TIM1 register.

TIM1_DCR register DBL bit has defined DMA's continuous transmission length, namely the transmission register number. In case of reading and writing TIM1_DMAR, the timer identifies DBL and determines the number of registers to be transmitted. TIM1_DCR register DBA bit has defined the DMA transmission base address, and the beginning offset of TIM1_CR1 register address (00000 for TIM1_CR1, 00001 for TIM1_CR2...00110 for TIM1_CCMR1, etc.)

For example: DMA continuous transmission mode is used to update the content of CCR1, CCR2 and CCR3 register in case of update event. Specific configuration:

1. Configure the corresponding DMA channel.
2. Configure TIM1_DCR register DBA=01101, configure DMA base address, select the register address with offset address TIM1_CCR1.
3. Configure TIM1_DCR register DBA=00010, configure transmission length 3.
4. Configure TIM1_DCR register UDE=1, enable update event DMA request.
5. Configure TIM1_CR1 register CEN=1, start the counter.
6. Enable DMA channel.

In this example, in case of an update event, DMA will transmit the ready data of the corresponding memory address to CCR1, CCR2 and CCR3 register or transmit the value of CCR1, CCR2 and CCR3 register to the corresponding memory address.

10.5 Register

Table 10-7 TIM1 register overview

Offset	Acronym	Register Name	Reset
0x00	TIM1_CR1	Control Register 1	0x0000
0x04	TIM1_CR2	Control Register 2	0x0000
0x08	TIM1_SMCR	Slave Mode Control Register	0x0000
0x0C	TIM1_DIER	DMA/Interrupt Enable Register (DMA is only suitable for the chip with built in DMA)	0x0000 0000
0x10	TIM1_SR	Status Register	0x0000 0000
0x14	TIM1_EGR	Event Generation Register	0x0000 0000
0x18	TIM1_CCMR1	Compare Mode Register 1	0x0000
0x1C	TIM1_CCMR2	Compare Mode Register 2	0x0000

Offset	Acronym	Register Name	Reset
0x20	TIM1_CCER	Compare Enable Register	0x0000
0x24	TIM1_CNT	Counter	0x0000
0x28	TIM1_PSC	Prescaler	0x0000
0x2C	TIM1_ARR	Auto Reload Register	0x0000
0x30	TIM1_RCR	Repeat Count Register	0x0000
0x34	TIM1_CCR1	Compare Register 1	0x0000
0x38	TIM1_CCR2	Compare Register 2	0x0000
0x3C	TIM1_CCR3	Compare Register 3	0x0000
0x40	TIM1_CCR4	Compare Register 4	0x0000
0x44	TIM1_BDTR	Break And Dead-Time Register	0x0000 0000
0x48	TIM1_DCR	DMA Control Register (only suitable for the chip with built in DMA)	0x0000
0x4C	TIM1_DMAR	DMA Address Register Of Continuous Mode (only suitable for the chip with built in DMA)	0x0000
0x54	TIM1_CCMR3	Compare Mode Register 3	0x0000
0x58	TIM1_CCR5	Compare Register 5	0x0000
0x5C	TIM1_PDER	PWM Phase Shift/DMA Repeat Update Request Enable Register	0x0000
0x60~0x70	TIM1_CCRxFALL	PWM Phase Shift Count Down Compare Register	0x0000
0x74	TIM1_BKINF	Break Input Filter Register	0x0000 0000

10.5.1 TIM1_CR1 Control Register 1

Address offset: 0x00
Reset value: 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved						CKD		ARPE	CMS		DIR	OPM	URS	UDIS	CEN
						rw		rw	rw		rw	rw	rw	rw	rw

Bit	Field	Description
15：10	Reserved	Reserved, must be kept at reset value.
9：8	CKD	Clock division Division ratio between the timer clock (CK_INT) frequency and the dead-time and sampling clock used by the dead-time generators and the digital filters (ETR, TIx). 00：$t_{DTS} = t_{INT\_CK}$ 01：$t_{DTS} = 2x\ t_{INT\_CK}$ 10：$t_{DTS} = 4x\ t_{INT\_CK}$ 11：Reserved, do not program this value
7	ARPE	Auto reload preload enable 0: Disable the shadow register of TIM1_ARR register 1: Enable the shadow register of TIM1_ARR register
6：5	CMS	Center alignment mode selection 00: Edge alignment mode. Count direction depends on DIR bit 01: Central alignment mode 1. The counter alternatively conducts up and down count. The channel is in output mode. Only in down count, compare interrupt flag bit is set 10: Central alignment mode 2. The counter alternatively conducts up and down count. The channel is in output mode. Only in up count, compare interrupt flag bit is set 11: Central alignment mode 3. The counter alternatively conducts up and down count. The channel is in output mode. In up and down count, compare interrupt flag bit is set Note: During count, the alignment mode change is disabled.

Bit	Field	Description
4	DIR	Count direction 0: up count 1: down count Note: When the counter is configured as central alignment mode, the bit is read only.
3	OPM	one-pulse mode 0: Disable one-pulse mode. In case of update event, the counter count continues 1: Enable one-pulse mode. In case of the next update event (clear CEN bit), the counter count stops
2	URS	Update request source This bit is set and cleared by software, select update event source. 0: The event below may generate a update interrupt or DMA request: - Counter overflow/underflow - Set UG bit - Update generation through the slave mode controller 1: Only in counter overflow/underflow, generate update interrupt or DMA request
1	UDIS	Update disables This bit is used to enable or disable the update event 0: Update event (UEV) enabled. 1: Update event disabled. The Update event is not generated, shadow registers keep their value(ARR, PSC, CCRx). However, the counter and the prescaler are reinitialized if the EGR.UG bit is set, the counter is reinitialized if a hardware reset is received from the slave mode controller.
0	CEN	Counter enable 0: Counter disabled 1: Counter enabled Note: External clock, gated mode can work only if the CEN bit has been previously set by software. However trigger mode can set the CEN bit automatically by hardware.

10.5.2 TIM1_CR2 Control Register 2

Address offset:0x04
Reset value: 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OIS4N	OIS4	OIS3N	OIS3	OIS2N	OIS2	OIS1N	OIS1	Res.	MMS			CCDS	CCUS	Res.	CCPC
rw	rw	rw	rw	rw	rw	rw	rw		rw			rw	rw		rw

Bit	Field	Description
15	OIS4N	Output idle state 4 (OC4N output). Refer to OIS1N bit.
14	OIS4	Output idle state 4 (OC4 output). Refer to OIS1 bit.
13	OIS3N	Output idle state 3 (OC3N output). Refer to OIS1N bit.
12	OIS3	Output idle state 3 (OC3 output). Refer to OIS1 bit.
11	OIS2N	Output idle state 2 (OC2N output). Refer to OIS1N bit.
10	OIS2	Output idle state 2 (OC2 output). Refer to OIS1 bit.
9	OIS1N	(Output idle state 1) (OC1N output) 0: In case of MOE =0, OC1N =0 after dead-time 1: In case of MOE =0, OC1N =1 after dead-time Note: After setting LOCK (TIM1_BKR register) level 1, 2 or 3, this bit cannot be changed.
8	OIS1	(Output idle state 1) (OC1 output) 0: In case of MOE=0, if OC1N is implemented, OC1=0 after dead-time 1: In case of MOE=0, if OC1N is implemented, OC1 = 1 after dead-time Note: After setting LOCK (TIM1_BKR register) level 1, 2 or 3, this bit cannot be changed.
7	Reserved	Reserved, must be kept at reset value.

Bit	Field	Description
6：4	MMS	Master mode selection These bits control TRGO signal selection, used to select the sync information sent to the slave timers in master mode: 000: Reset TIM1_EGR register UG bit generate one pulse trigger output (TRGO). 001: Enable The Counter Enable signal CNT_EN is used as trigger output (TRGO). It is useful to start several timers at the same time or to control a window in which a slave timer is enable. The Counter Enable signal is generated by a logic OR between CEN control bit and the trigger input when configured in gated mode. When the Counter Enable signal is controlled by the trigger input, there is a delay on TRGO, except if the master/slave mode is selected. 010: Update Update event is selected as TRGO. 011: Compare When a compare match occurred, send a positive pulse as TRGO. 100: Compare OC1REF signal is used as trigger output (TRGO) 101: Compare OC2REF signal is used as trigger output (TRGO) 110: Compare OC3REF signal is used as trigger output (TRGO) 111: Compare OC4REF signal is used as trigger output (TRGO)
3	CCDS	DMA request source selection 0: when CCx event occurs, send CCx DMA request 1: when update event occurs, send CCx DMA request Note: It's only suitable for the product with the built in DMA
2	CCUS	Compare control update selection 0: CCPC=1, they are updated by setting the COMG=1 only. 1：CCPC=1, they are updated by setting the COMG=1 or when a rising edge occurs on TRGI. Note: This bit acts only on channels that have a complementary output.
1	Reserved	Reserved, must be kept at reset value.
0	CCPC	Compare preloaded control bit 0: CCxE, CCxNE and OCxM bit preload disable 1: CCxE, CCxNE and OCxM bit preload enable Note: This bit acts only on channels that have a complementary output.

10.5.3 TIM1_SMCR Slave Mode Control Register

Address offset: 0x08
Reset value: 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ETP	ECE	ETPS		ETF				MSM	TS			OCCS	SMS		
rw	rw	rw		rw				rw	rw			rw	rw		

Bit	Field	Description
15	ETP	External trigger polarity The bit selects ETR signal polarity. 0: High level or rising edge is active 1: Low level or falling edge is active Note: It's only suitable for supporting the external trigger product
14	ECE	External clocks enable The bit enables external clock mode 2. 0: external clock mode 2 disabled 1: external clock mode 2 enabled, the counter is clocked by any active edge on the ETRFsignal. Note 1: It's only suitable for supporting the external trigger product. Note 2: Configuring ECE =1 has the same effect as configuring SMS =111 and TS =111. Note 3: TS≠111, reset mode, gate mode and trigger mode can be used together with the external clock mode 2. Note 4: If external clock mode 1 and external clock mode 2 are enabled at the same time, the external clock input is ETRF.

Bit	Field	Description
13：12	ETPS	External trigger prescaler The external trigger signal ETRP frequency must be lower than 1/4 of TIM1PCLK frequency. When inputting the quicker external clock, use the prescaler to reduce ETRP frequency. 00: Prescaler OFF 01: ETRP frequency divided by 2 10: ETRP frequency divided by 4 11: ETRP frequency divided by 8 Note: It's only suitable for supporting the external trigger product
11：8	ETF	External trigger filter These bits define ETRP signal sampling frequency and ETRP digital filter bandwidth. In fact, the digital filter is an event counter, which generate an output jump after N event. 0000: No filter, f_{DTS} sampling 001: Sampling frequency $f_{sampling}=f_{INT\_CK}$, N =2 0010: Sampling frequency $f_{sampling}=f_{INT\_CK}$, N =4 0011: Sampling frequency $f_{sampling}=f_{INT\_CK}$, N =8 0100: Sampling frequency $f_{sampling}=f_{DTS/2}$, N =6 0101: Sampling frequency $f_{sampling}=f_{DTS/2}$, N =8 0110: Sampling frequency $f_{sampling}=f_{DTS/4}$, N =6 0101: Sampling frequency $f_{sampling}=f_{DTS/2}$, N =8 0110: Sampling frequency $f_{sampling}=f_{DTS/4}$, N =6 0111: Sampling frequency $f_{sampling}=f_{DTS/4}$, N =8 1000: Sampling frequency $f_{sampling}=f_{DTS/8}$, N =6 1001: Sampling frequency $f_{sampling}=f_{DTS/8}$, N =8 1010: Sampling frequency $f_{sampling}=f_{DTS/16}$, N =5 1011: Sampling frequency $f_{sampling}=f_{DTS/16}$, N =6 1100: Sampling frequency $f_{sampling}=f_{DTS/16}$, N =8 1101: Sampling frequency $f_{sampling}=f_{DTS/32}$, N =5 1110: Sampling frequency $f_{sampling}=f_{DTS/32}$, N =6 1111: Sampling frequency $f_{sampling}=f_{DTS/32}$, N =8 Note: It's only suitable for supporting the external trigger product
7	MSM	Master/slave mode 0: No action 1: The effect of an event on the trigger input (TRGI) is delayed allowing a perfect synchronization between the current timer and its slaves (through TRGO). It is useful if we want to synchronize several timers on a single external event.
6：4	TS	Trigger selection Trigger input source selection. 000：Internal trigger 0 (ITR0) 001：Internal trigger 1 (ITR1) 010：Internal trigger 2 (ITR2) 011：Internal trigger 3 (ITR3) 100：Reserved 101：Reserved 110：Reserved 111：External trigger input (ETR) Note: After the slave mode enable, these bits cannot be changed
3	OCCS	Timer compare output signal (OCxREF) clear selection In PWM mode, clear the compare output signal (OCxREF) 0: External trigger signal as clear signal 1: Comparator output (COMP) as clear signal Note: It's only applicable to products that support external trigger or have a built-in comparator (COMP)

Bit	Field	Description
2：0	SMS	Slave mode selection When external signals are selected the active edge of the trigger signal (TRGI) is linked to the polarity selected on the external input. 000: Close slave mode - In case of CEN =1, the prescaler is directly driven by the internal clock. 001: Reserved 010: Reserved 011: Reserved 100: Reset mode -The rising edge of the selected trigger input (TRGI) reinitialize the counter and generate an update event. 101: Gate mode - When the trigger input (TRGI) is high, the counter count begins and generate an update event. When the trigger input turns low, the counter count stops (but without reset) and generates an update event. The counter start and stop are controlled. 110: Trigger mode -The counter starts in the rising edge of the trigger input TRGI (but without reset) and generate an update event. Only the counter start is controlled. 111: External clock mode 1 -The rising edge of the selected trigger input (TRGI) drives the counter and generate an update event.

Table 10-8 TIMx internal trigger connection

Slave timer	ITR0	ITR1	ITR2	ITR3
TIM1	-	TIM2_TRGO	TIM14_OC1REF	TIM13_OC1REF
TIM2	TIM1_TRGO	-	TIM14_OC1REF	TIM13_OC1REF

10.5.4 TIM1_DIER DMA/Interrupt Enable Register

Address offset: 0x0C
Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved														CC5DE	CC5IE
														rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	TDE	COMDE	CC4DE	CC3DE	CC2DE	CC1DE	UDE	BIE	TIE	COMIE	CC4IE	CC3IE	CC2IE	CC1IE	UIE
	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bit	Field	Description
31：18	Reserved	Reserved, must be kept at reset value.
17	CC5DE	Compare 5 DMA requests enable 0: Compare 5 interrupt disable 1: Compare 5 interrupt enable Note: It's only suitable for the product the built in DMA and channel 5.
16	CC5IE	Compare 5 interrupt enable 0: Compare 5 interrupt disable 1: Compare 5 interrupt enable
15	Reserved	Reserved, must be kept at reset value.
14	TDE	Trigger DMA request enable 0: Trigger DMA request disable 1: Trigger DMA request enable Note: It's only suitable for the built in DMA product.
13	COMDE	COM DMA request enable 0: COM DMA request disable 1: COM DMA request enable Note: It's only suitable for the built in DMA product.
12	CC4DE	Compare 4 DMA request enable 0: Compare 4 DMA request disable 1: Compare 4 DMA request enable Note: It's only suitable for the built in DMA product.

Bit	Field	Description
11	CC3DE	Compare 3 DMA requests enable 0: Compare 3 DMA request disable 1: Compare 3 DMA requests enable Note: It's only suitable for the built in DMA product.
10	CC2DE	Compare 2 DMA request enable 0: Compare 2 DMA request disable 1: Compare 2 DMA request enable Note: It's only suitable for the built in DMA product.
9	CC1DE	Compare 1 DMA request enable 0: Compare 1 DMA request disable 1: Compare 1 DMA request enable Note: It's only suitable for the built in DMA product.
8	UDE	Update DMA request enable 0: Update DMA request disable 1: Update DMA request enable Note: It's only suitable for the built in DMA product.
7	BIE	Break interrupts enable 0: Break interrupts disable 1: Break interrupts enable
6	TIE	Trigger interrupts enable 0: Break interrupts disable 1: Break interrupts enable
5	COMIE	Enable COM interrupt 0: COM interrupts disable 1: COM interrupts enable
4	CC4IE	Enable compare 4 interrupts 0: Compare interrupt 4 disable 1: Compare interrupt 4 enable
3	CC3IE	Enable compare 3 interrupts 0: Compare interrupt 3 disable 1: Compare interrupt 3 enable
2	CC2IE	Enable compare 2 interrupts 0: Compare interrupt 2 disable 1: Compare interrupt 2 enable
1	CC1IE	Enable compare 1 interrupt 0: Compare interrupt 1 disable 1: Compare interrupt 1 enable
0	UIE	Enable update interrupt 0: Update event interrupt disable 1: Update event interrupt enable

10.5.5 TIM1_SR Status Register

Address offset: 0x10
Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															CC5IF
															rw0c
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								BIF	TIF	COMIF	CC4IF	CC3IF	CC2IF	CC1IF	UIF
								r_w0c							

Bit	Field	Description
31：17	Reserved	Reserved, must be kept at reset value.
16	CC5IF	Compare 5 interrupt flags Refer to CC1IF description.
15：8	Reserved	Reserved, must be kept at reset value.
7	BIF	Break interrupt flag When the break input is active, the bit is set by hardware. If the Break input is inactive, the bit is set as 0 by software 0: No break event occurred 1: Detect active level on Break input

Bit	Field	Description
6	TIF	Trigger interrupt flag In case of trigger event (When the slave mode counter is in any other mode except for the gate mode, detect the active edge in TRGI input terminal, or detect any edge in gate mode), the bit is set by hardware. It's cleared by software. 0: No trigger event occurred 1: Trigger interrupt pending
5	COMIF	COM interrupt flag In case of COM event (compare control bit: CCxE, CCxNE, OCxM updated), the bit is set by hardware. It is cleared by software. 0：No COM event occurred 1: COM interrupt pending
4	CC4IF	Compare 4 interrupt flag Refer to CC1IF description
3	CC3IF	Compare 3 interrupt flag Refer to CC1IF description
2	CC2IF	Compare 2 interrupt flag Refer to CC1IF description.
1	CC1IF	Compare 1 interrupt flag When the counter value and compare value match, the bit is set by hardware except for the central alignment mode (the bit is set in the central alignment mode according to TIM1_CR1.CMS[1:0]). It's cleared by software. 0: No match 1: TIM1_CNT value and TIM1_CCR1 value match
0	UIF	Update interrupt flag In case of update event, the bit is set by hardware. It's cleared by software. 0: No update interrupt occurred 1: update interrupt pending This bit is set by hardware when the registers are updated: - In case of TIM1_CR1 register UDIS=0 and REP_CNT=0, when the counter generates overflow/underflow. -In case of TIM1_CR1 register UDIS=0, URS=0, when TIM1_EGR register UG =1. - In case of TIM1_CR1 register UDIS=0, URS=0, when update generation through the slave mode controller.

10.5.6 TIM1_EGR Event Generation Register

Address offset: 0x14
Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															CC5G
															w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								BG	TG	COMG	CC4G	CC3G	CC2G	CC1G	UG
								w	w	w	w	w	w	w	w

Bit	Field	Description
31：17	Reserved	Reserved, must be kept at reset value.
16	CC5G	Compare 5 generation Refer to CC1G description.
15：8	Reserved	Reserved, must be kept at reset value.
7	BG	Break generation 0: No action 1: generate a break event. MOE =0, BIF =1; when producing the corresponding interrupt and DMA, generate the corresponding interrupt and DMA. It's cleared by hardware.
6	TG	Trigger generation 0: No action 1: generate a trigger event. TIM1_SR register TIF =1. When enabling the corresponding interrupt and DMA, generate the corresponding interrupt and DMA. It's auto cleared by hardware.

Bit	Field	Description
5	COMG	Compare control update generation 0: No action 1: Compare event control update. It's auto cleared by hardware. In case of CCPC =1, enable update CCxE, CCxNE and OCxM bit. Note: This bit is only active for the complementary output channel.
4	CC4G	Compare 4 generation Refer to CC1G description.
3	CC3G	Compare 3 generation Refer to CC1G description.
2	CC2G	Compare 2 generation Refer to CC1G description.
1	CC1G	Compare 1 generation Refer to CC1G description. This bit is set by software. It generates a compare event, and is auto cleared by hardware. 0: No action 1：Generate a compare event in channel CC1: Set CC1IF =1. When enabling the corresponding interrupt and DMA, generate the corresponding interrupt and DMA.
0	UG	Update event generation 0：No action 1: Initialize the counter, and generate an update event. It's auto cleared by hardware. When selecting the central alignment or up counting mode, the counter is clear; otherwise (down counting mode) the counter auto reload value is loaded. The prescaler counter is cleared at the same time.

10.5.7 TIM1_CCMR1 Compare Mode Register 1

Address offset: 0x18
Reset value: 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OC2CE	OC2M			OC2PE	OC2FE	Reserved		OC1CE	OC1M			OC1PE	OC1FE	Reserved	
rw	rw			rw	rw			rw	rw			rw	rw		

Bit	Field	Description
15	OC2CE	Channel 2 output and compare clear enable Refer to OC1CE description.
14：12	OC2M	Channel 2 output and compare mode Refer to OC1M description.
11	OC2PE	Channel 2 output and compare preload enable Refer to OC1PE description.
10	OC2FE	Channel 2 output and compare quick enable Refer to OC1FE description.
9：8	Reserved	Reserved, must be kept at reset value.
7	OC1CE	Channel 1 compare clear enable 0: OC1REF is not affected by ETR 1: When detecting ETR as active level, clear OC1REF

Bit	Field	Description
6：4	OC1M	Channel 1 output compare mode The bit has defined the output reference signal OC1REF action. OC1REF has determined OC1 and OC1N value. OC1REF is active at high level. The active level of OC1, OC1N depends on CC1P and CC1NP bit. 000：Freeze. TIM1_CCR1 and TIM1_CNT compare results has no effect on OC1REF. 001: Set as high when configuration. When TIM1_CNT value and TIM1_CCR1 value are same, enforce OC1REF as high level 010: Set as low when configuration. When TIM1_CNT value and TIM1_CCR1 value are same, enforce OC1REF as low level 011: Toggle when match. When TIM1_CCR1=TIM1_CNT, OC1REF toggle. 100: Enforce as low. Enforce OC1REF at low level 101: Enforce as high. Enforce OC1REF at high level 110：PWM mode 1. During up count, in case of TIM1_CNT<TIM1_CCR1, enforce OC1REF is at high level. Or else, it's at low level. During down count, in case of TIM1_CNT > TIM1_CCR1, enforce OC1REF is at low level. Or else, it's at high level. 111：PWM mode 2. During up count, in case of TIM1_CNT<TIM1_CCR1, channel 1 enforce OC1REF is at low level. Or else, it's at high level. During down count, in case of TIM1_CNT > TIM1_CCR1, enforce OC1REF is at high level. Or else, it's at low level. Note 1: In case of LOCK level 3 (TIM1_BDTR register LOCK bit), the bit cannot be changed. Note 2: In PWM mode 1 or PWM mode 2, only when the compare result changes or it changes over from the freeze mode to PWM mode in the output compare mode, OC1REF level may change.
3	OC1PE	Channel 1 output compare preload enable 0: Disable TIM1_CCR1 register preload function. The value written into TIM1_CCR1 register becomes active immediately 1: Enable TIM1_CCR1 register preload function. Only read and write the preload register. TIM1_CCR1 preload value becomes active at the update event Note 1: When the LOCK level is 3 (TIM1_BDTR register LOCK bit), the bit cannot be changed. Note 2: Only in the one-pulse mode (TIM1_CR1 register OPM= 1), it has no influence whether the preload register is set. Under other scenarios, it's required to set the preload register. Otherwise, the follow up action is not certain.
2	OC1FE	Channel 1 output compare quick enable In case of the bit is set 1, when the channel is configured as PWM mode, it speeds up response of the compare output to the trigger time. The output channel deems the active edge of the trigger input signal as one compare match. Therefore, OC is set as compare level, but is irrelevant to the compare result. 0: channel 1 output compare fast enable. 1: channel 1 output compare fast disable.
1：0	Reserved	Reserved, must be kept at reset value.

10.5.8 TIM1_CCMR2 Compare Mode Register 2

Address offset: 0x1C
Reset value: 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OC4CE	OC4M			OC4PE	OC4FE	Reserved		OC3CE	OC3M			OC3PE	OC3FE	Reserved	
rw	rw			rw	rw			rw	rw			rw	rw		

Bit	Field	Description
15	OC4CE	Channel 4 output compare clear enable Refer to OC3CE description
14：12	OC4M	Channel 4 output compare mode Refer to OC3M description
11	OC4PE	Channel 4 output compare preload enable Refer to OC3PE description
10	OC4FE	Channel 4 output compare rapid enable Refer to OC3FE description
9：8	Reserved	Reserved, must be kept at reset value.
7	OC3CE	Channel 3 output compare clear enable 0: OC3REF is not affected by ETR input 1: When detecting ETR input is at active level, clear OC3REF

Bit	Field	Description
6：4	OC3M	Channel 3 output compare mode The bit has defined the output reference signal OC3REF action. OC3REF has determined OC3 and OC3N value. OC3REF is active at high level. The active level of OC3, OC3N depends on CC3P and CC3NP bit. 000：Freeze. TIM1_CCR3 and TIM1_CNT compare results has no effect on OC3REF. 001: Set as high when configuration. When TIM1_CNT value and TIM1_CCR3 value are same, enforce OC3REF as high level 010: Set as low when configuration. When TIM1_CNT value and TIM1_CCR3 value are same, enforce OC3REF as low level 011: Toggle when match. When TIM1_CCR3=TIM1_CNT, OC3REF toggle. 100: Enforce as low. Enforce OC3REF at low level 101: Enforce as high. Enforce OC3REF at high level 110: PWM mode 1. During up count, in case of TIM1_CNT<TIM1_CCR3, enforce OC3REF as high level, otherwise its low level; during down count, in case of TIM1_CNT > TIM1_CCR3, enforce OC3REF as low level, otherwise its high level. 111: PWM mode 2. During up count, in case of TIM1_CNT<TIM1_CCR3, enforce OC3REF as low level, otherwise its high level; during down count, in case of TIM1_CNT > TIM1_CCR3, enforce OC3REF as high level, otherwise its low level. Note 1: In case of LOCK level 3 (TIM1_BDTR register LOCK bit), the bit cannot be changed. Note 2: In PWM mode 1 or PWM mode 2, only when the compare result changes or it changes over from the freeze mode to PWM mode in the output compare mode, OC3REF level may change.
3	OC3PE	Channel 3 output compare preload enable 0: Disable TIM1_CCR3 register preload function, the value written into TIM1_CCR3 register becomes active immediately 1: Enable TIM1_CCR3 register preload function. The read and write only works on the preload register. TIM1_CCR3 preload value becomes active when the update event is available Note 1: When the LOCK level is set 3 (TIM1_BDTR register LOCK bit), the bit cannot be changed. Note 2: Only in the one-pulse mode (TIM1_CR1 register OPM= 1), it's not required to set the preload register. Under other scenarios, it's required to set the preload register. Otherwise, the follow up action is not certain.
2	OC3FE	Channel 3 output compare quick enable In case of the bit 1, when the channel is configured as PWM mode, it speeds up response of the compare output to the trigger time. The output channel deems the active edge of the trigger input signal as one compare match. Therefore, OC is set as compare level, but is irrelevant to the compare result. 0: channel 3 output compare fast disable. 1: channel 3 output compare fast enable.
1：0	Reserved	Reserved, must be kept at reset value.

10.5.9 TIM1_CCER Compare Enable Register

Address offset: 0x20
Reset value: 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CC4NP	CC4NE	CC 4P	CC 4E	CC3NP	CC3NE	CC3P	CC3E	CC2NP	CC2NE	CC2P	CC2E	CC1NP	CC1NE	CC1P	CC1E
rw															

Bit	Field	Description
15	CC4NP	Channel 4 complementary output polarity Refer to CC1NP description
14	CC4NE	Channel 4 complementary output enable Refer to CC1NE description
13	CC4P	Channel 4 compare output polarity Refer to CC1P description.
12	CC4E	Channel 4 compare enable Refer to CC1E description
11	CC3NP	Channel 3 complementary output polarity Refer to CC1NP description
10	CC3NE	Channel 3 complementary output enable Refer to CC1NE description
9	CC3P	Channel 3 output polarity Refer to CC1P description

Bit	Field	Description
8	CC3E	Channel 3 output enable Refer to CC1E description
7	CC2NP	Channel 2 complementary output polarity Refer to CC1NP description
6	CC2NE	Channel 2 complementary output enable Refer to CC1NE description
5	CC2P	Channel 2 output polarity Refer to CC1P description
4	CC2E	Channel 2 output enable Refer to CC1E description
3	CC1NP	Channel 1 complementary output polarity This bit has defined the input signal polarity: 0: OC1N is active at high level 1: OC1N is active at low level Note: When LOCK level (TIM1_BDTR register LCCK bit) is set 3 or 2, the bit cannot be changed.
2	CC1NE	Channel 1 complementary output enable 0: Close channel 1 complementary output. OC1N output disable. 1: OC1N signal exports to the corresponding output pin. The output level relies on the values in MOE, OSSI, OSSR, OIS1, OIS1N and CC1E.
1	CC1P	Channel 1 output polarity This bit defines the output signal polarity: 0: OC1 is active at high level 1: OC1 is active at low level Note: When LOCK level (TIM1_BDTR register LCCK) is set 3 or 2, this bit cannot be changed.
0	CC1E	Channel 1 output enable 0: Close. OC1 output disable 1: Enable. The output level relies on the values in MOE, OSSI, OSSR, OIS1, OIS1N and CC1NE.

10.5.10 TIM1_CNT Counter

Address offset: 0x24
Reset value: 0x0000

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

Bit	Field	Description
15：0	CNT	Counter value

10.5.11 TIM1_PSC Prescaler

Address offset: 0x28
Reset value: 0x0000

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

Bit	Field	Description
15：0	PSC	Prescaler value Counter clock frequency (ck_cnt) =$f_{CK\_PSC/（PSC+1）}$ In case of update event, PSC value is loaded into the current prescaler register.

10.5.12 TIM1_ARR Auto Reload Register

Address offset: 0x2C
Reset value: 0x0000

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

Bit	Field	Description
15：0	ARR	Auto reload value These bits define the auto reload value of the counter. When auto reload value is 0, the counter doesn't work.

10.5.13 TIM1_RCR Repeat Count Register

Address offset: 0x30
Reset value: 0x0000

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

Bit	Field	Description
15：8	REP_CNT	Real-time value written by the repeat counter In the repeat count mode, it's to change real-time phase move of the detection point of update interrupt flag bit (UIF) Note: After the update event, write these bits. Write REP_CNT before the update event will make the bit move inactive.
7：0	REP	Repeat counter value The repeat counter value defines the generation speed of the update event. When the repeat counter value reduces to 0, it generate the update event. If the update interrupt is enabled, it will simultaneously affect the update interrupt generation speed. The written REP value becomes active at the next update event. In PWM mode, (REP+1) corresponds to: In the edge alignment mode, the number of PWM period In the central alignment mode, the number of PWM half period

10.5.14 TIM1_CCR1 Compare Register 1

Address offset: 0x34
Reset value: 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR1															
rw															

Bit	Field	Description
15：0	CCR1	Channel 1 compare value If the preload function is not selected in TIM1_CCMR1 register (OC1PE bit), the write value is immediately transmitted to the current compare shadow register. Or else, the write value is only loaded into the compare register in case of update event. The current compare shadow register is involved in the compare with the counter TIM1_CNT, and the compare result is reflected to the output signal of OC1 port.

10.5.15 TIM1_CCR2 Compare Register 2

Address offset: 0x38
Reset value: 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR2															
rw															

Bit	Field	Description
15：0	CCR2	Channel 2 compare value Refer to CCR1 description.

10.5.16 TIM1_CCR3 Compare Register 3

Address offset: 0x3C
Reset value: 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR3															
rw															

Bit	Field	Description
15：0	CCR3	Channel 3 compare value Refer to CCR1 description.

10.5.17 TIM1_CCR4 Compare Register 4

Address offset: 0x40
Reset value: 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR4															
rw															

Bit	Field	Description
15：0	CCR4	Channel 4 compare value Refer to CCR1 description.

10.5.18 TIM1_BDTR Break and Dead-Time Register

Address offset: 0x44
Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															DOE
															rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MOE	AOE	BKP	BKE	OSSR	OSSI	LOCK		DTG							
rw	rw	rw	rw	rw	rw	rw		rw							

Note: According to the lock setting, DOE, AOE, BKP, BKE, OSSI, OSSR and DTG bits can be write protected; it's necessary to configure them during the first write into TIM1_BDTR register. Details are given in the chapter of complementary output and dead-time insert.

Bit	Field	Description
31：17	Reserved	Reserved, must be kept at reset value.

Bit	Field	Description
16	DOE	Direct output enable When the break is active, MOE is set 0, active. 0：After break output, wait for a dead time and export the idle state (output disable). 1：Immediately export the idle state (output disable). Note: When LOCK level 1 (TIM1_BDTR register LOCK bit) is set, the bit cannot be changed.
15	MOE	Main output enable When the channel x is configured as output, according to AOE bit setting value, the bit can be cleared or auto set by software. When the break input is active, the bit is cleared by hardware. 0：Disable OCx and OCxN output or enforce as the idle state (output disable). 1：When setting the corresponding enable bit (TIM1_CCER register CCxE, CCxNE), OCx and OCxN output enable
14	AOE	Auto output enable 0: MOE cannot be set by hardware 1: MOE can be set by software or auto set by hardware at the next update event when the break is inactive Note: When LOCK level 1 (TIM1_BDTR register LOCK bit) is set, the bit cannot be changed.
13	BKP	Break input polarity 0: Break input is active low 1: Break input is active high Note: When LOCK level 1 (TIM1_BDTR register LOCK bit) is set, the bit cannot be changed.
12	BKE	Break function enable 0：Break input disabled 1：Break input enabled Note 1: When LOCK level 1 (TIM1_BDTR register LOCK bit) is set, the bit cannot be changed. Note 2: Break input includes pin input and comparator compare result input. Before break function is enabled, configure BKIN_SEL bit in TIM1_BKINF register to select break source.
11	OSSR	Off state selection in the run mode This bit only applies in case of MOE =1 and the channel is in complementary output. 0: When the timer inactive, disable OC/OCN output 1: When the timer inactive, in case of CCxE = 1 or CCxNE = 1, first enable OC/OCN and export inactive level, and then set OC/OCN enable output signal Note: When LOCK level 2 (TIM1_BDTR register LOCK bit) is set, the bit cannot be changed.
10	OSSI	Off state selection in the idle mode This bit only applies in case of MOE =0 and the channel is in output. 0: When the timer inactive, disable OC/OCN output 1: When the timer inactive, in case of CCxE = 1 or CCxNE = 1, first OC/OCN exports inactive level, and then set OC/OCN enable output signal. Note: When LOCK level 2 (TIM1_BDTR register LOCK bit) is set, the bit cannot be changed.
9：8	LOCK	Lock configuration This bit defines the register write protection function. 00: Write protection function closes, the register doesn't have the write protection 01: Lock level 1, unable to write TIM1_BDTR register DOE, DTG, BKE, BKP and AOE bit and TIM1_CR2 register OISx/OISxN bit 10: Lock level 2, unable to write lock level 1 bit, or CC polarity level as well as OSSR/OSSI bit 11：Lock level 3, unable to write lock level 2 bit, or CC control bit Note: The LOCK bits can be written only once after the reset. Once the TIM1_BDTR register has been written, their content is frozen until the next reset.
7：0	DTG	Dead-time generator setup adjustment These bits define the dead-time duration of the complementary output. Note: When LOCK level (TIM1_BDTR register LOCK bit) is set 1, 2 or 3, the bit cannot be changed.

10.5.19 TIM1_DCR DMA Control Register

Address offset: 0x48

Reset value: 0x0000

Note: This register is only applicable for the built in DMA product. Details are given in the 10.4.8 DMA.

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

Bit	Field	Description
15：13	Reserved	Reserved, must be kept at reset value.
12：8	DBL	DMA continuous transmission length These bits define the DMA access register number in continuous mode 00000: 1 transmission 00001: 2 transmission 00010: 3 transmission .... 10001：18 transmission
7：5	Reserved	Reserved, must be kept at reset value.
4：0	DBA	DMA base address These bits define the first address of DMA access TIM1_DMAR register in continuous mode. DBA defines the offset value from the TIM1_CR1 register address： 00000：TIM1_CR1 00001：TIM1_CR2 00010：TIM1_SMCR .....

10.5.20 TIM1_DMAR DMA Address Register of Continuous Mode

Address offset: 0x4C

Reset value: 0x0000

Note: This register is only applicable for the built in DMA product. Details are given in the 10.4.8 DMA.

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

Bit	Field	Description
15：0	DMAB	DMA register for burst accesses DMA continuous transmission register's write to TIM1_DMAR register will lead to access of the register in the addresses below： TIM1_CR1 address + DBA + DMA index， Where TIM1_CR1 address is the address of TIM1_CR1 register; DBA is the base address defined by TIM1_DCR register; DMA index is the offset auto controlled by DMA. It depends on the DBL defined by TIM1_DCR register.

10.5.21 TIM1_CCMR3 Compare Mode Register 3

Address offset: 0x54

Reset value: 0x0000

The channel only applies to the output compare mode

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved												OC5PE	Reserved		
												rw			

Bit	Field	Description
15：4	Reserved	Reserved, must be kept at reset value.
3	OC5PE	Output compare 5 preload enable 0: Disable TIM1_CCR5 register preload function, and the value written into TIM1_CCR5 register will become effective immediately. 1: Enable TIM1_CCR5 register preload function. Only read and write the preload register. TIM1_CCR5 preload value becomes active at the update event Note 1: When LOCK level 3 is set (TIM1_BDTR register LOCK level), this bit cannot be changed.
2：0	Reserved	Reserved, must be kept at reset value.

10.5.22 TIM1_CCR5 Compare Register 5

Address offset: 0x58
Reset value: 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR5															
rw															

Bit	Field	Description
15：0	CCR5	Compare 5 value CC5 channel is only configured as output: If the preload function is not selected in the TIM1_CCMR3 register (OC5PE bit), the write value will immediately be transmitted to the corresponding compare shadow register. Or else, only in case of the update event, the preload value will be transmitted to the corresponding compare shadow register. The compare shadow register is involved in the counter TIM1_CNT compare. Because CC5 channel is the internal channel and cannot be transmitted to the pin, the compare result is used for the internal trigger event.

10.5.23 TIM1_PDER PWM Phase Shift/DMA Repeat Update Request Enable Register

Address offset: 0x5C
Reset value: 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved										CCR5_SHIFT_EN	CCR4_SHIFT_EN	CCR3_SHIFT_EN	CCR2_SHIFT_EN	CCR1_SHIFT_EN	CCDREPE
										rw	rw	rw	rw	rw	rw

Bit	Field	Description
15：6	Reserved	Reserved, must be kept at reset value.
5	CCR5_SHIFT_EN	Enable channel 5 output PWM phase shift bit 0: Channel 5 output PWM phase shift disable 1: Channel 5 output PWM phase shift enable Details given in CCRxFALL register phase shift operation
4	CCR4_SHIFT_EN	Enable channel 4 output PWM phase shift bit 0: Channel 4 output PWM phase shift disable 1: Channel 4 output PWM phase shift enable Details given in CCRxFALL register phase shift operation
3	CCR3_SHIFT_EN	Enable channel 3 output PWM phase shift bit 0: Channel 3 output PWM phase shift disable 1: Channel 3 output PWM phase shift enable Details given in CCRxFALL register phase shift operation
2	CCR2_SHIFT_EN	Enable channel 2 output PWM phase shift bit 0: Channel 2 output PWM phase shift disable 1: Channel 2 output PWM phase shift enable Details given in CCRxFALL register phase shift operation

Bit	Field	Description
1	CCR1_SHIFT_EN	Enable channel 1 output PWM phase shift bit 0: Channel 1 output PWM phase shift disable 1: Channel 1 output PWM phase shift enable Details given in CCRxFALL register phase shift operation
0	CCDREPE	Enable DMA to generate the update request at each underflow or overflow 0：DAM update request is generated according to the repeat count register value. 1：Enable DMA to generate the update request at each underflow or overflow

10.5.24 TIM1_CCRxFALL PWM Phase Shift Count Down Compare Register

Address offset: 0x60 ~ 0x70
Reset value: 0x0000

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

Bit	Field	Description
15：0	CCRxFALL	Channel x compare value during count down in PWM central alignment mode PWM phase shift function: Enable PDER register PWM phase shift. According to the required phase shift, configure CCRxFALL and CCRx, PWM exports programmable phase shift waveform, and can shift left or right.

10.5.25 TIM1_BKINF Break Input Filter Register

Address offset: 0x74
Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	COMPBKIN_SEL		IOBKIN_SEL							Res.	BKINF				BKINFE
	rw		rw								rw				rw

Bit	Field	Description
31：15	Reserved	Reserved, must be kept at reset value.
14：13	COMPBKIN_SEL	COMP break input channel selects COMPBKIN_SEL[1]: 0: COMP2_OUT break input disable 1: COMP2_OUT break input enable COMPBKIN_SEL[0]: 0: COMP1_OUT break input disable 1: COMP1_OUT break input enable

Bit	Field	Description
12：6	IOBKIN_SEL	IO break input channel selects IOBKIN_SEL[6]: 0: TIM1_BKIN7 break input disable 1: TIM1_BKIN7 break input enable IOBKIN_SEL[5]: 0: TIM1_BKIN6 break input disable 1: TIM1_BKIN6 break input enable IOBKIN_SEL[4]: 0: TIM1_BKIN5 break input disable 1: TIM1_BKIN5 break input enable IOBKIN_SEL[3]: 0: TIM1_BKIN4 break input disable 1: TIM1_BKIN4 break input enable IOBKIN_SEL[2]: 0: TIM1_BKIN3 break input disable 1: TIM1_BKIN3 break input enable IOBKIN_SEL[1]: 0: TIM1_BKIN2 break input disable 1: TIM1_BKIN2 break input enable IOBKIN_SEL[0]: 0: TIM1_BKIN1 break input disable 1: TIM1_BKIN1 break input enable
5	Reserved	Reserved, must be kept at reset value.
4：1	BKINF	Break input filter 0000: 2 cycles 0001: 4 cycles 0010: 8 cycles 0011: 16 cycles 0100: 32 cycles 0101: 64 cycles 0110: 128 cycles 0111: 256 cycles 1000: 384 cycles 1001: 512 cycles 1010: 640 cycles 1011: 768 cycles 1100: 896 cycles 1101: 1024 cycles 1110: 1152 cycles 1111: 1280 cycles Note: Before changing the sampling frequency of break digital filter, turn off the break filter function.
0	BKINFE	Break input filter enable 1: Break input filter enable 0: Break input filter disable Note: Enable the filter after the break condition is configured. Break filter is used for level active signal.

11 TIM2 General-Purpose Timer

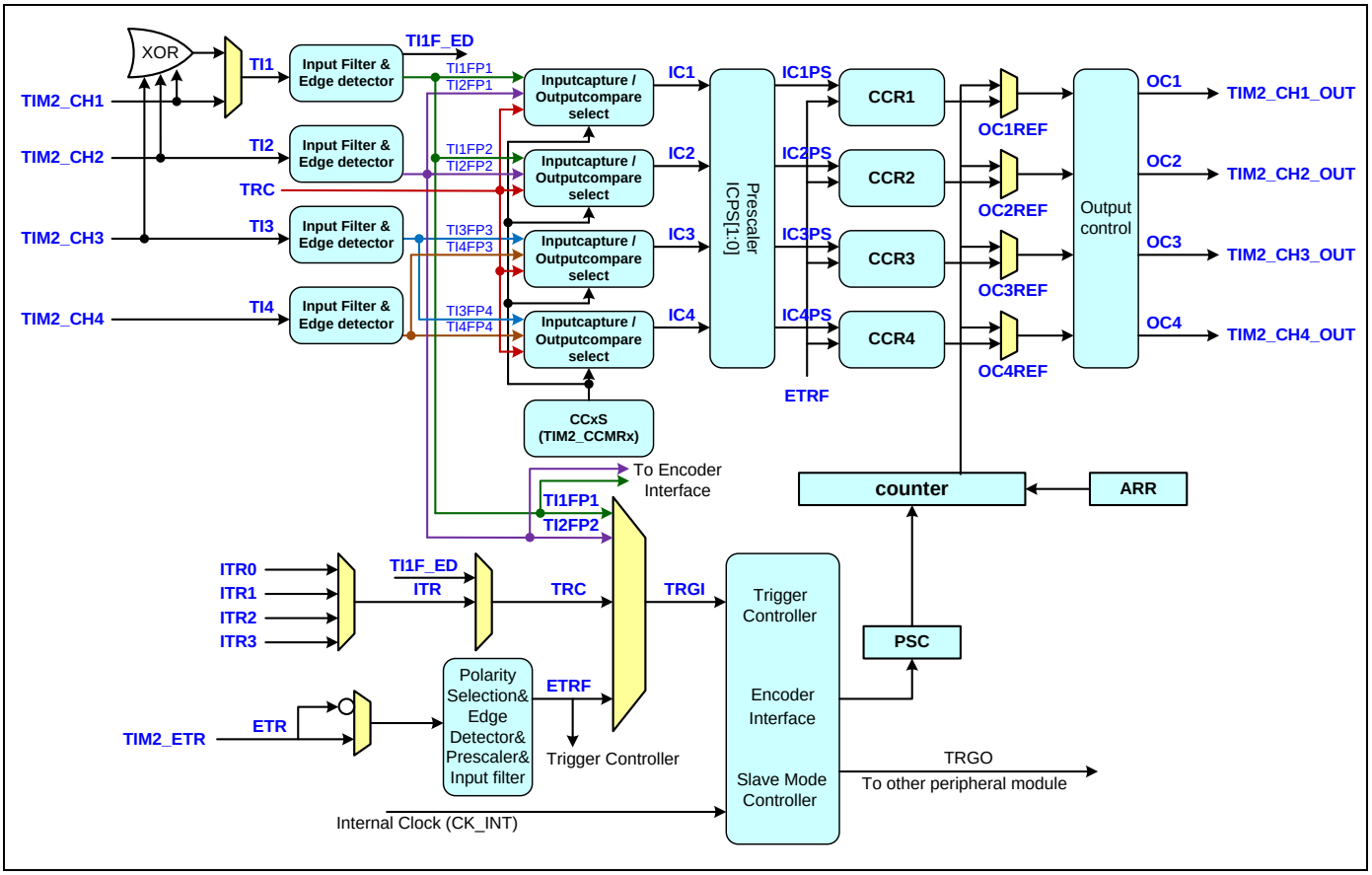
32-bit General-Purpose Timer include TIM2 、TIM5，they have the same function, this product only equipped with TIM2, which are uniformly described as TIMx in this chapter. Some diagrams take TIM2 as an example for illustration.

11.1 Overview

TIMx is composed of a 16-bit real-time programmable prescaler and a 32-bit auto reload counter with the programmable counting direction, and can conveniently provide with the counting and timing function. The counter is driven by a programmable prescaler. The general-purpose timer can be used for a variety of purposes, such as input capture function (measuring input signal pulse width or frequency, PWM input, etc.), output compare function (PWM output, one-pulse mode, etc.)

11.2 Function block diagram

Figure 11-1 TIMx block diagram



TIMx structure block diagram is mainly composed of input capture unit, output compare unit, and counter unit.

11.3 Main characteristics

- 16-bit real-time programmable prescaler, division factor: 1-65536.
- Clock source: Internal clock source, external clock input (TIx, ETRx), internal trigger input (ITRx).
- 32 bit auto reload counter (counting direction: up, down, up/down).
- Synchronization circuit to control the timer with external signals and to interconnect several timers to each other.
- Input capture: measure input signal pulse width or period.
- Trigger input can serve as external clock or be managed cycle-by-cycle.
- Support interfaces such as encoder and hall-sensor.
- 4 output channels
- Output compares (control output waveform or indicator counter already finishes timing).
- PWM output (edge alignment or central alignment mode)
- One-pulse mode.
- Interrupt/DMA request event: update event, trigger event, input capture, output compare.

11.4 Function description

11.4.1 Clock

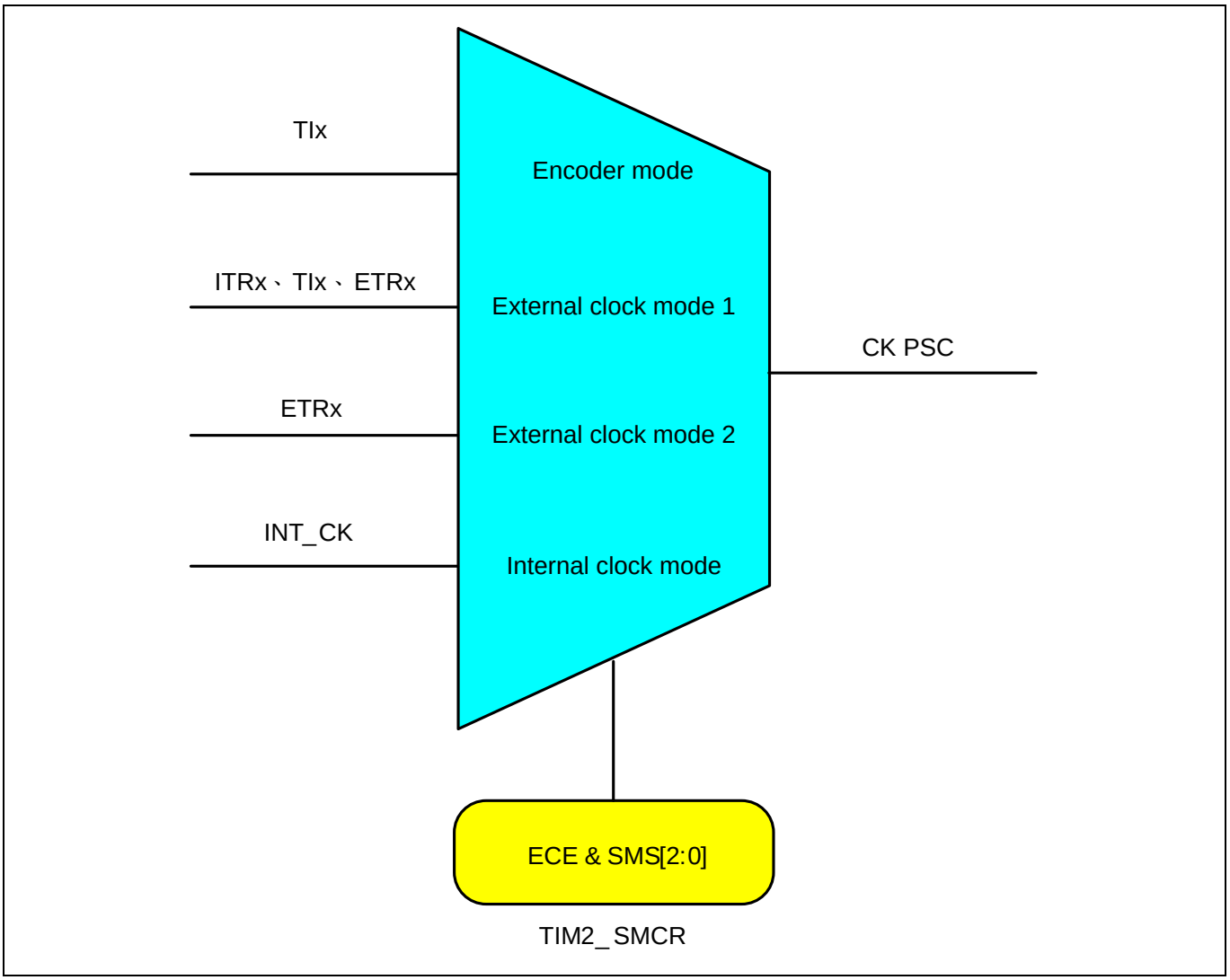
11.4.1.1 Clock selection

Several types of clock sources of the counter below:

- Internal clock (INT_CK)
- External clock mode 1: External trigger input TRGI (including TIx, ITRx, ETRx)
- External clock mode 2: External trigger input ETR (including ETRx)
- Encoder mode

The diagram of several types of clock choice above:

Figure 11-2 Clock selection



11.4.1.1.1 Internal clock source (INT _CK)

In case of SMS =000 of TIMx_SMCR register, disable the slave mode. Enable the counter, and the prescaler clock is driven by internal clock.

11.4.1.1.2 External clock mode 1(external trigger input TRGI, including TIx, ITRx, ETRx)

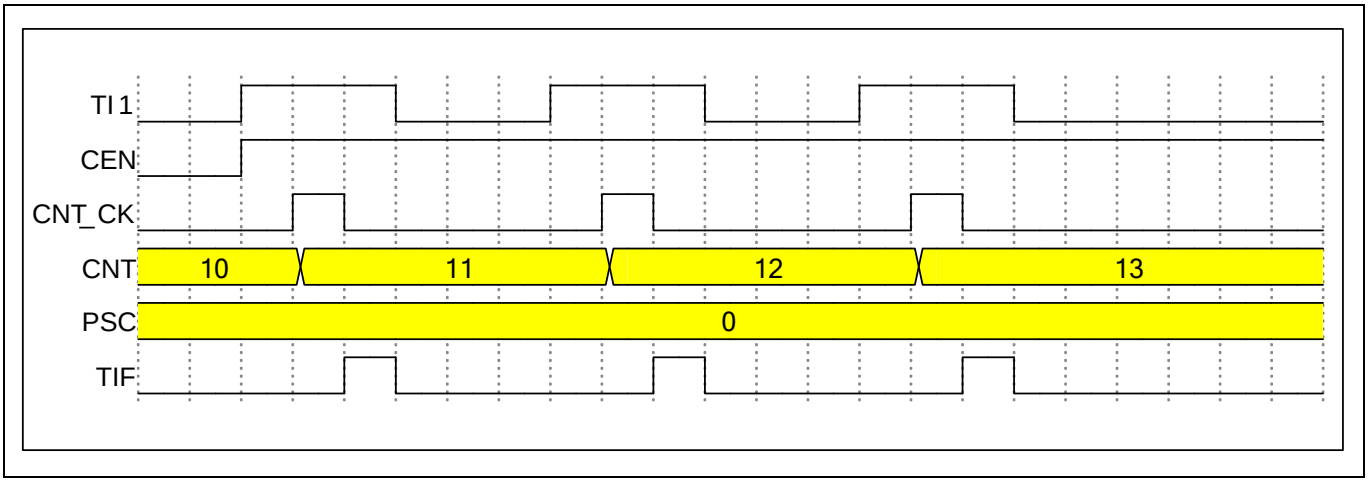
In case of SMS = 111 of TIMx_SMCR register, select the external clock mode 1 (TRGI). The counter is driven by the rising edge or falling edge of the selected input signal.

