CH32X315/X305 Reference Manual

V1.2

https://wch-ic.com

Overview

CH32X315 is a multi-channel ADC microcontroller designed based on the QingKe RISC-V core. The core supports 480MHz zero-wait operation and supports single-precision floating point instructions. CH32X315 has built-in 4 sets of high-speed 12-bit ADC conversion units, providing 48 direct input channels, supporting scan mode, and can be used with analog switch chips to expand the number of channels to 8 times and automatically switch; Features a built-in USB3.0 SuperSpeed controller and PHY, USB2.0 High-speed controller and PHY, and Type-C/PD controller and PHY, supporting USB3.2 Gen 1, USBSS Device functionality, USBHS Host and USBHS Device functionality, Type-C, and PDUSB fast charging; It provides a wealth of peripheral resources, including a DMA controller, ARGB single-wire RGB driver, multiple timer groups, 4 USART serial ports, 2 I2C interfaces, and 3 SPI interfaces.

The CH32X305 omits the USB3.0 (USBSS) module found in the CH32X315.

This manual provides detailed usage information of CH32X315 and X305 chips for users' application development.

For the device characteristics of CH32X315 and X305 series chips, please refer to the *CH32X315DS0*.

RISC-V core version comparison overview

Features / Core versions	Instruction set	Hardware stack levels	Interrupt nesting levels	Fast interrupt channels	Assembly line	Vector table model	Extended instruction (XW)	Memory protection
V3F	RV32IMAFCB	2	2	4	3	Address or command	Supported	1/2-wire

For information about the RISC-V core, please refer to the QingKeV3 microprocessor manual: *QingKeV3_Processor_Manual*

Abbreviated description of the bit attribute in the register:

Register bit attributes	Property description
RF	Read-only attribute, read a fixed value.
RO	Read-only attribute, changed by hardware.
RZ	Read-only attribute, auto bit clear 0 after read operation.
WO	Write only attribute (not readable, read value uncertain)
WA	Write-only attribute, writable in Safe mode.
WZ	Write only attribute, auto bit clear 0 after write operation.
RW	Readable and writable.
RWA	Readable, writable in Safe mode.
RW1	Readable, write 1 valid, write 0 invalid.
RW0	Readable, write 0 valid, write 1 invalid.
RW1T	Readable, write 0 invalid, write 1 flipped.
RW1Z	Readable; write 1 clears this bit; write 0 invalid.
SC	Auto cleared.

Chapter 1 Memory and Bus Architecture

1.1 Bus Architecture

This series of chips is an industrial-grade general-purpose microcontroller designed based on the QingKe RISC-V3F core. The core, arbitration unit, DMA module, SRAM storage and other parts of its architecture interact through multiple sets of buses. The system block diagram is shown in Figure 1-1-1 and Figure 1-1-2.

Figure 1-1-1 CH32X315 system block diagram

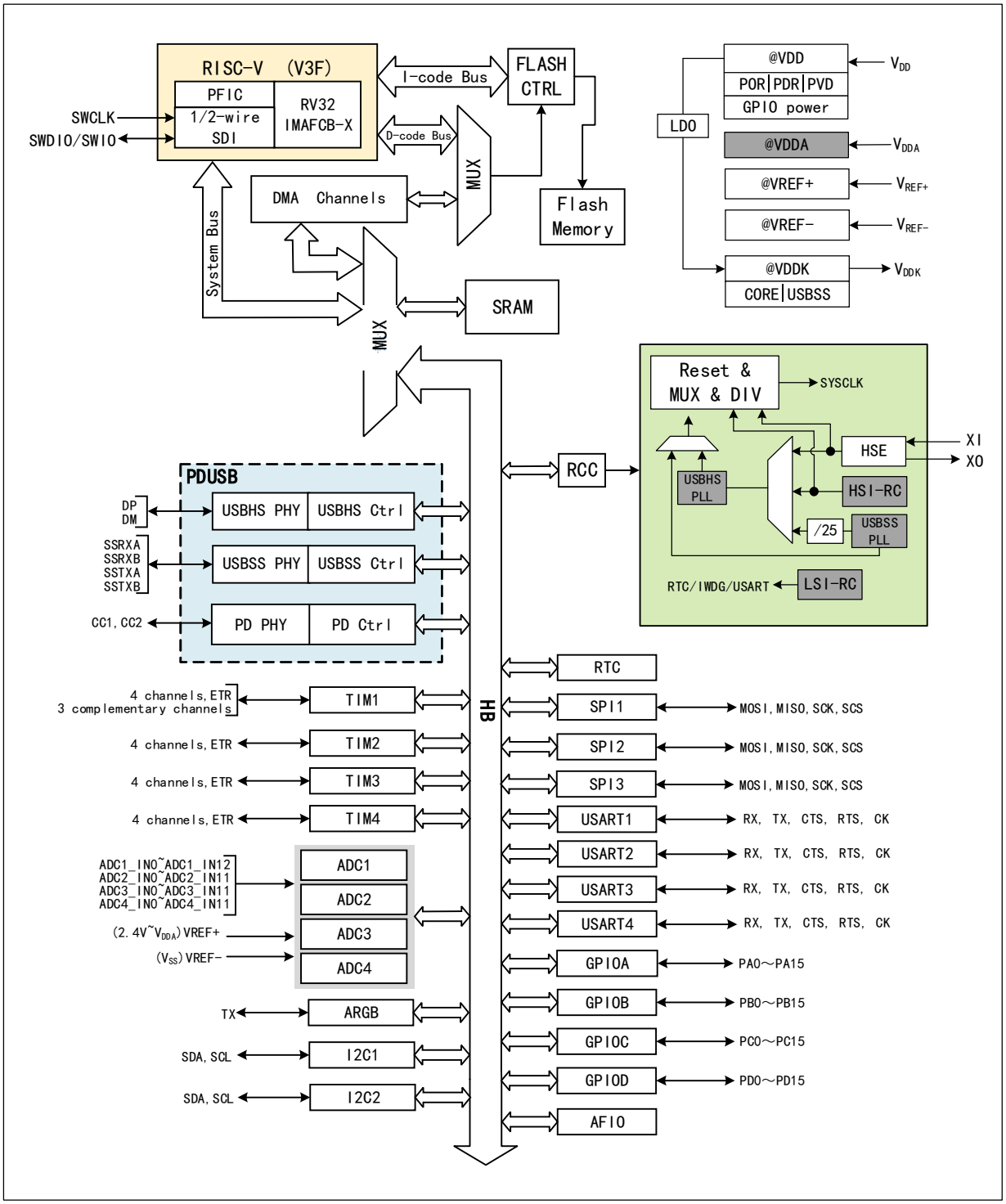
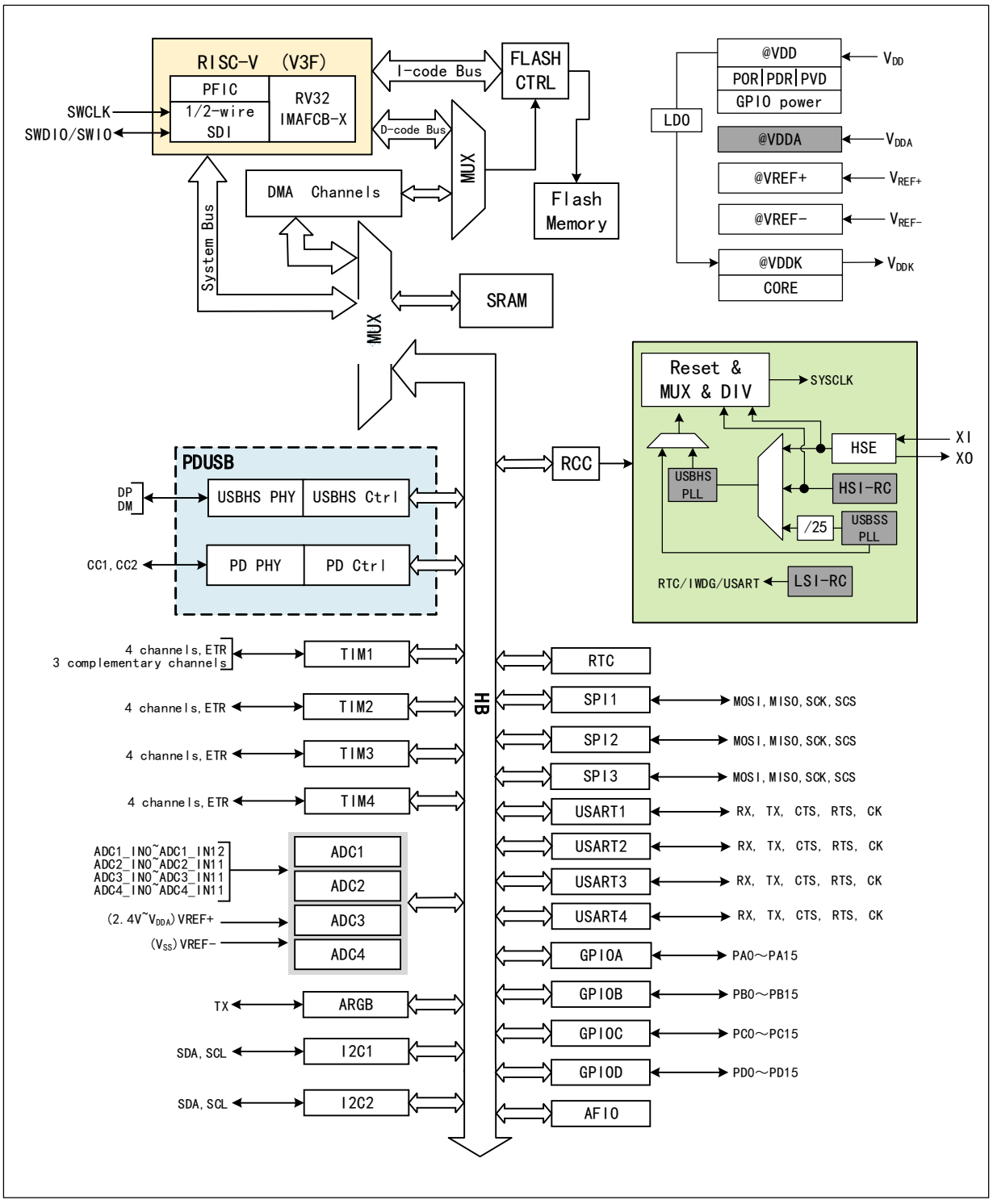


Figure 1-1-2 CH32X305 System block diagram



The system is equipped with: General-purpose DMA controller to reduce the CPU burden and improve efficiency; clock tree hierarchy management to reduce the total power consumption of peripherals, as well as data protection mechanisms, clock security system protection mechanisms and other measures to increase system stability.

- The instruction bus (I-Code) connects the core and the FLASH instruction interface, and prefetching instructions are completed on this bus.
- The data bus (D-Code) connects the kernel to the FLASH data interface and is used for constant loading and

debugging.

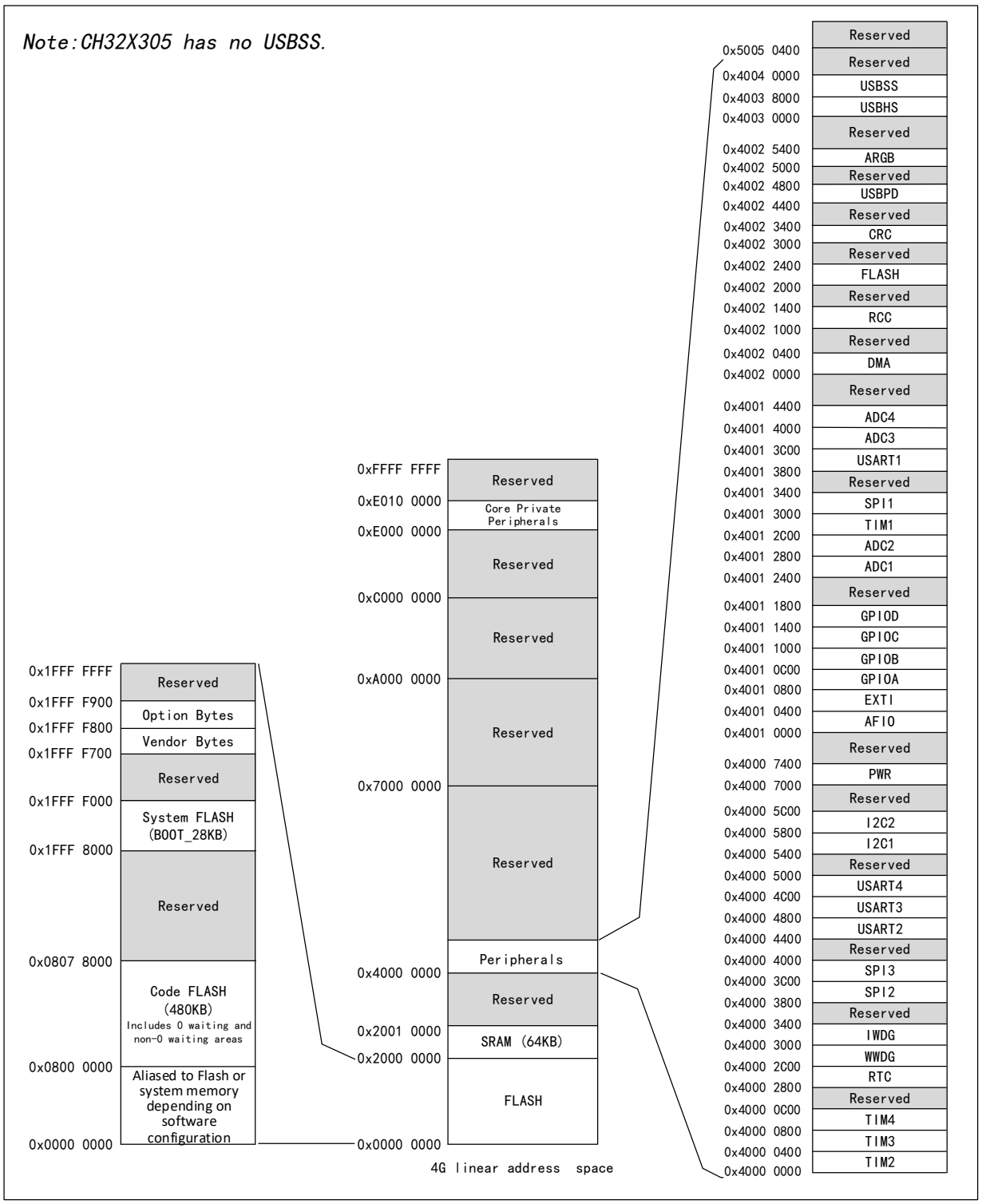
- The system bus connects the core to the bus matrix and is used to coordinate access to the core, DMA, SRAM and peripherals.
- The DMA bus is responsible for connecting the HB master control interface of the DMA to the bus matrix. The bus access objects are FLASH data, SRAM and peripherals.
- The bus matrix is responsible for the access coordination between the system bus, data bus, DMA bus, SRAM and HB bridge.

1.2 Memory Map

The chip contains program memory, data memory, core registers, peripheral registers, and more, all addressed in a 4GB linear space.

System storage stores data in small-end format, i.e., low bytes are stored at the low address and high bytes are stored at the high address.

Figure 1-2 Storage image



1.2.1 Memory Allocation

Built-in 64K bytes SRAM area for storing data, data will be lost after power failure.

It features 480KB of built-in program flash memory (Code FLASH), also known as the user area, which is used to store user applications and constant data. This includes 192KB of zero-wait program execution area and 288KB of non-zero-wait area.

Built-in 28K bytes system storage area (System FLASH), that is, BOOT area, used for system boot program storage (manufacturer solidified boot loader).

256 bytes are used for the system non-volatile configuration information storage area and are used for manufacturer configuration word storage. They are solidified before leaving the factory and cannot be modified by the user.

128 bytes are used for user-defined information storage area and used for user option byte storage.

For the usage and precautions of the BOOT function of the chip, please refer to the *CH32X315 Evaluation Board Reference*.

Chapter 2 Power Control (PWR)

2.1 Overview

The system operating voltage V_{DD} range is 2.8~3.6V, which supplies power to I/O pins, system voltage regulator LDO and built-in USB2.0 PHY. During normal operation, V_{DD} supplies power to the built-in system voltage regulator LDO and outputs regulated power on the V_{DDK} pin.

V_{DDK} powers the core circuitry and the built-in USB3.0 PHY.

V_{DDA} supplies power to the high-frequency RC oscillator, ADC, and analog portion of the PLL. The V_{DDA} voltage must be the same as the V_{DD} voltage.

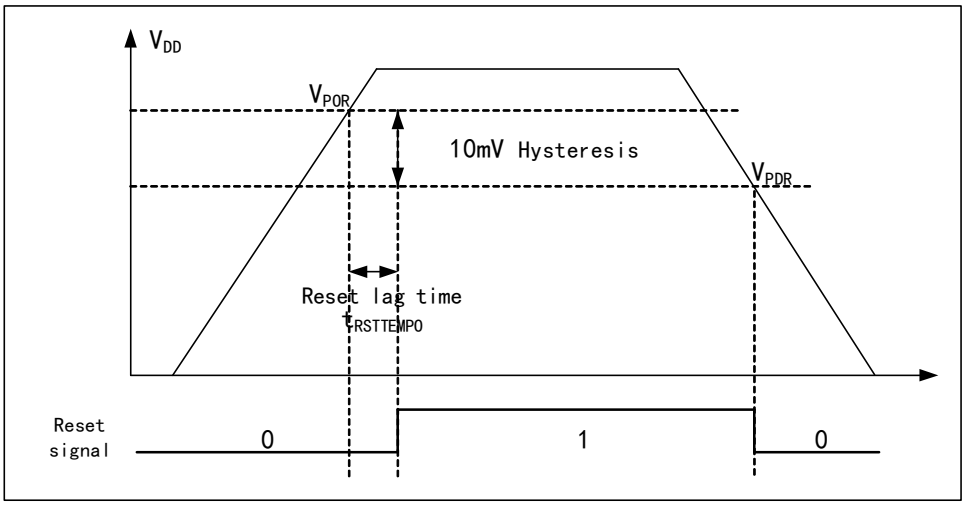
V_{REF+} and V_{REF-} serve as the positive and negative reference voltages of the ADC respectively, and V_{REF+} must not be higher than the V_{DDA} voltage.

2.2 Power Management

2.2.1 Power-on Reset and Power-down Reset

The system integrated a power-on reset POR and a power-down reset PDR circuit. When the chip supply voltage V_{DD} falls below the corresponding threshold voltage, the system is reset by the relevant circuit, and no additional external reset circuit is required. Please refer to the corresponding datasheet for the parameters of the power-on threshold voltage V_{POR} and the power-down threshold voltage V_{PDR}.

Figure 2-1 Schematic diagram of the operation of POR and PDR



2.2.2 Programmable Voltage Detector

The programmable voltage detector, PVD, is mainly used to monitor the change of the main power supply of the system, compared with the threshold voltage set by PLS[10] of the power control register PWR_CTLR, can be generated in order to notify the system to carry out the pre-dropout operation.

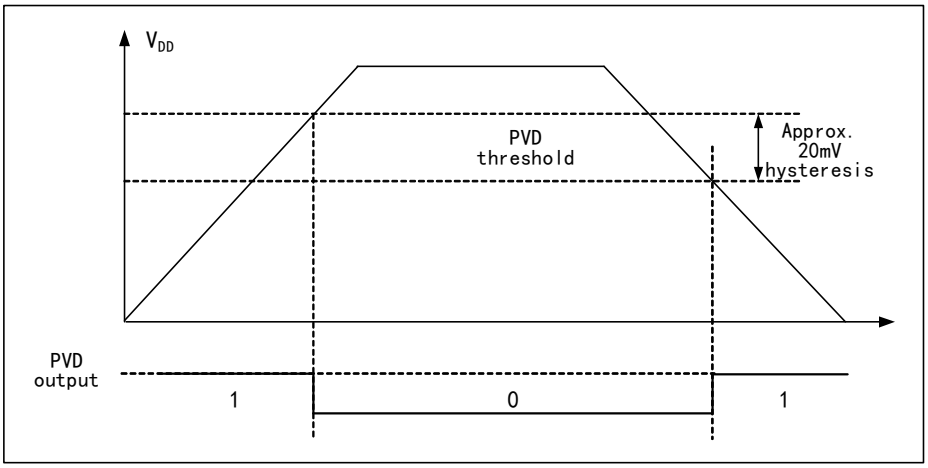
The specific configuration is as follows:

1) Set the PLS[1:0] fields of the PWR_CTLR register to select the voltage threshold to be monitored.

2) Set the PVDE bit of the PWR_CTLR register to enable the PVD function.

3) Read the PVD0 bit of PWR_CSR status register to obtain the relationship between the current system main power supply and the threshold value set by PLS[1:0], and perform the corresponding soft processing. When the V_{DD} voltage is higher than the PLS[1:0] setting threshold, the PVD0 position is 0; when the V_{DD} voltage is lower than the PLS[1:0] setting threshold, the PVD0 position is 1.

Figure 2-2 Schematic diagram of PVD operation



2.3 Low-power Modes

After a system reset, the microcontroller is in a normal operating state (Run mode), where system power can be saved by reducing the system main frequency or turning off the unused peripheral clock or reducing the operating peripheral clock. If the system does not need to work, you can set the system to enter low-power mode and let the system jump out of this state by specific events.

Microcontrollers currently offer 2 low-power modes, divided in terms of operating differences between processors, peripherals, voltage regulators, etc.

- Sleep mode: The core stops running and all peripherals (including core private peripherals) are still running.
- Stop mode: Stops all clocks and the system continues to run after waking up.

Table 2-1 Low-power mode list

Mode	Entry	Wake-up source	Effect on clock	Voltage regulator
SLEEP	WFI	Any interrupt	Core clock OFF, no effect on other clocks	Normal
	WFE	Wake-up event		
STOP	Set SLEEPDEEP to 1 WFI or WFE	Any external interrupt/event (EXTI signal), external reset signal on NRST, IWDG reset, where the EXTI signal includes one of the 64 external I/O ports, the output of PVD, RTC alarm clock, USBPD wake-up signal, USART wake-up signal or USB wake-up signal.	Disable HSE, HIS, PLL and peripheral clock	Normal: LPDS=0 or low-power consumption: LPDS=1

2.3.1 Low-power Configuration Options

- WFI and WFE

WFI: The microcontroller is woken up by an interrupt source with interrupt controller response, and the interrupt service function will be executed first after the system wakes up (except for microcontroller reset).

WFE: The wakeup event triggers the microcontroller to exit low-power mode. Wake-up events include:

1) Configure an external or internal EXTI line to event mode, when no interrupt controller needs to be configured.
2) Or configure an interrupt source, equivalent to a WFI wakeup, where the system prioritizes the execution of the interrupt service function.
3) Or configure the SEVONPEND bit to turn on peripheral interrupt enable, but not interrupt enable in the interrupt controller, and the interrupt pending bit needs to be cleared after the system wakes up.
4) The debugger's debugging request.

- SLEEPONEXIT

  Enable: After executing the WFI or WFE instruction, the microcontroller ensures that all pending interrupt services are exited and then enters low-power mode.

  Disable: The microcontroller enters low-power mode immediately after executing the WFI or WFE command.

- SEVONPEND

  Enable: All interrupts or wake-up events can wake up the low-power consumption entered by executing WFE.

  Disable: Only interrupts or wake-up events enabled in the interrupt controller can wake up the low-power consumption entered by executing WFE.

2.3.2 Sleep Mode

In this mode, all IO pins keep their state in Run mode and all peripheral clocks are normal, so try to turn off useless peripheral clocks before entering Sleep mode to reduce low-power consumption. This mode takes the shortest time to wake up.

Enter: Configure core register control bit SLEEPDEEP=0, power control register PDDS=0, execute WFI or WFE, optionally SEVONPEND and SLEEPONEXIT.

Exit: Arbitrary interrupt or wakeup event.

2.3.3 Stop Mode

Stop mode is a combination of peripheral clock control mechanisms based on the core's deep sleep mode (SLEEPDEEP) and allows the voltage regulator to operate in a much lower power consumption state. In this mode the high frequency clock (HSE/HSI/PLL) domain is switched off, the SRAM and register contents are maintained and the IO pin state is held. The system can continue to run after this mode wakes up and the HSI is called the default system clock.

If flash programming is in progress, the system does not enter stop mode until access to memory is complete; if access to the HB is in progress, the system does not enter stop mode until access to the HB is complete.

The voltage regulator LDO can be entered into a low-power mode from the normal mode through the configuration bit LPDS=1, thereby achieving a lower power consumption state.

Modules that can work in stop mode: independent watchdog (IWDG), real-time clock (RTC), low-frequency clock (LSI).

Enter: Set the core register control bit SLEEPDEEP to 1; optionally set the LPDS bit; execute WFI or WFE; optionally set SEVONPEND and SLEEPONEXIT.

Exit: Any external interrupt/event (EXTI signal), external reset signal on NRST, IWDG reset. The EXTI signal includes one of the 64 external I/O ports, the output of PVD, RTC alarm clock, USBPD wake-up signal, USART wake-up signal or USB wake-up signal.

In stop mode, the LPDS bit is optional, LPDS=0, the voltage regulator operates in normal mode; LPDS=1, the voltage regulator operates in low-power mode.

2.4 Register Description

Table 2-2 PWR-related registers list

Name	Access address	Description	Reset value
R32_PWR_CTLR	0x40007000	Power control register	0x00002D00
R32_PWR_CSR	0x40007004	Power control/status register	0x00000000

2.4.1 Power Control Register (PWR_CTLR)

Offset address: 0x00

31	30	29	28	27	26	25	24	23	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		LDO_VDD12A[2:0]			LDO_VDDK[2:0]			PLAN	PLS[1:0]		PVDE	Reserved			LPDS

Bit	Name	Access	Description	Reset value
[31:14]	Reserved	RO	Reserved	0
[13:11]	LDO_VDD12A[2:0]	RW	Adjust the voltage value of the internal analog power supply. *Note: It needs to be consistent with LDO_VDDK.*	101b
[10:8]	LDO_VDDK[2:0]	RW	Adjust the digital core voltage value and VDDK voltage value.	101b
7	PLAN	RW	Power adjustment bit preset function enable bit: 1: Enable the power preset function, the LDO_VDD12A/LDO_VDDK configuration value only takes effect during stop mode; 0: Turn off the power preset function, the LDO_VDD12A/LDO_VDDK configuration value takes effect immediately.	0
[6:5]	PLS[1:0]	RW	PVD voltage monitoring threshold setting. See the Electrical Characteristics section of the data sheet for details. 00: Rising edge 2.79V/falling edge 2.77V; 01: Rising edge 2.91V/falling edge 2.88V; 10: Rising edge 3.01V/falling edge 2.99V; 11: Rising edge 3.09V/falling edge 3.07V.	0

4	PVDE	RW	Power supply voltage monitoring function enable bit: 1: Enable the power supply voltage monitoring function; 0: Disable the power supply voltage monitoring function.	0
[3:1]	Reserved	RO	Reserved	0
0	LPDS	RW	In stop mode, the voltage regulator operating mode selection bit. 1: The voltage regulator works in low-power consumption mode; 0: The voltage regulator works in normal mode.	0

2.4.2 Power Control/Status Register (PWR_CSR)

Offset address: 0x04

31	30	29	28	27	26	25	24	23	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															PVDO

Bit	Name	Access	Description	Reset value
[31:1]	Reserved	RO	Reserved	0
0	PVDO	RO	PVD output status flag. This bit is valid when PVDE=1 in the PWR_CTLR register. 1: V_{DD} is lower than the PVD falling edge threshold set by PLS[1:0]; 0: V_{DD} is higher than the PVD rising edge threshold set by PLS[1:0].	0

Chapter 3 Reset and Clock Control (RCC)

The controller provides different forms of resets and configurable clock tree structures based on the division of power areas and peripheral power management considerations in the application. This section describes the scope of each clock in the system.

3.1 Main Features

- Multiple reset forms
- Multiple clock sources, bus clock management
- Built-in external crystal oscillation monitoring and clock security system
- Independent management of each peripheral clock: Reset, On, Off
- Supports internal clock output

3.2 Reset

The controller provides 2 forms of reset: Power Reset, System Reset.

3.2.1 Power Reset

When a power reset occurs, all registers will be reset.

The conditions for its production include:

- Power-on/power-off reset (POR/PDR reset)

3.2.2 System Reset

When a system reset occurs, all registers except the reset flag and the backup domain in the control/status register RCC_RSTSCKR are reset. The source of the reset event is identified by looking at the reset status flag bit in the RCC_RSTSCKR register.

The conditions for its production include:

- Low level signal on NRST pin (External reset)
- Window watchdog count terminated (WWDG reset)
- Independent watchdog count terminated (IWDG reset)
- Software reset (SW reset)
- USBPD reset
- ADC reset

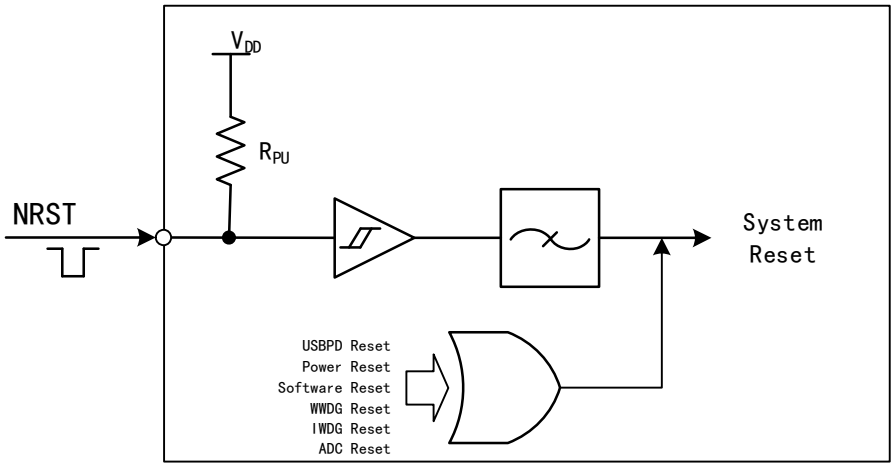
Window/independent watchdog reset: Triggered by window/independent watchdog peripheral timer count cycle overflow, see the corresponding chapter for a detailed description.

Software reset: This product resets the system through the SYSRST position 1 of the interrupt configuration register PFIC_CFGR in the programmable interrupt controller PFIC or the RSTSYS position 1 of the configuration register PFIC_SCTLR, refer to the corresponding chapter.

USBPD Reset: When PD_RST_EN is 1, the chip supports the reset generated by the Hard Reset of the USBPD signal frame. If IE_RX_RESET is also 1, the Reset generated by Cable Reset of signal frames is also supported. USBPD does not have a reset flag, but the resulting reset effect is the same as a software reset.

ADC reset: With the ADC Watchdog reset enabled, ADC reset occurs when ADC data is greater than the watchdog high threshold or less than the watchdog low threshold.

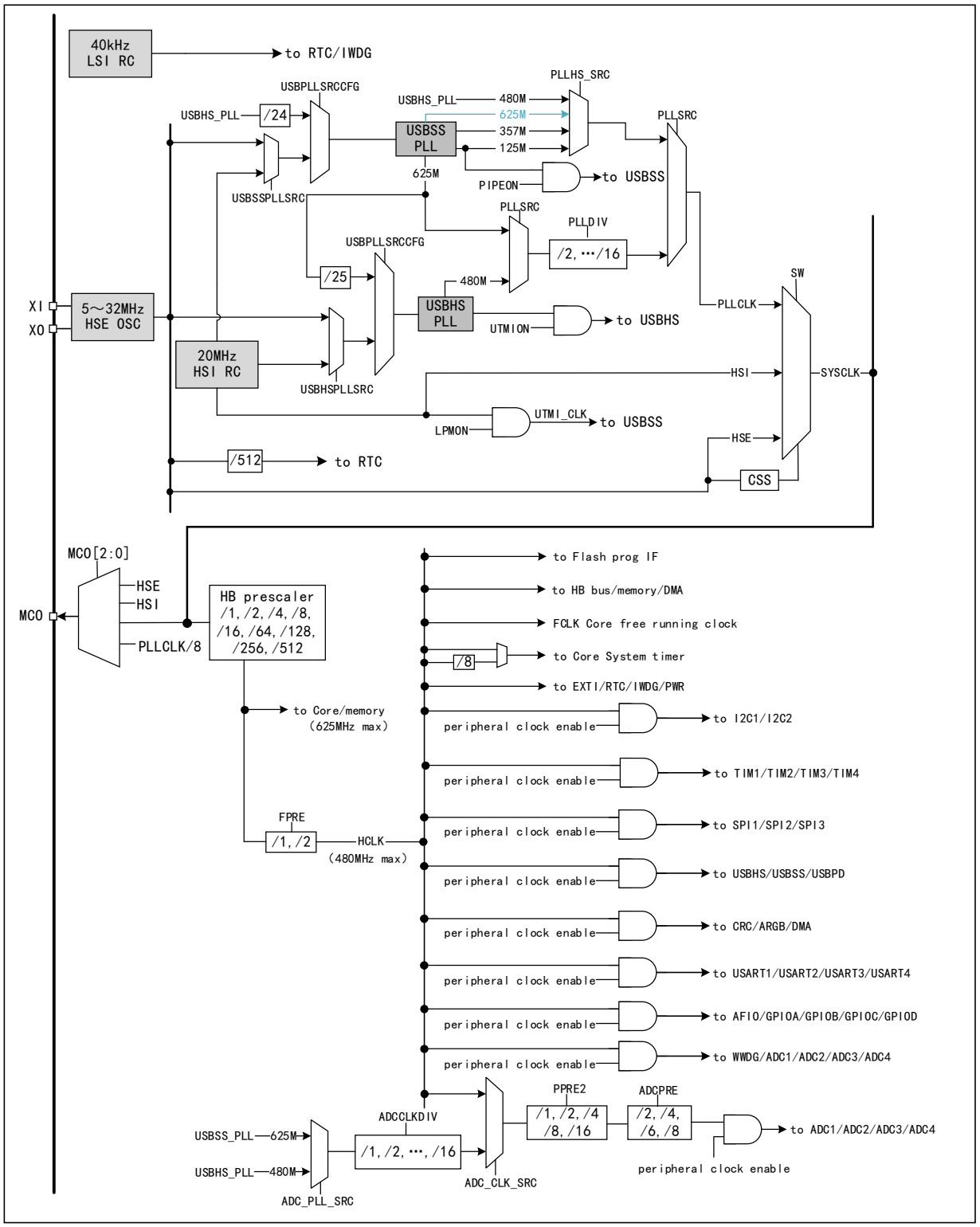
Figure 3-1 System reset structure



3.3 Clock

3.3.1 System Clock Structure

Figure 3-2 Clock tree block diagram



Note: The blue marked part in the above picture only applies to chips whose fifth digit of the lot number is greater than 0.

3.3.2 High-speed Clock (HSI/HSE)

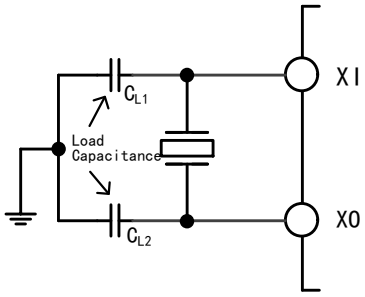
HSI is a high-speed clock signal generated by the RC oscillator of 20MHz in the system. The HIS RC oscillator can provide the system clock without any external devices. Its start-up time is very short, but the clock frequency accuracy is poor. The HSI is turned on and off by setting the HSION bit in the RCC_CTLR register, and the HSIRDY bit indicates whether the HSIRC oscillator is stable. The system defaults to HSION and HSIRDY setting 1 (it is recommended that you do not turn it off). If the HSIRDYIE bit of the RCC_INTR register is set, the corresponding interrupt will be generated.

Note: If the HSE crystal oscillator fails, the HSI clock will be used as a backup clock source (clock security system).

HSE is an external high-speed clock signal, including external crystal/ceramic resonator generation or external high-speed clock input.

- External Crystal/Ceramic Resonator (HSE Crystal): An external oscillator provides a more accurate clock source for the system. Further information can be found in the Electrical Characteristics section of the data sheet. The HSE crystal can be turned on and off by setting the HSEON bit in the RCC_CTLR register. The HSERDY bit indicates whether the HSE crystal oscillation is stable. The hardware only sends the clock to the system after the HSERDY bit is set to 1.
  If the HSERDYIE bit in the RCC_INTR register is set, the appropriate interrupt will be generated.

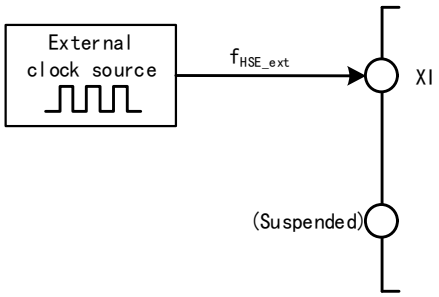
Figure 3-3 High-speed external crystal circuit



Note: The load capacitance needs to be as close as possible to the oscillator pins and the capacitance value selected according to the crystal manufacturer's parameters.

- External High Speed Clock Source (HSE Bypass): In this mode, the clock source is directly sent to the XI pin from the outside, and the XO pin is left floating. Supports up to 25MHz frequency. The application needs to set the HSEBYP bit when the HSEON bit is 0, turn on the HSE bypass function, and then set the HSEON bit.

Figure 3-4 High-speed clock source circuit



3.3.3 Low-speed Clock (LSI)

LSI is a low-speed clock signal generated by a RC oscillator of about 40KHz in the system. It can be kept running

in both downtime and standby modes, providing clock references for RTC clocks, independent watchdogs, and wake-up units. For further information, please refer to the electrical characteristics section of the data manual. LSI can be turned on and off by setting the LSION bit in the RCC_RSTSCKR register, and then check whether the LSI RC oscillation is stable by querying the LSIRDY bit, and the hardware sends the clock in after LSIRDY position 1. If the LSIRDYIE bit of the RCC_INTR register is set, the corresponding interrupt will be generated.

3.3.4 PLL Clock

By configuring the RCC_CFGR0 register, the internal PLL clock can select 5 clock sources and frequency multiplication factors. These settings must be completed before each PLL is started. Once the PLL is started, these parameters cannot be changed.

The USBHS_PLLON bit in the RCC_CTLR register is set to be enabled and disabled. The USBHS_PLLRDY bit indicates whether the USBHS_PLL clock is stable. The hardware only sends the clock to the system after the USBHS_PLLRDY bit is set to 1. Set the USBHSPLLRDYCFG bit in the RCC_CFGR2 register to configure the clock stabilization time.

The USBSS_PLLON bit in the RCC_CTLR register is set to be enabled and disabled. The USBSS_PLLRDY bit indicates whether the USBSS_PLL clock is stable. The hardware only sends the clock to the system after the USBSS_PLLRDY bit is set to 1. Set the USBSSPLLRDYCFG bit in the RCC_CFGR2 register to configure the clock stabilization time.

PLL clock source:

- USBHS_PLL (480MHz) clock input
- USBSS_PLL (125MHz) clock input
- USBSS_PLL (357MHz) clock input
- USBSS_PLL (625MHz) clock input

3.3.5 Bus/Peripheral Clock

3.3.5.1 System Clock (SYSCLK)

By configuring the RCC_CFGR0 register SW[1:0] bit to configure the system clock source, SWS[1:0] indicates the current system clock source.

- HSI as the system clock.
- HSE as the system clock.
- PLL clock as the system clock.

After the controller is reset, the default HSI clock is selected as the system clock source. The switching between clock sources does not occur until the target clock source is ready.

3.3.5.2 HB Bus Peripheral Clock (HCLK)

By configuring HPRE[3:0] of the RCC_CFGR0 register, you can configure the clock of the HB bus. These bus clocks determine the access clock base of the peripheral interfaces mounted below them. Applications can adjust different values to reduce power consumption when some peripherals are working.

Different peripheral modules can be reset and restored to the initial state through each bit in the RCC_HBRSTR, RCC_HB1PRSTR and RCC_HB2PRSTR registers.

The communication clock interface of different peripheral modules can be turned on or off separately through each

bit in the RCC_HBPCENR, RCC_HB1PCENR and RCC_HB2PCENR registers. When using a peripheral, you first need to turn on its clock enable bit before you can access its register.

3.3.5.3 Independent Watchdog Clock

If the independent watchdog has been started by the hardware configuration or software, the LSI oscillator will be forced to open and cannot be turned off. After the LSI oscillator is stabilized, the clock is supplied to the IWDG.

3.3.5.4 Microcontroller Clock Output (MCO)

The microcontroller allows clock signals to be output to MCO pins. The alternate push-pull output mode is configured in the corresponding GPIO port register. by configuring the RCC_CFGR0 register MCO[3:0] bit, the following 4 clock signals can be selected as MCO clock output:

- System clock (SYSCLK) output
- HSI clock output
- HSE clock output
- PLL clock output through 8 frequency division

3.3.6 Clock Security System

The clock security system is a running protection mechanism of the controller, which can switch to the HSI clock in the case of HSE clock transmission failure, and generate interrupt notification, allowing application software to complete the rescue operation.

Activate the clock security system by setting CSSON position 1 of the RCC_CTLR register. At this point, the clock monitor will be enabled after the HSE oscillator startup (HSERDY=1) delay and turned off after the HSE clock is turned off. Once the HSE clock fails during the operation of the system, the HSE oscillator will be turned off, the clock failure event will be sent to the brake input of the advanced timer (TIM1), and the clock security interrupt will be generated, CSSF position 1, and the application will enter the NMI unshielded interrupt. By setting the CSSC bit, the CSSF bit flag can be cleared and the NMI interrupt suspension bit can be revoked.

If the current HSE is used as the system clock, or the current HSE is used as the PLL input clock, and the PLL is used as the system clock, the clock safety system will automatically switch the system clock to the HSI oscillator when the HSE fails, and shut down the HSE oscillator and PLL.

3.3.7 RTC Calibration

This function must configure the PC8 pin to be used as a normal IO port.

- Pulse Output

Configure the ASOE bit of the RTC_CTLRL register, turn on the RTC pulse output, set the ASOS bit, and select second pulse output or alarm pulse output.

- RTC calibration

After configuring the CCO bit of the RTC_CTLRL register, the internal RTC clock will be divided by 64 and output to the PC8 pin. Through actual testing, the software cooperates with modifying the RTCCAL[6:0] bits to adjust the clock and calibrate the RTC.