For example: count up at the rising edge on TI1 input. Specific configuration:

1. Configure TIMx_CCMR1 register CC1S=01, CC1 channel is configured as input, and IC1 is mapped on TI1; Configure TIMx_CCMR1 register IC1F[3：0], set input filter bandwidth; configure TIMx_CCER register CC1P=0, and select the rising edge as the active edge.
2. Configure TIMx_SMCR register TS=101, select TI1 as the trigger input source; configure TIMx_SMCR register SMS=111, select the external clock mode 1.
3. Configure TIMx_CR1 register DIR=0, select up counting mode; configure TIMx_CR1 register CEN=1 and start the counter.

TI1 input the active edge, count up once and TIF flag bit is set by hardware. The delay between the active edge of TI1 and actual clock of counter depends on the TI1 input pin sync circuit.

Figure 11-3 Control circuit in external clock mode 1



Note：TIMx_PSC must be kept at reset value when select the external clock mode.

11.4.1.1.3 External clock mode 2(external trigger input ETR, including ETRx)

In case of TIMx_SMCR register ECE=1, enable external clock mode 2, and the counter is driven by the active edge of ETR signal.

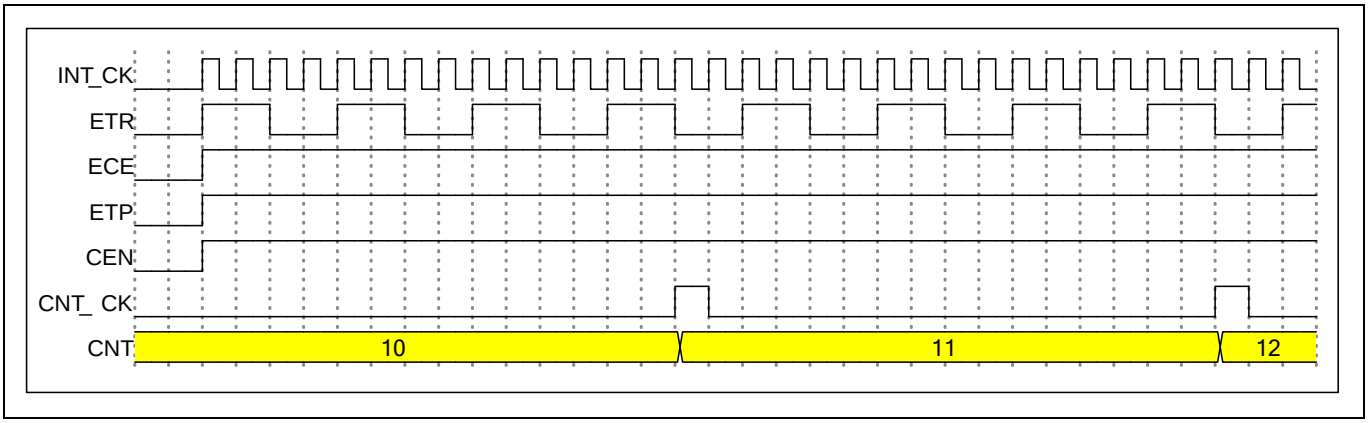
For example: count up once of each 4 falling edge on ETR Specific configuration:

1. Configure TIMx_SMCR register ETF [3:0] =0010, count up once of each 4 active edges on ETR; configure TIMx_SMCR register ETP=1, select falling edge as active edge; configure TIMx_SMCR register ECE=1, select external clock mode 2.

2. Configure TIMx_CR1 register DIR=0, select up counting mode; Configure TIMx_CR1 register CEN=1, start the counter.

The delay between the falling edge of ETR and actual clock of counter depends on the sync circuit of ETR signal terminal.

Figure 11-4 Control circuit in external clock mode 2



Note：TIMx_PSC must be kept at reset value when select the external clock mode.

11.4.1.1.4 Encoder mode

Refer to the chapter of slave mode-encoder interface for details.

11.4.1.2 Time base unit

TIMx time base unit mainly includes: Counter register (TIMx_CNT), prescaler register (TIMx_PSC), auto reload register (TIMx_ARR).

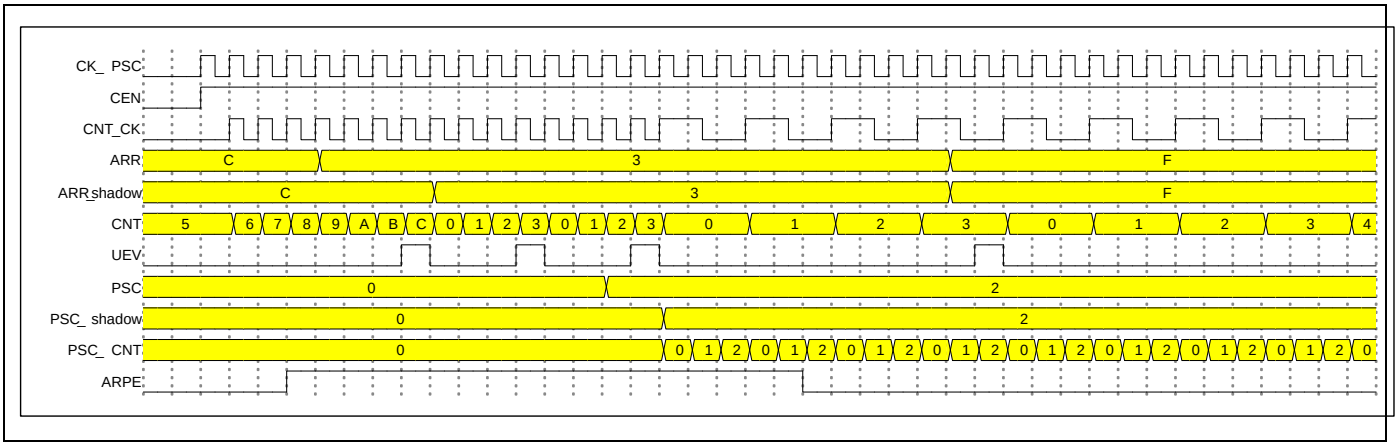
The counter unit is composed of a 32 bit counter with its related auto reload register, the counter

can count up, down or both up and down.

The counter clock is provided by prescaler. The prescaler is composed of the prescaler counter with its related register. The division factor is 1-65536. It can write any time, and gets active at the next update.

The auto preload register has a 32 bit shadow register with preload function. Set TIMx_CR1 register ARPE bit to select the content of ARR register are transferred into the shadow register permanently or at each update event.

Figure 11-5 Auto preload



11.4.1.3 Counter modes

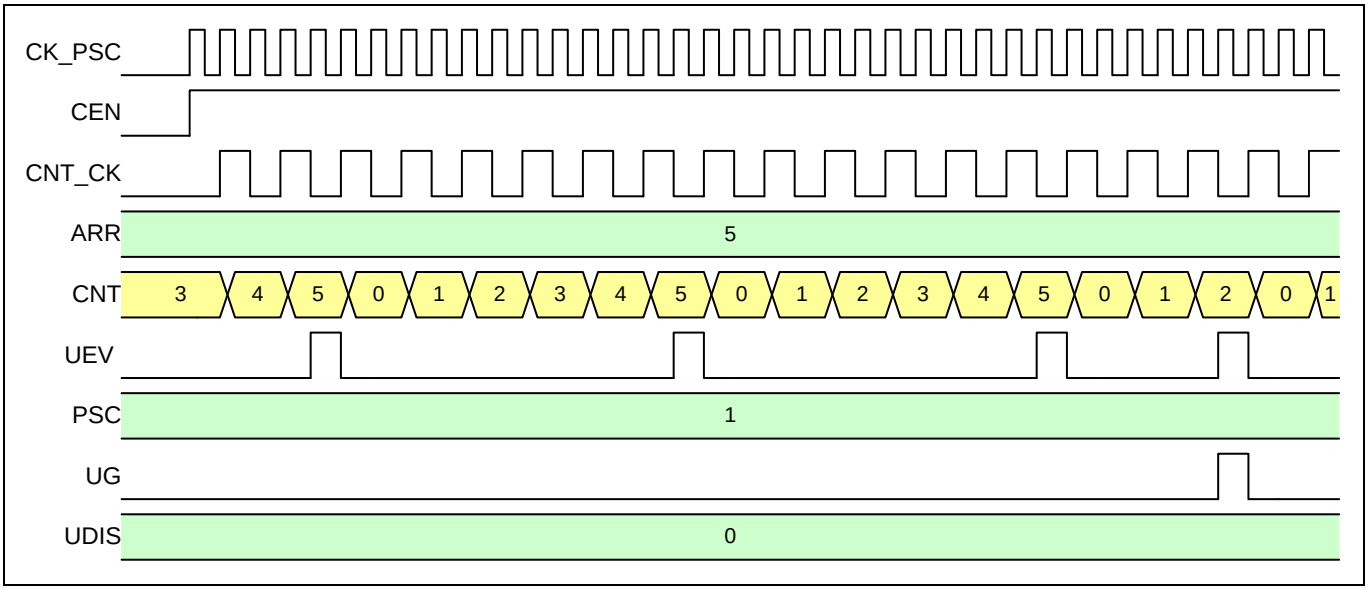
Configure DIR bit and CMS bit of TIMx_CR1 register to select the count mode. There are three count modes, up counting mode, down counting mode and central alignment mode (up/ down counting mode). Detailed descriptions are provided for each count mode.

11.4.1.3.1 Up counting mode

Configure TIMx_CR1 register CMS=0, DIR＝0, select up counting mode.

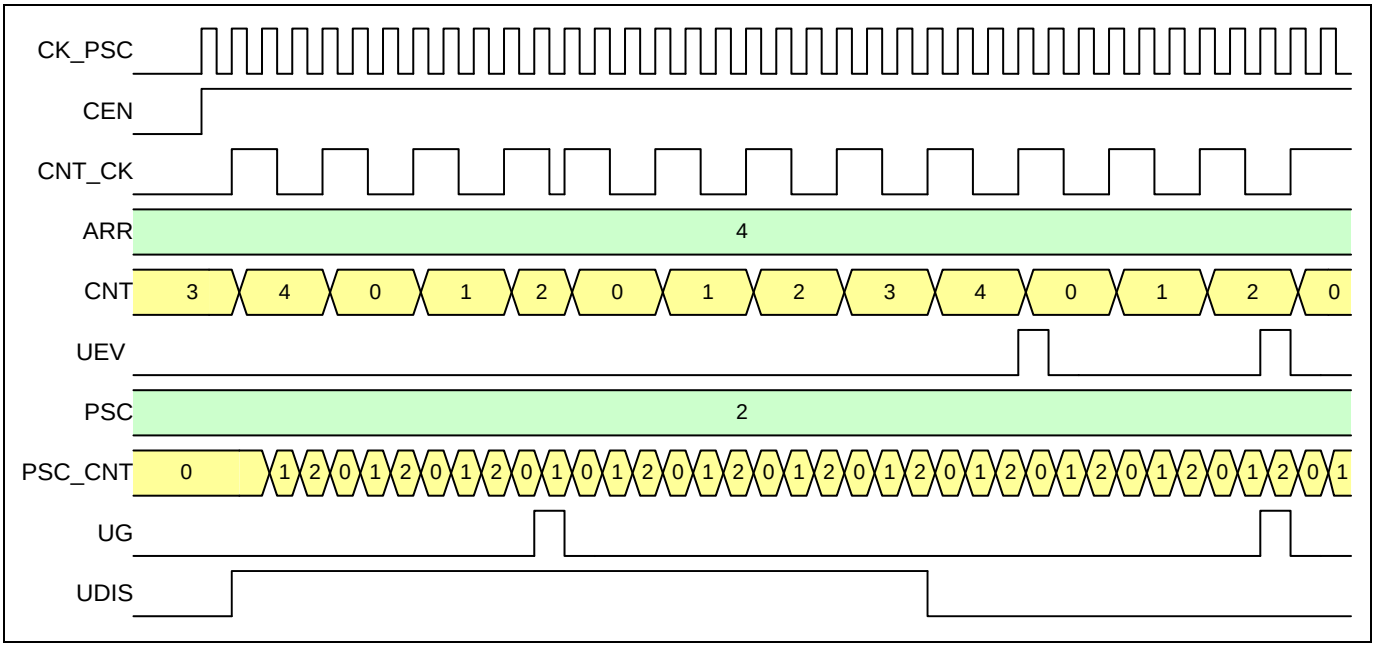
In up counting mode, enable TIMx_CR1 register CEN, and the counter begins up count from 0 to TIMx_ARR value and generates a counter overflow event (update). The counter begins up count again from 0. Set TIMx_EGR register UG =1, also generate an update event.

Figure 11-6 Up counting mode（UDIS=0）



Configuring TIMx_CR1 register UDIS=1, disable update event. In case of counter overflow event, the update event is not generated. Configuring UG=1, the update event is not generated. However, the counter and prescaler counter will initialize, and begin up count from 0.

Figure 11-7 Up counting mode (UDIS =1, disable update event)



Note: In case of update event:

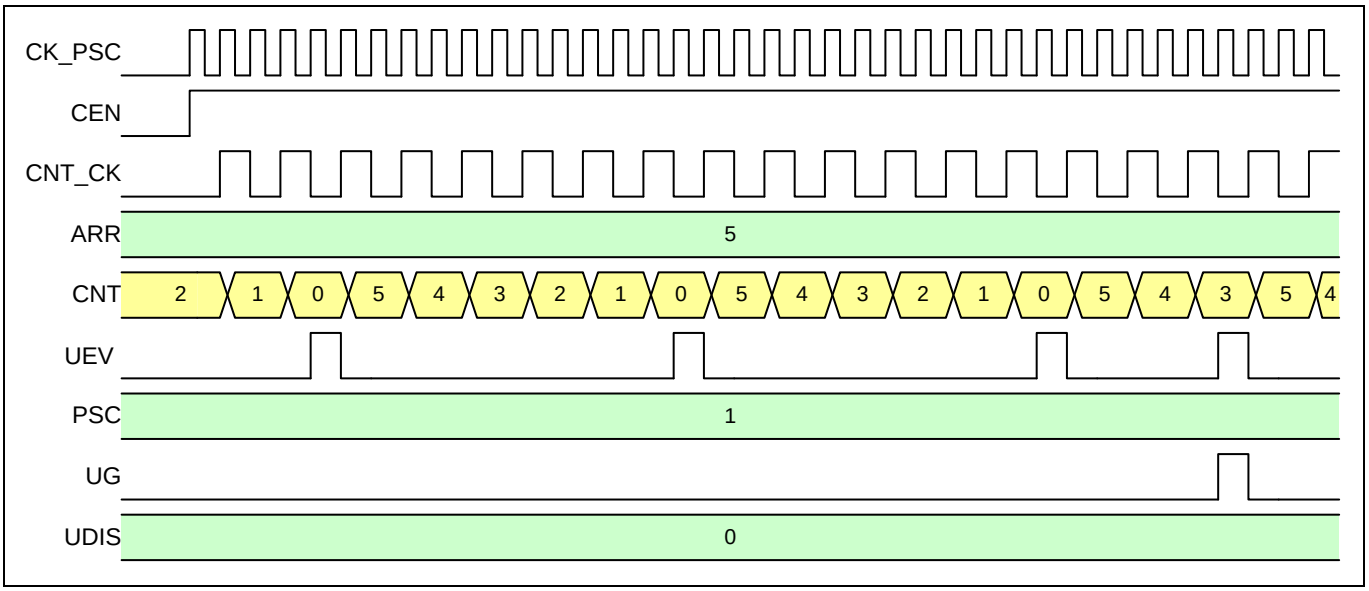
- ARR register value is loaded into ARR shadow register.
- Prescaler preload value becomes active.

11.4.1.3.2 Down counting mode

Configure TIMx_CR1 register CMS=0, DIR＝1, select down counting mode.

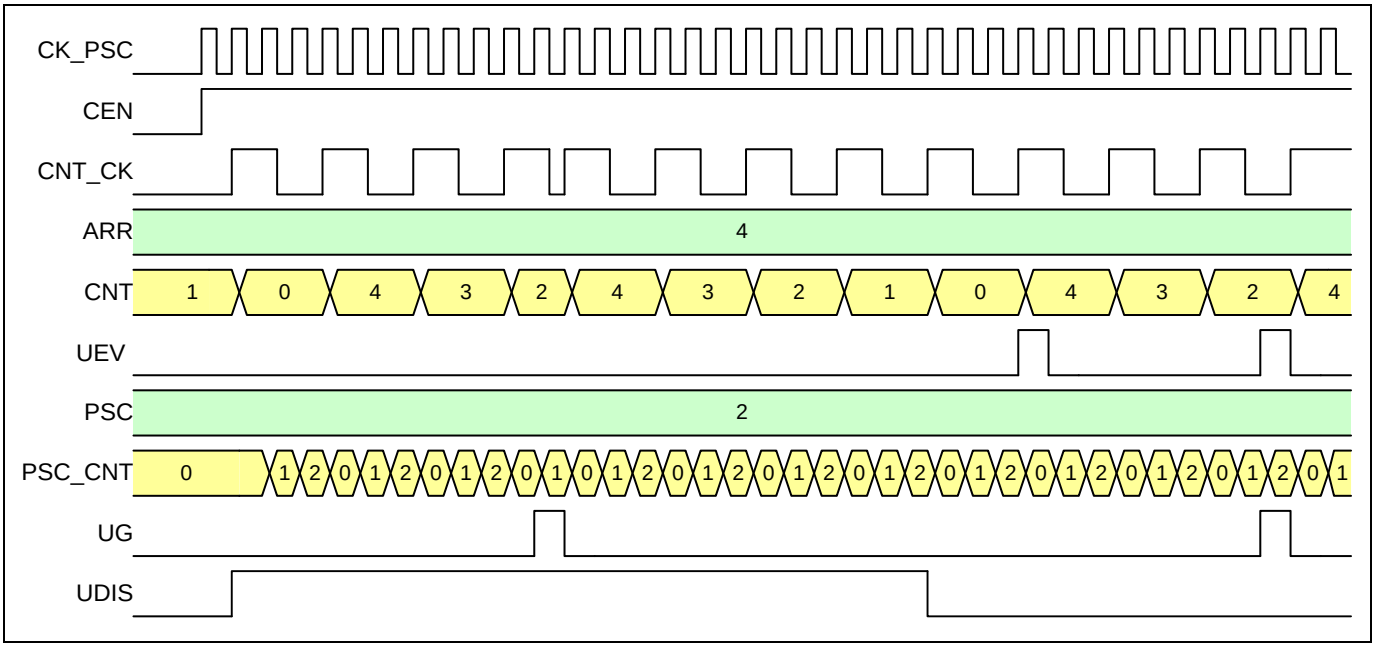
In down counting mode, the counter begins down count from the auto preload value TIMx_ARR. When counting to 0, generate an underflow event (update event). Set TIMx_EGR register UG =1, also generate an update event. After the update event, begin down count (TIMx_CR1 register UDIS=0) from the auto load value TIMx_ARR.

Figure 11-8 Down counting mode（UDIS＝0）



Configuring TIMx_CR1 register UDIS=1, disable update event. When the counter generates the underflow event, it doesn't generate an update event. Configuring UG=1 also doesn't generate update events. However, the counter and prescaler counter will initialize, and count begins from TIMx_ARR.

Figure 11-9 Down counting mode (UDIS=1 disables update event)



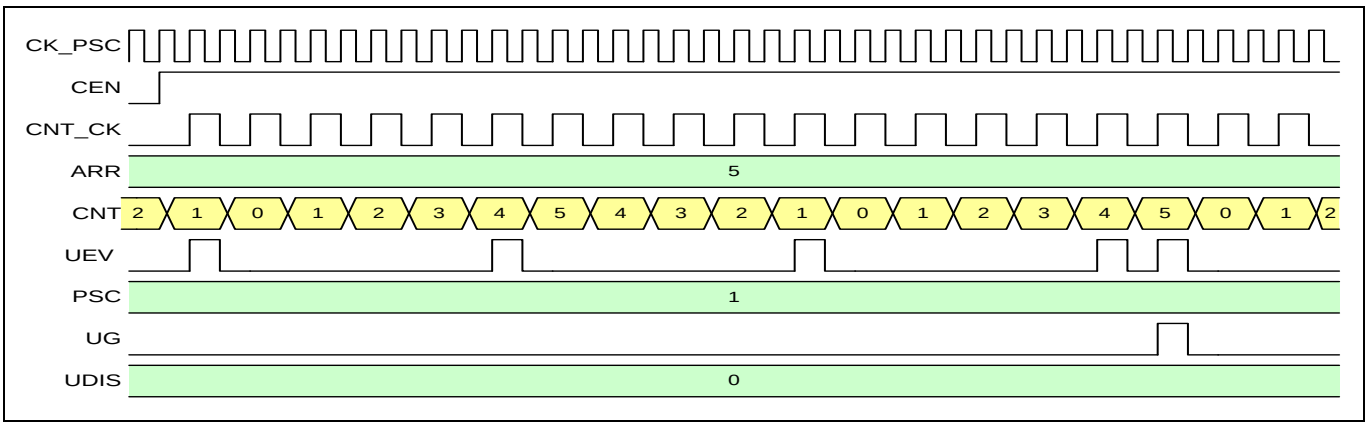
11.4.1.3.3 Central count mode (up/down count mode)

Configure TIMx_CR1 register CMS ≠ 0 (write DIR failure), and select central alignment count mode.

Central alignment counts mode, up count and down count run alternatively. In case of an up count to ARR-1, generate an overflow event. Begin down count to 1 from ARR, generate an underflow event, and begin up count from 0.

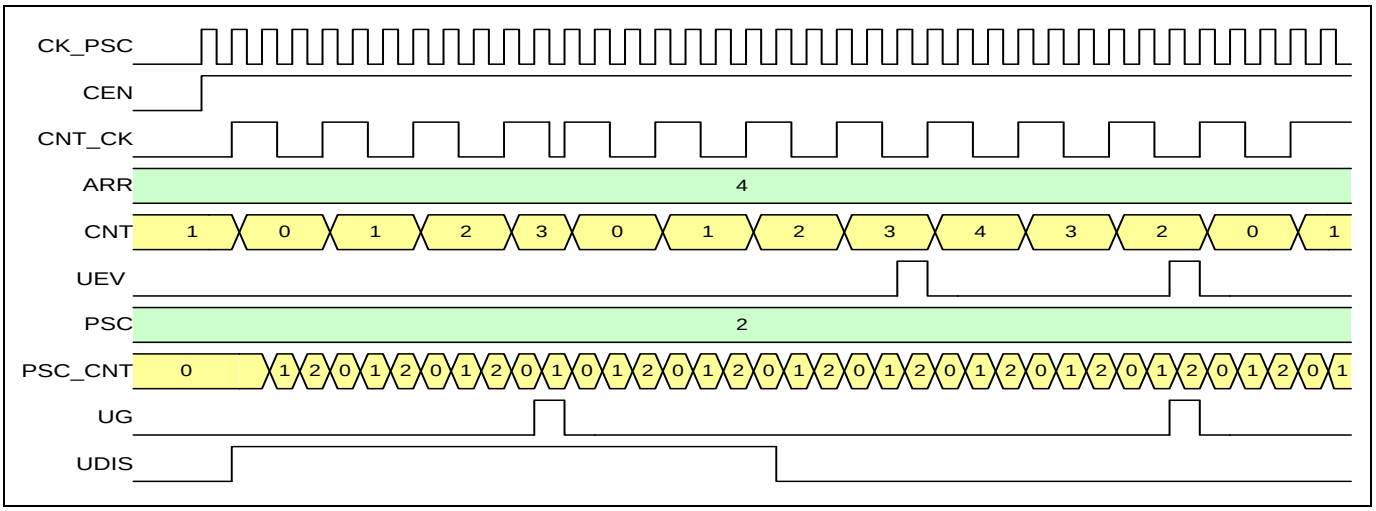
Setting TIMx_EGR register UG =1 also generate an update event. After update event, the counter begins up count again from 0 (TIMx_CR1 register UDIS=0).

Figure 11-10 Central count mode（UDIS=0）



When configuring TIMx_CR1 register UDIS=1, disable update event. When the counter generates the overflow or underflow event, it doesn't generate update event. Configuring UG=1 also doesn't generate update event. However, the counter and prescaler counter will initialize, and the count begins from 0.

Figure 11-11 Central count mode (UDIS=1 disables update event)

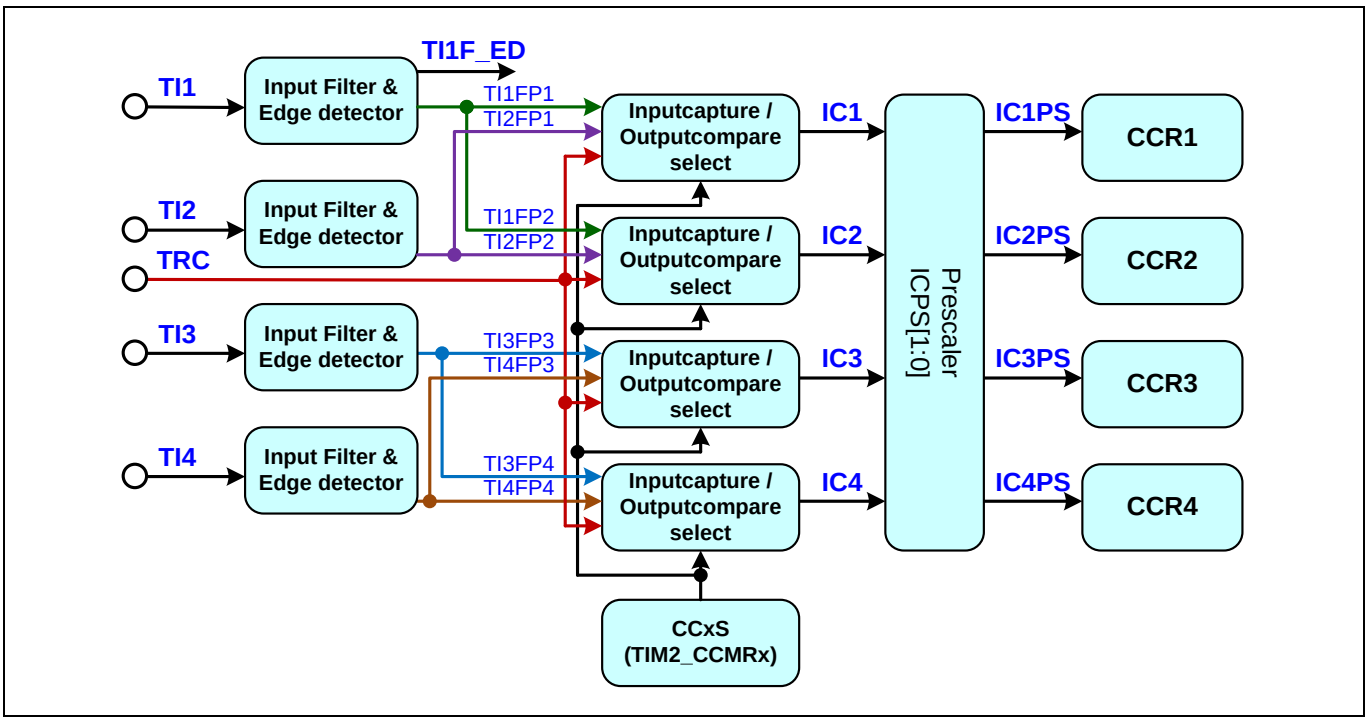


11.4.2 Input capture

11.4.2.1 Input capture

The input capture is composed of digital filter, multiplex, prescaler, etc. Its block diagram:

Figure 11-12 TIMx input capture block diagram



Configure TIMx_CCMRx register ICxF, set the filter width of a digital filter (filter sampling frequency and digital filter width are shown in the table below). When the input signal width is greater than the filter width, the input signal is active. When the digital filter samples the input signal of the input pin TIx, generate a filtered signal TIxF. Through the edge detector with the optional polarity, generate an effective signal TIxFPx, this signal can serve as the trigger input signal of the slave mode counter. This signal generates ICxPS via the prescaler for trigger input capture event.

Table 11-1 Digital filter width and ICxF correlation

IC1F[3：0]	Sample frequency and filter length	IC1F[3：0]	Sample frequency and filter length
0000	No filter, sample by f_{DTS}	1000	Sampling frequency $f_{sampling}=f_{DTS}$ /8, N=6
0001	Sampling frequency fsampling=f_{INT_CK}, N=2	1001	Sampling frequency $f_{sampling}=f_{DTS}$ /8, N=8
0010	Sampling frequency fsampling=f_{INT_CK}, N=4	1010	Sampling frequency $f_{sampling}=f_{DTS}$ /16，N=5
0011	Sampling frequency fsampling=f_{INT_CK}, N=8	1011	Sampling frequency $f_{sampling}=f_{DTS}$/16, N=6
0100	Sampling frequency fsampling=f_{DTS} /2, N=6	1100	Sampling frequency $f_{sampling}=f_{DTS}$ /16, N=8
0101	Sampling frequency fsampling=f_{DTS}/2, N=8	1101	Sampling frequency $f_{sampling}=f_{DTS}$ /32, N=5
0110	Sampling frequency $f_{sampling}=f_{DTS}$ /4, N=6	1110	Sampling frequency $f_{sampling}=f_{DTS}$ /32, N=6
0111	Sampling frequency $f_{sampling}=f_{DTS}$ /4, N=8	1111	Sampling frequency $f_{sampling}=f_{DTS}$ /32, N=8

In the input capture mode, when detecting the active edge of signal ICx, the counter's current value is locked in the corresponding shadow register, and then loaded into the corresponding capture/compare register. When enabling interrupt or DMA, and producing the capture event, generate the corresponding interrupt or DMA request. At the capture event, the capture flag bit CCxIF of the status register (TIMx_SR) will be set. Configure CCxIF =0 or read TIMx_CCRx data, clear CCxIF flag bit. When CCxIF is not cleared, next input capture event generated, the repeat capture flag CCxOF will be set. Configure CCxOF =0, can clear CCxOF flag bit.

For example, Sample TI1 input signal's active edge. At TI1 rising edge, capture the current counter value and lock it in the TIMx_CCR1 register, procedure steps as follows:

1. Configure TIMx_CCMR1 register CC1S=01, CC1 channel is configured as input, and IC1 is mapped on TI1.
2. Configure TIMx_CCMR1 register IC1F [3:0], configure the filter width of digital filter (configure upon need).
3. Configure TIMx_CCER register CC1P=0, select capture occurs in the rising edge of TI1 signal.
4. Configure TIMx_CCMR1 register IC1PSC [1:0], select the prescaler factor.
5. Configure TIMx_CCER register CC1E = 1, channel 1 capture enable.
6. Configure TIMx_DIER register CC1IE=1, enable channel 1 interrupt request; if the chip has the built in DMA, configure CC1DE=1 of the TIMx_DIER register to allow the DMA request on capture/compare channel 1.

Note：

- When the channel is configured as input mode, TIMx_CCRx register becomes read only.
- In case of more than two continuous captures and CCxIF flag is not cleared, the repeat capture flag CCxOF is set. In order to avoid losing the capture information that may be generated before the repeat capture flag CCxOF is set. It is recommended to read data before reading the repeat capture flag.
- Set the corresponding CCxG bit of TIMx_EGR register, generate the input capture interrupt or DMA request through software.

11.4.2.2 PWM capture

Differences of PWM input mode configuration with ordinary input capture:

- Two ICx signals are active on edges with opposite polarity and mapped on the same TIx input.
- Configure the slave mode counter at reset mode, set one TIxFP as trigger input signal.

For example: Measure TI1 PWM signal period (TIMx_CCR1 register) and duty cycle (TIMx_CCR2 register), the measuring value depends on the internal clock INT_Ck frequency and prescaler value. Specific steps:

1. Configure TIMx_CR1 register DIR =0, select counter as up counting mode.
2. Configure TIMx_CCMR1 register CC1S = 01, map IC1 on TI1, and select active input of TIMx_CCR1.
3. Configure TIMx_CCER register CC1P =0, select active polarity of TI1FP1 (active on rising

edge) (capture the counter value to TIMx_CCR1, and clear the counter).

4. Configure TIMx_CCMR1 register CC2S =10, map IC2 on TI1, select active input of TIMx_CCR2.

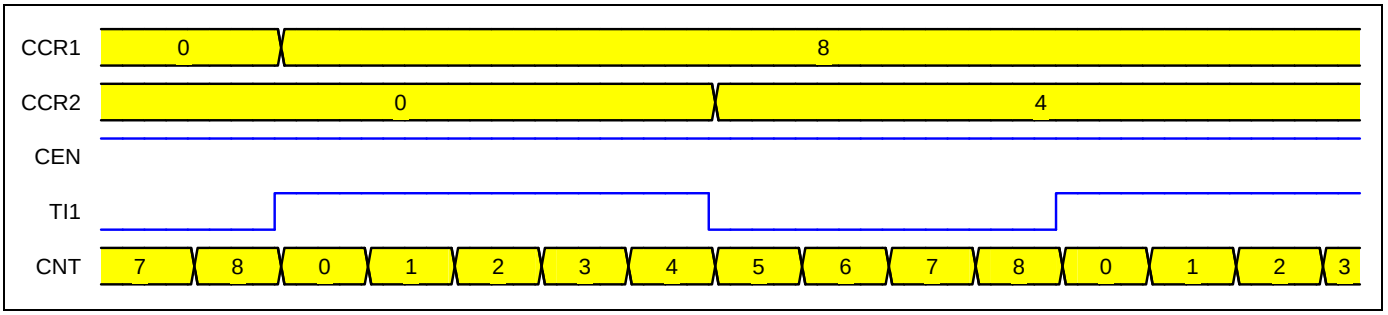
5. Configure TIMx_CCER register CC2P =1, and select active polarity of TI2FP2 (active on falling edge) (capture counter value to TIMx_CCR2).

6. Configure TIMx_SMCR register TS = 101, select TI1FP1 as the active trigger input signal.

7. Configure TIMx_SMCR SMS = 100, set the slave mode counter as reset mode.

8. Configure TIMx_CCER register CC1E=1 and CC2E = 1. Enable CC1 channel and CC2 channel capture.

Figure 11-13 PWM input mode sequence

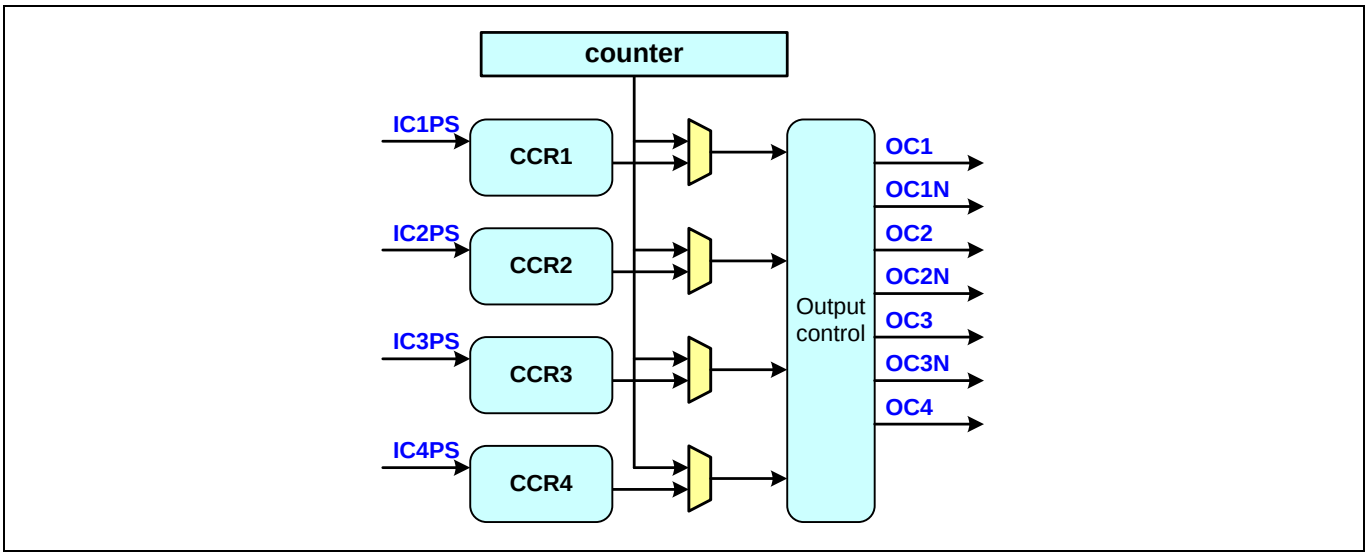


Note: The slave mode counter only connects TI1FP1 and TI2FP2, PWM input mode is only applicable to TIMx_CH1/TIMx_CH2 port input signal.

11.4.3 Output compare

The compare output of the capture/compare channel is composed of the comparator, output control circuit and capture/compare register. Its structure is shown below:

Figure 11-14 Output compare block diagram



In output compare mode, the content of capture/compare register is loaded into the shadow register and then compare the content of shadow register with the current value of counter. The capture/compare module includes a capture/compare register (preload register) and a shadow register. When reading and write, only programmed the capture/compare register.

11.4.3.1 Force output

Configure TIMx_CCMRx register CCxS = 00 and set the channel CCx as output mode. Configure TIMx_CCMRx register OCxM bit, and directly enforce the output compare signal as active or inactive

level, independently of the output compare result. Configure TIMx_CCMRx register OCxM = 100, and enforce output compare signal as inactive level. And OCxREF is enforced at a low level. Configure TIMx_CCMRx register OCxM = 101, and enforce output compare signal as active level. And OCxREF is enforced as high level (OCxREF always active as high level).

Note: In force output mode, TIMx_CCRx shadow register and counter output compare is still performed. The corresponding flag bit of compare result will also be changed. When enable the corresponding interrupt and DMA request, it will also generate the corresponding interrupt and DMA request.

11.4.3.2 Output compare

In the output compare mode, when the values of counter and capture compare register matches, different waveform can be exported according to the OCxM bit of TIMx_CCMRx register.

For example, when the value of counter and capture/compare register matches, the functions in the output compare mode:

1. When compare match, OCxM value is different, output channel x signal OCx run differently:
   - OCxM = 000: OCx signal keeps its level.
   - OCxM = 001: OCx signal is set as active level.
   - OCxM = 010: OCx signal is set as inactive level.
   - OCxM = 011: OCx signal toggle.
2. When match, set the flag bit of status register (CCxIF bit of TIMx_SR register).
3. Configure TIMx_DIER register CCxIE =1, generate an interrupt when match.
4. Configure TIMx_DIER register CCxDE =1, generate a DMA request when match (only suitable for the built in DMA product).

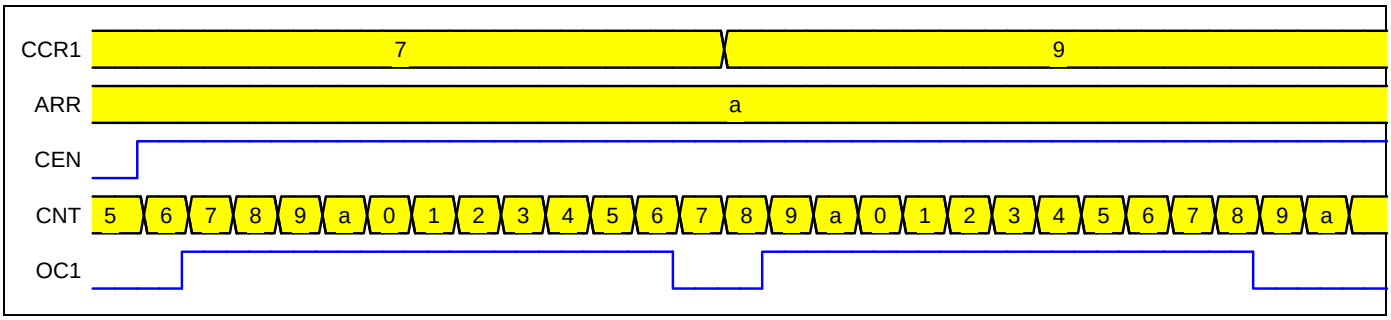
The output compare mode can also be used to export a single pulse (one-pulse mode).

Configuration steps of channel 1 output compare mode:

1. Configure counter clock (select clock source, configure prescaler factor).
2. Configure TIMx_ARR and TIMx_CCR1 register.
3. Configure TIMx_DIER register CC1IE =1, enable capture/compare 1 interrupt.
4. Configure output mode:
   - Configure TIMx_CCMR1 register OC1M = 011, toggle when OC1 compare match.
   - Configure TIMx_CCMR1 register OC1PE = 0, disable preload function of TIMx_CCR1 register.
   - Configure TIMx_CCMR1 register CC1P = 1, OC1 is active low polarity.
   - Configure TIMx_CCER register CC1E = 1, enable channel 1 output, OC1 signal is exported to the corresponding output pin.
5. Configure TIMx_CR1 register CEN =1, enable the counter.

Configure TIMx_CCMRx register OCxPE=0, disable TIMx_CCRx register preload function, write TIMx_CCRx register any time, and the write value updated immediately. Configure TIMx_CCMRx register OCxPE=1, and enable TIMx_CCRx register preload function, Updated only at the next update event. The figure below is an example.

Figure 11-15 Output compare mode, OC1 signals toggle when match



Note: In the output compare mode, the update event has no effect on output result. In the force

output mode, the compare output between TIMx_CCRx shadow register and counter still performed. The corresponding flag bit of compare result is also be changed. When enabling the corresponding interrupt and DMA request, it still generates the corresponding interrupt and DMA request.

11.4.3.3 PWM output

In the PWM mode, according to the value of TIMx_ARR register and TIMx_CCRx register, generate a frequency and duty cycle controllable PWM waveform.

Configure OCxM=110 or OCxM=111 of TIMx_CCMRx register corresponding to the channel x, select channel x to PWM mode 1 or PWM mode 2. In the PWM mode, the counter will always be compared with CCRx. According to configuration and compare results, channel x exports different signals. Therefore, TIMx can generate 4 PWM output signals with independent duty under the same frequency. In PWM mode, enable the preload function of TIMx_CCRx and TIMx_ARR. The written value of TIMx_CCRx preload register and TIMx_ARR preload register will become effective at the next update event, and load the corresponding shadow register. Therefore, in the PWM mode, before enabling counter, it's required to set TIMx_EGR UG=1, and generate the update event to initialize all registers.

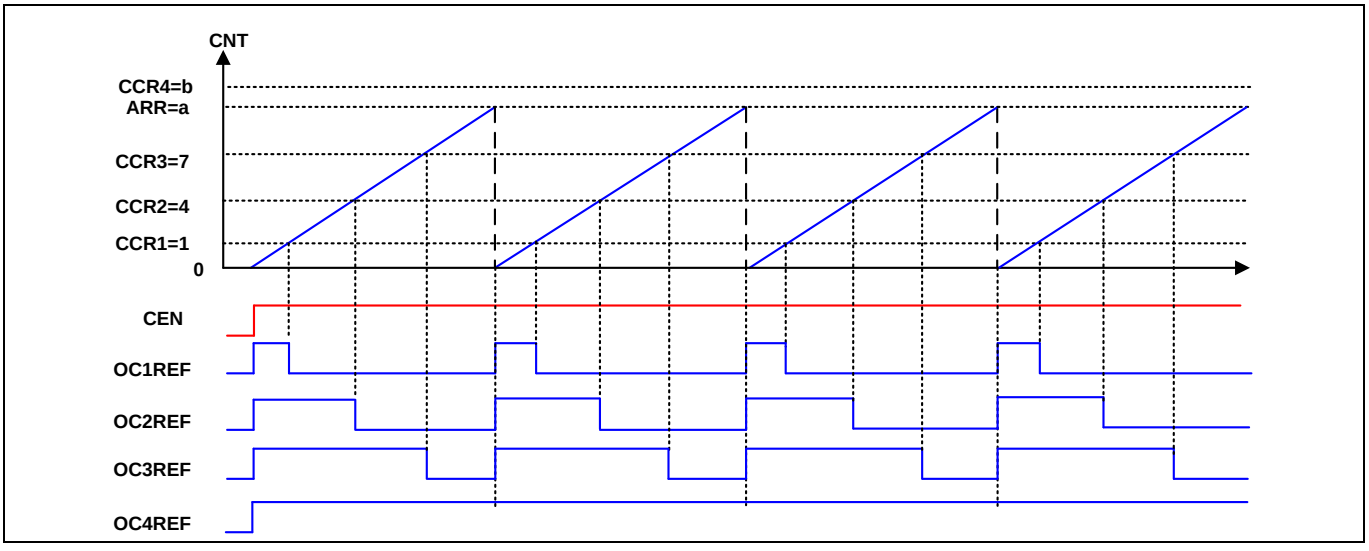
Configure TIMx_CCER register CCxP, select OCx's active polarity. Configure TIMx_CCER register CCxE bit and control OCx output enable. Configure TIMx_CR1 register CMS bit, and select edge alignment or central alignment PWM signal.

- CMS=00, edge alignment mode, configure DIR, select count mode (up or down).
- CMS=01, central alignment mode 1.
- CMS=10, central alignment mode 2.
- CMS=11, central alignment mode 3.

11.4.3.3.1 PWM edge alignment mode - up counting mode

Based on up counting mode configuration, configure TIMx_CCMRx register CCxS =00, select output mode, OCxM =110, select PWM mode 1. In the case of TIMx_CNT ＜TIMx_CCRx, channel x (OCxREF) is in active level. Otherwise, it's inactive level. If TIMx_CCRx compare value is greater than the auto reloads value (TIMx_ARR), OCxREF stays at an active level. When the value of the comparison is 0, OCxREF is at an inactive level. The figure below is the example of waveform in PWM mode 1 during edge alignment up count in case of CCR1 =1, CCR2 =4, CCR3 =7, CCR4 =b and ARR =a.

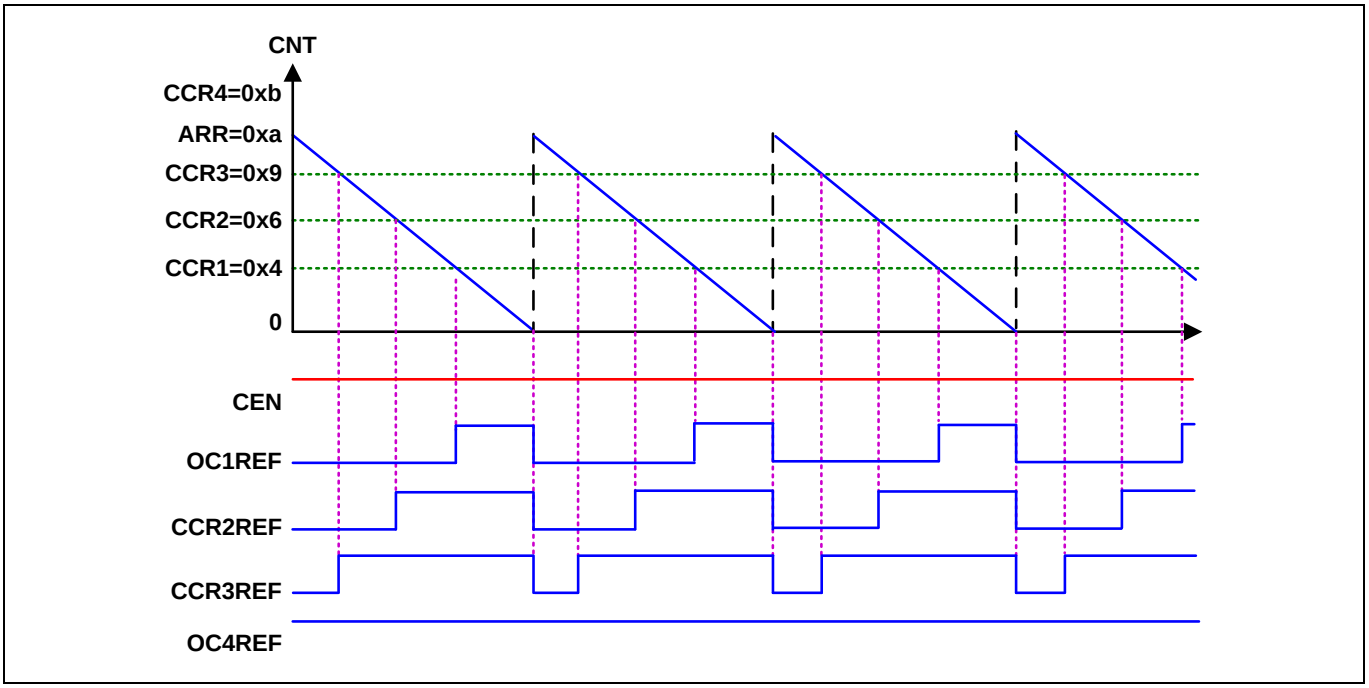
Figure 11-16 Waveform in PWM mode 1 during edge alignment up count



11.4.3.3.2 PWM edge alignment mode -down counting mode

Based on the down counting mode configuration, configure TIMx_CCMRx register CCxS=00, select output mode, OCxM=110, select PWM mode 1. In the case of TIMx_CNT > TIMx_CCRx, channel x (OCxREF) is at an inactive level. Or else, it's at an active level. The figure is an example of a waveform in PWM mode 1 during edge alignment down count in case of CCR1=4, CCR2=6, CCR3=9, CCR4=b and ARR=a.

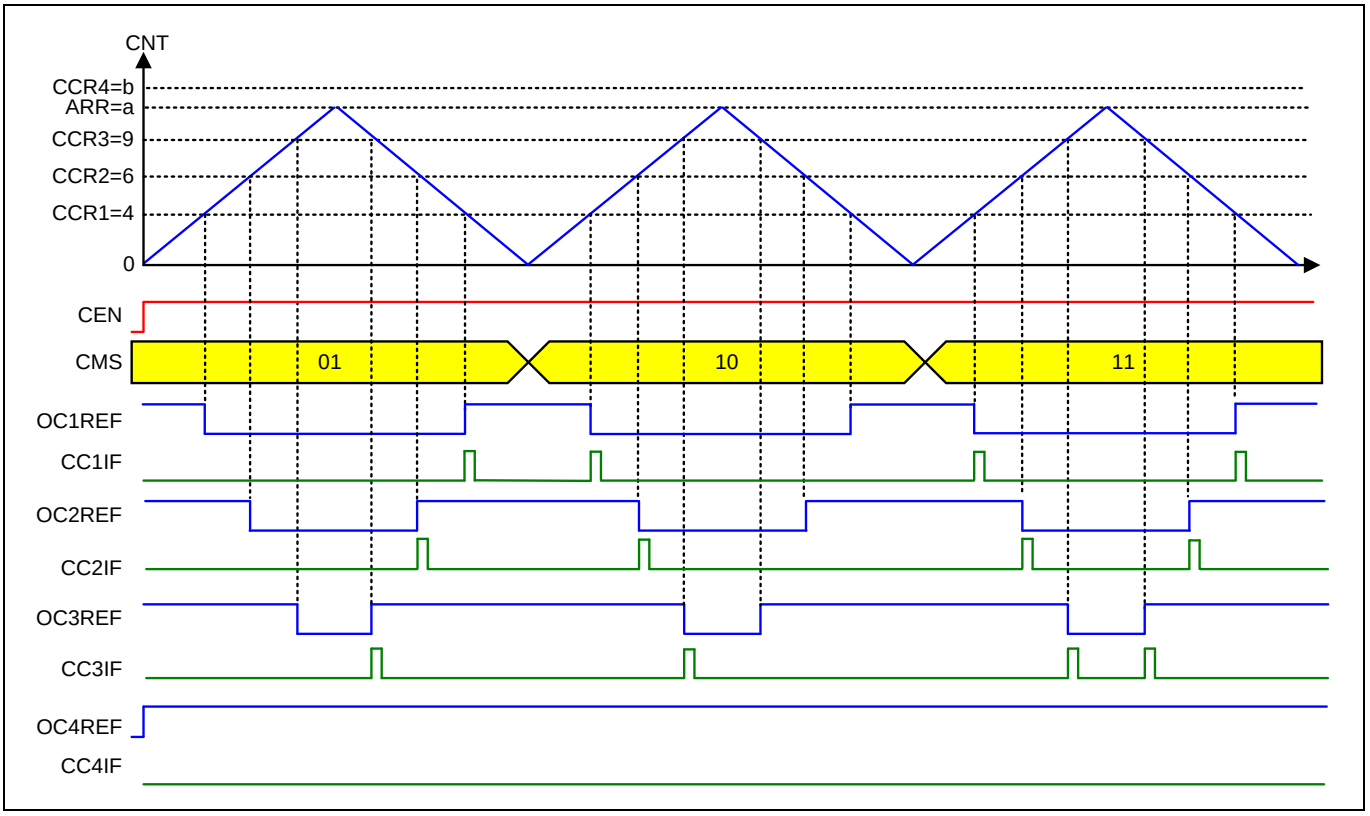
Figure 11-17 Waveform in PWM mode 1 during edge alignment down count



11.4.3.3.3 PWM central alignment mode

Configure TIMx counter as central alignment count mode. Configure TIMx_CCMRx register CCxS=00 and select output mode. According to the CMS with different configurations, the output compare interrupt flag bit is set during up count (CMS =01), set during down count (CMS =10) or set during up or down count (CMS =11). The figure below is an example of waveform in central alignment PWM mode 1 (CMS =1) during CCR1=4, CCR2=6, CCR3=9, CCR4=b, ARR=a.

Figure 11-18 Waveform in central alignment PWM mode 1

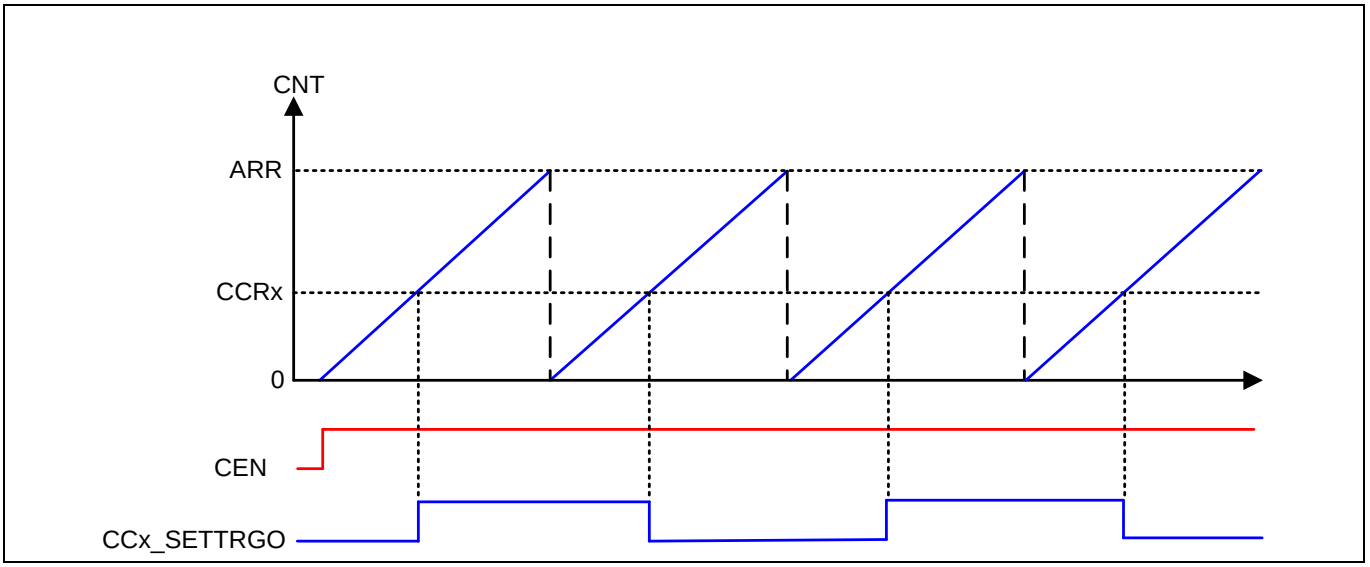


11.4.3.3.4 Trigger Source Output

In PWM mode, CCx_SETTRGO signal can be used to trigger modules such as ADC. This chapter only introduces trigger logic of the SETTRGO signal, please refer to the ADC chapter for the detailed trigger source selection, trigger edge selection, and other information.

In edge alignment mode, every time compare match (the current count value of TIMx_CNT is equal to TIMx_CCRx), CCx_SETTRGO undergoes a flip. The following figure shows in edge aligned up counting mode, example of CCx_SETTRGO output.

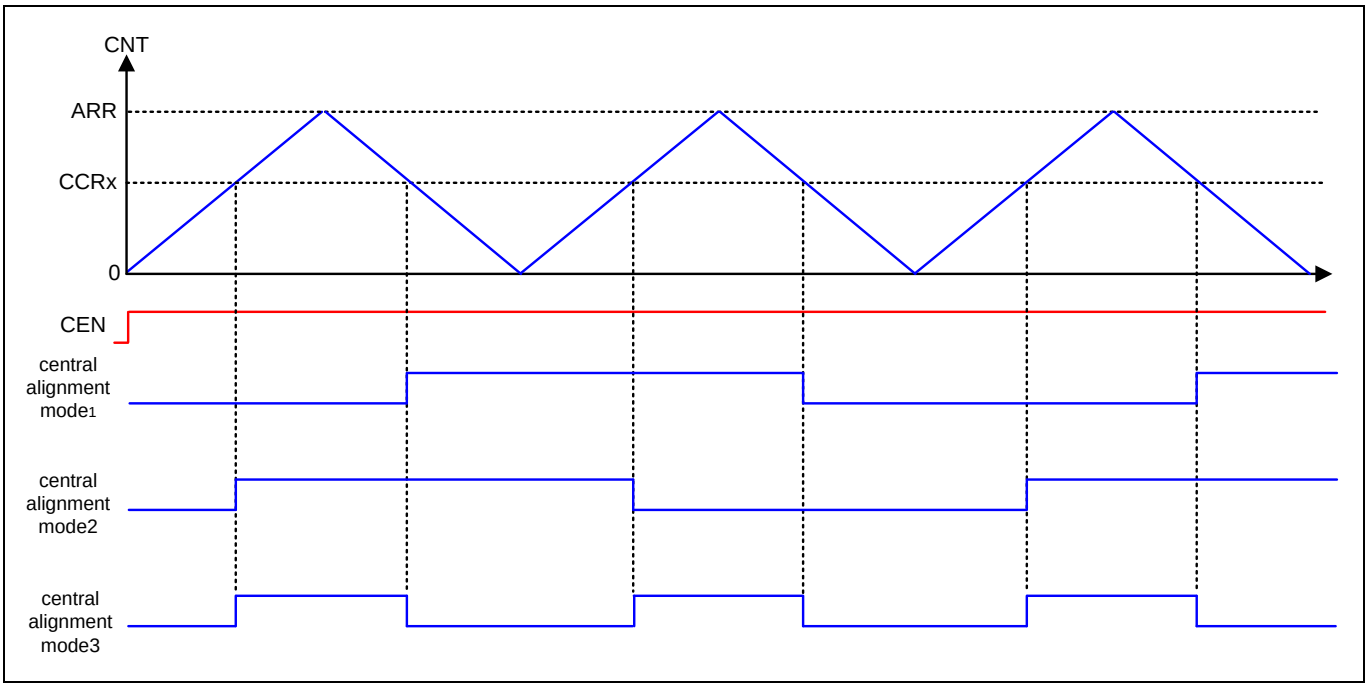
Figure 11-19 example of CCx_SETTRGO output with edge aligned up counting mode



In central alignment mode 1, when compare match the down counting cycles, CCx_SETTRGO undergoes a flip. In central alignment mode 2, when comparing match the up-counting cycles, CCx_SETTRGO undergoes a flip. In central alignment mode 3, flip occurs when the up or down counting cycles match. The following figure shows in central alignment mode, an example of

CCx_SETTRGO output.

Figure 11-20 example of CCx_SETTRGO output with central alignment mode



11.4.3.4 External event clear OCxREF

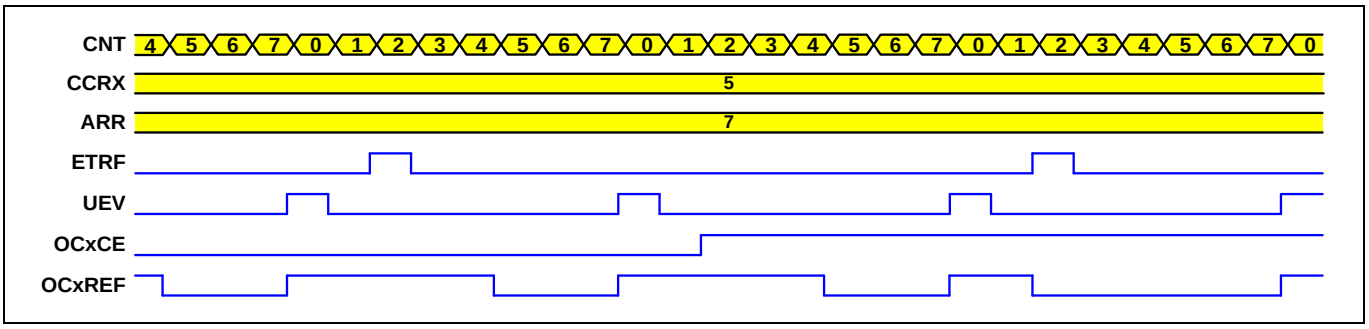
When configuring TIMx_CCMR register OCxCE=1, OCxREF can be driven low by the active level of ETRF input terminal till the next update event (UEV). The function can only be used in the output mode and PWM mode, it does not work in the force output mode.

For example, when OCxREF signal is connected to an external input, ETR configuration:

1. Configure TIMx_SMCR register ETPS [1:0] =00, close the external trigger prescaler.
2. Configure TIMx_SMCR register ECE=0, disable external clock mode 2.
3. Configure TIMx_SMCR register ETF [3:0] and ETP, configure ETR signal trigger polarity and filter width.

In the figure below, when ETRF input change triggers ETRF as high, it corresponds to different OCxCE value, and OCxREF signal action (PWM mode).

Figure 11-21 The external event clear OCxREF



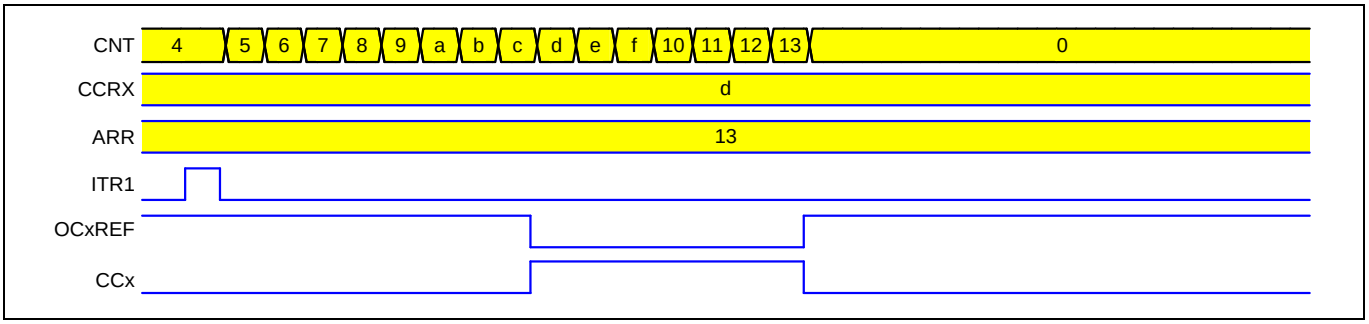
11.4.3.5 One-pulse output

In the one-pulse mode (OPM), the counter responds to a stimulus and generates a pulse with width programmable. Configure TIMx_CR1 register OPM =1. Select one-pulse mode, trigger signal's active edge or configuring CEN =1 can enable the counter. CEN =1 remains until the next update event or configure CEN =0.

The necessary condition to generate pulse is the difference of compare value and initial value of the counter. Necessary configuration prior to counter enable:

- Up counting mode: Counter CNT < CCRx ≤ ARR.
- Down counting mode: Counter CNT > CCRx.

Figure 11-22 one-pulse mode



For example, when TI2 detects the rising edge, and after latency t_{DELAY}, OC2 generates a positive pulse with the length of t_{PULSE}.

Configure TI2FP2 serves as the trigger source:

1. Configure TIMx_CCMR1 register CC2S = 01, map TI2FP2 in TI2.
2. Configure TIMx_CCER register CC2P = 0, detect TI2FP2 rising edge.
3. Configure TIMx_SMCR register TS = 110, TI2FP2 as the slave mode counter trigger (TRGI).
4. Configure TIMx_SMCR register SMS = 110, select trigger mode, TI2FP2 enables the counter.

OPM waveform is determined by TIMx_ARR and TIMx_CCR1 (consider clock frequency and counter prescaler): t_{DELAY} is determined by TIMx_CCR1 register value and CNT initial value; TIMx_ARR - TIMx_CCR1 value is t_{PULSE}.

Next example is generating a negative pulse. In case of compare and match, generate a waveform from 1 to 0. When the counter preload value is achieved, generate a waveform from 0 to 1:

1. Configure TIMx_CCMR1 register OC1M = 111, select PWM mode 2.
2. Configure TIMx_CCER register CC1P = 1, export active low level.
3. Configure TIMx_CCMR1 OC1PE = 1 and TIMx_CR1 register ARPE, enable preload register.
4. Configure TIMx_CCR1 register and TIMx_ARR register.
5. Configure TIMx_EGR register UG =1, generate an update event.
6. Wait for an external trigger event on TI2.

In this case, TIMx_CR1 registers DIR=0, CMS=0, OPM= 1, stop counting in the next update event (when the counter toggles to 0 from the auto load value).

11.4.3.5.1 OCx fast enable

OCx fast enable is a special scenario of one-pulse mode. In the one-pulse mode, set TIMx_CCMR register OCxFE=1, and enforce OCxREF to directly respond to stimulus without relying on the compare result of counter and compare value. The output waveform is the same as the waveform when compare and match. Thus, it can remove the compare time and export the compare result fast. OCx rapid output enables only becomes active in the PWM mode.

11.4.4 Slave mode

11.4.4.1 Encoder interface

The encoder interface mode means the counter count under interaction of orthogonal signals of

TI1 and TI2. At the change of input source, the count direction is auto modified by hardware. Configure TIMx_SMCR register SMS bit and select input source. According to input source difference, divide the encoder interface into 3 modes: SMS =001, encoder interface mode 1; SMS =010, encoder interface mode 2; SMS =011, encoder interface mode 3; the specific count in the three modes is expressed below. Two input TI1 and TI2 are used as the interface of orthogonal encoder.

In the encoder mode, prior to the counter enable, configure the ARR register. In the encoder interface mode, it's like using an external clock with direction selection. The counter counts continuously between 0 to the auto load value of TIMx _ARR register (up count and down count are controlled by external clock).

Note: The encoder mode doesn't support the external clock mode 2.

In the encoder interface code, the counter is auto changed according to the speed and direction of increment encoder. Therefore, the counter content always points to the encoder position. The count direction corresponds to the direction of the connected sensor. The table below presents all possible combinations, assuming that TI1 and TI2 are not exchanged at the same time.

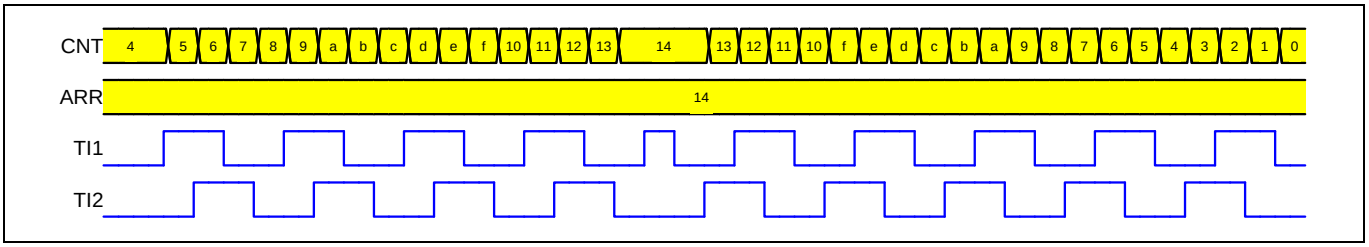
Table 11-2 Relations of count direction and encoder signal

Count mode	Relative level (TI1FP1 relative to TI2, TI2FP2 relative to TI1)	TI1FP1 signal		TI2FP2 signal	
		Rising	Falling	Rising	Falling
Encoder interface mode 1 (only count at TI2)	High level	-	-	up count	down count
Encoder interface mode 1 (only count at TI2)	Low level	-	-	down count	up count
Encoder interface mode 2 (only count at TI1)	High level	down count	up count	-	-
Encoder interface mode 2 (only count at TI1)	Low level	up count	down count	-	-
Encoder interface mode 3 (only count at TI1 and TI2)	High level	down count	up count	up count	down count
Encoder interface mode 3 (only count at TI1 and TI2)	Low level	up count	down count	down count	up count

The counter configuration and sequence figure in the encoder interface mode is below. We can see the count signal generation and direction control in the figure. Specific configuration:

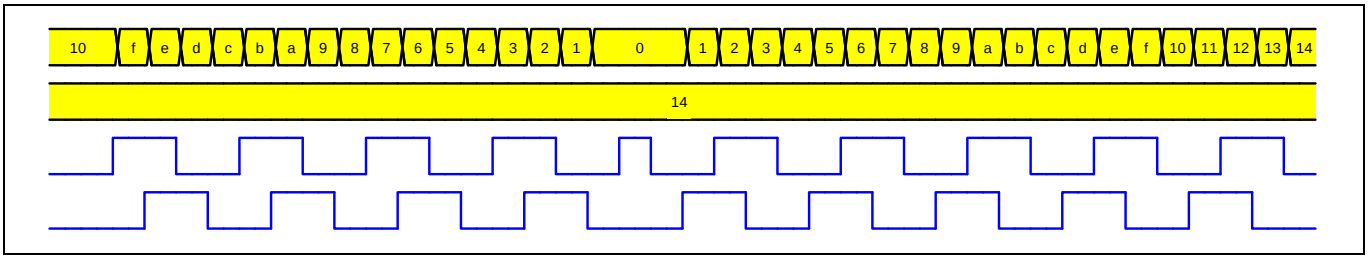
1. Configure TIMx_CCMR register CC1S=01, map IC1FP1 on TI1.
2. Configure TIMx_CCMR register CC2S=01, map IC2FP2 on TI2.
3. Configure TIMx_CCER register CC1P =0, IC1 non-inverted, IC1=TI1.
4. Configure TIMx_CCER register CC2P =0, IC2 non-inverted, IC1=TI2.
5. Configure TIMx_SMCR register SMS =011, select encoder mode 3. According to the input level of another signal, the counter counts at the edge of TI1FP1 and TI2FP2.
6. Configure TIMx_CR1 register CEN =1, enable the counter.

Figure 11-23 Counter time sequence in the encoder code



Counter sequence when the IC1FP1 polarity inverted (CC1P = 1, another configuration unchanged).

Figure 11-24 IC1FP1 polarity inverted sequence of encoder interface mode



In the encoder interface mode, the counter can provide the current position information of the sensor. Use another timer in the capture mode to measure the interval period of two encoder events and acquire dynamic information (speed, acceleration, and deceleration). According to the interval period of two encoder events, regularly read the counter. Lock the counter value in the third input capture register (the acquire signal should be cyclic and generated by another timer). When the chip has the built in DMA, read the value upon DMA request.

11.4.4.2 Reset mode

Configure TIMx_SMCR register SMS=100, select slave reset mode. In this mode, the TRGI input event will clear and restart the counter.

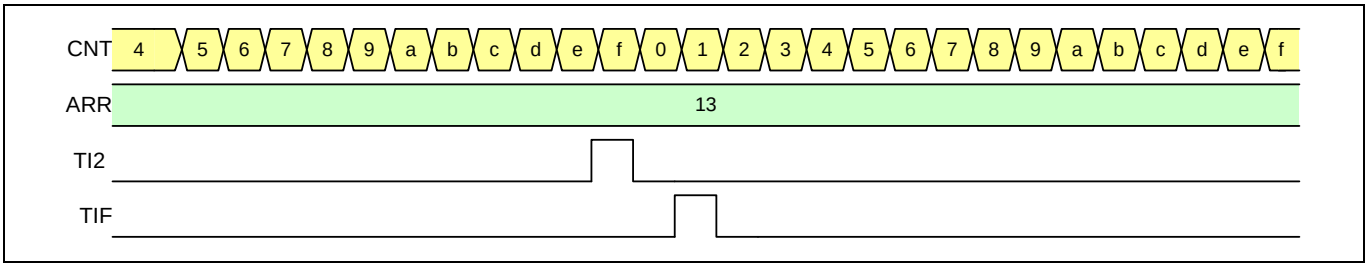
For example, TI2 input terminal falling edge trigger counter restart:

1. Configure TIMx_CCMR1 CC2S=01. CC2 channel is configured as input mode. IC2 is mapped on T12. Configure TIMx_CCER register CC2P=1, detect the falling edge.
2. Configure TIMx_SMCR register SMS = 100 and select the slave mode as reset mode. Configure TIMx_SMCR register TS =110, select timer input 2 after filter (TI2FP2) as the trigger input of sync timer.
3. Configure TIMx_CR1 register DIR=0, select the count direction as the up count; configure PSC=0, no division; configure CEN=1, enable the counter.

The counter clock source is provided by the internal clock. When detecting T12 falling edge, clear and restart the counter. The trigger interrupt flag is set by hardware.

TIMx_ARR = 0x13 time sequence in reset mode

Figure 11-25 Control time sequence of reset mode



11.4.4.3 Gate mode

Configure TIMx_SMCR register SMS=101, select slave gate mode. In this mode, select active level (0: high level active, 1: low level active) according to the TIMx_CCER register CCxP value. When TRGI input is active level, the counter always starts. Otherwise, the counter stops (but without reset operation), and the counter start and stop are countable.

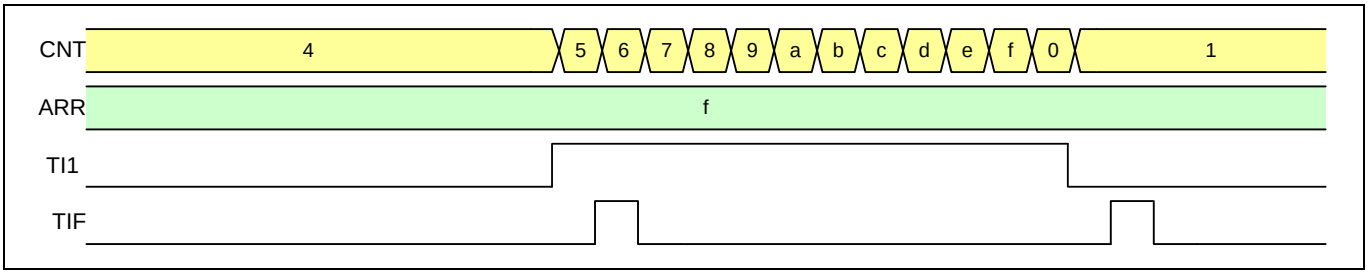
For example, the counter only counts when TI1 is high:

1. Configure CC1S=01 of TIMx_CCMR1 register. CC1 channel is configured as input mode. IC1 is mapped on TI1. Configure TIMx_CCER register CC1P=0, detect the high level of TI1.
2. Configure TIMx_SMCR register SMS=101, select slave gate mode; configure TIMx_SMCR register TS=101, select the timer input after filter 1 (TI1FP1) as the sync counter trigger input.
3. Configure TIMx_CR1 register DIR=0, select count direction as the up count; configure PSC=0, no division; configure CEN=1, enable the counter.

The counter clock source is provided by the internal clock. When detecting TI1 high level, the counter count starts. When TI1 is at a low level, the counter stops. When the counter starts or stops, set TIF as 1.

TIMx_ARR=0xf time sequence in gate mode.

Figure 11-26 Control time sequence of gate mode



11.4.4.4 Trigger mode

Configure TIMx_SMCR register SMS=110, and select slave trigger mode. Select active edge (0: rising edge active, 1: falling edge active) according to the TIMx_CCER register CCxP value. When TRGI input is active edge, the counter starts counting. The timer startup is controlled and the stop is not controlled.

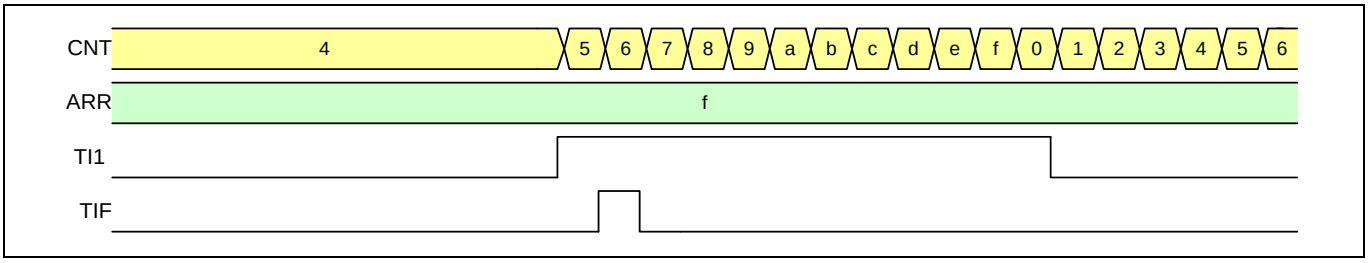
For example, the counter count starts at the rising edge of TI1 input:

1. Configure CC1S=01 of TIMx_CCMR1 register. CC1 channel is configured as input mode. IC1 is mapped on TI1. Configure TIMx_CCER register CC1P=0, and detect the rising edge.
2. Configure TIMx_SMCR register SMS = 110, select slave trigger mode; configure TIMx_SMCR register TS=101, select the timer input 1 (TI1FP1) after filter as the trigger input of counter.
3. Configure TIMx_CR1 register DIR =0, select count direction as the up count; configure PSC=0, no division.

The counter clock source is provided by internal clock. When detecting TI1 rising edge, the counter count starts.

TIMx_ARR =0xf sequence in the trigger mode.

Figure 11-27 Control time sequence of trigger mode



11.4.4.5 External clock mode 2+ slave mode

When the clock source selects the external clock mode 2, and ETR signal is used as external clock input, it can be used together with the other slave mode. In this mode, the slave mode only supports the reset mode, gate mode, trigger mode, and not supports the external clock mode 1 and encoder mode.

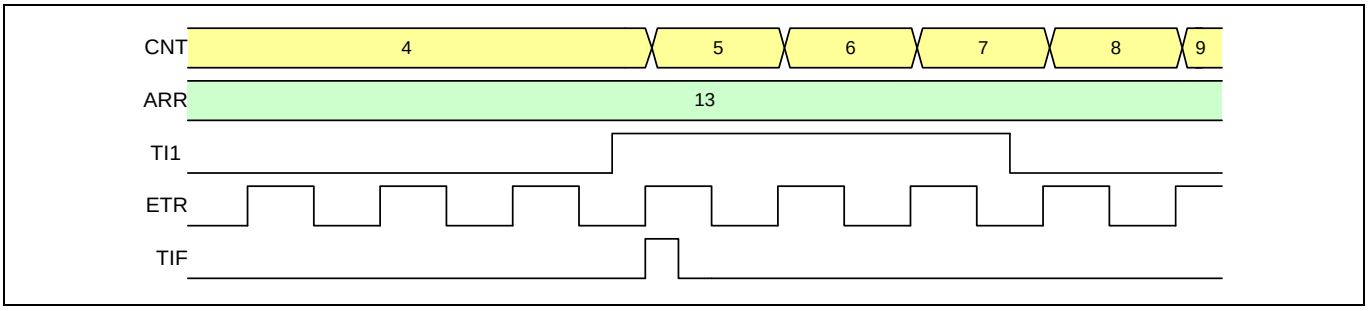
For example, in the trigger mode, the counter counts once at each rising edge of ETR:

1. Configure TIMx_SMCR register ETF = 0000, don't use the digital filter; configure TIMx_SMCR register ETPS = 00, close the prescaler; configure TIMx_SMCR register ETP =0, detect ETR rising edge; configureTIMx_SMCR register ECE = 1, enable external clock mode 2.
2. Configure TIMx_CCMR1 register CC1S=01, CC1 channel is configured as input, and IC1 mapped on TI1 as the input capture source; configure TIMx_CCER register CC1P=0, selected rising edge is active.
3. Configure TIMx_SMCR register SMS = 110, select slave trigger mode. Configure TIMx_SMCR register TS = 101, select TI1 as input source.
4. Configure TIMx_CR1 register DIR=0, select the count direction as up count. Configure PSC=0, no division.

The counter count begins at the rising edge of TI1, and set TIF. The latency between the ETR signal rising edge and actual reset time of counter depends on the sync circuit of ETR input terminal.

TIMx_ARR=13 time sequence in external clock mode 2 + slave mode (trigger mode).

Figure 11-28 Control time sequence of external clock mode 2 +slave mode (trigger mode)



11.4.5 Timer synchronization

Different timers are internally connected, achieve timer cascading or synchronization.

Detailed descriptions are given in the relevant chapter of TIM1.

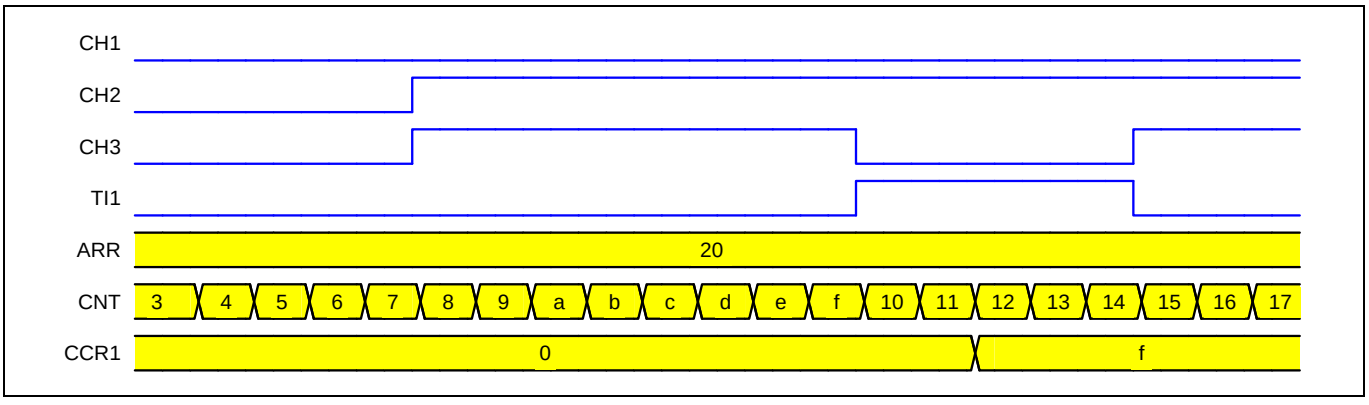
11.4.6 Timer XOR function

Configure TIMx_CR2 registerTI1S =1, connect the TIMx_CH1, TIMx_CH2 and TIMx_CH3 pin to TI1 input terminal via XOR for all input modes of the timer.

For example, TIMx_CH1, TIMx_CH2 and TIMx_CH3 pins are connected to TI1 input terminal via XOR. Sample TI1 input signal's active edge. When TI1 rising edge is active, capture the current counter value, and lock it in the TIMx_CCR1 register. Specific configuration:

1. Configure TIMx_CR2 register TI1S=1, configure three inputs of the timer and connect it to TI1 input channel via XOR.
2. Configure TIMx_CCMR1 register CC1S=01, CC1 channel is configured as input, IC1 is mapped on TI1.
3. Configure TIMx_CCMR1 register IC1F [3:0], configure digital filter width (configure upon need).
4. Configure TIMx_CCER register CC1P=0, select capture in the rising edge of TI1 signal.
5. Configure TIMx_CCMR1 register IC1PS, select prescaler factor.
6. Configure TIMx_CCER register CC1E = 1, enable input/capture channel 1 capture.
7. Configure TIMx_CR1 register CEN=1, enable the counter.

Figure 11-29（TI1 XOR input）input capture waveform



Hall interface circuit

Detailed descriptions are given in the relevant chapter of TIM1.

11.4.7 Debug mode

In the debug mode, configure DBG_CR register DBG_TIMx_STOP=1, TIMx counter stops counting. (Details in the debug section)

11.4.8 Interrupt

TIMx interrupt includes: capture/compare 1 interrupt, capture/compare 2 interrupt, capture/compare 3 interrupt, capture/compare 4 interrupt, update interrupt, trigger interrupt. When the corresponding interrupt enable bit is set, generate the corresponding event and interrupt.

Table 11-3 List of interrupt events

Interrupt event	Flag bit	Enable bit
Capture/compare 1 interrupt	CC1IF	CC1IE
Capture/compare 2 interrupt	CC2IF	CC2IE
Capture/compare 3 interrupt	CC3IF	CC3IE
Capture/compare 4 interrupt	CC4IF	CC4IE
Update interrupt	UIF	UIE
Trigger interrupt	TIF	TIE

11.4.9 DMA

TIMx can generate multiple requests in case of a single event. The main objective is to read multiple registers based on the period by reprogramming some registers of TIMx without software.

TIMx_DCR and TIMx_DMAR registers are relevant to DMA mode. The objective of DMA counter is unique, and must point to the TIMx_DMAR register. After enabling DMA and at the specified TIMx event, TIMx will send request to DMA. Each write to TIMx_DMAR register will be redirected to one TIMx register.

TIMx_DCR register DBL bit has defined DMA's continuous transmission length, namely the transmission register number. In case of reading and writing TIMx_DMAR, the timer identifies DBL and determines the number of registers to be transmitted. TIMx_DCR register DBA bit has defined the DMA transmission base address, and the beginning offset of TIMx_CR1 register address (00000 for TIMx_CR1, 00001 for TIMx_CR2...00110 for TIMx_CCMR1, etc.)

For example: DMA continuous transmission mode is used to update the content of CCR1, CCR2 and CCR3 register in case of update event. Specific configuration:

1. Configure the corresponding DMA channel.
2. Configure TIMx_DCR register DBA=01101, configure DMA base address, select the register address with offset address TIMx_CCR1.
3. Configure TIMx_DCR register DBA=00010, configure transmission length 3.
4. Configure TIMx_DCR register UDE=1, enable update event DMA request.
5. Configure TIMx_CR1 register CEN=1, start the counter.
6. Enable DMA channel.

In this example, in case of one update event, DMA will transmit the ready data of the corresponding memory address to CCR1, CCR2 and CCR3 register or transmit the value of CCR1, CCR2 and CCR3 register to the corresponding memory address.

11.5 Register

Table 11-4 TIMx register overview

Offset	Acronym	Register Name	Reset
0x00	TIMx_CR1	Control Register 1	0x0000
0x04	TIMx_CR2	Control Register 2	0x0000
0x08	TIMx_SMCR	Slave Mode Control Register	0x0000
0x0C	TIMx_DIER	DMA/Interrupt Enable Register (DMA is only suitable for the chip with built in DMA)	0x0000
0x10	TIMx_SR	Status Register	0x0000
0x14	TIMx_EGR	Event Generation Register	0x0000
0x18	TIMx_CCMR1	Capture/Compare Mode Register 1	0x0000
0x1C	TIMx_CCMR2	Capture/Compare Mode Register 2	0x0000
0x20	TIMx_CCER	Capture/Compare Enable Register	0x0000
0x24	TIMx_CNT	Counter	0x0000 0000
0x28	TIMx_PSC	Prescaler	0x0000
0x2C	TIMx_ARR	Auto Reload Register	0x0000 0000
0x34	TIMx_CCR1	Capture/Compare Register 1	0x0000 0000
0x38	TIMx_CCR2	Capture/Compare Register 2	0x0000 0000
0x3C	TIMx_CCR3	Capture/Compare Register 3	0x0000 0000
0x40	TIMx_CCR4	Capture/Compare Register 4	0x0000 0000
0x48	TIMx_DCR	DMA Control Register (only suitable for the chip with built in DMA)	0x0000
0x4C	TIMx_DMAR	DMA Address Register Of Continuous Mode (only suitable for the chip with built in DMA)	0x0000
0x50	TIMx_OR	Input Option Register	0x0000

11.5.1 TIMx_CR1 Control Register 1

Address offset: 0x00
Reset value: 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved						CKD		ARPE	CMS		DIR	OPM	URS	UDIS	CEN
						rw		rw	rw		rw	rw	rw	rw	rw

Bit	Field	Description
15：10	Reserved	Reserved, must be kept at reset value.
9：8	CKD	Clock division Division ratio between the timer clock (CK_INT) frequency and the dead-time and sampling clock used by the dead-time generators and the digital filters (ETR, TIx). 00：$t_{DTS} = t_{INT\_CK}$ 01：t_{DTS} =2x t_{INT_CK} 10：t_{DTS} =4x t_{INT_CK} 11：Reserved, do not program this value
7	ARPE	Auto reload preload enable 0: Disable the shadow register of TIMx_ARR register 1: Enable the shadow register of TIMx_ARR register

Bit	Field	Description
6：5	CMS	Center alignment mode selection 00: Edge alignment mode. Count direction depends on DIR bit 01: Central alignment mode 1. The counter alternatively conducts up and down count. The channel is in output mode. Only in down count, compare interrupt flag bit is set 10: Central alignment mode 2. The counter alternatively conducts up and down count. The channel is in output mode. Only in up count, compare interrupt flag bit is set 11: Central alignment mode 3. The counter alternatively conducts up and down count. The channel is in output mode. In up and down count, compare interrupt flag bit is set Note: During count, the alignment mode change is disabled.
4	DIR	Count direction 0: up count 1: down count Note: When the counter is configured as central alignment mode or encoder mode, the bit is read only.
3	OPM	one-pulse mode 0: Disable one-pulse mode. In case of update event, the counter count continues 1: Enable one-pulse mode. In case of the next update event (clear CEN bit), the counter count stops
2	URS	Update request source This bit is set and cleared by software, select update event source. 0: The event below may generate a update interrupt or DMA request: - Counter overflow/underflow - Set UG bit - Update generation through the slave mode controller 1: Only in counter overflow/underflow, generate update interrupt or DMA request
1	UDIS	Update disable This bit is used to enable or disable the update event 0: Update event (UEV) enabled. 1: Update event disabled. The Update event is not generated, shadow registers keep their value (ARR, PSC, and CCRx). However the counter and the prescaler are reinitialized if the EGR.UG bit is set, the counter is reinitialized if a hardware reset is received from the slave mode controller.
0	CEN	Counter enable 0: Counter disabled 1: Counter enabled Note: External clock, gated mode and encoder mode can work only if the CEN bit has been previously set by software. However trigger mode can set the CEN bit automatically by hardware.

11.5.2 TIMx_CR2 Control Register 2

Address offset: 0x04
Reset value: 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								TI1S	MMS			CCDS	Reserved		
								rw	rw			rw			

Bit	Field	Description
15：8	Reserved	Reserved, must be kept at reset value.
7	TI1S	TI1 selection 0: TIMx_CH1 pin is connected to TI1 input 1: TIMx_CH1, TIMx_CH2 and TIMx_CH3 pins are connected to the TI1 input after XOR combination

Bit	Field	Description
6：4	MMS	Master mode selection These bits control TRGO signal selection, used to select the sync information sent to the slave timers in master mode: 000: Reset TIMx_EGR register UG bit generate one pulse trigger output (TRGO). 001: Enable The Counter Enable signal CNT_EN is used as trigger output (TRGO). It is useful to start several timers at the same time or to control a window in which a slave timer is enable. The Counter Enable signal is generated by a logic OR between CEN control bit and the trigger input when configured in gated mode. When the Counter Enable signal is controlled by the trigger input, there is a delay on TRGO, except if the master/slave mode is selected. 010: Update Update event is selected as TRGO. 011: Capture/compare When a capture or a compare match occurred, send a positive pulse as TRGO. 100: Compare OC1REF signal is used as trigger output (TRGO) 101: Compare OC2REF signal is used as trigger output (TRGO) 110: Compare OC3REF signal is used as trigger output (TRGO) 111: Compare OC4REF signal is used as trigger output (TRGO)
3	CCDS	DMA request source selection 0: when CCx event occurs, send CCx DMA request 1: when update event occurs, send CCx DMA request Note: It's only suitable for the product with the built in DMA
2：0	Reserved	Reserved, must be kept at reset value.

11.5.3 TIMx_SMCR Slave Mode Control Register

Address offset: 0x08
Reset value: 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ETP	ECE	ETPS		ETF				MSM	TS			OCCS	SMS		
rw	rw	rw		rw				rw	rw			rw	rw		

Bit	Field	Description
15	ETP	External trigger polarity The bit selects ETR signal polarity. 0: High level or rising edge is active 1: Low level or falling edge is active Note: It's only suitable for supporting the external trigger product
14	ECE	External clock enable The bit enables the external clock mode 2. 0: external clock mode 2 disabled 1: external clock mode 2 enabled, The counter is clocked by any active edge on the ETRFsignal. Note 1: It's only suitable for supporting the external trigger product. Note 2: Configuring ECE =1 has the same effect with configuring SMS =111 and TS =111. Note 3: TS≠111, reset mode, gate mode and trigger mode can be used together with the external clock mode 2. Note 4: If external clock mode 1 and external clock mode 2 are enabled at the same time, the external clock input is ETRF.
13：12	ETPS	External trigger prescaler The external trigger signal ETRP frequency must be lower than 1/4 of TIMxPCLK frequency. When input the quicker external clock, use the prescaler to reduce ETRP frequency. 00: Prescaler OFF 01: ETRP frequency divided by 2 10: ETRP frequency divided by 4 11: ETRP frequency divided by 8 Note: It's only suitable for supporting the external trigger product

Bit	Field	Description
11：8	ETF	External trigger filter These bits define ETRP signal sampling frequency and ETRP digital filter bandwidth. In fact, the digital filter is an event counter, which generate an output jump after N event. 0000: No filter, f_{DTS} sampling 001: Sampling frequency $f_{sampling}=f_{INT\_CK}$, N =2 0010: Sampling frequency $f_{sampling}=f_{INT\_CK}$, N =4 0011: Sampling frequency $f_{sampling}=f_{INT\_CK}$, N =8 0100: Sampling frequency $f_{sampling}=f_{DTS/2}$, N =6 0101: Sampling frequency $f_{sampling}=f_{DTS/2}$, N =8 0110: Sampling frequency $f_{sampling}=f_{DTS/4}$, N =6 0101: Sampling frequency $f_{sampling}=f_{DTS/2}$, N =8 0110: Sampling frequency $f_{sampling}=f_{DTS/4}$, N =6 0111: Sampling frequency $f_{sampling}=f_{DTS/4}$, N =8 1000: Sampling frequency $f_{sampling}=f_{DTS/8}$, N =6 1001:Sampling frequency $f_{sampling}=f_{DTS/8}$, N =8 1010: Sampling frequency $f_{sampling}=f_{DTS/16}$, N =5 1011: Sampling frequency $f_{sampling}=f_{DTS/16}$, N =6 1100: Sampling frequency $f_{sampling}=f_{DTS/16}$, N =8 1101: Sampling frequency $f_{sampling}=f_{DTS/32}$, N =5 1110: Sampling frequency $f_{sampling}=f_{DTS/32}$, N =6 1111: Sampling frequency $f_{sampling}=f_{DTS/32}$, N =8 Note: It's only suitable for supporting the external trigger product
7	MSM	Master/slave mode 0: No action 1: The effect of an event on the trigger input (TRGI) is delayed to allow a perfect synchronization between the current timer and its slaves (through TRGO). It is useful if we want to synchronize several timers on a single external event.
6：4	TS	Trigger selection Trigger input source selection. 000：Internal trigger 0 (ITR0) 001：Internal trigger 1 (ITR1) 010：Internal trigger 2 (ITR2) 011：Internal trigger 3 (ITR3) 100：TI1 edge detector (TI1F_ED) 101：Timer input after filter 1 (TI1FP1) 110：Timer input after filter 2 (TI2FP2) 111：External trigger input (ETR) More details on ITRx are given in the table below. Note: After the slave mode enable, these bits cannot be changed
3	OCCS	Timer compare output signal(OCxREF) clear selection In PWM mode, clear the compare output signal(OCxREF) 0: External trigger signal as clear signal 1: Comparator output(COMP) as clear signal Note: It's only applicable to products that support external trigger or have a built-in comparator(COMP)
2：0	SMS	Slave mode selection When external signals are selected the active edge of the trigger signal (TRGI) is linked to the polarity selected on the external input. 000: Close slave mode - In case of CEN =1, the prescaler is directly driven by the internal clock. 001: Encoder mode 1 - Based on TI1FP1 level, the counter conducts up/down count along TI2FP2 edge. 010: Encoder mode 2 - Based on TI2FP2 level, the counter conducts up/down count along TI1FP1edge. 011: Encoder mode 3- Based on another input level, the counter conducts up/down count along TI1FP1 and TI2FP2 edge. 100: Reset mode -The rising edge of the selected trigger input (TRGI) reinitialize the counter and generate an update register signal. 101: Gate mode - When the trigger input (TRGI) is high, the counter count begins and generate an update event. When the trigger input turns low, the counter count stops (but without reset) and generate an update event. The counter start and stop are controlled. 110: Trigger mode -The counter starts in the rising edge of the trigger input TRGI (but without reset) and generate an update event. Only the counter start is controlled. 111: External clock mode 1 -The rising edge of the selected trigger input (TRGI) drives the counter and generate an update event. Note: If TI1F_EN is selected as trigger input (TS =100), don't use the gate mode. This is because that EI1F_ED exports a pulse at each TI1F change. However, in the gate mode, it's required to check the trigger input level.

Table 11-5 TIMx internal trigger connection

Slave timer	ITR0	ITR1	ITR2	ITR3
TIM1	-	TIM2_TRGO	TIM14_OC1REF	TIM13_OC1REF
TIM2	TIM1_TRGO	-	TIM14_OC1REF	TIM13_OC1REF

11.5.4 TIMx_DIER DMA/Interrupt Enable Register

Address offset: 0x0C
Reset value: 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	TDE	Res.	CC4DE	CC3DE	CC2DE	CC1DE	UDE	Res.	TIE	Res.	CC4IE	CC3IE	CC2IE	CC1IE	UIE
	rw		rw	rw	rw	rw	rw		rw		rw	rw	rw	rw	rw

Bit	Field	Description
15	Reserved	Reserved, must be kept at reset value.
14	TDE	Trigger DMA request enable 0: Trigger DMA request disable 1: Trigger DMA request enable Note: It's only suitable for the built in DMA product.
13	Reserved	Reserved, must be kept at reset value.
12	CC4DE	Capture/compare 4 DMA request enable 0: Capture/compare 4 DMA request disable 1: Capture/compare 4 DMA request enable Note: It's only suitable for the built in DMA product.
11	CC3DE	Capture/compare 3 DMA request enable 0: Capture/compare 3 DMA request disable 1: Capture/compare 3 DMA request enable Note: It's only suitable for the built in DMA product.
10	CC2DE	Capture/Compare 2 DMA request enable 0: Capture/compare 2 DMA request disable 1: Capture/compare 2 DMA request enable Note: It's only suitable for the built in DMA product.
9	CC1DE	Capture/compare 1 DMA request enable 0: Capture/compare 1 DMA request disable 1: Capture/compare 1 DMA request enable Note: It's only suitable for the built in DMA product.
8	UDE	Update DMA request enable 0: Update DMA request disable 1: Update DMA request enable Note: It's only suitable for the built in DMA product.
7	Reserved	Reserved, must be kept at reset value.
6	TIE	Trigger interrupt enable 0: Break interrupt disable 1: Break interrupt enable
5	Reserved	Reserved, must be kept at reset value.
4	CC4IE	Enable capture/compare 4 interrupt 0: Capture/compare interrupt 4 disable 1: Capture/compare interrupt 4 enable
3	CC3IE	Enable capture/compare 3 interrupt 0: Capture/compare interrupt 3 disable 1: Capture/compare interrupt 3 enable
2	CC2IE	Enable capture/compare 2 interrupt 0: Capture/compare interrupt 2 disable 1: Capture/compare interrupt 2 enable
1	CC1IE	Enable capture/compare 1 interrupt 0: Capture/compare interrupt 1 disable 1: Capture/compare interrupt 1 enable
0	UIE	Enable update interrupt 0: Update event interrupt disable 1: Update event interrupt enable

11.5.5 TIMx_SR Status Register

Address offset: 0x10
Reset value: 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved			CC4OF	CC3OF	CC2OF	CC1OF	Reserved		TIF	Res.	CC4IF	CC3IF	CC2IF	CC1IF	UIF
			r_w0c						r_w0c		r_w0c				

Bit	Field	Description
15：13	Reserved	Reserved, must be kept at reset value.
12	CC4OF	Capture/compare 4 over capture flag Refer to CC1OF description.
11	CC3OF	Capture/compare 3 over capture flag Refer to CC1OF description.
10	CC2OF	Capture/compare 2 over capture flag Refer to CC1OF description.
9	CC1OF	Capture/compare 1 over capture flag Only when the channel 1 is configured as input capture mode and CC1F is configured as 1, and the capture event occurs again. The flag is set by hardware. Write 0 may clear the bit. 0: No over capture 1: Over capture
8：7	Reserved	Reserved, must be kept at reset value.
6	TIF	Trigger interrupt flag In case of trigger event (When the slave mode counter is in any other mode except for the gate mode, detect the active edge in TRGI input terminal, or detect any edge in gate mode), the bit is set by hardware. It's cleared by software. 0: No trigger event occurred 1: Trigger interrupt pending
5	Reserved	Reserved, must be kept at reset value.
4	CC4IF	Capture/compare 4 interrupt flag Refer to CC1IF description
3	CC3IF	Capture/compare 3 interrupt flag Refer to CC1IF description
2	CC2IF	Capture/compare 2 interrupt flag Refer to CC1IF description.
1	CC1IF	Capture/compare 1 interrupt flag Channel 1 in output mode: When the counter value and compare value match, the bit is set by hardware except for the central alignment mode (the bit is set in the central alignment mode according to TIMx_CR1.CMS [1:0]). It's cleared by software. 0: No match 1: TIMx_CNT value and TIMx_CCR1 value match Channel 1 as output mode: In case of capture event, the bit is set by hardware. It's cleared by software or reading TIMx_CCR1 register 0: No input capture occurred 1: Counter value is captured to TIMx_CCR1
0	UIF	Update interrupt flag In case of update event, the bit is set by hardware. It's cleared by software. 0: No update interrupt occurred 1: update interrupt pending This bit is set by hardware when the registers are updated: - In case of TIMx_CR1 register UDIS=0 and REP_CNT=0, when the counter generates overflow/underflow. -In case of TIMx_CR1 register UDIS=0, URS=0, when TIMx_EGR register UG =1. - In case of TIMx_CR1 register UDIS=0, URS=0, when update generation through the slave mode controller.

11.5.6 TIMx_EGR Event Generation Register

Address offset: 0x14
Reset value: 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved									TG	Res.	CC4G	CC3G	CC2G	CC1G	UG
									w		w	w	w	w	w

Bit	Field	Description
15：7	Reserved	Reserved, must be kept at reset value.
6	TG	Trigger generation 0: No action 1: generate a trigger event. TIMx_SR register TIF =1. When enabling the corresponding interrupt and DMA, generate the corresponding interrupt and DMA. It's auto cleared by hardware.
5	Reserved	Reserved, must be kept at reset value.
4	CC4G	Capture/compare 4 generation Refer to CC1G description.
3	CC3G	Capture/compare 3 generation Refer to CC1G description.
2	CC2G	Capture/compare 2 generation Refer to CC1G description.
1	CC1G	Capture/compare 1 generation Refer to CC1G description. This bit is set by software. It generates a capture/compare event, and is auto cleared by hardware. 0: No action 1：Generate a capture/compare event in channel CC1: When the channel CC1 is configured as output: Set CC1IF =1. When enabling the corresponding interrupt and DMA, generate the corresponding interrupt and DMA. When the channel CC1 is configured input: The current value of the counter is captured in TIMx_CCR1 register, set CC1IF =1. When enabling the corresponding interrupt and DMA, generate the corresponding interrupt and DMA. When CC1IF is set, set CC1OF =1.
0	UG	Update event generation 0：No action 1: Initialize the counter, and generate an update event. It's auto cleared by hardware. When selecting the central alignment or up counting mode, the counter is clear; otherwise (down counting mode) the counter auto reload value is loaded. The prescaler counter is cleared at the same time.

11.5.7 TIMx_CCMR1 Capture/Compare Mode Register 1

Address offset: 0x18
Reset value: 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OC2CE	OC2M			OC2PE	OC2FE	CC2S		OC1CE	OC1M			OC1PE	OC1FE	CC1S	
IC2F				IC2PSC		CC2S		IC1F				IC1PSC		CC1S	
rw	rw			rw	rw	rw		rw	rw			rw	rw	rw	

The channel may be used for input (capture mode) or output (compare mode). The channel direction is defined by the corresponding CCxS. The register's bit other than CCxS functions differently in the input mode and output mode. OCxx describes the channel function in the output mode. ICxx describes the channel function in the input mode.

Output compare mode:

Bit	Field	Description
15	OC2CE	Channel 2 output and compare clear enable Refer to OC1CE description.