3.4 Register Description

Table 3-1 RCC-related registers list

Name	Access address	Description	Reset value
R32_RCC_CTLR	0x40021000	Clock control register	0x00000003
R32_RCC_CFGR0	0x40021004	Clock configuration register 0	0x00000000
R32_RCC_INTR	0x40021008	Clock interrupt register	0x00000000
R32_RCC_HB2PRSTR	0x4002100C	HB2 peripheral reset register	0x00000000
R32_RCC_HB1PRSTR	0x40021010	HB1 peripheral reset register	0x00000000
R32_RCC_HBPCENR	0x40021014	HB peripheral clock enable register	0x00000000
R32_RCC_HB2PCENR	0x40021018	HB2 peripheral clock enable register	0x00000000
R32_RCC_HB1PCENR	0x4002101C	HB1 peripheral clock enable register	0x00000000
R32_RCC_RSTSCKR	0x40021024	Control/status register	0x08000000
R32_RCC_HBRSTR	0x40021028	HB peripheral reset register	0x00000000
R32_RCC_CFGR2	0x4002102C	Clock configuration register 2	0x10000001

3.4.1 Clock Control Register (RCC_CTLR)

Offset address: 0x00

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PLL_GATE	Reserved	PI_XI_STAT	PIPEON	USBSS_PLLRDY	USBSS_PLLON	USBHS_PLLRDY	USBHS_PLLON	Reserved				CSSON	HSEBYP	HSERDY	HSEON

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

Bit	Name	Access	Description	Reset value
31	PLL_GATE	RW	PLL clock gating enable: 1: Disable; 0: Enable.	0
30	Reserved	RO	Reserved	0
29	PI_XI_STAT	RO	Crystal oscillator indication bit: 1: With crystal oscillator; 0: Without crystal oscillator.	0
28	PIPEON	RW	PIPE clock gating enable: 1: Enable; 0 Disable.	0
27	USBSS_PLLRDY	RO	USBSS_PLL clock ready lock flag bit: : USBSS_PLL clock is locked; 0: USBSS_PLL clock is not locked.	0
26	USBSS_PLLON	RW	USBSS_PLL clock enable control bit: 1: Enable USBSS_PLL clock; 0: Turn off USBSS_PLL clock.	0

			The hardware is automatically cleared after entering deep sleep;	
25	USBHS_PLLRDY	RO	USBHS_PLL clock ready lock flag bit: 1: USBHS_PLL clock is locked; 0: USBHS_PLL clock is not locked.	0
24	USBHS_PLLON	RW	USBHS_PLL clock enable control bit: 1: Enable USBHS_PLL clock; 0: Turn off USBHS_PLL clock. The hardware is automatically cleared after entering deep sleep;	0
[23:20]	Reserved	RO	Reserved	0
19	CSSON	RW	Clock security system enable control bit: 1: Enable clock security system. When the HSE is ready (HSERDY is set to 1), the hardware turns on the clock monitoring function of the HSE. If the HSE abnormality is found, the CSSF flag and NMI interrupt are triggered; when the HSE is not ready, the hardware turns off the clock monitoring function of the HSE. 0: Disable clock security system.	0
18	HSEBYP	RW	External high-speed crystal bypass control bit: 1: Bypass external high-speed crystal/ceramic resonator (use external clock source); 0: Do not bypass high-speed external crystal/ceramic resonator. *Note: This bit needs to be written when HSEON is 0.*	0
17	HSERDY	RO	External high-speed crystal oscillation stable ready flag (set by hardware): 1: The external high-speed crystal oscillation is stable; 0: The external high-speed crystal oscillation is not stable. *Note: After the HSEON bit is cleared to 0, this bit requires 1 HSE cycle to be cleared to 0.*	0
16	HSEON	RW	External high-speed crystal oscillator enable control bit: 1: Enable HSE oscillator; 0: Turn off HSE oscillator. *Note: After entering the stop low power mode, this bit is cleared by hardware.*	0
[15:9]	Reserved	RO	Reserved	0
8	LPMON	RW	LPM clock switch: 1: On; 0: Off.	0
[7:2]	Reserved	RO	Reserved.	0
1	HSIRDY	RO	Internal high-speed clock (20MHz) stable ready flag	1

			(set by hardware): 1: The internal high-speed clock (20MHz) is stable; 0: The internal high-speed clock (20MHz) is not stable. *Note: After the HSION bit is cleared to 0, this bit requires 6 HSI cycles to be cleared to 0.*	
0	HSION	RW	Internal high-speed clock (20MHz) enable control bit: 1: Enable HSI oscillator; 0: Disable HSI oscillator. *Note: When returning from stop mode or the external oscillator HSE used as the system clock fails, this bit is set by hardware to start the internal 20MHz RC oscillator.*	1

3.4.2 Clock Configuration Register 0 (RCC_CFGR0)

Offset address: 0x04

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved		PLLHS_SRC[1:0]		MCO[3:0]				UTMION	PLLSRC[1:0]		PLLDIV[3:0]				Reserved

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADCPRE[1:0]		PPRE2[2:0]			Reserved		FPRE	HPRE[3:0]				SWS[1:0]		SW[1:0]	

Bit	Name	Access	Description	Reset value
[31:30]	Reserved	RO	Reserved	0
[29:28]	PLLHS_SRC[1:0]	RW	PLL Clock selection (Used in conjunction with the PLLSRC[1:0] bits): 00: Select the USBSS PLL 125MHz clock as the PLL clock; 01: Select the USBSS PLL 357MHz clock as the PLL clock; 10: Selects the USBSS PLL 625MHz clock as the PLL clock; 11: Selects the USBHS PLL 480MHz clock as the PLL clock. *Note: (1) This bit is active when PLLSRC[1:0] = 01b or PLLSRC[1:0] = 11b.* *(2) 10b is only applicable to chips whose fifth digit of the lot number is greater than 0.*	0
[27:24]	MCO[3:0]	RW	Microcontroller MCO pin clock output control: 00xx: No clock output; 0100: system clock (SYSCLK) output; 0101: Internal 20MHz RC oscillator clock (HSI) output; 0110: External oscillator clock (HSE) output;	0

			0111: PLL clock output after dividing by 8; Others: Reserved.	
23	UTMION	RW	UTMI clock gating enable: 1: Enable; 0: Disable.	0
[22:21]	PLLSRC[1:0]	RW	PLLCLK clock selection: 00: Select the USBHS PLL 480MHz divided clock as the PLL clock; x1: Select the clock via the PLLHS_SRC[1:0] bits; 10: Select the USBSS PLL 625MHz divided clock as the PLL clock.	0
[20:17]	PLLDIV[3:0]	RW	Divider factors for the USBSS PLL 625MHz clock and the USBHS PLL 480MHz clock (used in conjunction with the PLLSRC[1:0] bits): 0000: Divided by 6; 0001: Divided by 2; 0010: Divided by 3; 0011: Divided by 4; 0100: Divided by 5; 0101: Divided by 6; 0110: Divided by 7; 0111: Divided by 8; 1000: Divided by 9; 1001: Divided by 10; 1010: Divided by 11 1011: Divided by 12; 1100: Divided by 13; 1101: Divided by 14; 1110: Divided by 15; 1111: Divided by 16.	5
16	Reserved	RO	Reserved	0
[15:14]	ADCPRE[1:0]	RW	ADC clock prescaler control lower 2 bits (used in conjunction with PPRE2[2:0] bits): 00: Divided by 2; 01: Divided by 4; 10: Divided by 6; 11: Divided by 8. *Note: The maximum ADC clock should not exceed 80MHz.*	0
[13:11]	PPRE2[2:0]	RW	ADC clock prescaler control high 3 bits (used in conjunction with ADCPRE[1:0] bits): 0xx: No frequency division; 100: Divided by 2; 101: Divided by 4;	0

			110: Divided by 8; 111: Divided by 16.	
[10:9]	Reserved	RO	Reserved	0
8	FPRE	RW	HCLK input clock source prescaler control: 1: Divided by 2; 0: No frequency division.	0
[7:4]	HPRE[3:0]	RW	HB clock source prescaler control: 0xxx: SYSCLK does not divide frequency; 1000: SYSCLK divided by 2; 1001: SYSCLK divided by 4; 1010: SYSCLK divided by 8; 1011: SYSCLK divided by 16; 1100: SYSCLK divided by 64; 1101: SYSCLK divided by 128; 1110: SYSCLK divided by 256; 1111: SYSCLK divided by 512.	0
[3:2]	SWS[1:0]	RO	System clock (SYSCLK) status (hardware set): 00: The system clock source is HSI; 01: The system clock source is HSE; 10: The system clock source is PLL; 11: Not available.	0
[1:0]	SW[1:0]	RW	Select system clock source: 00: HSI is used as the system clock; 01: HSE is used as the system clock; 10: PLL output is used as the system clock; 11: Not available. *Note: When the clock safety system is enabled (CSSON=1), when returning from stop mode or the external oscillator HSE used as the system clock fails, HSI is forced to be selected as the system clock by hardware.*	0

3.4.3 Clock Interrupt Register (RCC_INTR)

Offset address: 0x08

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved								CSSC	Reserved			HSE RDYC	HSIR DYC	Reserved	LSIR DYC

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				HSE RDYIE	HSI RDYIE	Reserved	LSI RDYIE	CSSF	Reserved			HSE RDYF	HSI RDYF	Reserved	LSI RDYF

Bit	Name	Access	Description	Reset value

[31:24]	Reserved	RO	Reserved	0
23	CSSC	WO	Clear clock security system interrupt flag (CSSF): 1: Clear CSSF interrupt flag; 0: No effect.	0
[22:20]	Reserved	RO	Reserved	0
19	HSERDYC	WO	Clear HSE oscillator ready interrupt flag: 1: Clear HSERDYF interrupt flag; 0: No effect.	0
18	HSIRDYC	WO	Clear HSI oscillator ready interrupt flag: 1: Clear HSIRDYF interrupt flag; 0: No effect.	0
17	Reserved	RO	Reserved	0
16	LSIRDYC	WO	Clear LSI oscillator ready interrupt flag: 1: Clear LSIRDYF interrupt flag; 0: No effect.	0
[15:12]	Reserved	RO	Reserved	0
11	HSERDYIE	RW	HSE ready interrupt enable: 1: Enable HSE ready interrupt; 0: Disable HSE ready interrupt.	0
10	HSIRDYIE	RW	HSI ready interrupt enable: 1: Enable HSI ready interrupt; 0: Disable HSI ready interrupt.	0
9	Reserved	RO	Reserved	0
8	LSIRDYIE	RW	LSI ready interrupt enable: 1: Enable LSI ready interrupt; 0: Disable LSI ready interrupt.	0
7	CSSF	RO	Clock security system interrupt flag bit: 1: HSE clock failure, resulting in clock security interrupt CSSI. 0: No clock security system interrupt. Set by hardware, clear when software write CSSC bit 1.	0
[6:4]	Reserved	RO	Reserved	0
3	HSERDYF	RO	HSE clock ready interrupt flag: 1: HSE clock ready generates an interrupt; 0: No HSE clock ready interrupt. Set by hardware, clear when software write HSERDYC bit 1.	0
2	HSIRDYF	RO	HSI clock ready interrupt flag: 1: HSI clock ready generates an interrupt; 0: No HSI clock ready interrupt. Set by hardware, clear when software write HSIRDYC bit 1.	0
1	Reserved	RO	Reserved	0
0	LSIRDYF	RO	LSI clock ready interrupt flag:	0

			1: LSI clock ready generates an interrupt; 0: No LSI clock ready interrupt. Set by hardware, clear when software write LSIRDYC bit 1.	

3.4.4 HB2 Peripheral Reset Register (RCC_HB2PRSTR)

Offset address: 0x0C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															ADC4 RST

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADC3 RST	USART1RST	Reserved	SPI1 RST	TIM1 RST	ADC2 RST	ADC1 RST	Reserved			IOPD RST	IOPC RST	IOPB RST	IOPA RST	Reserved	AFIO RST

Bit	Name	Access	Description	Reset value
[31:17]	Reserved	RO	Reserved	0
16	ADC4RST	RW	ADC4 module reset control: 1: Reset module; 0: No effect.	0
15	ADC3RST	RW	ADC3 module reset control: 1: Reset module; 0: No effect.	0
14	USART1RST	RW	USART1 interface reset control: 1: Reset module; 0: No effect.	0
13	Reserved	RO	Reserved	0
12	SPI1RST	RW	SPI1 interface reset control: 1: Reset module; 0: No effect.	0
11	TIM1RST	RW	TIM1 module reset control: 1: Reset module; 0: No effect.	0
10	Reserved	RO	Reserved.	0
9	ADC1RST	RW	ADC1 module reset control: 1: Reset module; 0: No effect.	0
[8:6]	Reserved	RO	Reserved	0
5	IOPDRST	RW	IO's PD port module reset control: 1: Reset module; 0: No effect.	0
4	IOPCRST	RW	IO's PC port module reset control: 1: Reset module; 0: No effect.	0
3	IOPBRST	RW	IO's PB port module reset control: 1: Reset module; 0: No effect.	0
2	IOPARST	RW	IO's PA port module reset control: 1: Reset module; 0: No effect.	0
1	Reserved	RO	Reserved	0
0	AFIORST	RW	I/O auxiliary function module reset control: 1: Reset module; 0: No effect.	0

3.4.5 HB1 Peripheral Reset Register (RCC_HB1PRSTR)

Offset address: 0x10

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved			PWRRST	Reserved					I2C2RST	I2C1RST	Reserved	USART4RST	USART3RST	USART2RST	Reserved

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SPI3RST	SPI2RST	Reserved		WWDGRST	Reserved								TIM4RST	TIM3RST	TIM2RST

Bit	Name	Access	Description	Reset value
[31:29]	Reserved	RO	Reserved	0
28	PWRRST	RW	Power interface module reset control: 1: Reset module; 0: No effect.	0
[27:23]	Reserved	RO	Reserved	0
22	I2C2RST	RW	I2C2 module reset control: 1: Reset module; 0: No effect.	0
21	I2C1RST	RW	I2C1 module reset control: 1: Reset module; 0: No effect.	0
20	Reserved	RO	Reserved	0
19	USART4RST	RW	USART4 module reset control: 1: Reset module; 0: No effect.	0
18	USART3RST	RW	USART3 module reset control: 1: Reset module; 0: No effect.	0
17	USART2RST	RW	USART2 module reset control: 1: Reset module; 0: No effect.	0
16	Reserved	RO	Reserved	0
15	SPI3RST	RW	SPI3 module reset control: 1: Reset module; 0: No effect.	0
14	SPI2RST	RW	SPI2 interface reset control: 1: Reset module; 0: No effect.	0
[13:12]	Reserved	RO	Reserved	0
11	WWDGRST	RW	WWDG reset control: 1: Reset module; 0: No effect.	0
[10:3]	Reserved	RO	Reserved	0
2	TIM4RST	RW	Timer 4 module reset control: 1: Reset module; 0: No effect.	0
1	TIM3RST	RW	Timer 3 module reset control: 1: Reset module; 0: No effect.	0
0	TIM2RST	RW	Timer 2 module reset control: 1: Reset module; 0: No effect.	0

3.4.6 HB Peripheral Clock Enable Register (RCC_HBPCENR)

Offset address: 0x14

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
USBSSEN	ARGBEN	Reserved	USBPDEN	USBHSEN	Reserved				CRCEN	Reserved					DMA1EN

Bit	Name	Access	Description	Reset value
[31:16]	Reserved	RO	Reserved	0
15	USBSSEN	RW	USBSS module clock enable: 1: Module clock on; 0: Module clock off.	0
14	ARGBEN	RW	ARGB module clock enable: 1: Module clock on; 0: Module clock off.	0
13	Reserved	RO	Reserved	0
12	USBPDEN	RW	USBPD module clock enable: 1: Module clock on; 0: Module clock off.	0
11	USBHSEN	RW	USBHS module clock enable: 1: Module clock on; 0: Module clock off.	0
[10:7]	Reserved	RO	Reserved	0
6	CRCEN	RW	CRC module clock enable: 1: Module clock on; 0: Module clock off.	0
[5:1]	Reserved	RO	Reserved	0
0	DMA1EN	RW	DMA1 module clock enable: 1: Module clock on; 0: Module clock off.	0

3.4.7 HB2 Peripheral Clock Enable Register (RCC_HB2PCENR)

Offset address: 0x18

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															ADC4EN

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADC3EN	USART1EN	Reserved	SPI1EN	TIM1EN	ADC2EN	ADC1EN	Reserved			IOPDEN	IOPCEN	IOPBEN	IOPAEN	Reserved	AFIOEN

Bit	Name	Access	Description	Reset value
[31:17]	Reserved	RO	Reserved	0
16	ADC4EN	RW	ADC4 module clock enable: 1: Module clock on; 0: Module clock off.	0
15	ADC3EN	RW	ADC3 module clock enable:	0

			1: Module clock on; 0: Module clock off.	
14	USART1EN	RW	USART1 module clock enable: 1: Module clock on; 0: Module clock off.	0
13	Reserved	RO	Reserved	0
12	SPI1EN	RW	SPI1 module clock enable: 1: Module clock on; 0: Module clock off.	0
11	TIM1EN	RW	TIM1 module clock enable: 1: Module clock on; 0: Module clock off.	0
10	ADC2EN	RW	ADC2 module clock enable: 1: Module clock on; 0: Module clock off.	0
9	ADC1EN	RW	ADC1 module clock enable: 1: Module clock on; 0: Module clock off.	0
[8:6]	Reserved	RO	Reserved	0
5	IOPDEN	RW	IO'PD port module clock enable: 1: Module clock on; 0: Module clock off.	0
4	IOPCEN	RW	IO'PC port module clock enable: 1: Module clock on; 0: Module clock off.	0
3	IOPBEN	RW	IO'PB port module clock enable: 1: Module clock on; 0: Module clock off.	0
2	IOPAEN	RW	IO'PA port module clock enable: 1: Module clock on; 0: Module clock off.	0
1	Reserved	RO	Reserved	0
0	AFIOEN	RW	IO auxiliary function module clock enable: 1: Module clock on; 0: Module clock off.	0

3.4.8 HB1 Peripheral Clock Enable Register (RCC_HB1PCENR)

Offset address: 0x1C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved			PWREN	Reserved					I2C2EN	I2C1EN	Reserved	USART4EN	USART3EN	USART2EN	Reserved

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SPI3EN	SPI2EN	Reserved		WWDGEN	Reserved								TIM4EN	TIM3EN	TIM2EN

Bit	Name	Access	Description	Reset value
[31:29]	Reserved	RO	Reserved	0
28	PWREN	RW	Power interface module clock enable: 1: Module clock on; 0: Module clock off.	0
[27:23]	Reserved	RO	Reserved	0
22	I2C2EN	RW	I2C2 module clock enable: 1: Module clock on; 0: Module clock off.	0
21	I2C1EN	RW	I2C1 module clock enable:	0

			1: Module clock on; 0: Module clock off.	
20	Reserved	RW	Reserved	0
19	USART4EN	RW	USART4 module clock enable: 1: Module clock on; 0: Module clock off.	0
18	USART3EN	RW	USART3 module clock enable: 1: Module clock on; 0: Module clock off.	0
17	USART2EN	RW	USART2 module clock enable: 1: Module clock on; 0: Module clock off.	0
16	Reserved	RO	Reserved	0
15	SPI3EN	RW	SPI3 module clock enable: 1: Module clock on; 0: Module clock off.	0
14	SPI2EN	RW	SPI2 module clock enable: 1: Module clock on; 0: Module clock off.	0
[13:12]	Reserved	RO	Reserved	0
11	WWDGEN	RW	WWDG clock enable: 1: Module clock on; 0: Module clock off.	0
[10:3]	Reserved	RO	Reserved	0
2	TIM4EN	RW	Timer 4 module clock enable: 1: Module clock on; 0: Module clock off.	0
1	TIM3EN	RW	Timer 3 module clock enable: 1: Module clock on; 0: Module clock off.	0
0	TIM2EN	RW	Timer 2 module clock enable: 1: Module clock on; 0: Module clock off.	0

3.4.9 Control/Status Register (RCC_RSTSCKR)

Offset address: 0x24

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved	WWDGRSTF	IWDGRSTF	SFTRSTF	PORRSTF	PINRSTF	Reserved	RMVF	ADC2AWDRSTF	ADC1AWDRSTF	USBPDRSTF	ADC4AWDRSTF	ADC3AWDRSTF	LKUPRSTF	Reserved	

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

Bit	Name	Access	Description	Reset value
31	Reserved	RO	Reserved	0
30	WWDGRSTF	RO	Window watchdog reset flag: 1: Occurrence of a window watchdog reset. 0: No window watchdog reset occurs. Set to 1 by hardware when a window watchdog reset occurs; cleared by software writing of the RMVF bit.	0
29	IWDGRSTF	RO	Independent watchdog reset flag: 1: Occurrence of an independent watchdog reset.	0

			0: No independent watchdog reset occurs. Set to 1 by hardware when an independent watchdog reset occurs; cleared by software writing of the RMVF bit.	
28	SFTRSTF	RO	Software reset flag: 1: Software reset occurs. 0: No software reset occurs. Set to 1 by hardware when a software reset occurs; software write RMVF bit cleared.	0
27	PORRSTF	RO	Power-up/power-down reset flag: 1: Power-up/power-down reset occurs. 0: No power-up/power-down reset occurs. Set to 1 by hardware when power-up/power-down reset occurs; cleared by software writing of RMVF bit.	1
26	PINRSTF	RO	External manual reset (NST pin) flag: 1: Occurrence of NST pin reset. 0: No NST pin reset occurs. Set to 1 by hardware when NST pin reset occurs; cleared by software writing of RMVF bit.	0
25	Reserved	RO	Reserved	0
24	RMVF	WO	Clear reset flag control: 1: Clear the reset flag. 0: No effect.	0
23	ADC2AWDRSTF	RO	ADC2 watchdog reset flag: 1: ADC2 watchdog reset occurs; 0: Normal. When an ADC2 watchdog reset occurs, it is set to 1 by hardware; software writes the RMVF bit to clear it.	0
22	ADC1AWDRSTF	RO	ADC1 watchdog reset flag: 1: ADC1 watchdog reset occurs; 0: Normal. When an ADC1 watchdog reset occurs, it is set to 1 by hardware; software writes the RMVF bit to clear it.	0
21	USBPDRSTF	RO	USBPD reset flag: 1: USBPD reset occurs; 0: Normal. Set by hardware when a USBPD reset occurs; software writes the RMVF bit to clear it.	0
20	ADC4AWDRSTF	RO	ADC4 watchdog reset flag: 1: ADC4 watchdog reset occurs; 0: Normal. When an ADC4 watchdog reset occurs, it is set to 1 by hardware; software writes the RMVF bit to clear it.	0
19	ADC3AWDRSTF	RO	ADC3 watchdog reset flag: 1: ADC3 watchdog reset occurs;	0

			0: Normal. When an ADC3 watchdog reset occurs, it is set to 1 by hardware; software writes the RMVF bit to clear it.	
18	LKUPRSTF	RO	LOCKUP reset flag: 1: LOCKUP occurs and causes system reset; 0: Normal. When a LOCKUP reset occurs, it is set to 1 by hardware; software writes the RMVF bit to clear it.	0
[17:2]	Reserved	RO	Reserved	0
1	LSIRDY	RO	Internal low-speed clock (LSI) stabilization ready flag bit (set by hardware): 1: The internal low-speed clock (40kHz) is stabilized; 0: The internal low-speed clock (40kHz) is not stabilized. *Note: After the LSION bit is cleared to 0, the bit takes 3 LSI cycles to clear to 0.*	0
0	LSION	RW	Internal low-speed clock (LSI) enable control bits: 1: Enables the LSI (40kHz) oscillator; 0: Disables the LSI (40kHz) oscillator.	0

Note: Except for BIT1, which is cleared by power-on reset, other write-clear reset flags can be cleared.

3.4.10 HB Peripheral Reset Register (RCC_HBRSTR)

Offset address: 0x28

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

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

Bit	Name	Access	Description	Reset value
[31:16]	Reserved	RO	Reserved	0
15	USBSSRST	RW	USBSS module reset control: 1: Reset module; 0: No effect.	0
14	ARGBRST	RW	ARGB module reset control: 1: Reset module; 0: No effect.	0
13	Reserved	RO	Reserved	0
12	USBPDRST	RW	USBPD module reset control: 1: Reset module; 0: No effect.	0
11	USBHSRST	RW	USBHS module reset control: 1: Reset module; 0: No effect.	0
[10:0]	Reserved	RO	Reserved	0

3.4.11 Clock Configuration Register 2 (RCC_CFGR2)

Offset address: 0x2C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ADCCLKDIV[3:0]				ADCPLLSRC	ADCDUTYMD[1:0]		ADCSRC	Reserved		USBSSPLLRDYCFG[1:0]		USBHSPLLRDYCFG[1:0]		HSERDYCFG[1:0]	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved						USBSSPLLSRC	USBHSPLLSRC	USBPLLSRCCFG[1:0]		USBSSPLLREFSEL[2:0]			USBHSPLLREFSEL[2:0]		

Bit	Name	Access	Description	Reset value
[31:28]	ADCCLKDIV[3:0]	RW	ADC input clock source selection USBHS_PLL480M or USBSS PLL625M prescaler control bits: 0000: No frequency division; 0001: Divided by 2; 0010: Divided by 3; 0011: Divided by 4; 0100: Divided by 5; 0101: Divided by 6; 0110: Divided by 7; 0111: Divided by 8; 1000: Divided by 9; 1001: Divided by 10; 1010: Divided by 11; 1011: Divided by 12; 1100: Divided by 13; 1101: Divided by 14; 1110: Divided by 15; 1111: Divided by 16.	0x1
27	ADCPLLSRC	RW	ADC input clock source selection (used in conjunction with the ADCSRC bit): 1: Select the USBSSPLL 625M clock as the ADC input clock source; 0: Select the USBHSPLL 480M clock as the ADC input clock source. *Note: When ADC is used, this bit is valid when ADCSRC=1.*	0
[26:25]	ADCDUTYMD[1:0]	RW	ADC clock duty cycle mode selection: 00: 50%; 01: 75%; 10: 62.5%; 11: 50%.	0
24	ADCSRC	RW	ADC input clock selection:	0

			1: Select the ADC input clock through the ADCPLLSRC bit; 0: Select HCLK as the ADC input clock.	
[23:22]	Reserved	RO	Reserved	0
[21:20]	USBSSPLLRDYCFG [1:0]	RW	USBSS PLL clock stabilization time configuration bits: 00: 21.6us; 01: 16us; 10: 35.2us; 11: 60us.	0
[19:18]	USBHSPLLRDYCFG [1:0]	RW	USBHS PLL clock stabilization time configuration bits: 00: 18.4us; 01: 13.6us; 10: 25.6us; 11: 45.6us.	0
[17:16]	HSERDYCFG[1:0]	RW	HSE clock stabilization time configuration bits: 00: 307.2us; 01: 204.8us; 10: 409.6us; 11: 512us.	0
[15:10]	Reserved	RO	Reserved	0
9	USBSSPLLSRC	RW	USBSS PLL clock source selection (Used in conjunction with the USBPLLSRCCFG[1:0] bits): 1: HSI; 0: HSE.	0
8	USBHSPLLSRC	RW	USBHS PLL clock source selection (Used in conjunction with the USBPLLSRCCFG[1:0] bits): 1: HSI; 0: HSE.	0
[7:6]	USBPLLSRCCFG[1:0]	RW	USBHS PLL clock source selection: 01: USBSS PLL 625MHz clock divided by 25; Other: Select the clock via the USBHSPLLSRC bit. USBSS PLL clock source selection: 10: USBHS PLL 480MHz clock divided by 24; Others: Select the clock via the USBSSPLLSRC bit.	0
[5:3]	USBSSPLLREFSEL [2:0]	RW	USBSS_PLL reference clock selection, writing is only allowed when the system clock source is HSE or HSI: 000: 20MHz; 001: 24MHz; 010: 25MHz; 011: 30MHz; 100: 32MHz;	0

			101: 40MHz; 110: 60MHz; 111: 80MHz.	
[2:0]	USBHSPLLREFSEL [2:0]	RW	USBHS_PLL reference clock selection, writing is only allowed when the system clock source is HSE or HSI: 000: 12MHz; 001: 20MHz; 010: 24MHz; 011: 25MHz; 100: 32MHz; Others: Reserved.	0x1

Chapter 4 Addressable RGB (ARGB)

The chip has a built-in set of ARGB (Addressable RGB) modules that support configurable duty cycle and period of the output PWM waveform, and support DMA.

The ARGB module can realize timing control of single data line RGB light strips and supports cascading. Compared with traditional RGB, it can support driving more different types of ARGB light control ICs to produce more light effects.

4.1 Main Features

- Support configurable output of code 0 and code 1 high- and low-level time and period
- Support DMA

4.2 Function Description

4.2.1 Polling and Interrupts

When using polling or interrupt mode to drive this peripheral, it is important to note that the ARGB internal buffer is only a single byte. During a data transmission cycle, the hardware will not detect whether new data is written to the R32_ARGB_DATAR register; when the 1-byte data transmission is completed, if the new data cannot be filled in time, the hardware will repeatedly send the last byte that has been sent.

Therefore, in order to ensure that the data sent is correct, the real-time nature of data writing must be ensured, and the writing operation of the next byte needs to be completed within the time window when TXEIF is set and BYTEIF is set.

Configuration phase

- Configure the MODE bit and LSB bit of the R32_ARGB_CTRL register to select the data frame structure.
- Configure the R32_ARGB_CCRL, R32_ARGB_CCRH, R32_ARGB_DAT_ARR, and R32_ARGB_RST_ARR registers to set the physical protocol parameters.
- Configure R32_ARGB_DAT_CYC to configure the data frame sending length.
- Enable the TX_EN bit in the R32_ARGB_CTRL register.

If the interrupt mode is used: Since the status of the TXEIF flag needs to be monitored during data transmission, the TXEIE bit of the R32_ARGB_CTLR register needs to be additionally set and the ARGB interrupt must be enabled.

Data transmitting phase

- Check the STATUS bit of the R32_ARGB_STATR register to confirm that the current controller is in the IDLE state.
- Write data to the R32_ARGB_DATAR register to trigger a new round of frame transmission.
- Wait for the TXEIF bit to be set and write the next set of data within the valid time window. When using the interrupt mode, the next set of data needs to be written in the interrupt service function under the condition that the TXE interrupt is triggered.
- Query the TCIF flag bit or software counting to determine whether all current data frames have been sent.
- If you need to update the display content of the light control IC, just repeat the above configuration and sending

steps.

4.2.2 DMA

The DMA configuration process is different from the polling method. The main difference is: in DMA mode, the detection and data filling of TXEIE during the data sending phase are completed by the ARGB peripheral and the DMA controller hardware. Under the condition that the bus is not continuously occupied by extremely high priority, the hardware can always complete data filling in time. Therefore, it is recommended to use DMA to drive external devices, which can improve the reliability of data transmission.

Configuration phase

- Configure the MODE bit and LSB bit of the R32_ARGB_CTRL register, select the data frame structure, and enable the DMA_EN bit.
- Configure the R32_ARGB_CCRL, R32_ARGB_CCRH, R32_ARGB_DAT_ARR, and R32_ARGB_RST_ARR registers to set the physical protocol parameters.
- Configure the R32_ARGB_DAT_CYC register to set the data frame transmission length.
- Configure DMA related parameters and set the DMA transmission bit width to 1 byte.

Data transmitting phase

- Enable the TX_EN bit of the R32_ARGB_CTRL register to start data transmission.
- Query the TCIF flag bit, or wait for the TCIF and RSTIF flag bits according to the frame protocol to determine the current transmission status.
- If you need to update the display content of the light control IC, just repeat the above configuration and sending steps.

4.3 Register Description

Table 4-1 ARGB-related registers list

Name	Access address	Description	Reset value
R32_ARGB_CTLR	0x40025000	ARGB control register	0x00000000
R32_ARGB_DAT_ARR	0x40025004	ARGB 0/1 code cycle configuration register	0x00000FFF
R32_ARGB_RST_ARR	0x40025008	ARGB RESET code width configuration register	0x0001FFFF
R32_ARGB_CCRL	0x4002500C	ARGB code 0 high level T0H width configuration register	0x00000000
R32_ARGB_CCRH	0x40025010	ARGB code 1 high level T1H width configuration register	0x00000000
R32_ARGB_DAT_CYC	0x40025014	ARGB data refresh cycle length configuration register	0x00000000
R32_ARGB_DATAR	0x40025018	ARGB data register	0x00000000
R32_ARGB_STATR	0x4002501C	ARGB status register	0x00000000
R32_ARGB_CNTR	0x40025020	ARGB counter register	0x00000000

4.3.1 ARGB Control Register (R32_ARGB_CTLR)

Offset address: 0x00

31	30	29	28	27	26	25	24	23	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		TXEIE	HALFIE	BYTEIE	TCIE	RSTIE	FRHIE	Reserved				LSB	MODE	DMAEN	TE

Bit	Name	Access	Description	Reset value
[31:14]	Reserved	RO	Reserved	0
13	TXEIE	RW	Transmit buffer empty interrupt enable: 1: Enabled; 0: Disabled.	0
12	HALFIE	RW	Interrupt is enabled when half of the data refresh cycle is transmitted: 1: Enabled; 0: Disabled.	0
11	BYTEIE	RW	Byte transmission complete interrupt enable: 1: Enabled; 0: Disabled.	0
10	TCIE	RW	Transmit complete interrupt enable: 1: Enabled; 0: Disabled.	0
9	RSTIE	RW	RESET code transmission completion interrupt enable: 1: Enabled; 0: Disabled.	0
8	FRHIE	RW	Interrupt enable at the end of data refresh cycle: 1: Enabled; 0: Disabled.	0
[7:4]	Reserved	RO	Reserved	0
3	LSB	RW	Data transmission mode enable bit: 1: Low-order data is transmitted first; 0: High-order data is transmitted first.	0
2	MODE	RW	Transmission mode configuration bits: 1: Transmit the data code first, then the RESET code; 0: Transmit the RESET code first, then the data code.	0
1	DMAEN	RW	DMA enable: 1: Enabled; 0: Disabled.	0
0	TE	RW	Transmit enable: 1: Enabled;	0

			0: Disabled.	

4.3.2 ARGB 0/1 Code Cycle Configuration Register (R32_ARGB_DAT_ARR)

Offset address: 0x04

31	30	29	28	27	26	25	24	23	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				DAT_ARR[11:0]											

Bit	Name	Access	Description	Reset value
[31:12]	Reserved	RO	Reserved	0
[11:0]	DAT_ARR[11:0]	RW	0/1 code cycle, unit: system clock cycle.	0xFFF

4.3.3 ARGB RESET Code Width Configuration Register (R32_ARGB_RST_ARR)

Offset address: 0x08

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

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

Bit	Name	Access	Description	Reset value
[31:17]	Reserved	RO	Reserved.	0
[16:0]	RST_ARR[16:0]	RW	Reset code cycle, unit: system clock cycle.	0x1FFFF

4.3.4 ARGB Code 0 High Level T0H Width Configuration Register (R32_ARGB_CCRL)

Offset address: 0x0C

31	30	29	28	27	26	25	24	23	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				CCRL[11:0]											

Bit	Name	Access	Description	Reset value
[31:12]	Reserved	RO	Reserved	0
[11:0]	CCRL[11:0]	RW	Code 0 high level T0H width, unit: system clock cycle.	0

4.3.5 ARGB Code 1 High Level T1H Width Configuration Register (R32_ARGB_CCRH)

Offset address: 0x10

31	30	29	28	27	26	25	24	23	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				CCRH[11:0]											

Bit	Name	Access	Description	Reset value
[31:12]	Reserved	RO	Reserved	0
[11:0]	CCRH[11:0]	RW	Code 1 high level T1H width, unit: system clock cycle.	0

4.3.6 ARGB Data Refresh Cycle Length Configuration Register (R32_ARGB_DAT_CYC)

Offset address: 0x14

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

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

Bit	Name	Access	Description	Reset value
[31:16]	Reserved	RO	Reserved	0
[15:0]	DAT_CYC[15:0]	RW	ARGB data refresh cycle length, unit: bytes.	0

4.3.7 ARGB Data Register (R32_ARGB_DATAR)

Offset address: 0x18

31	30	29	28	27	26	25	24	23	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								DATA[7:0]							

Bit	Name	Access	Description	Reset value
[31:8]	Reserved	RO	Reserved	0
[7:0]	DATA[7:0]	RW	ARGB transmit data.	0

4.3.8 ARGB Status Register (R32_ARGB_STATR)

Offset address: 0x1C

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved					BIT_CNT[2:0]			STATE[1:0	TX	HA	BY	TCI	RS	FR	

		]	EIF	LFIF	TEIF	F	TIF	HIF

Bit	Name	Access	Description	Reset value
[31:16]	BYTE_CNT[15:0]	RO	Transmit byte counter.	0
[15:11]	Reserved	RO	Reserved	0
[10:8]	BIT_CNT[2:0]	RO	Transmit bit counter.	0
[7:6]	STATE[1:0]	RO	State machine indication signal: 00: Idle; 01: Data sending phase; 10: RESET code transmitting phase; 11: Reserved.	0
5	TXEIF	RO	Transmit buffer empty interrupt flag, writing DATAR will clear this flag. *Note: 1 bit in this position indicates that the content in RDATA has been fetched into the shift register, the RDATA register has been released, and new data can be written.*	0
4	HALFIF	RW1	The interrupt flag is transmitted halfway through the data refresh cycle and cleared by writing 1. *Note: When this bit is set, it means that the number of bytes sent in this round of data transmitting cycle has reached half of DAT_CYC.*	0
3	BYTEIF	RW1	Byte transmission completion interrupt flag, clear by writing 1. *Note: When this bit is set, it means that the contents of the shift register have been transmitted.*	0
2	TCIF	RW1	Send completion interrupt flag, clear by writing 1. *Note: When this bit is set, it means that the number of bytes transmitted in this round of data sending cycle has reached the number set by DAT_CYC.*	0
1	RSTIF	RW1	RESET code transmission completion interrupt flag, clear by writing 1. *Note: Setting this bit indicates the completion of the RST phase of the current cycle.*	0
0	FRHIF	RW1	The interrupt flag after the data refresh cycle is sent is cleared by writing 1. *Note: This position indicates the end of the current cycle.*	0

4.3.9 ARGB Counter Register (R32_ARGB_CNTR)

Offset address: 0x20

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

Reserved	CNT[16]

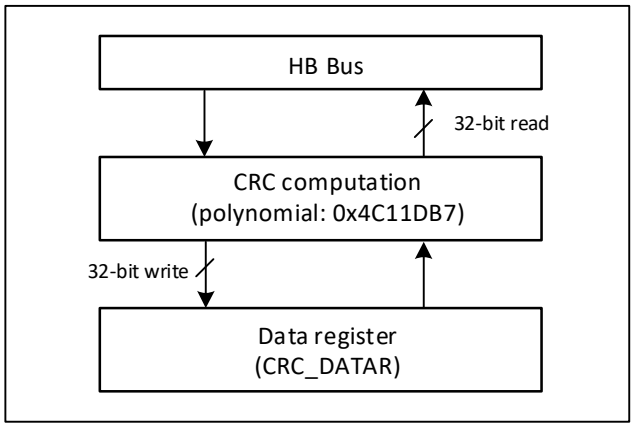
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNT[15:0]															

Bit	Name	Access	Description	Reset value
[31:17]	Reserved	RO	Reserved	0
[16:0]	CNT[16:0]	RO	ARGB base counter.	0

Chapter 5 Cyclic Redundancy Check (CRC)

The cyclic redundancy check (CRC) computation unit is used to obtain the result of the CRC computation for any 32-bit data based on a fixed generating polynomial. It is generally used in the field of data storage and data communication to verify the correctness of data. The system provides hardware CRC calculation unit which can greatly save CPU and RAM resources to improve efficiency.

Figure 5-1 CRC structure diagram



5.1 Main Features

- CRC32 polynomial (0x4C11DB7): $X^{32}+X^{26}+X^{23}+X^{22}+X^{16}+X^{12}+X^{11}+X^{10}+X^{8}+X^{7}+X^{5}+X^{4}+X^{2}+X+1$;
- Same 32-bit register as input for data and output for CRC32 calculation
- Single conversion time: 4 HB clock cycles (HCLK)

5.2 Function Description

- CRC unit reset

To start a CRC calculation for a new data set, the CRC calculation unit needs to be reset. Writing a 1 to the RST bit of the control register CRC_CTLR will reset the data register by the hardware, restoring the initial value 0xFFFFFFFF.

- CRC calculation

The CRC unit calculates the CRC result of the previous CRC calculation and the CRC result of the newly involved data. The CRC_DATAR data register, for which a write operation will feed new data to the hardware calculation unit; and a read operation will get the value of the latest round of CRC calculation. The hardware calculation interrupts the system write operation, so new values can be written continuously.

Note: The CRC unit calculates the entire 32-bit data, not byte-by-byte.

- Independent data buffer

The CRC unit provides an 8-bit independent data register, CRC_IDATAR, which is used for the application code to temporarily store 1 byte of data independent of the CRC unit reset.

5.3 Register Description

Table 5-1 CRC-related registers list

Name	Access address	Description	Reset value
R32_CRC_DATAR	0x40023000	Data register	0xFFFFFFFF
R8_CRC_IDATAR	0x40023004	Independent data buffer	0x00
R32_CRC_CTLR	0x40023008	Control register	0x00000000

5.3.1 Data Register (CRC_DATAR)

Offset address: 0x00

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

Bit	Name	Access	Description	Reset value
[31:0]	DR[31:0]	RW	Write raw data; read calculations.	0xFFFFFFFF

5.3.2 Independent Data Buffer (CRC_IDATAR)

Offset address: 0x04

7	6	5	4	3	2	1	0
IDR[7:0]							

Bit	Name	Access	Description	Reset value
[7:0]	IDR[7:0]	RW	An 8-bit general-purpose register that can be used as a data cache, this register is not affected by the RST field of the control register.	0

5.3.3 Control Register (CRC_CTLR)

Offset address: 0x08

31	30	29	28	27	26	25	24	23	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															RST

Bit	Name	Access	Description	Reset value
[31:1]	Reserved	RO	Reserved	0
0	RST	WO	CRC calculation unit reset control, write 1	0

			execution, hardware auto clear, after execution, data register is 0xFFFFFFFF.	

Chapter 6 Real Time Clock (RTC)

The Real Time Clock (RTC) is a standalone timer module with a programmable counter up to 32 bits, which can be used with software to realize the real time clock function, and the counter value can be modified to reconfigure the current time and date of the system.

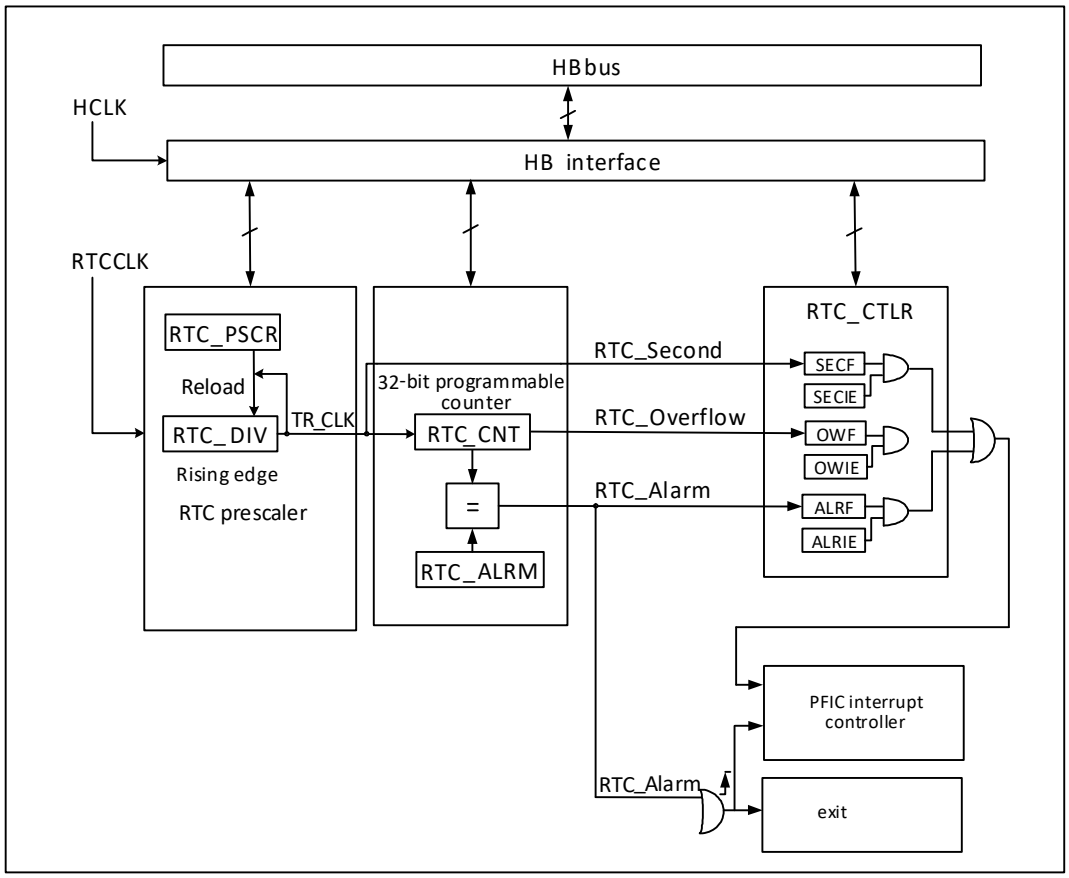
6.1 Main Features

- Prescaler coefficients up to 2^{20}
- 32-bit programmable counter
- Multiple clock sources, interrupts
- Independent reset

6.2 Function Description

6.2.1 Overview

Figure 6-1 RTC structure diagram



As shown in Figure 6-1, the RTC module is mainly composed of three parts: HB bus interface, frequency divider and counter, and control and status register. After RTCCK is input into the divider (RTC_DIV), it is divided into TR_CLK. It is worth noting that the internal part of the frequency divider (RTC_DIV) is a self-decreasing counter. When it decrements to overflow, it will output a TR_CLK, and then take the preset value from the reload value register (RTC_PSCR) and reload it into the frequency divider. Reading the frequency divider actually reads its real-

time value (read only), and writing the frequency division coefficient should be written to the reload value register (RTC_PSCR). Generally, the period of TR_CLK is set to 1 second. TR_CLK will trigger the second event and at the same time cause the main counter (RTC_CNT) to increase by 1. When the main counter increases to be consistent with the value of the alarm register, the alarm event will be triggered. When the main counter increments to overflow, an overflow event will be triggered. The above 3 events can trigger interrupts and are controlled by corresponding interrupt enable bits.

6.2.2 Special Read/Write Register Operation

Due to the special use of the real-time clock, RTC and HB buses are independent, and the reading of RTC by HB is not necessarily real-time. Reading the register of RTC through HB must be after HB startup and passing through a RTC rising edge. This situation may occur after system reset and power reset, after waking up from Stop mode. It is convenient to wait for the RSF bit of the control register (CTLR) to be set high. The write operator for RTC must wait for the end of the last write operation and must enter configuration mode. The specific steps are as follows:

1) Query the RTOFF bit until it becomes 1.
2) Set the CNF bit and enter the configuration mode.
3) Write to one or more RTC registers.
4) Clear the CNF bit, exit the configuration mode, and the HB interface starts to write the RTC register.
5) Query the RTOFF bit until it becomes 1.

6.3 Register Description

Table 6-1 RTC-related registers list

Name	Access address	Description	Reset value
R16_RTC_CTLRH	0x40002800	RTC control register high	0x0000
R16_RTC_CTLRL	0x40002804	RTC control register low	0x0020
R16_RTC_PSCRH	0x40002808	Prescaler reload register high	0x000X
R16_RTC_PSCRL	0x4000280C	Prescaler reload register low	0xXXXX
R16_RTC_DIVH	0x40002810	Divider register high	0x000X
R16_RTC_DIVL	0x40002814	Divider register low	0xXXXX
R16_RTC_CNTH	0x40002818	RTC counter register high	0xXXXX
R16_RTC_CNTL	0x4000281C	RTC counter register low	0xXXXX
R16_RTC_ALRMH	0x40002820	Alarm clock register high	0xXXXX
R16_RTC_ALRML	0x40002824	Alarm clock register low	0xXXXX

6.3.1 RTC Control Register High (RTC_CTLRH)

Offset address: 0x00

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

Bit	Name	Access	Description	Reset value
[15:3]	Reserved	RO	Reserved	0
2	OWIE	RW	Overflow interrupt enable: 1: Enable; 0: Disable.	0
1	ALRIE	RW	Alarm interrupt enable: 1: Enable; 0: Disable.	0
0	SECIE	RW	Second interrupt enable: 1: Enable; 0: Disable.	0

6.3.2 RTC Control Register Low (RTC_CTLRL)

Offset address: 0x04

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved														RTCCAL[6:5]	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RTCCAL[4:0]					ASOS	ASOE	CCO	RTCSEL	RTCEN	RTOFF	CNF	RSF	OWF	ALRF	SECF

Bit	Name	Access	Description	Reset value
[31:18]	Reserved	RO	Reserved	0
[17:11]	RTCCAL[6:0]	RW	RTC calibration value register, the value of this register indicates how many of every 2^{20} clock pulses are skipped. This function is used to calibrate the RTC clock. The RTC clock can be slowed down by 0~121ppm.	0
10	ASOS	RW	PC8 pin alarm/second pulse output selection: 1: Output second pulse; 0: Output alarm pulse.	0
9	ASOE	RW	PC8 pin enable pulse output bit: 1: Enable the output of alarm pulse or second pulse; 0: Disable the output of alarm pulse or second pulse.	0
8	CCO	RW	Calibration clock output selection bit: 1: PC8 pin outputs the RTC clock divided by 64; 0: Does not output the calibration clock.	0
7	RTCSEL	RW	RTC clock source selection: 1: HSE is used as the RTC clock after being divided by 512; 0: LSI is used as the RTC clock.	0
6	RTCEN	RW	RTC clock enable control: 1: Enable RTC clock;	0

			0: Turn off RTC clock.	
5	RTOFF	RO	RTC operation status indication bit, indicates the execution status of the last operation to the RTC, the operation to the RTC must wait for this bit to be 1. 1: The last operation on the RTC has been completed; 0: The last operation on the RTC is still in progress.	1
4	CNF	RW	Configuration flag bit, writing this bit by 1 enters the configuration mode, thereby allowing values to be written to the Counter (R16_RTC_CNTx), the Alarm Clock Register (R16_RTC_ALRMx) and the Prescaler Reload Value Register (R16_RTC_PSCRx). The write operation is performed only after the bit is written 1 and re-cleared 0 by the software: 1: Enter configuration mode; 0: Exit configuration mode and start updating the RTC registers.	0
3	RSF	RW0	Register synchronization flag bit, before reading and writing registers such as pre-division frequency (PSCRx), alarm clock (ALRMx) and counter (CNTx) of RTC module, make sure that this bit has been set by hardware to make sure that these registers have been synchronized; when reading and writing these registers, or after PB1 reset or PB1 clock stops, the first step should reset this bit. 1: The register is synchronized; 0: The register is not synchronized.	0
2	OWF	RW0	Counter overflow flag, this bit is set by hardware when the 32-bit counter overflows. An overflow interrupt is also generated if the OWIE bit is set. This bit can only be cleared by software and cannot be set by software.	0
1	ALRF	RW0	Alarm Clock Flag, this bit is set by hardware when the counter value reaches the value of the Alarm Clock Register (ALRMx), and an alarm clock interrupt is also generated if the Alarm Clock Interrupt Enable Bit (ALRIE) is set. This bit can only be cleared by software and cannot be set by software.	0
0	SECF	RW0	The second event flag, when the clock is divided by the prescaler after each falling edge, will cause the counter to increment by one, and at the same time generate a second event, this bit will be set, if the second interrupt is enabled (SECIE is set), and at the same time will also generate a second interrupt. This bit can only be cleared by software and cannot be set	0

			by software.	