Bit	Field	Description
14：12	OC2M	Channel 2 output and compare mode Refer to OC1M description.
11	OC2PE	Channel 2 output and compare preload enable Refer to OC1PE description.
10	OC2FE	Channel 2 output and compare quick enable Refer to OC1FE description.
9：8	CC2S	Channel 2 capture/compare 2 selection This bit defines the channel direction and input signal selection. Only when the channel closes, these bits can be written: 00：Channel 2 is configured as output 01：Channel 2 is configured as input, IC2 is mapped on TI2 10：Channel 2 is configured as input, IC2 is mapped on TI1 11：Channel 2 is configured as input, IC2 is mapped on TRC. The mode only works when the internal generator input is selected (TIMx_SMCR register TS bit selection)
7	OC1CE	Channel 1 capture/compare clear enable 0: OC1REF is not affected by ETR 1: When detecting ETR as active level, clear OC1REF
6：4	OC1M	Channel 1 output compare mode The bit has defined the output reference signal OC1REF action. OC1REF has determined OC1 and OC1N value. OC1REF is active at high level. The active level of OC1, OC1N depends on CC1P and CC1NP bit. 000：Freeze. TIMx_CCR1 and TIMx_CNT compare results has no effect on OC1REF. 001: Set as high when configuration. When TIMx_CNT value and TIMx_CCR1 value are same, enforce OC1REF as high level 010: Set as low when configuration. When TIMx_CNT value and TIMx_CCR1 value are same, enforce OC1REF as low level 011: Toggle when match. When TIMx_CCR1=TIMx_CNT, OC1REF toggle. 100: Enforce as low. Enforce OC1REF at low level 101: Enforce as high. Enforce OC1REF at high level 110：PWM mode 1. During up count, in case of TIMx_CNT<TIMx_CCR1, enforce OC1REF is at high level. Or else, it's at low level. During down count, in case of TIMx_CNT > TIMx_CCR1, enforce OC1REF is at low level. Or else, it's at high level. 111：PWM mode 2. During up count, in case of TIMx_CNT<TIMx_CCR1, channel 1 enforce OC1REF is at low level. Or else, it's at high level. During down count, in case of TIMx_CNT > TIMx_CCR1, enforce OC1REF is at high level. Or else, it's at low level. Note 1: In case of LOCK level 3 (TIMx_BDTR register LOCK bit) and CC1S = 00 (the channel is configured as output), the bit cannot be changed. Note 2: In PWM mode 1 or PWM mode 2, only when the compare result changes or it changes over from the freeze mode to PWM mode in the output compare mode, OC1REF level may change.
3	OC1PE	Channel 1 output compare preload enable 0: Disable TIMx_CCR1 register preload function. The value written into TIMx_CCR1 register becomes active immediately 1: Enable TIMx_CCR1 register preload function. Only read and write the preload register. TIMx_CCR1 preload value becomes active at the update event Note 1: When the LOCK level is 3 (TIMx_BDTR register LOCK bit) and CC1S = 00 (when the channel is configured as output), the bit cannot be changed. Note 2: Only in the one-pulse mode (TIMx_CR1 register OPM= 1), it has no influence whether the preload register is set. Under other scenarios, it's required to set the preload register. Otherwise, the follow up action is not certain.
2	OC1FE	Channel 1 output compare quick enable In case of the bit is set 1, when the channel is configured as PWM mode, it speeds up response of the capture/compare output to the trigger time. The output channel deems the active edge of the trigger input signal as one compare match. Therefore, OC is set as compare level, but is irrelevant to the compare result. 0: channel 1 output compare fast enable. 1: channel 1 output compare fast disable.

Bit	Field	Description
1：0	CC1S	Channel 1 capture/compare selection This bit defines the channel direction and input signal selection. Only when the channel closes, these bits can be written: 00：Channel 1 is configured as output 01：Channel 1 is configured as input, IC1 is mapped on TI1 10：Channel 1 is configured as input, IC1 is mapped on TI2 11：Channel 1 is configured as input, IC1 is mapped on TRC. This mode only works when the internal trigger input is selected (TIMx_SMCR register TS bit selection)

Input capture mode:

Bit	Field	Description
15：12	IC2F	Input capture 2 filter Refer to IC1F description
11：10	IC2PSC	Input/capture 2 prescaler Refer to IC1PSC description
9：8	CC2S	Channel 2 capture/compare selection This bit defines the channel direction and input signal selection. Only when the channel closes, these bits can be written: 00：Channel 2 is configured as output 01：Channel 1 is configured as input, IC2 is mapped on TI2 10：Channel 2 is configured as input, IC2 is mapped on TI1 11：Channel 2 is configured as input, IC2 is mapped on TRC. This mode only works when the internal trigger input is selected (TIMx_SMCR register TS bit selection)
7：4	IC1F	Channel 1 input capture filter The digital filter is composed of an event counter. It records an output jump after N input events. These bits define IC1 input signal sampling frequency and digital filter length. 0000: No filter, f_{DTS} sampling 001: Sampling frequency $f_{sampling}$ =f_{INT_CK}, N =2 0010: Sampling frequency $f_{sampling}$ =f_{INT_CK}, N =4 0011: Sampling frequency $f_{sampling}$ =f_{INT_CK}, N =8 0100: Sampling frequency $f_{sampling}$ =$f_{DTS/2}$, N =6 0101: Sampling frequency $f_{sampling}$ =$f_{DTS/2}$, N =8 0110: Sampling frequency $f_{sampling}$ =$f_{DTS/4}$, N =6 0111: Sampling frequency $f_{sampling}$ =$f_{DTS/4}$, N =8 1000: Sampling frequency $f_{sampling}$ =$f_{DTS/8}$, N =6 1001:Sampling frequency $f_{sampling}$ =$f_{DTS/8}$, N =8 1010: Sampling frequency $f_{sampling}$ =$f_{DTS/16}$, N =5 1011: Sampling frequency $f_{sampling}$ =$f_{DTS/16}$, N =6 1100: Sampling frequency $f_{sampling}$ =$f_{DTS/16}$, N =8 1101: Sampling frequency $f_{sampling}$ =$f_{DTS/32}$, N =5 1110: Sampling frequency $f_{sampling}$ =$f_{DTS/32}$, N =6 1111: Sampling frequency $f_{sampling}$ =$f_{DTS/32}$, N =8
3：2	IC1PSC	Channel 1 input/capture prescaler This bit defines the ratio of the prescaler acting on IC1.The prescaler is reset as soon as CC1E=0(TIMx_CCER register). 00: No prescaler, the capture input detects each edge to trigger one capture 01: Trigger one capture for every 2 events 10: Trigger one capture for every 4 events 11: Trigger one capture for every 8 events
1：0	CC1S	Channel 1 capture/compare selection This bit defines the channel direction and input signal selection. Only when the channel closes, these bits can be written: 00: Channel 1 is configured as output 01: Channel 1 is configured as input, IC1 is mapped on TI1 10: Channel 1 is configured as input, IC1 is mapped on TI2 11: Channel 1 is configured as input, IC1 is mapped on TRC. This mode only works when the internal trigger input is selected (TIMx_SMCR register TS bit selection)

11.5.8 TIMx_CCMR2 Capture/Compare Mode Register 2

Address offset: 0x1C

Reset value: 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OC4CE	OC4M			OC4PE	OC4FE	CC4S		OC3CE	OC3M			OC3PE	OC3FE	CC3S	
IC4F				IC4PSC		CC4S		IC3F				IC3PSC		CC3S	
rw	rw			rw	rw	rw		rw	rw			rw	rw	rw	

Compare output mode:

Bit	Field	Description
15	OC4CE	Channel 4 output compare clear enable Refer to OC3CE description
14：12	OC4M	Channel 4 output compare mode Refer to OC3M description
11	OC4PE	Channel 4 output compare preload enable Refer to OC3PE description
10	OC4FE	Channel 4 output compare rapid enable Refer to OC3FE description
9：8	CC4S	Channel 4 capture/compare selection This bit defines the channel direction and input signal selection. Only when the channel closes, these bits can be written: 00: Channel 4 is configured as output 01: Channel 4 is configured as input, IC4 is mapped on TI4 10: Channel 4 is configured as input, IC4 is mapped on TI3 11: Channel 4 is configured as input, IC4 is mapped on TRC. This mode only works when the internal trigger input is selected (TIMx_SMCR register TS bit selection
7	OC3CE	Channel 3 output compare clear enable 0: OC3REF is not affected by ETR input 1: When detecting ETR input is at active level, clear OC3REF
6：4	OC3M	Channel 3 output compare mode The bit has defined the output reference signal OC3REF action. OC3REF has determined OC3 and OC3N value. OC3REF is active at high level. The active level of OC3, OC3N depends on CC3P and CC3NP bit. 000：Freeze. TIMx_CCR3 and TIMx_CNT compare results has no effect on OC3REF. 001: Set as high when configuration. When TIMx_CNT value and TIMx_CCR3 value are same, enforce OC3REF as high level 010: Set as low when configuration. When TIMx_CNT value and TIMx_CCR3 value are same, enforce OC3REF as low level 011: Toggle when match. When TIMx_CCR3=TIMx_CNT, OC3REF toggle. 100: Enforce as low. Enforce OC3REF at low level 101: Enforce as high. Enforce OC3REF at high level 110: PWM mode 1. During up count, in case of TIMx_CNT<TIMx_CCR3, enforce OC3REF as high level, otherwise its low level; during down count, in case of TIMx_CNT > TIMx_CCR3, enforce OC3REF as low level, otherwise its high level. 111: PWM mode 2. During up count, in case of TIMx_CNT<TIMx_CCR3, enforce OC3REF as low level, otherwise its high level; during down count, in case of TIMx_CNT > TIMx_CCR3, enforce OC3REF as high level, otherwise its low level. Note 1: In case of LOCK level 3 (TIMx_BDTR register LOCK bit) and CC3S = 00 (the channel is configured as output), the bit cannot be changed. Note 2: In PWM mode 1 or PWM mode 2, only when the compare result changes or it changes over from the freeze mode to PWM mode in the output compare mode, OC3REF level may change.
3	OC3PE	Channel 3 output compare preload enable 0: Disable TIMx_CCR3 register preload function, the value written into TIMx_CCR3 register becomes active immediately 1: Enable TIMx_CCR3 register preload function. The read and write only works on the preload register. TIMx_CCR3 preload value becomes active when the update event is available Note 1: When the LOCK level is set 3 (TIMx_BDTR register LOCK bit) and CC3S =0 (the channel is configured as output), the bit cannot be changed. Note 2: Only in the one-pulse mode (TIMx_CR1 register OPM= 1), it's not required to set the preload register. Under other scenarios, it's required to set the preload register. Otherwise, the follow up action is not certain.

Bit	Field	Description
2	OC3FE	Channel 3 output compare quick enable In case of the bit 1, when the channel is configured as PWM mode, it speeds up response of the capture/compare output to the trigger time. The output channel deems the active edge of the trigger input signal as one compare match. Therefore, OC is set as compare level, but is irrelevant to the compare result. 0: channel 3 output compare fast disable. 1: channel 3 output compare fast enable.
1：0	CC3S	Channel 3 capture/compare selection The bit has defined the channel direction and input signal selection. Only when the channel closes, these bits can be written: 00: Channel 3 is configured as input 01: Channel 3 is configured as input, IC3 is mapped on TI3 10: Channel 3 is configured as input, IC3 is mapped on TI4 11: Channel 3 is configured as input, IC3 is mapped on TRC. This mode only works when the internal trigger input is selected (TIMx_SMCR register TS bit selection)

Input capture mode:

Bit	Field	Description
15：12	IC4F	Input capture 4 filter Refer to IC3F description
11：10	IC4PSC	Input/capture 4 prescaler Refer to IC3PSC description
9：8	CC4S	Channel 4 capture/compare selection This bit defines the channel direction and input signal selection. Only when the channel closes, these bits can be written: 00：Channel 4 is configured as output 01：Channel 4 is configured as input, IC4 is mapped on TI4 10：Channel 4 is configured as input, IC2 is mapped on TI3 11：Channel 4 is configured as input, IC4 is mapped on TRC. This mode only works when the internal trigger input is selected (TIMx_SMCR register TS bit selection)
7：4	IC3F	Channel 3 input capture filter The digital filter is composed of an event counter. It records an output jump after N input events. These bits define IC1 input signal sampling frequency and digital filter length. 0000: No filter, f_{DTS} sampling 001: Sampling frequency $f_{sampling}$ =f_{INT_CK}, N =2 0010: Sampling frequency $f_{sampling}$ =f_{INT_CK}, N =4 0011: Sampling frequency $f_{sampling}$ =f_{INT_CK}, N =8 0100: Sampling frequency $f_{sampling}$ =$f_{DTS/2}$, N =6 0101: Sampling frequency $f_{sampling}$ =$f_{DTS/2}$, N =8 0110: Sampling frequency $f_{sampling}$ =$f_{DTS/4}$, N =6 0111: Sampling frequency $f_{sampling}$ =$f_{DTS/4}$, N =8 1000: Sampling frequency $f_{sampling}$ =$f_{DTS/8}$, N =6 1001:Sampling frequency $f_{sampling}$ =$f_{DTS/8}$, N =8 1010: Sampling frequency $f_{sampling}$ =$f_{DTS/16}$, N =5 1011: Sampling frequency $f_{sampling}$ =$f_{DTS/16}$, N =6 1100: Sampling frequency $f_{sampling}$ =$f_{DTS/16}$, N =8 1101: Sampling frequency $f_{sampling}$ =$f_{DTS/32}$, N =5 1110: Sampling frequency $f_{sampling}$ =$f_{DTS/32}$, N =6 1111: Sampling frequency $f_{sampling}$ =$f_{DTS/32}$, N =8
3：2	IC3PSC	Channel 3 input/capture prescaler This bit defines the ratio of the prescaler acting on IC3.The prescaler is reset as soon as CC3E=0(TIMx_CCER register). 00: No prescaler, the capture input detects each edge to trigger one capture 01: Trigger one capture for every 2 events 10: Trigger one capture for every 4 events 11: Trigger one capture for every 8 events

Bit	Field	Description
1：0	CC3S	Channel 3 capture/compare selection This bit defines the channel direction and input signal selection. Only when the channel closes, these bits can be written: 00: Channel 3 is configured as output 01: Channel 3 is configured as input, IC3 is mapped on TI3 10: Channel 3 is configured as input, IC3 is mapped on TI4 11: Channel 3 is configured as input, IC3 is mapped on TRC. This mode only works when the internal trigger input is selected (TIMx_SMCR register TS bit selection)

11.5.9 TIMx_CCER Capture/Compare Enable Register

Address offset: 0x20
Reset value: 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CC4NP	Res.	CC4P	CC4E	CC3NP	Res.	CC3P	CC3E	CC2NP	Res.	CC2P	CC2E	CC1NP	Res.	CC1P	CC1E
rw		rw	rw	rw		rw	rw	rw		rw	rw	rw		rw	rw

Bit	Field	Description
15	CC4NP	Channel 4 input/capture complementary output polarity Refer to CC1NP description
14	Reserved	Reserved, must be kept at reset value.
13	CC4P	Channel 4 capture/compare output polarity Refer to CC1P description.
12	CC4E	Channel 4 capture/compare enable Refer to CC1E description
11	CC3NP	Channel 3 input/capture complementary output polarity Refer to CC1NP description
10	Reserved	Reserved, must be kept at reset value.
9	CC3P	Channel 3 input capture output polarity Refer to CC1P description
8	CC3E	Channel 3 input/capture output enable Refer to CC1E description
7	CC2NP	Channel 2 input/capture complementary output polarity Refer to CC1NP description
6	Reserved	Reserved, must be kept at reset value.
5	CC2P	Channel 2 input capture output polarity Refer to CC1P description
4	CC2E	Channel 2 input/capture output enable Refer to CC1E description
3	CC1NP	Channel 1 input/capture complementary output polarity When channel 1 is configured as output, this bit has defined the input signal polarity: 0: OC1N is active at high level 1: OC1N is active at low level When channel 1 is configured as input, CC1P/CC1NP match use has defined the input signal polarity and level. Details are given in the ICx polarity/level selection table. Note: When LOCK level (TIMx_BDTR register LCCK bit) is set 3 or 2 and CC1S = 00 (channel is configured as output), the bit cannot be changed.
2	Reserved	Reserved, must be kept at reset value.
1	CC1P	Channel 1 input/capture output polarity When channel 1 is configured as output, the bit defines the output signal polarity: 0: OC1 is active at high level 1: OC1 is active at low level When channel 1 is configured as input, CC1P/CC1NP match use has defined the input signal polarity and level. Details are given in the ICx polarity/level selection table. Note: When LOCK level (TIMx_BDTR register LCCK) is set 3 or 2, this bit cannot be changed.

Bit	Field	Description
0	CC1E	Channel 1 input/capture output enable When channel 1 is configured as output: 0: Close. OC1 output disable 1: Enable. The output level relies on the values in MOE, OSSI, OSSR, OIS1, OIS1N and CC1NE. CC1channel is configured as input: This bit determines whether the input capture function is enabled. 0：Capture disable 1：Capture enable

In input mode, ICx polarity/level selection is given in the Table below:

Table 11-6 ICx polarity/level selection

CCxP	CCxNP	ICx polarity/level
0	0	Rising edge active/high level active
1	0	Falling edge active/low level active
1	1	Rising edge or falling edge active/low level active
0	1	Reserved

11.5.10 TIMx_CNT Counter

Address offset: 0x24
Reset value: 0x0000 0000

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

Bit	Field	Description
31：0	CNT	Counter value

11.5.11 TIMx_PSC Prescaler

Address offset: 0x28
Reset value: 0x0000

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

Bit	Field	Description
15：0	PSC	Prescaler value Counter clock frequency (ck_cnt) =f_{CK_PSC}/（PSC+1） In case of update event, PSC value is loaded into the current prescaler register.

11.5.12 TIMx_ARR Auto Reload Register

Address offset: 0x2C
Reset value: 0x0000 0000

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

Bit	Field	Description
31：0	ARR	Auto reload value These bits define the auto reload value of the counter. When auto reload value is 0, the counter doesn't work.

11.5.13 TIMx_CCR1 Capture/Compare Register 1

Address offset: 0x34
Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CCR1															
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR1															
rw															

Bit	Field	Description
31：0	CCR1	Channel 1 capture/compare value When channel 1 is configured as input: CCR1 value determines the counter value of the previous capture event (this register is only readable). When channel 1 is configured as output: If the preload function is not selected in TIMx_CCMR1 register (OC1PE bit), the write value is immediately transmitted to the current capture/compare shadow register. Or else, the write value is only loaded into the capture/compare register in case of update event. The current capture/compare shadow register is involved in the compare with the counter TIMx_CNT, and the compare result is reflected to the output signal of OC1 port.

11.5.14 TIMx_CCR2 Capture/Compare Register 2

Address offset: 0x38
Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CCR2															
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR2															
rw															

Bit	Field	Description
31：0	CCR2	Channel 2 capture/compare value Refer to CCR1 description.

11.5.15 TIMx_CCR3 Capture/Compare Register 3

Address offset: 0x3C
Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CCR3															
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR3															
rw															

Bit	Field	Description
31：0	CCR3	Channel 3 capture/compare value Refer to CCR1 description.

11.5.16 TIMx_CCR4 Capture/Compare Register 4

Address offset: 0x40
Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CCR4															
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR4															
rw															

Bit	Field	Description
31：0	CCR4	Channel 4 capture/compare value Refer to CCR1 description.

11.5.17 TIMx_DCR DMA Control Register

Address offset: 0x48
Reset value: 0x0000
Note: This register is only applicable for the built in DMA product. Details are given in the 10.4.8 DMA.

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

Bit	Field	Description
15：13	Reserved	Reserved, must be kept at reset value.
12：8	DBL	DMA continuous transmission length These bits define the DMA access register number in continuous mode 00000: 1 transmission 00001: 2 transmission 00010: 3 transmission .... 10001：18 transmission
7：5	Reserved	Reserved, must be kept at reset value.

Bit	Field	Description
4：0	DBA	DMA base address These bits define the first address of DMA access TIMx_DMAR register in continuous mode. DBA defines the offset value from the TIMx_CR1 register address： 00000：TIMx_CR1 00001：TIMx_CR2 00010：TIMx_SMCR .....

11.5.18 TIMx_DMAR DMA Address Register of Continuous Mode

Address offset: 0x4C
Reset value: 0x0000
Note: This register is only applicable for the built in DMA product. Details are given in the 10.4.8 DMA.

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

Bit	Field	Description
15：0	DMAB	DMA register for burst accesses DMA continuous transmission register's write to TIMx_DMAR register will lead to access of the register in the addresses below： TIMx_CR1 address + DBA + DMA index， Where TIMx_CR1 address is the address of TIMx_CR1 register; DBA is the base address defined by TIMx_DCR register; DMA index is the offset auto controlled by DMA. It depends on the DBL defined by TIMx_DCR register.

11.5.19 TIMx_OR Input Option Register

Address offset: 0x50
Reset value: 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								TI4_RMP		Reserved				ETR_RMP	
								rw						rw	

Bit	Field	Description
15:8	Reserved	Reserved, must be kept at reset value.
7:6	TI4_RMP	00：GPIO input or COMP input 01：LSI clock input 10：reserved 11：reserved
5:2	Reserved	Reserved, must be kept at reset value.
1:0	ETR_RMP	00：GPIO input Others reserved

12 TIM6 Basic Timer

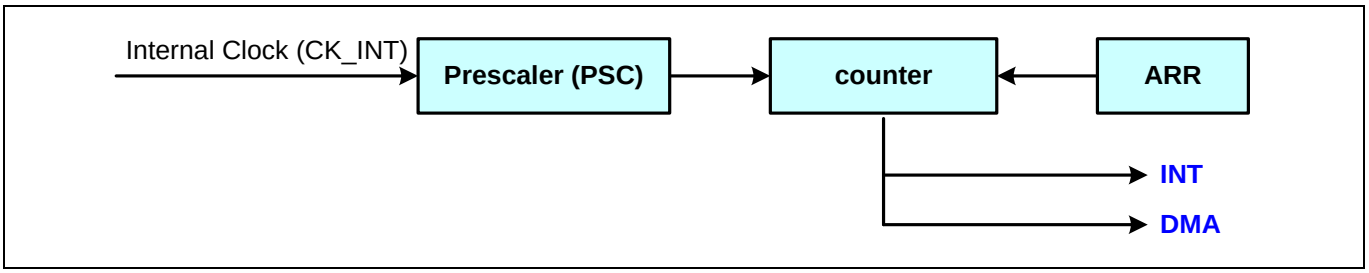
16 bit Basic Timer include TIM6 and TIM7， they have the same function, this product only equipped with TIM6, which are uniformly described as TIMx in this chapter. Some diagrams take TIM6 as an example for illustration.

12.1 Overview

TIMx is composed of a 16 bit real-time programmable prescaler and a 16 bit auto reload counter, and can conveniently provide the counting and timing function. The counter is driven by a programmable prescaler.

12.2 Function block diagram

Figure 12-1 TIMx structure



The figure above is TIMx structure block diagram.

12.3 Main characteristics

- 16 bit real-time programmable prescaler, division factor: 1-65536
- 16 bit auto reload counter (counting direction: up)
- Interrupt/DMA request event: update event

12.4 Function description

12.4.1 Clock

12.4.1.1 Clock selection

The counter clock is provided by the internal clock (INT_CK).

12.4.1.2 Time base unit

TIMx time base unit mainly includes: Counter register (TIMx_CNT), prescaler register (TIM1_PSC) and auto reload register (TIMx_ARR).

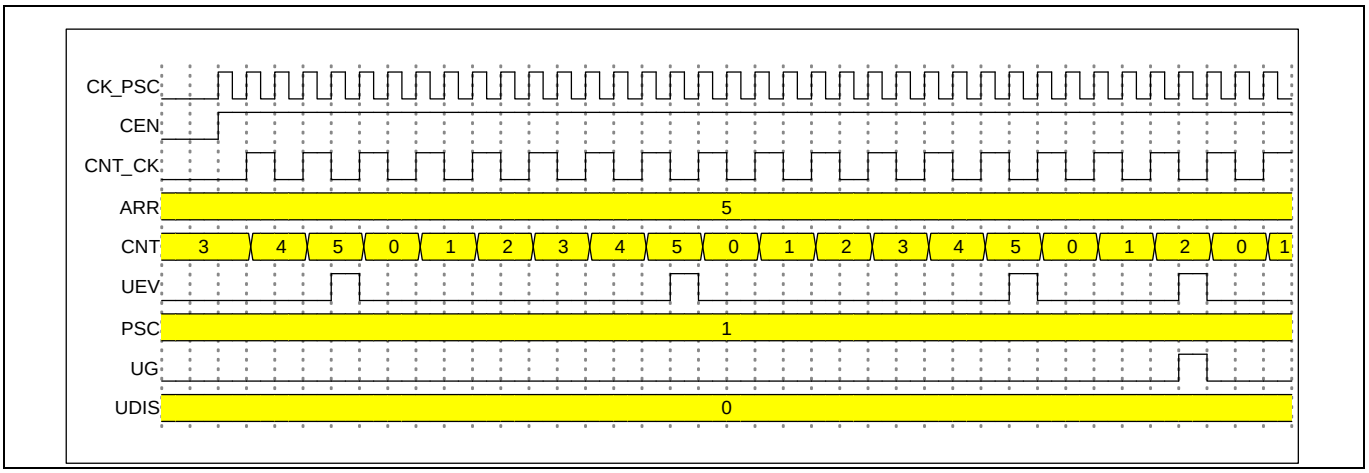
The counter unit is composed of a 16 bit counter with its related auto reload register. The counter can count up.

The counter clock is provided by prescaler. The prescaler is composed of the prescaler counter with is related register. The division factor is 1-65536. It can write any time, and gets valid at the next

update.

The auto preload register has a 16 bit shadow register with preload function. Set TIMx_CR1 register ARPE bit to select the content of ARR register are transferred into the shadow register permanently or at each update event.

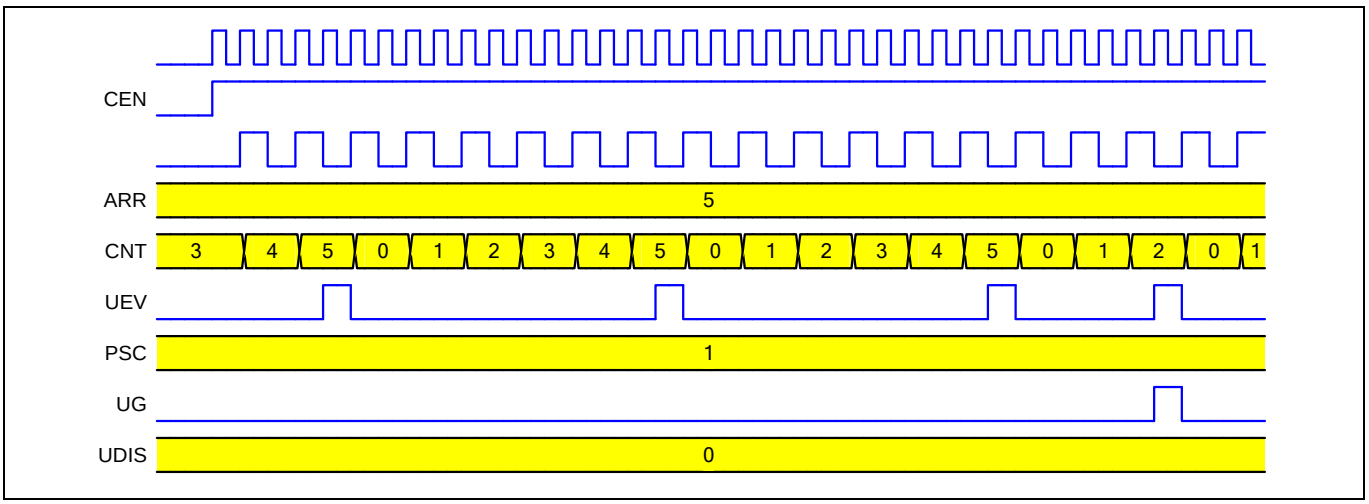
Figure 12-2 Auto preload



12.4.1.3 Counter modes

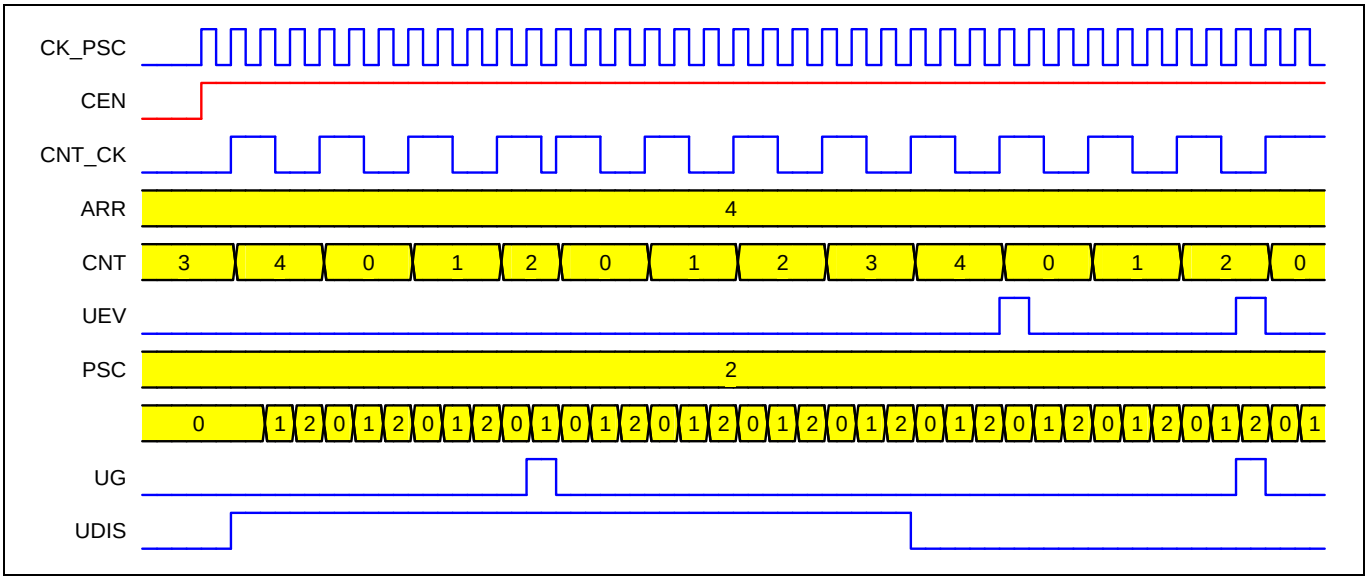
TIMx only supports the up-counting mode. Enable TIMx_CR1 register CEN, and the counter begins up count from 0 toTIMx_ARR value and generates a counter overflow event (update event). The counter begins up count again from 0. Set TIMx_EGR register UG =1, also generate an update event.

Figure 12-3 up counting mode (UDIS=0)



Configuring TIMx_CR1 register UDIS=1, disable update event. In case of counter overflow event, the update event is not generated. Configuring UG=1, the update event is not generated. However, the counter and prescaler counter will initialize, and begin up count from 0.

Figure 12-4 up counting mode (UDIS =1, disable update event)



Note: In case of update event

- ARR register value is loaded into ARR shadow register.
- Prescaler preload value becomes active.

12.4.2 Debug mode

In the debug mode, configure DBG_CR register DBG_TIMx_STOP=1,TIMx counter stops counting. (Details in the debug section)

12.4.3 Interrupt

TIMx only provides the update interrupt. When the update interrupt enable bit is set, generate the corresponding interrupt.

Table 12-1 List of interrupt events

Interrupt event	Flag bit	Enable bit
update interrupt	UIF	UIE

12.4.4 DMA

TIMx can generate the DMA request in case of an update event.

12.5 Register

Table 12-2 TIMx register overview

Offset	Acronym	Register Name	Reset
0x00	TIMx_CR1	Control Register 1	0x0000
0x0C	TIMx_DIER	DMA/Interrupt Enable Register (DMA is only suitable for the chip with built in DMA)	0x0000
0x10	TIMx_SR	Status Register	0x0000
0x14	TIMx_EGR	Event Generation Register	0x0000
0x24	TIMx_CNT	Counter	0x0000

Offset	Acronym	Register Name	Reset
0x28	TIMx_PSC	Prescaler	0x0000
0x2C	TIMx_ARR	Auto Reload Register	0x0000

12.5.1 TIMx_CR1 Control Register 1

Address offset: 0x0
Reset value: 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								ARPE	Reserved			OPM	URS	UDIS	CEN
								rw				rw	rw	rw	rw

Bit	Field	Description
15：8	Reserved	Reserved, must be kept at reset value.
7	ARPE	Auto reload preload enable 0: Disable the shadow register of TIMx_ARR register 1: Enable the shadow register of TIMx_ARR register
6：4	Reserved	Reserved, must be kept at reset value.
3	OPM	One-pulse mode 0: Disable One-pulse mode. In case of update event, the counter count continues 1: Enable one-pulse mode. In case of the next update event (clear CEN bit), the counter count stops
2	URS	Update request source The software is configured in the bit, select update event source. 0：The event below may generate a update interrupt or DMA request: - Counter overflow - Set UG bit 1: Only in counter overflow, generate a update interrupt or DMA request
1	UDIS	Update disable This bit is used to enable or disable the update event 0: Enable update event (UEV) 1: Disable update event. The Update event is not generated, shadow registers keep their value (ARR, PSC, and CCRx). However the counter and the prescaler are reinitialized if the EGR.UG bit is set.
0	CEN	Counter enable 0: Counter disabled 1: Counter enabled

12.5.2 TIMx_DIER DMA/Interrupt Enable Register

Address offset: 0x0C
Reset value: 0x0000

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

Bit	Field	Description
15：9	Reserved	Reserved, must be kept at reset value.
8	UDE	Update DMA request enable 0: Update DMA request disable 1: Update DMA request enable Note: It's only suitable for the built in DMA product.
7：1	Reserved	Reserved, must be kept at reset value.
0	UIE	Update interrupt enable 0: Update event interrupt disable 1: Update event interrupt enable

12.5.3 TIMx_SR Status Register

Address offset: 0x10
Reset value: 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															UIF
															r_w0c

Bit	Field	Description
15：1	Reserved	Reserved, must be kept at reset value.
0	UIF	Update interrupt flag In case of update event, the bit is set by hardware. It's cleared by software. 0: No update interrupt occurred 1: update interrupt pending This bit is set by hardware when the registers are updated: - In case of TIMx_CR1 register UDIS=0, when the counter generates overflow. -In case of TIMx_CR1 register UDIS=0, URS=0, when TIMx_EGR register UG =1.

12.5.4 TIMx_EGR Event Generation Register

Address offset: 0x14
Reset value: 0x0000

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

Bit	Field	Description
15：1	Reserved	Reserved, must be kept at reset value.
0	UG	Update generation event 0: No action 1: Initialize counter, and generate an update event. It's auto cleared by the hardware.

12.5.5 TIMx_CNT Counter

Address offset: 0x24
Reset value: 0x0000

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

Bit	Field	Description
15：0	CNT	Counter value

12.5.6 TIMx_PSC Prescaler

Address offset: 0x28
Reset value: 0x0000

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

rw

Bit	Field	Description
15：0	PSC	Prescaler value Counter clock frequency (ck_cn) =$f_{CK\_PSC/（PSC+1）}$ In case of update event, PSC value is loaded into the current prescaler register.

12.5.7 TIMx_ARR Auto Reload Register

Address offset: 0x2C
Reset value: 0x0000

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

Bit	Field	Description
15：0	ARR	Auto reload value These bits define the auto reload value of the counter. When auto reload value is 0, the counter doesn't work.

13 TIM13/14 Basic Timer

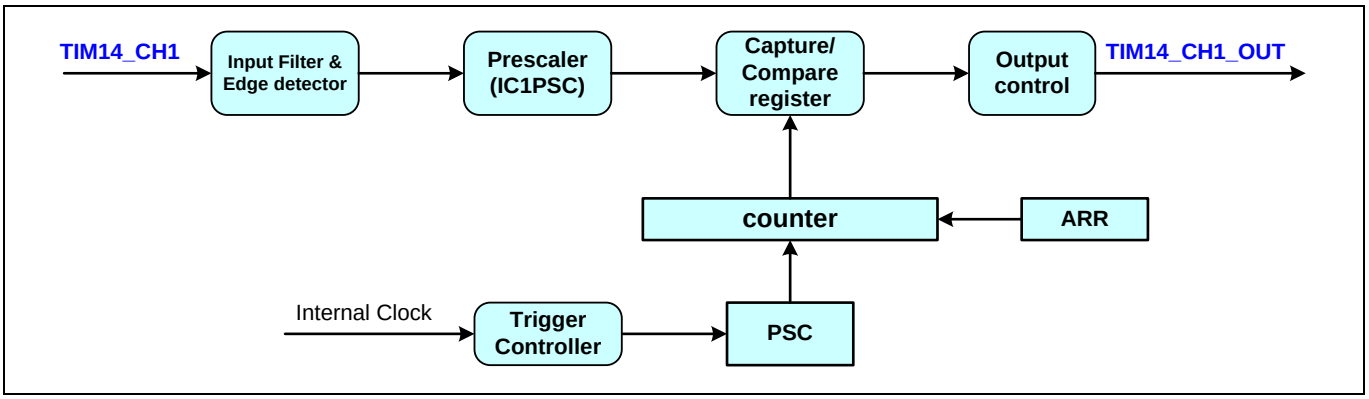
16 bit Basic Timer include TIM13 、TIM14，they have the same function, this product is equipped with TIM13 and TIM14, which are uniformly described as TIMx in this chapter. Some diagrams take TIM14 as an example for illustration.

13.1 Overview

TIMx is composed of a 16 bit real-time programmable prescaler and a 16-bit auto reload counter, and can conveniently provide with the counting and timing function. The counter is driven by a programmable prescaler. The basic timer can be used for a variety of purposes, such as input capture function (measuring input signal pulse width or frequency, etc.), output compare function (PWM output, compare output, etc.)

13.2 Function block diagram

Figure 13-1 TIMx block diagram



TIMx structure block diagram is mainly composed of input capture unit, output compare unit, and counter unit.

13.3 Main characteristics

- 16 bit real-time programmable prescaler, division factor: 1-65536.
- 16 bit auto reload counter (counting direction: up).
- Input capture: measure input signal pulse width or period.
- Output compare (control output waveform or indicator counter already finishes timing).
- One-pulse mode.
- PWM output (edge alignment mode)
- Interrupt/DMA request event: update event, input capture, output compare.

13.4 Function description

13.4.1 Clock

13.4.1.1 Clock selection

The clock of the counter is provided by the internal clock (INT_CK).

13.4.1.2 Time base unit

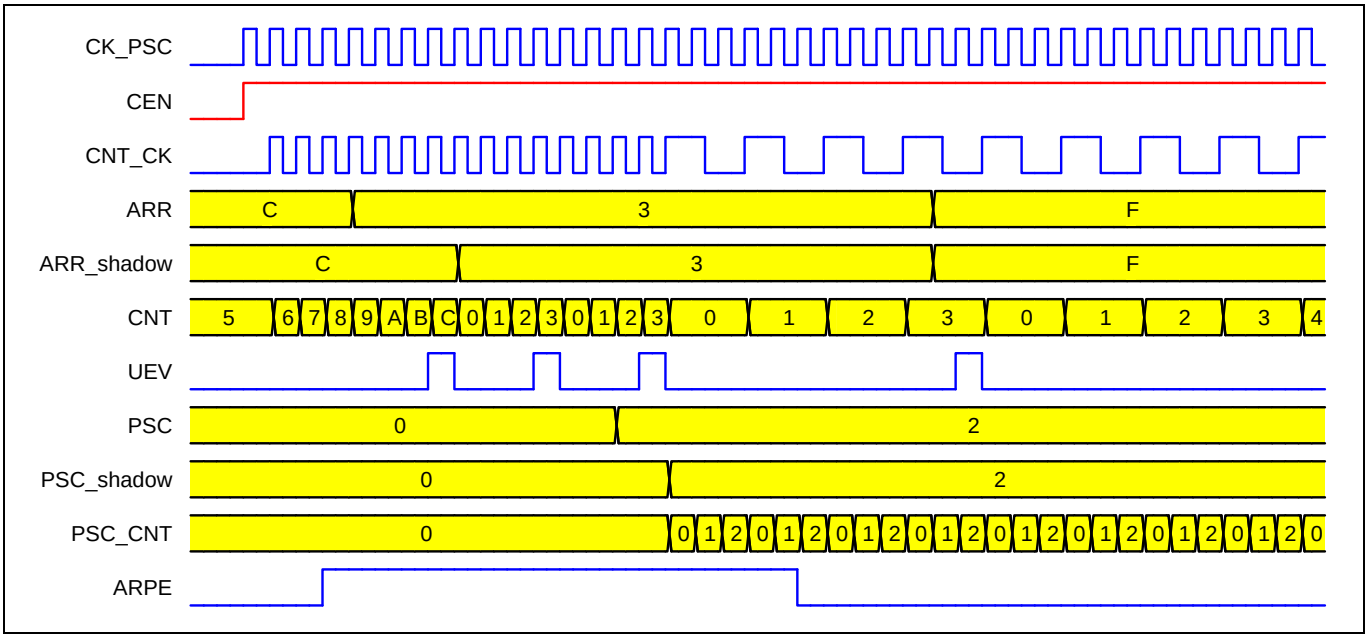
TIMx time base unit mainly includes: Counter register (TIMx_CNT), prescaler register (TIMx_PSC), auto reload register (TIMx_ARR).

The counter unit is composed of a 16 bit counter with its related auto reload register, the counter can count up.

The counter clock is provided by prescaler. The prescaler is composed of the prescaler counter with its related register. The division factor is 1-65536. It can write any time, and gets active at the next update.

The auto preload register has a 16 bit shadow register with preload function. Set TIMx_CR1 register ARPE bit to select the content of ARR register are transferred into the shadow register permanently or at each update event.

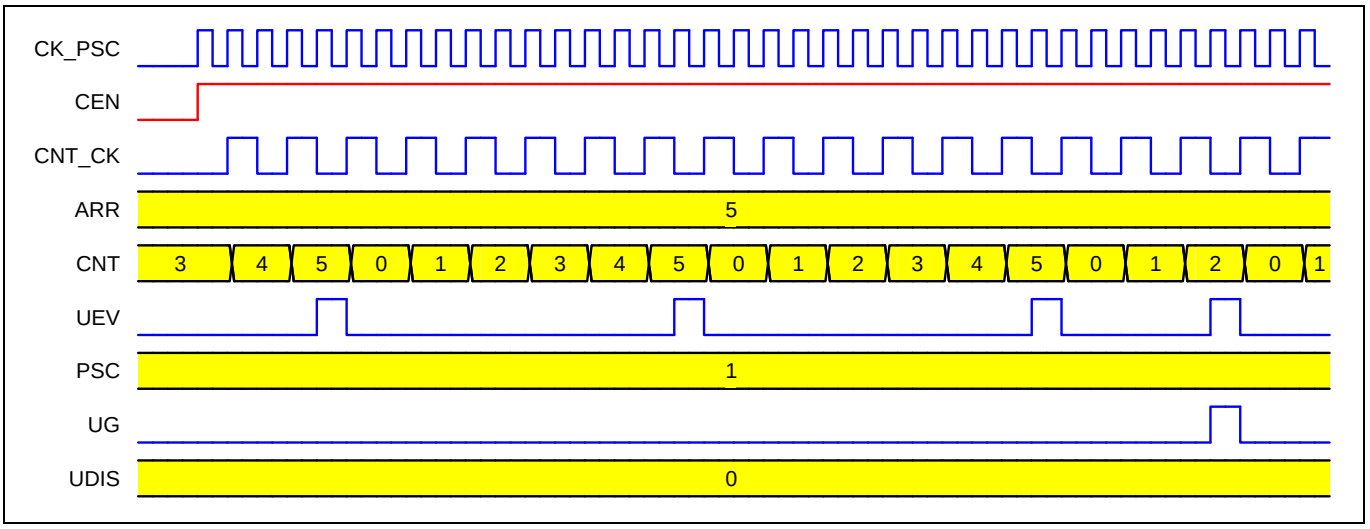
Figure 13-2 Auto preload



13.4.1.3 Counter modes

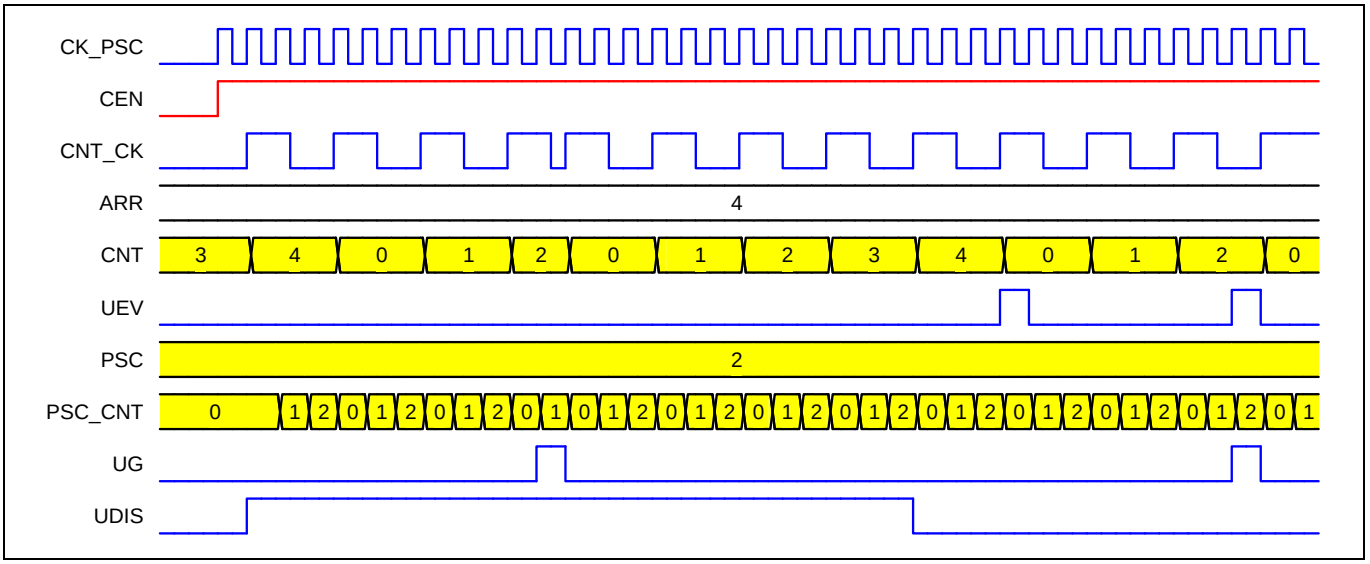
TIMx supports only the up counting mode. Enable TIMx_CR1 register CEN, and the counter begins up count from 0 to TIMx_ARR value and generates a counter overflow event (update). The counter begins up count again from 0. Set TIMx_EGR register UG =1, also generate an update event.

Figure 13-3 Up counting mode（UDIS=0）



Configuring TIMx_CR1 register UDIS=1, disable update event. In case of counter overflow event, the update event is not generated. Configuring UG=1, the update event is not generated. However, the counter and prescaler counter will initialize, and begin up count from 0.

Figure 13-4 Up counting mode (UDIS =1, disable update event)



Note: In case of update event:

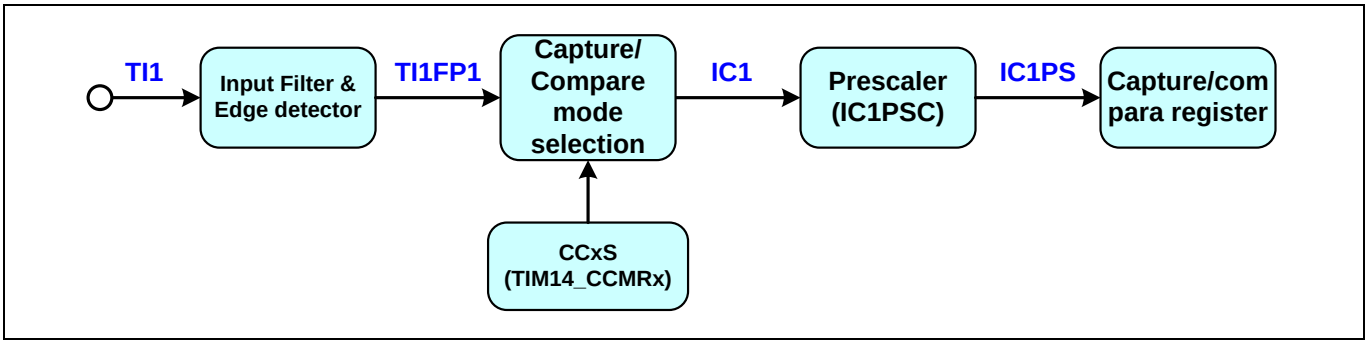
- ARR register value is loaded into ARR shadow register.
- Prescaler preload value becomes active.

13.4.2 Input capture

13.4.2.1 Input capture

The input capture is composed of digital filter, edge detector, prescaler, capture compare register, etc. Its block diagram:

Figure 13-5 TIMx input capture block diagram



Configure TIMx_CCMR1 register IC1F, set the filter width of a digital filter (filter sampling frequency and digital filter width are shown in the table below). When the input signal width is greater than the filter width, the input signal is active. When the digital filter samples the input signal of the input pin TI1, generate a filtered signal TI1F. Through the edge detector with the optional polarity, generate an effective signal IC1. This signal generates IC1PS via the prescaler for trigger input capture event.

Table 13-1 Digital filter width and IC1F correlation

IC1F[3：0]	Sample frequency and filter length	IC1F[3：0]	Sample frequency and filter length
0000	No filter, sample by f_{DTS}	1000	Sampling frequency $f_{sampling}=f_{DTS}$ /8, N=6
0001	Sampling frequency fsampling=f_{INT_CK}, N=2	1001	Sampling frequency $f_{sampling}=f_{DTS}$ /8, N=8
0010	Sampling frequency fsampling=f_{INT_CK}, N=4	1010	Sampling frequency $f_{sampling}=f_{DTS}$ /16，N=5
0011	Sampling frequency fsampling=f_{INT_CK}, N=8	1011	Sampling frequency $f_{sampling}=f_{DTS}$/16, N=6
0100	Sampling frequency fsampling=f_{DTS} /2, N=6	1100	Sampling frequency $f_{sampling}=f_{DTS}$ /16, N=8
0101	Sampling frequency fsampling=f_{DTS}/2, N=8	1101	Sampling frequency $f_{sampling}=f_{DTS}$ /32, N=5
0110	Sampling frequency $f_{sampling}=f_{DTS}$ /4, N=6	1110	Sampling frequency $f_{sampling}=f_{DTS}$ /32, N=6
0111	Sampling frequency $f_{sampling}=f_{DTS}$ /4, N=8	1111	Sampling frequency $f_{sampling}=f_{DTS}$ /32, N=8

In the input capture mode, when detecting the active edge of signal IC1, the counter's current value is locked in the corresponding shadow register, and then loaded into the corresponding capture/compare register. When enabling interrupt or DMA, and producing the capture event, generate the corresponding interrupt or DMA request. At the capture event, the capture flag bit CC1IF of the status register (TIMx_SR) will be set. Configure CC1IF =0 or read TIMx_CCR1 data, clear CC1IF flag bit. When CC1IF is not cleared, next input capture event generated, the repeat capture flag CC1OF will be set. Configure CC1OF =0, can clear CC1OF flag bit.

For example, Sample TI1 input signal's active edge. At TI1 rising edge, capture the current counter value and lock it in the TIMx_CCR1 register, procedure steps as follows:

1. Configure TIMx_CCMR1 register CC1S=01, CC1 channel is configured as input, and IC1 is mapped on TI1.
2. Configure TIMx_CCMR1 register IC1F [3:0], configure the filter width of digital filter (configure upon need).
3. Configure TIMx_CCER register CC1P=0, select capture occurs in the rising edge of TI1 signal.
4. Configure TIMx_CCMR1 register IC1PSC [1:0], select the prescaler factor.
5. Configure TIMx_CCER register CC1E = 1, channel 1 capture enable.
6. Configure TIMx_DIER register CC1IE=1, enable channel 1 interrupt request; if the chip has the built in DMA, configure CC1DE=1 of the TIMx_DIER register to allow the DMA request on capture/compare channel 1.

Note：

- When the channel is configured as input mode, TIMx_CCR1 register becomes read only.
- In case of more than two continuous captures and CC1IF flag is not cleared, the repeat capture flag

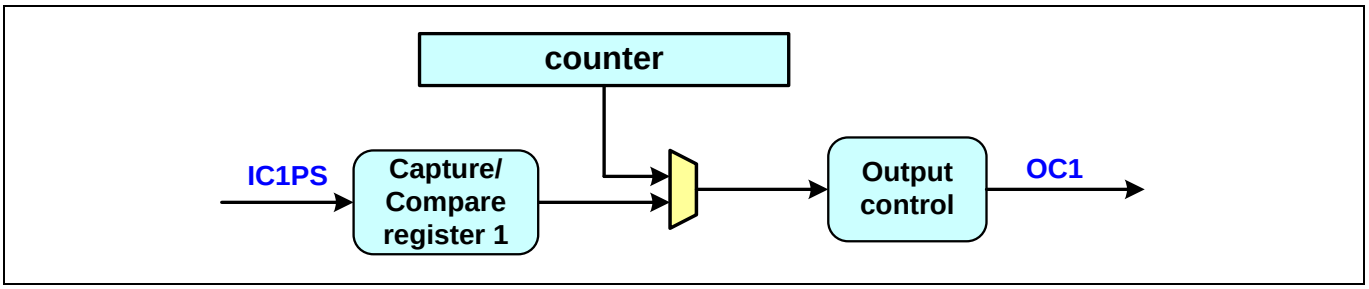
CC1OF is set. In order to avoid losing the capture information that may be generated before the repeat capture flag CC1OF is set. It is recommended to read data before reading the repeat capture flag.

- Set the corresponding CC1G bit of TIMx_EGR register, generate the input capture interrupt or DMA request through software.

13.4.3 Output compare

The compare output of the capture/compare channel is composed of the comparator, output control circuit and capture/compare register. Its structure is shown below:

Figure 13-6 Output compare block diagram



In output compare mode, the content of capture/compare register is loaded into the shadow register, and then compare the content of shadow register with the current value of counter. The capture/compare module includes a capture/compare register (preload register) and a shadow register. When read and write, only programmed the capture/compare register.

13.4.3.1 Force output

Configure TIMx_CCMR1 register CC1S = 00 and set the channel CC1 as output mode. Configure TIMx_CCMR1 register OC1M bit, and directly enforce the output compare signal as active or inactive level, independently of the output compare result. Configure TIMx_CCMR1 register OC1M = 100, and enforce output compare signal as inactive level. And OC1REF is enforced as low level. Configure TIMx_CCMRx register OC1M = 101, and enforce output compare signal as active level. And the OC1REF is enforced as high level (OC1REF always active as high level).