6.3.3 Prescaler Reload Register High (RTC_PSCRH)

Offset address: 0x08

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved												PRL[19:16]			

Bit	Name	Access	Description	Reset value
[15:4]	Reserved	RO	Reserved	0
[3:0]	PRL[19:16]	WO	Reload value high	x

6.3.4 Prescaler Reload Register Low (RTC_PSCRL)

Offset address: 0x0C

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

Bit	Name	Access	Description	Reset value
[15:0]	PRL[15:0]	WO	The reloading value is low. The actual frequency division factor is (PRLL[19:0] + 1). If the RTC input frequency is 32768Hz, then this value is set to 0x7FFF, you can divide the signal with a period of 1 second.	X

6.3.5 Divider Register High (RTC_DIVH)

Offset address: 0x10

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved												DIV[19:16]			

Bit	Name	Access	Description	Reset value
[15:4]	Reserved	RO	Reserved	0
[3:0]	DIV[19:16]	RO	Divider register high.	x

6.3.6 Divider Register Low (RTC_DIVL)

Offset address: 0x14

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

Bit	Name	Access	Description	Reset value
[15:0]	DIV[15:0]	RO	Divider register low. DIV is actually a self-subtractive counter. Every time a clock comes to RTC_CLK, the DIV counter is subtracted by 1.	X

			After the overflow, a TR_CLK is output and the value is reloaded from the PSCR. DIV can only read and read the remaining value of the counter of the current frequency divider.	

6.3.7 RTC Counter High (RTC_CNTH)

Offset address: 0x18

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

Bit	Name	Access	Description	Reset value
[15:0]	CNT[31:16]	RW	Counter High	X

6.3.8 RTC Counter Low (RTC_CNTL)

Offset address: 0x1C

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

Bit	Name	Access	Description	Reset value
[15:0]	CNT[15:0]	RW	Counter low, the core device of the RTC timer, the clock is provided by the TRCLK (the period is generally set to 1 second). Calculate the current time by reading CNT[31:0]. To write this value, you need to enter configuration mode.	X

6.3.9 Alarm Register High (RTC_ALRMH)

Offset address: 0x20

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

Bit	Name	Access	Description	Reset value
[15:0]	ALR[31:16]	WO	Alarm register high	X

6.3.10 Alarm Register Low (RTC_ALRML)

Offset address: 0x24

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

Bit	Name	Access	Description	Reset value
[15:0]	ALR[15:0]	WO	Alarm clock register is low. An alarm clock event occurs when the value of the alarm clock register	X

			ALRM[31:0] is the same as that of the counter CNT[31:0]. Changing this value requires you to enter configuration mode.	

Chapter 7 Independent Watchdog (IWDG)

The system is equipped with an independent watchdog (IWDG) to detect software failures caused by logic errors and external environment interference. The IWDG clock source comes from LSI and can be run independently of the main program, so it is suitable for situations with low precision requirements.

7.1 Main Features

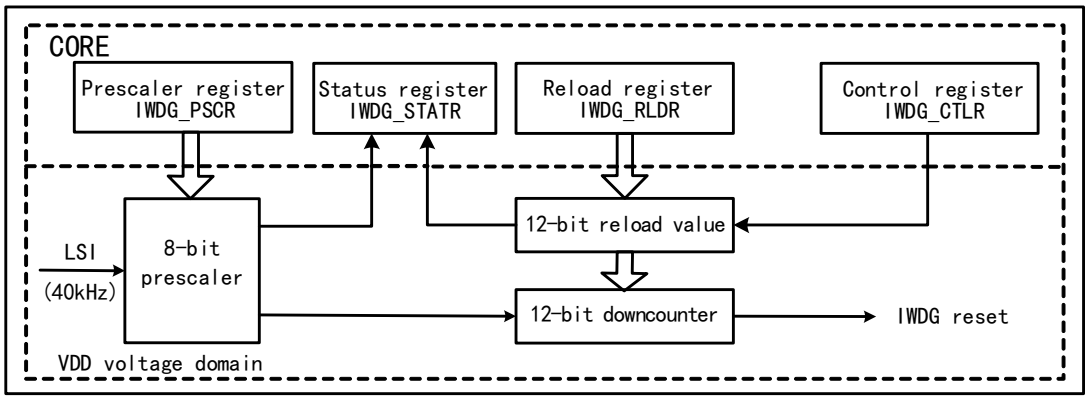
- 12-bit self-subtractive counter.
- Clock source LSI frequency division, can run in low-power mode.
- Reset condition: counter value reduced to 0

7.2 Functional Description

7.2.1 Principle and Application

The stand-alone watchdog is clocked from the LSI clock and functions in shutdown and standby modes. When the watchdog counter decrements itself to 0, a system reset will be generated, so the timeout is (reload value + 1) clock.

Figure 7-1 Structure block diagram of Independent Watchdog



- Enable independent watchdog

After the system reset, the watchdog is OFF. Write 0xCCCC to the IWDG_CTLR register to enable the watchdog, and then it can no longer be disabled unless a reset occurs.

If the hardware independent watchdog enable bit (IWDGSW) is enabled in user option bytes, the IWDG is permanently enabled after the microcontroller reset.

- Watchdog configuration

Inside the watchdog is a 12-bit counter running progressively. When the value of the counter is reduced to 0, a system reset will occur. To enable the IWDG function, you need to perform the following actions:

1) Count time base: IWDG clock source LSI, through the IWDG_PSCR register to set the LSI frequency division value clock as the IWDG count time base. The operation method first writes 0x5555 to the IWDG_CTLR register, and then modifies the frequency division value in the IWDG_PSCR register. The PVU bit in the IWDG_STATR

register indicates the update status of the frequency division value, and the frequency division value can only be modified and read out when the update is completed.

2) Reload value: used to update the current value of the counter in the independent watchdog, and the counter is decremented by this value. The operation method first writes 0x5555 to the IWDG_CTLR register, and then modifies the IWDG_RLDR register to set the target reload value. The RUV bit in the IWDG_STATR register indicates the update status of the reload value, and the IWDG_RLDR register can be modified and read out only after the update is completed.

3) Watchdog enable: write 0xCCCC to the IWDG_CTLR register to turn on the watchdog function.

4) Feeding the dog: that is, before the watchdog counter decreases to 0, refresh the current counter value to prevent system reset. Write 0xAAAA to the IWDG_CTLR register and have the hardware update the IWDG_RLDR register value to the watchdog counter. This action needs to be performed regularly after the watchdog function is turned on, otherwise the watchdog reset action will occur.

7.2.2 Debug Mode

When the system enters the debug mode, the IWDG counter can be configured by the debug module register to continue working or stop.

7.3 Register Description

Table 7-1 IWDG-related registers

Name	Access address	Description	Reset value
R16_IWDG_CTLR	0x40003000	Control register	0x0000
R16_IWDG_PSCR	0x40003004	Prescaler register	0x0000
R16_IWDG_RLDR	0x40003008	Reload register	0x0FFF
R16_IWDG_STATR	0x4000300C	Status register	0x0000

7.3.1 IWDG Control Register (IWDG_CTLR)

Offset address: 0x00

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

Bit	Name	Access	Description	Reset value
[15:0]	KEY[15:0]	WO	Operate the key value lock. 0xAAAA: Feed the dog. Load the IWDG_RLDR register value into the independent watchdog counter. 0x5555: Allow modification of R16_IWDG_PSCR and R16_IWDG_RLDR registers. 0xCCCC: Start the watchdog, which is not subject to this restriction if the hardware watchdog is enabled (user option byte configuration).	0

7.3.2 Prescaler Register (IWDG_PSCR)

Offset address: 0x04

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

Bit	Name	Access	Description	Reset value
[15:3]	Reserved	RO	Reserved	0
[2:0]	PR[2:0]	RW	IWDG clock division factor, write 0x5555 to KEY before modifying this field. 000: Divided by 4; 001: Divided by 8; 010: Divided by 16; 011: Divided by 32; 100: Divided by 64; 101: Divided by 128; 110: Divided by 256; 111: Divided by 256. IWDG count time base = LSI/division factor. *Note: Before reading the value of this field, make sure that the PVU bit in the IWDG_STATR register is 0, otherwise the read value is invalid.*	000b

7.3.3 Reload Value Register (IWDG_RLDR)

Offset address: 0x08

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

Bit	Name	Access	Description	Reset value
[15:12]	Reserved	RO	Reserved	0
[11:0]	RL[11:0]	RW	Counter reload value. Write 0x5555 to KEY before modifying this field. When 0xAAAA is written to KEY, the value of this field is loaded into the counter by the hardware, and the counter is then decremented from that value. *Note: Before reading and writing the field value, make sure that the RUV bit in the IWDG_STATR register is 0, otherwise it is invalid to read and write this field.*	0xFFF

7.3.4 Status Register (IWDG_STATR)

Offset address: 0x0C

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

Bit	Name	Access	Description	Reset value
[15:2]	Reserved	RO	Reserved	0
1	RVU	RO	The reinstall value updates the flag bit. Hardware setting or clearing 0. 1: Reload value update is in progress. 0: Reload update ends (up to 5 LSI cycles). *Note: The reload value register IWDG_RLDR can be read and written only after the RVU bit has been cleared.*	0
0	PVU	RO	Clock frequency division coefficient updates flag bits. Hardware setting or clearing 0. 1: Clock division value update is in progress. 0: Clock division value update ends (up to 5 LSI cycles). *Note: The frequency division factor register IWDG_PSCR can be read and written only after the PVU bit has been cleared.*	0

Note: After the pre-division or reinstallation value is updated, you do not have to wait for RVU or PVU to reset, you can continue to execute the following code. (This write operation continues to be completed even in low power mode.)

Chapter 8 Window Watchdog (WWDG)

The window watchdog is generally used to monitor the software failures of the system, such as external interference, unforeseen logic errors and so on. It needs to refresh the counter (feed the dog) within a specific window time (with upper and lower limits), otherwise the watchdog circuit will produce a system reset earlier or later than this window time.

8.1 Main Features

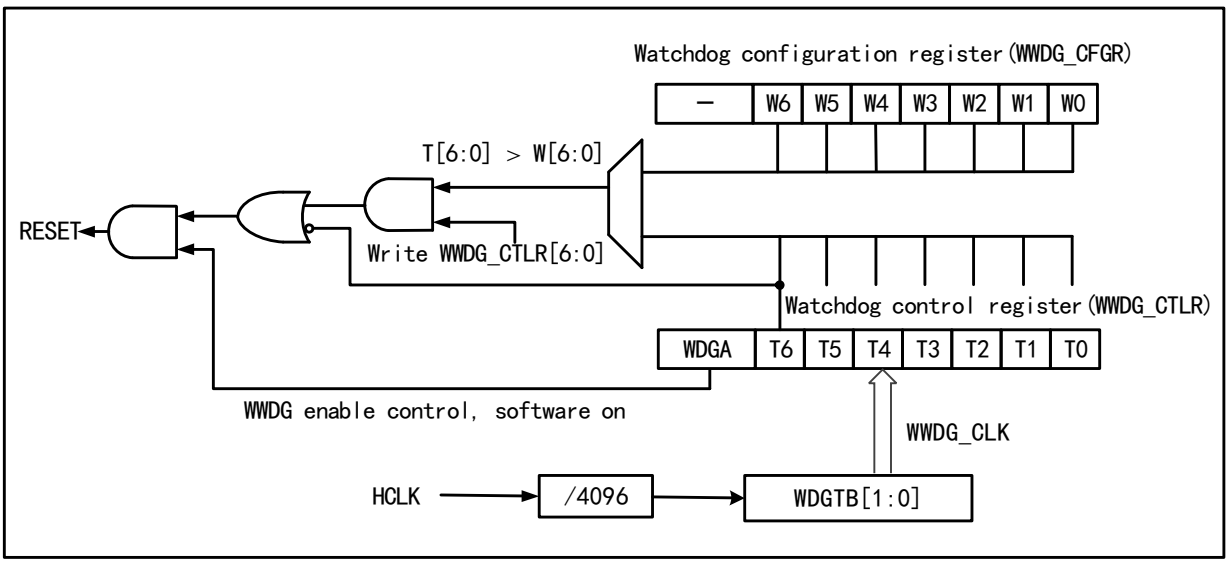
- Programmable 7-bit self-subtractive counter.
- Double conditional reset: the current counter value is less than 0x40, or the counter value is reloaded outside the window time.
- Wake up advance notice function (EWI), which is used to feed the dog in time to prevent system reset

8.2 Function Description

8.2.1 Principle and Application

The window watchdog runs based on a 7-bit decrement counter, which is mounted on the HB bus to count the frequency division of the time-based WWDG_CLK source (HCLK/4096) clock, and the frequency division factor is set in the WDGTB[1:0] field in the configuration register WWDG_CFGR. The decrement counter is in a state of free operation, regardless of whether the watchdog function is turned on or not, the counter has been cyclically decreasing counting. As shown in figure 8-1, the internal structure block diagram of the window watchdog.

Figure 8-1 Window Watchdog structure diagram



- Enable window watchdog

After the system reset, the watchdog is disabled. Set the WDGA bit in the WWDG_CTLR register to switch on the watchdog, and then it can no longer be disabled unless a reset occurs.

Note: WWDG clock source can be disabled by setting the RCC_HB1PCENR register to suspend WWDG_CLK counting and indirectly stop the watchdog function. Or reset the WWDG module by setting the RCC_HB1PRSTR register, which is equivalent to the function of reset.

- Watchdog configuration

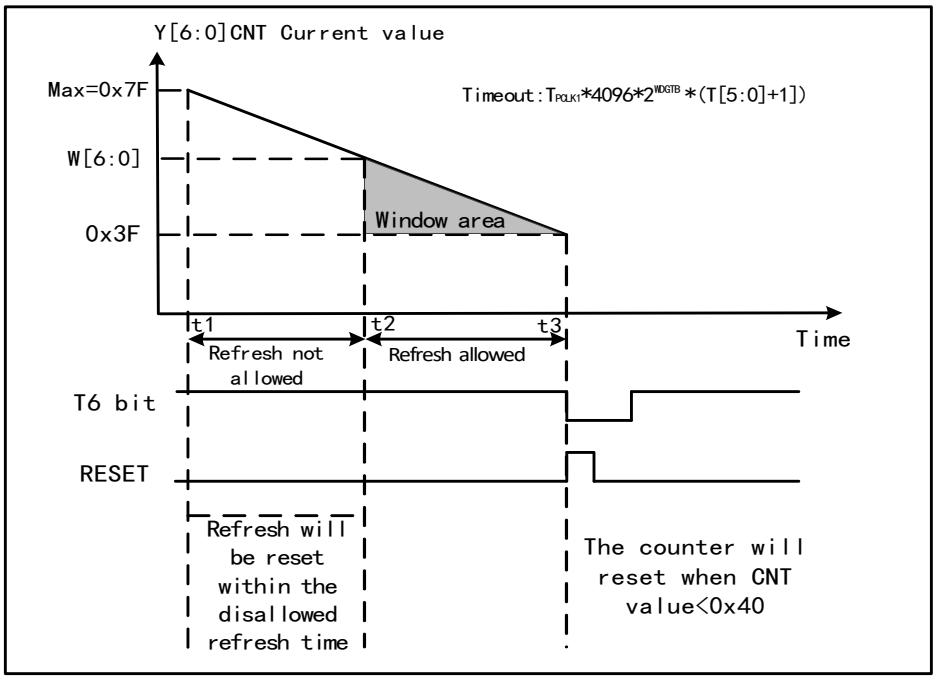
Inside the watchdog is a 7-bit counter running in a continuous cycle, which supports read and write access. To use the watchdog reset function, you need to perform the following actions:

1) Counter time base: The WDGTB[1:0] bits in the WWDG_CFGR register. Note to switch on the WWDG module clock of the RCC unit.
2) Window counter: Set the W[6:0] bits in the WWDG_CFGR register. This counter is used to be compared with the current counter by hardware, the value is configured by the user software and will not change. It serves as the maximum value of window time.
3) Watchdog enable: The WDGA bit in the WWDG_CTLR register is set to 1 by software, and the watchdog function is enabled to reset the system.
4) Feed dog: Refresh the current counter value and configure the T[6:0] bits in the WWDG_CTLR register. This action needs to be executed in the periodic window time after the watchdog function is enabled. Otherwise, the watchdog reset action occurs.

- Feed dog window time

As shown in Figure 8-2, the gray area is the detector window area of the window watchdog. Its maximum timeout (t_2) corresponds to the time point when the current counter value reaches the window value W[6:0]. Its minimum timeout (t_3) corresponds to the time point when the current counter value reaches 0x3F. Within this area time ($t_2<t<t_3$), the feed dog operation can be performed (write T[6:0]) to refresh the current counter value.

Figure 8-2 Counting mode of window watchdog



- Watchdog reset:

1) When the feed dog operation is not performed in time, the value of the T[6:0] counter changes from 0x40 to 0x3F, a "Window Watchdog Reset" occurs, and a system reset occurs. I.e., when T6-bit is detected as 0 by hardware, the system reset occurs.

Note: The application program can write 0 to the T6-bit by software to implement system reset, which is equivalent to software reset function.

2) When the counter refresh action is executed when the feed dog operation is disabled, i.e., when write operation is performed on the T[6:0] bits when $t_1 \leq t \leq t_2$, a "Window Watchdog Reset" occurs, and a system reset occurs.

- Early wake-up

To prevent the system from resetting due to failure to refresh the counter in time, the watchdog module provides early wake-up interrupt (EWI) notification. When the counter is reduced to 0x40, an early wake-up signal is generated, and the WEIF flag is set to 1. If the EWI bit is set, the window watchdog will be triggered to interrupt at the same time. At this point, there is a counter clock cycle (self-reduced to 0x3F) from the hardware reset, during which the application can immediately feed the dog.

8.2.2 Debug Mode

When the system enters debug mode, the WWDG counter can either continues to work normally or stops, depending on the debugging module register.

8.3 Register Description

Table 8-1 WWDG-related registers

Name	Access address	Description	Reset value
R16_WWDG_CTLR	0x40002C00	Control register	0x007F
R16_WWDG_CFGR	0x40002C04	Configuration register	0x007F
R16_WWDG_STATR	0x40002C08	Status register	0x0000

8.3.1 WWDG Control Register (WWDG_CTLR)

Offset address: 0x00

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

Bit	Name	Access	Description	Reset value
[15:8]	Reserved	RO	Reserved	0
7	WDGA	RW1	The window watchdog resets the enable position. 1: Enable the watchdog function (can generate reset signal). 0: Disable the watchdog function. Software write 1 is on, but only hardware is allowed to clear 0 after reset.	0
[6:0]	T[6:0]	RW	7-bit self-subtractive counter, minus 1 per $4096*2^{WDGTB}$ HCLK cycle. When the counter is reduced from 0x40 to 0x3F, that is, when T6 jumps to 0, a watchdog reset is generated.	0x7F

8.3.2 WWDG Configuration Register (WWDG_CFGR)

Offset address: 0x04

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

Bit	Name	Access	Description	Reset value
[15:10]	Reserved	RO	Reserved	0
9	EWI	RW1	Early wakeup interrupt: If it set to 1, interrupt is generated when the counter reaches 0x40. It can only be cleared by hardware after reset.	0
[8:7]	WDGTB[1:0]	RW	Window watchdog time base: 00: Divided by 1, counter time base =HCLK/4096; 01: Divided by 2, counter time base =HCLK/4096/2; 10: Divided by 4, counter time base = HCLK/4096/4; 11: Divided by 8, counter time base = HCLK/4096/8.	0
[6:0]	W[6:0]	RW	Window watchdog 7-bit window value. It is used to be compared with the counter value. The feed dog operation can be performed only when the counter value is less than the window value and is greater than 0x3F.	0x7F

8.3.3 WWDG Status Register (WWDG_STATR)

Offset address: 0x08

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

Bit	Name	Access	Description	Reset value
[15:1]	Reserved	WO	Reserved	0
0	EWIF	RW0	Early wakeup interrupt flag. When the counter reaches 0x40, this bit is set by hardware, and it must be cleared by software. User setting is invalid. Even if the EWI is not set, this bit is still set as usual when the event occurs.	0

Chapter 9 Interrupt and Events (PFIC)

The built-in Programmable Fast Interrupt Controller (PFIC) supports up to 256 interrupt vectors. The current system manages 50 peripheral interrupt channels and 16 core interrupt channels, and the rest are retained.

9.1 Main Features

9.1.1 PFIC

- Each interrupt request has an independent trigger and mask control bit, and a dedicated status bit
- Programmable multi-level interrupt nesting, maximum nesting depth 2 levels
- Dedicated fast interrupt in and out mechanism, hardware automatic stack and recovery, maximum hardware stack depth level 2, no instruction overhead
- Vector Table Free (VTF) interrupt response mechanism, 4-channel programmable direct interrupt vector address

9.2 SysTick

The core comes with a 32-bit addition and subtraction counter (SysTick), which supports HCLK or HCLK/8 as a time base, has a higher priority, and can be used as a time benchmark after calibration.

9.3 Vector Table of Interrupt and Exception

Table 9-1 Vector table

No.	Priority	Type	Name	Description	Entrance address
0	-6	fixed	RESET	Reset interrupt	0x00000000
1	-	-	-	-	0x00000004
2	-5	fixed	NMI	Non-maskable interrupt	0x00000008
3	-4	fixed	HardFault	Abnormal interruption	0x0000000C
4	-	-	-	-	0x00000010
5	-3	fixed	Ecall-M	Machine mode callback interrupt	0x00000014
6-7	-	-	-	-	0x00000018- 0x0000001C
8	-2	fixed	Ecall-U	User mode callback interrupt	0x00000020
9	-1	fixed	BreakPoint	Breakpoint callback interrupt	0x00000024
10-11	-	-	-	-	0x00000028- 0x0000002C
12	0	programmable	SysTick	SysTick interrupt	0x00000030
13	-	-	-	-	0x00000034
14	1	programmable	SW	Software interrupt	0x00000038
15	-	-	-	-	0x0000003C
16	2	programmable	WWDG	Window watchdog timer interrupt	0x00000040
17	3	programmable	EXTI15_8	EXTI line [15:8] interrupt	0x00000044
18	4	programmable	FLASH	FLASH global interrupt	0x00000048

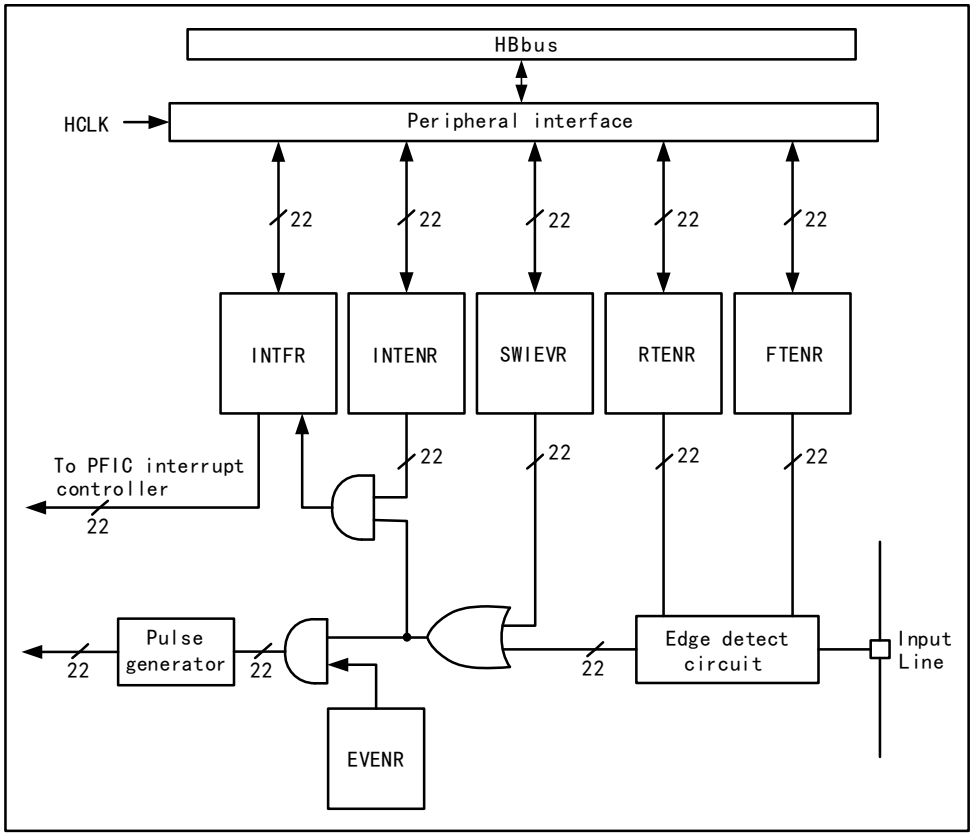
19	5	programmable	RCC	Reset and clock control interrupts	0x0000004C
20	6	programmable	EXTI7_0	EXTI line [7:0] interrupt	0x00000050
21	7	programmable	SPI2	SPI2 global interrupt	0x00000054
22	8	programmable	DMA1_CH1	DMA channel 1 global interrupt	0x00000058
23	9	programmable	DMA1_CH2	DMA channel 2 global interrupt	0x0000005C
24	10	programmable	DMA1_CH3	DMA channel 3 global interrupt	0x00000060
25	11	programmable	DMA1_CH4	DMA channel 4 global interrupt	0x00000064
26	12	programmable	DMA1_CH5	DMA channel 5 global interrupt	0x0000006C
27	13	programmable	DMA1_CH6	DMA channel 6 global interrupt	0x00000070
28	14	programmable	DMA1_CH7	DMA channel 7 global interrupt	0x00000074
29	15	programmable	DMA1_CH8	DMA channel 8 global interrupt	0x00000078
30	16	programmable	DMA1_CH9	DMA channel 9 global interrupt	0x0000007C
31	17	programmable	DMA1_CH10	DMA channel 10 global interrupt	0x00000080
32	18	programmable	DMA1_CH11	DMA channel 11 global interrupt	0x00000084
33	19	programmable	USART2	USART2 global interrupt	0x00000088
34	20	programmable	I2C1_EV	I^2C1 event interrupt	0x0000008C
35	21	programmable	I2C1_ER	I^2C1 error interrupt	0x00000090
36	22	programmable	USART1	USART1 global interrupt	0x00000094
37	23	programmable	SPI1	SPI1 global interrupt	0x00000098
38	24	programmable	TIM1_UP	TIM1 update interrupt	0x0000009C
39	25	programmable	TIM1_CC	TIM1 capture compare interrupt	0x000000A0
40	26	programmable	TIM1_TRG_COM	TIM1 trigger and communication interruption	0x000000A4
41	27	programmable	TIM1_BRK	TIM1 brake interrupt	0x000000A8
42	28	programmable	TIM2	TIM2 global interrupt	0x000000AC
43	29	programmable	I2C2_EV	I^2C2 event interrupt	0x000000B0
44	30	programmable	I2C2_ER	I^2C2 error interrupt	0x000000B4
45	31	programmable	USBPD	USBPD global interrupt	0x000000B8
46	32	programmable	USBPDWakeUP	USBPD wake-up interrupt	0x000000BC
47	33	programmable	ARGB	ARGB global interrupt	0x000000C0
48	34	programmable	TIM3	TIM3 global interrupt	0x000000C4
49	35	programmable	TIM4	TIM4 global interrupt	0x000000C8
50	36	programmable	SPI3	SPI3 global interrupt	0x000000CC
51	37	programmable	USART3	USART3 global interrupt	0x000000D0
52	38	programmable	USART4	USART4 global interrupt	0x000000D4
53	39	programmable	USBSS	USBSS global interrupt	0x000000D8
54	40	programmable	USBSS_LINK	USBSS LINK global interrupt	0x000000DC
55	41	programmable	USBHS	USBHS global interrupt	0x000000E0
56	42	programmable	USBHSWakeUP	USBHS wake-up interrupt	0x000000E4
57	43	programmable	USBSSWakeUP	USBSS wake-up interrupt	0x000000E8

58	44	programmable	PVD	Supply voltage detection interrupt	0x000000EC
59	45	programmable	ADC1	ADC1 global interrupt	0x000000F0
60	46	programmable	ADC2	ADC2 global interrupt	0x000000F4
61	47	programmable	ADC3	ADC3 global interrupt	0x000000F8
62	48	programmable	RTCAlarm	RTC alarm interrupt	0x000000FC
63	49	programmable	RTC	Real time clock interrupt	0x00000100
64	50	programmable	USARTWakeUP	USART wake-up interrupt	0x00000104
65	51	programmable	ADC4	ADC4 global interrupt	0x00000108

9.4 External Interrupt and Event Controller (EXTI)

9.4.1 Overview

Figure 9-1 External interrupt (EXTI) interface block diagram



As can be seen from figure 9-1, the trigger source of the external interrupt can be either a software interrupt (SWIEVR) or an actual external interrupt channel, and the signal of the external interrupt channel is first screened by the edge detection circuit. As long as one of the software interrupts or external interrupt signals is generated, it will be output to both event enabling and interrupt enabling circuits through the OR gate circuit in the diagram. As long as an interrupt is enabled or an event is enabled, an interrupt or event will occur. The six registers of EXTI are accessed by the processor through the HB interface.

9.4.2 Wakeup Event

The system can wake up sleep patterns caused by WFE instructions through wake-up events. Wake-up events are

generated through the following 2 configurations:

- Enable an interrupt in the register of the peripheral, but not in the PFIC of the core, as well as the SEVONPEND bit in the core. Reflected in EXTI, it enables EXTI interrupts, but does not enable EXTI interrupts in PFIC, while enabling SEVONPEND bits. When CPU wakes up from WFE, the interrupt flag bit and PFIC hang bit of EXTI need to be cleared.
- Enable an EXTI channel to be an event channel, and after CPU wakes up from WFE, there is no need to clear the operation of the interrupt flag bit and PFIC hang bit.

9.4.3 Description

The use of external interrupt needs to configure the corresponding external interrupt channel, that is, select the appropriate trigger edge to enable the corresponding interrupt. When a set trigger edge appears on the external interrupt channel, an interrupt request will be generated and the corresponding interrupt flag bit will be set. Write 1 to the flag bit to clear it.

Use external hardware interrupt steps:

1) Configure GPIO operation.

2) Configure the interrupt enable bit (EXTI_INTENR) of the corresponding external interrupt channel.

3) Configure trigger edge (EXTI_RTENR or EXTI_FTENR), select rising edge trigger, falling edge trigger or double edge trigger.

4) Configure EXTI interrupts in the PFIC of the core to ensure that they respond correctly.

To use external hardware events:

1) Configure GPIO operation.

2) Configure the event enable bit (EXTI_EVENR) of the corresponding external interrupt channel.

3) Configure trigger edge (EXTI_RTENR or EXTI_FTENR), select rising edge trigger, falling edge trigger or double edge trigger.

Use software interrupt/event steps:

1) Enable external interrupt (EXTI_INTENR) or external event (EXTI_EVENR).

2) If you use the interrupt service function, you need to set the EXTI interrupt in the PFIC of the core.

3) Set the software interrupt trigger (EXTI_SWIEVR), that is, an interrupt will occur.

9.4.4 External Event Map

Table 9-2 EXTI interrupt map

External interrupt/event line	Mapping event description
EXTI0~EXTI15	Px0~Px15 (x=A/B/C/D), any of the IO ports can be enabled for external interrupt/event functionality, configured by the AFIO_EXTICRx register.
EXTI16	PVD event: Exceed the voltage detector threshold
EXTI17	USBHS wakeup event
EXTI18	USBPD wakeup event
EXTI19	USBSS wakeup event
EXTI20	RTC wakeup event
EXTI21	USART wakeup event

9.5 Register Description

9.5.1 EXTI Registers

Table 9-3 EXTI-related registers

Name	Access address	Description	Reset value
R32_EXTI_INTENR	0x40010400	Interrupt enable register	0x00000000
R32_EXTI_EVENR	0x40010404	Event enable register	0x00000000
R32_EXTI_RTENR	0x40010408	Rising edge trigger enable register	0x00000000
R32_EXTI_FTENR	0x4001040C	Falling edge trigger enable register	0x00000000
R32_EXTI_SWIEVR	0x40010410	Software interrupt event register	0x00000000
R32_EXTI_INTFR	0x40010414	Interrupt flag register	0x0000XXXX

9.5.1.1 Interrupt Enable Register (EXTI_INTENR)

Offset address: 0x00

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved										MR21	MR20	MR19	MR19	MR17	MR16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MR15	MR14	MR13	MR12	MR11	MR10	MR9	MR8	MR7	MR6	MR5	MR4	MR3	MR2	MR1	MR0

Bit	Name	Access	Description	Reset value
[31:22]	Reserved	RO	Reserved	0
[21:0]	MRx	RW	Enable the interrupt request signal for external interrupt channel x: 1: Enable interrupt for this channel; 0: Mask interrupt for this channel.	0

9.5.1.2 Event Enable Register (EXTI_EVENR)

Offset address: 0x04

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved										MR21	MR20	MR19	MR18	MR17	MR16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MR15	MR14	MR13	MR12	MR11	MR10	MR9	MR8	MR7	MR6	MR5	MR4	MR3	MR2	MR1	MR0

Bit	Name	Access	Description	Reset value
[31:22]	Reserved	RO	Reserved	0
[21:0]	MRx	RW	Enable the event request signal for external interrupt channel x: 1: Enable event for this channel;	0

			0: Mask event for this channel.	

9.5.1.3 Rising Edge Trigger Enable Register (EXTI_RTENR)

Offset address: 0x08

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved										TR21	TR20	TR19	TR18	TR17	TR16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TR15	TR14	TR13	TR12	TR11	TR10	TR9	TR8	TR7	TR6	TR5	TR4	TR3	TR2	TR1	TR0

Bit	Name	Access	Description	Reset value
[31:22]	Reserved	RO	Reserved	0
[21:0]	TRx	RW	Enable rising edge triggering of external interrupt channel x: 1: Enable rising edge triggering for this channel; 0: Disable rising edge triggering for this channel.	0

9.5.1.4 Falling Edge Trigger Enable Register (EXTI_FTENR)

Offset address: 0x0C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved										TR21	TR20	TR19	TR18	TR17	TR16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TR15	TR14	TR13	TR12	TR11	TR10	TR9	TR8	TR7	TR6	TR5	TR4	TR3	TR2	TR1	TR0

Bit	Name	Access	Description	Reset value
[31:22]	Reserved	RO	Reserved	0
[21:0]	TRx	RW	Enable falling edge triggering of external interrupt channel x: 1: Enable falling edge triggering for this channel; 0: Disable falling edge triggering for this channel.	0

9.5.1.5 Software Interrupt Event Register (EXTI_SWIEVR)

Offset address: 0x10

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved										SWIER 21	SWIER 20	SWIER 19	SWIER 18	SWIER 17	SWIER 16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SWIER15	SWIER 14	SWIER13	SWIER12	SWIER11	SWIER 10	SWIER 9	SWIER 8	SWIER 7	SWIER 6	SWIER 5	SWIER 4	SWIER 3	SWIER 2	SWIER 1	SWIER 0

Bit	Name	Access	Description	Reset value

[31:22]	Reserved	RO	Reserved	0
[21:0]	SWIERx	RW	A software interrupt is set on the corresponding externally triggered interrupt channel. Setting it here causes the interrupt flag bit (EXTI_INTFR) to be set to the corresponding position bit, and an interrupt or event is generated if the interrupt enable (EXTI_INTENR) or event enable (EXTI_EVENR) is turned on.	0

9.5.1.6 Interrupt Flag Register (EXTI_INTFR)

Offset address: 0x14

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved										IF21	IF20	IF19	IF18	IF17	IF16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IF15	IF14	IF13	IF12	IF11	IF10	IF9	IF8	IF7	IF6	IF5	IF4	IF3	IF2	IF1	IF0

Bit	Name	Access	Description	Reset value
[31:22]	Reserved	RO	Reserved	0
[21:0]	IFx	RW1	Interrupt Flag Bit, this location bit flag indicates that the corresponding external interrupt has occurred. Writing 1 clears this bit.	0

9.5.2 PFIC Registers

Table 9-4 PFIC-related registers

Name	Access address	Description	Reset value
R32_PFIC_ISR1	0xE000E000	PFIC interrupt enable status register 1	0x0000032C
R32_PFIC_ISR2	0xE000E004	PFIC interrupt enable status register 2	0x00000000
R32_PFIC_ISR3	0xE000E008	PFIC interrupt enable status register 3	0x00000000
R32_PFIC_IPR1	0xE000E020	PFIC interrupt pending status register 1	0x00000000
R32_PFIC_IPR2	0xE000E024	PFIC interrupt pending status register 2	0x00000000
R32_PFIC_IPR3	0xE000E028	PFIC interrupt pending status register 3	0x00000000
R32_PFIC_ITHRESDR	0xE000E040	PFIC interrupt priority threshold configuration register	0x00000000
R32_PFIC_CFGR	0xE000E048	PFIC interrupt configuration register	0x00000000
R32_PFIC_GISR	0xE000E04C	PFIC interrupt global status register	0x00000000
R32_PFIC_VTFIDR	0xE000E050	PFIC VTF interrupt ID configuration register	0xXXXXXXXX
R32_PFIC_VTFADDRR0	0xE000E060	PFIC VTF interrupt 0 address register	0xXXXXXXXX
R32_PFIC_VTFADDRR1	0xE000E064	PFIC VTF interrupt 1 address register	0xXXXXXXXX
R32_PFIC_VTFADDRR2	0xE000E068	PFIC VTF interrupt 2 address register	0xXXXXXXXX
R32_PFIC_VTFADDRR3	0xE000E06C	PFIC VTF interrupt 3 address register	0xXXXXXXXX
R32_PFIC_IENR1	0xE000E100	PFIC interrupt enable set register 1	0x00000000

R32_PFIC_IENR2	0xE000E104	PFIC interrupt enable set register 2	0x00000000
R32_PFIC_IENR3	0xE000E108	PFIC interrupt enable set register 3	0x00000000
R32_PFIC_IRER1	0xE000E180	PFIC interrupt enable clear register 1	0x00000000
R32_PFIC_IRER2	0xE000E184	PFIC interrupt enable clear register 2	0x00000000
R32_PFIC_IRER3	0xE000E188	PFIC interrupt enable clear register 3	0x00000000
R32_PFIC_IPSR1	0xE000E200	PFIC interrupt pending set register 1	0x00000000
R32_PFIC_IPSR2	0xE000E204	PFIC interrupt pending set register 2	0x00000000
R32_PFIC_IPSR3	0xE000E208	PFIC interrupt pending set register 3	0x00000000
R32_PFIC_IPRR1	0xE000E280	PFIC interrupt pending clear register 1	0x00000000
R32_PFIC_IPRR2	0xE000E284	PFIC interrupt pending clear register 2	0x00000000
R32_PFIC_IPRR3	0xE000E288	PFIC interrupt pending clear register 3	0x00000000
R32_PFIC_IACTR1	0xE000E300	PFIC interrupt activation register 1	0x00000000
R32_PFIC_IACTR2	0xE000E304	PFIC interrupt activation register 2	0x00000000
R32_PFIC_IACTR3	0xE000E308	PFIC interrupt activation register 3	0x00000000
R32_PFIC_IPRIORx	0xE000E400	PFIC interrupt priority configuration register	0x00000000
R32_PFIC_WAKEIP0	0xE000E720	PFIC wake-up instruction pointer register 0	0x00000001
R32_PFIC_CSTAR0	0xE000E780	PFIC core status register 0	0x00000000
R32_PFIC_EENR	0xE000EC80	PFIC event enable register	0xFFFFFFFF
R32_PFIC_EPR	0xE000EC84	PFIC event pending register	0x00000000
R32_PFIC_EWUPR	0xE000EC88	PFIC event wake-up register	0x00000000
R32_PFIC_SCTLR	0xE000ED10	PFIC system control register	0x00000000

Note: 1. NMI, HardFault, ECALL-M, ECALL-U, BREAKPOINT interrupts are always enabled by default.

2. ECALL-M, ECALL-U, BREAKPOINT are all EXC cases, and the status is indicated by EXC status bit 3.

3. NMI and EXC support interrupt suspend clear and set operations, but not interrupt enable clear and set operations.

4. ECALL-M, ECALL-U, and BREAKPOINT do not support interrupt hang clear and set, interrupt enable clear and set operations.