Note: In force output mode, TIMx_CCR1 shadow register and counter output compare is still performed. The corresponding flag bit of compare result will also be changed. When enable the corresponding interrupt and DMA request, it will also generate the corresponding interrupt and DMA request.

13.4.3.2 Output compare

In the output compare mode, when the values of counter and capture compare register are matches, different waveform can be exported according to the OC1M bit of TIMx_CCMR1 register.

For example, when the value of counter and capture/compare register matches, the functions in the output compare mode:

1. When compare match, OC1M value is different, output channel 1 signal OC1 run differently:
   - OC1M = 000: OC1signal keeps its level.
   - OC1M = 001: OC1 signal is set as active level.
   - OC1M = 010: OC1 signal is set as inactive level.
   - OC1M = 011: OC1 signal toggle.
2. When match, set the flag bit of status register (CC1IF bit of TIMx_SR register).
3. Configure TIMx_DIER register CC1IE =1, generate an interrupt when match.

4. Configure TIMx_DIER register CC1DE =1, generate a DMA request when match (only suitable for the built in DMA product).

The output compare mode can also be used to export a single pulse (one-pulse mode).

Configuration steps of channel 1 output compare mode:

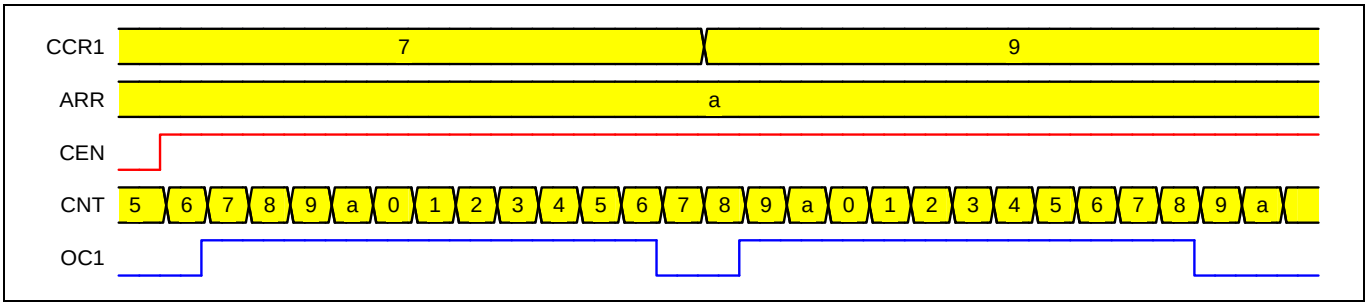
1. Configure counter clock (select clock source, configure prescaler factor).
2. Configure TIMx_ARR and TIMx_CCR1 register.
3. Configure TIMx_DIER register CC1IE =1, enable capture/compare 1 interrupt.
4. Configure output mode:

- Configure TIMx_CCMR1 register OC1M = 011, toggle when OC1 compare match.
- Configure TIMx_CCMR1 register OC1PE = 0, disable preload function of TIMx_CCR1 register.
- Configure TIMx_CCMR1 register CC1P = 1, OC1 is active low polarity.
- Configure TIMx_CCER register CC1E = 1, enable channel 1 output, OC1 signal is exported to the corresponding output pin.

5. Configure TIMx_CR1 register CEN =1, enable the counter.

Configure TIMx_CCMR1 register OC1PE=0, disable TIMx_CCR1 register preload function, write TIMx_CCR1 register any time, and the write value updated immediately. Configure TIMx_CCMR1 register OC1PE=1, enable TIMx_CCR1 register preload function, Updated only at the next update event. The figure below is an example.

Figure 13-7 Output compare mode, OC1 signals toggle when match



Note: In the output compare mode, the update event has no effect on output result. In the force output mode, the compare output between TIMx_CCR1 shadow register and counter still performed. The corresponding flag bit of compare result is also be changed. When enabling the corresponding interrupt and DMA request, it still generates the corresponding interrupt and DMA request.

13.4.3.3 PWM output

In the PWM mode, according to the value of TIMx_ARR register and TIMx_CCR1 register, generate a frequency and duty cycle controllable PWM waveform.

Configure OC1M=110 or OC1M=111 of TIMx_CCMR1 register corresponding to the channel 1, select channel 1 to PWM mode 1 or PWM mode 2. In the PWM mode, the counter will always compare with CCR1. According to configuration and compare results, the channel x exports different signals. Therefore, TIMx can generate 1 PWM output signals with independent duty under the same frequency. In PWM mode, enable the preload function of TIMx_CCR1 and TIMx_ARR. The written value of TIMx_CCR1 preload register and TIMx_ARR preload register will become effective at the next update event, and load the corresponding shadow register. Therefore, in the PWM mode, before enabling counter, it's required to set TIMx_EGR UG=1, and generate the update event to initialize all registers.

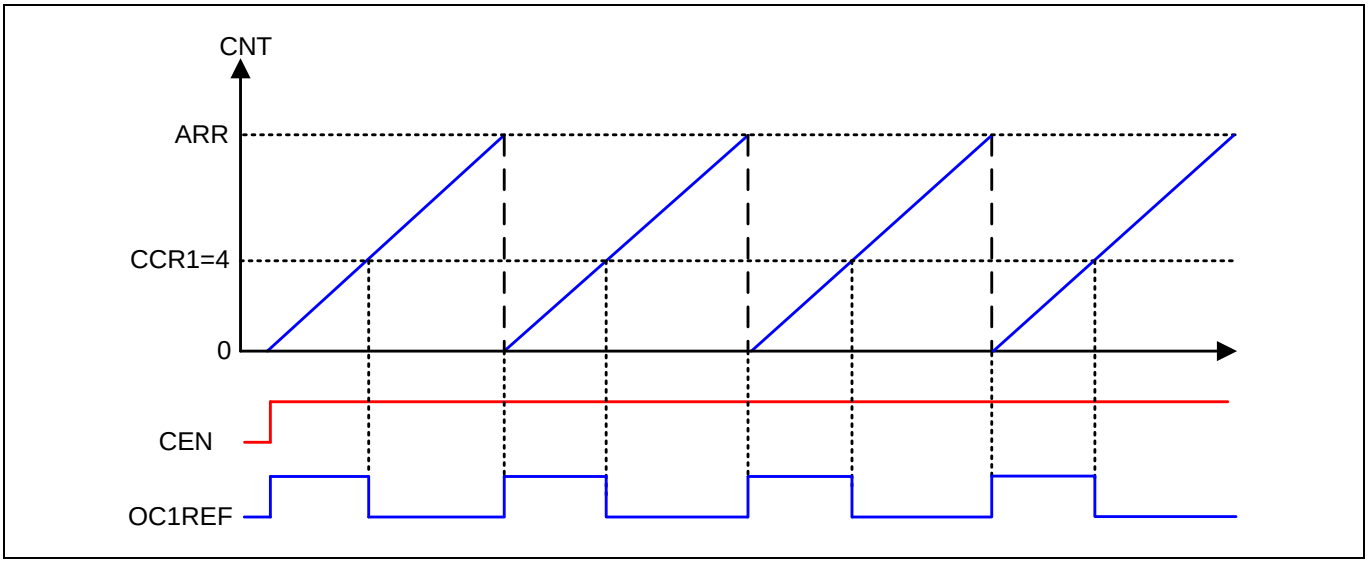
Configure TIMx_CCER register CC1P, select OC1's active polarity. Configure TIMx_CCER register CC1E to control OC1 output enable.

13.4.3.3.1 PWM edge alignment mode - up counting mode

Based on up counting mode configuration, configure TIMx_CCMR1 register CC1S =00, select output mode, OC1M =110, select PWM mode 1. In case of TIMx_CNT ＜TIMx_CCR1, channel 1 (OC1REF) is in active level. Otherwise, it's inactive level. If TIMx_CCR1 compare value is greater than

the auto reload value (TIMx_ARR), OC1REF stays in active level. When the compare value is 0, OC1REF is at inactive level. The figure below is the example of waveform in PWM mode 1 during edge alignment up count in case of CCR1 =4, and ARR =a.

Figure 13-8 Waveform in PWM mode 1 during edge alignment up count



13.4.3.4 One-pulse output

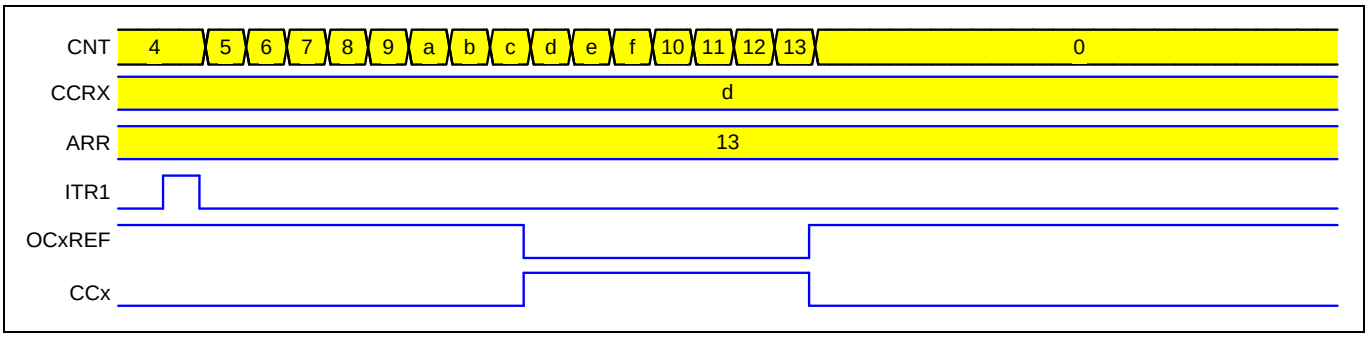
In the one-pulse mode (OPM), the counter responds a stimulus, and generate a pulse with width programmable. Configure TIMx_CR1 register OPM =1. Select one-pulse mode, trigger signal's active edge or configuring CEN =1 can enable the counter. CEN =1 remains until the next update event or configure CEN =0.

The necessary condition to generate pulse is the difference of compare value and initial value of the counter. Necessary configuration prior to counter enable:

- Up counting mode: Counter CNT < CCR1 ≤ ARR.

The figure below is one pulse mode:

Figure 13-9 one-pulse mode



13.4.4 Debug mode

In the debug mode, configure DBG_CR register DBG_TIMx_STOP=1, TIMx counter stops counting. (Details in the debug section)

13.4.5 Interrupt

TIMx interrupt includes: capture/compare 1 interrupt and update interrupt. When the corresponding interrupt enable bit is set, generate the corresponding event and interrupt.

Table 13-2 List of interrupt events

Interrupt event	Flag bit	Enable bit
Capture/compare 1 interrupt	CC1IF	CC1IE
Update interrupt	UIF	UIE

13.4.6 DMA

TIMx can generate DMA requests when a capture/compare event or an update event occurs.

13.5 Register

Table 13-3 TIMx register overview

Offset	Acronym	Register Name	Reset
0x00	TIMx_CR1	Control Register 1	0x0000
0x0C	TIMx_DIER	DMA/Interrupt Enable Register (DMA is only suitable for the chip with built in DMA)	0x0000
0x10	TIMx_SR	Status Register	0x0000
0x14	TIMx_EGR	Event Generation Register	0x0000
0x18	TIMx_CCMR1	Capture/Compare Mode Register 1	0x0000
0x20	TIMx_CCER	Capture/Compare Enable Register	0x0000
0x24	TIMx_CNT	Counter	0x0000
0x28	TIMx_PSC	Prescaler	0x0000
0x2C	TIMx_ARR	Auto Reload Register	0x0000
0x34	TIMx_CCR1	Capture/Compare Register 1	0x0000
0x44	TIMx_BDTR	Break And Dead-Time Register	0x0000

13.5.1 TIMx_CR1 Control Register 1

Address offset：0x00
Reset value：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved						CKD		ARPE	Reserved			OPM	URS	UDIS	CEN
						rw		rw				rw	rw	rw	rw

Bit	Field	Description
15：10	Reserved	Reserved, must be kept at reset value.
9：8	CKD	Clock division Division ratio between the timer clock (CK_INT) frequency and the dead-time and sampling clock used by the dead-time generators and the digital filters (TI1). 00：$t_{DTS} = t_{INT\_CK}$ 01：t_{DTS} =2x t_{INT_CK} 10：t_{DTS} =4x t_{INT_CK} 11：Reserved, do not program this value

Bit	Field	Description
7	ARPE	Auto reload preload enable 0: Disable the shadow register of TIMx_ARR register 1: Enable the shadow register of TIMx_ARR register
6：4	Reserved	Reserved, must be kept at reset value.
3	OPM	one-pulse mode 0: Disable one-pulse mode. In case of update event, the counter count continues 1: Enable one-pulse mode. In case of the next update event (clear CEN bit), the counter count stops
2	URS	Update request source Software configures this bit, select update event source. 0: The event below may generate an update interrupt or DMA request: - Counter overflow - Set UG bit 1: Only in counter overflow, generate update interrupt or DMA request
1	UDIS	Update disable This bit is used to enable or disable the update event 0: Update event (UEV) enabled. 1: Update event disabled. The Update event is not generated, shadow registers keep their value (ARR, PSC, and CCR1). However the counter and the prescaler are reinitialized if the EGR.UG bit is set.
0	CEN	Counter enable 0: Counter disabled 1: Counter enabled

13.5.2 TIMx_DIER DMA/Interrupt Enable Register

Address offset：0x0C

Reset value：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved						CC1DE	UDE	Reserved						CC1IE	UIE
						rw	rw							rw	rw

Bit	Field	Description
15：10	Reserved	Reserved, must be kept at reset value.
9	CC1DE	Capture/compare 1 DMA request enable 0: Capture/compare 1 DMA request disable 1: Capture/compare 1 DMA request enable Note: It's only suitable for the built in DMA product.
8	UDE	Update DMA request enable 0: Update DMA request disable 1: Update DMA request enable Note: It's only suitable for the built in DMA product.
7：2	Reserved	Reserved, must be kept at reset value.
1	CC1IE	Enable capture/compare 1 interrupt 0: Capture/compare interrupt 1 disable 1: Capture/compare interrupt 1 enable
0	UIE	Enable update interrupt 0: Update event interrupt disable 1: Update event interrupt enable

13.5.3 TIMx_SR Status Register

Address offset：0x10

Reset value：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved						CC1OF	Reserved							CC1IF	UIF
						r_w0c								r_w0c	r_w0c

Bit	Field	Description
15：10	Reserved	Reserved, must be kept at reset value.
9	CC1OF	Capture/compare 1 over capture flag Only when the channel 1 is configured as input capture mode and CC1F is configured as 1, and the capture event occurs again. The flag is set by hardware. Write 0 may clear the bit. 0: No over capture 1: Over capture
8：2	Reserved	Reserved, must be kept at reset value.
1	CC1IF	Capture/compare 1 interrupt flag Channel 1 in output mode: When the counter value and compare value match, the bit is set by hardware, It's cleared by software. 0: No match 1: TIMx_CNT value and TIMx_CCR1 value match Channel 1 as output mode: In case of capture event, the bit is set by hardware. It's cleared by software or reading TIMx_CCR1 register 0: No input capture occurred 1: Counter value is captured to TIMx_CCR1
0	UIF	Update interrupt flag In case of an update event, the bit is set 1 by hardware. It's cleared 0 by software. 0: No update interrupt occurred 1: update interrupt pending TIM1_EGR register UG =1 or counter overflow generate update event.

13.5.4 TIMx_EGR Event Generation Register

Address offset：0x14
Reset value：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved														CC1G	UG
														w	w

Bit	Field	Description
15：2	Reserved	Reserved, must be kept at reset value.
1	CC1G	Capture/compare 1 generation This bit is set by software. It generates a capture/compare event, and is auto cleared by hardware. 0: No action 1：Generate a capture/compare event in channel CC1: When the channel CC1 is configured as output: Set CC1IF =1. When enabling the corresponding interrupt and DMA, generate the corresponding interrupt and DMA. When the channel CC1 is configured input: Set CC1IF =1. When enabling the corresponding interrupt and DMA, generate the corresponding interrupt and DMA. When CC1IF is set, set CC1OF =1.
0	UG	Update generation 0：No action 1: Initialize the counter, and generate an update event. It's auto cleared by hardware.

13.5.5 TIMx_CCMR1 Capture/Compare Mode Register 1

Address offset：0x18
Reset value：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								Res.	OC1M			OC1PE	Res.	CC1S	

	IC1F		IC1PSC		CC1S
	rw	rw	rw	rw	rw

The channel may be used for input (capture mode) or output (compare mode). The channel direction is defined by the corresponding CC1S. The register's other bits function differently in the input mode and output mode. OC1x describes the channel function in the output mode. IC1x describes the channel function in the input mode.

Output compare mode:

Bit	Field	Description
15：7	Reserved	Reserved, must be kept at reset value.
6：4	OC1M	Output compare mode The bit has defined the output reference signal OC1REF action. OC1REF has determined OC1 value. OC1REF is active at high level. The active level of OC1 depends on CC1P bit. 000：Freeze. TIMx_CCR1 and TIMx_CNT compare results has no effect on OC1REF. 001: Set as high when configuration. When TIMx_CNT value and TIMx_CCR1 value are same, enforce OC1REF as high level 010: Set as low when configuration. When TIMx_CNT value and TIMx_CCR1 value are same, enforce OC1REF as low level 011: Toggle when match. When TIMx_CCR1=TIMx_CNT, OC1REF toggle. 100: Enforce as low. Enforce OC1REF at low level 101: Enforce as high. Enforce OC1REF at high level 110：PWM mode 1. During up count, in case of TIMx_CNT<TIMx_CCR1, enforce OC1REF is at high level. Or else, it's at low level. 111：PWM mode 2. During up count, in case of TIMx_CNT<TIMx_CCR1, channel 1 enforce OC1REF is at low level. Or else, it's at high level. Note: In PWM mode 1 or PWM mode 2, only when the compare result changes or it changes over from the freeze mode to PWM mode in the output compare mode, OC1REF level may change.
3	OC1PE	Output compare preload enable 0: Disable TIMx_CCR1 register preload function. The value written into TIMx_CCR1 register becomes valid immediately 1: Enable TIMx_CCR1 register preload function. Only read and write the preload register. TIMx_CCR1 preload value becomes valid during the update event Note: Only in the one-pulse mode (TIMx_CR1 register OPM= 1), it has no influence whether the preload register is set. Under other scenarios, it's required to set the preload register. Otherwise, the follow up action is not certain.
2	Reserved	Reserved, must be kept at reset value.
1：0	CC1S	Channel 1 Capture/Compare selection This bit defines the channel direction and input signal selection. Only when the channel closes, these bits can be written: 00：Channel 1 is configured as output 01：Channel 1 is configured as input 10：Reserved 11：Reserved

Input capture mode

Bit	Field	Description
15：8	Reserved	Reserved, must be kept at reset value.

Bit	Field	Description
7：4	IC1F	Channel 1 input capture filter The digital filter is composed of an event counter. It records an output jump after N input events. These bits define IC1 input signal sampling frequency and digital filter length. 0000：No filter, f_{DTS} sampling 0001：Sampling frequency $f_{sampling}=f_{INT\_CK}$，N=2 0010：Sampling frequency $f_{sampling}=f_{INT\_CK}$，N=4 0011：Sampling frequency $f_{sampling}=f_{INT\_CK}$，N=8 0100：Sampling frequency $f_{sampling}=f_{DTS}$ /2，N=6 0101：Sampling frequency $f_{sampling}=f_{DTS}$ /2，N=8 0110：Sampling frequency $f_{sampling}=f_{DTS}$ /4，N=6 0111：Sampling frequency $f_{sampling}=f_{DTS}$ /4，N=8 1000：Sampling frequency $f_{sampling}=f_{DTS}$ /8，N=6 1001：Sampling frequency $f_{sampling}=f_{DTS}$ /8，N=8 1010：Sampling frequency $f_{sampling}=f_{DTS}$ /16，N=5 1011：Sampling frequency $f_{sampling}=f_{DTS}$ /16，N=6 1100：Sampling frequency $f_{sampling}=f_{DTS}$ /16，N=8 1101：Sampling frequency $f_{sampling}=f_{DTS}$ /32，N=5 1110：Sampling frequency $f_{sampling}=f_{DTS}$ /32，N=6 1111：Sampling frequency $f_{sampling}=f_{DTS}$ /32，N=8
3：2	IC1PSC	Channel 1 input capture prescaler This bit defines the ratio of the prescaler acting on IC1.The prescaler is reset as soon as CC1E=0(TIMx_CCER register). 00: No prescaler, the capture input detects each edge to trigger one capture 01: Trigger one capture for every 2 events 10: Trigger one capture for every 4 events 11: Trigger one capture for every 8 events
1：0	CC1S	Channel 1 capture/Compare selection This bit defines the channel direction and input signal selection. Only when the channel closes, these bits can be written: 00: Channel 1 is configured as output 01: Channel 1 is configured as input 10: Reserved 11: Reserved

13.5.6 TIMx_CCER Capture/Compare Enable Register

Address offset：0x20

Reset value：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved												CC1NP	Res.	CC1P	CC1E
												rw		rw	rw

Bit	Field	Description
15：4	Reserved	Reserved, must be kept at reset value.
3	CC1NP	Channel 1 input capture 1 polarity This bit is not valid when Channel 1 is configured as an output. When channel 1 is configured as input, CC1P/CC1NP match use has defined the input signal polarity and level. Details are given in the IC1 polarity/level selection table.
2	Reserved	Reserved, must be kept at reset value.
1	CC1P	Channel 1 capture/Compare output polarity When Channel 1 is configured as an output, this bit defines the output signal polarity. 0: OC1 active at high level 1: OC1 active at low level When channel 1 is configured as input, CC1P/CC1NP match use has defined the input signal polarity and level. Details are given in the IC1 polarity/level selection table.

Bit	Field	Description
0	CC1E	Channel 1 capture/compare output enable When channel 1 is configured as an output. 0: off. OC1 output is disabled 1: on. OC1 signal is output to the corresponding output pin CC1 Channel configured as input. This bit determines whether the input capture function is enabled. 0: Capture disable 1: Capture enable

The polarity/level selection of IC1 in the input mode is shown in the following table:

Table 13-4 IC1 polarity/level selection table

CC1P	CC1NP	IC1 polarity/level
0	0	Rising edge active/high level active
1	0	Falling edge active/low level active
1	1	Rising or falling edge active/low level active
0	1	Reserved

13.5.7 TIMx_CNT Counter

Address offset：0x24
Reset value：0x0000

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

Bit	Field	Description
15：0	CNT	Count value

13.5.8 TIMx_PSC Prescaler

Address offset：0x28
Reset value：0x0000

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

Bit	Field	Description
15：0	PSC	Prescaler value Counter clock frequency（ck_cnt）=f_{CK_PSC}/（PSC+1） In case of an update event, PSC value is loaded into the current prescaler register.

13.5.9 TIMx_ARR Auto Reload Register

Address offset：0x2C
Reset value：0x0000

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

Bit	Field	Description
15：0	ARR	Auto-reload value These bits define the auto reload value of the counter. When auto reload value is 0, the counter doesn't work.

13.5.10 TIMx_CCR1 Capture/Compare Register 1

Address offset：0x34
Reset value：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR1															
rw															

Bit	Field	Description
15：0	CCR1	Channel 1 capture/compare value When channel 1 is configured as input: CCR1 value determines the counter value of the previous capture event (this register is only readable). When channel 1 is configured as output: If the preload function is not selected in TIMx_CCMR1 register (OC1PE bit), the write value is immediately transmitted to the current capture/compare shadow register. Or else, the write value is only loaded into the capture/compare register in case of update event. The current capture/compare shadow register is involved in the compare with the counter TIMx_CNT, and the compare result is reflected to the output signal of OC1 port.

13.5.11 TIMx_BDTR Break and Dead-Time Register

Address offset：0x44
Reset value：0x0000

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

Bit	Field	Description
15	MOE	Main output enable 0：Disable OC1 output or enforce as the idle state 1: If the corresponding enable bit (CC1E of TIMx_CCER register) is set, the OC1 output is turned on
14：0	Reserved	Reserved, must be kept at reset value.

14 IWDG Independent watchdog

14.1 Introduction

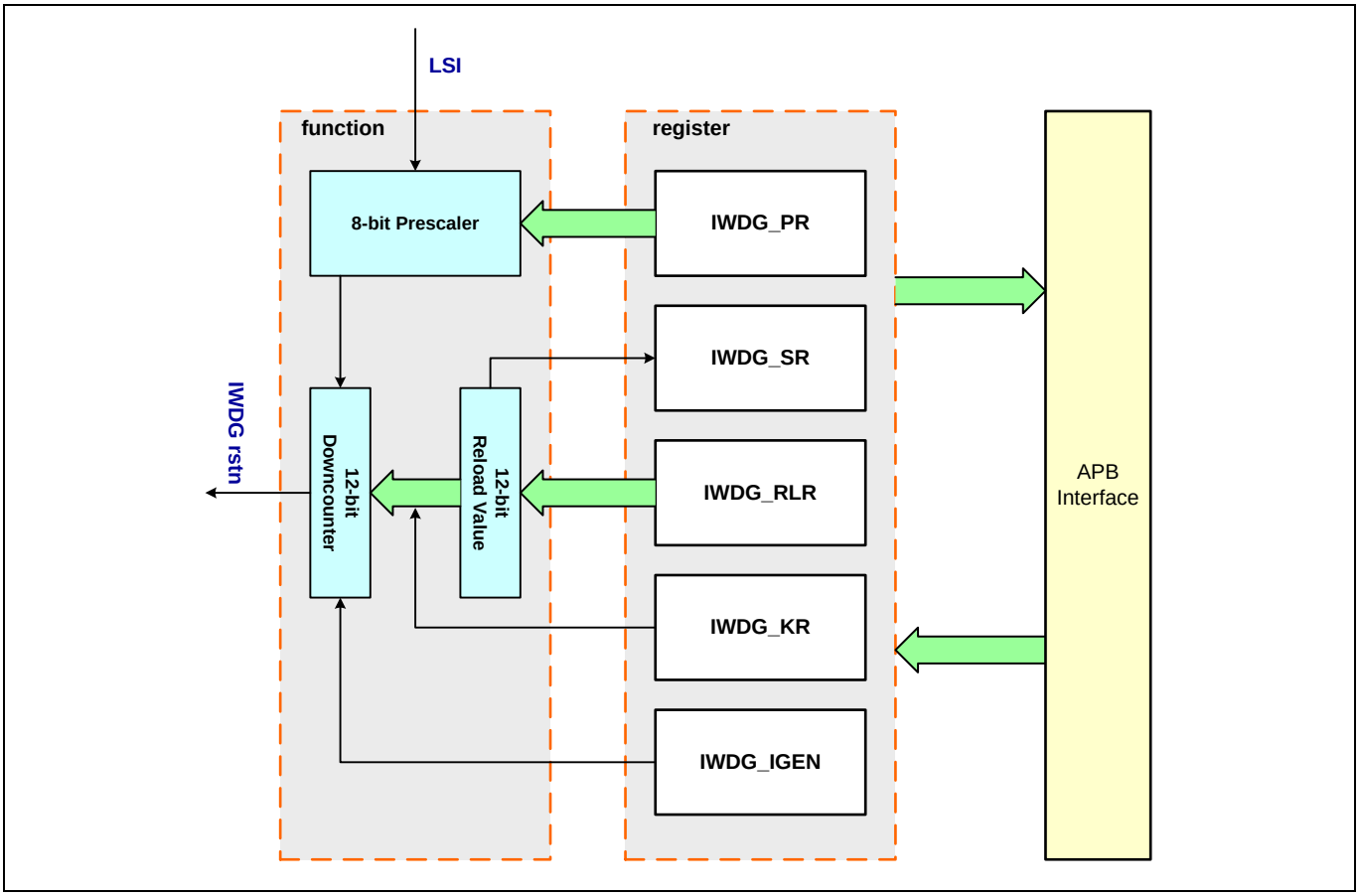
The independent watchdog is designed to detect and resolve malfunctions due to software failure. Its principle can be briefly described as follows: a system reset signal is generated when the independent watchdog (IWDG) counter decrements to a given value, thus triggering a system reset and improving the overall safety level of the system.

The independent watchdog is best suited to applications which require the watchdog to run as a totally independent process outside the main application but have lower timing accuracy constraints.

The independent watchdog is clocked by the internal low-speed clock (LSI) and thus stays active even if the main clock fails.

14.2 Functional block diagram

Figure 14-1 Functional block diagram



14.3 Main features

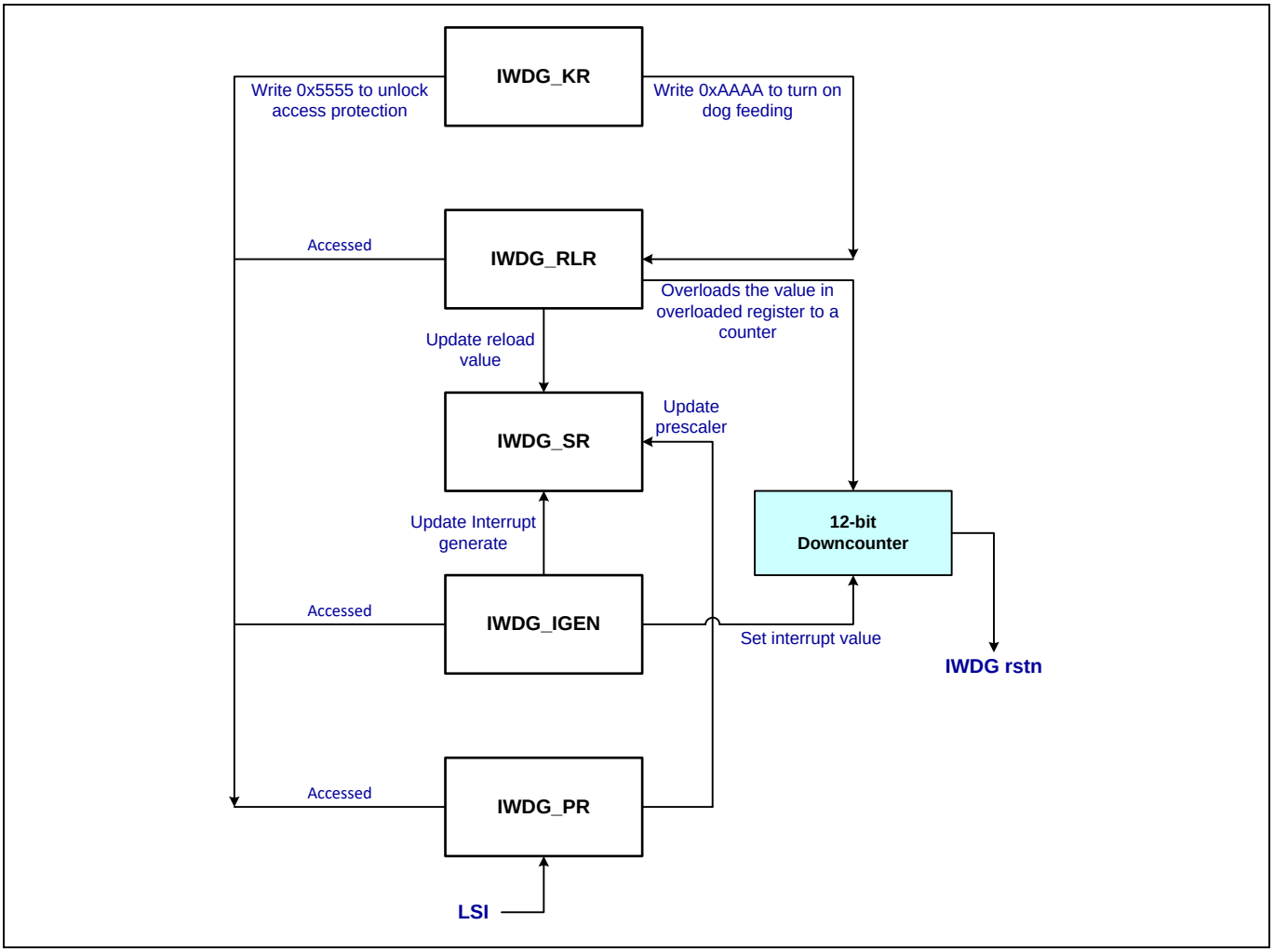
- The software watchdog is the default mode in the chip.
- The hardware watchdog can be enabled by programming the WDG_SW bit of the reset option byte register in the Flash. If enabled, the hardware watchdog operates automatically after it is powered on by a system reset and the internal counter starts decrementing.
- The LSI keeps working in low power modes, Refer to the PWR section.
- There is a 12-bit free-running down counter inside the watchdog. A system reset or an interrupt signal is generated when the down counter value reaches 0x0000.

14.4 Functional description

- The independent watchdog is started by writing the value 0xCCCC in the Key register (IWDG_KR). Meanwhile, the counter starts counting down from the reset value of 0xFFF. Either a system reset signal is generated when the counter reaches 0x0000, or an interrupt is generated when the counter reaches the value of IWDG_IGEN. The IRQ_SEL configuration decides which result will occur.
- Whenever 0xAAAA is written to IWDG_KR, the value in the reload register (IWDG_RLR) will be reloaded into the counter (commonly known as “feeding the dog”) to avoid the generation of a reset signal or interrupt signal.
- A reset signal or interrupt signal is generated if the compare/output program is abnormal, and the dog cannot be fed properly. Then a system reset, or system interrupt is generated.
- Access to the IWDG_PR, IWDG_RLR, and IWDG_IGEN registers is protected. These registers can be modified only by writing the code 0x5555 in the Key register (IWDG_KR). Write access to this register with a different value will break the sequence and register access will still be protected. It is the same case as the reload operation.
- Since the clock of the independent watchdog is provided by LSI, the watchdog can work in the low-power modes such as the Stop and Standby modes.
- The independent watchdog counts normally in the low-power modes. The system will exit the low-power mode if the watchdog is reset.
- In low-power mode, the soft watchdog can be disabled by configuring the RCC register to select whether to turn off the LSI clock upon entering Stop mode.

14.4.1 Flow block diagram

Figure 14-2 Flow block diagram



14.4.2 Independent watchdog timeout

Table 14-1 IWDG timeout (For example, the LSI clock frequency is 40 KHz)

Prescale Coefficient	PR[2:0] Bits	Min time (ms) RL[11:0]=0x000	Max time (ms)
/4	0	0.1	409.6
/8	1	0.2	819.2
/16	2	0.4	1638.4
/32	3	0.8	3276.8
/64	4	1.6	6553.6
/128	5	3.2	13107.2
/256	(6 or 7)	6.4	26214.4

Timeout (overflow) is calculated as below:
Tout= ((4×2^PR) ×RLR)/40
where: the unit of Tout is millisecond.
The clock frequency LSI = 40K; a watchdog clock period is the minimum timeout.
Maximum timeout = (maximum value of IWDG_RLR register) x watchdog clock period

14.4.3 Interrupt

- If the CR.IRQ_SEL is set, the IWDG generates an interrupt when the counter overflows.

14.5 Register

14.5.1 Overview of registers

Table 14-2 Overview of IWDG registers

Offset	Acronym	Register Name	Reset
0x00	IWDG_KR	Key register	0x00000000
0x04	IWDG_PR	Prescaler register	0x00000000
0x08	IWDG_RLR	Reload register	0x00000FFF
0x0C	IWDG_SR	Status register	0x00000000
0x10	IWDG_CR	Control register	0x00000000
0x14	IWDG_IGEN	Interrupt generate register	0x00000FFF
0x18	IWDG_CNT	Counter register	0x00000001

14.5.2 IWDG_KR Key Register

Address offset: 0x00
Reset value: 0x0000 0000

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

Bit	Field	Description
31:16	Reserved	Reserved, must be kept at reset value
15:0	KEY	Key value (write-only register) These bits must be written by software at regular intervals with 0xAAAA to feed the dog, otherwise a reset signal is generated to reset the system when the counter decrements to 0x0000. Writing 0x5555 by software may disable the protection and enable access to other configuration registers (IWDG_PR, IWDG_RLR, IWDG_CR (bit0), and IWDG_IGEN). Writing 0xCCCC by software starts the watchdog.

14.5.3 IWDG_PR Prescaler Register

Address offset: 0x04
Reset value: 0x0000 0000 (reset by system)

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

Bit	Field	Description
31:3	Reserved	always read as 0

Bit	Field	Description
2:0	PR	Prescaler divider These bits are set to select the prescaler dividers of the LSI clock. To modify prescaler dividers, the protection should be disabled at first (write 0x5555 to IWDG_KE). After the prescaler dividers are updated, The PVU bit in the register becomes 0. At this time, the value read from this register is valid. 000: prescaler divider = 4 100: prescaler divider = 64 001: prescaler divider = 8 101: prescaler divider = 128 010: prescaler divider = 16 110: prescaler divider = 256 011: prescaler divider = 32 111: prescaler divider = 256

14.5.4 IWDG_RLR Reload Register

Address offset: 0x08
Reset value: 0x0000 0FFF (reset by system)

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

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

Bit	Field	Description
31:12	Reserved	Reserved, must be kept at reset value
11:0	RL	Watchdog counter reload value These bits serve to define the reload value of the watchdog counter. This value will update to the counter each time the dog is fed (write 0xAAAA to IWDG_KR register). Subsequently, the watchdog counter counts down from this value. To modify the reload values, the protection should be disabled at first (write 0x5555 to IWDG_KR). After the reload values are updated, the RVU bit in the register becomes 0. At this time, the value read from this register is valid. The watchdog timeout period is a function of this reload value and the prescaler.

14.5.5 IWDG_SR Status Register

Address offset: 0x0C
Reset value: 0x0000 0000 (reset by RCC and system)

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.												UPDATE	IVU	RVU	PVU
												r	r	r	r

Bit	Field	Description
31:4	Reserved	Reserved, must be kept at reset value
3	UPDATE	Watchdog reload value update flag The UPDATE is set when 0xAAAA is written into the IWDG_KR register. It is cleared automatically when the watchdog counter is updated and the reload value is written into the counter.
2	IVU	Watchdog Interrupt Generate value update This bit is set by hardware to indicate that an update of the interrupt generate value is ongoing. It is cleared by hardware when the interrupt generate value update operation is completed in the VDD voltage domain (takes up to 5 oscillator cycles at LSI). The interrupt generate value can be updated only when IVU bit is cleared.

Bit	Field	Description
1	RVU	Watchdog counter reload value update This bit is set when an update of the reload value is ongoing. It is cleared when the reload value update operation is completed (takes up to 5 oscillator cycles at LSI). The reload value can be updated only when RVU bit is cleared.
0	PVU	Watchdog prescaler value update This bit is set when an update of the prescaler value is ongoing. It is cleared when the prescaler value update operation is completed (takes up to 5 oscillator cycles at LSI). The prescaler value can be updated only when PVU bit is cleared.

Note: If several reload values, prescaler values or interrupt generate values are used by application, it is necessary to disable the register protection (write 0x5555 to IWDG_KR) at first. Then configure the IWDG_PR, IWDG_RLR and IWDG_IGEN registers. Wait until the corresponding bits (PVU, RVU, IVU) in the status register are cleared, which indicates that the reload value, prescaler value, and interrupt generate values are configured. Next, feed the dog, or wait for a reset or interrupt signal which is generated from the auto decrement of the counter.

14.5.6 IWDG_CR Control Register

Address offset: 0x10
Reset value: 0x0000 0000(reset by RCC and system)

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.														IRQ_CLR	IRQ_SEL
														rw	rw

Bit	Field	Description
31:2	Reserved	Reserved, must be kept at reset value
1	IRQ_CLR	IWDG interrupt clear 1: Interrupt cleared 0: No effect, interrupt flag still pending Note: It is not necessary to disable the KEY protection before writing to this bit.
0	IRQ_SEL	IWDG overflow operation select 1: interrupt generation enabled after overflow 0: reset generation enabled after overflow

14.5.7 IWDG_IGEN Interrupt Generate Register

Address offset: 0x14
Reset value: 0x0000 0FFF (reset by system)

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

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

Bit	Field	Description
31:12	Reserved	Reserved, must be kept at reset value

Bit	Field	Description
11:0	IGEN	IWDG Interrupt Generate value These bits serve to define the interrupt generate value of the watchdog. An interrupt is generated when the counter counts down until it reaches this value. The protection should be disabled before changing it. After the value update operation is completed, the value read from this register is valid only when the IVU bit in the IWDG_SR register is cleared.

14.5.8 IWDG_CNT Counter Register

Address offset: 0x18
Reset value: 0x0000 0001(reset by RCC and system)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.													IWDG_CNT		
													r		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IWDG_CNT								IWDG_PS							
r								r							

Bit	Field	Description
31:19	Reserved	Reserved, must be kept at reset value
18:8	IWDG_CNT	IWDG counter value
7：0	IWDG_PS	Value of the prescaler counter of the IWDG clock

15 USART Universal Synchronous Asynchronous Receiver Transmitter

15.1 Introduction

The universal synchronous/asynchronous receiver transmitter (USART) offers a flexible means of full-duplex data exchange with external equipment. The USART offers a very wide range of baud rates using a built-in baud rate (including integer and decimal formats) generator.

It supports asynchronous mode (UART), synchronous mode. The single wire half duplex communication is supported in the asynchronous mode (UART). Both the asynchronous mode (UART) and synchronous mode support the Modem (CTS/RTS) operations.

15.2 USART features

- Supports DMA
- Supports full-duplex asynchronous communications and full-duplex clock synchronous communications
- Baud rate generator (including integer and decimal formats)
  - Programmable baud rate, available for a transmitter or receiver (with a minimum division factor of 1)
- Separate transmit and receive FIFO registers, with separate enable bits for transmitter and receiver
- Supports LSB/MSB reception and transmission modes
- Programmable data word length (8 or 9 bits)
- Configurable stop bits (1/2 stop bits)
- Configurable parity check bit (odd parity, even parity, without parity check)
- Supports the idle frame's generation (automatic output when TE is enabled) and receive detection
- Supports hardware automatic flow control (with CTS/RTS pin functions configured)
- Support exchange of Tx and Rx pins and inversion of Tx and Rx signals
- Supports following interrupting sources:
  - Tx data register empty (TXE)
  - CTS flag (CTS)
  - Transmission complete (TC)
  - Rx data valid (RXNE)
  - Rx buffer overrun (OVR)
  - Rx idle frame complete (IDLE)

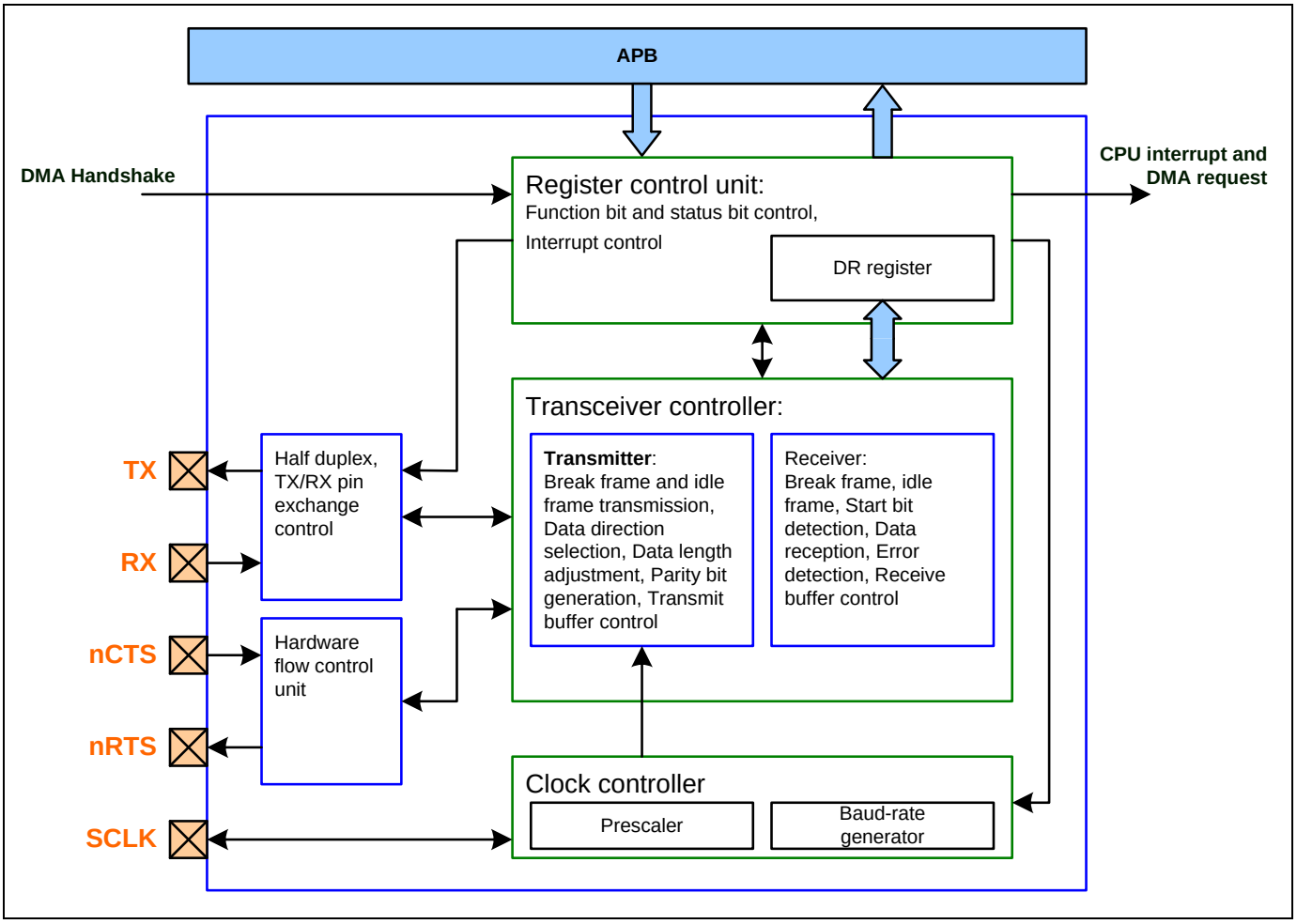
- Parity error (PE)
- Noise flag (NF) and framing error (FE)

15.3 USART functional description

15.3.1 Functional block diagram

The functional block diagram of USART can be referred to as follows: USART can be divided into register-related control unit, receive/transmit data controller, clock controller, hardware flow control unit, and pin control logic unit.

Figure 15-1 USART functional block diagram



15.3.2 Signal description

Signal Name	Type	Description
USART_SCLK	Output or input	Input or output clock pin in synchronous mode
USART_TX	Output or input	Transmit data pin, or half-duplex receive/transmit data pin
USART_RX	Input	Receive data pin (for full duplex)
USART_nCTS	Input	Clear transmit pin
USART_nRTS	Output	Request transmit pin

15.3.3 Functional description

The full-duplex communication requires a minimum of two pins for USART: Receive Data in (RX) and Transmit Data Out (TX).

RX: External serial data is transmitted to the USART receiver through this pin. Oversampling techniques are used for data recovery by discriminating between valid incoming data and noise which is generated during the transmission.

TX: The serial data generated internally in the USART transmitter is output through this pin. When the transmitter is enabled and nothing is to be transmitted, the TX pin is at a high level.

The idle state is the initial state of the bus prior to transmission or reception.

There is one start bit represented by '0'.

In the USART communication process, a data frame (8 or 9 bits) can be transmitted and received in least significant bit first (LSB) or in most significant bit first (MSB).

The stop bit indicates the end of a frame with a '1'. The number of bits can be configured as 1 or 2.

The internal baud rate generator is controlled by programming the baud rate register USART_BRR to obtain the desired baud rate for data transmission (refer to the baud rate calculation equation in section 15.3.5).

By configuring the SWAP bit in the USART_CR2 register, the interface of Tx and Rx pins can be swapped.

By configuring the RXTOG/TXTOG bit in the USART_CR2 register, the level signal at the receiving / transmitting terminals can be reversed (including the start bit and stop bit).

The following pins are required in hardware flow control mode:

- nCTS Clear to Send: when it is at the high level, it indicates that the current Rx end cannot perform data reception and the Tx end should stop the subsequent transmission.
- nRTS Request to Send: when it is in the low level, it indicates that the current Rx end can receive data.

In addition, the USART module supports synchronous mode (different from UART), and thus the following pins are required:

- SCLK transmitter clock output or clock input: This pin is used in synchronous mode. In synchronous mode, the clock input and output functions are supported, and the clock polarity and phase are configured via software.

15.3.4 Character description

Word length may be selected as being either 8 or 9 bits by programming the USART_CR1.DL bit. The TX pin is pulled down by the transmitter during the transmission start bit. It is pulled up during the transmission stop bit.

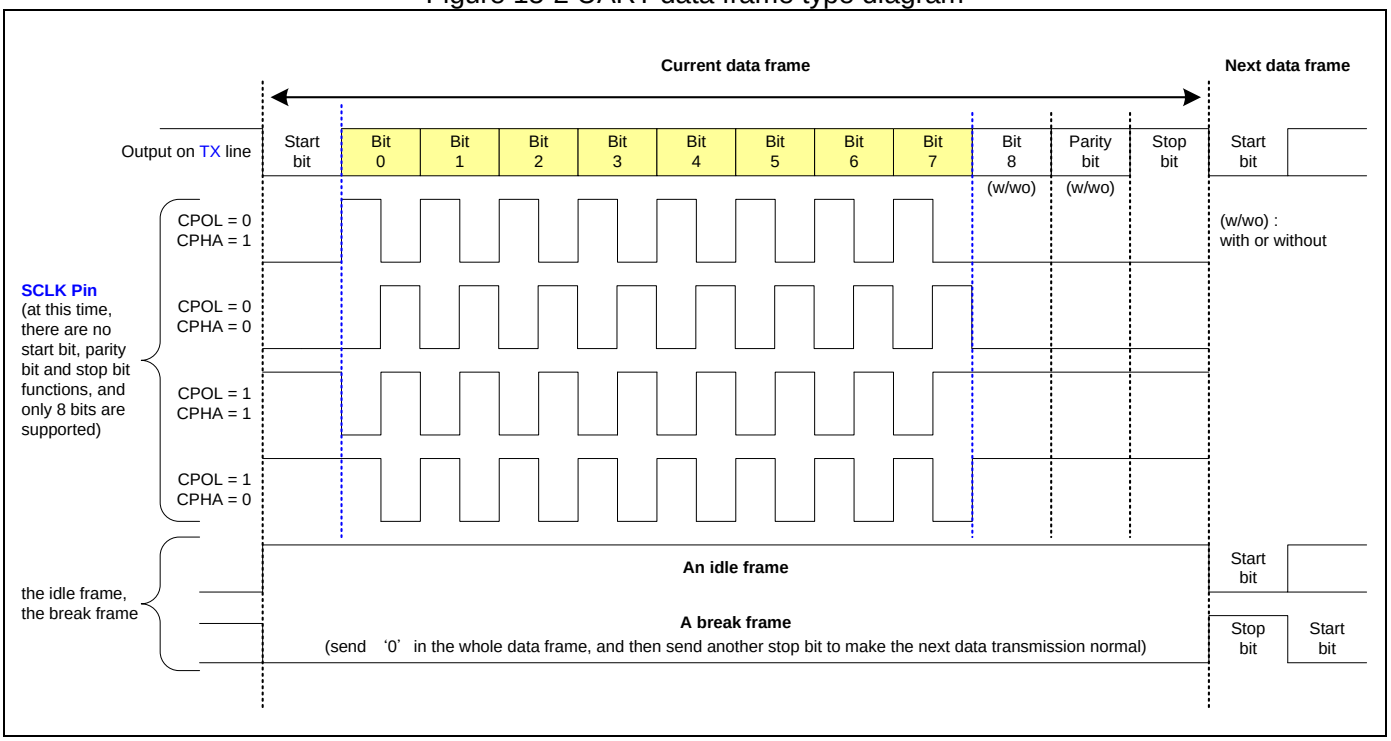
An idle frame is interpreted as an entire data frame of '1's containing the stop bit. The start bit of the next data frame follows the Idle frame.

A Break frame is interpreted as an entire data frame of '0's containing the stop bit. At the end of the Break frame, the transmitter sends another '1' stop bit to acknowledge the start bit of the next frame (generating a falling edge to be detected by the Rx end).

The clock generated by the baud rate generator is provided for the transmitter or receiver, when the enable bit is set respectively for the transmitter and receiver.

The following figure shows a diagram of data frame format, the break frame, and the Idle frame.

Figure 15-2 UART data frame type diagram



15.3.5 Baud rate generator

For each communication mode, the baud rate can be configured according to the following equation.

- For asynchronous mode (UART) and multiprocessor mode (including the multiprocessor under LIN protocol):

  The communication baud rate:

$$f_{baudrate} = \frac{\text{PCLK}}{N\times\ （\text{MFD} + \text{FFD/N}）}$$

  Error E (%):

$$E(\%) = \{\frac{\text{PCLK}}{f_{baudrate}\times N\times\ （\text{MFD} + \text{FFD/N}）} - 1\}\times 100$$

  In the above equation, PCLK is the frequency of internal clock source; MFD and FFD are the integer and fractional frequency division that baud rate configuration of the USART_BRR; N=8x (2-OVER8). Select the oversampling mode by configuring the USART_CR1.OVER8; when OVER8=1 (8x oversampling), FFD [3:0] only use the lower 3 bits and user should configure the FFD [3] bit to 1'h0.

- For synchronous mode:

  The communication baud rate:

$$f_{baudrate} = \frac{\text{PCLK}}{4\times\ \text{MFD}}$$

  In the above equation, PCLK is the frequency of the internal clock source; MFD is the integer baud rate configuration of the USART_BRR. In synchronous mode, fractional frequency division (FFD) is invalid, and user should configure FFD [3:0] to 4 'h0.