9.5.2.1 PFIC Interrupt Enable Status Register 1 (PFIC_ISR1)

Offset address: 0x00

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
INTENSTA[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INTENSTA15	INTENSTA14	INTENSTA13	INTENSTA12	Reserved		INTENSTA9	INTENSTA8	Reserved		INTENSTA5	Reserved	INTENSTA3	INTENSTA2	Reserved	

Bit	Name	Access	Description	Reset value
[31:12]	INTENSTA	RO	12#-31#Interrupt current enable status. 1: Current numbered interrupt is enabled; 0: Current numbered interrupt not enabled.	0
[11:10]	Reserved	RO	Reserved	0

[9:8]	INTENSTA	RO	8#-9# Interrupt current enable status. 1: Current numbered interrupt is enabled; 0: Current numbered interrupt not enabled.	0x3
[7:6]	Reserved	RO	Reserved	0
5	INTENSTA	RO	5# Interrupt current enable status. 1: Current numbered interrupt is enabled; 0: Current numbered interrupt not enabled.	1
4	Reserved	RO	Reserved	0
[3:2]	INTENSTA	RO	2#-3# Interrupt current enable status. 1: Current numbered interrupt is enabled; 0: Current numbered interrupt not enabled.	0x3
[1:0]	Reserved	RO	Reserved	0

9.5.2.2 PFIC Interrupt Enable Status Register 2 (PFIC_ISR2)

Offset address: 0x04

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
INTENSTA[63:48]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INTENSTA[47:32]															

Bit	Name	Access	Description	Reset value
[31:0]	INTENSTA	RO	32#-63# Interrupt current enable status. 1: Current numbered interrupt is enabled; 0: Current numbered interrupt not enabled.	0

9.5.2.3 PFIC Interrupt Enable Status Register 3 (PFIC_ISR3)

Offset address: 0x08

31	30	29	28	27	26	25	24	23	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															INTENSTA [65:64]

Bit	Name	Access	Description	Reset value
[31:2]	Reserved	RO	Reserved	0
[1:0]	INTENSTA	RO	64#-65# Interrupt current enable status. 1: Current numbered interrupt is enabled; 0: Current numbered interrupt not enabled.	0

9.5.2.4 PFIC Interrupt Pending Status Register 1 (PFIC_IPR1)

Offset address: 0x20

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

PENDSTA[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PENDSTA15	PENDSTA14	PENDSTA13	PENDSTA12	Reserved		PENDSTA9	PENDSTA8	Reserved		PENDSTA5	Reserved	PENDSTA3	PENDSTA2	Reserved	

Bit	Name	Access	Description	Reset value
[31:12]	PENDSTA	RO	12#-31#interrupts currently pending status. 1: Current numbered interrupt is pending; 0: Current numbered interrupt is not pending.	0
[11:10]	Reserved	RO	Reserved	0
[9:8]	PENDSTA	RO	8#-9# interrupts currently pending status. 1: Current numbered interrupt is pending; 0: Current numbered interrupt is not pending.	0
[7:6]	Reserved	RO	Reserved	0
5	PENDSTA	RO	5# interrupts currently pending status. 1: Current numbered interrupt is pending; 0: Current numbered interrupt is not pending.	0
4	Reserved	RO	Reserved	0
[3:2]	PENDSTA	RO	2#-3# interrupts currently pending status. 1: Current numbered interrupt is pending; 0: Current numbered interrupt is not pending.	0
[1:0]	Reserved	RO	Reserved	0

9.5.2.5 PFIC Interrupt Pending Status Register 2 (PFIC_IPR2)

Offset address: 0x24

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PENDSTA[63:48]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PENDSTA[47:32]															

Bit	Name	Access	Description	Reset value
[31:0]	PENDSTA	RO	32#-63# interrupts currently pending status. 1: Current numbered interrupt is pending; 0: Current numbered interrupt is not pending.	0

9.5.2.6 PFIC Interrupt Pending Status Register 3 (PFIC_IPR3)

Offset address: 0x28

31	30	29	28	27	26	25	24	23	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	PENDSTA [65:64]

Bit	Name	Access	Description	Reset value
[31:2]	Reserved	RO	Reserved	0
[1:0]	PENDSTA	RO	64#-65# interrupts currently pending status. 1: Current numbered interrupt is pending; 0: Current numbered interrupt is not pending.	0

9.5.2.7 PFIC Interrupt Priority Threshold Configuration Register (PFIC_ITHRESDR)

Offset address: 0x40

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

Bit	Name	Access	Description	Reset value
[31:8]	Reserved	RO	Reserved	0
[7:0]	THRESHOLD	RW	Interrupt priority threshold setting value. Interrupt priority values lower than the current set value do not perform interrupt service when hung; a 0 in this register indicates that the threshold register function is invalid. [7:4]: Priority threshold value. [3:0]: Reserved, fixed to 0.	0

9.5.2.8 PFIC Interrupt Configuration Register (PFIC_CFGR)

Offset address: 0x48

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

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

Bit	Name	Access	Description	Reset value
[31:16]	KEYCODE[15:0]	WO	Key control keyword is valid when writing to Beef, otherwise the operation is invalid.	0
[15:8]	Reserved	RO	Reserved	0
7	SYSRST	WO	Reset register, system reset when writing 0xBEEF0080, other values are invalid. *Note: Same function as PFIC_SCTLR register SYSRST bit.*	0
[6:0]	Reserved	RO	Reserved	0

9.5.2.9 PFIC Interrupt Global Status Register (PFIC_GISR)

Offset address: 0x4C

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EX_STATE[1:0]		LOCK_UP	DBG_MODE	GLOBL_IE	Reserved	GPENDSTA	GACTSTA	NESTSTA[7:0]							

Bit	Name	Access	Description	Reset value
[31:16]	Reserved	RO	Reserved	0
[15:14]	EX_STATE[1:0]	RO	Core status register, used to determine the core's running state 00: Running; 01: Light sleep; 10: Deep sleep; 11: Deep sleep (locked).	0
13	LOCK_UP	RO	Core lock status register, used to query whether the core is in a locked state. 1: Locked; 0: Not locked.	0
12	DBG_MODE	RO	Core debug mode register, used to query whether the core is in debug mode. 1: The core is in debug mode; 0: The core is in non-debug mode.	0
11	GLOBL_IE	RO	The core global interrupt enable register is used to query whether the core global interrupt is enabled. 1: The core global interrupt enable is on; 0: The core global interrupt enable is off.	0
10	Reserved	RO	Reserved	0
9	GPENDSTA	RO	Whether an interrupt is currently pending: 1: Yes; 0: No.	0
8	GACTSTA	RO	Whether the interrupt is executed currently: 1: Yes; 0: No.	0
[7:0]	NESTSTA	RO	The current interrupt nesting status is used to query the nesting status of kernel interrupts. Same as the nest_sta bit of the INEST_CTLR register of the CSR register.	0

9.5.2.10 PFIC VTF Interrupt ID Configuration Register (PFIC_VTFIDR)

Offset address: 0x50

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

VTFID3								VTFID2							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VTFID1								VTFID0							

Bit	Name	Access	Description	Reset value
[31:24]	VTFID3	RW	Configure VTF interrupt 3 ID	X
[23:16]	VTFID2	RW	Configure VTF interrupt 2 ID	X
[15:8]	VTFID1	RW	Configure VTF interrupt 1 ID	X
[7:0]	VTFID0	RW	Configure VTF interrupt 0 ID	X

9.5.2.11 PFIC VTF Interrupt 0 Address Register (PFIC_VTFADDRR0)

Offset address: 0x60

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ADDR0[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR0[15:1]															VTF0EN

Bit	Name	Access	Description	Reset value
[31:1]	ADDR0	RW	VTF interrupt 0 service program address bit[31:1].	X
0	VTF0EN	RW	VTF interrupt 0 enable bit: 1: Enabled VTF interrupt 0 channel; 0: Disabled.	0

9.5.2.12 PFIC VTF Interrupt 1 Address Register (PFIC_VTFADDRR1)

Offset address: 0x64

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ADDR1[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR1[15:1]															VTF1EN

Bit	Name	Access	Description	Reset value
[31:1]	ADDR1	RW	VTF interrupt 1 service program address bit[31:1].	X
0	VTF1EN	RW	VTF interrupt 1 enable bit: 1: Enabled VTF interrupt 1 channel; 0: Disabled.	0

9.5.2.13 PFIC VTF Interrupt 2 Address Register (PFIC_VTFADDRR2)

Offset address: 0x68

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ADDR2[31:16]															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR2[15:1]															VTF2EN

Bit	Name	Access	Description	Reset value
[31:1]	ADDR2	RW	VTF interrupt 2 service program address bit[31:1].	X
0	VTF2EN	RW	VTF interrupt 2 enable bit: 1: Enabled VTF interrupt 2 channel; 0: Disabled.	0

9.5.2.14 PFIC VTF Interrupt 3 Address Register (PFIC_VTFADDRR3)

Offset address: 0x6C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ADDR3[31:16]															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR3[15:1]															VTF3EN

Bit	Name	Access	Description	Reset value
[31:1]	ADDR3	RW	VTF interrupt 3 service program address bit[31:1].	X
0	VTF3EN	RW	VTF interrupt 3 enable bit: 1: Enabled VTF interrupt 3 channel; 0: Disabled.	0

9.5.2.15 PFIC Interrupt Enable Set Register 1 (PFIC_IENR1)

Offset address: 0x100

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INTEN15	INTEN14	INTEN13	INTEN12	Reserved											

Bit	Name	Access	Description	Reset value
[31:12]	INTEN	WO	12#-31# interrupt enable control. 1: Current numbered interrupt enable; 0: No effect.	0
[11:0]	Reserved	RO	Reserved	0

9.5.2.16 PFIC Interrupt Enable Set Register 2 (PFIC_IENR2)

Offset address: 0x104

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
INTEN[63:48]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INTEN[47:32]															

Bit	Name	Access	Description	Reset value
[31:0]	INTEN	WO	32#-63# interrupt enable control. 1: Current numbered interrupt enable; 0: No effect.	0

9.5.2.17 PFIC Interrupt Enable Set Register 3 (PFIC_IENR3)

Offset address: 0x108

31	30	29	28	27	26	25	24	23	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														INTEN [65:64]	

Bit	Name	Access	Description	Reset value
[31:2]	Reserved	RO	Reserved	0
[1:0]	INTEN	WO	64#-65# interrupt enable control. 1: Current numbered interrupt enable; 0: No effect.	0

9.5.2.18 PFIC Interrupt Enable Clear Register 1 (PFIC_IRER1)

Offset address: 0x180

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
INTRSET[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INTRSET15	INTRSET14	INTRSET13	INTRSET12	Reserved											

Bit	Name	Access	Description	Reset value
[31:12]	INTRSET	WO	12#-31# interrupt shutdown control. 1: Current numbered interrupt shutdown; 0: No effect.	0
[11:0]	Reserved	RO	Reserved	0

9.5.2.19 PFIC Interrupt Enable Clear Register 2 (PFIC_IRER2)

Offset address: 0x184

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
INTRSET[63:48]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INTRSET[47:32]															

Bit	Name	Access	Description	Reset value
[31:0]	INTRSET	WO	32#-63# interrupt shutdown control. 1: Current numbered interrupt shutdown; 0: No effect.	0

9.5.2.20 PFIC Interrupt Enable Clear Register 3 (PFIC_IRER3)

Offset address: 0x188

31	30	29	28	27	26	25	24	23	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														INTRSET [65:64]	

Bit	Name	Access	Description	Reset value
[31:2]	Reserved	RO	Reserved	0
[1:0]	INTRSET	WO	64#-65# interrupt shutdown control. 1: Current numbered interrupt shutdown; 0: No effect.	0

9.5.2.21 PFIC Interrupt Pending Set Register 1 (PFIC_IPSR1)

Offset address: 0x200

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PENDSET[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PEND SET15	PEND SET14	PEND SET13	PEND SET12	Reserved		PEND SET9	PEND SET8	Reserved		PEND SET5	Reserved	PEND SET3	PEND SET2	Reserved	

Bit	Name	Access	Description	Reset value
[31:12]	PENDSET	WO	12#-31# interrupt pending settings. 1: Current numbered interrupt pending; 0: No effect.	0
[11:10]	Reserved	RO	Reserved	0

[9:8]	PENDSET	RO	8#-9# interrupt pending settings. 1: Current numbered interrupt pending; 0: No effect.	0
[7:6]	Reserved	RO	Reserved	0
5	PENDSET	RO	5# interrupt pending settings. 1: Current numbered interrupt pending; 0: No effect.	0
4	Reserved	RO	Reserved	0
[3:2]	PENDSET	WO	2#-3# interrupt pending settings. 1: Current numbered interrupt pending; 0: No effect.	0
[1:0]	Reserved	RO	Reserved	0

9.5.2.22 PFIC Interrupt Pending Set Register 2 (PFIC_IPSR2)

Offset address: 0x204

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PENDSET[63:48]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PENDSET[47:32]															

Bit	Name	Access	Description	Reset value
[31:0]	PENDSET	WO	32#-63# interrupt pending settings. 1: Current numbered interrupt pending; 0: No effect.	0

9.5.2.23 PFIC Interrupt Pending Set Register 3 (PFIC_IPSR3)

Offset address: 0x208

31	30	29	28	27	26	25	24	23	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														PENDSET [65:64]	

Bit	Name	Access	Description	Reset value
[31:2]	Reserved	RO	Reserved	0
[1:0]	PENDSET	WO	64#-65# interrupt pending settings. 1: Current numbered interrupt pending; 0: No effect.	0

9.5.2.24 PFIC Interrupt Pending Clear Register 1 (PFIC_IPRR1)

Offset address: 0x280

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

PENDRST[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PENDRST15	PENDRST14	PENDRST13	PENDRST12	Reserved		PENDRST9	PENDRST8	Reserved		PENDRST5	Reserved	PENDRST3	PENDRST2	Reserved	

Bit	Name	Access	Description	Reset value
[31:12]	PENDRST	WO	12#-31# interrupt pending clear. 1: Current numbered interrupt clears the pending state; 0: No effect.	0
[11:10]	Reserved	RO	Reserved	0
[9:8]	PENDRST	RO	8#-9# interrupt pending clear. 1: Current numbered interrupt clears the pending state; 0: No effect.	0
[7:6]	Reserved	RO	Reserved	0
5	PENDRST	RO	5# interrupt pending clear. 1: Current numbered interrupt clears the pending state; 0: No effect.	0
4	Reserved	RO	Reserved。	0
[3:2]	PENDRST	WO	2#-3# interrupt pending clear. 1: Current numbered interrupt clears the pending state; 0: No effect.	0
[1:0]	Reserved	RO	Reserved	0

9.5.2.25 PFIC Interrupt Pending Clear Register 2 (PFIC_IPRR2)

Offset address: 0x284

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PENDRST[63:48]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PENDRST[47:32]															

Bit	Name	Access	Description	Reset value
[31:0]	PENDRST	WO	32#-36# interrupt pending clear. 1: Current numbered interrupt clears the pending state; 0: No effect.	0

9.5.2.26 PFIC Interrupt Pending Clear Register 3 (PFIC_IPRR3)

Offset address: 0x288

31	30	29	28	27	26	25	24	23	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														PENDRST [65:64]	

Bit	Name	Access	Description	Reset value
[31:2]	Reserved	RO	Reserved	0
[1:0]	PENDRST	WO	64#-65# interrupt pending clear. 1: Current numbered interrupt clears the pending state; 0: No effect.	0

9.5.2.27 PFIC Interrupt Activation Register 1 (PFIC_IACTR1)

Offset address: 0x300

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IACTS15	IACTS14	IACTS13	IACTS12	Reserved								IACTS 3	IACTS 2	Reserved	

Bit	Name	Access	Description	Reset value
[31:12]	IACTS	RO	12#-31# interrupt execution status. 1: Current numbered interrupt is executing; 0: Current numbered interrupt not executed.	0
[11:4]	Reserved	RO	Reserved	0
[3:2]	IACTS	RO	2#-3# interrupt execution status. 1: Current numbered interrupt is executing; 0: Current numbered interrupt not executed.	0
[1:0]	Reserved	RO	Reserved	0

9.5.2.28 PFIC Interrupt Activation Register 2 (PFIC_IACTR2)

Offset address: 0x304

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

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

Bit	Name	Access	Description	Reset value
[31:0]	IACTS	RO	32#-63# interrupt execution status. 1: Current numbered interrupt is executing;	0

			0: Current numbered interrupt not executed.	

9.5.2.29 PFIC Interrupt Activation Register 3 (PFIC_IACTR3)

Offset address: 0x308

31	30	29	28	27	26	25	24	23	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														IACTS [65:64]	

Bit	Name	Access	Description	Reset value
[31:2]	Reserved	RO	Reserved	0
[1:0]	IACTS	RO	64#-65# interrupt execution status. 1: Current numbered interrupt is executing; 0: Current numbered interrupt not executed.	0

9.5.2.30 PFIC Interrupt Priority Configuration Register (PFIC_IPRIORx) (x=0-63)

Offset address: 0x400 - 0x4FF

The controller supports 256 interrupts (0-255), and the priority of each interrupt is controlled by 8bits.

	31 - 24	23 - 16	15 - 8	7 - 0
IPRIOR63	PRIO_255	PRIO_254	PRIO_253	PRIO_252
…	…	…	…	…
IPRIORx	PRIO_(4x+3)	PRIO_(4x+2)	PRIO_(4x+1)	PRIO_(4x)
…	…	…	…	…
IPRIOR0	PRIO_3	PRIO_2	PRIO_1	PRIO_0

Bit	Name	Access	Description	Reset value
[2047:2040]	IP_255	RW	Same as IP_0 description.	0
…	…	…	…	…
[31:24]	IP_3	RW	Same as IP_0 description.	0
[23:16]	IP_2	RW	Same as IP_0 description.	0
[15:8]	IP_1	RW	Same as IP_0 description.	0
[7:0]	IP_0	RW	No.0 Interrupt Priority Configuration: When the preempt register value is 0, there is no preemption bit. When the preempt register value is 1, bit7 is the preemption bit. When the preempt register value is 2, bit7-6 are preemption bits.	0

			When the preempt register value is 3, bit7-5 is the preemption bit. The core supports up to 2 levels of nesting. When the nesting is full, preemption cannot continue. Bit[3:0] are fixed to 0.	

9.5.2.31 PFIC Wake-up Instruction Pointer Register 0 (PFIC_WAKEIP0)

Offset address: 0x720

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IP_RELOAD0[30:15]															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IP_RELOAD0[14:0]															SHUTDOWN0

Bit	Name	Access	Description	Reset value
[31:1]	IP_RELOAD0	RW	The PC address register when core C0 wakes up is used to reload the PC value after power-on wake-up and reset wake-up.	0
0	SHUTDOWN0	RW	The core C0 deep sleep (lock) cancellation register is used to cancel the lock of the core that enters the deep sleep (lock) state after power-on, allowing it to be woken up by event: 1: Locked; 0: Unlocked.	1

9.5.2.32 PFIC Core Status Register (PFIC_CSTAR0)

Offset address: 0x780

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CPU_ex_state_0		CPU_lock_up_0	CPU_dbg_mode_0	CPU_globl_ie_0	Reserved	CPU_irq_pend_0	CPU_irq_active_0	CPU_nest_sta_0							

Bit	Name	Access	Description	Reset value
[31:16]	Reserved	RO	Reserved.	0
[15:14]	CPU_ex_state_0	RO	Core C0 status register, used to obtain the running status of core C0: 00: Running; 01: Light sleep;	0

			10: Deep sleep; 11: Deep sleep (locked).	
13	CPU_lock_up_0	RO	Core C0 lock status register, used to query whether core C0 is in the locked state: 1: Locked; 0: Not locked.	0
12	CPU_dbg_mode_0	RO	The core C0 debug mode register is used to query whether the core C0 is in debug mode: 1: The core is in debug mode; 0: The core is in non-debug mode.	0
11	CPU_globl_ie_0	RO	The core C0 global interrupt enable register is used to query whether the core C0 global interrupt is enabled: 1: The core global interrupt enable is on; 0: The core global interrupt enable is off.	0
10	Reserved	RO	Reserved	0
9	CPU_irq_pend_0	RO	The core C0 interrupt pending flag register is used to query whether there are unhandled interrupts in the core C0: 1: There are unhandled interrupts in the core; 0: There are no unhandled interrupts in the core.	0
8	CPU_irq_active_0	RO	Core C0 interrupt active flag register is used to query whether core C0 is processing interrupts. 1: The core is processing interrupts; 0: The core is not processing interrupts.	0
[7:0]	CPU_nest_sta_0	RO	The core C0 interrupt nesting status register is used to query the nesting status of the core C0 interrupt. Same as the nest_sta bit of the INEST_CTLR register of the CSR register.	0

9.5.2.33 PFIC Event Enable Register (PFIC_EENR)

Offset address: 0xC80

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

Bit	Name	Access	Description	Reset value
[31:0]	EVENTEN	RW	31#-0# event wake-up enable: 1: Allow the corresponding event to wake up the core during sleep; 0: Mask the corresponding event, and the core	0xFFFFFFFF

			is not allowed to wake up when this event occurs. Supports up to 32 event sources, of which 0-7 are level triggers (high effective), and 8-31 are pulse triggers: 0#: Core pause request; 1#: Core interrupt request; 7#-2#: Reserved; 8#: Core entry interrupt; 9#: Core exit interrupt; 10#: New interrupt pending; (for shared interrupts, it does not distinguish whether the interrupt is routed to the local core) 11#: Core sending event; (Local core sending event) 12#: Core receiving event; (Other core sending event) 13#: System receiving event; (System configured event, such as GPIO flip event...) 31#-14#: Reserved.	

9.5.2.34 PFIC Event Pending Register (PFIC_EPR)

Offset address: 0xC84

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

Bit	Name	Access	Description	Reset value
[31:8]	EVENT_PEND	RW1Z	31#-8# Event pending status, writing 1 can be cleared: 1: The corresponding event has not been processed; 0: The corresponding event has not occurred.	0
[7:0]	EVENT_PEND	RO	7#-0# Event pending status, cannot be cleared: 1: The corresponding event has not been processed; 0: The corresponding event has not occurred.	0

9.5.2.35 PFIC Event Wakeup Register (PFIC_EWUPR)

Offset address: 0xC88

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

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

Bit	Name	Access	Description	Reset value
[31:0]	EVENT_WUP	RO	31#-0# Event wake-up register: 1: The corresponding event wakes the core from sleep; 0: The corresponding event is not an event that wakes up the core this time.	0

9.5.2.36 PFIC System Control Register (PFIC_SCTLR)

Offset address: 0XD10

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved										SETEVENT	SEVONPEND	WFITOWFE	SLEEPDEEP	SLEEPONEXIT	Reserved

Bit	Name	Access	Description	Reset value
31	SYSRST	RW1	System reset, automatically cleared to 0. Writing 1 is valid, writing 0 is invalid, and has the same effect as the PFIC_CFGR register.	0
[30:24]	Reserved	RO	Reserved	0
[23:16]	HART_ID	RO	Core ID register, used to obtain the core ID of the reading register: 1: This core is C1; 0: This core is C0.	0
[15:6]	Reserved	RO	Reserved	0
5	SENDEVENT	WO	Write event, generates a write event, which can be used to wake up other cores.	0
4	SEVONPEND	RW	When an event occurs or the interrupt is pending, the system can be woken up from the WFE instruction. If the WFE instruction is not executed, the system will be awakened immediately after the next execution of the instruction. 1: Enabled events and all interrupts (including unenabled interrupts) can wake up the system; 0: Only enabled events and enabled interrupts can wake up.	0

3	WFITOWFE	RW	The WFI command is executed as WFE, which is a custom instruction. After this register is set to 1, the core enters a low-power state after executing the WFI instruction, and can wake up the core after waiting for an external event to occur. 1: Wait for the event to wake up the core after the WFI instruction; 0: Wait for the interrupt to wake up the core after the WFI instruction.	0
2	SLEEPDEEP	RW	Controlling the system's low-power modes: 1: deepsleep 0: sleep	0
1	SLEEPONEXIT	RW	After control leaves the interrupt service routine, the system status: 1: The system enters the low power consumption mode; 0: The system enters the main program.	0
0	Reserved	RO	Reserved	0

9.5.3 STK Registers

Table 9-5 STK-related registers

Name	Access address	Description	Reset value
R32_STK_CTLR	0xE000F000	System count control register	0x00000000
R32_STK_SR	0xE000F004	System counter status register	0x00000000
R32_STK_CNTL	0xE000F008	System counter low register	0xXXXXXXXX
R32_STK_CMPLR	0xE000F010	Count/compare low register	0xXXXXXXXX

Note: Applied for general-purpose MCUs designed based on 32-bit RISC-V instruction set and architecture.

9.5.3.1 System Count Control Register (STK_CTLR)

Offset address: 0x00

31	30	29	28	27	26	25	24	23	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											MODE	STRE	STCLK	STIE	STE

Bit	Name	Access	Description	Reset value
[31:5]	Reserved	RO	Reserved	0
4	MODE	RW	Counting modes: 1: Counting down; 0: Counting up.	0
3	STRE	RW	Automatic reload counting enable bit:	0

			1: Count upwards to the comparison value and then count again from 0; after counting downwards to 0, count again from the comparison value; 0: Continue counting up/down. 0	
2	STCLK	RW	Counter clock source select bit: 1: HCLK for time base; 0: HCLK/8 for time base.	0
1	STIE	RW	Counter interrupt enable control bit: 1: Enable counter interrupt; 0: Disable counter interrupt.	0
0	STE	RW	System counter enable control bits: 1: Starts the system counter STK; 0: Turn off the system counter STK and the counter stops counting.	0

9.5.3.2 System Counter Status Register (STK_SR)

Offset address: 0x04

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

Bit	Name	Access	Description	Reset value
31	SWIE	RW	Software interrupt trigger enable (SW): 1: Trigger software interrupt; 0: Turn off trigger. *Note: This bit must be cleared to 0 after entering the software interrupt, otherwise it will always trigger.*	0
[30:1]	Reserved	RO	Reserved	0
0	CNTIF	RW	Count value comparison flag, clear by writing 0, invalid by writing 1: 1: counting up to reach the comparison value, counting down to 0; 0: not reaching the comparison value.	0

9.5.3.3 System Counter Low Register (STK_CNTL)

Offset address: 0x08

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

CNT[15:0]

Bit	Name	Access	Description	Reset value
[31:0]	CNT[31:0]	RW	Current counter count low 32-bit.	X

9.5.3.4 Count/Compare Low Register (STK_CMPLR)

Offset address: 0x10

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

Bit	Name	Access	Description	Reset value
[31:0]	CMP[31:0]	RW	Set the low 32 bits of the compare counter value. The STK interrupt will be triggered when the up-counting CMP register value is equal to the CNT counter value or the down-counting CNT counter value is 0.	X

9.5.4 Dedicated CSR Registers

Some Control and Status Registers (CSR) are defined in the RISC-V architecture to configure or identify or record the running status. CSR registers are internal registers within the core and use a dedicated 12-bit address space.

Note: Such registers marked with the "MRW, MRO, MRW1" attribute need to be accessed by the system in machine mode.

Table 9-6 List of dedicated CSR registers

Name	CSR address	Description	Reset value
MVENDORID	0xF11	Chip vendor ID register	0x00000000
MARCHID	0xF12	Chip structure ID register	0xDC68D86D
MIMPID	0xF13	Machine implementation ID register	0xDC688002
MHARTID	0xF14	Core ID register	0x00000000
MSTATUS	0x300	Machine status register	0x00000000
MISA	0x301	Machine instruction set registers	0x40901127
MTVEC	0x305	Machine mode exception base address register	0x00000000
MSCRATCH	0x340	Machine mode temporary register	0xXXXXXXXX
MEPC	0x341	Machine exception program counter register	0xXXXXXXXX
MCAUSE	0x342	Machine exception register	0xX00000XX
MTVAL	0x343	Machine trap value register	0xXXXXXXXX
TINFO	0x7A4	Trigger information register	0x00000004
DCSR	0x7B0	Debug control and status registers	0x40000000
DPC	0x7B1	Debug PC address register	0xXXXXXXXX
DSCRATCH0	0x7B2	Debug mode temporary register 0	0xXXXXXXXX

DSCRATCH1	0x7B3	Debug mode temporary register 1	0xXXXXXXXX
DBGMCU_0	0x7C0	Debug register 0	0x00000000
FFLAGS	0x001	Floating point cumulative exception flag register	0x00000000
FRM	0x002	Floating point rounding mode register	0x00000000
FCSR	0x003	Floating point control and status registers	0x00000000
CPU_RUN_CTLR	0xBC0	Processor operation control register	0x12370000
INEST_CTLR	0xBC1	Interrupt nesting control register	0x00000000
MIE	0xBC8	Machine interrupt register	0x00000000
UACCES_MSTATUS	0x800	User access to machine status register	0x00000000
HW_POPDM_CTLR	0x804	Hardware push control register	0x00000004

9.5.4.1 Chip Vendor ID Register (MVENDORID)

CSR address: 0xF11

Bit	Name	Access	Description	Reset value
[31:0]	mvendorid[31:0]	MRO	Chip vendor ID, fixed value is 0.	0

9.5.4.2 Chip Structure ID Register (MARCHID)

CSR address: 0xF12

Bit	Name	Access	Description	Reset value
[31:0]	marchid[31:0]	MRO	Chip structure ID, fixed value cannot be modified.	0xDC68D86D

9.5.4.3 Machine Implementation ID Register (MIMPID)

CSR address: 0xF13

Bit	Name	Access	Description	Reset value
[31:0]	mimpid[31:0]	MRO	Machine implementation ID, fixed value cannot be modified.	0xDC688002

9.5.4.4 Core ID Register (MHARTID)

CSR address: 0xF14

Bit	Name	Access	Description	Reset value
[31:0]	mhartid[31:0]	MRO	Core ID, indicating the core ID number of this core in the chip structure.	0

9.5.4.5 Machine Status Register (MSTATUS)

CSR address: 0x300

Bit	Name	Access	Description	Reset value
31	SD	MRO	Used to mark whether floating point instructions are in a dirty state.	0
[30:15]	Reserved	MRO	Reserved	0

[14:13]	FS[1:0]	MRW	Floating-point instruction set status, single-precision floating-point instructions (F) are allowed to execute when the value is not 0: 00: OFF; 01: Initial; 10: Clean; 11: Dirty.	0
[12:11]	MPP[1:0]	MRW	Save/restore the machine state before entering the exception, and update it to the machine state before the exception when entering the exception: 00: The machine status is set to U mode when exiting abnormally; 01: Reserved; 10: Reserved; 11: The machine status is set to M mode when exiting abnormally.	0
[10:8]	Reserved	MRO	Reserved.	0
7	MPIE	MRW	Save the interrupt enable status when entering the interrupt: 1: Enable global interrupts after exiting the interrupt; 0: Disable global interrupt after exiting the interrupt.	0
[6:4]	Reserved	MRO	Reserved.	0
3	MIE	MRW	Global interrupt is enabled, updated to MPIE value when exiting the interrupt: 1: Global interrupt is enabled, allowing the processor to respond to interrupt requests; 0: Global interrupt is disabled, and the processor is not allowed to respond to interrupt requests.	0
[2:0]	Reserved	MRO	Reserved.	0

9.5.3.6 Machine Instruction Set Registers (MISA)

CSR address: 0x301

Bit	Name	Access	Description	Reset value
[31:30]	XLEN[1:0]	MRO	Machine basic integer bit width and basic instruction bit width information, 32-bit wide core, fixed value is 1.	1
[29:24]	Reserved	MRO	Reserved	0
23	X	MRO	This core supports execution of custom instruction sets.	1
[22:21]	Reserved	MRO	Reserved	0

20	U	MRO	Support user mode operation.	1
[19:13]	Reserved	MRO	Reserved	0
12	M	MRO	This core supports execution of the multiplication instruction set.	1
[11:9]	Reserved	MRO	Reserved	0
8	I	MRO	This core supports execution of the basic integer instruction set.	1
[7:6]	Reserved	MRO	Reserved	0
5	F	MRO	This core supports execution of the single-precision floating-point instruction set.	1
[4:3]	Reserved	MRO	Reserved	0
2	C	MRO	This core supports the execution of compressed instruction sets.	1
1	B	MRO	This core supports a set of instructions for performing bit operations.	1
0	A	MRO	This core supports execution of atomic instruction sets.	1

9.5.4.7 Machine Trap-vector Base Address Register (MTVEC)

CSR address: 0x305

Bit	Name	Access	Description	Reset value
[31:10]	mtvec_base[21:0]	MRW	The high bits of the base address that the processor jumps to when an exception occurs, 1kB aligned, actual 32-bit base address BASE = {mtvec_base,10'h0}.	0
[9:2]	Reserved	MRO	Reserved	0
1	mode1	RW	Vector table mode 1: 1: The jump address is stored in the interrupt vector table; 0: The jump instruction is stored in the interrupt vector table.	0
0	mode0	RW	Vector table mode 0: 1: All interrupts/exception independent entries, the jump address is BASE+4*IRQ/EXC ID; 0: All interrupts/exception unified entries, jump to the exception jump base address.	0

9.5.4.8 Machine Mode Temporary Register (MSCRATCH)

CSR address: 0x340

Bit	Name	Access	Description	Reset value
[31:0]	mscratch[31:0]	MRW	Machine mode temporary registers.	X

9.5.4.9 Machine Exception Program Counter Register (MEPC)

CSR address: 0x341

Bit	Name	Access	Description	Reset value
[31:0]	mepc[31:0]	MRW	Save the PC address recovered when exiting the interrupt/exception, mepc[0] is fixed to 0.	X

9.5.4.10 Machine Exception Register (MCAUSE)

CSR address: 0x342

Bit	Name	Access	Description	Reset value
31	Interrupt	RW	Interrupt/exception flag bit: 1: Exception/interrupt is generated externally; 0: Exception/interrupt is generated by the core.	X
[30:8]	Reserved	RO	Reserved	0
[7:0]	Exception Code[7:0]	RW	Save the source information that generated the exception or interrupt.	X

Interrupt exception source code:

Interrupt/Exception	Exception Code	Interrupt/Exception Source
1	2	Reserve NMI
1	3	Machine mode software interrupt
1	4	User mode timer interrupt
1	5	Watchdog mode timer interrupt
1	6	Reserved
1	7	Machine mode timer interrupt
1	8	User mode external interrupt
1	11	Machine mode external interrupt
1	12	System timer interrupt
1	≥16	Reserve platform interrupt id
0	0	Instruction address misalignment
0	1	Command access error
0	2	Illegal instruction
0	3	Breakpoint
0	4	Read address is not aligned
0	5	Read access error
0	6	Write/atomic operation address non-alignment
0	7	Write/atomic operation access error
0	8	User mode ecall
0	11	Machine mode ecall
0	≥16	Reserved

9.5.4.11 Machine Trap Value register (MTVAL)

CSR address: 0x343

Bit	Name	Access	Description	Reset value

[31:0]	mtval[31:0]	RW	Save additional information about self-trap.	X

Additional information saved in the trap by the mtval register:

Exception type	Mcause	Save information
Instruction address misalignment	0	Instruction PC
Command access failed	1	Instruction PC
Illegal instruction	2	Instruction machine code
Read address is not aligned	4	Read data address
Read access failed	5	Read data address
Write/atomic access failed	6	Write data address
Write/atomic address non-alignment	7	Write data address

9.5.4.12 Trigger Information Register (TINFO)

CSR address: 0x7A4

Bit	Name	Access	Description	Reset value
[31:0]	tinfo[31:0]	MRO	Trigger information register, corresponding to the type value of the TDATA1 register, the type value is 2, and the second bit of tinfo is 1.	0x4

9.5.4.13 Debug Control and Status Registers (DCSR)

CSR address: 0x7B0

Bit	Name	Access	Description	Reset value
[31:28]	exdebug[3:0]	MRO	There are external debuggers that conform to the debugger protocol.	0x4
[27:16]	Reserved	MRO	Reserved	0
15	ebreakm	MRW	Enable the ebreak instruction to enter debugging mode in machine mode: 1: Run the ebreak instruction in machine mode, and the processor enters debugging mode; 0: Run the ebreak instruction in machine mode, and the processor enters an exception.	0
[14:13]	Reserved	MRO	Reserved	0
12	ebreaku	MRW	Enable the ebreak instruction to enter debug mode in user mode: 1: Run the ebreak instruction in user mode, and the processor enters debug mode; 0: Run the ebreak instruction in user mode, and the processor enters an exception.	0
[11:10]	Reserved	MRO	Reserved	0
9	stoptime	MRW	Timer pause enable: 1: The counter stops counting in debugging mode;	0

			0: The counter works normally in debugging mode.	
[8:6]	dcause[2:0]	MRW	Debug source register, used to mark the specific event that caused the debug mode to be entered 000: Reset value; 001: Debugging triggered by ebreak instruction in the program (priority 3); 010: Reserved; 011: Pause request generated by the debugger (priority 1); 100: Reserved; 101: Not implemented; 110: Not implemented; 111: Reserved.	0
[5:4]	Reserved	MRO	Reserved	0
3	nmip	MRO	Core NMI interrupt pending flag.	0
2	Reserved	MRO	Reserved	0
[1:0]	prv[1:0]	MRW	Record the machine status before entering debug mode: 00: Enter debugging mode from user mode, and the processor returns to user mode during recovery; 01, 10: Reserved; 11: Enter debugging mode from machine mode, and the processor returns to machine mode during recovery.	0

9.5.4.14 Debug PC Address Register (DPC)

CSR address: 0x7B1

Bit	Name	Access	Description	Reset value
[31:0]	dpc[31:0]	RW	Save the PC address restored when exiting debug mode.	X

When jumping to debugging mode caused by different debugging request sources, the address saved by dpc is as follows:

Inducement	Save address
ebreak	The instruction address of Ebreak.
Pause request	The address of the next instruction that should be executed when entering debug mode.

9.5.4.15 Debug Mode Temporary Register 0 (DSCRATCH0)

CSR address: 0x7B2

Bit	Name	Access	Description	Reset value
[31:0]	Dscratch0[31:0]	RW	Debug mode temporary register 0.	X

9.5.4.16 Debug Mode Temporary Register 1 (DSCRATCH1)

CSR address: 0x7B3

Bit	Name	Access	Description	Reset value
[31:0]	Dscratch1[31:0]	RW	Debug mode temporary register 1.	X

9.5.4.17 Debug Register 0 (DBGMCU_0)

CSR address: 0x7C0

Bit	Name	Access	Description	Reset value
[31:8]	dbg_mode_stop	MRW	The registered module stops working when entering debug mode: 1: The registered peripherals stop working when the core enters debug mode; 0: The registered peripherals work normally when the core enters debug mode.	0
[7:0]	Reserved	MRO	Reserved	0

9.5.4.18 Floating-point Cumulative Exception Flag Register (FFLAGS)

CSR address: 0x001

Bit	Name	Access	Description	Reset value
[31:5]	Reserved	RO	Reserved	0
[4:0]	FFlags[4:0]	RW	Cumulative exception flag representing exceptions that have occurred on floating-point instructions since the last reset (reset or software clear). name / bit / description: NV / 4 / Illegal operation DZ / 3 / Divide by zero OF / 2 / Overflow upward UF / 1 / Overflow downward NX / 0 / Imprecise	0

9.5.4.19 Floating-point Rounding Mode Register (FRM)

CSR address: 0x002

Bit	Name	Access	Description	Reset value
[31:3]	Reserved	RO	Reserved	0
[2:0]	Frm[2:0]	RW	Floating point operation rounding mode register: 000: Rounding towards the nearest value (rounding towards even numbers when in the middle value); 001: Round towards zero; 010: Round down;	0

			011: Round up; 100: Round to the nearest value (round to the maximum magnitude when in the middle value); 101, 110: Reserved; 111: Dynamic rounding, select the rounding mode according to the FRM bit configuration value in the instruction.	

9.5.4.20 Floating-point Control and Status Registers (FCSR)

CSR address: 0x003

Bit	Name	Access	Description	Reset value
[31:8]	Reserved	RO	Reserved	0
[7:5]	Frm[2:0]	RW	Floating point operation rounding mode register.	0
[4:0]	FFlags[4:0]	RW	Cumulative exception flags.	0

9.5.4.21 Processor Operation Control Register (CPU_RUN_CTLR)

CSR address: 0xBC0

Bit	Name	Access	Description	Reset value
[31:28]	fadd_clkdiv[3:0]	MRW	Floating point addition instruction frequency division coefficient, the default value is 1, that is, the floating-point addition instruction frequency is divided by 2 of the main frequency.	0x1
[27:24]	fmul_clkdiv[3:0]	MRW	Floating point addition instruction frequency division coefficient, the default value is 2, that is, the floating-point addition instruction frequency is divided by 3 of the main frequency.	0x2
[23:20]	fmac_clkdiv[3:0]	MRW	Floating point addition instruction frequency division coefficient, the default value is 3, that is, the floating-point addition instruction frequency is divided by 4 of the main frequency.	0x3
[19:16]	fdiv_clkdiv[3:0]	MRW	Floating point addition instruction frequency division coefficient, the default value is 4, that is, the floating point addition instruction frequency is divided by 8 of the main frequency.	0x7
[17:8]	Reserved	MRO	Reserved	0
7	nmi_ie	MRW	Non-maskable interrupt enable: 1: NMI interrupt is entered when an exception occurs and nesting overflow occurs;	0

			0: NMI interrupt is not entered when an exception occurs and nesting overflow occurs.	
6	int_fence	MRW	fence interrupt enable: 1: Clears interrupt requests when the fence instruction is executed; 0: Does not clear interrupt requests when the fence instruction is executed.	0
5	non_usta	MRW	UACCES_MSTATUS register enable: 1: Bits 3 and 7 of CSR address 0x800 are mapped to bits MIE and MPIE of the MSTATUS register, respectively; 0: CSR address 0x800 is a read-only register; the return value is the value of the MSTATUS register.	0
[4:2]	Reserved	MRO	Reserved	0
[1:0]	pipe_acc[1:0]	MRW	Fetch mode: 00: Prefetch is turned off. The instruction prefetch function is turned off to avoid invalid instruction fetch operations. There is at most 1 valid instruction on the CPU pipeline. This mode consumes the lowest power and reduces performance by about 2 to 3 times. 01: Prefetch enabled. When the instruction prefetch function is turned on, the CPU will continue to access the instruction memory until the number of instructions to be executed in the internal instruction buffer exceeds a certain number, or when the instruction buffer is full, instruction fetching will be suspended. This mode has high power consumption and strong performance; (CPU prediction failure will introduce redundant instruction fetch operations, and in some cases, the execution unit will introduce an additional bubble of 0 to 2 cycles, and the performance degradation of most programs will not be obvious); Others: Reserved.	0

Core	fadd_freq (max)	fmul_freq (max)	fmac_freq (max)	fdiv_freq (max)
V3F	80MHz	53MHz	40MHz	20MHz

Floating-point arithmetic instructions are multi-cycle instructions. For different types of floating-point instructions, the maximum operating frequency is as shown in the table above. According to the operating frequency of the

processor, the user needs to configure the floating-point operation division coefficient so that the divided frequency is not higher than the maximum operating frequency shown in the above table.

fxxx_freq(max)≥Core main frequency/fxxx_clkdiv

9.5.4.22 Interrupt Nesting Control Register (INEST_CTLR)

CSR address: 0xBC1

Bit	Name	Access	Description	Reset value
31	Reserved	RO	Reserved	0
30	nest_ovr	MRW	The interrupt/exception nesting overflow flag is set to 1 when the nesting depth exceeds the maximum depth. It is cleared by writing 1. *Note: An interrupt overflow occurs only when an instruction exception or an NMI interrupt is generated while executing a level-2 interrupt service routine. In this case, the exception and NMI interrupt are handled normally, but the CPU stack overflows, making it impossible to exit from the exception or NMI interrupt.*	0
[29:12]	Reserved	MRO	Reserved	0
[11:8]	nest_sta[3:0]	MRO	Nested interrupt status flags: 0000: No interrupts; 0001: Level 1 interrupt; 0011: Level 2 interrupt nesting; 0111: Level 3 interrupt nesting, overflow; Others: Reserved.	0
[7:2]	Reserved	MRO	Reserved	0
[1:0]	nest_max[1:0]	MRW	Maximum nesting level: 00: Nesting disabled (nesting feature turned off); 01: 2-level nesting (nesting enabled); 10: Write disabled; 11: Write disabled. *Note: (1) Writing 10b or 11b to this field sets the register to 01b;* *(2) When 11b is written, reading this register returns the chip's maximum nesting level.*	0

9.5.4.23 Machine Interrupt Register (MIE)

CSR address: 0xBC8

Bit	Name	Access	Description	Reset value
[31:5]	Reserved	RO	Reserved	0
[4:0]	nest_mie[4:0]	RW	The machine interrupt register stores the mie information of each level of interrupt in the multi-level interrupt nesting:	0

			status.mie=nest_mie[0]; status.mpie=nest_mie[1].	

9.5.4.24 User Access Machine Status Register (UACCES_MSTATUS)

CSR address: 0x800

Bit	Name	Access	Description	Reset value
31	SD	RO	Used to mark whether floating point instructions are in a dirty state.	0
[30:15]	Reserved	RO	Reserved	0
[14:13]	FS[1:0]	RO	Floating-point instruction set status, single-precision floating-point instructions (F) are allowed to execute when the value is not 0: 00: OFF; 01: Initial; 10: Clean; 11: Dirty.	0
[12:11]	MPP[1:0]	RO	Save/restore the machine state before entering the exception, and update it to the machine state before the exception when entering the exception: 00: The machine status is set to U mode when exiting abnormally; 01: Reserved; 10: Reserved; 11: The machine status is set to M mode when exiting abnormally.	0
[10:8]	Reserved	RO	Reserved	0
7	MPIE	RW	Save the interrupt enable status when entering the interrupt: 1: Enable global interrupts after exiting the interrupt; 0: Disable global interrupt after exiting the interrupt.	0
[6:4]	Reserved	RO	Reserved	0
3	MIE	RW	Global interrupt is enabled, updated to MPIE value when exiting the interrupt: 1: Global interrupt is enabled, allowing the processor to respond to interrupt requests; 0: Global interrupt is disabled, and the processor is not allowed to respond to interrupt requests.	0
[2:0]	Reserved	RO	Reserved	0

Note: This register is used to configure the MSTATUS register when the kernel is running in user mode. This address is available when the non_usta bit in the CPU_RUN_CTLR register is enabled. The register setting is the same as

the MSTATUS (0x300) register.

9.5.4.25 Hardware Stack Control Register (HW_POPDM_CTLR)

CSR address: 0x804

Bit	Name	Access	Description	Reset value
31	lock	RW	User mode lock flag. After locking, the user mode cannot rewrite the register configuration of this group: 1: Locked, only machine mode can be configured; 0: Unlocked, machine/user mode can be configured.	0
[30:6]	Reserved	RO	Reserved	0
5	hw_pop_off	RW	The one-time hardware pop is turned off and automatically resets after exiting the interrupt: 1: Shielding the hardware pop when exiting the interrupt next time; 0: Do not shield the hardware pop.	0
4	Reserved	RO	Reserved	0
[3:2]	preempt[1:0]	RW	The preemption priority bit width configuration register is used to configure the preemption priority bit width in the interrupt priority level: 00: The preemption bit width is 0, and interrupts of any priority cannot be nested; 01: The preemption bit width is 1, that is, the [7] bits of the interrupt priority register are the preemption priority level; 10: The preemption bit width is 2, that is, the [7:6] bits of the interrupt priority register are the preemption priority level; 11: The preemption bit width is 3, that is, the [7:5] bits of the interrupt priority register are the preemption priority level.	0x1
1	nest_en	RW	Interrupt nesting enable register: 1: Allow interrupt nesting 0: Do not allow interrupt nesting	0
0	hw_stk_en	RW	Hardware stack protection enable: 1: Use hardware stack protection when entering and exiting interrupts; 0: Do not use hardware stack protection when entering and exiting interrupts.	0

Chapter 10 GPIO and Alternate Functions (GPIO/AFIO)

The chip has 4 sets of built-in GPIO ports (PA0~PA15, PB0~PB15, PC0~PC15, PD0~PD15), with a total of 64 GPIO pins. Each pin can be configured by software as an output (push-pull or open-drain), input (with or without pull-up or pull-down), or a multiplexed peripheral function port.

All GPIO pins support controllable pull-up and pull-down resistors.

The PD4 and PD5 pins provide pull-up currents that are compatible with USBPD and Type-C specifications and are independently controlled by the PD controller. The PD4/CC1R and PD5/CC2R pins have built-in controllable Rd pull-down resistors defined by the Type-C specification. It is recommended to turn off the pull-down resistors when used as GPIO push-pull outputs.

All I/O pins in the system are powered by V_{DD} rated 3.3V and are shared with digital or analog multiplexing peripherals. Some pins can withstand 5V signal input. By changing the V_{DD} power supply, the high value of the I/O pin output level will be changed to adapt to the external communication interface level.

10.1 Main Features

Each pin of the port can be configured into one of the following modes:

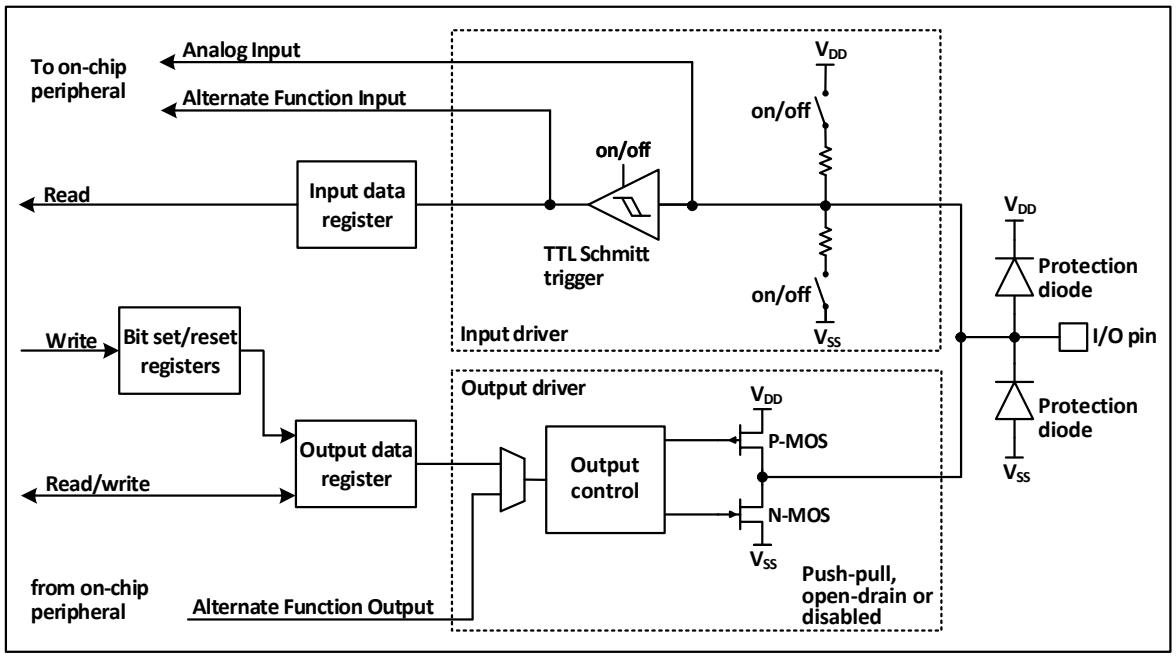
- Floating input
- Pull-up input
- Pull-down input
- Analog input
- Open-drain output
- Push-pull output
- Input and output of alternate function

Many pins have alternate functions, and many other peripherals map their own output and input channels to these pins. The specific application of these alternate pins needs to be with reference to each peripheral, and this chapter shall specify whether these pins are alternate and remapped.

10.2 Function Description

10.2.1 Overview

Figure 10-1 Basic structure of GPIO module



As shown in figure 10-1, the IO port structure, each pin has two protection diodes inside the chip, and the IO port can be divided into input and output drive modules. The input driver has a weak pull-up resistor, which can be connected to AD and other analog input peripherals; if you input to a digital peripheral, you need to go through a TTL Schmitt trigger, and then connect to the GPIO input register or other multiplexed peripherals. The output driver has a pair of MOS tubes, and the IO port can be configured to open leakage or push-pull output by configuring whether the upper and lower MOS tubes are enabled or not; the output driver can also be configured to be controlled by GPIO or by other peripherals that are reused.

10.2.2 GPIO Initialization Function

Just after the reset, the GPIO port is running in the initial state, at this time, most IO ports are running in the floating input state, but there are also HSE and other peripheral-related pins that run on the peripheral reuse function. For specific initialization functions, please refer to the relevant sections of the pin description.

10.2.3 External Interrupt

All GPIO ports can be configured with external interrupt input channels, but an external interrupt input channel can only be mapped to one GPIO pin, and the sequence number of the external interrupt channel must be consistent with the tag of the GPIO port, for example, PA1 (or PB1, PC1, PD1) can only be mapped to EXTI1, and EXTI1 can only accept the mapping of one of PA1, PB1, PC1 or PD1.

10.2.4 Alternate Function

Alternate function

The I/O pins are connected to the onboard peripherals/modules through a multiplexer that allows only one peripheral's Alternate Function (AF) to be connected to the I/O pin at a time. This ensures that there are no conflicts

between peripherals sharing the same I/O pins.

Each I/O pin has a multiplexer that takes up to 16 multiplexed function inputs (AF0 to AF15), which are configurable through the GPIOx_AFLR and GPIOx_AFHR registers:

After reset, the multiplexer is selected as Alternate Function 0 (AF0). Configure I/O through the GPIOx_CFGLR/GPIOx_CFGHR register in multiplexed mode.

The specific alternate function assignment of each pin is detailed in the *CH32X315DS0*.

Alternate function configuration

- To use the input direction alternate function, the port must be configured in input mode. It can be configured in floating, pull-up or pull-down mode according to the actual situation.
- To use the the output direction alternate function, the port must be configured in multiplexed output mode. Multiplexed push-pull or multiplexed open-drain mode can be configured according to the actual situation.
- For bidirectional alternate function, the port must be configured as multiplexed output mode, then the driver is configured as floating input mode

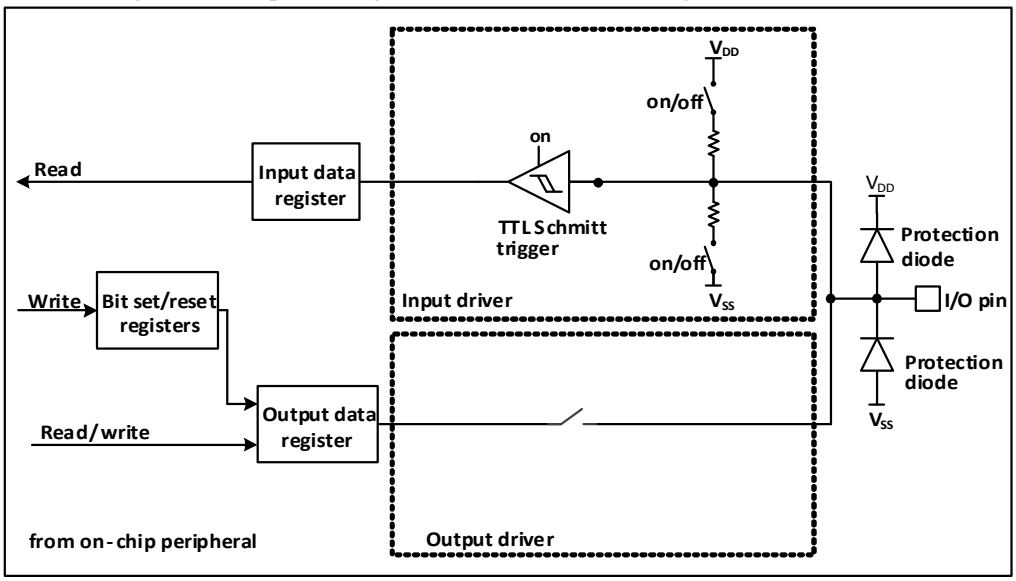
The same IO port may have multiple peripherals alternate to this pin. Therefore, in order to maximize the space for each peripheral, in addition to the default alternate pins, the alternate pins of the peripherals can also be remapped to other pins to avoid occupied pins.

10.2.5 Locking Mechanism

The locking mechanism can lock the configuration of the IO port. After a specific write sequence, the selected IO pin configuration is locked and cannot be changed until the next reset.

10.2.6 Input Configuration

Figure 10-2 Input configuration structure block diagram of GPIO module

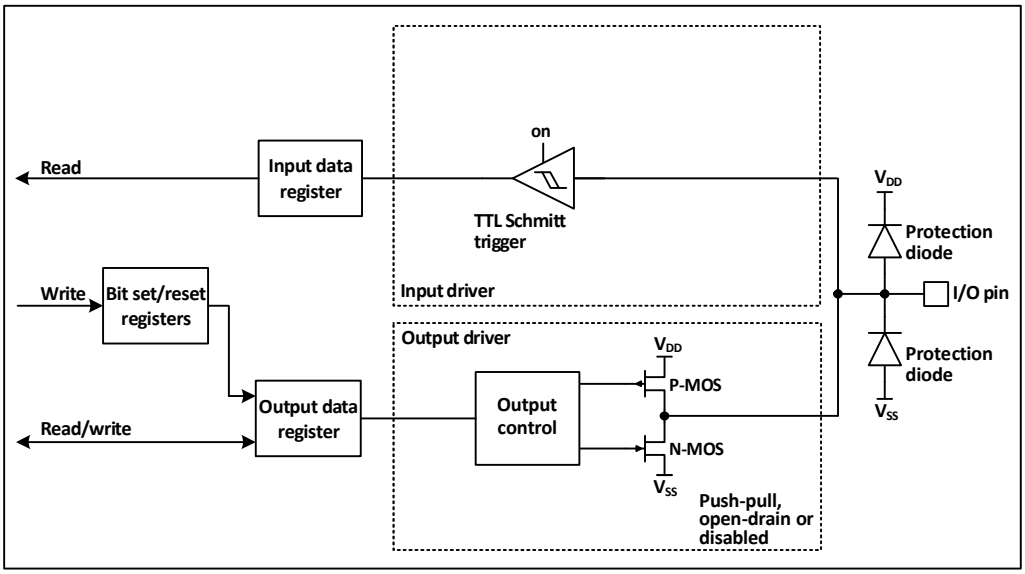


When the IO port is configured in input mode, the output driver is disconnected, the input pull-up and pull-down is optional, and the alternate function and analog input are not connected. The data on each IO port is sampled to the input data register at each HB clock, and the level state of the corresponding pin is obtained by reading the

corresponding bit of the input data register.

10.2.7 Output Configuration

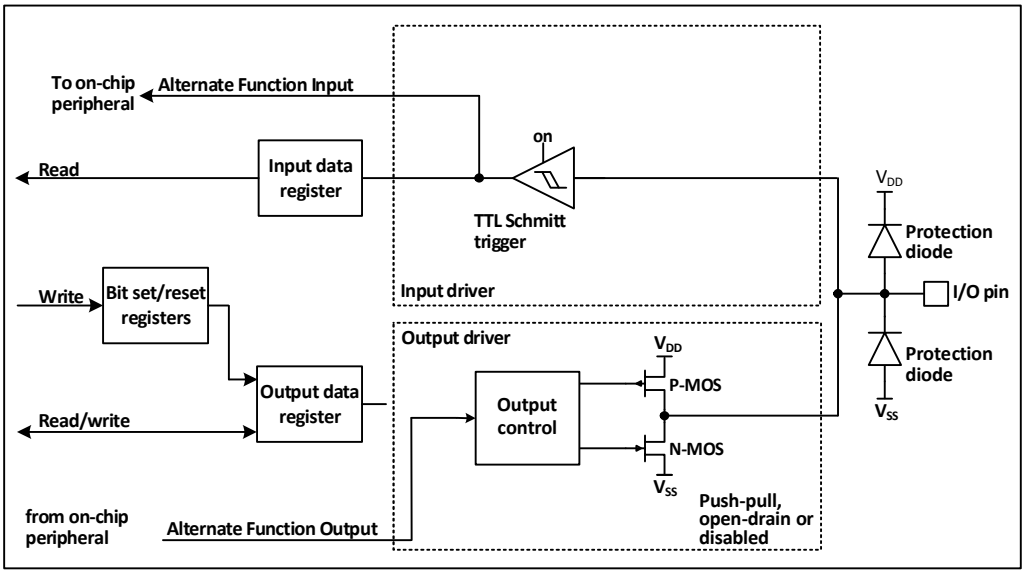
Figure 10-3 Output configuration structure block diagram of GPIO module



When the IO port is configured in output mode, a pair of MOS in the output driver can be configured in push-pull or open-drain mode as needed, without using the alternate function. The input-driven pull-up/pull-down resistor is disabled, the TTL Schmitt trigger is activated, and the level that appears on the IO pin will be sampled to the input data register at each HB clock, so reading the input data register will get the IO state, and in push-pull output mode, access to the output data register will get the last written value.

10.2.8 Alternate Function Configuration

Figure 10-4 Structure block diagram when GPIO module is multiplexed by other peripherals

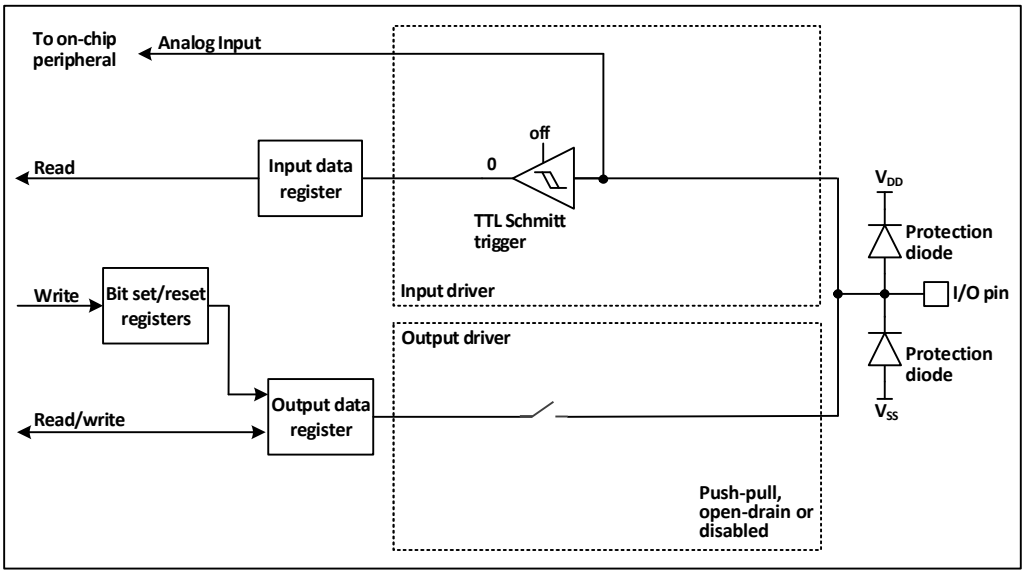


When the alternate function is enabled, the output driver is enabled and can be configured in open-drain or push-pull mode as needed, the Schmitt trigger is also turned on, the input and output lines of the alternate function are connected, but the output data register is disconnected, and the level that appears on the IO pin will be sampled to the input data register at each HB clock. In open-drain mode, reading the input data register will get the current state

of the IO port. In push-pull mode, reading the output data register will get the last written value.

10.2.9 Analog Input Configuration

Figure 10-5 Configuration structure block diagram of GPIO module as analog input



When the analog input is enabled, the output buffer is disconnected, the input of the Schmitt trigger in the input driver is disabled to prevent consumption on the IO port, the pull-up resistor is prohibited, and the read input data register will always be 0.

10.2.10 Peripheral GPIO Setting

The following table recommends the corresponding GPIO port configuration of each peripheral pin.

Table 10-1 Advanced-control timer (TIM1)

TIM1	Configuration	GPIO configuration
TIM1_CHx	Input capture channel x	Floating input
	Output compare channel x	Push-pull alternate output
TIM1_CHxN	Complementary output channel x	Push-pull alternate output
TIM1_BKIN	Break input	Floating input
TIM1_ETR	External trigger clock input	Floating input

Table 10-2 General-purpose timer (TIM2/3/4)

TIM2/3/4 pin	Configuration	GPIO configuration
TIM2/3/4_CHx	Input capture channel x	Floating input
	Output compare channel x	Push-pull alternate output
TIM2/3/4_ETR	External trigger clock input	Floating input

Table 10-3 Universal Synchronous/Asynchronous Receiver Transmitter (USART1/2/3/4)

USART pin	Configuration	GPIO configuration
USARTx_TX	Full-duplex mode	Push-pull alternate output
	Half-duplex synchronous mode	Open-drain alternate output
USARTx_RX	Full-duplex mode	Floating input or pull-up input

	Half-duplex synchronous mode	Not used
USARTx_CK	Synchronous mode	Push-pull alternate output
USARTx_RTS	Hardware flow control	Push-pull alternate output
USARTx_CTS	Hardware flow control	Floating input or pull-up input

Table 10-4 Serial Peripheral Interface (SPI1/2/3) module

SPI pin	Configuration	GPIO configuration
SPIx_SCK	Master mode	Push-pull alternate output
	Slave mode	Floating input
SPIx_MOSI	Full-duplex master mode	Push-pull alternate output
	Full-duplex slave mode	Floating input or pull-up input
	Simplex bidirectional data line/master mode	Push-pull alternate output
	Simplex bidirectional data line/slave mode	Not used
SPIx_MISO	Full duplex master mode	Floating input or pull-up input
	Full duplex slave mode	Push-pull alternate output
	Simplex bidirectional data line/master mode	Not used
	Simplex bidirectional data line/slave mode	Push-pull alternate output
SPIx_NSS	Hardware master or slave mode	Floating input or pull-up or pull-down input
	Hardware master mode/NSS output enable	Push-pull alternate output
	Software mode	Not used

Table 10-5 Inter IC Sound (I2C1/2) module

I2C pin	Configuration	GPIO configuration
I2Cx_SCL	I2C clock	Open-drain alternate output
I2Cx_SDA	I2C data	Open-drain alternate output

Table 10-6 Analog-to-Digital Converter (ADC)

ADC pin	GPIO configuration
ADC	Analog input

Table 10-7 USB PD controller

USBPD pin	GPIO configuration
USBPD_CC1/USBPD_CC2	Push-pull alternate output

Table 10-8 USB2.0 high-speed host device (USBHS) controller

USBHS pin	GPIO configuration
USBHS_DM/USBHS_DP	After the USB module is enabled, the multiplexed IO port will

	automatically connect to the internal USBHS transceiver.

Table 10-9 Other IO function settings

Pinout	Configuration function	GPIO configuration
RTC	RTC output	Push-pull alternate output
MCO	Clock output	Push-pull alternate output
EXTI	External interrupt input	Floating input or pull-up or pull-down input

10.2.11 Alternate Function Remapping GPIO Settings

10.2.11.1 XI/XO as GPIO port PD0/PD1

XI/XO can be used as GPIO PD0/PD1, which is realized by setting Remap Register 1 (AFIO_PCFR1).

10.3 Register Description

10.3.1 GPIO Registers

Unless otherwise specified, the GPIO registers must be operated in words (operate these registers in 32 bits).

Table 10-10 GPIO-related registers

Name	Access address	Description	Reset value
R32_GPIOA_CFGLR	0x40010800	PA port configuration register low	0x44444444
R32_GPIOB_CFGLR	0x40010C00	PB port configuration register low	0x44444444
R32_GPIOC_CFGLR	0x40011000	PC port configuration register low	0x44444444
R32_GPIOD_CFGLR	0x40011400	PD port configuration register low	0x44444444
R32_GPIOA_CFGHR	0x40010804	PA port configuration register high	0x44444444
R32_GPIOB_CFGHR	0x40010C04	PB port configuration register high	0x44444444
R32_GPIOC_CFGHR	0x40011004	PC port configuration register high	0x44444444
R32_GPIOD_CFGHR	0x40011404	PD port configuration register high	0x44444444
R32_GPIOA_INDR	0x40010808	PA port input data register	0x0000XXXX
R32_GPIOB_INDR	0x40010C08	PB port input data register	0x0000XXXX
R32_GPIOC_INDR	0x40011008	PC port input data register	0x0000XXXX
R32_GPIOD_INDR	0x40011408	PD port input data register	0x0000XXXX
R32_GPIOA_OUTDR	0x4001080C	PA port output data register	0x00000000
R32_GPIOB_OUTDR	0x40010C0C	PB port output data register	0x00000000
R32_GPIOC_OUTDR	0x4001100C	PC port output data register	0x00000000
R32_GPIOD_OUTDR	0x4001140C	PD port output data register	0x00000000
R32_GPIOA_BSHR	0x40010810	PA port set/reset register	0x00000000
R32_GPIOB_BSHR	0x40010C10	PB port set/reset register	0x00000000
R32_GPIOC_BSHR	0x40011010	PC port set/reset register	0x00000000
R32_GPIOD_BSHR	0x40011410	PD port set/reset register	0x00000000
R32_GPIOA_BCR	0x40010814	PA port reset register	0x00000000
R32_GPIOB_BCR	0x40010C14	PB port reset register	0x00000000
R32_GPIOC_BCR	0x40011014	PC port reset register	0x00000000

R32_GPIOD_BCR	0x40011414	PD port reset register	0x00000000
R32_GPIOA_LCKR	0x40010818	PA port lock configuration register	0x00000000
R32_GPIOB_LCKR	0x40010C18	PB port lock configuration register	0x00000000
R32_GPIOC_LCKR	0x40011018	PC port lock configuration register	0x00000000
R32_GPIOD_LCKR	0x40011418	PD port lock configuration register	0x00000000
R32_GPIOA_SPEED	0x4001081C	PA port speed register	0x00000000
R32_GPIOB_SPEED	0x40010C1C	PB port speed register	0x00000000
R32_GPIOC_SPEED	0x4001101C	PC port speed register	0x00000000
R32_GPIOD_SPEED	0x4001141C	PD port speed register	0x00000000

10.3.1.1 GPIO Configuration Register Low (GPIOx_CFGLR) (x=A/B/C/D)

Offset address: 0x00

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

Bit	Name	Access	Description	Reset value
[31:30] [27:26] [23:22] [19:18] [15:14] [11:10] [7:6] [3:2]	CNFy[1:0]	RW	(y=0-7), the configuration bits of port x, through which the corresponding ports are configured. When entering a mode (MODE=00b): 00: Analog input mode. 01: Floating input mode. 10: With pull-up mode. 11: Reserved. In output mode (MODE > 00b): 00: Universal push-pull output mode. 01: Universal open-leak output mode. 10: Alternate function push-pull output mode. 11: Alternate function open-leak output mode.	01b
[29:28] [25:24] [21:20] [17:16] [13:12] [9:8] [5:4] [1:0]	MODEy[1:0]	RW	(y=0-7), port x mode selection, configure the corresponding port through these bits. 00: Input mode. 01/10/11: Output mode.	00b

10.3.1.2 GPIO Configuration Register High (GPIOx_CFGHR) (x=A/B/C/D)

Offset address: 0x04

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

Bit	Name	Access	Description	Reset value
[31:30] [27:26] [23:22] [19:18] [15:14] [11:10] [7:6] [3:2]	CNFy[1:0]	RW	(y=8-15), the configuration bits for port x, by which the corresponding port is configured. When in input mode (MODE=00b): 00: Analog input mode; 01: Float input mode; 10: With pull-up and pull-down modes. 11: Reserved. In output mode (MODE>00b): 00: General-purpose push-pull output mode; 01: General-purpose open-drain output mode; 10: Alternate function push-pull output mode; 11: Alternate function open-drain output mode.	01b
[29:28] [25:24] [21:20] [17:16] [13:12] [9:8] [5:4] [1:0]	MODEy[1:0]	RW	(y=8-15), the mode bits of port x, by which the corresponding port is configured. 00: Input mode. 01/10/11: Output mode.	00b

10.3.1.3 Port Input Register (GPIOx_INDR) (x=A/B/C/D)

Offset address: 0x08

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IDR15	IDR14	IDR13	IDR12	IDR11	IDR10	IDR9	IDR8	IDR7	IDR6	IDR5	IDR4	IDR3	IDR2	IDR1	IDR0

Bit	Name	Access	Description	Reset value
[31:16]	Reserved	RO	Reserved	0
[15:0]	IDRy	RO	(y=0-15), port input data. These bits are read-only and can only be read out in 16-bit form. The value read out is the high and low state of the corresponding bit.	X

10.3.1.4 Port Output Register (GPIOx_OUTDR) (x=A/B/C/D)

Offset address: 0x0C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ODR15	ODR14	ODR13	ODR12	ODR11	ODR10	ODR9	ODR8	ODR7	ODR6	ODR5	ODR4	ODR3	ODR2	ODR1	ODR0

Bit	Name	Access	Description	Reset value
[31:16]	Reserved	RO	Reserved	0
[15:0]	ODRy	RW	For output mode: (y=0-15). the port outputs data. This data can only be manipulated in 16-bit form. the IO port externally outputs the values of these registers. For input modes with pull-down: 0: Pull-down input; 1: Pull-up input.	0

10.3.1.5 Port Set/Reset Register (GPIOx_BSHR) (x=A/B/C/D)

Offset address: 0x10

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BR15	BR14	BR13	BR12	BR11	BR10	BR9	BR8	BR7	BR6	BR5	BR4	BR3	BR2	BR1	BR0
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BS15	BS14	BS13	BS12	BS11	BS10	BS9	BS8	BS7	BS6	BS5	BS4	BS3	BS2	BS1	BS0

Bit	Name	Access	Description	Reset value
[31:16]	BRy	WO	(y=0-15), the corresponding OUTDR bits are cleared for these location bits; writing 0 has no effect. These bits can only be accessed in 16-bit format. If both BR and BS bits are set, the BS bit acts.	0
[15:0]	BSy	WO	(y=0-15), for which positional bits cause the corresponding OUTDR positional bits, and writing 0 has no effect. These bits can only be accessed in 16-bit format. If both BR and BS bits are set, the BS bit acts.	0

10.3.1.6 Port Reset Register (GPIOx_BCR) (x=A/B/C/D)

Offset address: 0x14

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

BR15	BR14	BR13	BR12	BR11	BR10	BR9	BR8	BR7	BR6	BR5	BR4	BR3	BR2	BR1	BR0

Bit	Name	Access	Description	Reset value
[31:16]	Reserved	RO	Reserved	0
[15:0]	BRy	WO	(y=0-15), the corresponding OUTDR bits are cleared for these location bits; writing 0 has no effect. These bits can only be accessed in 16-bit form.	0

10.3.1.7 Configuration Lock Register (GPIOx_LCKR) (x=A/B/C/D)

Offset address: 0x18

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LCK15	LCK14	LCK13	LCK12	LCK11	LCK10	LCK9	LCK8	LCK7	LCK6	LCK5	LCK4	LCK3	LCK2	LCK1	LCK0

Bit	Name	Access	Description	Reset value
[31:17]	Reserved	RO	Reserved	0
16	LCKK	RW	Lock key, which can be locked by writing a specific sequence, but it can be read out at any time. When it reads 0, it means that the lock is not in effect, and read 1 means that the lock is in effect. The write sequence of the lock key is: write 1-write 0-write 1-read 0-read 1, the last step is not necessary, but can be used to confirm that the lock key has been activated. Any error when writing a sequence does not activate a lock, and the value of LCK[15:0] cannot be changed when writing a sequence. After the lock is in effect, the configuration of the port can be changed only after the next reset.	0
[15:0]	LCKy	RW	(y=0-15), where 1 indicates that the configuration of the corresponding port is locked. These bits can only be changed before the LCKK is locked. Locked configurations refer to the configuration registers GPIOx_CFGLR and GPIOx_CFGHR.	0

Note: After the LOCK sequence is performed on the corresponding port bit, the configuration of the port bit cannot be changed until the next system reset.

10.3.1.8 Configuration Lock Register (GPIOx_LCKR) (x=A/B/C/D)

Offset address: 0x1C

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

SPEED15 [1:0]	SPEED14 [1:0]	SPEED13 [1:0]	SPEED12 [1:0]	SPEED11 [1:0]	SPEED10 [1:0]	SPEED9 [1:0]	SPEED8 [1:0]
15 14	13 12	11 10	9 8	7 6	5 4	3 2	1 0
SPEED7 [1:0]	SPEED6 [1:0]	SPEED5 [1:0]	SPEED4 [1:0]	SPEED3 [1:0]	SPEED2 [1:0]	SPEED1 [1:0]	SPEED0 [1:0]

Bit	Name	Access	Description	Reset value
[31:0]	SPEEDy[1:0]	RW	(y=0-15), port x speed selection, configure the corresponding port through these bits. When V_{DD}≥2.85V, CL=30pF: 00: The maximum frequency is 35MHz; 01: The maximum frequency is 45MHz; 10: The maximum frequency is 50MHz; 11: The maximum frequency is 55MHz. See the Electrical Characteristics section of the datasheet for details.	0

10.3.2 AFIO Registers

Unless otherwise specified, the AFIO registers must be operated in words (operate these registers in 32 bits).

Table 10-11 AFIO-related registers

Name	Access address	Description	Reset value
R32_AFIO_PCFR1	0x40010004	Remap register 1	0x00000000
R32_AFIO_EXTICR1	0x40010008	External interrupt configuration register 1	0x00000000
R32_AFIO_EXTICR2	0x4001000C	External interrupt configuration register 2	0x00000000
R32_AFIO_EXTICR3	0x40010010	External interrupt configuration register 3	0x00000000
R32_AFIO_EXTICR4	0x40010014	External interrupt configuration register 4	0x00000000
R32_AFIO_CR	0x40010018	Control register	0x00000000
R32_AFIO_GPIOA_AFLR	0x40010020	PA port alternate register low	0x00000000
R32_AFIO_GPIOA_AFHR	0x40010024	PA port alternate register high	0x00000000
R32_AFIO_GPIOB_AFLR	0x40010028	PB port alternate register low	0x00000000
R32_AFIO_GPIOB_AFHR	0x4001002C	PB port alternate register high	0x00000000
R32_AFIO_GPIOC_AFLR	0x40010030	PC port alternate register low	0x00000000
R32_AFIO_GPIOC_AFHR	0x40010034	PC port alternate register high	0x00000000
R32_AFIO_GPIOD_AFLR	0x40010038	PD port alternate register low	0x00000000
R32_AFIO_GPIOD_AFHR	0x4001003C	PD port alternate register high	0x00000000

10.3.2.1 Remap Register 1 (AFIO_PCFR1)

Offset address: 0x04

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

15	14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
PD0P	Reserved

D1_RM	

Bit	Name	Access	Description	Reset value
[31:27]	Reserved	RO	Reserved	0
[26:24]	SW_CFG[2:0]	WO	These bits are used to configure the IO ports for the SW function and the trace function. SDI is the debugging interface for accessing the core. It is always used as the SWD port after a system reset. 0xx: Enable SWD (SDI); 100: Disable SWD (SDI)as a GPIO function; Other: Invalid.	0
[23:16]	Reserved	RO	Reserved	0
15	PD0PD1_RM	RW	Pin PD0&PD1 remap bit, which can be read or written by the user. It controls whether the GPIO functions of PD0 & PD1 are remapped, i.e. PD0 & PD1 are mapped to XI&XO. 1: Pin is used as GPIO port; 0: Pin is used as crystal pin.	0
[14:0]	Reserved	RO	Reserved	0

10.3.2.2 External Interrupt Configuration Register 1 (AFIO_EXTICR1)

Offset address: 0x08

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

Bit	Name	Access	Description	Reset value
[31:16]	Reserved	RO	Reserved	0
[15:12] [11:8] [7:4] [3:0]	EXTIx[3:0]	RW	(x=0-3), external interrupt input pin configuration bit. Used to determine which pin the external interrupt pin is mapped to: 0000: xth pin of the PA pin; 0001: xth pin of the PB pin; 0010: xth pin of the PC pin; 0011: xth pin of the PD pin; Other: Reserved.	0000b

10.3.2.3 External Interrupt Configuration Register 2 (AFIO_EXTICR2)

Offset address: 0x0C

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

Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXTI7[3:0]				EXTI6[3:0]				EXTI5[3:0]				EXTI4[3:0]			

Bit	Name	Access	Description	Reset value
[31:16]	Reserved	RO	Reserved	0
[15:12] [11:8] [7:4] [3:0]	EXTIx[3:0]	RW	(x=4-7), external interrupt input pin configuration bit. Used to determine which port pin the external interrupt pin is mapped to: 0000: xth pin of the PA pin; 0001: xth pin of the PB pin; 0010: xth pin of the PC pin; 0011: xth pin of the PD pin; Other: Reserved.	0000b

10.3.2.4 External Interrupt Configuration Register 3 (AFIO_EXTICR3)

Offset address: 0x10

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXTI11[3:0]				EXTI10[3:0]				EXTI9[3:0]				EXTI8[3:0]			

Bit	Name	Access	Description	Reset value
[31:16]	Reserved	RO	Reserved	0
[15:12] [11:8] [7:4] [3:0]	EXTIx[3:0]	RW	(x=8-11), external interrupt input pin configuration bit. Used to determine which port pin the external interrupt pin is mapped to: 0000: xth pin of the PA pin; 0001: xth pin of the PB pin; 0010: xth pin of the PC pin; 0011: xth pin of the PD pin; Other: Reserved.	0000b

10.3.2.5 External Interrupt Configuration Register 4 (AFIO_EXTICR4)

Offset address: 0x14

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXTI15[3:0]				EXTI14[3:0]				EXTI13[3:0]				EXTI12[3:0]			

Bit	Name	Access	Description	Reset value

[31:16]	Reserved	RO	Reserved	0
[15:12] [11:8] [7:4] [3:0]	EXTIx[3:0]	RW	(x=12-15), external interrupt input pin configuration bit. Used to determine which port pin the external interrupt pin is mapped to: 0000: xth pin of the PA pin; 0001: xth pin of the PB pin; 0010: xth pin of the PC pin; 0011: xth pin of the PD pin; Other: Reserved.	0000b

10.3.2.6 Control Register (AFIO_CR)

Offset address: 0x1C

31	30	29	28	27	26	25	24	23	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													USBPDRISE [1:0]		USBPDHVT

Bit	Name	Access	Description	Reset value
[31:3]	Reserved	RO	Reserved	0
[2:1]	USBPDRISE[1:0]	RW	Tr adjustment of USBPDLV waveform, user adjustable: 00: Tr decreased by 11%; 01: Tr decreased by 5%; 10: Tr remained unchanged; 11: Tr increased by 7%.	0
0	USBPDHVT	RW	CC pin input channel threshold adjustment, used for PD communication USBPD CC pin high threshold input mode enable: 1: High threshold input, which can reduce power consumption during USBPD communication; 0: Normal GPIO threshold input.	0

10.3.2.7 Port Alternate Function Register Low (GPIOx_AFLR) (x=A/B/C/D)

Offset address: 0x20, 0x28, 0x30, 0x38

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

Bit	Name	Access	Description	Reset value

[31:28] [27:24] [23:20] [19:16] [15:12] [11:8] [7:4] [3:0]	AFRy[3:0]	RW	(y=0-7), alternate function selection of port x pin y: 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.	0

10.3.2.8 Port Alternate Function Register High (GPIOx_AFHR) (x=A/B/C/D)

Offset address: 0x24, 0x2C, 0x34, 0x3C

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

Bit	Name	Access	Description	Reset value
[31:28] [27:24] [23:20] [19:16] [15:12] [11:8] [7:4] [3:0]	AFRy[3:0]	RW	(y=8-15), alternate function selection of port x pin y: 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.	0

Chapter 11 Direct Memory Access Control (DMA)

Direct Memory Access (DMA) controllers provide a high-speed means of transferring data between a peripheral and memory or between memory and memory without CPU intervention, and data can be moved quickly through the DMA to conserve CPU resources for other operations.

Each channel of the DMA controller is dedicated to managing requests for memory access from one or more peripherals. There is also an arbiter to coordinate priorities between channels.

11.1 Main Features

- Multiple independent configurable channels
- Each channel is directly connected to dedicated hardware DMA request, and supports software trigger
- Support cyclic buffer management
- The priority of requests between multiple channels can be set by software programming (very high, high, medium and low level). When the priority settings are equal, it is determined by the channel number (the lower the channel number, the higher the priority)
- Support transmission from peripheral to memory, memory to peripheral, and memory to memory
- Flash memory, SRAM, peripheral SRAM and HB peripherals can all be used as the sources and targets of access
- Number of programmable data transfer byte: 65535 at most

11.2 Functional Description

11.2.1 DMA Channel Processing

1) Arbitration priority

DMA requests generated by multiple independent channels are logically or structurally input to the DMA controller, and currently only one channel request is answered. The arbitrator within the module selects the access of the peripheral / memory to be initiated according to the priority of the channel request.

In software management, applications can independently configure priority levels for each channel, including the very high, high, medium and low levels, by setting the PL[1:0] bit of the DMA_CFGRx register. When the software settings between the channels are the same, the modules will be selected according to the fixed hardware priority, and the lower channel number will have higher priority than the higher one.

2) DMA configuration

When the DMA controller receives a request signal, it accesses the requesting peripheral or memory to establish the peripheral or data transfer between the memory and the memory. It mainly includes the following three steps:

1) Data is taken from the memory address indicated by the peripheral data register or the current peripheral / memory address register, and the starting address of the first transmission is the peripheral base address or memory address specified by the DMA1_PADDRx or DMA1_MADDRx register.
2) The data is stored in the peripheral data register or the memory address indicated by the current peripheral / memory address register, and the starting address of the first transmission is the peripheral base address or memory

address specified in the DMA1_PADDRx or DMA1_MADDRx register.
3) Perform a decrement operation of the values in the DMA1_CNTRx register, which indicates the current number of outstanding operations.

Each channel has 3 DMA data transfer modes:

- Peripheral to memory (MEM2MEM=0, DIR=0)
- Memory to peripheral (MEM2MEM=0, DIR=1)
- Memory to memory (MEM2MEM=1)

Note: Memory-to-memory mode does not require a peripheral request signal, after configuring this mode (MEM2MEM=1), the channel is turned on (EN=1) to start data transfer. This method does not support cyclic mode.

The configuration process is as follows:
1) Set the first address of the peripheral register or the memory data address in the memory-to-memory mode (MEM2MEM=1) in the DMA1_PADDRx register. When an DMA request occurs, this address will be the source or destination address of the data transfer.
2) Set the memory data address in the DMA1_MADDRx register. When a DMA request occurs, the transmitted data will be read from or written to this address.
3) Set the amount of data to be transferred in the DMA1_CNTRx register. After each data transmission, this value decreases.
4) Set the channel priority in the PL [1:0] bit of the DMA1_CFGRx register.
5) Set the data transfer direction, loop mode, incremental mode of peripherals and memory, data width of peripherals and memory, transmission halfway, transmission completion, transmission error interrupt enable level in DMA1_CFGRx register.
6) Set the EN bit of the DMA1_CFGRx register and start the channel x.
Note: Control bits such as the data transfer direction (DIR), cycle mode (position), and peripheral and memory increment modes (MINC/PINC) in the DMA1_PADDRx, DMA1_MADDRx, DMA1_CNTRx, and DMA1_CFGRx registers can only be configured and written when the DMA channel is disabled.

3) Cycle mode

Set the CIRC location 1 of the DMA1_CFGRx register to enable the circular mode function of channel data transfer. In circular mode, when the number of data transfers becomes 0, the contents of the DMA1_CNTRx register are automatically reloaded to their initial values, the internal peripherals and memory address registers are also reloaded to the initial address values set by the DMA1_PADDRx and DMA1_MADDRx registers, and the DMA operation continues until the channel is closed or DMA mode is turned off.

4) DMA processing status

- Transfer more than half: corresponding to the HTIFx bit hardware setting in the DMA1_INTFR register. When the number of DMA transmissions is reduced to less than half of the initial setting value, more than half of the DMA transfer flag will be generated, and if HTIE is set in the DMA1_CFGRx register, an interrupt will occur. The hardware uses this flag to remind the application that it can prepare for a new round of data transfer.
- Transfer completed: corresponding to the TCIFx bit hardware setting in the DMA1_INTFR register. When the number of DMA transmissions is reduced to 0, the DMA transfer completion flag will be generated, and if TCIE is set in the DMA1_CFGRx register, an interrupt will occur.
- Transfer error: corresponding to the TEIFx bit hardware setting in the DMA1_INTFR register. Reading and writing a reserved address area will result in a DMA transmission error. At the same time, the module hardware

will automatically clear the EN bits of the DMA1_CFGRx register corresponding to the channel where the error occurred, and the channel is closed. If TEIE is set in the DMA1_CFGRx register, an interrupt will occur.

When querying the status of the DMA channel, the application can first access the GIFx bit of the DMA1_INTFR register, determine which channel has the DMA event, and then deal with the specific DAM event content of the channel.

11.2.2 Programmable Data Transmission Total Size/Data Bit Width/Alignment

The total amount of data transmitted by DMA in one round of each channel is programmable, up to 65535 times. The DMA1_CNTRx register indicates the number of transfers to be transmitted. In EN=0, the setting value is written, and after the EN=1 opens the DMA transmission channel, the register becomes read-only, and the value decreases after each transfer.

The transmission data values of peripherals and memory support the function of automatic increment of address pointer, and the pointer increment is programmable. The first transmitted data address they access is stored in the DMA1_PADDRx and DMA1_MADDRx registers. By setting the PINC bit or MINC location 1 of the DMA1_CFGRx register, you can turn on the peripheral address self-increment mode or the memory address self-increment mode, respectively. PSIZE[1:0] sets the peripheral address to take data size and address self-increase, and MSIZE[1:0] sets the memory address to take data size and address self-increase. There are 3 options: 8-bit, 16-bit, 32-bit. The specific data transfer methods are as follows:

Table 11-1 DMA transfer under different data bit width (PINC=MINC=1)

Source bit width	Target bit width	Transfer data Number	Source: address/data	Target: address/data	Transfer operation
8	8	4	0x00/B0 0x01/B1 0x02/B2 0x03/B3	0x00/B0 0x01/B1 0x02/B2 0x03/B3	● The source address increment is aligned with the bit width of the data set by the source, and the size of the value is equal to the bit width of the source data. ● The target address increment is aligned with the bit width of the target set data, and the size of the value is equal to the bit width of the target data. ● DMA transfer to the target side of the data based on the principle: data size is not enough to make up the high bit 0, data size overflow high bit removed ● Data storage mode: small end mode, low address stores low bytes, high address stores high bytes.
8	16	4	0x00/B0 0x01/B1 0x02/B2 0x03/B3	0x00/00B0 0x02/00B1 0x04/00B2 0x06/00B3	
8	32	4	0x00/B0 0x01/B1 0x02/B2 0x03/B3	0x00/000000B0 0x04/000000B1 0x08/000000B2 0x0C/000000B3	
16	8	4	0x00/B1B0 0x02/B3B2 0x04/B5B4 0x06/B7B6	0x00/B0 0x01/B2 0x02/B4 0x03/B6	
16	16	4	0x00/B1B0 0x02/B3B2 0x04/B5B4 0x06/B7B6	0x00/B1B0 0x02/B3B2 0x04/B5B4 0x06/B7B6	
16	32	4	0x00/B1B0	0x00/0000B1B0	

			0x02/B3B2 0x04/B5B4 0x06/B7B6	0x04/0000B3B2 0x08/0000B5B4 0x0C/0000B7B6	
32	8	4	0x00/B3B2B1B0 0x04/B7B6B5B4 0x08/BBBAB9B8 0x0C/BFBEBDBC	0x00/B0 0x01/B4 0x02/B8 0x03/BC	
32	16	4	0x00/B3B2B1B0 0x04/B7B6B5B4 0x08/BBBAB9B8 0x0C/BFBEBDBC	0x00/B1B0 0x02/B5B4 0x04/B9B8 0x06/BDBC	
32	32	4	0x00/B3B2B1B0 0x04/B7B6B5B4 0x08/BBBAB9B8 0x0C/BFBEBDBC	0x00/B3B2B1B0 0x04/B7B6B5B4 0x08/BBBAB9B8 0x0C/BFBEBDBC	

11.2.3 Dual Buffer Mode

Dual buffer mode is used for the channel of DMA1. In addition to having 2 memory pointers, the channel of double buffer mode still works the same as single buffer mode.

By setting the DOUBLE_MODE bit of the DMA1_CFGRx register to 1, the double buffer mode is enabled, the loop mode is automatically entered, and the memory address is automatically switched after completing a cycle (alternating between DMA1_MADDRx and DMA1_M1ADDRx).

The DMA controller swaps from one memory target to another at the end of each cycle, so that while the software is working on one memory area, the DMA transfer can fill or use a second memory area. Dual buffer channels work in both directions, as shown in the table below.

Table 11-3 Source and destination address registers in double buffer mode

DMA1_CFGRx.DIR	Direction	Source address	Destination address
0	Read from peripheral	DMA1_PADDRx	DMA1_MADDRx/DMA1_M1ADDRx
1	read from memory	DMA1_MADDRx/DMA1_M1ADDRx	DMA1_PADDRx

Note: Configuring memory-to-memory mode is disabled when dual-buffering mode is enabled.

11.2.4 DMA Request Mapping

The DMA controller provides 11 channels, each channel corresponds to multiple peripheral requests. By setting the corresponding DMA control bit in the corresponding peripheral register, the DMA function of each peripheral can be independently turned on or off. The specific corresponding relationship is shown in Figure 11-1.

Among them, the peripheral DMA channel can be remapped through the configuration bits of the R32_EXTEN_CTR register. The xx_TX_0/xx_RX_0 format is the default mapping, and the xx_TX_1/xx_RX_1 format is remapping.