15.3.6 Sampling

The built-in detection circuit of UART detects the start of a data frame and the RX pin is sampled. The UART uses a clock of 8 or 16 times the data baud rate to sample the data on the RX pin.

The USART_CR1.OVER8 bit can be configured to select whether the USART uses a clock of 16

or 8 times the data baud rate to sample the data on the RX pin.

When 8x oversampling (OVER8=1) is selected, a higher speed (up to fPCLK/8) can be obtained, but the maximum tolerance of the receiver to clock deviation will be reduced.

15.3.7 Parity check control

Parity control (generation of parity bit in transmission and parity checking in reception) can be enabled by setting the USART_CR1.PCE bit. The USART_CR1.PS bit is used to select odd parity or even parity for the data.

Even parity: the total number of '1s' inside the data and the parity bit is an even number.

Odd parity: the total number of '1s' inside the data and the parity bit is an odd number.

When parity control is effective：

- The transmitter will automatically generate a parity bit and output it before the stop bit.
- The receiver will detect the parity bit and judge whether it is wrong. If the parity bit is wrong, the hardware will automatically set USART_SR.PE flag, but the current received data will still be transferred from the shift register to USART_ DR register.

15.3.8 Transmitter

When the USART_CR1.TE bit is set, the transmitter is enabled, and the serial data is output on the TX pin. The transmit data register USART_DR and the internal transmit shift register form a double buffer structure to transmit data continuously. As for UART mode, the data length (8 bits or 9 bits) can be selected by configuring the USART_CR1.DL bit.

15.3.8.1 Character transmission

During a USART transmission, data is written from the USART_DR register, and the data frame byte shifts out least significant bit first (USART_CR1.MLS=0) or most significant bit first (USART_CR1.MLS=1) on the TX pin via the transmit shift register.

Data transmission order: a start bit, character, a parity bit (with or without), stop bits.

The number of stop bits can be configured in the register USART_CR2.STOP [1:0] as 1 or 2.

The USART_CR1.TE bit should not be reset before data transmission is complete. Otherwise, the baud rate generator will stop generating the clocks immediately. Subsequently, the latter part of the data being transmitted are lost.

15.3.8.2 Send break frame

Setting the USART_CR1.SBK bit to transmit a break frame. If the SBK bit is set to '1' during the transmission, the break frame can't be sent on the TX pin until completing the current character transmission.

SBK bit is cleared automatically by hardware when the break frame is completed and sending an additional high-level stop bit (to ensure that the start bit of the next frame data can be detected).

The length of break frames depends on the data frame length (CR2.DL), parity enable bit (CR1. PS) and stop bit (CR2. STOP). For example, when there is no parity bit and the stop bit is 1 bit, the break frame is 10 consecutives '0' bits if CR2.DL=0, while it is 11 consecutive '0' bits if CR2.DL=1.

15.3.8.3 Transmission configuration procedure

Refer to the following procedures to configure the USART to transmit data frames.

1. Configure the pin functions required by USART.
2. Enable the USART (USART_CR1.UE=1).
3. Configure the USART_BRR.

4. Configure the USART_CR1, USART_CR2, and USART_CR3 registers according to the requirements such as data frame transmission.
5. Enable the transmitter (USART_CR1.TE=1); set USART_CR1.TXEIEN=1 if the transmit data register empty interrupt is needed.
6. Wait until the transmit data register is empty. Write the communication data to USART_DR, and the data is transferred to the transmit shift register. Then the transmission starts (when the CTS function is enabled, the data is transferred to the transmit shift register when the USART_CTS input is at the low level, and then the transmission starts).
7. Repeat step 6 if the data should be transmitted continuously.
8. Confirm that the transmission is complete by checking the USART_SR.TC bit. If TCIEN=1 is configured, a “transmit complete interrupt” is generated after the transmission of the last frame is complete.

Note: The USART's transmitter supports two types of interrupts, namely the transmit data register empty interrupt TXE and the transmit complete interrupt TC, which can be queried by the status bits in the USART_SR register. If TXEIEN=1, a TXE interrupt is generated when the value of USART_DR register is transferred to the transmit shift register. If TCIEN=1, a TC interrupt is generated when the last bit of data is transmitted, and no further data is written to the USART_DR register.

15.3.9 Receiver

The data register USART_DR and the internal receive shift register form a double buffer structure to receive data continuously.

As for UART mode, the data length (8 bits or 9 bits) can be selected by configuring the USART_CR1.DL bit.

After the receiver enabled bit USART_CR.RE is set to '1' and a start bit is detected, the data on the RX pin is received into the receive shift register. When a data frame is received, the data is transferred from the receive shift register to the data register USART_DR. Meanwhile, the status flag RXNE will be set to '1'. If RXNEIEN=1, the RXNE interrupt request is allowed. When the CPU or DMA uses this request to read the received data, the data can be read only once per request.

Data reception order: start bit -> data bit (MSB/LSB) -> parity bit (with or without) -> stop bits.

15.3.9.1 Receive break frame

When the USART receiver recognizes a break frame, it sets the USART_SR.FE flag (equivalent to receiving a '0' in the stop bit).

15.3.9.2 Receive idle character

When UART works normally, the receiver will set the USART_SR.IDLE flag when it receives an idle frame.

Configure IDLEIEN=1 to allow the IDLE interrupt request.

15.3.9.3 Reception configuration procedure

Refer to the following procedures to configure the USART to receive data frames.

1. Configure the functional pins required by USART.
2. Enable the USART (USART_CR1.UE=1).
3. Configure the USART_BRR.
4. Configure the USART_CR1, USART_CR2, and USART_CR3 registers according to the requirements such as data frame.
5. Configure the USART_BRR to set the communication baud rate (not necessary if the clock source is external).
6. Enable the receiver (USART_CR1.RE=1), set USART_CR1.RXNEIEN=1 if the receive interrupt is needed.
7. When a start bit is detected, the receiver receives the data into the receive shift register and

checks the parity and stop bits. There are three error flags in total: PE, FE, and ORE. When no error occurs, the received data is transferred from the receive shift register to the USART_DR register, and the RXNE flag is set to '1'.

8. The received data can be read through the RXNE interrupt. Repeat step 7 if the data should be received continuously.

9. If any receive error has been detected during reception, the corresponding error flag can be set.

Note: To prevent overflow error, the RXNE bit must be cleared (software reads the data register USART_DR) before the end of the next character reception. Data reception can no longer be performed when any of the PE, FE, or ORE reception errors occur, but data reception can also be restarted by clearing all error flags.

- When an overrun error occurs, the received data is lost and the ORE status bit is set to '1', but the RXNE interrupt is not generated.
- When a parity error occurs, the received data is transferred to USART_DR and the PE status bit is set to '1', but the RXNE interrupt is not generated.
- When a framing error occurs, the received data is transferred to USART_DR and the FE status bit is set to '1', but the RXNE interrupt is not generated.

15.3.10 Synchronous mode

By configuring USART_CR1.SAS bit to '1' to enable synchronous mode (Clock pin function will be valid at the same time).

In synchronous mode, users should configure USART_ CR2.HDSEL bit as '0'.

Synchronous mode supports master mode and slave mode: In master mode, the clock generated by the internal baud rate generator, it is used and output at the same time. In slave mode, the clock is input by SCLK pin. In the synchronous mode, USART can achieve data communication with SPI (At this time, users should configure the clock polarity and clock phase of SPI and USART to be consistent).

15.3.10.1 Clock description

Configure the USART_CR2.CLKEN bit to '1' to enable the clock pin function. In addition, select whether to use the internal baud rate clock or the input clock from the SCLK pin for data communication according to the USART_CR3.CKINE bit.

When the internal baud rate clock is selected, a synchronous clock can be output via the SCLK pin.

The transmission and reception of 1 data frame consist of 8 clock pulses.

When both RE and TE are '0', the clock output is stopped and fixed at the level configured by USART_CR2.CPOL.

The clock polarity is selected by configuring the USART_CR2.CPOL bit; the external clock phase is selected by configuring the USART_CR2.CPHA bit.

15.3.10.2 Clock synchronization description

When the SCLK pin is used as the clock output of the transmitter, the clock is output only in the data segment. 8 clock pulses are output for one frame of data. After the last bit is sent, the communication line holds the value of the last bit, and the clock output is fixed at a high or low level (determined by the CPOL bit).

The USART receiver works differently in synchronous mode than in asynchronous mode. If RE=1, the data is sampled on the changing SCLK edge (rising or falling edge, depending on the CPOL and CPHA bit configuration) without any oversampling. Sufficient setup time and hold time must be ensured at this point to comply with the timing requirements (similar to SPI protocol).

When the internal clock source is used, the baud rate generated by the internal baud rate

generator is calculated in the following equation:

$$f_{baudrate} = \frac{\text{PCLK}}{4 \times \text{MFD}}$$

Where the communication baud rate is in MBps; PCLK is the frequency of the internal clock source; MFD is the integer baud rate configuration of the USART_BRR (Note that in synchronous mode MFD ≥ 2 should be configured). In synchronous mode, fractional frequency division (FFD) is invalid, and users should configure FFD [3:0] to 4 'h0.

Therefore, when the internal clock source is used and MFD=2, the maximum baud rate for synchronous mode is PCLK/8 (MBps).

When the external clock source is used, the maximum frequency required for the external input clock is PCLK/8 (MHz), at which time the maximum baud rate is also PCLK/8 (MBps).

15.3.11 Single-wire half-duplex communication

Configure the USART_CR3.HDSEL bit to '1' to enable the single-wire half-duplex mode.

In single-wire half-duplex mode, the TX and RX pins are connected through the internal logic of the chip, meanwhile:

- The RX pin is left suspended and not involved in the transmission. The TX of the USART is directly connected to the TX of another USART during transmission.
- When data is being transmitted, the TX remains occupied until the stop bit is sent.
- TX is always released when no data is transmitted. Thus, it acts as a standard IO in idle or in reception. It means that the IO corresponding to TX must be configured as floating input (or output high open-drain) when not driven by the USART.

Apart from the configuration of the single-wire pins, the rest configuration is similar to what is done in normal transmission.

Before the communication, both USARTs have RXEN enabled to stay in the wait-to-receive state. When communication is required, both USARTs have to agree on which will be sent, with RE off and TE enabled in the USART_CR1 register of the sender. If both USARTs try to send data, a conflict will occur (the hardware will not block USARTs from sending when the transmitters enable bit TE is set, TX will transmit data as long as USART_DR is written).

15.3.12 Hardware flow control

The USART module supports controlling the serial data flow between two devices by using the nCTS input and the nRTS output.

RTS and CTS flow control can be enabled independently by writing respectively RTSE and CTSE bits in the USART_CR3 register.

If the RTS flow control is enabled, then nRTS outputs a low level and requests to send as long as the USART receiver is ready to receive new data. When the receive register is full or a receive error (overrun error, framing error, and parity error) is generated, nRTS outputs a high level, indicating that the sender should abort the transmission of the next data frame.

If the CTS flow control is enabled, then the transmitter checks the nCTS before transmitting the next frame. If the nCTS input is a low level, then the next data frame is transmitted immediately (assuming that data is ready to transmit, in other words, if TXE=0), else the transmission does not occur. Changes in nCTS during transmission do not affect the current frame of data being sent.

When USART_CR3.CTSE=1, the USART_SR.CTS flag will be automatically set by hardware whenever nCTS changes. Configure USART_CR3.CTSIEN =1 to enable CTS interrupt request.

15.3.13 Interrupts

The USART module supports the following interrupt sources:

Table 15-1 UART interrupt requests

Interrupt Event	Interrupt Status Bit	Enable Bit	UART	Synchronous mode
Transmit data register empty	TXE	TXEIEN	√	√
CTS flag	CTS	CTSIEN	√	√
Transmission complete	TC	TCIEN	√	√
Receive data register full	RXNE	RXNEIEN	√	√
Idle line detected	IDLE	IDLEIEN	√	-
Parity error	PE	PEIEN	√	-
Noise flag	NF	ERRIEN	√	-
Overrun error	ORE	ERRIEN	√	√
Framing error	FE	ERRIEN	√	-

Note: "√" means the interrupt is used. "-" indicates that the interrupt is not used.

15.3.14 DMA

The USART is capable of schlepping data using the DMA (DMA needs to be configured in advance). The USART_CR3.DMAMODE bit should be set to '1' to activate DMA mode.

Transmission using DMA: The DMA transport data to the USART_DR. After the UE/TE of USART_CR1 is enabled, the DMA is requested to transmit data to USART_DR whenever the transmit buffer is empty.

Reception using DMA: The DMA transport data away from the USART_DR. After the UE/RE of USART_CR1 is enabled, the DMA is requested to transport data whenever the receive buffer contains valid data (USART_SR.RXNE=1).

15.4 Register

15.4.1 Overview of registers

Table 15-2 Overview of USART registers

Offset	Acronym	Register Name	Reset
0x00	USART_SR	Status register	0x0000_00C0
0x04	USART_DR	Data register	0x0000_01FF
0x08	USART_BRR	Baud rate register	0x0000_0000
0x0C	USART_CR1	Control register 1	0x0000_0000
0x10	USART_CR2	Control register 2	0x0000_0000
0x14	USART_CR3	Control register 3	0x0000_6000

15.4.2 USART_SR status register

Address offset: 0x00

Reset value: 0x0000 00C0

31	30	29	28	27	26	25	24	23	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						CTS	Res.	TXE	TC	RXNE	IDLE	ORE	NF	FE	PE
						rcw0		r	r	rcw0	r	r	r	r	r

Bit	Field	Description
31:10	Reserved	Reserved, must be kept at reset value
9	CTS *Note 1	CTS flag 0: No change occurred on the CTS status line 1: A change occurred on the CTS status line
8	Reserved	Reserved, must be kept at reset value
7	TXE	Transmit data register Empty 0: Transmit data register is not empty (data is not transferred to the shift register) 1: Transmit data register is empty (data is transferred to the shift register) TXE bit is set and cleared automatically by hardware. The hardware will clear TXE when data is not transferred to the shift register (when written to DR register); the hardware will set this bit to '1' when data is transferred from DR to shift register.
6	TC	Transmission Complete 0: Transmission is not complete 1: Transmission is complete TC clearing condition: Write the transmit data to the data register when TE=1. TC setting condition: TE=0 or the transmit data register USART_DR is not updated when the last bit of the data frame is sent.
5	RXNE *Note 1	Receive data register not empty 0: Valid data is not received 1: Valid data is received Note: RXNE bit is set and cleared by hardware. It can also be cleared by writing '0' to it. RXNE bit is set automatically by hardware when the valid data is received. It is cleared by hardware after reading the received data.
4	IDLE *Note 2	IDLE frame detected 0: No Idle frame is detected 1: Idle frame is detected This bit is set automatically by hardware when an idle frame is detected.
3	ORE *Note 2	OverRun error 0: No overrun error 1: Overrun error is detected Note: This bit is set automatically by hardware when a new data frame is received while RXNE=1 (there already exists readable data).
2	NF *Note 2	Noise detected flag 0: No noise is detected 1: Noise is detected Note: This bit is set automatically by hardware when noise is detected on a received signal line.
1	FE *Note 2	Framing error 0: No framing error is detected 1: Framing error occurred This bit is set automatically by hardware when: In asynchronous (UART) mode, the stop bit of the received data frame is low. Note: The received data will be transferred from the shift register to the data register when FE=1, but no RXNE interrupt request signal will be generated, and the subsequent data reception will be stopped.
0	PE *Note 2	Parity error 0: No parity error 1: Parity error This bit is set automatically by hardware when a parity error occurs during data reception. Note: The received data will be transferred from the shift register to the data register when PE=1, but no RXNE interrupt request signal will be generated, and the subsequent data reception will be stopped.

*Note 1: This bit can be cleared via software by writing '0' to it.

*Note 2: This bit can be cleared by software sequence (reading the status register and then performing a read access to the USART_DR data register).

15.4.3 USART_DR data register

Address offset: 0x04

Reset value: 0x0000 01FF

31	30	29	28	27	26	25	24	23	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							DR[8:0]								
							rw								

Bit	Field	Description
31:9	Reserved	Reserved, must be kept at reset value
8:0	DR[8:0]	Transmit /Receive data register Contains the Received or Transmitted data character, depending on whether it is read from or written to. When reading, it means receiving data; when transmitting, it means sending data. The most significant bit DR [8], is valid only in asynchronous (UART) mode when the data length is set to 9 bits (DL=1).

15.4.4 USART_BRR baud rate register

Address offset: 0x08
Reset value: 0x0000 0000

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

Bit	Field	Description
31:16	Reserved	Reserved, must be kept at reset value
19:4	MFD [15:0]	Mantissa frequency division of baudrate These 16 bits define the mantissa division of the USART baud rate. Before transmission or reception is enabled (TE or RE set to 1), this value must be changed according to the baud rate requirement.
3:0	FFD[3:0]	Fraction frequency division of baudrate These 4 bits define the fraction division of the USART baud rate. Before transmission or reception is enabled (TE or RE set to 1), this value must be configured according to the baud rate requirement. Note: When FFD [3:0] =4'h0, the fractional division function is disabled. In asynchronous (UART) mode and if USART_CR1.OVER8=1, FFD [3] is invalid so please configure FFD [3] =0. In synchronous mode, the fractional division is invalid, and should configure FFD [3:0] =4'h0.

15.4.5 USART_CR1 control register 1

Address offset: 0x0C
Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved														SAS	MLS
														rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OVER8	Res.	UE	DL	Res.	PCE	PS	PEIEN	TXEIEN	TCIEN	RXNEIEN	IDLEIEN	TE	RE	Res.	SBK
rw		rw	rw		rw	rw	rw	rw	rw	rw	rw	rw	rw		rw

Bit	Field	Description
31:18	Reserved	Reserved, must be kept at reset value

Bit	Field	Description
17	SAS	Synchronous/Asynchronous mode selection 0: UART mode (asynchronous) 1: Synchronous mode (synchronous) Note: User should configure this bit at TE=0 and RE=0.
16	MLS	MSB/LSB mode selection 0: LSB mode 1: MSB mode Note: User should configure this bit at TE=0 and RE=0.
15	OVER8	UART oversampling mode 0: 16x oversampling 1: 8x oversampling Note: User should configure this bit at TE=0 and RE=0.
14	Reserved	Reserved, must be kept at reset value
13	UE	USART enable 0: USART prescaler and pin outputs disabled 1: USART enabled When this bit is cleared, the USART prescaler and outputs are stopped and the current transmission is stopped in order to reduce power consumption. Note: This bit is set and cleared by software.
12	DL	Data length 0: 8 bits 1: 9 bits Note: User should configure this bit at TE=0 and RE=0.
11	Reserved	Reserved, must be kept at reset value
10	PCE	Parity control enable 0: Parity control disabled 1: Parity control enabled Note: This bit is set or cleared by software. It must be kept at reset value in synchronous mode.
9	PS	Parity selection 0: Even parity 1: Odd parity Note: This bit is set and cleared by software. It is only valid when PCE=1.
8	PEIEN	PE interrupt enable 0: PE interrupt request disabled 1: PE interrupt request enabled Note: This bit is set and cleared by software.
7	TXEIEN	TXE interrupt enable 0: TXE interrupt request disabled 1: TXE interrupt request enabled Note: This bit is set and cleared by software.
6	TCIEN	Transmission complete interrupt enable 0: TC interrupt request disabled 1: TC interrupt request enabled Note: This bit is set and cleared by software.
5	RXNEIEN	RXNE interrupt enable 0: RXNE interrupt request disabled 1: RXNE interrupt request enabled Note: This bit is set and cleared by software.
4	IDLEIEN	IDLE interrupt enable 0: IDLE interrupt request disabled 1: IDLE interrupt request enabled Note: This bit is set and cleared by software.
3	TE	Transmitter enable 0: Transmitter disabled 1: Transmitter enabled Note: This bit is set and cleared by software. In synchronous mode, if simultaneous transmission and reception are required, user must configure both TE and RE bits at the same time to ensure proper timing sequence of clock and data.
2	RE	Receiver enable 0: Receiver disabled 1: Receiver enabled Note: This bit is set and cleared by software. In synchronous mode, if simultaneous transmission and reception are required, user must configure both RE and TE bits at the same time to ensure proper timing sequence of clock and data.
1	Reserved	Reserved, must be kept at reset value

Bit	Field	Description
0	SBK	Send break 0：No break frame is transmitted 1: Break frame will be transmitted This bit set is used to send break frame. It can be set by software and cleared automatically by hardware after sending the break frame.

15.4.6 USART_CR2 control register 2

Address offset: 0x10
Reset value: 0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SWAP	Res.	STOP[1:0]		Res.	CPOL	CPHA	Res.								
rw		rw			rw	rw									

Bit	Field	Description
31:15	Reserved	Reserved, must be kept at reset value
15	SWAP	Swap pin between input and output 0: IO pin functions are not swapped 1: Input and output of IO pin functions are swapped Note: Once the SWAP bit is set, the MODE in the GPIOx_CRL register needs to be changed, eg. the original input mode changes to the output mode.
14	Reserved	Reserved, must be kept at reset value
13:12	STOP[1:0]	STOP bit UART mode: 00: 1 stop bit 10: 2 stop bits 01: Reserved 11: Reserved
11	Reserved	Reserved, must be kept at reset value
10	CPOL	Clock polarity 0: The clock is at a low level when idle 1: The clock is at a high level when idle Note: This bit works in conjunction with the CPHA bit to produce the desired clock/data relationship (only valid in synchronous mode).
9	CPHA	Clock phase 0: The first clock transition is the first data capture edge 1: The second clock transition is the first data capture edge. Note: This bit works in conjunction with the CPOL bit to produce the desired clock/data relationship (only valid in synchronous mode).
8:0	Reserved	Reserved, must be kept at reset value

15.4.7 USART_CR3 control register 3

Address offset: 0x14
Reset value: 0x0000 6000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.		TXTOG	RXTOG	Reserved											CKINE
		rw	rw												rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.				ONEBIT	CTSIEN	CTSE	RTSE	Res.	DMAMODE	Res.		HDSEL	Res.		ERRIEN
				rw	rw	rw	rw		rw			rw			rw

Bit	Field	Description
31:30	Reserved	Reserved, must be kept at reset value
29	TXTOG	Transmit toggle bit 0: Transmit toggle disabled 1: Transmit signal level's toggle enabled
28	RXTOG	Receive toggle bit 0: Receive toggle disabled 1: Receive signal level's toggle enabled
27:17	Reserved	Reserved, must be kept at reset value
16	CKINE	Clock input permission in synchronous mode This bit is used to control whether the clock is input externally (This bit is valid when CR1. SAS bit is' 1 '): 0: Clock is not input externally 1: Clock is input externally Note: User should configure this bit at TE=0 and RE=0.
15:12	Reserved	Reserved, must be kept at reset value
11	ONEBIT	UART one sample bit method enable 0: Three samples (majority decision) 1: One sample Note: User should configure this bit at TE=0 and RE=0. When one sample bit method is selected, the noise detected flag (USART_SR.NF) will be invalid.
10	CTSIEN	CTS interrupt enable 0: CTS interrupt request disabled 1: CTS interrupt request enabled
9	CTSE	CTS enable 0: CTS hardware flow control disabled 1: CTS hardware flow control enabled
8	RTSE	RTS enable 0: RTS hardware flow control disabled 1: RTS hardware flow control enabled
7	Reserved	Reserved, must be kept at reset value
6	DMAMODE	DMA mode enable bit 0: Select polling or interrupt mode 1: Select DMA mode
5:4	Reserved	Reserved, must be kept at reset value
3	HDSEL	Single wire Half-duplex selection 0: Full-duplex mode 1: Half-duplex mode
2:1	Reserved	Reserved, must be kept at reset value
0	ERRIEN	Error interrupt enable 0: Error interrupt request disabled 1: Error interrupt request enabled Error interrupts include FE, ORE, and NF. Note: when using DMA to read or write to DR (USART_CR3.DMAMODE=1), user should configure ERRIEN=1 to allow interrupt requests that sending to CPU for USART's abnormal communication.

16 ADC Analog-to-digital converter

16.1 Introduction

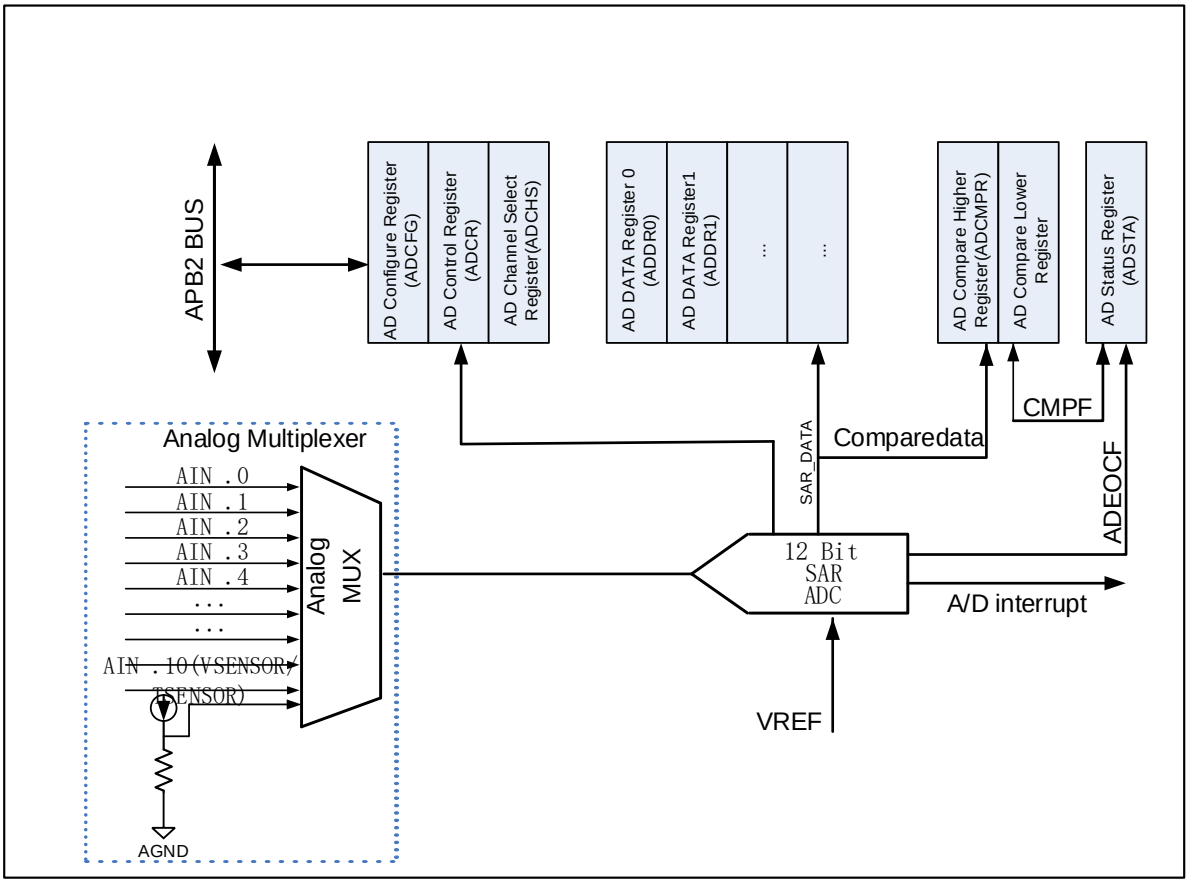
The 12-bit ADC is a successive approximation analog-to-digital converter (SAR), which converts an analog signal to a digital signal.

ADC has 11 channels allowing it to measure signals from external and internal signal sources. Among them, ADC has 10 external input channels and 1 internal channel. These ADC channels are available for single-shot, one-cycle and continuous conversions. Depending on the method, a normal channel conversion, any channel conversion, or injected channel conversion can be selected.

The ADC input clock is generated from the APB clock (PCLK) divided by a prescaler and it must not exceed 16M.

16.2 Functional block diagram

17 Figure 16-1 ADC system block diagram



Note: T_SENSOR (temperature sensor) and V_SENSOR (internal reference voltage) are in the ADC's AIN10 channel.

17.1 Main features

- Conversion rate up to 1Msps
- Normal channel conversion supported
  - Single conversion mode: one conversion is performed in the specified channel

- One-cycle scan mode: one-cycle conversion is performed in all specified channels (from low sequence number channel to high sequence number channel, or from high sequence number channel to low sequence number channel)
- Continuous scan mode: continuous one-cycle scan conversions are performed until the software stops A/D conversion. A/D conversion must stop if the conversion channel needs to be changed. The conversion can restart after configuring the corresponding register

- Any channel conversion supported
  - Single conversion mode: one conversion is performed in the specified channel
  - One-cycle scan mode: one-cycle conversion is performed in all specified channels according to the channel configuration
  - Continuous scan mode: continuous one-cycle scan conversions are performed until the software stops A/D conversion. The user doesn't have to stop conversion if the channel needs to be changed during that period. The conversion in a new configured channel will start in the next scan cycle after configuring the corresponding channel register
- Injected channel conversion supported
  - Auto-injection: in any channel conversion mode, the injected channels are automatically converted after the operation of any channel is complete.
- Programmable channel sampling time
- Up to 12-bit programmable resolution SAR
- DMA transfer supported
- A/D conversion start condition
  - Software start
  - Trigger start with configurable trigger delay
- Analog watchdog function. The conversion result is compared with the specified threshold; when the conversion result exceeds the specified threshold, an interrupt is generated if the ADC_ADCR.AWDIE is set

17.2 Interrupt

ADC interrupts and flag bits are shown in the following table. When an event occurs, the corresponding interrupt is generated if the interrupt enable bit is set.

Table 16-1 Interrupts

Interrupt	Flag Bit
Analog watchdog comparator interrupt	AWDIF
End of sequential conversion interrupt	EOSIF
End of conversion interrupt	EOCIF
End of sample interrupt	EOSMPIF
End of injected sequence interrupt	JEOSIF
End of injected conversion interrupt	JEOCIF
End of injected sample interrupt	JEOSMPIF

17.3 DMA

A/D conversion results are stored in the data register ADC_ADDATA. This data allow DMA access when multiple channels are converted continuously to avoid conversion data loss.

After DMA is enabled, a DMA request will be generated when the conversion is finished, which is to transfer the conversion data from the ADC_ADDATA register to the destination specified by the software.

Each channel has its corresponding data register ADC_ADDRn, which can be accessed to get its respective conversion result.

17.4 Functional description

17.4.1 Clock

The ADC input clock is synchronous with PCLK. ADC clock control bit in the RCC controller should be set before using the ADC.

17.4.2 Data offset

The data from a non-injected channel doesn't have a data offset value.

The converted value from an injected channel decreased by the offset defined in the ADC_JOFRn register is then stored in the injected channel data registers ADC_JADDATA and ADC_JDRn. The value decreased by the offset may be a negative value. Therefore, the converted result from the injected channel is a signed value (the SEXT bit in the data alignment represents the extended sign value).

If ADC_JOFRn.JOFFSET is not set to 0, then the data in the injected channel data registers ADC_JADDATA and ADC_JDRn are signed values, otherwise they are unsigned values.

The following table describes the data resolution in relation to the left aligned data offset:

Table 16-2 Data resolution in relation to the left aligned data offset

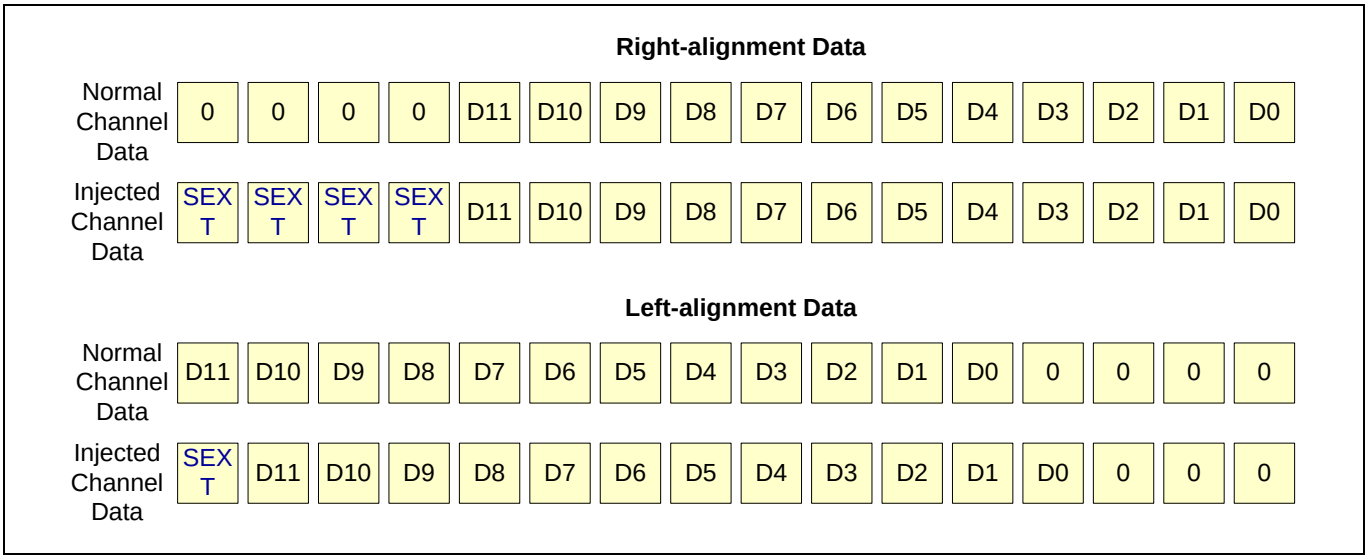
Resolution option	Analog conversion data SAR_DATA, left shift	Offset value	Result	Description
12bit	SAR_DATA[11:0]	ADC_JOFRn.JOFFESET	12-bit signed data	-
11bit	SAR_DATA[11:1],0	ADC_JOFRn.JOFFESET	11-bit signed data	User must set ADC_JOFRn.JOFFSET[0] to '0'
10bit	SAR_DATA[11:2],00	ADC_JOFRn.JOFFESET	10-bit signed data	User must set ADC_JOFRn.JOFFSET[1:0] to '00'
9bit	SAR_DATA[11:3],000	ADC_JOFRn.JOFFESET	9-bit signed data	User must set ADC_JOFRn.JOFFSET[2:0] to '000'
8bit	SAR_DATA[11:4],0000	ADC_JOFRn.JOFFESET	8-bit signed data	User must set ADC_JOFRn.JOFFSET[3:0] to '0000'

17.4.3 Data alignment

ADC_ADCR.ALIGN bit selects the data after conversion is stored in a left-alignment or right-alignment manner.

See the diagram below:

Figure 16-2 Data alignment style



17.4.4 Programmable resolution

The ADC conversion significance can be configured using the ADC_ADCFG.RSLTCTL [2:0] bits. The significant data bits are aligned from the right bits of 12-bit data by default. Low resolution configuration may accelerate the data conversion speed.

17.4.5 Programmable sampling time

The sampling time of ADC conversion channels can be configured by SAMPCTL0~SAMPCTL10 in ADC_SMPR1 and ADC_SMPR2 registers.

The ADC conversion clock ADC_CLK is generated from the PCLK divided by a prescaler. The prescale coefficient can be determined by setting the ADC_ADCFG. ADCPRE bit, i.e. PCLK/ (ADCPRE +2) divided by a prescaler to be the ADC conversion clock. ADC samples the input voltage for a number of ADC_CLK cycles. The number of sampling cycles (m) can be modified using the ADC_SMPR1 and ADC_SMPR2 registers. Set the ADC resolution as n bits (n=8, 9, 10, 11, 12). The sampling cycles of each channel is m*T (T is the clock cycle of ADC module).

The conversion frequency is calculated as follows:

$F_{sample} = F_{ADC\_CLK} /(m+n+0.5)$.

Suppose the resolution (n) is set to 12 bits, and the sampling cycle of each channel (m) is 3.5T, then $F_{sample}=F_{ADC\_CLK}/16$.

The total conversion time is calculated as follows:

T_{CONV} = Sampling time + 12.5 conversion cycles

Example: With an ADC_CLK = 16MHz and a sampling time of 3.5T, T_{CONV}= (3.5+12.5)*T=16*T, the total conversion time is 1μs.

17.4.6 Data channel register

After the ADC conversion is complete, the conversion result from a non-injected channel is stored in the ADC_ADDATA register, and ADC_ADDATA.CHANNELSEL indicates the channel number corresponding to the current data.

The conversion result from an injected channel is stored in the ADC_JADDATA register, and ADC_JADDATA.JCHANNELSEL indicates the injected channel number corresponding to the current data.

17.4.7 Channel selection

ADC has 8 external input channels 0~9, internal temperature sensor and internal 1.2V reference voltage channel 10.

Different registers can be used to enable channels in different operating modes: ADC_ADCHS register can be set for normal channel conversion. ADC_ANY_CFG, ADC_CHANY0, and ADC_CHANY1 registers can be set for any channel conversion; the sequence of any channel conversion is organized from highest to lowest priority among CHANY_SEL0 to 15 bits in the register. JSQR register can be set for injection channel conversion; the sequence of injection channel conversion is organized from highest to lowest priority among JSQ0 to 3 bits in the register.

17.5 ADC on-off control

The ADC_ADCFG.ADEN bit controls the A/D converter. When the ADEN bit is set to 0, the analog converter enters Power Down mode. The A/D converter wakes up from the Power Down mode if the ADEN is set to 1. The ADC_ADCFG.ADEN bit can be cleared to stop the conversion and then the ADC enters Power Down mode, after which the ADC consumes almost no power.

17.5.1 Normal channel conversion

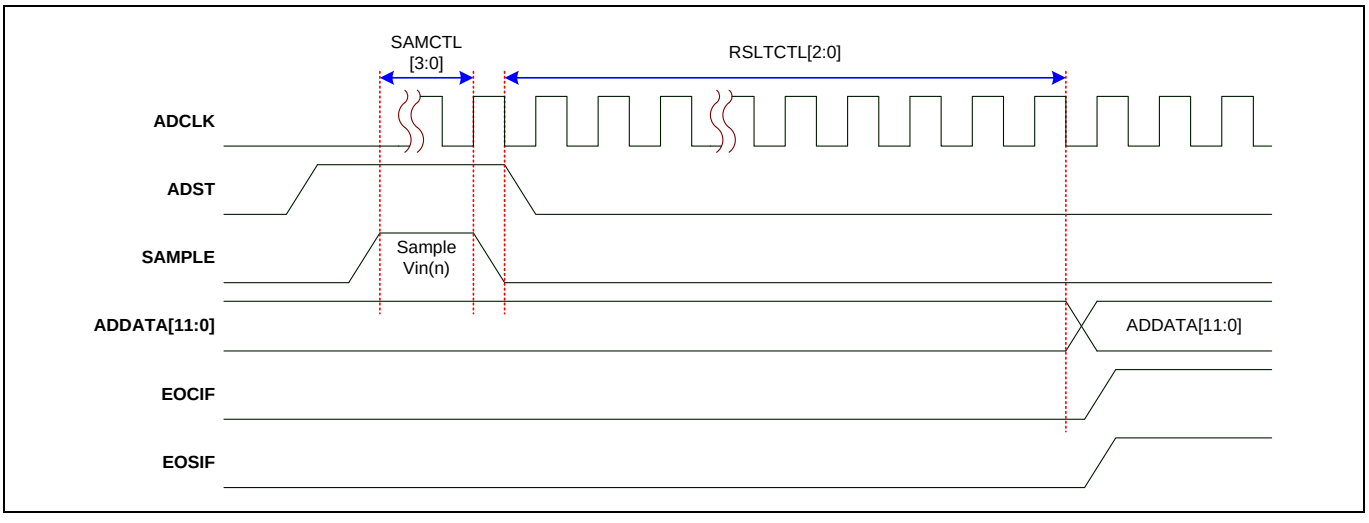
17.5.1.1 Single sampling

A single conversion is selected by configuring ADC_ADCR.ADMODE. The configured channel enters the idle state after one conversion. The specific procedures are as follows:

- The A/D conversion is started by setting the ADC_ADCR.ADST bit via external trigger input, or timer trigger or software.
- By default, the sampling output SAMPL is cleared after 2.5 ADC clocks since it is set.
- After A/D conversion, the SAR conversion result will be stored in the data registers ADC_ADDATA and ADC_ADDRn.
- At that time, the ADC_ADSTA.EOSIF bit in the status register is set to 1; meanwhile, an AD EOC (End of Conversion) interrupt request is generated if the ADC_ADCR.EOSIE bit in the control register is set to 1.
- ADC_ADCR.ADST bit remains 1 during the A/D conversion period. Once the A/D channel conversion is complete, the ADC_ADCR.ADST bit is automatically cleared by hardware and the A/D converter enters the idle state.

Note: In Single conversion mode, the channel of lowest sequence is converted while others are ignored, if more than one channel is enabled via software.

Figure 16-3 Single conversion mode timing diagram



17.5.1.2 One-cycle scan mode

ADC_ADCR.ADMODE is configured as one-cycle conversion. In One-cycle scan mode, scan channel direction can be selected by setting ADC_ADCR.SCANDIR. The A/D conversion is performed in sequence for channels enabled by ADC_ADCHS.CHENy(y=0~8), with the following operation steps:

- Start the A/D conversion either by trigger with the configuration of delay or by setting the ADC_ADCR.ADST bit via software. The default direction is from the lowest sequence number channel to the highest sequence number channel.
- The channel outputs are organized according to the conversion channel configuration.
- After the end of A/D conversion in each channel, the converted result SAR_DATA will be loaded to the data register (ADC_ADDATA and ADC_ADDRn) of corresponding channel in order. The ADC_ADSTA_EXT.EOCIF flag is set when the conversion in the current channel is complete. At this time, an end of conversion interrupt request is generated if the EOC interrupt bit is set. At the end of sequential conversion, the ADC_ADSTA.EOSIF (end of sequential conversion interrupt) flag is set. If the end of sequential conversion interrupt is enabled, then the corresponding request will be generated.
- Once the final A/D conversion is complete, the ADC_ADCR.ADST bit is automatically cleared by hardware and the A/D converter enters the idle state.

Figure 16-4 Enable channel conversion timing diagram in One-cycle scan mode (channel direction: from high to low)

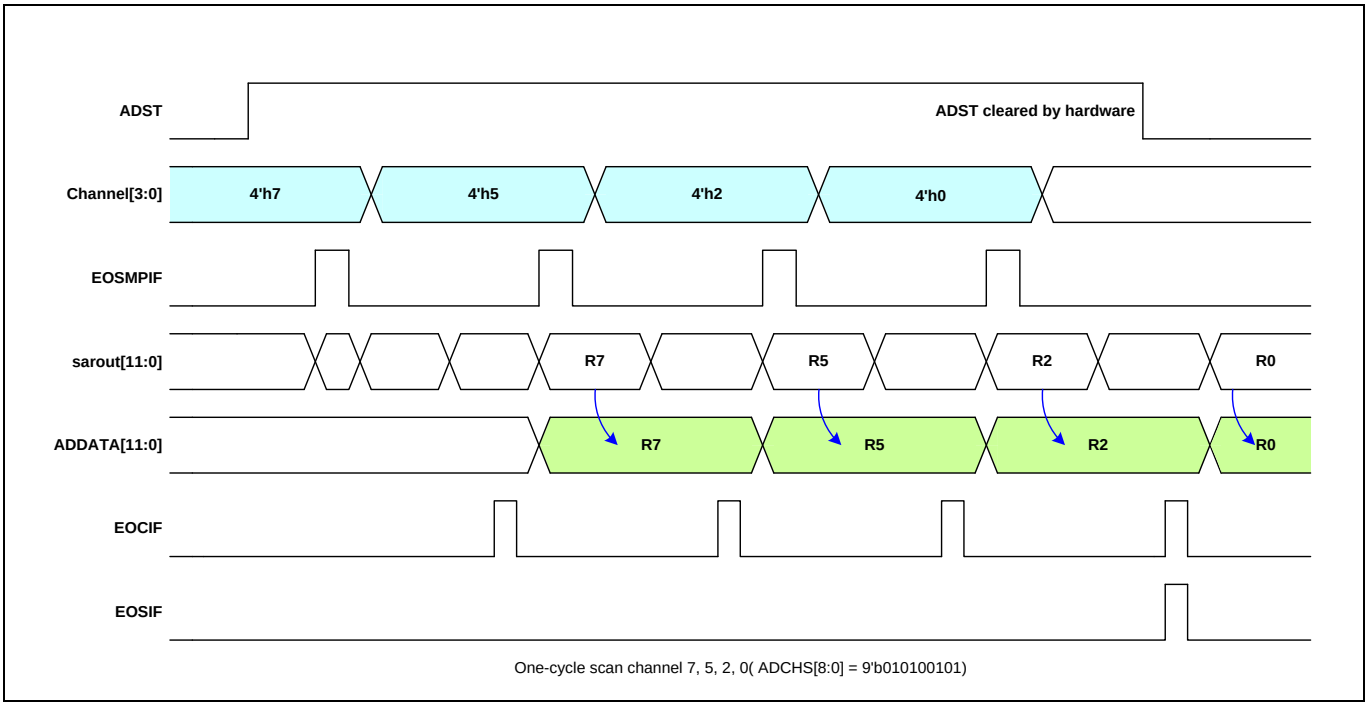
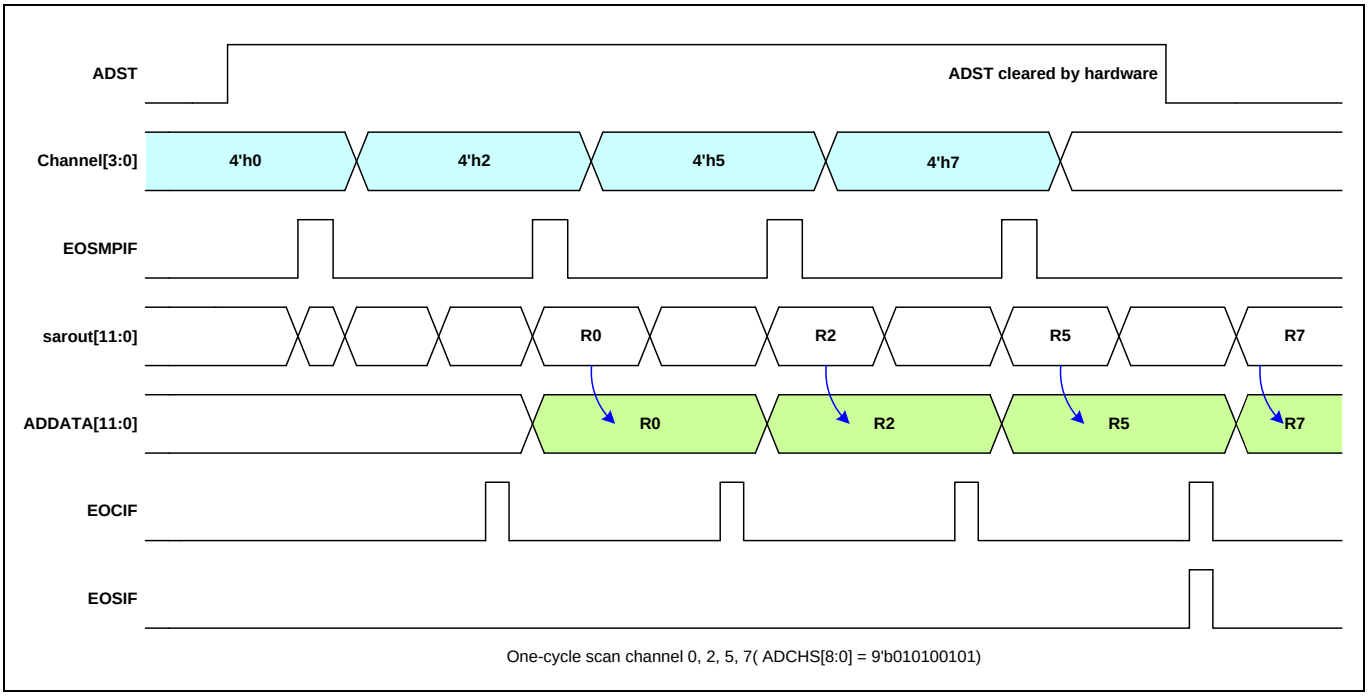


Figure 16-5 Enable channel conversion timing diagram in One-cycle scan mode (channel direction: from low to high)



17.5.1.3 Continuous scan mode

ADC_ADCR.ADMODE is configured as continuous conversion. In continuous scan mode, scan channel direction can be selected by setting ADC_ADCR.SCANDIR. The A/D conversion is performed in sequence for channels enabled by ADC_ADCHS.CHENy(y=0~8), with the following operation steps:

- Start the A/D conversion either by trigger with the configuration of delay or by setting the ADC_ADCR.ADST bit via software. The default direction is from the lowest sequence number channel to the highest sequence number channel.
- The channel outputs are organized according to the conversion channel configuration.
- After the end of A/D conversion in each channel, the converted result SAR_DATA will be loaded to the data register (ADC_ADDATA and ADC_ADDRn) of corresponding channel in order. The ADC_ADSTA_EXT.EOCIF flag is set when the conversion in the current channel is complete. At this time, an end of conversion interrupt request is generated if the EOC interrupt bit is set. At the end of sequential conversion, the ADC_ADSTA.EOSIF (end of sequential conversion interrupt) flag is set. If the end of sequential conversion interrupt is enabled, then the corresponding request will be generated.
- The A/D conversion continues as long as the ADC_ADCR.ADST bit remains 1. Once the ADC_ADCR.ADST bit is cleared and the conversion stops, the converter enters the idle state. Once the ADC_ADCR.ADST bit is cleared, the A/D converter will put an end to the current conversion.

Figure 16-6 Enable channel conversion timing diagram in continuous scan mode (channel direction: from low to high)

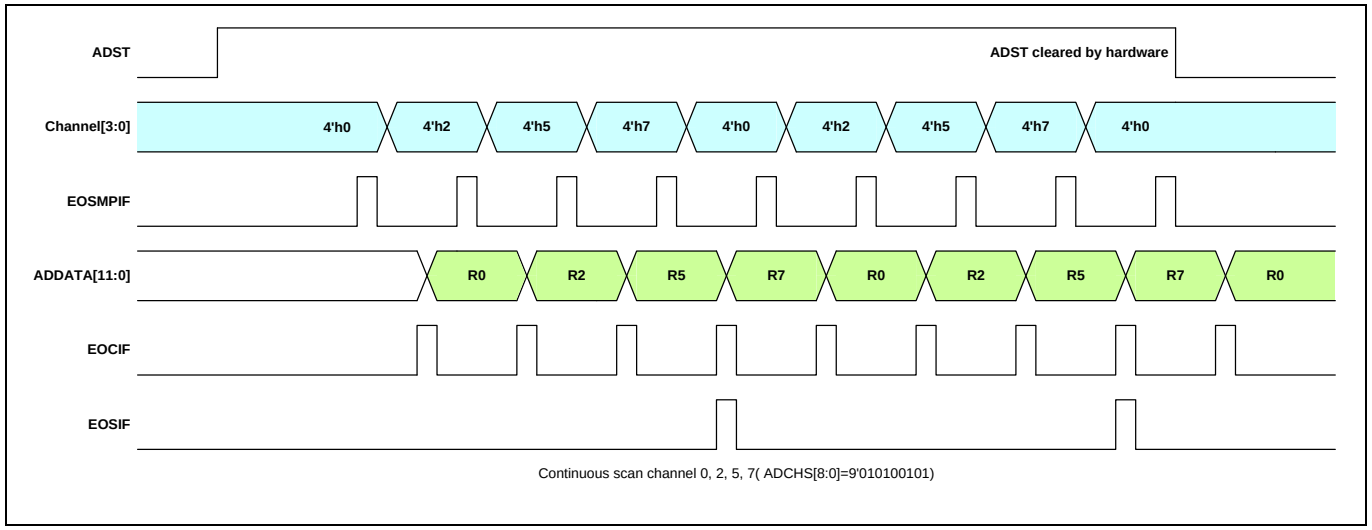
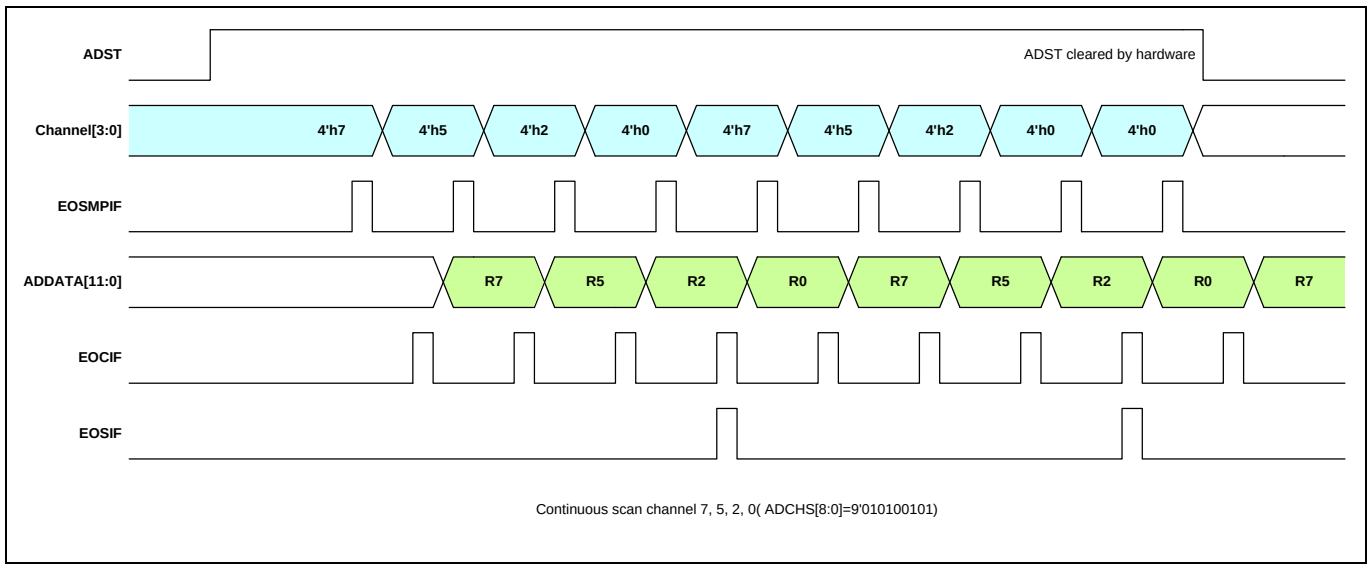


Figure 16-7 Enable channel conversion timing diagram in continuous scan mode (channel direction: from high to low)



17.5.2 Any channel conversion

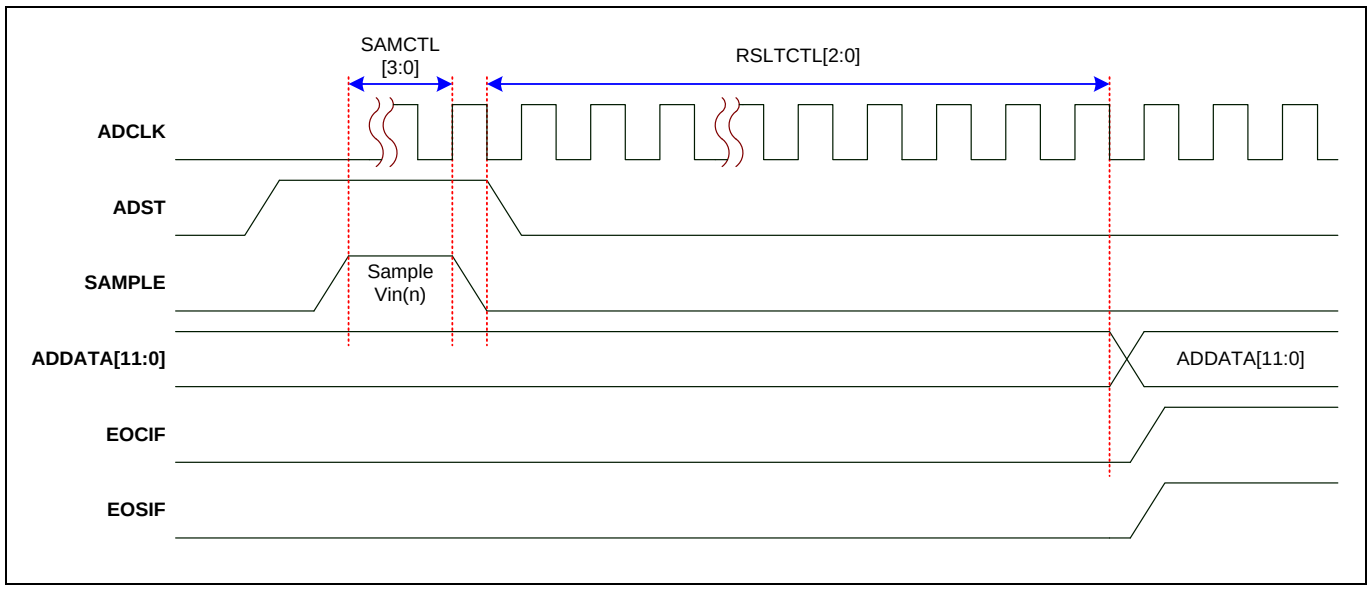
17.5.2.1 Single conversion mode

A single conversion is selected by configuring ADC_ADCR.ADMODE. The configured channel enters the idle state after one conversion. The specific procedures are as follows:

- Set by software the ADC_ANY_CFG, ADC_CHANY0, and ADC_CHANY1 registers, desired channels, and the ADC_ANY_CR.CHANY_MDEN bit.
- Set the ADC_ADCR.ADST bit by trigger with the configuration of trigger delay or by software. Conversion channels can be arbitrarily configured through registers ADC_CHANY0 and ADC_CHANY1.
- By default, the sampling output SAMPL is cleared after 2.5 ADC clocks since it is set.
- After the end of A/D conversion, the converted result SAR_DATA will be loaded to ADC_ADDATA and ADC_ADDR0. The ADC_ADSTA.EOSIF flag is set when the conversion in the channel is complete. At this time, an end of conversion interrupt request is generated if the EOC interrupt is enabled.

- Once the final A/D channel conversion is complete, the ADST bit is automatically cleared by hardware and the A/D converter enters the idle state.

Figure 16-8 Channel conversion timing diagram in Single conversion mode

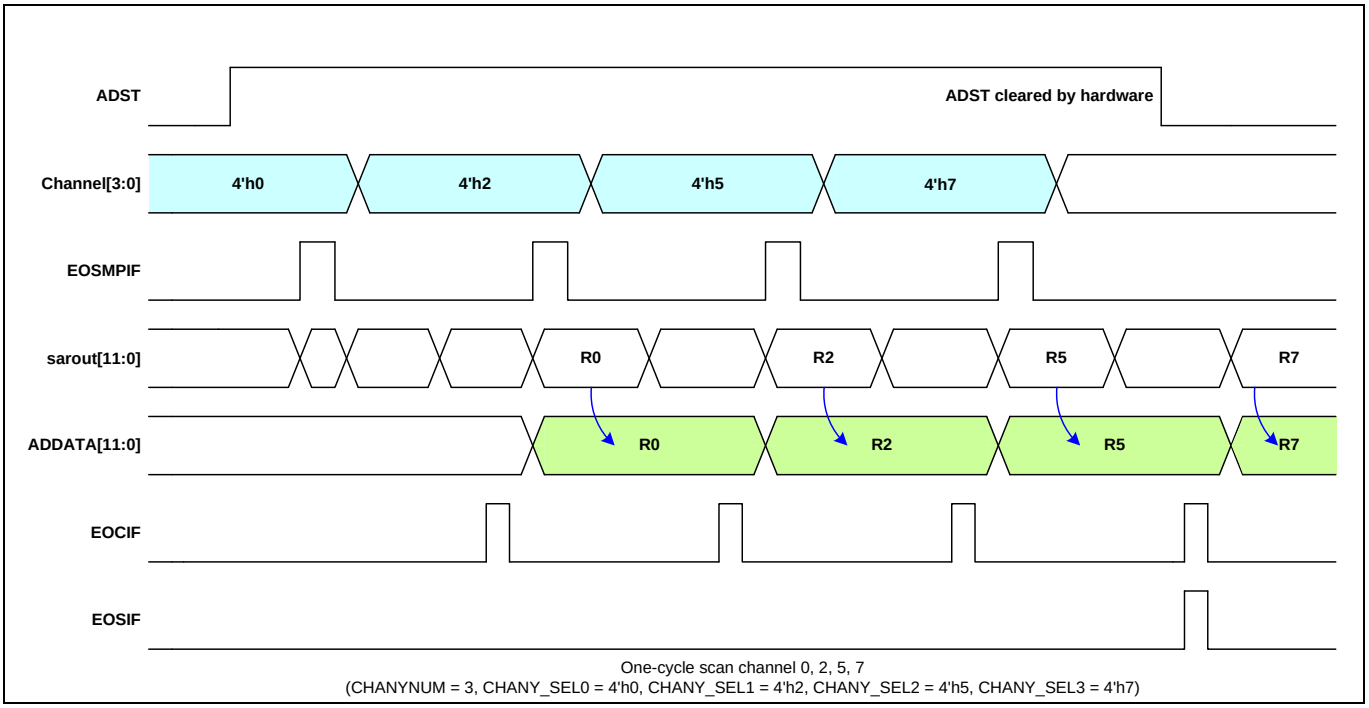


17.5.2.2 One-cycle scan mode

A one-cycle conversion is selected by configuring ADC_ADCR.ADMODE. In One-cycle scan mode the A/D converter does one conversion according to the software configuration. The specific procedures are as follows:

- Set by software the ADC_ANY_CFG, ADC_CHANY0, and ADC_CHANY1 registers, desired channels and quantity, and the ADC_ANY_CR.CHANY_MDEN bit.
- Set the ADC_ADCR.ADST bit by trigger with the configuration of trigger delay or by software. The number of A/D conversion channels is configured by ADC_ANY_CFG.CHANY_NUM. The input channels for each conversion can be arbitrarily configured through CHANY_SEL0 to CHANY_SEL15 bits in the register, which can be identical or not.
- The channel outputs are organized according to the conversion channel configuration.
- After the end of A/D conversion in each channel, the converted result SAR_DATA will be loaded to the data register (ADC_ADDATA and ADC_ADDRn) of corresponding channel in order. The ADC_ADSTA_EXT.EOCIF flag is set when the conversion in the current channel is complete. At this time, an end of conversion interrupt request is generated if the EOC interrupt bit is set. At the end of sequential conversion, the ADC_ADSTA.EOSIF (end of sequential conversion interrupt) flag is set. If the end of sequential conversion interrupt is enabled, then the corresponding request will be generated.
- Once the final A/D channel conversion is complete, the ADST bit is automatically cleared by hardware and the A/D converter enters the idle state.
- If the software updates registers ADC_ANY_CFG, ADC_CHANY0, and ADC_CHANY1 during the A/D conversion period, these settings will not be applied immediately. They will be applied when the conversion of the final channel is complete. The next conversion is started by setting the ADC_ADCR.ADST bit via software.

Figure 16-9 Channel conversion timing diagram In One-cycle scan mode



17.5.2.3 Continuous scan mode

Continuous conversions are selected by configuring ADC_ADCR.ADMODE. In continuous scan mode the A/D converter does continuous conversions according to the software configuration until the software stops conversion. The specific procedures are as follows:

- Set by software the ADC_ANY_CFG, ADC_CHANY0, and ADC_CHANY1 registers, desired channels and quantity, and the ADC_ANY_CR.CHANY_MDEN bit.
- Set the ADC_ADCR.ADST bit by trigger with the configuration of trigger delay or by software. The number of A/D conversion channels is configured by ADC_ANY_CFG.CHANY_NUM. The input channels for each conversion can be arbitrarily configured through CHANY_SEL0 to CHANY_SEL15 bits in the register, which can be identical or not.
- The channel outputs are organized according to the conversion channel configuration.
- After the end of A/D conversion in each channel, the converted result SAR_DATA will be loaded to the data register (ADC_ADDATA and ADC_ADDRn) of corresponding channel in order. The ADC_ADSTA_EXT.EOCIF flag is set when the conversion in the current channel is complete. At this time, an end of conversion interrupt request is generated if the EOC interrupt bit is set. At the end of sequential conversion, the ADC_ADSTA.EOSIF flag is set. If the end of sequential conversion interrupt is enabled, then the corresponding request will be generated.
- The A/D conversion continues as long as the ADC_ADCR.ADST bit remains 1. Once the ADC_ADCR.ADST bit is cleared by software, the current A/D conversion stops after it is complete, and the A/D converter enters the idle state.
- If the user updates registers ADC_ANY_CFG, ADC_CHANY0, and ADC_CHANY1 during the A/D conversion period, these settings will not be applied immediately. They will be applied when the conversion of the final channel is complete, i.e. the time a conversion in the new channel starts in the next scan cycle.

Figure 16-10 Channel conversion timing diagram in Continuous scan mode

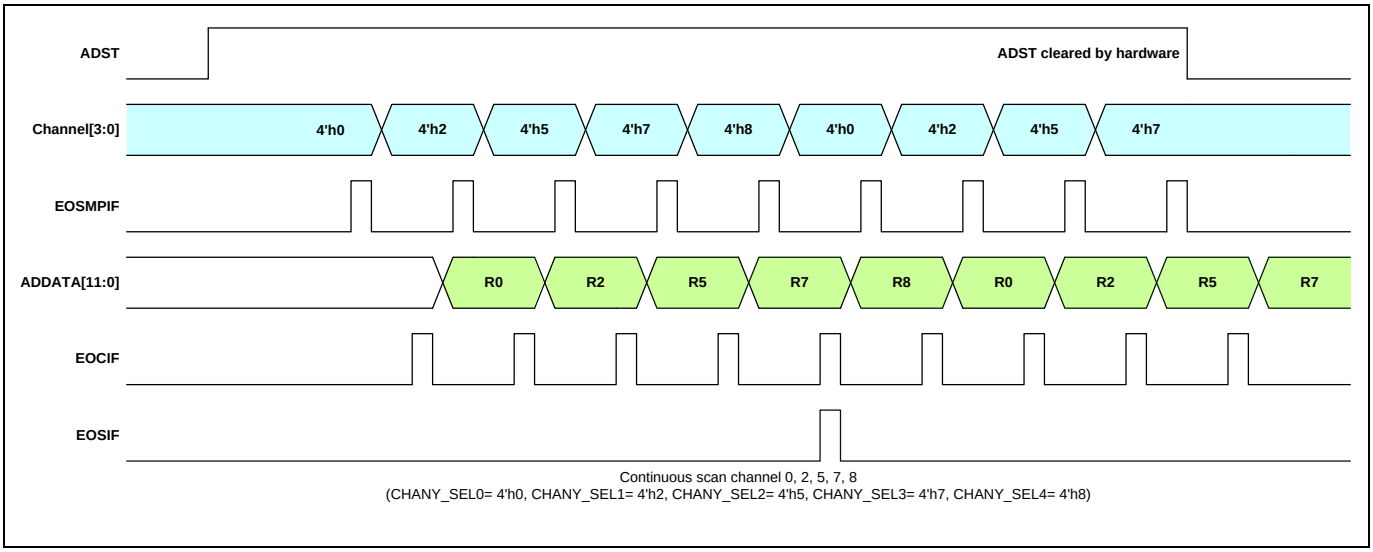
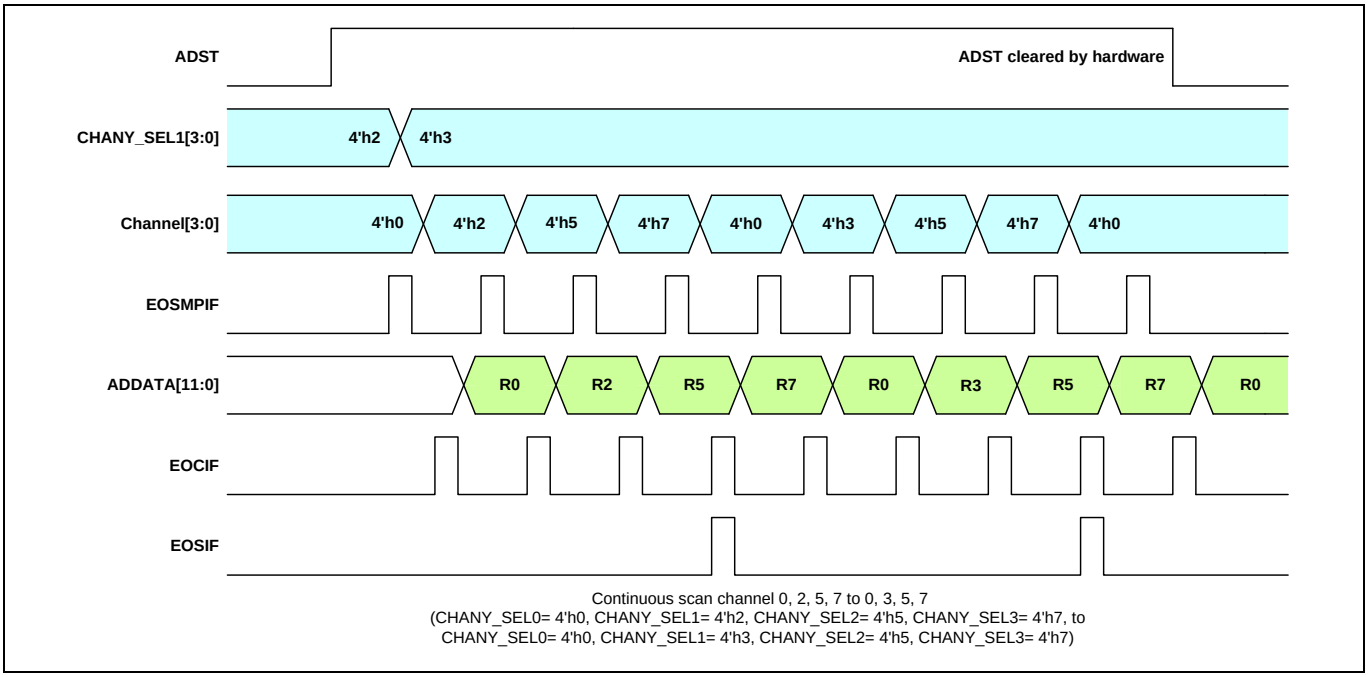


Figure 16-11 Timing diagram with dynamically updated configuration in Continuous scan mode



17.5.3 Injected channel conversion

17.5.3.1 Auto-injection conversion

In the any channel conversion mode, the injected channels are automatically converted after the operation of any channel is complete, if the auto-injection conversion is enabled; if any channel is in continuous scan mode, the A/D conversion can only be stopped by resetting the ADC_ADCR.ADST bit.

- Set by software the ADC_ANY_CFG, ADC_CHANY0, ADC_CHANY1, and ADC_JSQR registers, the ADC_ANY_CFG.JAUTO bit, desired channels and quantity, and finally the ADC_ANY_CR.JCEN and ADC_ANY_CR.CHANY_MDEN bits.
- Set the ADC_ADCR.ADST bit by trigger with the configuration of trigger delay or by software. The number of any A/D conversion channels is configured by ADC_ANY_CFG.CHANY_NUM. The input

channels for each conversion can be arbitrarily configured through CHANY_SEL0 to CHANY_SEL15 bits in the register. The injected channels are converted after the operation of any channel is complete. The number of conversion channels is configured by JCHANY_NUM. The input channels for each conversion can be configured through JSQ0 to JSQ3 bits in the register.

- The channel outputs are organized according to the conversion channel configuration.
- After the end of A/D conversion in each channel, the converted result SAR_DATA will be loaded to the data register (ADC_ADDATA and ADC_ADDRn, ADC_JDATA and ADC_JDRn) of corresponding channel in order. The ADC_ADSTA_EXT.EOCIF and ADC_ADSTA_EXT.JEOCIF flags are set when the conversion in the current channel is complete. At this time, an end of conversion interrupt request is generated if the EOC interrupt is enabled. At the end of continuous conversion, the ADC_ADSTA.EOSIF and ADC_ADSTA_EXT.JEOSIF flags are set. If the end of sequential conversion interrupt is enabled, then the corresponding request will be generated.
- If the scan is in discontinuous mode, the ADC_ADCR.ADST bit is automatically cleared by hardware once the conversion is complete; otherwise, the A/D conversion continues as long as the ADC_ADCR.ADST bit remains 1. Once the ADC_ADCR.ADST bit is cleared, the current A/D conversion stops after it is complete, and the A/D converter enters the idle state.

Figure 16-12 Auto-injection conversion timing diagram in One-cycle scan mode

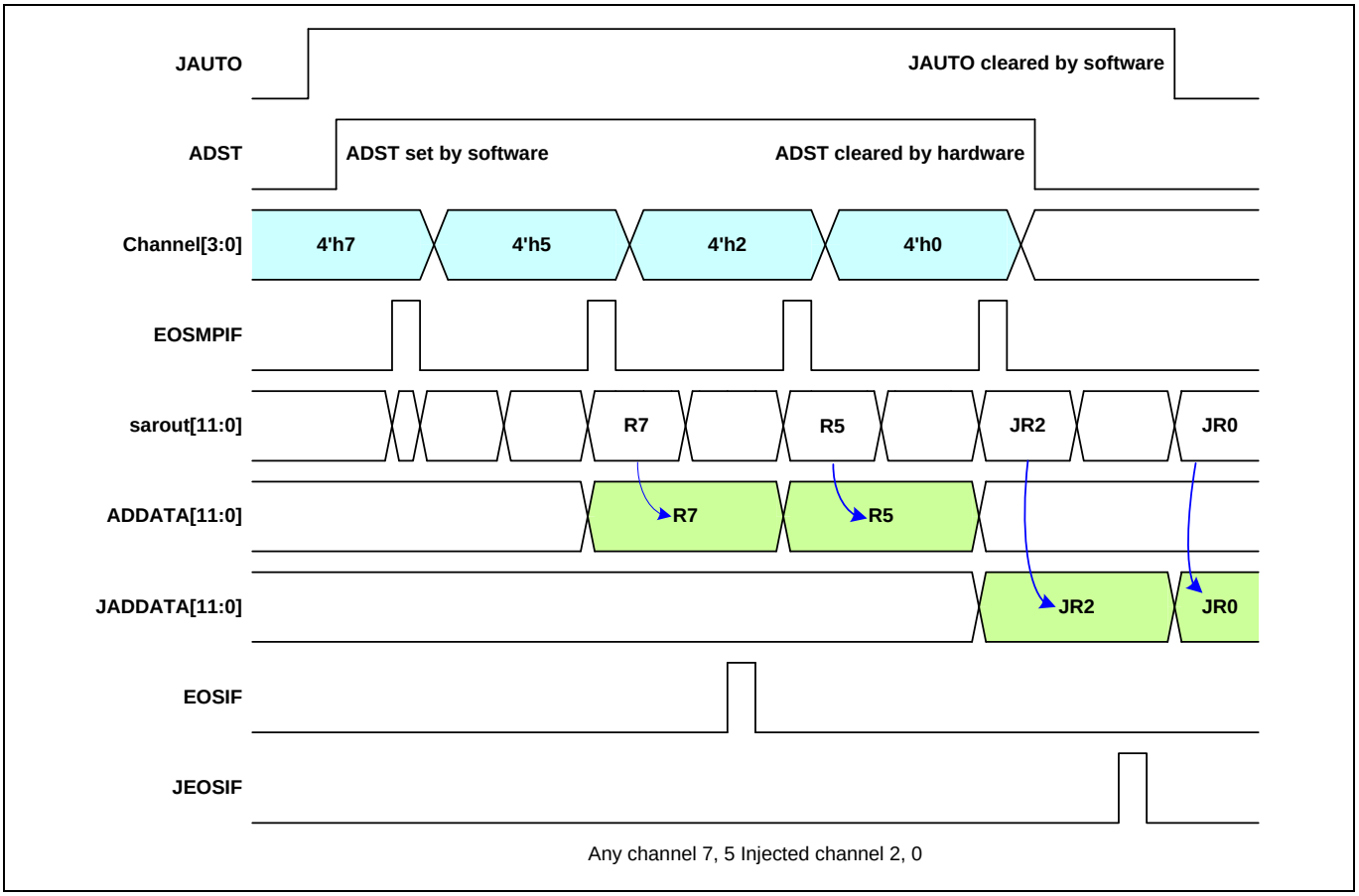
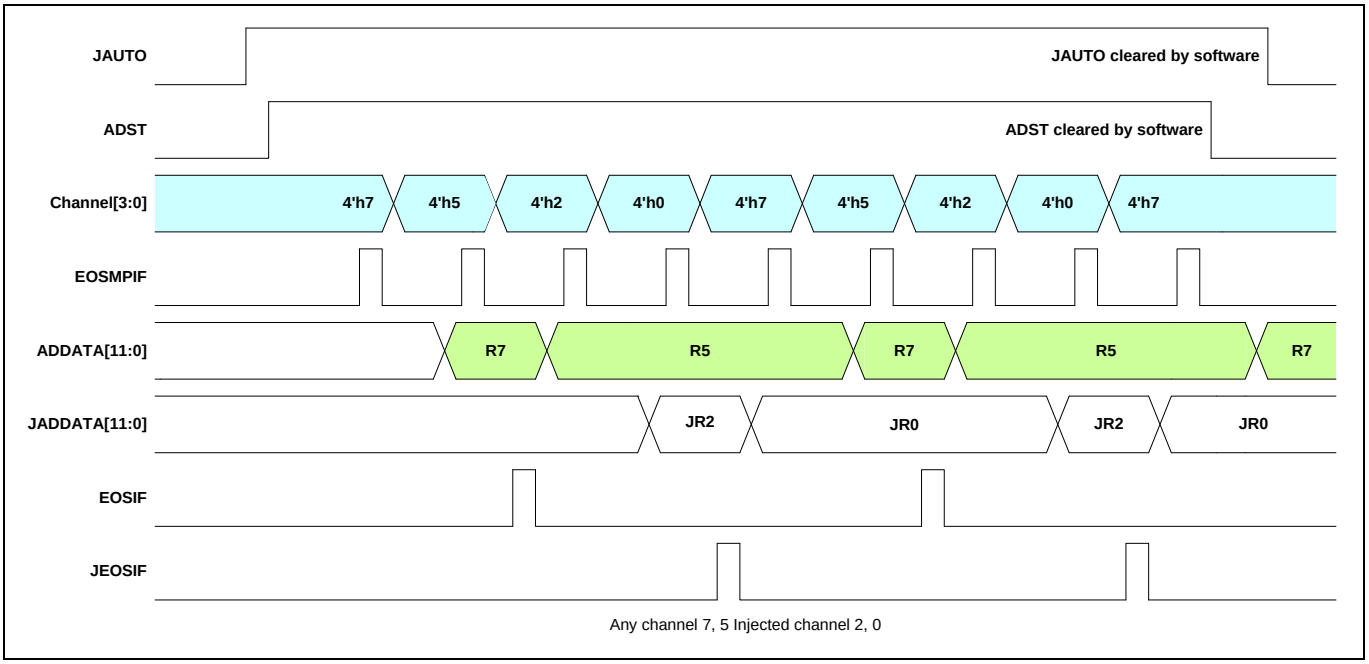


Figure 16-13 Auto-injection conversion timing diagram in Continuous scan mode



17.5.3.2 Event injection working mode

If any channel conversion is currently in progress in any channel conversion mode and an injection event occurs (including software and trigger) after the automatic injection is turned off, the current conversion will stop immediately and the injected channel conversion will start. Wait for all injected channels to complete conversion according to the configuration of register ADC_JSQR and then complete the remaining arbitrary channel conversion until ADC_ADCR. ADST bit is cleared by software or hardware; if there is no current conversion, the injected channel conversion will start directly.

- Configure by software the ADC_ANY_CFG, ADC_CHANY0, ADC_CHANY1 and ADC_JSQR registers and set ADC_ANY_CR.JCEN and ADC_ANY_CR.CHANY_MDEN after the configuration of the channels and quantities to be converted is completed.
- Set the ADC_ADCR.ADST bit by trigger or software and the trigger event can configure trigger delay. The number of any A/D conversion channel is configured by ADC_ANY_CFG.CHANY_NUM, and the input channel for each conversion can be configured arbitrarily by register bit CHANY_SEL0 to CHANY_SEL15.
- The injected channel conversion and the number of conversion channel are configured by JCHANY_NUM. The input channel for each conversion can be configured through JSQ0 to JSQ3 bits in the register.
- The channel output is organized in order according to the conversion channel configuration.
- After the end of A/D conversion in each channel, the converted result SAR_DATA will be loaded to the data register (ADC_ADDATA and ADC_ADDRn, ADC_JDATA and ADC_JDRn) of corresponding channel in order. The ADC_ADSTA_EXT.EOCIF and ADC_ADSTA_EXT.JEOCIF flags are set when the conversion in the current channel is completed. At this time, an end of conversion interrupt request is generated if the EOC interrupt is enabled. At the end of continuous conversion, the ADC_ADSTA.EOSIF and ADC_ADSTA_EXT.JEOSIF flags are set. If the end of sequential conversion interrupt is enabled, then the corresponding request will be generated.

- If the scan is in discontinuous mode, the ADC_ADCR.ADST bit is automatically cleared by hardware once the conversion is completed; otherwise, the A/D conversion continues as long as the ADC_ADCR.ADST bit remains 1. Once the ADC_ADCR.ADST bit is cleared, the current A/D conversion stops after it is completed, and the A/D converter enters the idle state.

Figure 16-14 Event injection channel conversion timing diagram at any channel conversion 1

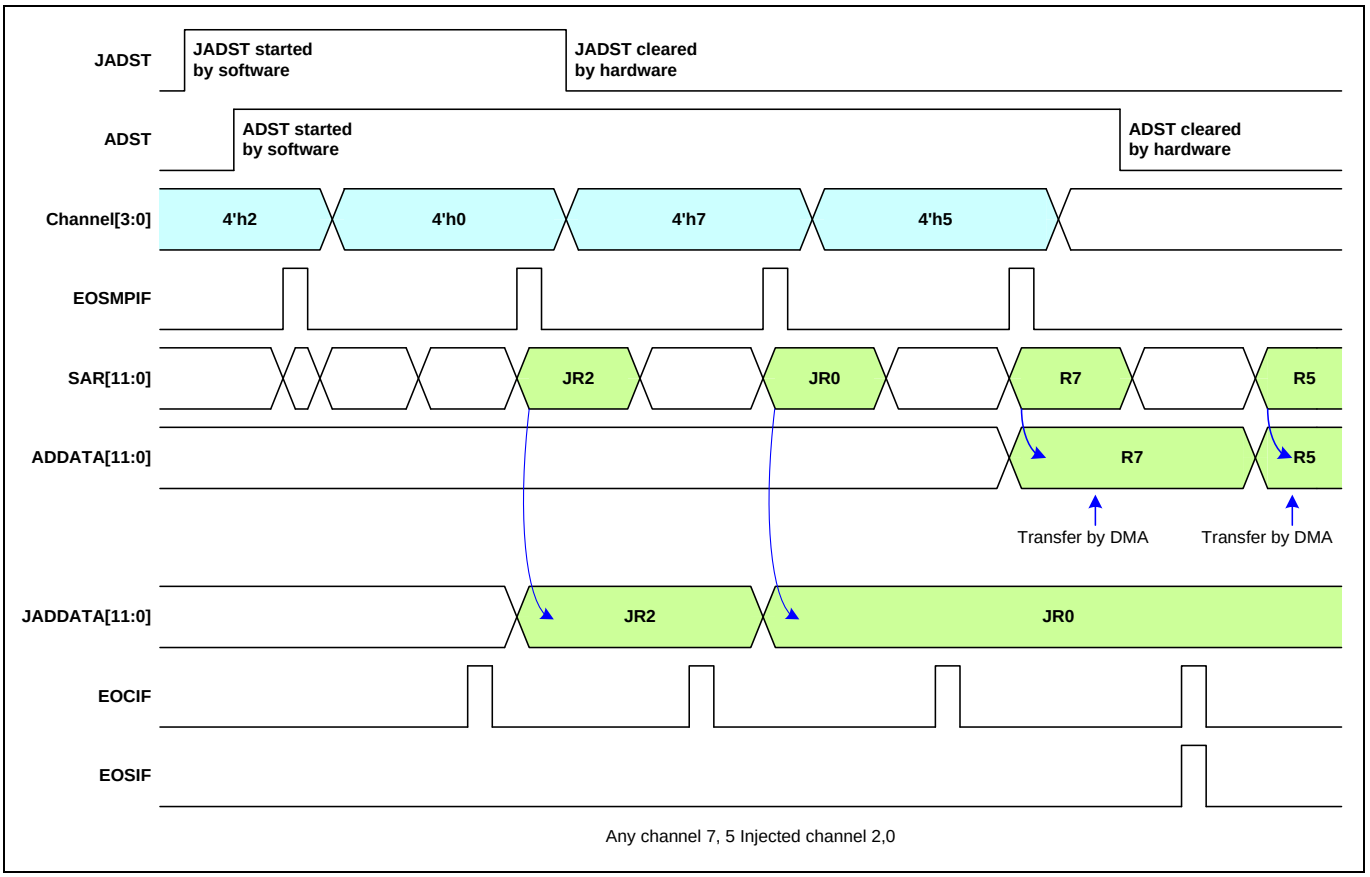
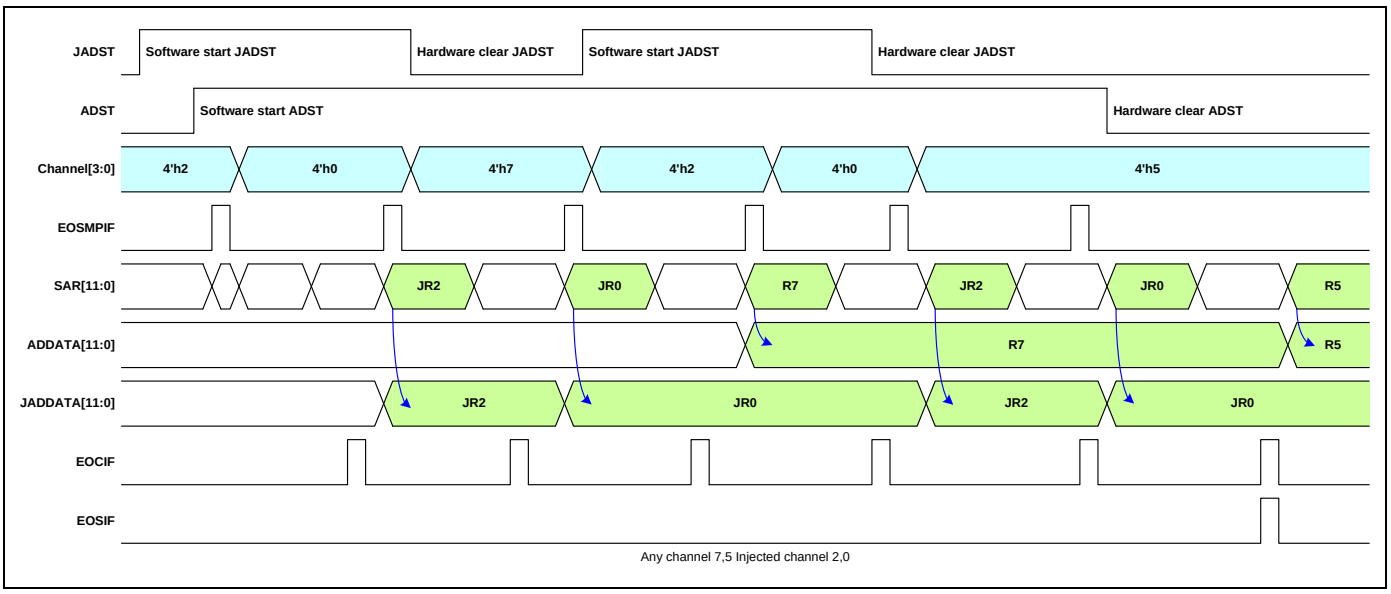


Figure 16-15 Event injection channel conversion timing diagram at any channel conversion 2



17.5.4 ADC trigger signal

In addition to software, the trigger sources for ADC conversion include timers and external events.

After a trigger signal is generated, the sampling starts after a delay of N PCLK clock cycles (configurable). In case of trigger scan mode, only the sampling in the first channel is delayed and the rest start immediately after the last conversion is complete.

If the ADC_ADCR.TRGEN bit is set then external events are able to trigger a conversion of non-injected channel.

An external trigger source of non-injected conversion can be selected by setting the ADC_ADCR.TRGSEL bit.

Refer to the description of ADC_ADCR.TRGSEL in the A/D control register for details of external trigger source selection. Refer to relevant description of ADC_ADCR.TRGSHIFT in the AD control register for details of external trigger delay control.

17.5.5 Analog watchdog

The watchdog compares the data before the data alignment. An upper limit comparison register and a lower limit comparison register are provided by the watchdog comparator. The monitoring channel can be selected by setting the ADC_ADCR.CMPCH bit in the register. Providing ADC_ADCMPR.CPMHDATA is greater than or equal to ADC_ADCMPR.CPMLDATA, and if analog watchdog is enabled on non-injected channels (ADC_ADCFG.AWDEN), and the result from non-injected channels is greater than or equal to the ADC_ADCMPR.CMPHDATA value or less than the ADC_ADCMPR.CMPLDATA value, the ADC_ADSTA.AWDIF bit in the status register is set to 1; if analog watchdog is enabled on injected channels (ADC_ADCFG.JAWDEN), and the result from injected channels is greater than or equal to the ADC_ADCMPR.CMPHDATA value or less than the ADC_ADCMPR.CMPLDATA value, the ADC_ADSTA.AWDIF bit in the status register is also set to 1.

Providing ADC_ADCMPR.CPMHDATA is less than ADC_ADCMPR.CPMLDATA, and if analog watchdog is enabled on non-injected channels (ADC_ADCFG.AWDEN), and the result from non-injected channels is equal to the ADC_ADCMPR.CPMHDATA value or between two specified values, the ADC_ADSTA. AWDIF bit in the status register is set to 1; if analog watchdog is enabled on injected channels (ADC_ADCFG.JAWDEN), and the result from injected channels is equal to the ADC_ADCMPR.CPMHDATA value or between two specified values, the ADC_ADSTA. AWDIF bit in the status register is also set to 1.

An interrupt request is generated if the ADC_ADCR.AWDIE bit in the control register is set.

17.5.6 Internal temperature sensor

The built-in temperature sensor can be used to measure the internal temperature changes (TA) of the device. If accurate temperature readings are needed, an external temperature sensor part should be used.

The temperature sensor is enabled by setting the ADC_ADCFG.VTSEN bit; the temperature sensor is disabled by clearing the VTSEN bit.

A temperature sensor channel can be selected by setting the ADC_ADCHS.CHENTS bit.

The temperature value can be calculated as follows:

T (℃) = ((Value * V_{DDA}-V_{25}*3300) / (4096*Avg_Slope)) +25

V_{DDA}: The VDDA voltage of the ADC when it is currently sampled, in mV

V_{25}: Converted value of the temperature sensor at 25℃, stored in flash space 0x1FFFF7E6.

Value: Value is the converted result of ADC.

Avg_Slope: Average Slope for curve between Temperature vs. Voltage (given in mV/℃)

Refer to the Temperature Sensor section in the data sheet for the typical values of V_{25} and Avg_Slope.

17.5.7 Internal voltage sensor

The internal voltage signal source channel of ADC is connected to an internal reference voltage Vref (about 1.2V). This channel converts the 1.2V reference voltage output to a digital value, and the Vref will be tested and recorded at the factory with 3.3V as the supply voltage of the MCU, and the calibration value is stored in the lower 12 bits of the flash space 0x1FFFF7E0. The user can calculate the current system analog reference voltage value based on the recorded calibration value. The calculation formula is as below:

Vref = (Vref_cal *3.3V) / 4096

Vref/ADC_ADDR(Vsensor) = VDDA/4096

VDDA = (Vref_cal * 3.3V) / ADC_ADDR(VSensor)

The variables are defined as follows:

Vref: Internal reference voltage value in V;

Vref_cal: Internal reference voltage factory-calibrated value which is stored in the lower 12 bits of the flash space 0x1FFFF7E0;

ADC_ADDR(Vsensor): The ADC internal voltage channel conversion value.

The internal reference voltage has an independent enable bit that can be enabled or disabled by setting the ADC_ADCFG.VTSEN bit in the register.

17.6 Register description

17.6.1 Overview of registers

Table 16-3 Overview of ADC register

Offset	Acronym	Register Name	Reset
0x00	ADC_ADDATA	Data Register	0x00000000
0x04	ADC_ADCFG	Configuration Register	0x00000000
0x08	ADC_ADCR	Control Register	0x00000000
0x0C	ADC_ADCHS	Channel Selection Register	0x00000000
0x10	ADC_ADCMPR	Analog Watchdog Compare Register	0x00000000
0x14	ADC_ADSTA	Status Register	0x00000000
0x18~0x40	ADC_ADDR 0~10	Channel Data Register	0x00000000
0x58	ADC_ADSTA_EXT	Extended State Register	0x00000000
0x5C	ADC_CHANY0	Any Channel Selection Register 0	0x00000000
0x60	ADC_CHANY1	Any Channel Selection Register 1	0x00000000
0x64	ADC_ANY_CFG	Any Channel Configuration Register	0x00000000
0x68	ADC_ANY_CR	Any Channel Control Register	0x00000000
0x70	ADC_SMPR1	Sample Configuration Register 1	0x00000000
0x74	ADC_SMPR2	Sample Configuration Register 2	0x00000000
0x7C~0x88	ADC_JOFR0~3	Injected Channel Data Offset Register	0x00000000
0x8C	ADC_JSQR	Injected Sequence Register	0x00000000
0x90	ADC_JADDATA	Injected Data Register	0x00000000
0xB0~0xBC	ADC_JDR0~3	Injected Channel Data Register	0x00000000
0xF0	ADC_LDATA	Last Conversion Data Register	0x00000000
0xF4	ADC_TRGSUPR	External Rule Trigger Event Suppression Register	0x00000000

17.6.2 ADC_ADDATA Data Register

Address offset: 0x00
Reset value: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.										VAILD	OVERRUN	CHANNELSEL			
										r	r	r			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA															
r															

Bit	Field	Description
31:22	Reserved	Reserved, must be kept at reset value.
21	VALID	Valid flag for data (read-only) 1: DATA[11:0] bits valid 0: DATA[11:0] bits invalid This bit is set by hardware after the conversion is complete. It is cleared automatically after reading the ADC_ADDATA register.
20	OVERRUN	Data overrun flag (read-only) 1: DATA[11:0] for last conversion is overrun 0: DATA[11:0] for last conversion is not overrun Before the data is loaded to the DATA [11:0] bits in the data register, OVERRUN is set to '1' if the DATA [11:0] for last conversion is not read. This bit is cleared automatically after reading the ADC_ADDATA register.
19:16	CHANNELSEL	4 bits show the channel corresponding to the data in the current data register (Channel Selection) 0000: indicates that the corresponding channel of the current data register is channel 0 0001: indicates that the corresponding channel of the current data register is channel 1 0010: indicates that the corresponding channel of the current data register is channel 2 0011: indicates that the corresponding channel of the current data register is channel 3 0100: indicates that the corresponding channel of the current data register is channel 4 0101: indicates that the corresponding channel of the current data register is channel 5 0110: indicates that the corresponding channel of the current data register is channel 6 0111: indicates that the corresponding channel of the current data register is channel 7 1000: indicates that the corresponding channel of the current data register is channel 8 1001: indicates the corresponding channel of the current data register is channel 9 1010: indicates the corresponding channel of the current data register is channel 10 (corresponding to the conversion data of internal temperature sensor or internal reference voltage) Others: invalid
15:0	DATA	12-bit A/D conversion result from the current channel (Conversion Data) The alignment style is selected via software.

17.6.3 ADC_ADCFG Configuration Register

Address offset: 0x04
Reset value: 0x00000000

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

Res.	ADCPREL	Res.	RSLTCTL	ADCPREH	VTSEN	AWDEN	ADEN
	rw		rw	rw	rw	rw	rw

Bit	Field	Description
31:17	Reserved	Reserved, must be kept at reset value.
16	JAWDEN	A/D Analog Watchdog Enable for Injected Channels 1: A/D analog watchdog enabled for injected channels 0: A/D analog watchdog disabled for injected channels
15	Reserved	Reserved, must be kept at reset value
14	ADCPREL	ADC Prescaler Low Bits Prescale coefficient ADCPRE={ADCPREH, ADCPREL}
13: 10	Reserved	Reserved, must be kept at reset value
9: 7	RSLTCTL	ADC conversion data resolution selection 000: 12 significant bits 001: 11 significant bits 010: 10 significant bits 011: 9 significant bits 100: 8 significant bits Others: reserved
6: 4	ADCPREH	ADC Prescaler High Bits Prescale coefficient ADCPRE={ADCPREH, ADCPREL} ADC clock division: div= (ADCPRE+2)
3:2	VTSEN	Internal reference voltage and Temperature Sensor enable (Voltage Sensor and Temperature Sensor Enable) 00: Internal voltage sensor and temperature sensor disabled 01: temperature sensor enabled 10: Internal voltage sensor enabled 11: Internal voltage sensor enabled
1	AWDEN	Analog watchdog enable on non-injected channels 1: analog watchdog enabled on non-injected channels 0: analog watchdog disabled on non-injected channels
0	ADEN	A/D conversion enable (ADC enable) 1: enabled 0: disabled

17.6.4 ADC_ADCR Control Register

Address offset: 0x08
Reset value: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.				EOCIE	EOSMPIE	TRG_EDGE		Res.		TRGSHIFT			TRGSELH		SCANDIR
				rw	rw	rw				rw			rw		rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CMPCH				ALIGN	ADMD		ADST	Res.	TRGSELL			DMAEN	TRGEN	AWDIE	EOSIE
rw				rw	rw		rw		rw			rw	rw	rw	rw

Bit	Field	Description
31:28	Reserved	Reserved, must be kept at reset value
27	EOCIE	Interrupt Enable for End of Conversion 1: interrupt enabled for end of single A/D conversion 0: interrupt disabled for end of single A/D conversion An A/D EOC interrupt request is generated if the EOCIF bit is set.
26	EOSMPIE	Interrupt Enable for End of Sample 1: interrupt enabled for end of A/D sampling 0: interrupt disabled for end of A/D sampling An A/D end of sample interrupt request is generated if the EOSMPIF bit is set.
25 : 24	TRG_EDGE	Trigger Sources Edge Selection 00: Double-edge trigger 01: Falling edge trigger 10: Rising edge trigger 11: Reserved

Bit	Field	Description
23:22	Reserved	Reserved, must be kept at reset value.
21:19	TRGSHIFT	External Trigger Sources Shift For Sample After a trigger event is generated, the sampling starts after a delay of N PCLK clock cycles (N=1, 2, 3…). 000: no delay 001: 4 cycles 010: 16 cycles 011: 32 cycles 100: 64 cycles 101: 128 cycles 110: 256 cycles 111: 512 cycles In case of trigger scan, the sampling in the rest channels starts immediately after the last conversion is complete.
18 :17	TRGSELH	External Trigger Sources Select For High Bits TRGSEL={TRGSELH,TRGSELL}
16	SCANDIR	ADC Channel Scan Direction Only valid for normal channel conversion 1: Backward scan from high sequence channel to low sequence channel by the ADC channel selection register 0: Upward scan from low sequence channel to high sequence channel by the ADC channel selection register
15:12	CMPCH	Compare Channel Selection For Analog Watchdog 0000: select to compare the conversion result from channel 0 0001: select to compare the conversion result from channel 1 0010: select to compare the conversion result from channel 2 0011: select to compare the conversion result from channel 3 0100: select to compare the conversion result from channel 4 0101: select to compare the conversion result from channel 5 0110: select to compare the conversion result from channel 6 0111: select to compare the conversion result from channel 7 1000: select to compare the conversion result from channel 8 1001: select to compare the conversion result from channel 9 1010: select to compare the conversion result from channel 10 1111: select to compare the conversion result from all scan channels Others: invalid
11	ALIGN	Data Alignment style 1: left aligned 0: right aligned
10 : 9	ADMD	A/D conversion mode (ADC Mode) 00: single conversion 01: one-cycle conversion 10: continuous scan 11: Reserved When the conversion mode is changed, the software has to clear the ADST bit.
8	ADST	A/D conversion start 1: conversion start 0: end of conversion or idle state There are two ways of clearing ADST: In the single or one-cycle mode, ADST is automatically cleared by hardware after conversion. In continuous scan mode, the A/D converter does continuous conversions until the software writes ‘0’ to ADST or the system is reset.
7	Reserved	Reserved, must be kept at reset value.

Bit	Field	Description
6: 4	TRGSELL	External Trigger sources Select for Low bits TRGSEL={TRGSELH,TRGSELL} ADC selects external trigger source TRGSEL: 00000： TIM1_CC1 00001： TIM1_CC2 00010： TIM1_CC3 00011： TIM2_CC2 00101： TIM1_CC4 and TIM1_CC5 00111： EXTI 11 01000： TIM1_TRGO 01001： TIM13_CC1 01010： TIM14_CC1 01011： TIM2_CC1 01101： TIM2_TRGO 01111： EXTI 15 10000： TIM1_CC4 10001： TIM1_CC5 Others: invalid
3	DMAEN	DMA 使能（Direct Memory Access Enable） 1： DMA 请求使能 0： DMA 请求禁止
2	TRGEN	External Hardware Trigger Sources Enable 1: external trigger enabled for A/D conversion 0: external trigger disabled for A/D conversion
1	AWDIE	Interrupt Enable of Analog Watchdog 1: A/D analog watchdog interrupt enabled 0: A/D analog watchdog interrupt disabled
0	EOSIE	Interrupt Enable for End of Sequence on non-injected channels 1: ADC interrupt enabled 0: ADC interrupt disabled An A/D EOC interrupt request is generated if the EOSIF bit is set.

17.6.5 ADC_ADCHS Channel Selection Register

Address offset: 0x0C
Reset value: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.					CHEN10	CHEN9	CHEN8	CHEN7	CHEN6	CHEN5	CHEN4	CHEN3	CHEN2	CHEN1	CHEN0
					rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bit	Field	Description
31:11	Reserved	Reserved, must be kept at reset value
10	CHEN10	Temperature Sensor or Voltage Sensor Enable 1: enabled 0: disabled
9	CHEN9	Analog Input Channel 9 Enable 1: enabled 0: disabled
8	CHEN8	Analog Input Channel 8 Enable 1: enabled 0: disabled
7	CHEN7	Analog Input Channel 7 Enable 1: enabled 0: disabled
6	CHEN6	Analog Input Channel 6 Enable 1: enabled 0: disabled
5	CHEN5	Analog Input Channel 5 Enable 1: enabled 0: disabled

Bit	Field	Description
4	CHEN4	Analog Input Channel 4 Enable 1: enabled 0: disabled
3	CHEN3	Analog Input Channel 3 Enable 1: enabled 0: disabled
2	CHEN2	Analog Input Channel 2 Enable 1: enabled 0: disabled
1	CHEN1	Analog Input Channel 1 Enable 1: enabled 0: disabled
0	CHEN0	Analog Input Channel 0 Enable 1: enabled 0: disabled

Note: 1) If all channels are disabled, the default conversion channel is 0 when the conversion starts according to SCANDIR configuration.

2) This register can only be operated when ADC_ADSTA.BUSY is invalid.

17.6.6 ADC_ADCMPR Analog Watchdog Compare Register

Address offset: 0x10

Reset value: 0x00000000

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

Bit	Field	Description
31:28	Reserved	Reserved, must be kept at reset value
27: 16	CMPHDATA	Compare High Threshold Data For Analog Watchdog The high threshold of comparison for the watchdog
15:12	Reserved	Reserved, must be kept at reset value.
11 : 0	CMPLDATA	Compare Low Threshold Data For Analog Watchdog The low threshold of comparison for the watchdog

17.6.7 ADC_ADSTA Status Register

Address offset: 0x14

Reset value: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.			OVERRUN									Res.			VALID
			r												r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VALID								CHANNEL				Res.	BUSY	AWDIF	EOSIF
r								r					r	rc_w1	rc_w1

Bit	Field	Description
31:29	Reserved	Reserved, must be kept at reset value
28:20	OVERRUN	Data overrun flag for channel 0 to channel 10(Overrun Flag)
19:17	Reserved	Reserved, must be kept at reset value
16:8	VALID	Valid flag for channel 0 to channel 10(Valid Flag)

Bit	Field	Description
7:4	CHANNEL	Current Convert Channel Note: these bits indicate the channel in conversion when BUSY=1; these bits indicate the channel for the next conversion when BUSY=0.
3	Reserved	Reserved, must be kept at reset value
2	BUSY	Non-Injected Conversion Busy/Idle 1 = A/D converter is busy 0 = A/D converter is idle
1	AWDIF	Analog Watchdog Flag This flag bit is cleared by writing '1'. 1: Analog watchdog event occurs 0: Analog watchdog event doesn't occur
0	EOSIF	End of Sequential Conversion Flag This bit is set by hardware at the end of sequential conversion, and cleared by software. 1: A/D conversion complete 0: A/D conversion incomplete This flag bit is cleared by writing '1'.

17.6.8 ADC_ADDRn Channel Data Register (n=0~10)

Address offset: 0x18~0x40
Reset value: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.										VAILD	OVERRUN	Res.			
										r	r				
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA															
r															

Bit	Field	Description
31:22	Reserved	Reserved, must be kept at reset value.
21	VALID	Valid flag (read-only) 1: DATA[11:0] bits valid 0: DATA[11:0] bits invalid This bit is set by hardware after the conversion in the corresponding channel is complete. It is cleared automatically after reading the ADC_ADDRn register.
20	OVERRUN	Data overrun flag (read-only) 1: DATA [11:0] data is overrun 0: DATA [11:0] for last conversion result OVERRUN is set to '1' if the DATA [11:0] data for last conversion is not read before it is loaded to the DATA [11:0] bits in the data register. This bit is cleared by hardware automatically after reading the ADC_ADDRn register.
19:16	Reserved	Reserved, must be kept at reset value.
15 : 0	DATA	12-bit A/D conversion result from the channel n (Conversion Data) The alignment style is selected via software.

17.6.9 ADC_ADSTA_EXT Extended State Register

Address offset: 0x58
Reset value: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.										JBUSY	JEOSIF	JEOCIF	JEOSMPIF	EOCIF	EOSMPIF
										r	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.															

Bit	Field	Description
31:8	Reserved	Reserved, must be kept at reset value.

Bit	Field	Description
21	JBUSY	Injected Conversion Busy/Idle Flag 1: Injected channel of A/D converter is busy 0: Injected channel of A/D converter is idle
20	JEOSIF	End of Injected Sequence Flag This bit is set by hardware at the end of sequential conversion, and cleared by software. 1: A/D conversion complete 0: A/D conversion incomplete This flag bit is cleared by writing '1'.
19	JEOCIF	End of Injected Conversion Flag This bit is set by hardware at the end of conversion, and cleared by software. 1: A/D conversion complete 0: A/D conversion incomplete This flag bit is cleared by writing '1'.
18	JEOSMPIF	End of Injected Sample Flag This bit is set by hardware at the end of sampling, and cleared by software. 1: A/D sampling complete 0: A/D sampling incomplete This flag bit is cleared by writing '1'.
17	EOCIF	End of Conversion Flag This bit is set by hardware at the end of conversion, and cleared by software. 1: A/D conversion complete 0: A/D conversion incomplete This flag bit is cleared by writing '1'.
16	EOSMPIF	End of Sample Flag This bit is set by hardware at the end of sampling, and cleared by software. 1: A/D sampling complete 0: A/D sampling incomplete This flag bit is cleared by writing '1'.
15:0	Reserved	Reserved, must be kept at reset value.