Figure 11-1 DMA request mapping

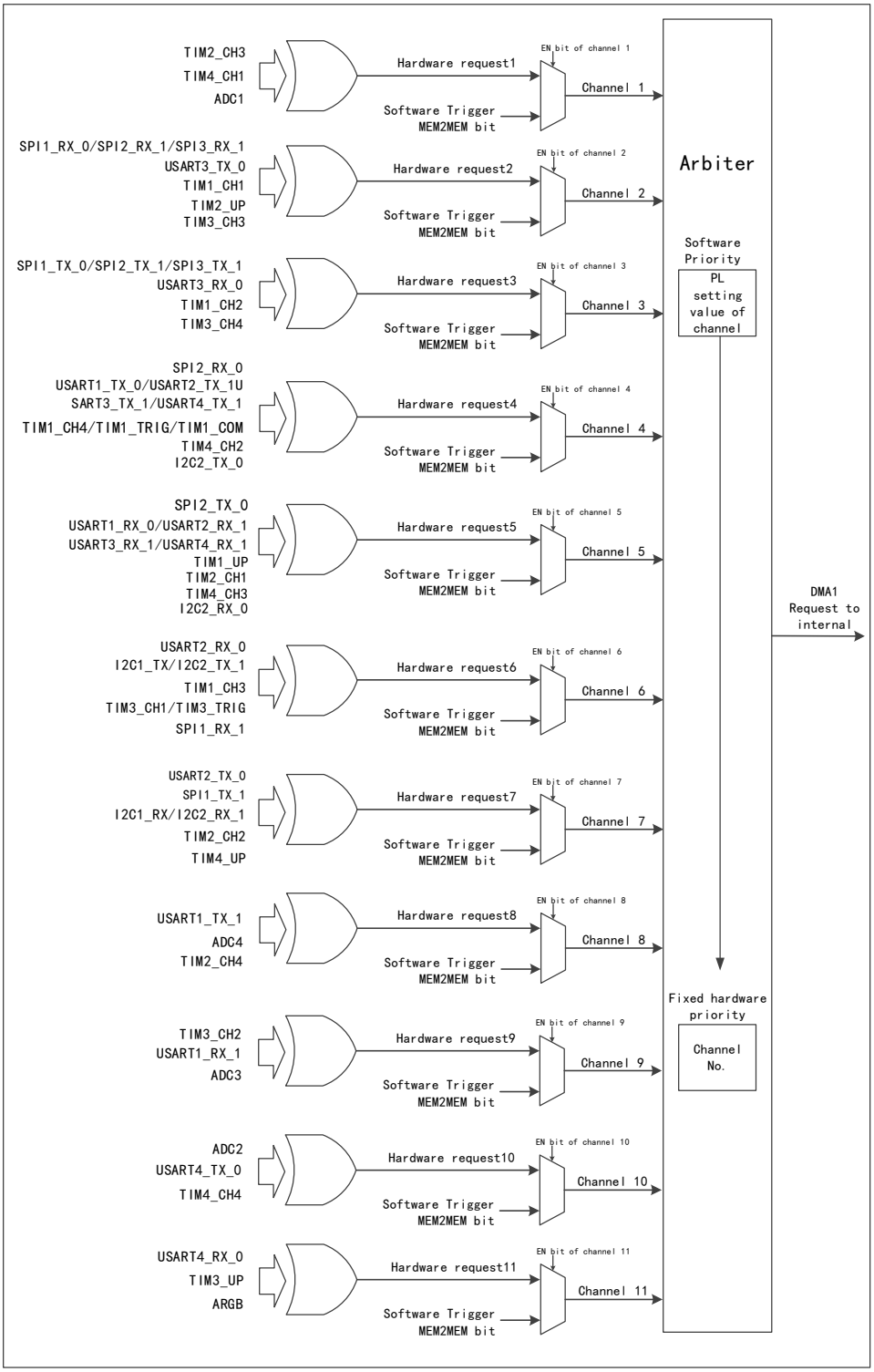


Table 11-2 Peripheral mapping table for each channel of DMA1

Peripheral	Channel 1	Channel 2	Channel 3	Channel 4	Channel 5	Channel 6	Channel 7
SPI1		SPI1_RX_0	SPI1_TX_0			SPI1_RX_1	SPI1_TX_1
SPI2		SPI2_RX_1	SPI2_TX_1	SPI2_RX_0	SPI2_TX_0		
SPI3		SPI3_RX_1	SPI3_TX_1				
USART1				USART1_T	USART1_R		

				X_0	X_0		
USART2				USART2_TX_1	USART2_RX_1	USART2_RX_0	USART2_TX_0
USART3		USART3_TX_0	USART3_RX_0	USART3_TX_1	USART3_RX_1		
USART4				USART4_TX_1	USART4_RX_1		
I2C1						I2C1_TX	I2C1_RX
I2C2				I2C2_TX_0	I2C2_RX_0	I2C2_TX_1	I2C2_RX_1
TIM1		TIM1_CH1	TIM1_CH2	TIM1_CH4 TIM1_TRIG TIM1_COM	TIM1_UP	TIM1_CH3	
TIM2	TIM2_CH3	TIM2_UP			TIM2_CH1		TIM2_CH2
TIM3		TIM3_CH3	TIM3_CH4			TIM3_CH1 TIM3_TRIG	
TIM4	TIM4_CH1			TIM4_CH2	TIM4_CH3		TIM4_UP
ADC1	ADC1						

Peripheral	Channel 8	Channel 9	Channel 10	Channel 11
ADC2			ADC2	
ADC3		ADC3		
ADC4	ADC4			
USART1	USART1_TX_1	USART1_RX_1		
USART4			USART4_TX_0	USART4_RX_0
TIM2	TIM2_CH4			
TIM3		TIM3_CH2		TIM3_UP
TIM4			TIM4_CH4	
ARGB				ARGB
SPI3	SPI3_RX_0	SPI3_TX_0		

11.3 Register Description

Table 11-4 DMA1 registers

Name	Access address	Description	Reset value
R32_DMA1_INTFR	0x40020000	DMA1 interrupt status register	0x00000000
R32_DMA1_INTFCR	0x40020004	DMA1 interrupt flag clear register	0x00000000
R32_DMA1_CFGR1	0x40020008	DMA1 channel 1 configuration register	0x00000000
R32_DMA1_CNTR1	0x4002000C	DMA1 channel 1 transfer data number register	0x00000000
R32_DMA1_PADDR1	0x40020010	DMA channel 1 peripheral address register	0x00000000
R32_DMA1_MADDR1	0x40020014	DMA channel 1 memory address register	0x00000000

R32_DMA1_M1ADDR1	0x40020018	DMA1 channel 1 memory address register 1	0x00000000
R32_DMA1_CFGR2	0x4002001C	DMA channel 2 configuration register	0x00000000
R32_DMA1_CNTR2	0x40020020	DMA channel 2 transfer data number register	0x00000000
R32_DMA1_PADDR2	0x40020024	DMA channel 2 peripheral address register	0x00000000
R32_DMA1_MADDR2	0x40020028	DMA channel 2 memory address register	0x00000000
R32_DMA1_M1ADDR2	0x4002002C	DMA1 channel 2 memory address register 1	0x00000000
R32_DMA1_CFGR3	0x40020030	DMA channel 3 configuration register	0x00000000
R32_DMA1_CNTR3	0x40020034	DMA channel 3 transfer data number register	0x00000000
R32_DMA1_PADDR3	0x40020038	DMA channel 3 peripheral address register	0x00000000
R32_DMA1_MADDR3	0x4002003C	DMA channel 3 memory address register	0x00000000
R32_DMA1_M1ADDR3	0x40020040	DMA1 channel 3 memory address register 1	0x00000000
R32_DMA1_CFGR4	0x40020044	DMA channel 4 configuration register	0x00000000
R32_DMA1_CNTR4	0x40020048	DMA channel 4 transfer data number register	0x00000000
R32_DMA1_PADDR4	0x4002004C	DMA channel 4 peripheral address register	0x00000000
R32_DMA1_MADDR4	0x40020050	DMA channel 4 memory address register	0x00000000
R32_DMA1_M1ADDR4	0x40020054	DMA1 channel 4 memory address register 1	0x00000000
R32_DMA1_CFGR5	0x40020058	DMA1 channel 5 configuration register	0x00000000
R32_DMA1_CNTR5	0x4002005C	DMA1 channel 5 transfer data number register	0x00000000
R32_DMA1_PADDR5	0x40020060	DMA1 channel 5 peripheral address register	0x00000000
R32_DMA1_MADDR5	0x40020064	DMA1 channel 5 memory address register	0x00000000
R32_DMA1_M1ADDR5	0x40020068	DMA1 channel 5 memory address register 1	0x00000000
R32_DMA1_CFGR6	0x4002006C	DMA1 channel 6 configuration register	0x00000000
R32_DMA1_CNTR6	0x40020070	DMA1 channel 6 transfer data number register	0x00000000
R32_DMA1_PADDR6	0x40020074	DMA1 channel 6 peripheral address register	0x00000000
R32_DMA1_MADDR6	0x40020078	DMA1 channel 6 memory address register	0x00000000
R32_DMA1_M1ADDR6	0x4002007C	DMA1 channel 6 memory address register 1	0x00000000
R32_DMA1_CFGR7	0x40020080	DMA1 channel 7 configuration register	0x00000000
R32_DMA1_CNTR7	0x40020084	DMA1 channel 7 transfer data number register	0x00000000
R32_DMA1_PADDR7	0x40020088	DMA1 channel 7 peripheral address register	0x00000000
R32_DMA1_MADDR7	0x4002008C	DMA1 channel 7 memory address register	0x00000000
R32_DMA1_M1ADDR7	0x40020090	DMA1 channel 7 memory address register 1	0x00000000
R32_DMA1_CFGR8	0x40020094	DMA1 channel 8 configuration register	0x00000000
R32_DMA1_CNTR8	0x40020098	DMA1 channel 8 transfer data number register	0x00000000
R32_DMA1_PADDR8	0x4002009C	DMA1 channel 8 peripheral address register	0x00000000
R32_DMA1_MADDR8	0x400200A0	DMA1 channel 8 memory address register	0x00000000
R32_DMA1_M1ADDR8	0x400200A4	DMA1 channel 8 memory address register 1	0x00000000
R32_DMA1_CFGR9	0x400200A8	DMA1 channel 9 configuration register	0x00000000
R32_DMA1_CNTR9	0x400200AC	DMA1 channel 9 transfer data number register	0x00000000
R32_DMA1_PADDR9	0x400200B0	DMA1 channel 9 peripheral address register	0x00000000
R32_DMA1_MADDR9	0x400200B4	DMA1 channel 9 memory address register	0x00000000
R32_DMA1_M1ADDR9	0x400200B8	DMA1 channel 9 memory address register 1	0x00000000
R32_DMA1_CFGR10	0x400200BC	DMA1 channel 10 configuration register	0x00000000
R32_DMA1_CNTR10	0x400200C0	DMA1 channel 10 transfer data number	0x00000000

		register	
R32_DMA1_PADDR10	0x400200C4	DMA1 channel 10 peripheral address register	0x00000000
R32_DMA1_MADDR10	0x400200C8	DMA1 channel 10 memory address register	0x00000000
R32_DMA1_M1ADDR10	0x400200CC	DMA1 channel 10 memory address register 1	0x00000000
R32_DMA1_CFGR11	0x400200D0	DMA1 channel 11 configuration register	0x00000000
R32_DMA1_CNTR11	0x400200D4	DMA1 channel 11 transfer data number register	0x00000000
R32_DMA1_PADDR11	0x400200D8	DMA1 channel 11 peripheral address register	0x00000000
R32_DMA1_MADDR11	0x400200DC	DMA1 channel 11 memory address register	0x00000000
R32_DMA1_M1ADDR11	0x400200E0	DMA1 channel 11 memory address register 1	0x00000000
R32_DMA1_EXTEM_INTFR	0x400200E4	DMA1 extended interrupt status register	0x00000000
R32_DMA1_EXTEM_INTFCR	0x400200E8	DMA1 extended interrupt flag clear register	0x00000000

11.3.1 DMA1 Interrupt Flag Register (DMA1_INTFR)

Offset address: 0x00

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TEIF8	HTIF8	TCIF8	GIF8	TEIF7	HTIF7	TCIF7	GIF7	TEIF6	HTIF6	TCIF6	GIF6	TEIF5	HTIF5	TCIF5	GIF5
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TEIF4	HTIF4	TCIF4	GIF4	TEIF3	HTIF3	TCIF3	GIF3	TEIF2	HTIF2	TCIF2	GIF2	TEIF1	HTIF1	TCIF1	GIF1

Bit	Name	Access	Description	Reset value
31/27/23/19/15/11/7/3	TEIFx	RO	Transmission error flag for channel x (x= 1/2/3/4/5/6/7/8): 1: A transmission error occurred on channel x. 0: There is no transmission error on channel x. The hardware is set, and the software writes the CTEIFx bit to clear this flag.	0
30/26/22/18/14/10/6/2	HTIFx	RO	Transmission more than half flag of channel x (x= 1/2/3/4/5/6/7/8): 1: More than half of the transmission events occurred on channel x. 0: No more than half of the transmission is on channel x. The hardware is set, and the software writes the CHTIFx bit to clear this flag.	0
29/25/21/17/13/9/5/1	TCIFx	RO	Transmission completion flags for channel x (x=1/2/3/4/5/6/7/8): 1: A transmission completion event was generated on channel x; 0: No transmission completion event on channel x. Hardware sets and software writes the CTCIFx bit to clear this flag.	0
28/24/20/16	GIFx	RO	Global interrupt flag for channel x (x=1/2/3/4/5/6/7/8):	0

/12/8/4/0			1: TEIFx or HTIFx or TCIFx was generated on channel x; 0: No TEIFx or HTIFx or TCIFx has been generated on channel x. Hardware sets and software writes the CGIFx bit to clear this flag.	

11.3.2 DMA1 Interrupt Flag Clear Register (DMA1_INTFCR)

Offset address: 0x04

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CTEIF8	CHTIF8	CTCIF8	CGIF8	CTEIF7	CHTIF7	CTCIF7	CGIF7	CTEIF6	CHTIF6	CTCIF6	CGIF6	CTEIF5	CHTIF5	CTCIF5	CGIF5
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CTEIF4	CHTIF4	CTCIF4	CGIF4	CTEIF3	CHTIF3	CTCIF3	CGIF3	CTEIF2	CHTIF2	CTCIF2	CGIF2	CTEIF1	CHTIF1	CTCIF1	CGIF1

Bit	Name	Access	Description	Reset value
31/27/23/19/15/11/7/3	CTEIFx	WO	Clear the transmission error flag for channel x (x=1/2/3/4/5/6/8): 1: Clear the TEIFx flag in the DMA1_INTFR register; 0: No effect.	0
30/26/22/18/14/10/6/2	CHTIFx	WO	Clear the transmit half flag for channel x (x=1/2/3/4/5/6/8): 1: Clear the HTIFx flag in the DMA1_INTFR register; 0: No effect.	0
29/25/21/17/13/9/5/1	CTCIFx	WO	Clear the transmission completion flag for channel x (x=1/2/3/4/5/6/7/8): 1: Clear the TCIFx flag in the DMA1_INTFR register; 0: No effect.	0
28/24/20/16/12/8/4/0	CGIFx	WO	Clear the global interrupt flag for channel x (x=1/2/3/4/5/6/7/8): 1: Clears the TEIFx/HTIFx/TCIFx/GIFx flags in the DMA1_INTFR register; 0: No effect.	0

11.3.3 DMA1 Channel x Configuration Register (DMA1_CFGRx) (x=1-11)

Offset address: 0x08 + (x-1)*20

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															FLAG_CUR_MEM
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DOUBLE_MOD	MEM2MEM	PL[1:0]		MSIZE[1:0]		PSIZE[1:0]		MINC	PINC	CIRC	DIR	TEIE	HTIE	TCIE	EN

E												

Bit	Name	Access	Description	Reset value
[31:17]	Reserved	RO	Reserved	0
16	FLAG_CUR_MEM	RW	Memory address selection setting, the hardware will automatically switch in double buffer mode: 1: M1ADDR (addressing using DMA_M1ADDRx pointers); 0: MADDR (addressing using DMA_MADDRx pointers). *Note: This bit is only used in double buffering mode. After the channel is enabled, this bit is equivalent to a status flag to indicate the storage area as the current target.*	0
15	DOUBLE_MODE	RW	Double buffering mode enabled: 1: Enable the double buffering mode, it will automatically enter the loop mode, and the memory address will be switched after one completion; 0: Turn off the double buffering mode.	0
14	MEM2MEM	RW	Memory to memory mode enable: 1: Enable memory-to-memory data transfer mode; 0: Not a memory-to-memory data transfer.	0
[13:12]	PL[1:0]	RW	Channel priority settings: 00: Low; 01: Medium; 10: High; 11: Very high.	00b
[11:10]	MSIZE[1:0]	RW	Memory address data width setting: 00: 8-bit; 01: 16-bit; 10: 32 bits; 11: Reserved.	00b
[9:8]	PSIZE[1:0]	RW	Peripheral address data width setting: 00: 8-bit; 01: 16-bit; 10: 32 bits; 11: Reserved.	00b
7	MINC	RW	Memory address increment mode enables: 1: Enable memory address increment operation; 0: Memory address remains unchanged operation.	0
6	PINC	RW	Peripheral address increment mode enables: 1: Enable memory address increment operation; 0: Peripheral address remains unchanged operation.	0
5	CIRC	RW	DMA channel cyclic mode enable: 1: Enable cyclic operation; 0: Perform a single operation.	0
4	DIR	RW	Data transmission direction: 1: Read from memory. 0: Read from the peripheral.	0
3	TEIE	RW	Transmission error interrupt enable control: 1: Enable transmission error interrupt. 0: Disable transmission error interrupt.	0

2	HTIE	RW	More than half of the transmission interruption enables control: 1: Enable transmission for more than half an interruption. 0: Disable transmission for more than half an interruption.	0
1	TCIE	RW	Transfer complete interrupt enable control: 1: Enable transmission completion interrupt; 0: Disable transmission completion interrupt.	0
0	EN	RW	Channel enable control: 1: Channel enabled; 0: Channel disabled. When a DMA transmission error occurs, it will be cleared to 0 automatically by hardware, and channel is disabled.	0

11.3.4 DMA1 Channel x Transfer Data Number Register (DMA1_CNTRx) (x=1-11)

Offset address: 0x0C + (x-1)*20

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

Bit	Name	Access	Description	Reset value
[31:16]	Reserved	RO	Reserved	0
[15:0]	NDT[15:0]	RW	Number of data transfers, range 0-65535. Indicating the remaining number of transfers (the register contents are decremented after each DMA transfer). When the channel is in loop mode, the contents of the register are automatically reloaded to the previously configured value.	0

Note: Indicating the current number of transfers. When the register content is 0, no data transfer occurs regardless of whether the channel is open or not.

11.3.5 DMA1 Channel x Peripheral Address Register (DMA1_PADDRx) (x=1-11)

Offset address: 0x10 + (x-1)*20

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

Bit	Name	Access	Description	Reset value
[31:0]	PA[31:0]	RW	The base address of the peripheral, which is used as the source or destination address of the peripheral data transmission. When PSIZE [1:0] = '01' (16-bit), the module automatically ignores bit0, and the operation address automatically aligns with 2 bytes; when PSIZE[1:0] ='10'	0

			(32-bit), the module automatically ignores bit [1:0], and the operation address automatically aligns with 4 bytes.	

Note: This register can only be changed when EN=0, and cannot be written when EN=1.

11.3.6 DMA1 Channel x Memory Address Register (DMA1_MADDRx) (x=1-11)

Offset address: 0x14 + (x-1)*20

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

Bit	Name	Access	Description	Reset value
[31:0]	MA[31:0]	RW	Memory data address, as the source or destination address for data transmission. When MSIZE[1:0] = '01' (16-bit), the module automatically ignores bit0, and the operation address automatically aligns with 2 bytes; when MSIZE [1:0] ='10' (32-bit), the module automatically ignores bit [1:0], and the operation address automatically aligns with 4 bytes.	0

Note: This register can only be changed when EN=0, and cannot be written when EN=1.

11.3.7 DMA1 Channel x Memory Address Register 1 (DMA1_M1ADDRx) (x=1-11)

Offset address: 0x18 + (x-1)*20

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
M1A[31:0]																															

Bit	Name	Access	Description	Reset value
[31:0]	M1A[31:0]	RW	Memory data address 1, as the source or destination address for data transmission. When MSIZE[1:0] = '01' (16-bit), the module automatically ignores bit0, and the operation address automatically aligns with 2 bytes; when MSIZE[1:0] ='10' (32-bit), the module automatically ignores bit [1:0], and the operation address automatically aligns with 4 bytes.	0

11.3.8 DMA1 Extended Interrupt Status Register (DMA1_EXTEM_INTFR)

Offset address: 0xE4

31	30	29	28	27	26	25	24	23	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				TEIF11	HTIF11	TCIF11	GIF11	TEIF10	HTIF10	TCIF10	GIF10	TEIF9	HTIF9	TCIF9	GIF9

Bit	Name	Access	Description	Reset value
[31:12]	Reserved	RO	Reserved	0
11/7/3	TEIFx	RO	Transmission error flag of channel x (x=9/10/11): 1: A transmission error occurred on channel x; 0: There is no transmission error on channel x. Set by hardware, software clears this flag by writing to the CTEIFx bit.	0
10/6/2	HTIFx	RO	Transmission over half flag of channel x (x=9/10/11): 0: There is no transmission over half on channel x; 1: A transmission over half event occurs on channel x. Set by hardware, software clears this flag by writing to the CHTIFx bit.	0
9/5/1	TCIFx	RO	Transmission completion flag of channel x (x=9/10/11): 1: A transmission completion event is generated on channel x; 0: There is no transmission completion event on channel x. Set by hardware, software writes the CTCIFx bit to clear this flag.	0
8/4/0	GIFx	RO	Global interrupt flag of channel x (x=9/10/11): 1: TEIFx or HTIFx or TCIFx occurred on channel x; 0: TEIFx or HTIFx or TCIFx did not occur on channel x. Set by hardware, software clears this flag by writing to the CGIFx bit.	0

11.3.9 DMA1 Extended Interrupt Flag Clear Register (DMA1_EXTEM_INTFCR)

Offset address: 0xE8

31	30	29	28	27	26	25	24	23	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				CTEIF 11	CHTIF 11	CTCIF 11	CGIF 11	CTEIF 10	CHTIF 10	CTCIF 10	CGIF 10	CTEIF 9	CHTIF 9	CTCIF 9	CGIF 9

Bit	Name	Access	Description	Reset value
[31:12]	Reserved	RO	Reserved	0
11/7/3	CTEIFx	WO	Clear the transmission error flag of channel x (x=9/10/11): 1: Clear the TEIFx flag in the DMA_INTFR register; 0: No effect.	0
10/6/2	CHTIFx	WO	Clear the half-transmission flag of channel x (x=9/10/11): 1: Clear the HTIFx flag in the DMA_INTFR register;	0

			0: No effect.	
9/5/1	CTCIFx	WO	Clear the transfer completion flag of channel x (x=9/10/11): 1: Clear the TCIFx flag in the DMA_INTFR register; 0: No effect.	0
8/4/0	CGIFx	WO	Clear the global interrupt flag of channel x (x=9/10/11): 1: Clear the TEIFx/HTIFx/TCIFx/GIFx flag in the DMA_INTFR register; 0: No effect.	0

Chapter 12 Analog-to-digital Converter (ADC)

The chip has built-in 4 groups of 12-bit high-speed analog/digital converters (ADC1/ADC2/ADC3/ADC4). The 4 groups of ADC provide a total of 4*12=48 external channel samples. ADC1 also provides 1 internal channel (ADC1_IN12). The internal reference voltage V_{REFINT} is connected to the ADC1_IN12 input channel.

The sampling rate of ADC1/ADC2/ADC3/ADC4 can reach 5Msps, supports programmable channel sampling time, can realize single, continuous, scan or intermittent conversion, and supports multiple ADC conversion modes. Providing an analog watchdog function allows very precise monitoring of one or more selected channels for monitoring channel signal voltages. Supports external event-triggered conversion, and trigger sources include internal signals of on-chip timers and external pins. Supports the use of DMA operations.

ADC1/ADC2/ADC3/ADC4 support scan mode. After the scan is completed, the round count can be output through GPIO (ADCS0/PA10, ADCS1/PA11, ADCS2/PA12). In matrix analog signal acquisition applications, the signals generated in this mode can be added with 8-to-1 analog switch chips CH448 to achieve 8 times the number of ADC channels, supporting automatic switching of up to 8*48=384 ADC channels.

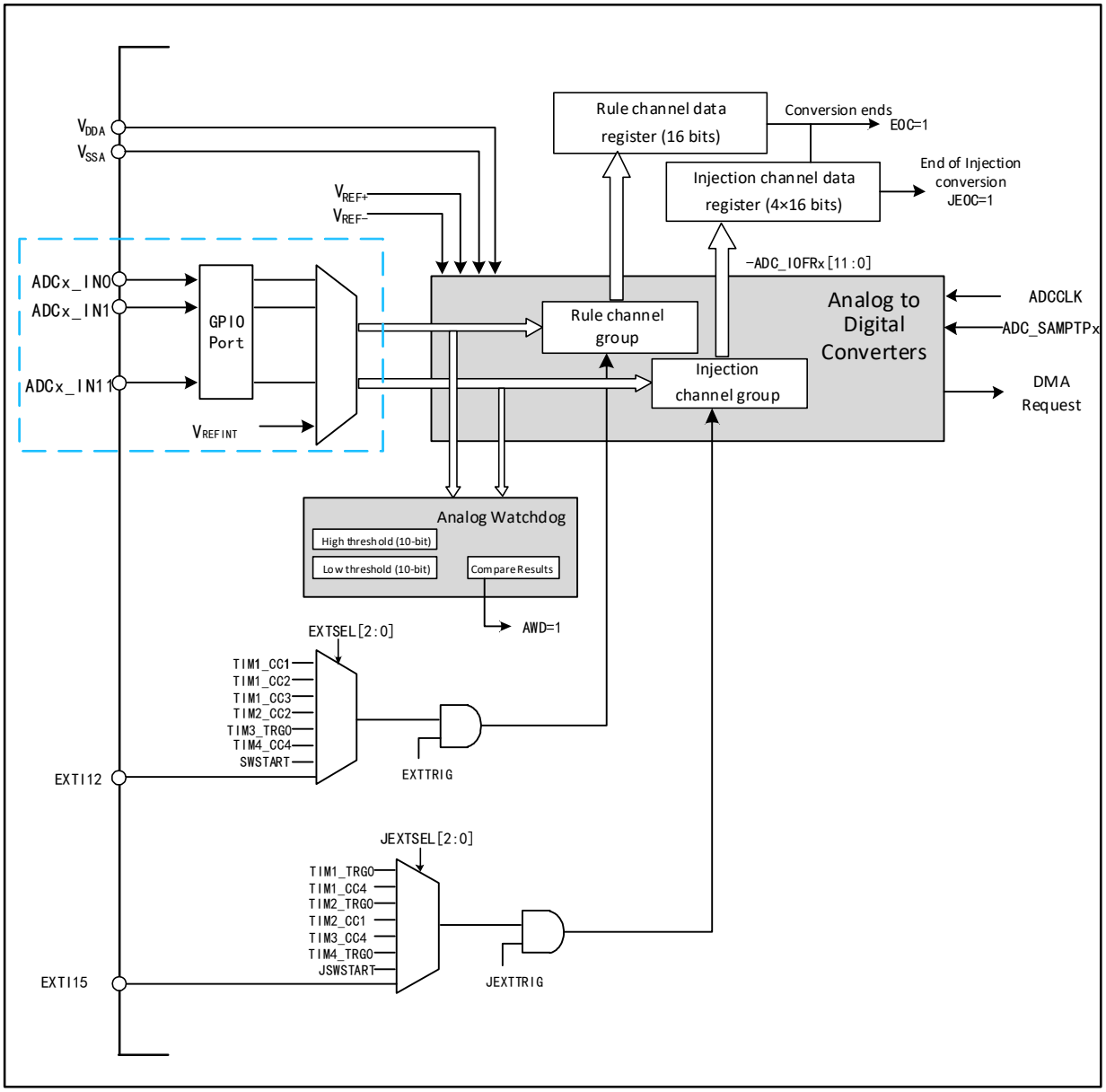
12.1 Main Features

- 12-bit resolution
- 48 external channels and 1 internal signal sources for sample
- Multiple sampling conversion modes for multiple channels: single, continuous, scan, trigger, discontinuous, etc.
- Data alignment mode: Left alignment, right alignment
- Support programmable sampling time
- Both regular conversion and injected conversion support external trigger
- Analog watchdog monitors the channel voltage
- Support dual mode
- Sampling rate up to 5Msps, supports multiple ADC conversion modes
- Support switching channels in advance to facilitate stable signal sampling
- Supports 4 groups of ADC cascades, equivalent to 20Msps sampling rate
- ADC channel input range: $V_{SS} \leq V_{IN} \leq V_{DDA}$

12.2 Functional Description

12.2.1 Module Structure

Figure 12-1 ADC module block diagram



Note: In the blue dotted box part of the figure, all external input channels of ADC1, ADC2, ADC3 and ADC4 are different, and only ADC1 provides the internal reference voltage V_{REFINT}.

12.2.2 ADC Configuration

1) Module power on

An ADON bit of 1 in the ADCx_CTLR2 register indicates that the ADC module is powered on. When the ADC module enters the power-on state (ADON=1) from the power-off mode (ADON=0), it needs to be delayed for a period of time for t_{STAB} to stabilize the module. After that, the ADON bit is written again as 1, which is used as the startup signal for the software to start the ADC conversion. By clearing the ADON bit to 0, you can terminate the current conversion and put the ADC module in power-off mode, where ADC consumes almost no power.

2) Sample clock

The module's register operation is based on the HCLK (HB bus) clock, and the clock reference ADCCLK of its conversion unit comes from HCLK or PLL. The frequency division is configured by the ADCPRE[1:0] field and PPRE2[2:0] field of the RCC_CFGR0 register, and the maximum cannot exceed 80MHz.

3) Channel configuration

Each ADC module provides 12 external channels for a total of 48 channels, of which ADC1 provides an additional internal channel. They can be configured into 2 transformation groups: rule groups and injection groups. To achieve a group conversion consisting of a series of conversions in any order on any number of channels.

Conversion group:

- Rule group: consists of up to 16 transformations. Regular channels and their conversion order are set in the ADCx_RSQRx register. The total number of translations in the rule group should be written to L[3:0] in the ADCx_RSQR1 register.
- Injection group: composed of up to 4 transformations. The injection channels and their conversion order are set in the ADC_ISQR register. The total number of translations in the injection group should be written to JL[1:0] in the ADC_ISQR register.

Note: If the ADCx_RSQRx or ADCx_ISQR register is changed during the conversion, the current conversion is terminated and a new startup signal will be sent to ADC to convert the newly selected group.

ADC1 provides 1 internal channel:

- V_{REFINT} internal reference voltage: Connect the ADC1_IN12 channel.

4) Calibration

Initialize the calibration register by writing the RSTCAL bit of the ADCx_CTLR2 register to 1, and wait for the RSTCAL hardware to clear to 0 to indicate that the initialization is complete. Set the CAL bit to start the calibration function. Once the calibration is completed, the hardware will automatically clear the CAL bit and store the calibration code in ADCx_RDATAR. After that normal conversion functions can begin. It is recommended to perform an ADC calibration when the ADC module is powered on.

Note: (1) Before starting the calibration, you must ensure that the ADC module is powered on (ADON=1) for at least 2 ADC clock cycles.

(2) During ADC calibration, the sampling time must be at least 1us.

5) Programmable sample time

ADC uses several ADCCLK cycles to sample the input voltage, and the number of sampling cycles of the channels can be changed by the SMPx[2:0] bit in the ADCx_SAMPTR1 and ADCx_SAMPTR2 registers. Each channel can be sampled at different times.

The total conversion time is calculated as follows:

$$T_{CONV} = \text{Sampling time} + 12.5T_{ADCCLK}$$

The sampling time of the ADC has 8 fixed gears to choose from by default. If the set gears cannot cover the application requirements, the programmable sampling time function can be used. After the programmable sampling time function is turned on, the adjustable sampling time range is 3.5~242.5 ADC clock cycles, and the sampling time adjustment step is 1 ADC clock cycle. The sampling time configuration is as follows:

UDP	SMPx[2:0]	Sampling time	Description
0	000	3.5	x is 0~13; When UDPx is 0, the sampling time is configured by the SMPx[2:0] register. When UDPx is 1, the sampling time is configured by the SMPUDP[7:0] register.
	001	5.5	
	010	7.5	
	011	13.5	
	100	20.5	
	101	30.5	
	110	45.5	
	111	68.5	
1	-	Programmable	

Notes on programmable sampling time function:

1) *Each channel can independently turn on the programmable sampling time function;*
2) *Each channel with programmable sampling time enabled shares the same sampling time configuration;*
3) *The sampling time during calibration is configurable, with a default of 242.5 ADC clock cycles;*
4) *The value of the configuration register SMPUDP[7:0] is equal to the sampling time + 12.5. For example, when SMPUDP=0xFF, the conversion cycle is 255 ADC clock cycles, and the sampling time is 255-12.5=242.5 ADC clock cycles;*
5) *When the programmable sampling time function is enabled, if SMPUDP<=16, the actual sampling time is 3.5 ADC clock cycles.*

ADC's regular channel conversion supports DMA function. The value converted by the regular channel is stored in a single data register ADCx_RDATAR. In order to prevent the data in the ADCx_RDATAR register from not being removed in time when converting multiple regular channels continuously, the DMA function of the ADC can be turned on. The hardware will generate a DMA request at the end of the conversion of the regular channel (EOC is set) and transfer the converted data from the ADCx_RDATAR register to the user-specified destination address.

After the channel configuration of the DMA controller module is completed, write the DMA bit of the ADCx_CTLR2 register to 1 to enable the DMA function of the ADC.

Note: The injection group transformation does not support the DMA feature.

6) Data alignment

The ALIGN bit in the ADCx_CTLR2 register selects the data storage alignment after ADC conversion. 12-bit data supports left and right alignment modes.

The data register ADCx_RDATAR of the regular group channel stores the actual converted 12-bit digital value, while the data register ADCx_IDATARx of the injected group channel is the value written after the actual converted data minus the offset of the ADCx_IOFRx register, so there are positive and negative SIGNB.

Figure 12-2 Data left alignment

Rule group data register

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

Injected group data register

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

Figure 12-3 Data right alignment

Rule group data register

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

Injected group data register

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

12.2.3 External Trigger Source

The start event of an ADC transformation can be triggered by an external event. If the EXTTRIG or JEXTTRIG bit of the ADCx_CTLR2 register is set, the conversion of the rule group or injection group channel can be triggered by external events, respectively. At this point, the configuration of the EXTSEL[2:0] and JEXTSEL[2:0] bits determines the external event sources of the rule group and the injection group.

Note: When the external trigger signal is selected for ADC rule or injection conversion, only its rising edge can initiate conversion.

Table 12-1 External trigger sources of regular group channel

EXTSEL[2:0]	Trigger source	Type
000	CC1 event of timer 1	Internal signal from on-chip timers
001	CC2 event of timer 1	
010	CC3 event of timer 1	
011	CC2 event of timer 2	
100	TRGO event of timer 3	
101	CC4 event of timer 4	
110	EXTI line 12	From external pin
111	SWSTART position 1 software trigger	Software control bit

Table 12-2 External trigger sources of injected group channel

JEXTSEL[2:0]	Trigger source	Type
000	TRGO event of timer 1	Internal signal from on-chip timers
001	CC4 event of timer 1	
010	TRGO event of timer 2	
011	CC1 event of timer 2	
100	CC4 event of timer 3	
101	TRGO event of timer 4	
110	EXTI line 15	From external pin
111	JSWSTART position 1 software trigger	Software control bit

12.2.4 Conversion Mode

Table 12-3 Conversion mode combination

ADCx_CTLR1 and ADCx_CTLR2 register control bits					ADC conversion mode
CONT	SCAN	DISCEN/JDISCEN	IAUTO	Start event	
0	0	0	0	ADON bit set to 1	Single channel mode: A regular channel performs a single conversion.
				External	Single channel mode: A regular channel or a

				trigger mode	channel of an injection channel performs a single conversion.
	1	0	0	ADON bit set to 1 or external trigger mode	Single scan mode: performs a single conversion on all selected regular group channels (ADCx_RSQRx) or all injection group channels (ADCx_ISQR) one by one in sequence. Trigger injection method: during the rule group channel conversion, all the transformations of the injection group channel can be inserted, and then the rule group channel conversion can be continued; however, the rule group channel transformation will not be inserted when the conversion is injected into the group channel.
			1	ADON bit set to 1 or external trigger mode	Single scan mode: performs a single conversion on all selected regular group channels (ADCx_RSQRx) or all injection group channels (ADCx_ISQR) one by one in sequence. Automatic injection mode: after the rule group channel is converted, the injection group channel is automatically converted. *Note: the external trigger signal of the injection channel is not allowed during the conversion process.*
	0	1 (DISCEN and JDISCEN cannot be 1 at the same time)	0	External trigger mode	Single break mode: each time an event is started, a short sequence of channel number conversions (the number defined by DISCNUM [2:0]) is performed until all selected channel conversions are completed. *Note: Rule group and injection group select this mode control bit as DISCEN and JDISCEN respectively. Discontinuous mode cannot be configured for both rule group and injection group. Discontinuous mode can only be used for one set of transformations.*
			1	-	Disable such mode.
	1	1	X	-	No such mode.
1	0	0	0	ADON bit set to 1 or external trigger mode	Continuous single channel/scan mode: repeat a new round of conversion at the end of each round until the CONT is cleared 0.
	1	0	0		
			1		

Note: The external trigger events of the rule group and the injection group are different, and the 'ADON' bit can

only start the rule group channel transformation, so the start events of the rule group and injection group channel transformation are independent.

1) Single channel conversion mode

In this mode, only one conversion is performed for the current 1 channel. This mode performs conversion on the channels sorted first in the rule group or injection group, where it can be started by setting ADON position 1 of the ADCx_CTLR2 register (for regular channels only) or by external triggering (for regular channels or injection channels). Once the conversion of the selected channel is completed:

If the rule group channel is converted, the conversion data is stored in the 16-bit ADCx_RDATAR register, the EOC flag is set, and if the EOCIE bit is set, the ADC interrupt will be triggered.

If the injection group channel is converted, the conversion data is stored in the 16-bit ADCx_IDATAR1 register, the EOC and JEOC flags are set, and if the JEOCIE or EOCIE bit is set, the ADC interrupt will be triggered.

2) Single scan mode conversion

Enter ADC scan mode by setting the scan bit of the ADCx_CTLR1 register to 1. This mode is used to scan a set of analog channels and perform a single conversion one by one for all channels selected by the ADCx_RSQRx register (for regular channels) or ADCx_ISQR (for injection channels). When the current channel conversion ends, the next channel of the same group is automatically converted.

In the scanning mode, according to the state of JAUTO bits, it can be divided into trigger injection mode and automatic injection mode.

- Trigger injection

The JAUTO bit is 0, when the trigger event of the injection group channel conversion occurs in the process of scanning the rule group channel, the current conversion is reset, and the sequence of the injection channel is carried out in a single scan mode. After all the selected injection group channel scan conversion is completed, the last interrupted rule group channel conversion is restored.

If the start event of the regular channel occurs when scanning the channel sequence of the injection group, the conversion of the injection group will not be interrupted, but the conversion of the rule sequence will be performed after the conversion of the injection sequence is completed.

Note: when using triggered injection transformations, you must ensure that the interval between triggered events is longer than the injection sequence. For example, if it takes 28 ADCCLK to complete the conversion of the injection sequence, the minimum time between events to trigger the injection channel is 29 ADCCLK.

- Automatic injection

The JAUTO bit is 1, and after scanning all the channel translations selected by the rule group, the channel selected by the injection group is converted automatically. This method can be used to convert up to 20 conversion sequences in ADCx_RSQRx and ADCx_ISQR registers.

In this mode, the external trigger (JEXTTRIG=0) of the injection channel must be disabled.

Note: When the ADC clock pre-division factor (ADCPRE[1:0]) is 4 to 8, 1 ADCCLK interval is automatically inserted when switching from regular conversion to injection sequence or from injection conversion to regular sequence; when the ADC clock pre-division factor is 2, there is a delay of 2 ADCCLK intervals.

3) Single discontinuous mode conversion

Enter the break mode of the rule group or injection group by setting the DISCEN or JDISCEN bit of the ADCx_CTLR1 register to 1. This mode distinguishes scanning a complete set of channels in the scan mode, but divides a group of channels into multiple short sequences, and each external trigger event will perform a scan conversion of a short sequence.

The length of the short sequence n (n < = 8) is defined in the DISCNUM[2:0] of the ADCx_CTLR1 register. When DISCEN is 1, it is the discontinuous mode of the rule group, and the total length to be converted is defined in the L[3:0] of the ADCx_RSQR1 register. When JDISCEN is 1, it is the discontinuous mode of the injection group, and the total length to be converted is defined in the JL[1:0] of the ADCx_ISQR register. Both rule group and injection group cannot be set to discontinuous mode at the same time.

Example of rule group discontinuity mode:
DISCEN=1, DISCNUM[2:0] = 3, L[3:0] = 8, to be switched channel = 1, 3, 2, 5, 8, 4, 10, 6.
The first external trigger: the conversion sequence is: 1, 3, 2, and EOC events are generated at the same time..
The second external trigger: the conversion sequence is: 5, 8, 4, and EOC events are generated at the same time..
The third external trigger: the conversion sequence is: 10, 6, and EOC events are generated at the same time.
The 4th external trigger: the conversion sequence is: 1, 3, 2, and EOC events are generated at the same time..

Example of injection group discontinuity mode:
JDISCEN=1, JL[1:0] = 3, Channel to be changed = 1, 3, 2.
The 1st external trigger: the conversion sequence is: 1.
The 2nd external trigger: conversion sequence is: 3.
The 3rd external trigger: the conversion sequence is: 2, and both EOC and JEOC events are generated.
The 4th external trigger: conversion sequence is: 1.

Note:
1. When a rule group or injection group is converted in discontinuous mode, the conversion sequence does not automatically start from scratch at the end of the conversion sequence. When all subgroups are converted, the next trigger event initiates the conversion of the first subgroup.
2. You cannot use both automatic injection (JAUTO=1) and discontinuous mode.
3. Discontinuous mode cannot be set for both rule groups and injection groups, and can only be used for one set of transformations.
4. In the intermittent mode of the injection group, the number of injection channels to be converted after external triggering is 1.

4) Continuous conversion

In continuous conversion mode, another conversion is started as soon as the current ADC conversion is completed, and the conversion does not stop on the last channel of the selection group, but continues again from the first channel of the selection group. Boot events in this mode include external trigger events and ADON position 1. After setting boot, CONT position 1 needs to be set.

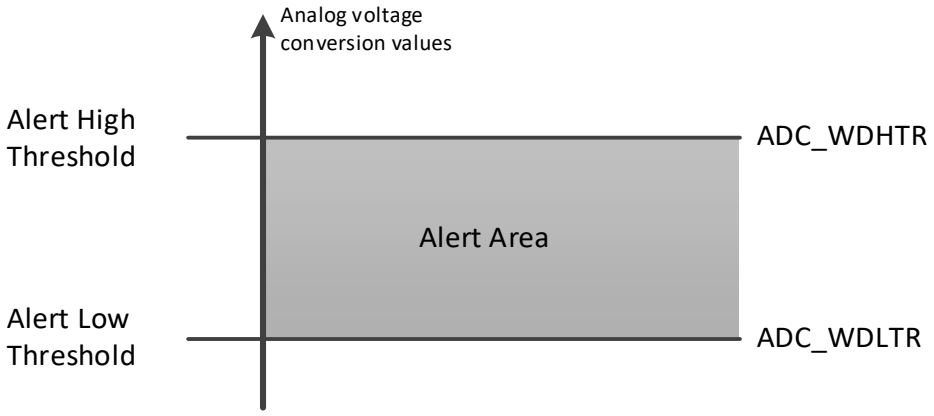
If a regular channel is converted, the converted data is stored in the ADCx_RDATAR register, the end of the conversion indicates that the EOC is set, and if EOCIE is set, an interrupt is generated.

If an injection channel is converted, the converted data is stored in the ADCx_IDATARx register, the end of injection conversion flag JEOC is set, and an interrupt is generated if JEOCIE is set.

12.2.5 Analog Watchdog

If the analog voltage converted by the ADC is below the low threshold or above the high threshold, the AWD analog watchdog status bit is set. The threshold settings are located in the least significant 12 bits of the ADCx_WDHTR and ADCx_WDLTR registers. Allow the corresponding interrupt to be generated by setting the AWDIE bit in the ADCx_CTLR1 register.

Figure 12-4 Analog watchdog threshold area



Configure the AWDSGL, AWDEN, JAWDEN, and AWDCH[4:0] bits of the ADCx_CTLR1 register to select the channel for analog watchdog alert, see the following table for the relationship:

Table 12-4 Analog Watchdog channel selection

Analog Watchdog alert channel	ADC_CTLR1 register control bit			
	AWDSGL	AWDEN	JAWDEN	AWDCH[4:0]
No alert	Ignore	0	0	Ignore
All injection channels	0	0	1	Ignore
All rule channels	0	1	0	Ignore
All injection and rule channels	0	1	1	Ignore
Single injected channel	1	0	1	Determine channel No.
Single regular channel	1	1	0	Determine channel No.
Single injected and regular channel	1	1	1	Determine channel No.

12.2.6 Dual ADC Mode

In an ADC module chip with 2 ADCs, 2 ADCs can be used together to achieve dual ADC mode. In dual ADC mode, ADC1 is the master ADC and ADC2 is the slave ADC. By configuring DUALMOD[3:0] in ADC1_CTLR1 to select the mode, ADC1 and ADC2 trigger alternately or synchronously trigger conversion.

The dual ADC mode for ADC3 and ADC4 is used in the same way as for ADC1 and ADC2.

Note: In the dual ADC mode, when selecting an external trigger event, the user must enable the external trigger of

the master-slave ADC and set the master ADC to the corresponding trigger and the slave ADC to software trigger to prevent unnecessary triggering from causing the slave ADC to convert.

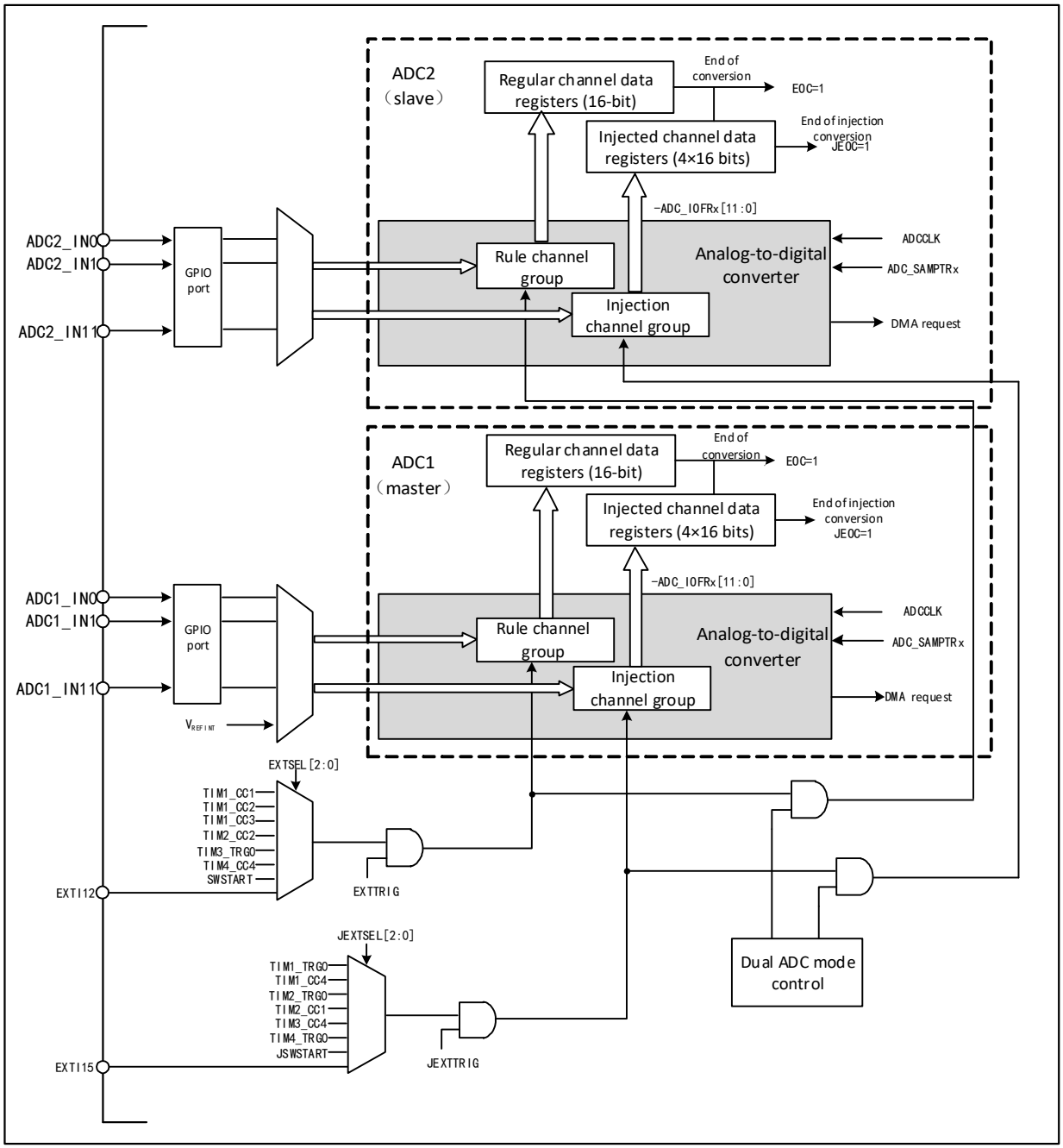
The following possible modes can be achieved through configuration:

- Independent mode
- Synchronous injection mode
- Synchronization rule mode
- Fast alternating mode
- Slow alternating mode
- Alternate trigger mode
- Synchronous rule mode + synchronous injection mode
- Synchronous rule mode + alternate trigger mode
- Synchronous injection mode + fast alternating mode
- Synchronous injection mode + slow alternating mode

Note: 1. In dual ADC mode, in order for the main data register to read the conversion data from the slave ADC, the DMA bit needs to be enabled.

2. Each ADC has independent DMA function.

Figure 12-5 Dual ADC block diagram



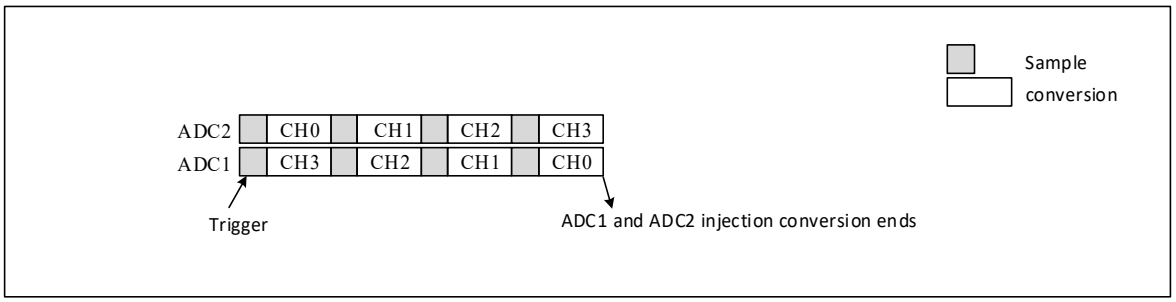
1) Independent mode

In this mode, the dual ADCs work asynchronously and independently of each other.

2) Synchronous injection mode

This mode is used to convert an injection channel group, set JEXTSEL[2:0] in ADC1_CTLR2 to select the trigger source, and it will also be used to trigger ADC2 synchronously. When the conversion is completed, the converted data are stored in ADCx_IDATARx of ADCx respectively, and if any ADC interrupt is enabled, a JEOC interrupt will be generated after the conversion is completed.

Figure 12-6 4-channel synchronous injection conversion



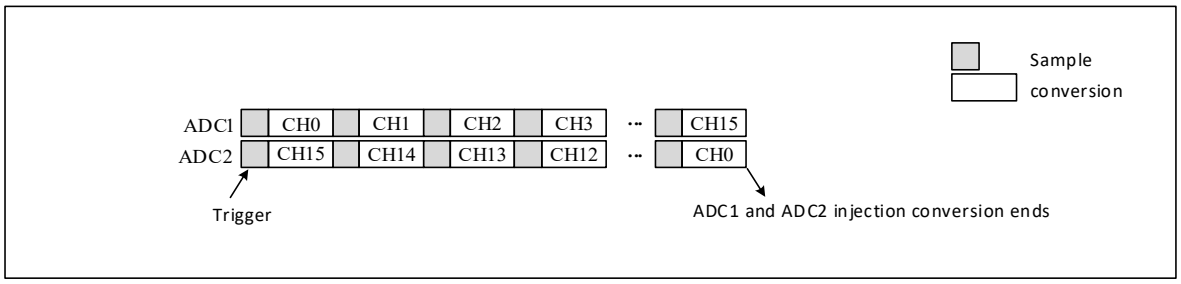
Note: 1. The conversion channels of ADC1 and ADC2 should not overlap at the same time.

2. In synchronous mode, ADC1 and ADC2 should have conversion sequences of the same length or the length of the longer conversion sequence between the two should be less than the trigger time interval to ensure that the conversion of both sequences can be completed every time the trigger is triggered.

3) Synchronization rule mode

This mode is used to convert the regular channel sequence. Set EXTSEL[2:0] in ADC1_CTLR2 to select the trigger source. It will also be used to trigger ADC2 synchronously. When the conversion is completed, a 32-bit DMA transfer request will be generated to transfer the contents of the data register ADC1_RDATAR to SRAM. The high 16 bits contain the ADC2 conversion data, and the low 16 bits contain the ADC1 conversion data. If either ADC interrupt is enabled, an EOC interrupt will be generated.

Figure 12-7 16-channel synchronous rule conversion



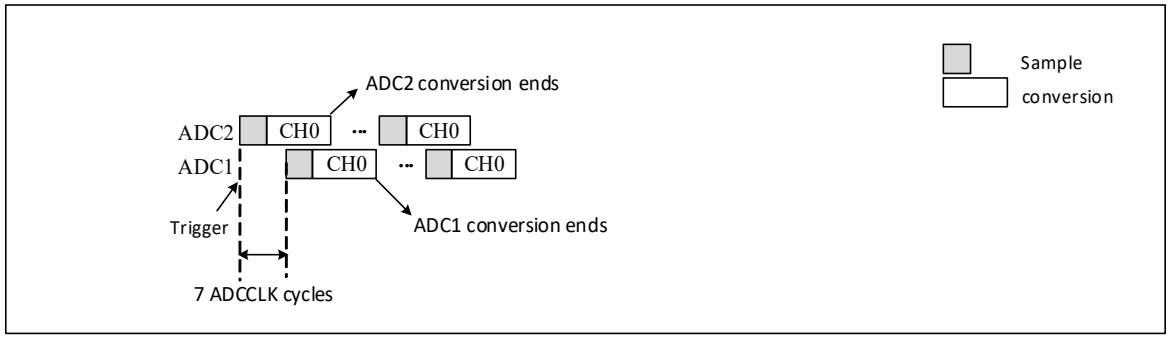
Note: 1. The conversion channels of ADC1 and ADC2 should not overlap at the same time.

2. In synchronous mode, ADC1 and ADC2 should have conversion sequences of the same length or the length of the longer conversion sequence between the two should be less than the trigger time interval to ensure that the conversion of both sequences can be completed every time the trigger is triggered.

4) Fast alternating mode

This mode is only applicable to rule channels (often only one channel). Set EXTSEL[2:0] in ADC1_CTLR2 to select the trigger source. When the trigger is generated, ADC2 will start conversion immediately, and ADC1 will start conversion after delaying 7 ADC clock cycles. If the continuous mode is enabled (CONT is set), the 2 ADCs will continuously convert the regular channels alternately. If the interrupt is enabled, ADC1 will generate an EOC interrupt. If DMA is enabled at the same time, a 32-bit DMA transfer request will be generated to transfer the contents of the data register ADC1_RDATAR to SRAM. The high 16 bits contain ADC2 conversion data, and the low 16 bits contain ADC1 conversion data.

Figure 12-8 Single channel fast alternating continuous conversion

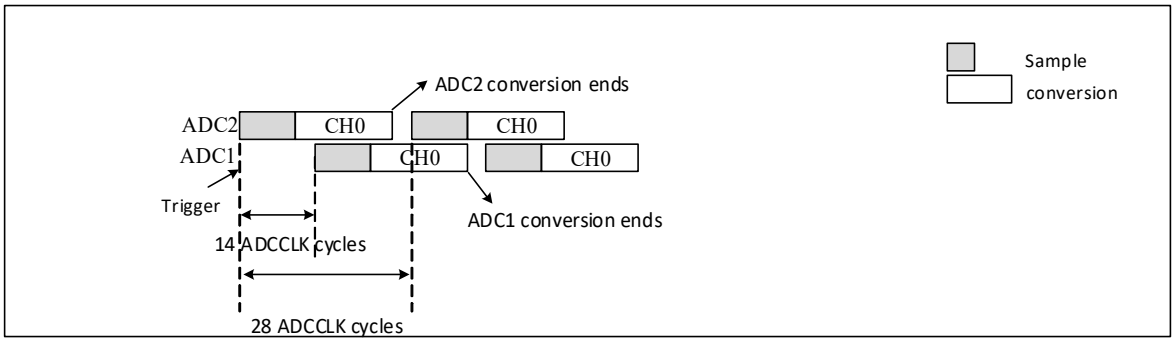


Note: The sampling time should be less than 7 ADC clock cycles to avoid the problem of overlapping sampling periods when ADC1 and ADC2 sample the same channel.

5) Slow alternating mode

This mode is only available for rule channels and only one channel. Set EXTSEL[2:0] in ADC1_CTLR2 to select the trigger source. ADC2 will start conversion immediately after the trigger is generated. ADC1 will start conversion after delaying 14 ADC clock cycles. ADC2 will start again after another delay of 14 ADC clock cycles, and so on. If the interrupt is enabled, ADC1 will generate an EOC interrupt. If DMA is enabled at the same time, a 32-bit DMA transfer request will be generated to transfer the contents of the data register ADC1_RDATAR to SRAM. The high 16 bits contain ADC2 conversion data, and the low 16 bits contain ADC1 conversion data.

Figure 12-9 Single channel slow alternating continuous conversion



Note: 1. The sampling time should be less than 14 ADC clock cycles to avoid overlapping with the next sampling cycle;

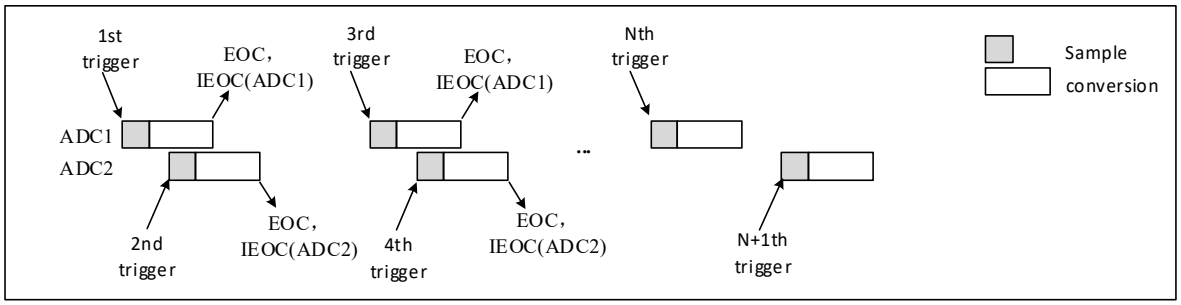
2. A new ADC2 conversion is automatically initiated after 28 ADC clock cycles;

3. The CONT bit does not need to be set.

6) Alternate trigger mode

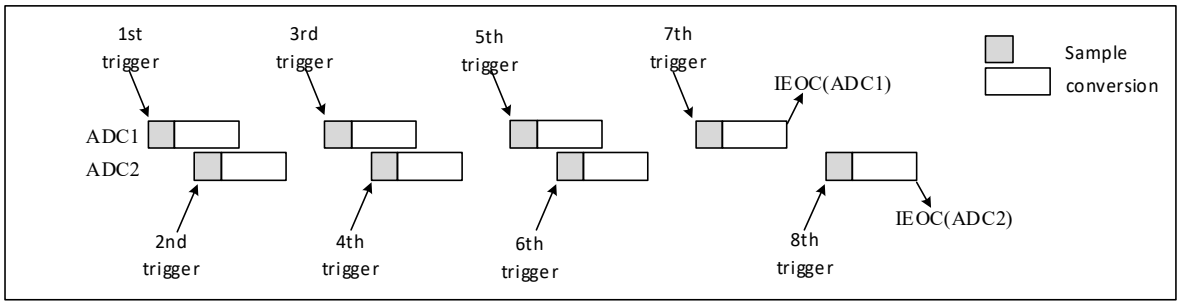
This mode is only applicable to the injection channel group. Set JEXTSEL[2:0] in ADC1_CTLR2 to select the trigger source. When the first trigger event occurs, all injection channels on ADC1 are converted, and when the second trigger event occurs, all injection channels on ADC2 are converted. The cycle repeats in sequence. If the interrupt is enabled, a JEOC interrupt will be generated after the conversion of all injection channels of ADC1 is completed, and a JEOC interrupt will be generated after the conversion of all injection channels of ADC2 is completed.

Figure 12-10 Each ADC injection channel group alternately triggers conversions



If the injection discontinuous mode is used at the same time, when the first trigger event occurs, the first injection channel on ADC1 is converted, and when the second trigger event occurs, the first injection channel on ADC2 is converted. And so on, back and forth. At the same time, if the interrupt is enabled, a JEOC interrupt will be generated when the conversion of all injection channels of ADC1 is completed, and a JEOC interrupt will be generated when the conversion of all injection channels of ADC2 is completed.

Figure 12-11 Each ADC injection channel alternately triggers conversions in discontinuous mode



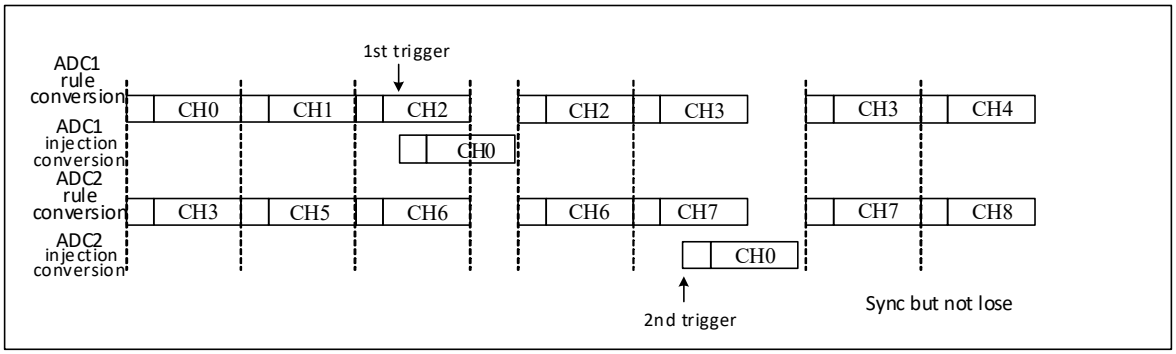
7) Synchronous rule mode + synchronous injection mode

In this mode, the synchronous conversion of rule groups can be interrupted to initiate the synchronous conversion of injection groups. Additionally, in this mode, you must ensure that the conversion sequences have the same duration, or that the trigger interval is longer than the duration of the longer of the two sequences.

8) Synchronous rule mode + alternate trigger mode

In this mode, the synchronous transition of rule sets can be interrupted to initiate an alternate-triggered transition of the injection group. When an injection event occurs, the alternate-triggered transition begins immediately. If a synchronous rule transition is in progress, all ADC rule transitions are paused and resume synchronously once the injection transition is complete.

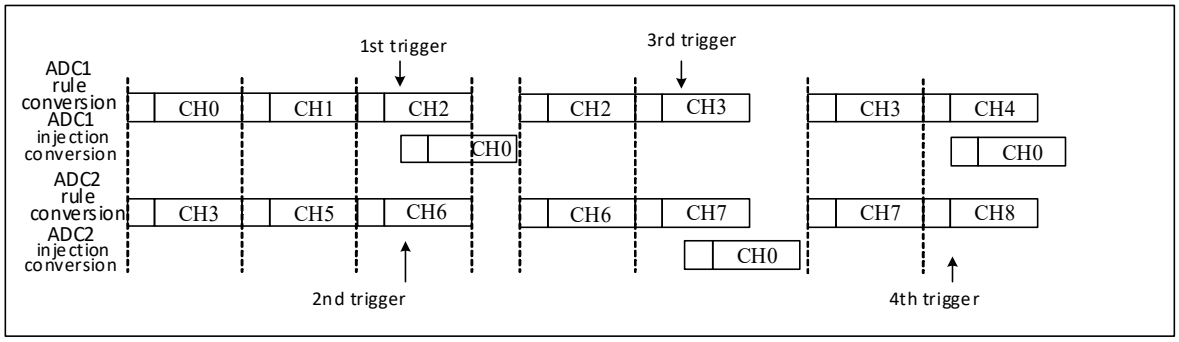
Figure 12-12 Alternate trigger injection channel conversion in synchronous rule mode



Note: In this mode, it is necessary to ensure that sequences with the same time length are converted accurately, or that the triggering time interval is longer than the longer of the two sequences.

If the injection trigger event occurs during the injection transition that interrupts the rule transition, the trigger event will be ignored, such as the second trigger described in the figure below.

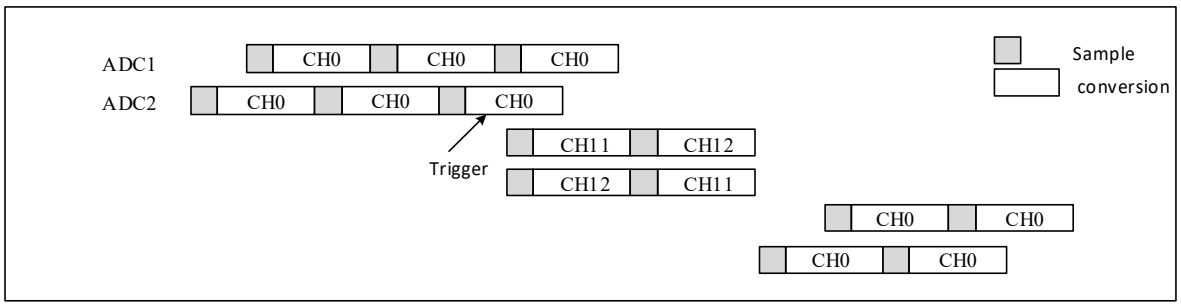
Figure 12-13 Trigger event occurs during the injection conversion



9) Synchronous injection mode + alternating mode

In this mode, injecting transitions can interrupt alternating transitions. When an injection event occurs, the alternation transition is interrupted, the injection transition is started, and after the injection transition ends, the alternation transition is resumed.

Figure 12-14 Trigger injection group conversion under alternating conversion



Note: When the ADC prescaler coefficient is 4, the sampling interval after alternate conversion recovery is no longer a uniform 7 ADC clocks, but instead alternates between 8 and 6 clock cycles.

12.2.7 Low-speed High-performance Mode

ADC defaults to 12-bit mode. After turning on low-speed and high-performance mode, 13/14/15-bit mode can be

selected, that is, 13-bit/14-bit/15-bit conversion results can be obtained in one conversion.

Notes on low speed and high-performance mode:

1) After turning on the low-speed high-performance mode, the single ADC conversion cycle (sampling time + conversion time) will increase, that is:
   13-bit mode conversion cycle = 12-bit mode conversion cycle * 2;
   14-bit mode conversion cycle = 12-bit mode conversion cycle * 4;
   15-bit mode conversion cycle = 12-bit mode conversion cycle * 8.
2) The use of low-speed high-performance mode is not restricted. No matter it is single mode, continuous mode, scan mode, intermittent mode and other working modes, low-speed high-performance mode is supported.

12.2.8 ADC Cascade

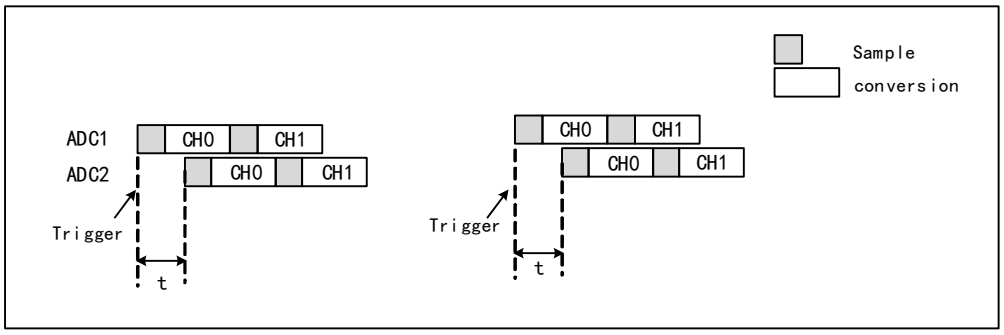
All ADCs can be cascaded, and the cascading order is configurable. Between the cascaded ADCs, the conversion delay of each level of ADC can be configured. It can convert simultaneously or delay the conversion in sequence; when used in groups, dual ADC mode is supported.

Select the previous-stage ADC by configuring the CONTS[1:0] bits of the ADCx_CTLR2 register.

ADC rule channel cascade configuration

If the ADC uses rule channel cascading, you can enable regular channel cascading through the REGUCON bit in the ADCx_CTLR2 register, and set parameters for trigger delay through the REGUDLY[5:0] field in the ADCx_CONCFG register. If the ADC2 cascade front stage is set to ADC1, ADC1 needs to provide a trigger source. If the delay time is set to t, the corresponding conversion relationship is as shown in Figure 12-15:

Figure 12-15 Rule channel cascade trigger delay

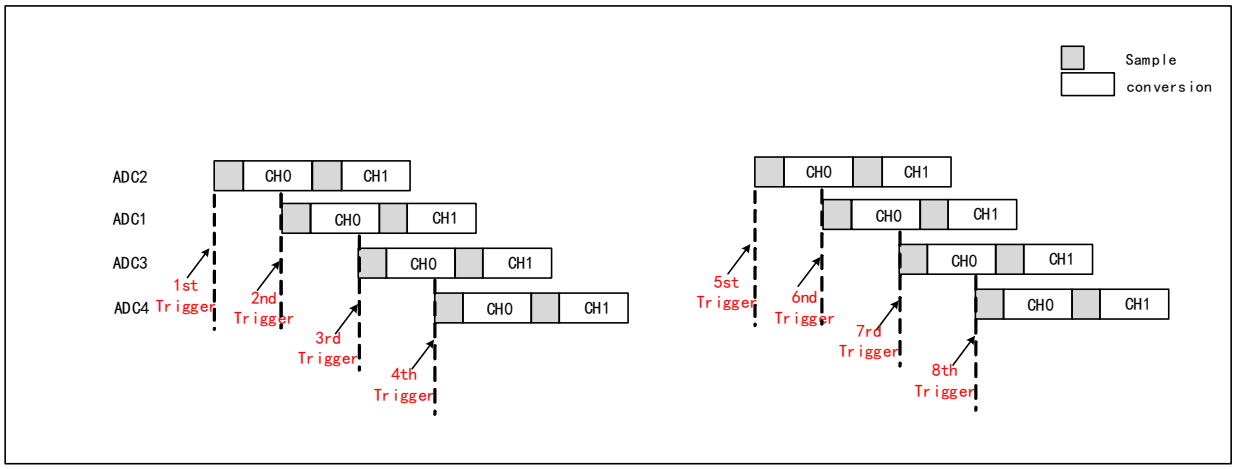


ADC injection channel cascade configuration

If the ADC uses injection channel cascade, it can be used with the injection trigger alternating cycle. Enable the injection channel cascade through the INJECON bit of the ADCx_CTLR2 register, set the parameters for trigger delay through the REGUDLY[5:0] field of the ADCx_CONCFG register, set the injection trigger alternating cycle period through the INJENUM[1:0] bits of the ADCx_CONCFG register, and configure the INJENUM[1:0] bits of the ADCx_CONCFG register to set the effective injection trigger sequence number.

If the ADC channels are cascaded and used in an alternating cycle of injection triggering, a multi-ADC alternating triggering mode can be achieved. If INJENUM[1:0]=11b and INJESEQ[1:0] are cascaded sequences in sequence, multi-ADC alternating triggering mode can be realized. The corresponding relationship is shown in Figure 12-16:

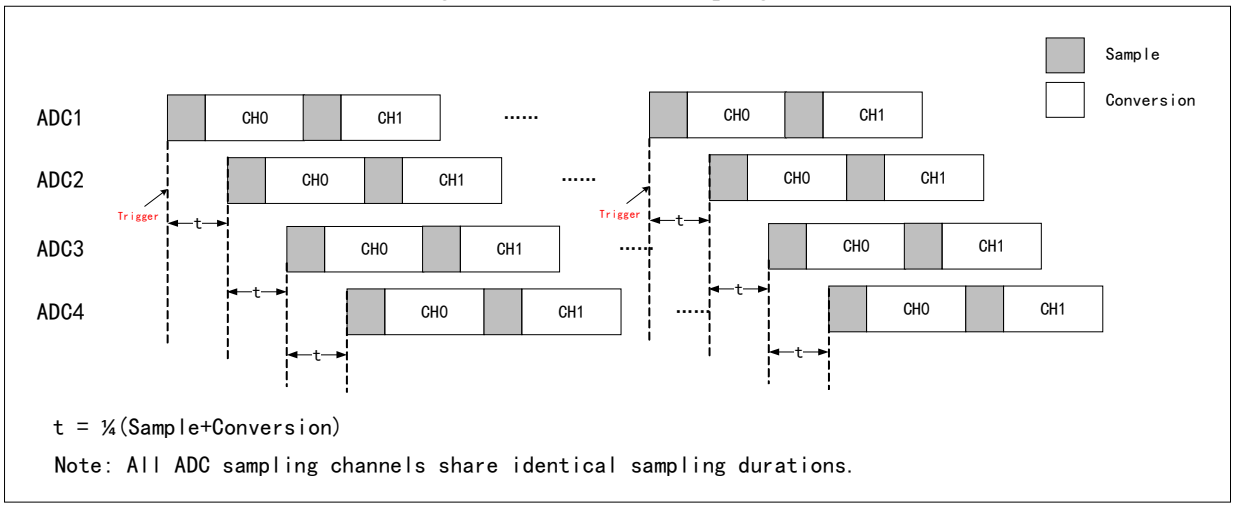
Figure 12-16 Multiple ADC alternate triggering



Application examples

In cascade mode, when using 4 groups of ADCs, the trigger delay can be configured to perform alternate sampling. If the trigger delay is 25% of the ADC conversion cycle time, the sampling rate can be increased by 4 times. That is, when 5Msps is used for sampling, the sampling rate can be increased to an equivalent 20Msps through cascade mode.

Figure 12-17 Increase sampling rate



12.2.9 ADC Scan Output

ADC scan output needs to configure the SCSCFG bit of the ADCx_SCANCNT register to select the rule channel or the injection channel for scan channel source selection. After the ADC scan is completed, the round count can be selected by setting the ADC_SCAN_SEL[1:0] bits of the R32_EXTEN_CTR register to select the ADCx scan cycle count output to GPIO. If SCANOE0, SCANOE1, and SCANOE2 are enabled, the relationship between the number of scans and the level of the GPIO port is as shown in Table 12-5:

Table 12-5 Relationship between ADC scan times and output level

Sequence	SCANOE0	SCANOE1	SCANOE2
0	Low level	Low level	Low level
1	High level	Low level	Low level

2	Low level	High level	Low level
3	High level	High level	Low level
4	Low level	Low level	High level
5	High level	Low level	High level
6	Low level	High level	High level
7	High level	High level	High level
8	Low level	Low level	Low level

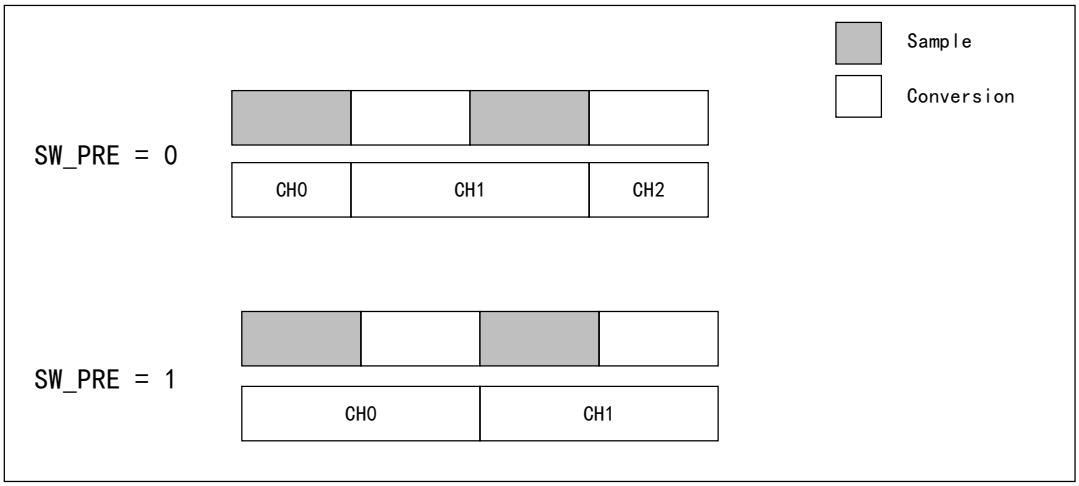
The signal generated in this mode can be added with one of the eight analog switch chips CH448 to achieve 8 times the number of ADC channels, supporting automatic switching of up to 8*48=384 ADC channels.

12.2.10 Channel Pre-switching

ADC contains 2 main steps: signal sampling and A/D analog-to-digital conversion. After the sampling step of the previous channel is completed, the ADC unit of CH32X315/X305 can be set to switch to the next channel by hardware in advance. During the A/D conversion process of the previous channel, the next channel completes the process of "switching-switching instant ringing-signal stabilization". This alternating parallel processing method of the 2-stage pipeline makes the signal voltage switched in advance stable enough when the next channel is sampled, thereby shortening the sampling time.

The channel pre-switching function is controlled by the SW_PRE bit of the ADCx_CTLR1 register, as shown in Figure 12-18.

Table 12-18 Channel pre-switching function



12.3 Register Description

Table 12-6 ADC1-related registers

Name	Access address	Description	Reset value
R32_ADC1_STATR	0x40012400	ADC1 status register	0x00000000
R32_ADC1_CTLR1	0x40012404	ADC1 control register 1	0x00000000
R32_ADC1_CTLR2	0x40012408	ADC1 control register 2	0x00000020
R32_ADC1_SAMPTR1	0x4001240C	ADC1 sample time configuration register 1	0x00000000

R32_ADC1_SAMPTR2	0x40012410	ADC1 sample time configuration register 2	0x00000000
R32_ADC1_IOFR1	0x40012414	ADC1 injected channel data offset register 1	0x00000000
R32_ADC1_IOFR2	0x40012418	ADC1 injected channel data offset register 2	0x00000000
R32_ADC1_IOFR3	0x4001241C	ADC1 injected channel data offset register 3	0x00000000
R32_ADC1_IOFR4	0x40012420	ADC1 injected channel data offset register 4	0x00000000
R32_ADC1_WDHTR	0x40012424	ADC1 watchdog high threshold register	0x00000FFF
R32_ADC1_WDLTR	0x40012428	ADC1 watchdog low threshold register	0x00000000
R32_ADC1_RSQR1	0x4001242C	ADC1 rule channel sequence register 1	0x00000000
R32_ADC1_RSQR2	0x40012430	ADC1 rule channel sequence register 2	0x00000000
R32_ADC1_RSQR3	0x40012434	ADC1 rule channel sequence register 3	0x00000000
R32_ADC1_ISQR	0x40012438	ADC1 injection channel sequence register	0x00000000
R32_ADC1_IDATAR1	0x4001243C	ADC1 injection data register 1	0x00000000
R32_ADC1_IDATAR2	0x40012440	ADC1 injection data register 2	0x00000000
R32_ADC1_IDATAR3	0x40012444	ADC1 injection data register 3	0x00000000
R32_ADC1_IDATAR4	0x40012448	ADC1 injection data register 4	0x00000000
R32_ADC1_RDATAR	0x4001244C	ADC1 rule data register	0x00000000
R32_ADC1_SMPMD	0x40012450	ADC1 sampling mode control register	0x00000000
R32_ADC1_UDP	0x40012454	ADC1 programmable sample time register	0x0000FF10
R32_ADC1_CONCFG	0x40012458	ADC1 cascade configuration register	0x00000000
R32_ADC1_SCANCNT	0x4001245C	ADC1 scan count configuration register	0x00000000

Table 12-7 ADC2-related registers

Name	Access address	Description	Reset value
R32_ADC2_STATR	0x40012800	ADC2 status register	0x00000000
R32_ADC2_CTLR1	0x40012804	ADC2 control register 1	0x00000000
R32_ADC2_CTLR2	0x40012808	ADC2 control register 2	0x000000D0
R32_ADC2_SAMPTR1	0x4001280C	ADC2 sample time configuration register 1	0x00000000
R32_ADC2_SAMPTR2	0x40012810	ADC2 sample time configuration register 2	0x00000000
R32_ADC2_IOFR1	0x40012814	ADC2 injected channel data offset register 1	0x00000000
R32_ADC2_IOFR2	0x40012818	ADC2 injected channel data offset register 2	0x00000000
R32_ADC2_IOFR3	0x4001281C	ADC2 injected channel data offset register 3	0x00000000
R32_ADC2_IOFR4	0x40012820	ADC2 injected channel data offset register 4	0x00000000
R32_ADC2_WDHTR	0x40012824	ADC2 watchdog high threshold register	0x00000FFF
R32_ADC2_WDLTR	0x40012828	ADC2 watchdog low threshold register	0x00000000
R32_ADC2_RSQR1	0x4001282C	ADC2 rule channel sequence register 1	0x00000000
R32_ADC2_RSQR2	0x40012830	ADC2 rule channel sequence register 2	0x00000000
R32_ADC2_RSQR3	0x40012834	ADC2 rule channel sequence register 3	0x00000000
R32_ADC2_ISQR	0x40012838	ADC2 injection channel sequence register	0x00000000
R32_ADC2_IDATAR1	0x4001283C	ADC2 injection data register 1	0x00000000
R32_ADC2_IDATAR2	0x40012840	ADC2 injection data register 2	0x00000000
R32_ADC2_IDATAR3	0x40012844	ADC2 injection data register 3	0x00000000
R32_ADC2_IDATAR4	0x40012848	ADC2 injection data register 4	0x00000000
R32_ADC2_RDATAR	0x4001284C	ADC2 rule data register	0x00000000

R32_ADC2_SMPMD	0x40012850	ADC2 sampling mode control register	0x00000000
R32_ADC2_UDP	0x40012854	ADC2 programmable sample time register	0x0000FF10
R32_ADC2_CONCFG	0x40012858	ADC2 cascade configuration register	0x00000000
R32_ADC2_SCANCNT	0x4001285C	ADC2 scan count configuration register	0x00000000

Table 12-8 ADC3-related registers

Name	Access address	Description	Reset value
R32_ADC3_STATR	0x40013C00	ADC3 status register	0x00000000
R32_ADC3_CTLR1	0x40013C04	ADC3 control register 1	0x00000000
R32_ADC3_CTLR2	0x40013C08	ADC3 control register 2	0x000000D0
R32_ADC3_SAMPTR1	0x40013C0C	ADC3 sample time configuration register 1	0x00000000
R32_ADC3_SAMPTR2	0x40013C10	ADC3 sample time configuration register 2	0x00000000
R32_ADC3_IOFR1	0x40013C14	ADC3 injected channel data offset register 1	0x00000000
R32_ADC3_IOFR2	0x40013C18	ADC3 injected channel data offset register 2	0x00000000
R32_ADC3_IOFR3	0x40013C1C	ADC3 injected channel data offset register 3	0x00000000
R32_ADC3_IOFR4	0x40013C20	ADC3 injected channel data offset register 4	0x00000000
R32_ADC3_WDHTR	0x40013C24	ADC3 watchdog high threshold register	0x00000FFF
R32_ADC3_WDLTR	0x40013C28	ADC3 watchdog low threshold register	0x00000000
R32_ADC3_RSQR1	0x40013C2C	ADC3 rule channel sequence register 1	0x00000000
R32_ADC3_RSQR2	0x40013C30	ADC3 rule channel sequence register 2	0x00000000
R32_ADC3_RSQR3	0x40013C34	ADC3 rule channel sequence register 3	0x00000000
R32_ADC3_ISQR	0x40013C38	ADC3 injection channel sequence register	0x00000000
R32_ADC3_IDATAR1	0x40013C3C	ADC3 injection data register 1	0x00000000
R32_ADC3_IDATAR2	0x40013C40	ADC3 injection data register 2	0x00000000
R32_ADC3_IDATAR3	0x40013C44	ADC3 injection data register 3	0x00000000
R32_ADC3_IDATAR4	0x40013C48	ADC3 injection data register 4	0x00000000
R32_ADC3_RDATAR	0x40013C4C	ADC3 rule data register	0x00000000
R32_ADC3_SMPMD	0x40013C50	ADC3 sampling mode control register	0x00000000
R32_ADC3_UDP	0x40013C54	ADC3 programmable sample time register	0x0000FF10
R32_ADC3_CONCFG	0x40013C58	ADC3 cascade configuration register	0x00000000
R32_ADC3_SCANCNT	0x40013C5C	ADC3 scan count configuration register	0x00000000

Table 12-9 ADC4-related registers

Name	Access address	Description	Reset value
R32_ADC4_STATR	0x40014000	ADC4 status register	0x00000000
R32_ADC4_CTLR1	0x40014004	ADC4 control register 1	0x00000000
R32_ADC4_CTLR2	0x40014008	ADC4 control register 2	0x000000D0
R32_ADC4_SAMPTR1	0x4001400C	ADC4 sample time configuration register 1	0x00000000
R32_ADC4_SAMPTR2	0x40014010	ADC4 sample time configuration register 2	0x00000000
R32_ADC4_IOFR1	0x40014014	ADC4 injected channel data offset register 1	0x00000000
R32_ADC4_IOFR2	0x40014018	ADC4 injected channel data offset register 2	0x00000000
R32_ADC4_IOFR3	0x4001401C	ADC4 injected channel data offset register 3	0x00000000

R32_ADC4_IOFR4	0x40014020	ADC4 injected channel data offset register 4	0x00000000
R32_ADC4_WDHTR	0x40014024	ADC4 watchdog high threshold register	0x00000FFF
R32_ADC4_WDLTR	0x40014028	ADC4 watchdog low threshold register	0x00000000
R32_ADC4_RSQR1	0x4001402C	ADC4 rule channel sequence register 1	0x00000000
R32_ADC4_RSQR2	0x40014030	ADC4 rule channel sequence register 2	0x00000000
R32_ADC4_RSQR3	0x40014034	ADC4 rule channel sequence register 3	0x00000000
R32_ADC4_ISQR	0x40014038	ADC4 injection channel sequence register	0x00000000
R32_ADC4_IDATAR1	0x4001403C	ADC4 injection data register 1	0x00000000
R32_ADC4_IDATAR2	0x40014040	ADC4 injection data register 2	0x00000000
R32_ADC4_IDATAR3	0x40014044	ADC4 injection data register 3	0x00000000
R32_ADC4_IDATAR4	0x40014048	ADC4 injection data register 4	0x00000000
R32_ADC4_RDATAR	0x4001404C	ADC4 rule data register	0x00000000
R32_ADC4_SMPMD	0x40014050	ADC4 sampling mode control register	0x00000000
R32_ADC4_UDP	0x40014054	ADC4 programmable sample time register	0x0000FF10
R32_ADC4_CONCFG	0x40014058	ADC4 cascade configuration register	0x00000000
R32_ADC4_SCANCNT	0x4001405C	ADC4 scan count configuration register	0x00000000

12.3.1 ADCx Status Register (ADC_STATR) (x=1/2/3/4)

Offset address: 0x00

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved					DELTA[10:0]										

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADC RSTF	Reserved										STRT	JSTRT	JEOC	EOC	AWD

Bit	Name	Access	Description	Reset value
[31:27]	Reserved	RO	Reserved	0
[26:16]	DELTA[10:0]	RO	Calibration deviation value.	0
15	ADCRSTF	RO	ADC reset status indication: 1: Resetting; 0: Reset completed.	0
[14:5]	Reserved	RO	Reserved	0
4	STRT	RW0	Regular channel conversion start status: 1: The regular channel conversion has started; 0: The regular channel conversion has not started. This bit is set by hardware and cleared by software (invalid if writing 1).	0
3	JSTRT	RW0	Injected channel conversion start status: 1: The injected channel conversion has started; 0: The injected channel conversion has not started; This bit is set by hardware and cleared by software (invalid if writing 1).	0
2	JEOC	RW0	Injected channel group conversion completion status:	0

			1: The conversion has completed; 0: The conversion has not completed. This bit is set to 1 by hardware (the conversion of all injected channels is completed), and cleared by software (invalid if writing 1).	
1	EOC	RW0	Conversion completion status: 1: The conversion has completed; 0: The conversion has not completed. This bit is set to 1 by hardware (the regular or injected channel group conversion ends), and is cleared by software (invalid if writing 1) or clearing when ADCx_RDATAR is read.	0
0	AWD	RW0	Analog watchdog flag bit: 1: The analog watchdog event occurs; 0: No analog watchdog event occurs. This bit is set to 1 by hardware (the conversion value is out of the ADCx_WDHTR and ADCx_WDLTR register range), and is cleared by software (invalid if writing 1).	0

12.3.2 ADCx Control Register 1 (ADCx_CTLR1) (x=1/2/3/4)

Offset address: 0x04

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SW_PRE	ANA_RST	AWD_RST	Reserved		BUFEN	Reserved		AWDEN	JAWDEN	Reserved		DUALMOD[3:0]			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DISCNUM[2:0]			JDISCEN	DISCEN	JAUTO	AWDSGL	SCAN	JEOCIE	AWDIE	EOCIE	AWDCH[4:0]				

Bit	Name	Access	Description	Reset value
31	SW_PRE	RW	Channel pre-switching off: 1: Channel pre-switching off; 0: Channel pre-switching on.	0
30	ANA_RST	RW	Analog module reset enable: 1: Enable; 0: Disable. *Note: Resetting the ADC analog circuit during ADC conversion will destroy the conversion result. This bit can be used with ADCxRST of RCC.*	0
29	AWD_RST	RW	ADC watchdog reset enable: 1: Enabled; 0: Disabled.	0
[28:27]	Reserved	RO	Reserved	0
26	BUFEN	RW	ADC BUFFER enable. 0: Disable input Buffer.	0

			1: Enable input Buffer.	
25	Reserved	RO	Reserved, the default value of this bit must remain unchanged.	0
24	Reserved	RO	Reserved	0
23	AWDEN	RW	Analog watchdog enable bit on regular channels: 1: Enable analog watchdog on regular channels; 0: Disable analog watchdog on regular channels.	0
22	JAWDEN	RW	Analog watchdog enable bit on injected channels: 1: Enable analog watchdog on injected channels; 0: Disable analog watchdog on injected channels.	0
[21:20]	Reserved	RO	Reserved	0
[19:16]	DUALMOD[3:0]	RW	Dual mode selection: 0000: Independent mode; 0001: Synchronization rules + synchronization injection mode; 0010: Synchronization rules + alternate trigger mode; 0011: Synchronous injection + fast alternating mode; 0100: Synchronous injection + slow alternating mode; 0101: Synchronous injection mode; 0110: Synchronization rule mode; 0111: Fast alternating mode; 1000: Slow alternating mode; 1001: Alternate trigger mode. *Note: These bits in ADC2/ADC4 are reserved bits, and any modification of the configuration bits should be performed with the dual mode turned off.*	0000b
[15:13]	DISCNUM[2:0]	RW	In discontinuous mode, the number of regular channels to be converted after external trigger: 000: 1 channel; … 111: 8 channels.	000b
12	JDISCEN	RW	Discontinuous mode enable bit on injected channel: 1: Enable discontinuous mode on the injected channel; 0: Disable discontinuous mode on the injected channel.	0
11	DISCEN	RW	Discontinuous mode enable level on the regular channel: 1: Enable discontinuous mode on regular channel. 0: Disable discontinuous mode on regular channel.	0
10	JAUTO	RW	After opening the regular channel, automatically convert the enable bit of the injected channel group: 1: Enable automatic injection channel group conversion. 0: Disable automatic injection channel group conversion. *Note: This mode needs to disable the external trigger function of the injection channel.*	0
9	AWDSGL	RW	In scan mode, analog watchdog enable bit on a single	0

			channel: 1: Enable analog watchdog on single channel (AWDCH[4:0] selection); 0: Disable analog watchdog on all channels.	
8	SCAN	RW	Scan mode enable bit: 1: Enable scan mode (continuous conversion of all channels selected by ADCx_IOFRx and ADCx_RSQRx); 0: Disable scan mode.	0
7	JEOCIE	RW	Injected channel group conversion completion interrupt enable bit: 1: Enable injected channel group transfer completion interrupt (JEOC flag); 0: Disable injected channel group transfer completion interrupt.	0
6	AWDIE	RW	Analog watchdog interrupt enable bit: 1: Enable analog watchdog interrupt; 0: Disable analog watchdog interrupt.	0
5	EOCIE	RW	Conversion completion (rule or injected channel group) interrupt enable bit; 1: Enable the transfer completion bit (EOC flag): 0: Disable the transfer completion interrupt.	0
[4:0]	AWDCH[4:0]	RW	Analog watchdog channel selection bit: 00000: Analog input channel 0; 00001: Analog input channel 1; … 01100: Analog input channel 12; 01101: Analog input channel 13.	00000b

12.3.3 ADCx Control Register 2 (ADCx_CTLR2) (x=1/2/3/4)

Offset address: 0x08

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved		CONTS[1:0]		INJECON	REGUCON	ENHANCE[1:0]		VREFE	SWSTART	JSWSTART	EXTTRIG	EXTSEL[2:0]			Reserved

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
JEXTTRIG	JEXTSEL[2:0]			ALIGN	Reserved		DMA	Reserved	CAL_AUTO	CAL_VOL[1:0]		RSTCAL	CAL	CONT	ADON

Bit	Name	Access	Description	Reset value
[31:30]	Reserved	RO	Reserved	0
[29:28]	CONTS[1:0]	RW	Pre-stage ADC selection bit: 00: Pre-stage ADC is ADC1;	0

			01: Pre-stage ADC is ADC2; 10: Pre-stage ADC is ADC3; 11: Pre-stage ADC is ADC4.	
27	INJECON	RW	Injection channel cascade enable bit: 1: Enable the injection channel cascade function, and the injection channel can be converted by the pre-stage ADC trigger; 0: Disable the injection channel cascade function.	0
26	REGUCON	RW	Rule channel cascade enable bit: 1: Enable the rule channel cascade function, the rule channel can be converted by the pre-stage ADC trigger; 0: Disable the rule channel cascade function.	0
[25:24]	ENHANCE[1:0]	RW	Enhanced mode configuration bits: 00: 12-bit mode; 01: 13-bit enhanced mode; 10: 14-bit enhanced mode; 11: 15-bit enhanced mode.	0
23	VREFE	RW	Internal voltage (V_{REFINT}) channel enable: 1: Enable V_{REFINT} channel; 0: Disable V_{REFINT} channel. *Note: This bit is only available for ADC1.*	0
22	SWSTART	RW	To start a rule channel conversion, you need to set the software trigger: 1: Start rule channel conversion. 0: Reset state. This bit is set by the software, and the hardware is cleared 0 after the conversion starts.	0
21	JSWSTART	RW	To start an injection channel conversion, you need to set the software trigger: 1: Start the injection channel conversion. 0: Reset state. This bit is set by the software, and the hardware or software clears 0 after the conversion starts.	0
20	EXTTRIG	RW	External trigger conversion mode enable for regular channels: 1: Enable conversion on external event; 0: Disable conversion on external event.	0
[19:17]	EXTSEL[2:0]	RW	External trigger event selection for regular channel: 000: CC1 event of timer 1; 001: CC2 event of timer 1; 010: CC3 event of timer 1; 011: CC2 event of timer 2; 100: TRGO event of timer 3; 101: CC4 event of timer 4;	000b

			110: EXTI line12; 111: SWSTART software trigger.	
16	Reserved	RO	Reserved	0
15	JEXTTRIG	RW	External trigger conversion mode enable for injected channels: 1: Enable conversion on external event; 0: Disable conversion on external event.	0
[14:12]	JEXTSEL[2:0]	RW	External trigger event select for injected channels: 000: TRGO event of timer 1; 001: CC4 event of timer 1; 010: TRGO event of timer 2; 011: CC1 event of timer 2; 100: CC4 event of timer 3; 101: TRGO event of timer 4; 110: EXTI line 15; 111: JSWSTART software trigger.	000b
11	ALIGN	RW	Data alignment: 1: Left alignment; 0: Right alignment.	0
[10:9]	Reserved	RO	Reserved	0
8	DMA	RW	Direct memory access (DMA) mode enable: 1: Enable DMA mode; 0: Disable DMA mode.	0
7	Reserved	RO	Reserved	0
6	CAL_AUTO	RW	Hardware automatic calibration function enable: 1: Enable hardware automatic calibration; 0: Disable hardware automatic calibration.	1
[5:4]	CAL_VOL[1:0]	RW	Calibration voltage configuration: 00: 0.2*Vref; 01: 0.4*Vref; 10: 0.6*Vref; 11: 0.8*Vref.	0
3	RSTCAL	RW	Reset calibration, this bit is set by software, and cleared by hardware after reset: 1: Initialize calibration register; 0: The calibration register initialized. *Note: If RSTCAL is set while the conversion is in progress, it takes extra cycles to clear the calibration register.*	0
2	CAL	RW	A/D calibration, set by software and cleared by hardware when the calibration is completed. 1: Enable the calibration: 0: Calibration completed.	0
1	CONT	RW	Continuous conversion enable: 1: Continuous conversion mode;	0

			0: Single conversion mode. If this bit is set, the conversion will continue until the bit is cleared.	
0	ADON	RW	A/D converter ON/OFF When this bit is 0, writing 1 will wake up the ADC from power-down mode; When this bit is set to 1, writing a 1 will trigger a conversion; if any other bits change, a new conversion will not be triggered. 1: Enable ADC and to start conversion; 0: Disable ADC conversion/calibration, and go to power down mode.	0

12.3.4 ADCx Sample Time Configuration Register 1 (ADCx_SAMPTR1) (x=1/2/3/4)

Offset address: 0x0C

31	30	29	28	27	26	25	24	23	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				SMP13[2:0]			SMP12[2:0]			SMP11[2:0]			SMP10[2:0]		

Bit	Name	Access	Description	Reset value
[31:12]	Reserved	RO	Reserved	0
[11:0]	SMPx[2:0]	RW	SMPx[2:0]: Sample time configuration of channel x: 000: 3.5 cycles; 001: 5.5 cycles; 010: 7.5 cycles; 011: 13.5 cycles; 100: 20.5 cycles; 101: 30.5 cycles; 110: 45.5 cycles; 111: 68.5 cycles; SMPx[2:0]: When the SMPMDx bit is set, the channel x sampling time is configured by the register SMPUDP: These bits are used to independently select the sample time of each channel, and the channel configuration value must remain unchanged during the sampling period.	000b

12.3.5 ADCx Sample Time Configuration Register 2 (ADCx_SAMPTR2) (x=1/2/3/4)

Offset address: 0x10

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved		SMP9[2:0]			SMP8[2:0]			SMP7[2:0]			SMP6[2:0]			SMP5[2:1]	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMP5[0]		SMP4[2:0]			SMP3[2:0]			SMP2[2:0]			SMP1[2:0]			SMP0[2:0]	

Bit	Name	Access	Description	Reset value
[31:30]	Reserved	RO	Reserved	0
[29:0]	SMPx[2:0]	RW	SMPx[2:0]: Sample time configuration of channel x: 000: 3.5 cycles; 001: 5.5 cycles; 010: 7.5 cycles; 011: 13.5 cycles; 100: 20.5 cycles; 101: 30.5 cycles; 110: 45.5 cycles; 111: 68.5 cycles; SMPx[2:0]: When the SMPMDx bit is set, the channel x sampling time is configured by the register SMPUDP: These bits are used to independently select the sample time of each channel, and the channel configuration value must remain unchanged during the sampling period.	000b

12.3.6 ADCy Injected Channel Data Offset Register (ADCy_IOFRx) (y=1/2/3/4, x=1/2/3/4)

Offset address: 0x14 + (x-1)*4

31	30	29	28	27	26	25	24	23	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	JOFFSETx[14:0]														

Bit	Name	Access	Description	Reset value
[31:15]	Reserved	RO	Reserved	0
[14:0]	JOFFSETx[14:0]	RW	Data offset for injected channel x. When the injected channel is converted, these bits define the value to be subtracted from the original conversion data. The result of the conversion can be read in the ADCx_IDATARx register.	0

12.3.7 ADCx Watchdog High Threshold Register (ADCx_WDHTR) (x=1/2/3/4)

Offset address: 0x24

31	30	29	28	27	26	25	24	23	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	HT[14:0]														

Bit	Name	Access	Description	Reset value
[31:15]	Reserved	RO	Reserved	0
[14:0]	HT[14:0]	RW	Analog watchdog high threshold set bits.	0xFFF

Note: The values of WDHTR and WDLTR can be changed during the conversion, but they will take effect in the next conversion.