17.6.10 ADC_CHANY0 Any Channel Selection Register 0

Address offset: 0x5C
Reset value: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CHANY_SEL7				CHANY_SEL6				CHANY_SEL5				CHANY_SEL4			
rw				rw				rw				rw			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CHANY_SEL3				CHANY_SEL2				CHANY_SEL1				CHANY_SEL0			
rw				rw				rw				rw			

Bit	Field	Description
31:28	CHANY_SEL7	7th Conversion Select for Any Channel sequence 0000~1010: configure any channel from channel 0 to channel 10 as an input channel in the operation of any channel
27:24	CHANY_SEL6	6th Conversion Select for Any Channel sequence 0000~1010: configure any channel from channel 0 to channel 10 as an input channel in the operation of any channel
23:20	CHANY_SEL5	5th Conversion Select for Any Channel sequence 0000~1010: configure any channel from channel 0 to channel 10 as an input channel in the operation of any channel
19:16	CHANY_SEL4	4th Conversion Select for Any Channel sequence 0000~1010: configure any channel from channel 0 to channel 10 as an input channel in the operation of any channel
15:12	CHANY_SEL3	3th Conversion Select for Any Channel sequence 0000~1010: configure any channel from channel 0 to channel 10 as an input channel in the operation of any channel
11:8	CHANY_SEL2	2th Conversion Select for Any Channel sequence 0000~1010: configure any channel from channel 0 to channel 10 as an input channel in the operation of any channel
7:4	CHANY_SEL1	1th Conversion Select for Any Channel sequence 0000~1010: configure any channel from channel 0 to channel 10 as an input channel in the operation of any channel

Bit	Field	Description
3:0	CHANY_SEL0	0th Conversion Select for Any Channel sequence 0000~1010: configure any channel from channel 0 to channel 10 as an input channel in the operation of any channel

Note: In the one-cycle or continuous scan modes, the ADC_CHANY0 shadow register is started by hardware. When ADC is working, only the shadow register is updated if the ADC_CHANY0 value is changed. Only when ADC starts to convert the last channel, the value of the shadow register will update to ADC_CHANY0, so as to implement the dynamic channel switch.

17.6.11 ADC_CHANY1 Any Channel Selection Register 1

Address offset: 0x60
Reset value: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CHANY_SEL15				CHANY_SEL14				CHANY_SEL13				CHANY_SEL12			
rw				rw				rw				rw			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CHANY_SEL11				CHANY_SEL10				CHANY_SEL9				CHANY_SEL8			
rw				rw				rw				rw			

Bit	Field	Description
31:28	CHANY_SEL15	15th Conversion Select for Any Channel sequence 0000~1010: configure any channel from channel 0 to channel 10 as an input channel in the operation of any channel
27:24	CHANY_SEL14	14th Conversion Select for Any Channel sequence 0000~1010: configure any channel from channel 0 to channel 10 as an input channel in the operation of any channel
23:20	CHANY_SEL13	13th Conversion Select for Any Channel sequence 0000~1010: configure any channel from channel 0 to channel 10 as an input channel in the operation of any channel
19:16	CHANY_SEL12	12th Conversion Select for Any Channel sequence 0000~1010: configure any channel from channel 0 to channel 10 as an input channel in the operation of any channel
15:12	CHANY_SEL11	11th Conversion Select for Any Channel sequence 0000~1010: configure any channel from channel 0 to channel 10 as an input channel in the operation of any channel
11:8	CHANY_SEL10	10th Conversion Select for Any Channel sequence 0000~1010: configure any channel from channel 0 to channel 10 as an input channel in the operation of any channel
7:4	CHANY_SEL9	9th Conversion Select for Any Channel sequence 0000~1010: configure any channel from channel 0 to channel 10 as an input channel in the operation of any channel
3:0	CHANY_SEL8	8th Conversion Select for Any Channel sequence 0000~1010: configure any channel from channel 0 to channel 10 as an input channel in the operation of any channel

Note: In the one-cycle or continuous scan modes, the ADC_CHANY1 shadow register is started by hardware. When ADC is working, only the shadow register is updated if the ADC_CHANY1 value is changed. Only when ADC starts to convert the last channel, the value of the shadow register will update to ADC_CHANY1, so as to implement the dynamic channel switch.

17.6.12 ADC_ANY_CFG Any Channel Configuration Register

Address offset: 0x64
Reset value: 0x00000000

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

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

Bit	Field	Description
31:4	Reserved	Reserved, must be kept at reset value
3: 0	CHANY_NUM	Number of Any Channel Mode: 0000: CHANY_SEL0 channel 0001: CHANY_SEL0~CHANY_SEL1 channels 0010: CHANY_SEL0~CHANY_SEL2 channels 0011: CHANY_SEL0~CHANY_SEL3 channels 0100: CHANY_SEL0~CHANY_SEL4 channels 0101: CHANY_SEL0~CHANY_SEL5 channels 0110: CHANY_SEL0~CHANY_SEL6 channels 0111: CHANY_SEL0~CHANY_SEL7 channels 1000: CHANY_SEL0~CHANY_SEL8 channels 1001: CHANY_SEL0~CHANY_SEL9 channels 1010: CHANY_SEL0~CHANY_SEL10 channels Others: invalid

Note: In the one-cycle or continuous scan modes, the ADC_NUM shadow register is started by hardware. When ADC is working, only the shadow register is updated if the ADC_ NUM value is changed. Only when ADC starts to convert the last channel, the value of the shadow register will update to ADC_ NUM, so as to implement the dynamic channel switch.

17.6.13 ADC_ANY_CR Any Channel Control Register

Address offset: 0x68
Reset value: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.														JTRG_EDGE	
														rw	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
JTRGSHIFT			JTRGSEL					JTRGEN	JADST	JAUTO	JEOSIE	JEOCIE	JEOSMPIE	JCEN	CHANY_MDEN
rw			rw					rw	rw	rw	rw	rw	rw	rw	rw

Bit	Field	Description
31:18	Reserved	Reserved, must be kept at reset value
17:16	JTRG_EDGE	External Trigger sources Edge Selection for Injected Channels 00: Double-edge trigger 01: Falling edge trigger 10: Rising edge trigger 11: Trigger blocked
15:13	JTRGSHIFT	External Trigger Shift Sample for Injected Channels After a trigger signal is generated, the sampling starts after a delay of N PCLK clock cycles. In case of trigger scan, the sampling in the rest channels starts immediately after the last sampling is complete. 000: no delay 001: 4 cycles 010: 16 cycles 011: 32 cycles 100: 64 cycles 101: 128 cycles 110: 256 cycles 111: 512 cycles

Bit	Field	Description
12:8	JTRGSEL	ADC External Trigger Sources Selection For Injected Channel 00000： TIM1_CC1 00001： TIM1_CC2 00010： TIM1_CC3 00011： TIM2_CC2 00101： TIM1_CC4 and TIM1_CC5 00111： EXTI 11 01000： TIM1_TRGO 01001： TIM13_CC1 01010： TIM14_CC1 01011： TIM2_CC1 01101： TIM2_TRGO 01111： EXTI 15 10000： TIM1_CC4 10001： TIM1_CC5 Others: invalid
7	JTRGEN	External Hardware Trigger Enable For Injected Channels 1: external trigger enabled for A/D conversion 0: external trigger disabled for A/D conversion
6	JADST	ADC Start for Injected Channels: enable JCEN first 1: Injected channel conversion start 0: Injected channel end of conversion or entering the idle state There are two ways of clearing JADST: JADST is automatically cleared by hardware after the conversion of injected channel is complete; JADST is cleared if JCEN is 0; it is also cleared after a system reset.
5	JAUTO	Automatic Injected Conversion 1: Automatic injected conversion enabled 0: Automatic injected conversion disabled
4	JEOSIE	Interrupt Enable for JEOS 1: end of A/D sequential conversion interrupt enabled 0: end of A/D sequential conversion interrupt disabled An A/D EOS interrupt request is generated if the JEOSIE bit is set.
3	JEOCIE	Interrupt Enable for JEOC 1: end of A/D conversion interrupt enabled 0: end of A/D conversion interrupt disabled An A/D EOC interrupt request is generated if the JEOCIF bit is set.
2	JEOSMPIE	Interrupt Enable for JEOSAMP 1: interrupt enabled for end of A/D sampling 0: interrupt disabled for end of A/D sampling An A/D end of conversion interrupt request is generated if the JEOSMPIF bit is set.
1	JCEN	Conversion Enable for Injected Channels 1: Injected conversion enabled 0: Injected conversion disabled
0	CHANY_ MDEN	Any Channel Mode Enable: 1: any channel mode enabled 0: any channel mode disabled

Note: In the any channel mode plus the one-cycle/continuous scan mode, wait until the ADC_ADCR.ADST and ADC_ADSTA.BUSY bits turn to 0 before closing the ADC. Then clear the ADC_ANY_CR.CHANY_MDEN bit.

17.6.14 ADC_SMPR1 Sample Configuration Register 1

Address offset: 0x70
Reset value: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SAMP7				SAMP6				SAMP5				SAMP4			
rw				rw				rw				rw			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SAMP3				SAMP2				SAMP1				SAMP0			
rw				rw				rw				rw			

Bit	Field	Description
31 : 0	SAMP7~SAMP0	Channel 0~7 Sample Time Selection These bits are used to independently select the sample time of each channel. In a sample cycle, the channel selection bit should remain unchanged. 0000: 2.5 cycles 0100: 42.5 cycles 0001: 8.5 cycles 0101: 56.5 cycles 0010: 14.5 cycles 0110: 72.5 cycles 0011: 29.5 cycles 0111: 240.5 cycles 1000: 3.5 cycles 1001: 4.5 cycles 1010: 5.5 cycles 1011: 6.5 cycles 1100: 7.5 cycles Others: reserved

17.6.15 ADC_SMPR2 Sample Configuration Register 2

Address offset: 0x74
Reset value: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.				SAMP10				SAMP9				SAMP8			
				rw				rw				rw			

Bit	Field	Description
31:12	Reserved	Reserved, must be kept at reset value
3: 0	SAMP8	Channel 8~10 Sample Time Selection These bits are used to independently select the sample time of each channel. In a sample cycle, the channel selection bit should remain unchanged. 0000: 2.5 cycles 0100: 42.5 cycles 0001: 8.5 cycles 0101: 56.5 cycles 0010: 14.5 cycles 0110: 72.5 cycles 0011: 29.5 cycles 0111: 240.5 cycles 1000: 3.5 cycles 1001: 4.5 cycles 1010: 5.5 cycles 1011: 6.5 cycles 1100: 7.5 cycles Others: reserved

17.6.16 ADC_JOFRn Injected Channel Data Offset Register (n=0~3)

Address offset: 0x7C~0x88
Reset value: 0x00000000

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

Res.	JOFFSET[11:0]
	rw

Bit	Field	Description
31:12	Reserved	Reserved, must be kept at reset value
11: 0	JOFFSET	Data Offset Register for 12-bit AD Injected Channel n The converted value from the injected channel n decreased by the JOFFSET offset value is then stored in the registers JADDATA and ADC_JDRn. Note: Writing to this bit is permitted when JADST=0.

17.6.17 ADC_JSQR Injected Sequence Register

Address offset: 0x8C
Reset value: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.										JNUM		JSQ3			
										rw		rw			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
JSQ3.	JSQ2					JSQ1				JSQ0					
rw	rw					rw				rw					

Bit	Field	Description
23:22	Reserved	Reserved, must be kept at reset value.
21:20	JNUM	Channel Number for Injected Sequence 00: JSQ0 channel 01: JSQ0~JSQ1 channels 10: JSQ0~JSQ2 channels 11: JSQ0~JSQ3 channels
19:15	JSQ3	3th Conversion for Injected Sequence 00000~01010: configure any channel from channel 0 to channel 10 as an injected channel Others: reserved
14:10	JSQ2	2th Conversion for Injected Sequence 00000~01010: configure any channel from channel 0 to channel 10 as an injected channel Others: reserved
9:5	JSQ1	1th Conversion for Injected Sequence 00000~01010: configure any channel from channel 0 to channel 10 as an injected channel Others: reserved
4:0	JSQ0	0th Conversion for Injected Sequence 00000~01010: configure any channel from channel 0 to channel 10 as an injected channel Others: reserved

Note: In the one-cycle or continuous scan modes, the ADC_JSQR shadow register is started by hardware. When ADC is working, only the shadow register is updated if the ADC_JSQR value is changed. Only when ADC starts to convert the last channel, the value of the shadow register will update to ADC_JSQR, so as to implement the dynamic channel switch.

17.6.18 ADC_JADDATA Injected Data Register

Address offset: 0x90
Reset value: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.									JVAILD	JOVERRUN	Res.	JCHANNELSEL			
									r	r		r			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
JDATA															

r

Bit	Field	Description
31:23	Reserved	Reserved, must be kept at reset value
22	JVALID	Valid Flag for Injected Channels (read-only) 1: JDATA[11:0] bits valid 0: JDATA[11:0] bits invalid This bit is set by hardware after the corresponding conversion is complete. It is cleared automatically after reading the ADC_JADDATA register.
21	JOVERRUN	Overrun Flag for Injected Channels (read-only) 1: JDATA [11:0] data is overrun 0: JDATA [11:0] for last conversion result Before the new conversion result is loaded to the register, JOVERRUN is set to '1' if the JDATA [11:0] is not read. This bit is cleared by hardware automatically after reading the ADC_JADDATA register.
20	Reserved	Reserved, must be kept at reset value.
19:16	JCHANNELSEL	4 bits show the injected channel corresponding to the current data (Injected Channel Selection) 0000 = conversion data of channel 0 0001 = conversion data of channel 1 0010 = conversion data of channel 2 0011 = conversion data of channel 3 0100 = conversion data of channel 4 0101 = conversion data of channel 5 0110 = conversion data of channel 6 0111 = conversion data of channel 7 1000 = conversion data of channel 8 1001 = conversion data of channel 9 1010 = conversion data of channel 10 (Conversion data of internal reference voltage or temperature Sensor) Others: invalid
15:0	JDATA	12-bit A/D conversion result of current injected channel (Transfer Data for Injected Channels) The alignment style is selected via software.

17.6.19 ADC_JDRn Injected Channel Data Register (n=0~3)

Address offset: 0xB0 ~ 0xBC
Reset value: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.									JVALID	JOVERRUN	Res.				
									r	r					
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
JDATA															
r															

Bit	Field	Description
31:23	Reserved	Reserved, must be kept at reset value
22	JVALID	Valid Flag for Injected Channels 1: JDATA[11:0] bits valid 0: JDATA[11:0] bits invalid This bit is set by hardware after the corresponding conversion is complete. It is cleared automatically after reading the ADC_JADR register.
21	JOVERRUN	Overrun Flag for Injected Channels 1: JDATA [11:0] data is overrun 0: JDATA [11:0] for last conversion result Before the new conversion result is loaded to the register, JOVERRUN is set to '1' if the JDATA [11:0] is not read. This bit is cleared by hardware automatically after reading the JDRn register.
20: 16	Reserved	Reserved, must be kept at reset value.
15: 0	JDATA	Conversion result of A/D injected channel n (Transfer Data for Injected Channels) The alignment style is selected via software. Please refer to the data alignment section.

17.6.20 ADC_LDATA Last Conversion Data Register

Address offset: 0xF0
Reset value: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.										LVAILD	LOVERRUN	LCHANNELSEL			
										r	r	r			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LDATA															
r															

Bit	Field	Description
31:22	Reserved	Reserved, must be kept at reset value.
21	LVALID	Data Valid Flag (read-only) 1: LCHANNELSEL and LDATA[11:0] bits valid 0: LCHANNELSEL and LDATA[11:0] bits invalid This bit is set by hardware after the corresponding conversion is completed. It is cleared automatically after reading the ADC_LDATA register.
20	LOVERRUN	Data Overrun Flag (read-only) 1: LDATA[11:0] The last conversion data is overrun 0: LDATA[11:0] The last conversion data is not overrun If the previous LDATA [11:0] was not read before data was loaded into LDATA [11:0] register, LOVERRUN will be set and this bit will be cleared automatically after reading ADC_LDATA register.
19:16	LCHANNELSEL	4 bits show the channel corresponding to the data of the last data register (Last Channel Selection) 0000 : Indicates that the channel of the last data register is channel 0 0001 : Indicates that the channel of the last data register is channel 1 0010 : Indicates that the channel of the last data register is channel 2 0011 : Indicates that the channel of the last data register is channel 3 0100 : Indicates that the channel of the last data register is channel 4 0101 : Indicates that the channel of the last data register is channel 5 0110 : Indicates that the channel of the last data register is channel 6 0111 : Indicates that the channel of the last data register is channel 7 1000 : Indicates that the channel of the last data register is channel 8 1001 : Indicates that the channel of the last data register is channel 9 1010 : Indicates that the channel of the last data register is channel 10
15:0	LDATA	12-bit A/D last channel conversion result (Last Conversion Data)

17.6.21 ADC_TRGSUPR External Rule Trigger Event Suppression Register

Address offset: 0xF4
Reset value: 0x00000000

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

Bit	Field	Description
31:8	Reserved	Reserved, must be kept at reset value

Bit	Field	Description
7	TRG_SUP_EN	Trigger suppression enable 0: Trigger suppression is disabled. When TRG_SUP_EN=0, the counter is triggered internally for clear flag. 1: Trigger suppression is enabled. No matter what value the TRG_SUP_CNT is set, the first rule trigger event triggers one ADC rule conversion regularly after TRG_SUP_EN is configured from 0 to 1
6:3	Reserved	Reserved, must be kept at reset value
2:0	TRG_SUP_CNT	Trigger suppression count target; each TRG_SUP_CNT+1 rule trigger event takes effect and conducts one ADC conversion only when TRG_SUP_EN=1. The internal count value will not be changed by suppressing TRG_SUP_CNT times to modify ADC trigger source or TRG_SUP_CNT value; only when TRG_SUP_EN=0, the internal count value will be forced for clear flag.

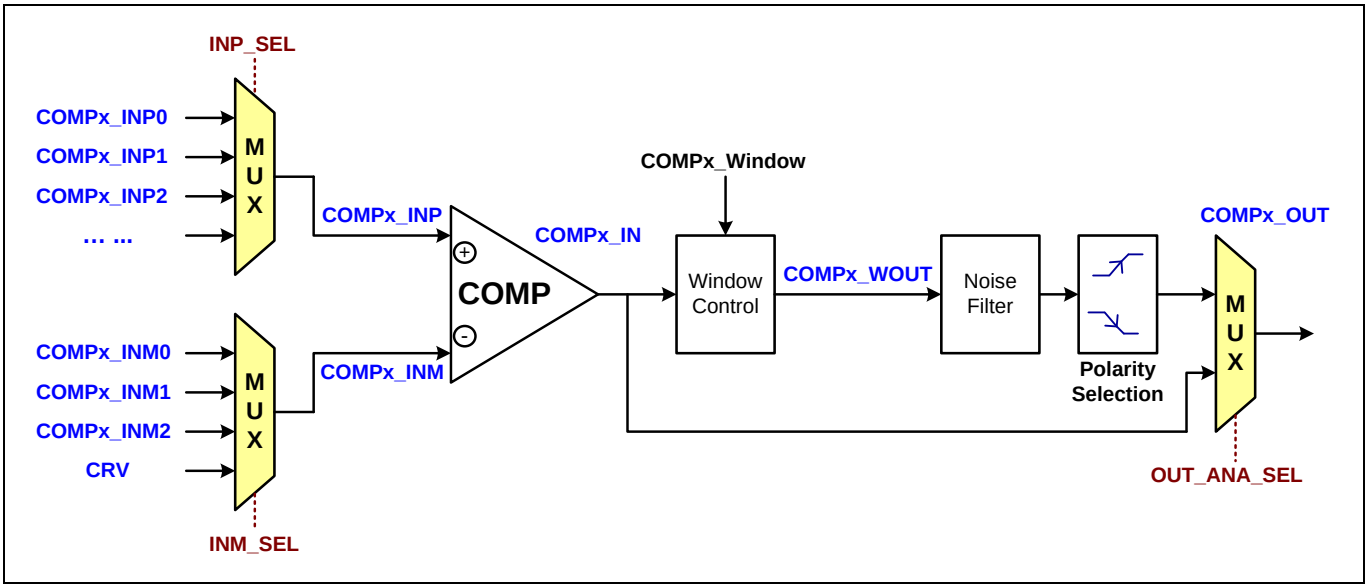
18 COMP Comparator

18.1 Overview

The chip embeds two general purpose comparators (COMP1, 2), that can be used independently. The comparator has an integrated digital filter, and its comparison result can be output to a timer or used to generate an interrupt, or a wake-up event triggering the low-power mode. It can also be used with a timer to combine with a PWM output from the timer, generating a cycle-by-cycle current control loop.

18.2 Functional block diagram

Figure 18-1 Comparator block diagrams



18.3 Main characteristics

- Analog input from alternate I/O pins;
- Programmable hysteresis voltage.
- Supports multiple types of rates and power consumption.
- Comparison result filtering is supported, and filtering cycle is configurable.
- Outputs to I/O pins or timers.
- Supports to wake up the CPU from Sleep and Stop modes through the external event EXTI.
- Each comparator supports 4 positive and 4 negative inputs with polling function:
- Supports the polling switch in fixed cycles.
- Supports the control of polling channel 1/2/3 or 1/2;
- Supports synchronized polling of positive and negative inputs or fixed negative inputs (optionally)

18.4 Functional description

18.4.1 COMP clocks and reset

The COMP input clock is synchronous with the APB1 CLK. Before using the comparator, the comparator clock should be enabled by setting the corresponding comparator clock enable bit in the RCC controller. Configuring the corresponding comparator reset control bit in the RCC controller allows you to reset the comparator via software.

18.4.2 COMP switch

Before using the comparator, the COMP is powered on by setting the EN bit in the COMPx_CSR register. The COMP is wakened up from the power-off state when EN bit is set to 1 and stops operation when EN bit is reset (EN set to 0).

18.4.3 COMP inputs and outputs

The I/Os used as comparator inputs must be configured in analog input mode in the GPIO registers. The COMP output could apply the filtering function (refer to the OFLT configuration in the COMPx_CSR register). The output can be internally redirected to a variety of timer input (refer to the OUT_SEL configuration in the COMPx_CSR register) or redirected to I/Os.

The control bit WE of COMPx_CSR can switch on or off the window control function. When the function is on and the signal 'COMPx_Window' is 1, COMPx_ IN will be sampled to COMPx_WOUT every one pclk. When the function is on and the signal 'COMPx_Window' is 0, COMPx_ WOUT will keep its present value (latest sampling value) unchanged. When the function is off, the analog signal "COMPx_IN" is directly output to the module "Noise Filter". Additionally, the signal "COMPx_Window" is a signal from timer selected by COMPx_CSR->WSEL. This function is used to filter the input when the input voltage of the comparator is invalid. It can also realize zero crossing detection in some PWM applications and filter the transient current pulse signal in motor applications. In practical application, COMPx_ WOUT may delay the analog input by up to one bus clock pclk. For some characteristic applications, the position of the window must be carefully considered, and the window width must be at least greater than one pclk clock cycle to ensure that the signal can be sampled.

18.4.4 COMP channel selection

Each COMP has 4 positive input channels to be selected from four external pins, and has 4 negative input channels to be selected from three external pins and the internal pin connected the CRV output. The CRV voltage can be the partial voltage value of the internal 1.2V reference voltage (V_{REFINT}) or the external voltage (VDDA).

In normal operation mode, the input channel of COMP is selected by software. In polling operation mode, comparison results of multiple channels are monitored in a time-shared manner via software polling. Logically, it is similar to simultaneous operation of several comparators.

In normal operation mode, the COMP compares selected signals from INP and INM ports. The specific procedures are as follows:

Set the INP_SEL bit and the INM_SEL bit in the COMPx_CSR register to select desired signals;

By setting the EN bit in the COMPx_CSR register, the COMP is powered-on and starts to work.

The comparison results are stored in the OUT bit of the COMPx_CSR register.

Moreover, it is required to set the CRV_SEL bit in the COMP_CRV register and then set the CRV_EN bit when the INM_SEL of COMP selects CRV (performed before the step 2 above).

In polling operation mode, the signals from the INP port in the COMP polls periodically. The FIXN bit in the COMPx_POLL register can be used to configure whether the signals from INM port follow the INP changes. The INM_SEL bit in the COMPx_CSR can also use to configure them. Please note that the INP_SEL bit in the COMPx_CSR will be invalid when the polling function is enabled. Similarly, if

the FIXN bit in the COMPx_POLL register determines the INM port follow the INP polling changes, then the INM_SEL bit in the COMPx_CSR will be invalid too. The specific procedures are as follows:

- Set the PERIOD bit in the COMPx_POLL register to select the desired polling wait cycles;
- Set the FIXN bit in the COMPx_POLL register to determine whether the signals from INM port follow the INP polling changes.
- Set the POLL_CH bit in the COMPx_POLL register to determine whether the desired polling channel is 1/2/3 or 1/2;
- Set the POLL_EN bit in the COMPx_POLL register to enable the polling function;
- By setting the EN bit in the COMPx_CSR register, the COMP is powered-on and starts to work.
- The polling comparison results are stored in the POUT bit of the COMPx_POLL register, where the results of polling channel 3/2/1 are individually stored in the POUT [2], POUT [1], and POUT [0].

18.4.5 Interrupt and wakeup

The comparator outputs can be connected to the events controller via hardware inside the chip. Each comparator has its own EXTI line and can generate events to exit the low-power mode. Refer to Interrupt and events section in the reference manual for more details.

18.4.6 Power mode

The comparator power consumption versus propagation delay can be adjusted to have the optimum trade-off for a given application.

The bit MODE in COMPx_CSR register can be programmed in 4 ways as follows:

- 00: High speed/full power;
- 01: Medium speed/medium power;
- 10: Low speed/low-power;
- 11: Very-low speed/ultra-low-power.

18.4.7 Comparator LOCK mechanism

The comparators can be used for safety purposes, such as over-current or thermal protection. For some specific applications, it is necessary to ensure that the comparator programming cannot be altered in case of spurious modification access or program counter corruption.

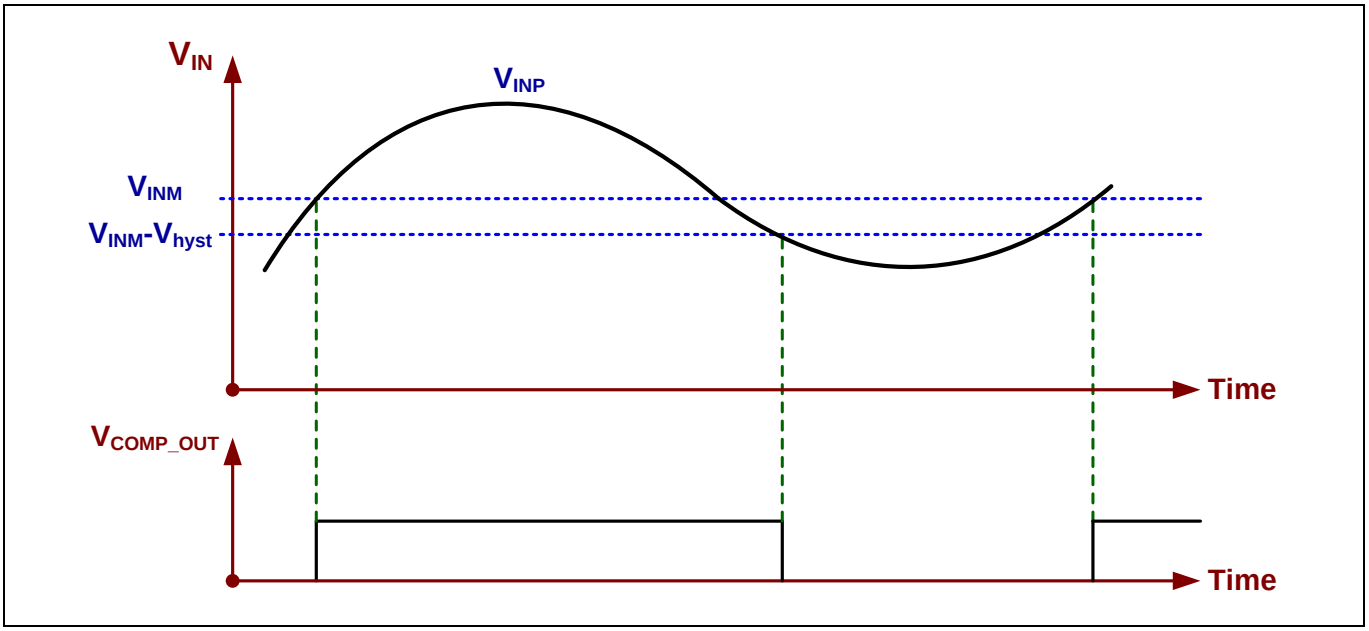
For this purpose, the comparator control status registers can be read-only.

Once the programming is completed, the LOCK bit is set to 1. This causes the whole COMPx_CSR register to become read-only, including the LOCK bit, which can only be reset by a MCU reset. (For details, please refer to the LOCK configuration in the COMPx_CSR register.)

18.4.8 Hysteresis voltage

To avoid invalid input in case of noisy signals, the comparator supports configurable hysteresis level (for details, please refer to the HYST configuration in the COMPx_CSR register).

Figure 18-2 Comparator hysteresis



18.5 Register

Table 18-1 Overview of COMP registers

Offset	Acronym	Register Name	Reset
0x0,0x8	COMPx_CSR (x=1, 2)	COMP x (x=1, 2) control and status register	0x0000000
0x40	COMP_CRV	COMP external reference voltage register	0x0000000
0x4, 0xC	COMPx_POLL (x=1, 2)	COMP x (x=1, 2) polling register	0x0000000

18.5.1 COMPx_CSR (COMP control and status register) (x=1, 2)

Address offset: 0x0, 0x8
Reset value: 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LOCK	OUT	OUT_ANA_SEL	WE	WSEL			Res.				OFLT			HYST	
rw	r	rw	rw	rw							rw			rw	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
POL	Res.	OUT_SEL				Res.	INP_SEL		INM_SEL		MODE			Res.	EN
rw		rw					rw		rw		rw				rw

Bit	Field	Description
31	LOCK	Comparator Register Lock This bit is write-once. It is set by software. It can be cleared by a system reset. Once set, it allows to have all control bits of comparator x as read-only. 1: COMPx_CSR, COMPx_POLL is read-only 0: COMPx_CSR, COMPx_POLL is read-write
30	OUT	Comparator x Output This bit indicates the comparator x output state. 1: Output is high (Positive input is greater than negative input in voltage) 0: Output is low (Positive input is less than negative input in voltage)

Bit	Field	Description
29	OUT_ANA_SEL	Comparator x output source selection 1: Select the analog output signal 0: Select the analog output signal that has been synchronized Note: When OUT_ANA_SEL=1, compared result can wake up CPU in low power mode as well as no clk by exit.
28	WE	Comparator x window control enable This bit enables window control function 1: window control function enabled 0: window control function disabled
27:25	WSEL	Comparator x Window input source selection 000:：TIM1_OC1REF 001：TIM1_TRGO 010：TIM2_TRGO 100:：TIM13_OC1REF 101:：TIM14_OC1REF Other: No selection
24 : 21	Reserved	Always read as 0.
20 : 18	OFLT	Comparator x Output Filter Period These bits control the comparator x output filter period. When the comparator output signal is shorter than the filter period width, it is considered as invalid to be filtered out, otherwise it is considered as valid. n indicates the number of Polling Wait Cycle configured with COMPx_CSR -> PERIOD. 111: 128 * n clock cycles 110: 64 * n clock cycles 101: 32 * n clock cycles 100: 16 * n clock cycles 011: 8 * n clock cycles 010: 4 * n clock cycles 001: 2 * n clock cycles 000: no filter
17 : 16	HYST	Comparator x Hysteresis These bits control the comparator x hysteresis level. When MODE = 00， 11: 85mV　10: 45mV 01: 22mV　00: 0mV When MODE = 01、10 或 11 时 11: 60mV　10: 32mV 01: 15mV　00: 0mV
15	POL	Comparator x Output Polarity This bit is used to select the comparator x output polarity. 1: Output is inverted from the terminal 0: Output is not inverted from the terminal
14	Reserved	Always read as 0.
13 : 10	OUT_SEL	Comparator x Output Selection These bits select the destination of the comparator x output. 0010: Timer 1 brake input 0110: Timer 1 Ocrefclear input 1000: Timer 2 input capture 4 1001: Timer 2 Ocrefclear input 1100: Timer 2 input capture 1 1101: Timer 13 input capture 1 1110: Timer 14 input capture 1 Other: No selection
9	Reserved	Always read as 0.
8: 7	INP_SEL	Comparator x Positive Input Selection These bits allows to select the source connected to the positive input end of the comparator x. Comparator x(x=1): 00: COMPx_INP0(PA3), corresponding to the Input Positive Channel 0; 01: COMPx_INP1(PA5), corresponding to the Input Positive Channel 1; 10: COMPx_INP2(PA6), corresponding to the Input Positive Channel 2; 11: COMPx_INP3(PB0), corresponding to the Input Positive Channel 3; Comparator x(x=2)： 00: COMPx_INP0(PB0), corresponding to the Input Positive Channel 0; 01: COMPx_INP1(PB2), corresponding to the Input Positive Channel 1; 10: COMPx_INP2(PA8), corresponding to the Input Positive Channel 2; 11: COMPx_INP3(PA9), corresponding to the Input Positive Channel 3;

Bit	Field	Description
6: 4	INM_SEL	Comparator x Negative Input Selection These bits allows to select the source connected to the negative input end of the comparator x. Comparator x(x=1): 000：COMPx_INM0（PA4）, corresponding to the Input Negative Channel 0; 001：COMPx_INM1（PA7）, corresponding to the Input Negative Channel 1; 010：COMPx_INM2（PA8）, corresponding to the Input Negative Channel 2; 011：COMPx_INM3（CRV）, corresponding to the Input Negative Channel 3; Comparator x(x=2): 000：COMPx_INM0（PA4）, corresponding to the Input Negative Channel 0; 001：COMPx_INM1（PA7）, corresponding to the Input Negative Channel 1; 010：COMPx_INM2（PA1）, corresponding to the Input Negative Channel 2; 011：COMPx_INM3（CRV）, corresponding to the Input Negative Channel 3; Others: no choice
3: 2	MODE	Comparator x Mode Select These bits select the power consumption mode of the comparator x. 11: Very-low speed/ultra-low power 10: Low speed/low-power 01: Medium speed/medium power 00: High speed/full power
1	Reserved	Always read as 0.
0	EN	Comparator x Enable This bit switches ON/OFF the comparator. 1: Comparator x enabled 0: Comparator x disabled

18.5.2 COMP_CRV (COMP external reference voltage register)

Address offset: 0x40
Reset value: 0x00000000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CRV_EN	Res.		CRV_SRC	Res.				CRV_SEL							
rw			rw					rw							

Bit	Field	Description
31:16	Reserved	Reserved and always read as 0
15	CRV_EN	Comparator Reference Voltage Enable 1: COMP external reference voltage (CRV) enabled 0: COMP external reference voltage (CRV) disabled
14:13	Reserved	Reserved and always read as 0
12	CRV_SRC	Comparator Reference Voltage Source Select 0: VREFINT (internal voltage) 1: VDDA (external voltage) Note: If you want to select internal voltage as CRV source, it is necessary to enable ADC voltage sensor by Configuring ADC_ADCFG.
11:8	Reserved	Reserved and always read as 0
7: 0	CRV_SEL	Comparator Reference Voltage Select V_{CRV} = VDDA*CRV_SEL/255 or V_{REFINT} *CRV_SEL/255

18.5.3 COMPx_POLL (COMP polling register) (x=1, 2)

Address offset: 0x4, 0xC
Reset value: 0x00000000

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

Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.					POUT			Res.	PERIOD			Res.	FIXN	POLL_CH	POLL_EN
					r				rw				rw	rw	rw

Bit	Field	Description
31:11	Reserved	Reserved and always read as 0
10 : 8	POUT	Polling Output This bit is read-only. It reflects the polling channel output state. POUT [0] corresponds to channel 1. POUT [1] corresponds to channel 2. POUT [2] corresponds to channel 3. 1: Output is high (positive input above negative input) 0: Output is low (positive input below negative input)
7	Reserved	Always read as 0.
6: 4	PERIOD	Polling Wait Cycle Switch to the next polling channel every n PCLK cycles. 111: 128 clock cycles 110: 64 clock cycles 101: 32 clock cycles 100: 16 clock cycles 011: 8 clock cycles 010: 4 clock cycles 001: 2 clock cycles 000: 1 clock cycles
3	Reserved	Always read as 0.
2	FIXN	Polling Negative Input Fix 1: Polling channel negative input is fixed. It is determined by the INM_SEL in the CSR register. When FIXN is 1, the INP_SEL is invalid. 0: Polling channel negative input is not fixed. It changes simultaneously with the INP channel. When FIXN is 0, the INM_SEL and INP_SEL is invalid.
1	POLL_CH	Comparator Polling Channel 1: polling channel 1/2/3 0: polling channel 1/2
0	POLL_EN	Comparator Polling Enable 1: Comparator polling enabled. 0: Comparator polling disabled.

19 OPAMP Operational Amplifier1/2

19.1 Overview

The chip embeds two operational amplifiers, whose inputs and outputs are connected to I/O. OPAMP can be connected to ADC and COMP by sharing I/O.

19.2 Main characteristics

- Rail to rail input and output
- Output connected to I/O

19.3 Functional description

19.3.1 Clock

The OPAMP clock controller is synchronous with PCLK1.

19.4 Register

19.4.1 Overview of registers

Table 19-1 Overview of OPAMP register

Offset	Acronym	Register Name	Reset
0x10	OPAMP_CR	Configuration register	0x0000002

19.4.2 OPAMP_CR Configuration register

Address offset: 0x00
Reset value: 0x00000002

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

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

Bit	Field	Description
31:1	Reserved	Reserved, must be kept at reset value
0	OPAEN	Enable OPA 0: Disable 1: Enable

20 DBG Debug Support

20.1 Introduction

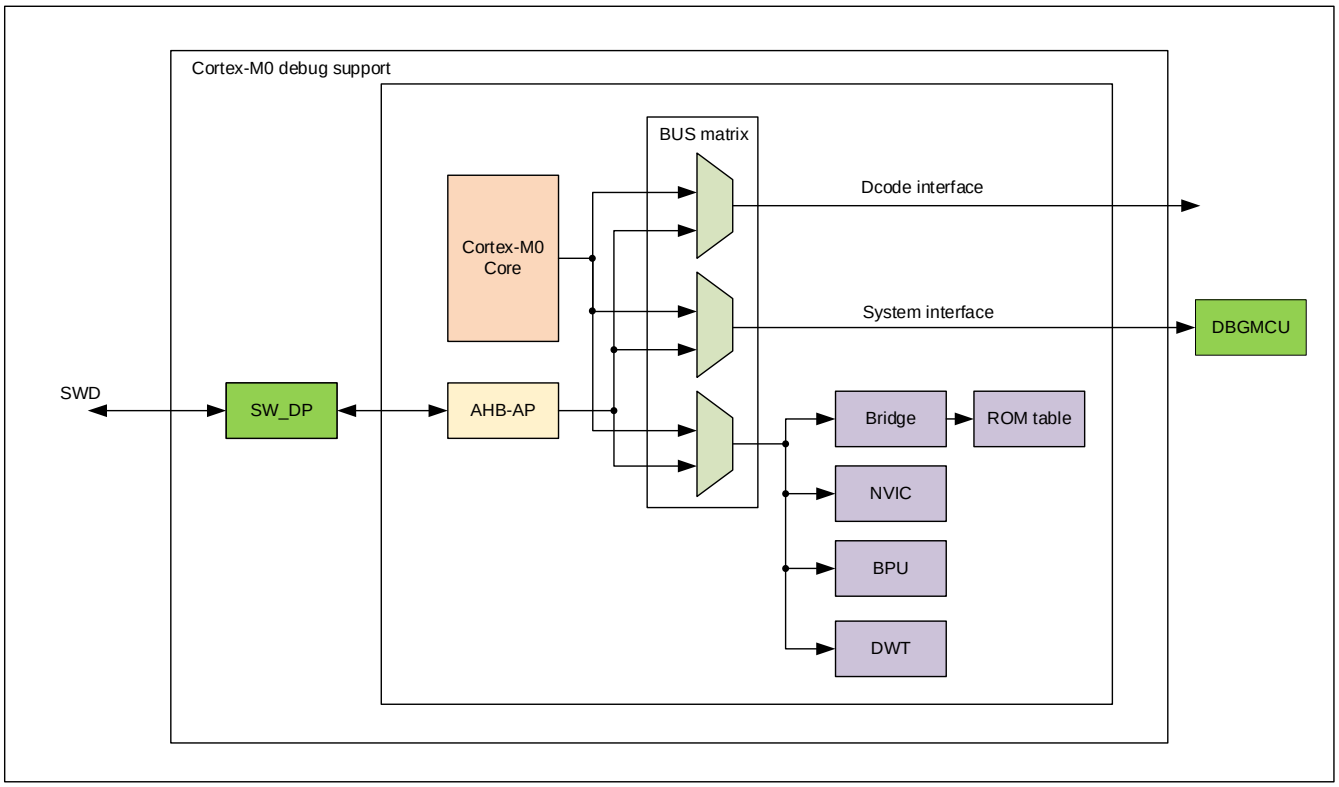
The MCU core includes the debug module mainly used for the function commissioning. When the core gets address (address breakpoint) or accesses data (data breakpoint), the hardware debug module may control the core to stop. The user can enquire the core internal status and system status. After enquiry, the core may continue to execute the current program.

When the chip and debugger connect and begin debugging, the debugger auto calls the core debug module for debug operation.

20.2 Function descriptions

20.2.1 Function block diagram

Figure 20-1 Debug function block diagram



Cortex-M0 core includes the debug unit, and it's composed of:

SWDP: SW debug port
BPU: Breakpoint debug unit
DWT: Data watch point and trace

20.2.2 SWD internal pull up and down

SWD pin input directly controls the debug mode, and cannot be floated. To guarantee that I/O level is controllable, SWD pin has the built in pull up and down resistance.

- SWDIO: Internal pull up
- SWCLK: Internal pull down

The software can use these I/O ports as the ordinary I/O port. At this time, the pull up/down function closes by default. Refer to the chapter of General Purpose Input/output GPIO.

20.2.3 SWJ debug port

The chip's two ordinary I/O ports are used as SWD-DP interface pin. These pins in different packages all support SWD debug port.

Table 20-1 SWD debug port pin

SWJ-DP port pin name	SW debug interface		Pin allocation
	Type	Debug function	
SWDIO	Input/output	Serial data input/output	PA13
SWCLK	Input	Serial clock	PA14

20.3 ID code and lock mechanism

The chip has many ID codes, as seen below:

Table 20-2 ID code

ID Name	Chip
DEV_ID	0x4C513000
CPU TAP SW ID	0x0BB11477

20.3.1 Microcontroller device ID code

The microcontroller contains the device ID code. This ID defines MCU chip version, and is mapped on the external APB bus. This code can be acquired by the user code and debug interface.

20.3.2 Cortex JEDEC-106 ID code

The microcontroller has a JEDEC-106ID code. It’s located in the 4KB ROM table mapped to the internal PBB bus address 0xE00FF000_0xE00FFFFF

.

20.4 SW debug port

20.4.1 SW protocol introduction

This sync serial protocol uses 2 pins: Clock signal (SWCLK) and two-way data signal (SWDIO) from host to target.

As the two-way data line, SWDIO must be connected to the pull up resistance (ARM's recommended value 100K). SWDIO pin has the built in pull up resistor, and the extra external resistor is not required.

Data transmission begins from the low bit, and allow reading and writing the register DPACC and APACC.

According to the protocol, when SWDIO changes direction, insert one conversion time (one Bit time by default, adjustment by SWCLK). In this period, any device cannot drive the signal line.

20.4.2 SW protocol series

One series contains three phases:

- The host sends 8 bit request packet;
- The target sends 3 bit acknowledgement response;
- According to the configuration direction, the host or target sends 33 bit (including one check bit) data;

Table 20-3 8bit request packet

Bit	Name	Description
0	Beginning	Must be 1
1	APnDP	0: Access DP 1: Access AP
2	RnW	0: Write request 1: Read request
4: 3	A[3 : 2]	DP or AP register address
5	Parity	Previous bit check bit
6	Stop	0
7	Park	Cannot be driven by host. Because of pull up, the target is always read as 1

Note: Each request packet is followed by 1 bit conversion time. For more information on DAPCC and APACC register, refer to the relevant CPU technology data sheet of ARM.

Table 20-4 3bit response package

Bit	Name	Description
0 .. 2	ACK	001: Fail 010: Wait 100: Success

Note When the response (ACK) signal is one of scenarios above, the response bit is followed by one conversion time.

Table 20-5 33bit data packet

Bit	Name	Description
0 .. 31	WDATA/RDATA	Read or write data
32	Parity	32 bit data parity check bit

Note: Wait for one conversion time after read data bit.

20.4.3 SW-DP status unit (Reset, Idle states, ID code)

SW-DP status unit uses the internal ID code to identify SW_DP. Observe JEP-106 standard. For specific information, please refer to the relevant ARM manual.

SW-DP status unit doesn't work until before the debugger reads ID.

- In case of power on reset, or DP switches from JTAG to SWD, or exceeds high level of 50 periods, SW-DP status unit is in the reset status;
- If the low level of at least 2 periods appear after RESET status, the status unit will switch to the IDLE status;
- When the status unit is in the reset status, first switch to the IDLE status, and read DP-SW ID register. Otherwise, the debugger cannot conduct other normal transmission, and ACK Fault will appear;

20.4.4 DP and AP read/write access

- DP read operation has no delay: The debugger directly acquires data (in case of ACK return success status) or waits (in case of ACK return wait status);
- AP read operation has latency. It means that the previous read operation result can only be acquired during the next operation. If the next operation is not access to AP, it's mandatory to read DP-RDBUFF register to acquire the previous read operation result;

- DP-CTRL/STAT register READOK flag bit is refreshed after AP read operation and RDBUFF read operation to inform the debugger whether AP read operation succeeds;
- SW-DP has the write buffer (DP and AP have the write buffer) so that the write operation can be accepted when other transmissions are ongoing. If the write buffer is full, the debugger will acquire one wait ACK response. Read IDCODE register, read CTRL/STAT register and write ABORT register operation remain accepted when the write buffer is full;
- Because SWCLK and HCLK are not in step, insert two extra SWCLK periods after the write operation (after parity check bit) to ensure that the internal write operation is correctly completed. These two extra clock periods should be inserted when the line is in IDLE status. This operation step is particularly important during writing CTRL/STAT register to propose one power on request. Otherwise, the next operation (valid operation after core power on) will execute immediately. It will lead to failure;

20.4.5 SW-DP register

In case of APnDP=0, access to these registers below.

Table 20-6 SW-DP register

A[3:2]	Read/write	SELECT register's CTRLSEL bit	Register	Description
00	Write		IDCODE	Fixed as 0x0BB1 1477（for identifying SW-DP）
00	Write		ABORT	
01	Read/Write	0	DP-CTRL/STAT	Request one system or debug power on operation; configure AP access operation mode; Control compare, check operation; Read some status bits (overflow, power on response).
01	Read/Write	1	WIRE CONTROL	Configure the serial communication physical layer protocol (such as conversion time length, etc.)
10	Read		READ RESEND	Allow to restore data from one error debug transmission rather than repeat the initial AP transmission.
10	Write		SELECT	Select current access port and valid 4 word register window.
11	Read/Write		READ BUFFER	This register will capture the data result of the previous read operation from AP. Therefore, the data can be acquired without again enabling the new AP transmission

20.4.6 SW-AP register

In case of APnDP=1, access and AP register access address is composed of two parts:

- A[3：2] value
- DP SELECT register's current value

20.5 MCU debug module (DBGMCU)

MCU debug module provides the debugger assist function below:

- Support low power mode
- Breakpoint timer and watchdog clock control

20.5.1 Debug support at low power mode

MCU has multiple low power modes. It can close CPU clock, reduce CPU power consumption, and enter the low power mode by executing WFE or WFI command. FCLK and AHB bus clock HCLK are mandatory for debug operation, and cannot be closed. MCU is equipped with some registers to change the low power mode characteristics, thus supporting the debug code in the low power mode. Specific configurations:

- Enter the sleep mode. In order to make HCLK and FCLK have the same clock, the debugger must set the DBGMUC_CR register DBG_SLEEP bit.

20.5.2 Support timer, watchdog

In case of the breakpoint, select the timer's work mode according to the application of timer and watchdog;

- Counter count continues. The application is in the PWM wave control motor
- Counter count stops. The application is in the watchdog counter.

20.6 Register

20.6.1 Register overview

Table 20-7 DBG register overview

Offset	Acronym	Register Name	Reset
0x00	DBG_IDCODE	DBG ID encode register	0x4C513000
0x04	DBG_CR	DBG control register	0x00000000

20.6.2 DBG_IDCODE ID Encode Register

Address offset: 0x40013400 (only support 32-bit access, read only)
复位值：0x4C513000

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

Bit	Field	Description
31:0	DEV_ID	Device Identifier Read only register, always read as reset value

20.6.3 DBG_CR Control Register

Address offset: 0x40013404 (only support 32 bit access)
Reset value: 0x0000 0000(POR reset only, not reset by system reset)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.									DBG_TIM6_STOP	Res.		DBG_TIM13_STOP	DBG_TIM14_STOP	Res.	

									rw			rw	rw		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	DBG_ TIM2_ PWM_OFF	DBG_TIM1_ PWM_OFF	Res.	DBG_TIM2_STOP	DBG_TIM1_STOP	Res.	DBG_IWDG_STOP	Res.							DBG_SLEEP
	rw	rw		rw	rw		rw								rw

Bit	Field	Description
31:23	Reserved	Reserved, must retain the reset value
22	DBG_TIM6_STOP	TIM6 stops work when the core enters the debug mode 0: TIM6 still works normally when the timer is selected 1: TIM6 stops when the timer is selected
21:20	Reserved	Reserved, must retain the reset value
19	DBG_TIM13_STOP	TIM13 stops work when the core enters the debug mode 0: TIM13 still works normally when the timer is selected 1: TIM13 stops when the timer is selected
18	DBG_TIM14_STOP	TIM14 stops work when the core enters the debug mode 0: TIM14 still works normally when the timer is selected 1: TIM14 stops when the timer is selected
17:15	Reserved	Reserved, must retain the reset value
14	DBG_ TIM2_ PWM_OFF	TIM2 PWM outputs 0 when the core enters the debug mode 0: TIM2 PWM normal outputs 1: TIM2 PWM outputs 0
13	DBG_TIM1_ PWM_OFF	TIM1 PWM outputs 0 when the core enters the debug mode 0: TIM1 PWM normal outputs 1: TIM1 PWM outputs 0
12	Reserved	Reserved, must retain the reset value
11	DBG_TIM2_STOP	TIM2 stops work when the core enters the debug mode 0: TIM2 still works normally when the timer is selected 1: TIM2 stops when the timer is selected
10	DBG_TIM1_STOP	TIM1 stops work when the core enters the debug mode 0: TIM1 still works normally when the timer is selected 1: TIM1 stops when the timer is selected
9	Reserved	Reserved, must retain the reset value
8	DBG_IWDG_STOP	Independent watchdog stops work This bit is not related to whether the core enters the debug status 0: The watchdog counter still works normally 1: The watchdog counter stops work
7:1	Reserved	Reserved, must retain the reset value
0	DBG_SLEEP	Debug sleep mode 0: In the sleep mode, clock FCLK opens, FCLK keeps the configured system clock by default, and HCLK closes. The sleep mode won't reset the configured clock system. When exiting the sleep mode, the software doesn't need to reconfigure the system clock 1: In the sleep mode, FCLK and HCLK clock is provided by the formally configured system.

21 Device Electronic Signature

21.1 Overview

Device electronic signature is an identifier (96bits) that is stored in the Flash memory system storage area to solely identify a microcontroller. Under no circumstances can the user modify the device electronic signature.

Device electronic signature can be read by software and used to achieve the following functions:

- Used as a password to improve the security of code in flash memory by using device electronic signatures in combination with software encryption and decryption algorithms when programming flash memory
- Used as serial number in terminal applications
- Activating the bootloader with safety mechanism

21.2 Register description

Base address: 0x1FFF F7E8

Table 21-1 Device electronic signature register overview

Offset	Acronym	Register Name	Reset
0x00	UID1	Unique identifier	0xXXXXXXXX
0x04	UID2	Unique identifier	0xXXXXXXXX
0x08	UID3	Unique identifier	0xXXXXXXXX

21.2.1 UID1 Unique Identifier

Address offset：0x00

Reset value: This value is written by default.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
U_ID（31：16）															
R															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
U_ID（15：0）															
r															

Bit	Field	Description
31：0	U_ID（31：0）	U_ID：31：0 unique ID bits

21.2.2 UID2 Unique Identifier

Address offset：0x04
Reset value: This value is written by default.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
U_ID（63：48）															
R															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
U_ID（47：32）															
R															

Bit	Field	Description
31：0	U_ID（63：32）	U_ID：63：32 unique ID bits

21.2.3 UID3 Unique Identifier

Address offset：0x08
Reset value: This value is written by default.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
U_ID（95：80）															
r															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
U_ID（79：64）															
r															

Bit	Field	Description
31：0	U_ID（95：64）	U_ID：95：64 unique ID bits

22 History Records

Table 22-1 History records

Date	Version	Content
2025/03/03	1.00	Initial Version