12.3.8 ADCx Watchdog Low Threshold Register (ADCx_WDLTR) (x=1/2/3/4)

Offset address: 0x28

31	30	29	28	27	26	25	24	23	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	LT[14:0]														

Bit	Name	Access	Description	Reset value
[31:15]	Reserved	RO	Reserved	0
[14:0]	LT[14:0]	RW	Analog watchdog low threshold set bits.	0

Note: The values of WDHTR and WDLTR can be changed during the conversion, but they will take effect in the next conversion.

12.3.9 ADCx Rule Channel Sequence Register 1 (ADCx_RSQR1) (x=1/2/3/4)

Offset address: 0x2C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved								L[3:0]				RSQ16[4:1]			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SQ16[0]	SQ15[4:0]					SQ14[4:0]					SQ13[4:0]				

Bit	Name	Access	Description	Reset value
[31:24]	Reserved	RO	Reserved	0
[23:20]	L[3:0]	RW	The number of channels to be converted in a rule channel conversion sequence: 0000-1111: 1-16 conversions.	0
[19:15]	SQ16[4:0]	RW	Number of the 16th conversion channel in the rule sequence (0-13).	0
[14:10]	SQ15[4:0]	RW	Number of the 15th conversion channel in the rule sequence (0-13).	0
[9:5]	SQ14[4:0]	RW	Numbering of the 14th conversion channel in the rule sequence (0-13).	0
[4:0]	SQ13[4:0]	RW	Numbering of the 13th conversion channel in the rule sequence (0-13).	0

12.3.10 ADCx Rule Channel Sequence Register 2 (ADCx_RSQR2) (x=1/2/3/4)

Offset address: 0x30

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved		SQ12[4:0]					SQ11[4:0]					SQ10[4:1]			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

SQ10[0]	SQ9[4:0]	SQ8[4:0]	SQ7[4:0]

Bit	Name	Access	Description	Reset value
[31:30]	Reserved	RO	Reserved	0
[29:25]	SQ12[4:0]	RW	Number of the 12th conversion channel in the rule sequence (0-13).	0
[24:20]	SQ11[4:0]	RW	Number of the 11th conversion channel in the rule sequence (0-13).	0
[19:15]	SQ10[4:0]	RW	Number of the 10th conversion channel in the rule sequence (0-13).	0
[14:10]	SQ9[4:0]	RW	Number of the 9th conversion channel in the rule sequence (0-13).	0
[9:5]	SQ8[4:0]	RW	Number of the 8th conversion channel in the rule sequence (0-13).	0
[4:0]	SQ7[4:0]	RW	Number of the 7th conversion channel in the rule sequence (0-13).	0

12.3.11 ADCx Rule Channel Sequence Register 3 (ADCx_RSQR3) (x=1/2/3/4)

Offset address: 0x34

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved		SQ6[4:0]					SQ5[4:0]					SQ4[4:1]			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SQ4[0]		SQ3[4:0]					SQ2[4:0]					SQ1[4:0]			

Bit	Name	Access	Description	Reset value
[31:30]	Reserved	RO	Reserved	0
[29:25]	SQ6[4:0]	RW	Number of the 6th conversion channel in the rule sequence (0-13).	0
[24:20]	SQ5[4:0]	RW	Number of the 5th conversion channel in the rule sequence (0-13).	0
[19:15]	SQ4[4:0]	RW	Number of the 4th conversion channel in the rule sequence (0-13).	0
[14:10]	SQ3[4:0]	RW	Number of the 3rd conversion channel in the rule sequence (0-13).	0
[9:5]	SQ2[4:0]	RW	Number of the 2nd conversion channel in the rule sequence (0-13).	0
[4:0]	SQ1[4:0]	RW	Number of the 1st conversion channel in the rule sequence (0-13).	0

12.3.12 ADCx Injected Channel Sequence Register (ADCx_ISQR) (x=1/2/3/4)

Offset address: 0x38

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

Reserved										JL[1:0]		JSQ4[4:1]			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
JSQ4[0]	JSQ3[4:0]					JSQ2[4:0]				JSQ1[4:0]					

Bit	Name	Access	Description	Reset value
[31:22]	Reserved	RO	Reserved	0
[21:20]	JL[1:0]	RW	Injects the number of channels to be converted in the channel conversion sequence: 00-11: 1-4 conversions.	0
[19:15]	JSQ4[4:0]	RW	Inject the number of the 4th conversion channel in the sequence (0-13). *Note: The software writes and assigns the channel number (0-13) as the 4th in the sequence to be converted.*	0
[14:10]	JSQ3[4:0]	RW	Inject the number of the 3rd conversion channel in the sequence (0-13).	0
[9:5]	JSQ2[4:0]	RW	Inject the number of the 2nd conversion channel in the sequence (0-13).	0
[4:0]	JSQ1[4:0]	RW	Inject the number of the 1st conversion channel in the sequence (0-13).	0

Note: Unlike regular conversion sequences, if the length of JL[1:0] is less than 4, the sequence order of conversions starts with (4-JL).

For example, when JL[1:0]=3 (4 injected transitions in the sequencer), the ADC will convert channels in the following order: JSQ1[4:0], JSQ2[4:0], JSQ3[4:0], and JSQ4[4:0];

When JL[1:0]=2 (3 injected transitions in the sequencer), the ADC will convert the channels in the following order: JSQ2[4:0], JSQ3[4:0] and JSQ4[4:0];

When JL[1:0]=1 (2 injected conversions in the sequencer), the ADC converts the channels in the following order: first JSQ3[4:0], then JSQ4[4:0];

When JL[1:0] = 0 (1 injection conversion in the sequencer), the ADC will convert only the JSQ4[4:0] channels.

If ADCx_ISQR[21:0]=10 00111 00011 00111 00010, the ADC will convert channels in the following order: JSQ2[4:0], JSQ3[4:0], and JSQ4[4:0], indicating that the scan conversions are performed in the following channel order: 7, 3, 7.

12.3.13 ADCy Injected Data Register (ADCy_IDATARx) (y=1/2/3/4, x=1/2/3/4)

Offset address: 0x3C + (x-1)*4

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

Bit	Name	Access	Description	Reset value
[31:16]	Reserved	RO	Reserved.	0
[15:0]	JDATA[15:0]	RO	Injected channel converted data (data left alignment or	0

			right alignment).	

12.3.14 ADCx Rule Data Register (ADCx_RDATAR) (x=1/2/3/4)

Offset address: 0x4C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ADC2DATA[15:0]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA[15:0]															

Bit	Name	Access	Description	Reset value
[31:16]	ADC2DATA[15:0]	RO	Data from the ADC2 conversion: In ADC1 (ADC3): In dual-mode, these bits contain the rule channel data from the ADC2 (ADC4) conversion. In ADC2 (ADC4): These bits are not used.	0
[15:0]	DATA[15:0]	RO	Rule channels transform data (data is left or right aligned).	0

12.3.15 ADCx Sampling Time Register (ADCx_SMPMD) (x=1/2/3/4)

Offset address: 0x50

31	30	29	28	27	26	25	24	23	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		SMPMD13	SMPMD12	SMPMD11	SMPMD10	SMPMD9	SMPMD8	SMPMD7	SMPMD6	SMPMD5	SMPMD4	SMPMD3	SMPMD2	SMPMD1	SMPMD0

Bit	Name	Access	Description	Reset value
[31:14]	Reserved	R0	Reserved	0
[13:0]	SMPMDx	RW	x=0-13, sampling mode configuration: 1: Channel x sampling time is configured by register SMPUDP; 0: Channel x sampling time is configured by register SMPx.	0

12.3.16 ADCx Programmable Sampling Time Register (ADCx_UDP) (x=1/2/3/4)

Offset address: 0x54

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

Bit	Name	Access	Description	Reset value
[31:17]	Reserved	R0	Reserved	0
[15:8]	SMPCAL[7:0]	RW	Calibrate the channel sampling time, the actual sampling time is SMPCAL-12.5 ADC clock cycles. When SMPCAL<16, the sampling time is fixed to 3.5 ADC clock cycles.	0xFF
[7:0]	SMPUDP[7:0]	RW	User programmable sampling time, the actual sampling time is SMPUDP-12.5 ADC clock cycles. When SMPUDP<16, the sampling time is fixed to 3.5 ADC clock cycles.	0x10

12.3.17 ADCx Cascade Configuration Register (ADCx_CONCFG) (x=1/2/3/4)

Offset address: 0x58

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved												INJENUM [1:0]		INJESEQ [1:0]	

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

Bit	Name	Access	Description	Reset value
[31:20]	Reserved	R0	Reserved	0
[19:18]	INJENUM[1:0]	RW	Injection trigger alternating cycle configuration bits: 00: 1 injection trigger cycle; 01: 2 injection trigger cycles; 10: 3 injection trigger cycles; 11: 4 injection trigger cycles. *Note: When INJENUM=0 and INJESEQ=0, each injection trigger will start the ADC injection group conversion.*	0
[17:16]	INJESEQ[1:0]	RW	Valid injection trigger sequence number configuration bits: 00: The first trigger is valid; 01: The second trigger is valid; 10: The third trigger is valid; 11: The fourth trigger is valid. *Note: Used in conjunction with INJENUM, and INJESEQ cannot be greater than INJENUM, otherwise the injection group conversion cannot be triggered.*	0
[15:14]	Reserved	RO	Reserved	0
[13:8]	INJEDLY[5:0]	RW	Injection channel cascade trigger delay, unit: ADC clock cycle.	0
[7:6]	Reserved	RO	Reserved	0

[5:0]	REGUDLY[5:0]	RW	Rule channel cascade trigger delay, unit: ADC clock cycle.	0

12.3.18 ADCx Scan Count Configuration Register (ADCx_SCANCNT) (x=1/2/3/4)

Offset address: 0x5C

31	30	29	28	27	26	25	24	23	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												SCANOE2	SCANOE1	SCANOE0	SCSCFG

Bit	Name	Access	Description	Reset value
[31:4]	Reserved	R0	Reserved	0
3	SCANOE2	RW	Scan cycle counter bit 2 output enable bit: 1: Enable; 0: Disable.	0
2	SCANOE1	RW	Scan cycle counter bit 1 output enable bit: 1: Enable; 0: Disable.	0
1	SCANOE0	RW	Scan cycle counter bit 0 output enable bit: 1: Enable; 0: Disable.	0
0	SCSCFG	RW	Scan channel source configuration bit: 1: Injection channel scan; 0: Rule channel scan.	0

Chapter 13 Advanced-control Timer (ADTM)

The advanced-control timer module contains a powerful 16-bit automatic reset timer (TIM1), which can be used to measure pulse width or generate pulses, PWM waves, etc. Used in motor control, power supply and other fields.

13.1 Main Features

Advanced-control timer (TIM1) features include:

- 16-bit auto-reload counter, supports up count, down count and up/down count
- 16-bit prescaler; the frequency division factor is dynamically adjustable from 1 to 65536
- 4 independent compare/capture channels
- Each compare/capture channel supports multiple working modes, such as: input capture, output compare, PWM generation and single pulse output
- Complementary outputs with programmable dead-time
- External signal to control timer
- Repetition counter to update the timer after the determination of the cycle
- Break signal input to put the timer's output signals in reset status or in a known status
- DMA generation in multiple modes
- Incremental encoder
- Cascade connection and synchronization between timers

13.2 Principle and Structure

This section describes the internal structure of the advanced-control timer.

13.2.1 Overview

As shown in Figure 13-1, the structure of the advanced-control timer can be roughly divided into 3 parts: Input clock part, core counterpart and compare/capture channel part.

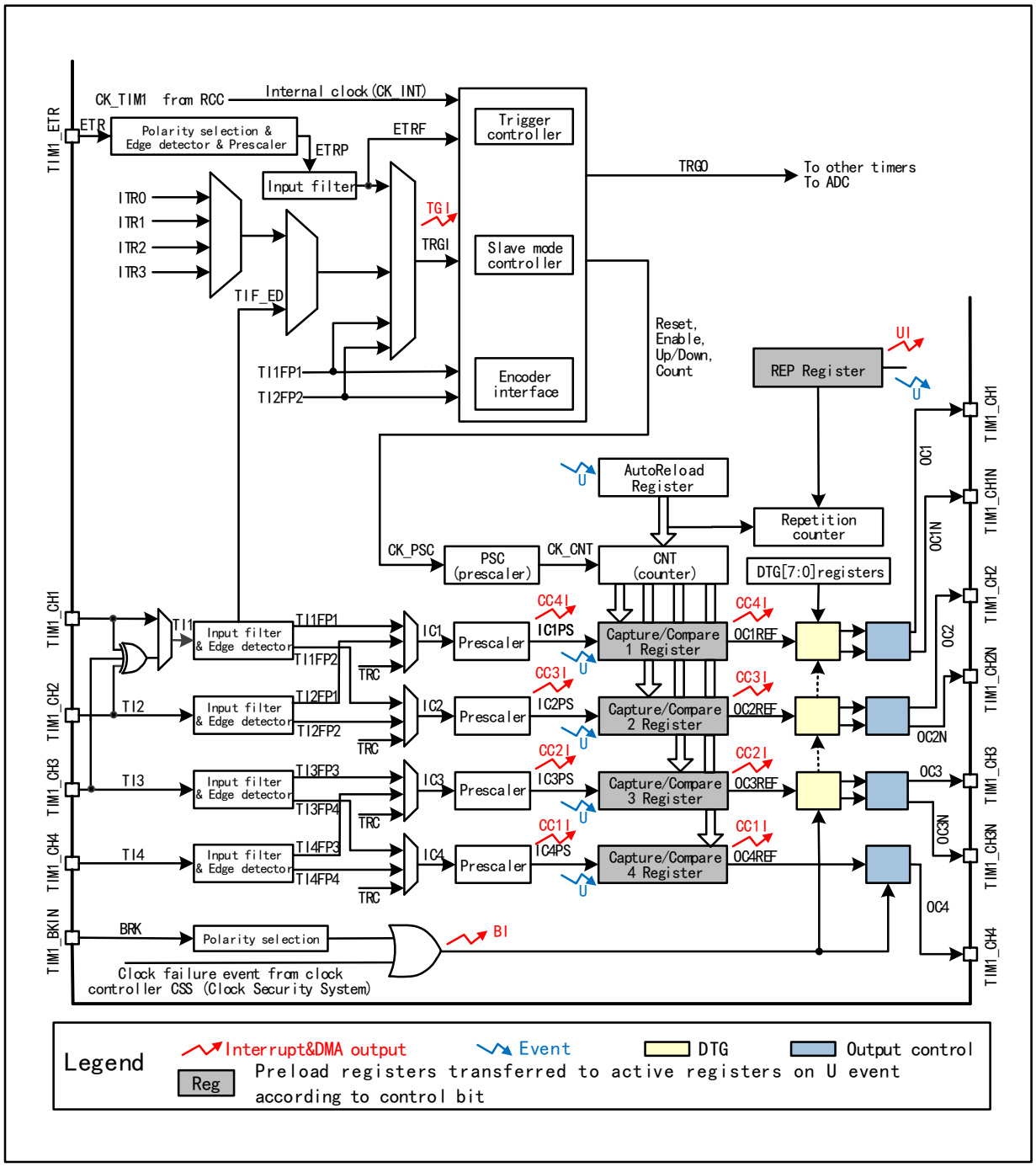
The advanced-control timer clock can come from PB bus clock (CK_INT), external clock input pin (TIM1_ETR), other timers with clock output function (ITRx), or the input end of compare capture channel (TIM1_CHx). These input clock signals will become CK_PSC clocks after various set filtering and frequency division operations, and will output to the core counterpart. In addition, these complex clock sources can also be output as TRGO to other timers, ADC and other peripherals.

The core of the advanced timer is a 16-bit counter (CNT). After CK_PSC is divided by the prescaler (PSC), it becomes CK_CNT and is output to CNT. CNT supports up counting mode, down counting mode and up and down counting mode. In down counting mode, the CNT is reloaded with the initial value after each counting cycle. At the end of each counting period under up-counting, CNT starts counting from 0. There is also an auxiliary counter that counts the number of times ATRLR reloads the initial value for CNT. When the number reaches the number set in the repetition count value register (RPTCR), a specific event can be generated.

The advanced-control timer has 4 groups of compare/capture channels. On each group of compare/capture channel, pulses can be inputted from its dedicated pins or output waveforms to the pins, i.e., the compare/capture channels support input and output modes. The input of each channel of the compare/capture register supports operations such as filtering, frequency division and edge detection, and supports mutual triggering between channels, and can also

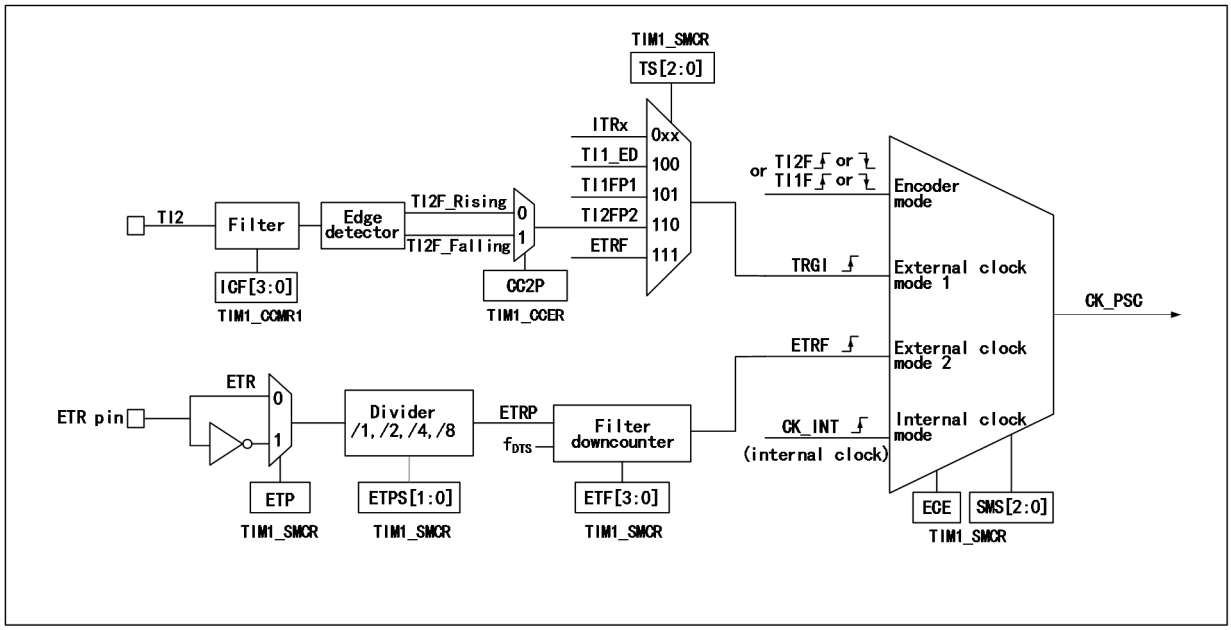
provide a clock for the core counter CNT. Each compare/capture channel has a set of compare/capture register (CHxCVR), which supports comparison with the main counter (CNT) so as to output pulse.

Figure 13-1 Structure block diagram of advanced-control timer



13.2.2 Clock Input

Figure 13-2 CK_PSC source block diagram of advanced-control timer



There are many clock sources for advanced-control timer CK_PSC, which can be divided into 4 categories:

1) The route of external clock pin (ETR) input clock: ETR→ETRP→ETRF;
2) Internal PB clock input route: CK_INT;
3) The route from the compare/capture channel pin (TIM1_CHx): TIM1_CHx→TIx→TIxFPx; this route is also used in encoder mode;
4) Input from other internal timers: ITRx;

The actual operation can be divided into 4 categories by determining the input pulse selection of the SMS from the CK_PSC source:

1) Select the internal clock source (CK_INT);
2) External clock source mode 1;
3) External clock source mode 2;
4) Encoder mode;

The 4 clock sources mentioned above can be selected by these 4 operations.

13.2.2.1 Internal Clock Source (CK_INT)

If the advanced-control timer is started when the SMS field is kept at 000b, then the internal clock source (CK_INT) is selected as the clock. At this moment, CK_INT is CK_PSC.

13.2.2.2 External Clock Source Mode 1

If SMS is set to 111b, the external clock source mode1 is enabled. When external clock source 1 is enabled, TRGI is selected as the source of CK_PSC. It is worth noting that you need to configure TS to select the source of TRGI. For TS, the following pulses can be used as the clock sources:

1) Internal Trigger (ITRx, x is 0, 1, 2, 3);
2) Signal of compare/capture 1 after passing through the edge detector (TI1F_ED);
3) Signals TI1FP1 and TI2FP2 of compare/capture channel;
4) Signal ETRF from external clock pin.

13.2.2.3 External Clock Source Mode 2

Use external trigger mode 2 to count on every rising or falling edge of the external clock pin input. When the ECE bit is set, the external clock source mode 2 is used. When the external clock source mode2 is used, ETRF is selected as CK_PSC. The ETR pin passes through the optional inverter (ETP) and frequency divider (ETPS) to become ETRP, and then passes through the filter (ETF) to become ETRF.

When ECE bit is set and the SMS is set to 111b, it means that the TS selects ETRF as the input.

13.2.2.4 Encoder Mode

Set SMS as 001b, 010b and 011b to enable the encoder mode. After enabling the encoder mode, you may choose to use another transition edge as a signal for signal output at a certain level in TI1FP1 and TI2FP2. This mode is used when the external encoder is used. Refer to Section 13.3.10 for specific functions.

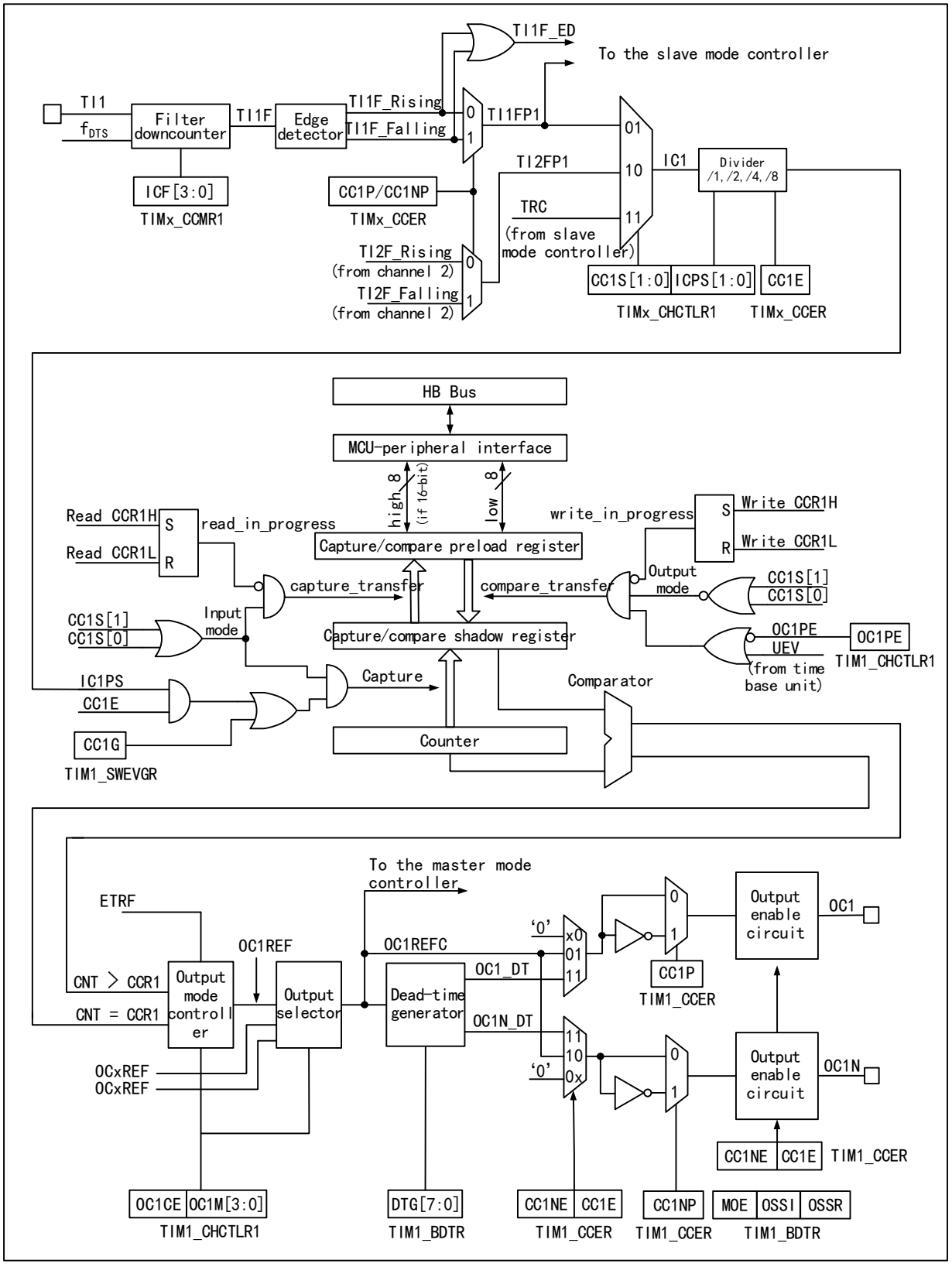
13.2.3 Counter and Periphery

CK_PSC inputs to the prescaler (PSC) for frequency division. PSC is 16 bits, and the actual frequency division factor is equivalent to the value of R16_TIM1_PSC+1. CK_PSC will become CK_CNT after PSC. The changed value of R16_TIM1_PSC will not take effect in real time, but will be updated to the PSC after the update event. Update events include clearing and resetting the UG bit. The core of the timer is a 16-bit counter (CNT). CK_CNT will eventually be inputted to CNT. CNT supports up-counting mode, down-counting mode and up/down counting mode, and there is an automatic reload value register (ATRLR) which re-loads the initial value for CNT after each counting cycle is completed. In addition, there is an auxiliary counter that records the number of times that ATRLR reloads the initial value for CNT. When the number of times reaches the number set in the repeat count register (RPTCR), a specific event can be generated.

13.2.4 Compare/Capture Channel and Periphery

The compare/capture channel is the main component of the timer to achieve complex functions. Its core is the compare/capture register, supplemented by the digital filtering of the peripheral input part, frequency division and channel multiplexing, the output part comparator and output control.

Figure 13-3 Structure block diagram of compare/capture channel



The block diagram of the compare/capture channel is as shown in Figure 13-3. After the signal is inputted from the channel x pin, it can be selected as TIx (the source of TI1 may be more than CH1. See the timer structure block diagram 13-1). Take TI1 as the example: TI1 passes through the filter (ICF[3:0]) to generate TI1F, and then is divided into TI1F_Rising and TI1F_Falling after passing through the edge detector. These 2 signals are selected

(CC1P) to generate TI1FP1, TI1FP1 and TI2FP1 from channel 2 are sent to CC1S together to be selected as IC1, and then sent to the compare/capture register after going through the ICPS frequency division.

The compare/capture register is composed of a preload register and a shadow register, and only the preload register is operated during reading and writing. In the capture mode, the capture occurs on the shadow register, and then copied to the preload register; in the comparison mode, the content of the preload register is copied to the shadow register, and then the content of the shadow register is compared with the core counter (CNT).

13.3 Function and Implementation

The advanced-control timer complex functions are implemented by the operation of compare/capture channel, clock input circuit, counter and peripheral parts of the timer. The timer's clock input can come from multiple clock sources including the input of the compare/capture channel. The operation of compare/capture channel and the clock source selection directly determines its function. The compare/capture channel is bidirectional and can work in input and output modes.

13.3.1 Counter Mode

Incremental counting mode

In incremental counting mode, the counter counts from 0 to the automatic reload value (The contents of the R16_TIM1_ATRLR register), then counts again from 0 and generates an overflow event on the counter.

If a repeat counter is used, an update event (UEV) is generated when the number of iterations of the increment count reaches the number of times programmed in the repeat counter register (R16_TIM1_RPTCR+1). Otherwise, an update event is generated each time the counter is overflowed.

An update event is also generated when you set the UG location 1 of the TIMx_SWEVGR register (through the software or using the slave mode controller).

The UEV event can be disabled by setting UDIS location 1 in the R16_TIM1_CTLR1 register by the software. This avoids updating the shadow register when a new value is written to the preloaded register. No update event occurs until the UDIS bit is written to 0. However, both the counter and the prescaler counter re-count from 0 (while the prescaler ratio remains the same). In addition, if the URS bit (update request selection) in the R16_TIM1_CTLR1 register is set to 1, setting the UG location 1 generates an update event UEV, but does not set the UIF flag to 1 (Therefore, no interrupts or DMA requests are sent). In this way, if the counter is cleared to zero when a capture event occurs, there will be no update interrupt and capture interrupt at the same time.

When an update event occurs, all registers are updated and the update flag (UIF bit in the R16_TIM1_INTFR register) is set to 1 (Depending on the URS bit):
1) The contents of the R16_TIM1_RPTCR register will be reloaded in the repeat counter.
2) The automatic overload shadow register will be updated with the preloaded value (R16_TIM1_ATRLR).
3) The preload value (Contents of the R16_TIM1_PSC register) will be reloaded in the buffer of the prescaler

Decreasing counting mode

In decrement counting mode, the counter decrements the count from the automatic overload value (The contents of the R16_TIM1_ATRLR register) to 0, then re-starts the count from the automatic overload value and generates a counter underflow event.

If a repeat counter is used, an update event (UEV) is generated when the number of repetitions of the decrement count reaches the number of times programmed in the repeat counter register (R16_TIM1_RPTCR+1). Otherwise, an update event is generated each time the counter is overflowed.

An update event is also generated when you set the UG location 1 of the R16_TIM1_EGR register (through the software or using the slave mode controller).

The UEV update event can be disabled by setting UDIS location 1 in the R16_TIM1_CTLR1 register by the software. This avoids updating the shadow register when a new value is written to the preloaded register. No update event occurs until the UDIS bit is written to 0. However, the counter starts counting again from the current automatic reload value, while the prescaler counter starts counting again at 0 (But the pre-division ratio remains the same).

In addition, if the URS bit (Update request selection) in the R16_TIM1_CTLR1 register is set to 1, setting the UG location 1 generates an update event UEV, but does not set the UIF flag to 1 (Therefore, no interrupts or DMA requests are sent). In this way, if the counter is cleared to zero when a capture event occurs, there will be no update interrupt and capture interrupt at the same time.

When an update event occurs, all registers are updated and the update flag (UIF bit in the R16_TIM1_INTFR register) is set to 1 (Depending on the URS bit):
1) the contents of the R16_TIM1_RPTCR register will be reloaded in the repeat counter.
2) the preload value (Contents of the R16_TIM1_PSC register) will be reloaded in the buffer of the prescaler.
3) the automatic reload activity register will be updated with the preloaded value (The contents of the R16_TIM1_ATRLR register).
Note: The automatic overload register is updated before the counter is overloaded, so the next count cycle is the desired new cycle length.

Center alignment mode (Increment/decrement count)
In center alignment mode, the counter counts from 0 to the automatic overload value (The contents of the R16_TIM1_ATRLR register)-1, generating a counter overflow event; then counting down to 1 from the automatic overload value and generating a counter underflow event. Then recount starts at 0.

The center alignment mode is valid when the CMS bit in the R16_TIM1_CTLR1 register is not '00'. When a channel is configured in output mode, its output comparison interrupt flag is set to 1 in the following mode, that is, counter decrement count (Center alignment mode 1, CMS = '01'), counter increment count (Center alignment mode 2, CMS = '10'), and counter increment / decrement count (Center alignment mode 3, CMS = '11').

In this mode, the DIR direction bit of the R16_TIM1_CTLR1 register is not writable, but is updated by the hardware and indicates the current counter direction.

An update event is generated each time a counter overflow and underflow occurs, or an update event can be generated by using UG location 1 in the R16_TIM1_SWEVGR register (Through the software or using the slave mode controller). In this case, the counter and the prescaler counter will start counting again from 0.

The UEV update event can be disabled by setting UDIS location 1 in the R16_TIM1_CTLR1 register by the software. This avoids updating the shadow register when a new value is written to the preloaded register. No update event occurs until the UDIS bit is written to 0. However, the counter still increments and decrements the count based on the current automatic overload value.

In addition, if the URS bit (Update request selection) in the R16_TIM1_CTLR1 register is set to 1, setting the UG

location 1 generates a UEV update event, but the UIF flag is not set to 1 (Therefore, no interrupts or DMA requests are sent). In this way, if the counter is cleared to zero when a capture event occurs, there will be no update interrupt and capture interrupt at the same time.

When an update event occurs, all registers are updated and the update flag (UIF bit in the R16_TIM1_INTFR register) is set to 1 (Depending on the URS bit):
1) The contents of the R16_TIM1_RPTCR register will be reloaded in the repeat counter.
2) The preload value (Contents of the R16_TIM1_PSC register) will be reloaded in the buffer of the prescaler.
3) The automatic reload activity register will be updated with the preloaded value (The contents of the R16_TIM1_ATRLR register). Note that if the update operation is triggered by an overflow on the counter, the automatic overload register is updated before the counter is overloaded, so the next count cycle is the desired new cycle length (The counter is overloaded with new values).

13.3.2 Input Capture Mode

The input capture mode is one of basic functions of timer. The principle of the input capture mode is that when a certain edge on the ICxPS signal is detected, a capture event will occur, and the current value of the counter will be latched into the compare/capture register (R16_TIM1_CHCTLRx). When a capture event occurs, CCxIF (In R16_TIM1_INTFR) bit will be set. If an interrupt or DMA is enabled, a corresponding interrupt or DMA will be generated. If CCxIF is already set when a capture event occurs, then the CCxOF bit will be set. CCxIF can be cleared by software or by hardware through reading the compare/capture register. CCxOF is cleared by the software.

Take an example of channel 1 to illustrate the steps to use the input capture mode, as follows:
1) Configure CC1S and select the source of IC1 signal. For example, it is set to 10b, and TI1FP1 is selected as the source of IC1, and the default setting cannot be used. CC1S defaults to use the compare capture module as the output channel;
2) Configure the IC1F register; these bits set the sampling frequency of the TI1 input and the length of the digital filter. The digital filter consists of an event counter that generates an output transition after recording N events.
3) Configure CC1P bit and set the polarity of TI1FP1. For example, maintain CC1P bit to be low and select the jump of rising edge;
4) Configure IC1PS and set IC1 signal as the frequency division factor between IC1PS. For example, maintain the IC1PS as 00b without frequency division;
5) Configure the CC1E bit to allow to capture the core counter (CNT) value to the compare/capture register.
6) Configure the CC1IE and CC1DE bits as needed to decide whether to enable interrupt or DMA.

After these operations, the compare & capture channel configuration is completed.

When TI1 inputs a captured pulse, the value of the core counter (CNT) will be recorded in the compare/capture register, and CC1IF will be set. When CC1IF has been set before, the CCIOF bit will also be set. If CC1IE is set, then an interrupt will be generated; if CC1DE is set, a DMA request will be generated. An input capture event can be generated by software through writing the event generation register (TIM1_SWEVGR).

13.3.3 Compare Output Mode

The compare output mode is one of basic functions of timer. The principle of the compare output mode is to output a specific change or waveform when the value of the core counter (CNT) is consistent with the value of the compare/capture register. OCxM (in R16_TIM1_CHCTLRx) and the CCxP bit (in R16_TIM1_CCER) determine whether the output is determined high or low level or level inversion. When a comparison consistent event is

generated, the CCxIF bit will be also set. If the CCxIE bit is preset, an interrupt will be generated; if the CCxDE bit is preset, a DMA request will be generated.

The procedure of compare output mode configuration is as follows:
1) Configure the clock source and auto-reload value of the core counter (CNT);
2) Set the count value to be compared to the compare/capture register (R16_TIM1_CHxCVR);
3) If an interrupt needs to be generated, set the CCxIE bit;
4) Keep OCxPE as 0 and disable the preload register of the compare register;
5) Set the output mode, and set OCxM and CCxP bit;
6) Enable the output and set the CCxE bit;
7) Set the CEN bit to start the timer.

13.3.4 Forced Output Mode

The output mode of the compare/capture channel of the timer can be forced to output a certain level by software, instead of relying on the shadow register and the core counter of the compare/capture register.

The specific means is to set OCxM to 100b, which means to force OCxREF to be low; or to set OCxM to 101b, which means setting OCxREF to a high value by force.

It should be noted that if OCxM is set to 100b or 101b by force, the comparison process between the internal core counter and the compare/capture register will be still in progress, the corresponding flag bit will be still set, and interrupts and DMA request will still be generated.

13.3.5 PWM Input Mode

The PWM input mode is used to measure the duty cycle and frequency of the PWM, which is a special case of the input capture mode. The operation is the same as the input capture mode except for the following differences: PWM occupies 2 compare/capture channels, and the input polarity of the 2 channels is set to opposite. One of the signals is set to trigger input, and SMS is set to reset mode.

For example, to measure the cycle and frequency of the PWM wave input from TI1, the following operations are required:
1) Set TI1 (TI1FP1) as the input of IC1 signal. Set CC1S to 01b;
2) Set TI1FP1 as the rising edge valid. Keep CC1P to 0;
3) Set TI1 (TI1FP2) as the input of IC2 signal. Set CC2S to 10b;
2) Set TI1FP2 as the falling edge valid. Set CC2P to 1;
5) The source of the clock source is TI1FP1. Set TS to 101b;
6) Set SMS to reset mode, i.e., 100b;
7) Enable the input capture. Set CC1E and CC2E bits;
In this way, the value of the compare/capture register 1 is the cycle of PWM, and the value of the compare/capture register 2 is its duty cycle.

Note: Since only TI1FP1 and TI2FP2 are connected to the slave mode controller, only TIM1_CH1/TIM1_CH2 can be used for PWM input mode.

13.3.6 PWM Output Mode

The PWM output mode is one of basic functions of timer. The most common method of PWM output mode is to

use the reload value to determine the PWM frequency, and to use the capture comparison register to determine the duty cycle. Set 110b or 111b in OCxM to use PWM mode 1 or mode 2, set the OCxPE bit to enable the preload register, and finally set the ARPE bit. Since the value of the preload register can be sent to the shadow register when an update event occurs, it is necessary to set the UG bit to initialize all registers before the core counter starts counting. In the PWM mode, the core counter and the compare/capture register are always being compared. According to the CMS bit, the timer can output edge-aligned or center-aligned PWM signals.

- Edge alignment

When the edge alignment is used, the core counter counts up or down. In the scenario of PWM mode 1, when the value of the core counter is greater than that of the compare/capture register, OCxREF will be high; when the value of the core counter is less than the compare capture register (such as When the core counter increases to the value of R16_TIM1_ATRLR and returns to all 0s), OCxREF drops to low.

- Central alignment

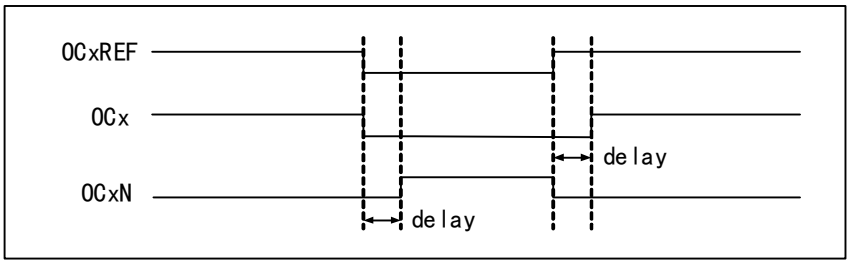
When the center-aligned mode is used, the core counter will run in a mode where up counting and down counting are performed alternately, and OCxREF performs rising and falling jumps when the values of the core counter and the compare/capture register are consistent. However, in 3 types of central alignment mode of comparison flag, the bit setting timing is different somewhat. When the center-alignment mode is used, it is the best to generate a software update flag (setting the UG bit) before starting the core counter.

13.3.7 Complementary Output and Dead-time

The compare/capture channel generally has 2 output pins (compare/capture channel 4 has only one output pin), to output 2 complementary signals (OCx and OCxN). OCx and OCxN can be independently set by the CCxP and CCxNP bits. The output enable is set independently through CCxE and CCxNE, and the dead-time and other controls are performed through the MOE, OIS, OISN, OSSI and OSSR bits. Meanwhile, OCx and OCxN outputs are enabled to insert into the dead-time, each channel has a 10-bit dead-time generator. If there is a break circuit, set the MOE bit. OCx and OCxN are generated by OCxREF in association. If OCx and OCxN are both high and effective, then OCx will be the same as OCxREF, but the rising edge of OCx is equivalent to OCxREF with a delay. OCxN is opposite to OCxREF, and its rising edge has a delay relative to the falling edge of the reference signal, and if the delay is greater than the effective output width, the corresponding pulse will not be generated.

Figure 13-4 shows the relationship between OCx, OCxN and OCxREF, and shows the dead-time.

Figure 13-4 Complementary output and dead-time



13.3.8 Break Signal

When the break signal is generated, the output enable signal and the invalid level will be modified according to the MOE, OIS, OISN, OSSI and OSSR bits. But OCx and OCxN will not be at the effective level at any time. The break event source can come from the break input pin, or it can be a clock failure event, and the clock failure event will be generated by CSS (Clock Security System).

After the system reset, the break function will be disabled by default (MOE bit is low). Setting the BKE bit can enable the break function. The polarity of the input break signal can be set by setting BKP. The BKE and BKP signals can be written at the same time. There will be an APB clock delay before the actual write, so you need to wait for an PB cycle to read the written value correctly.

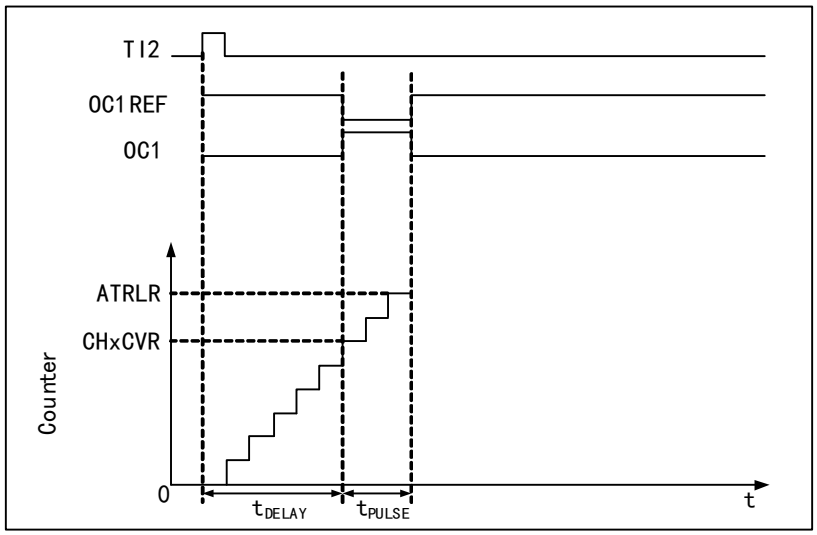
When the selected level appears on the break pin, the system will generate the following actions:
1) The MOE bit is asynchronously cleared, and the output is set to the invalid status, idle status or reset status according to the setting of the OSSI bit;
2) After MOE is cleared, each output channel will output the level determined by OISx;
3) During the supplementary output: the output will be in an invalid status, depending on the polarity;
4) If BIE is set, an interrupt will be generated when BIF is set; if the BDE bit is set, a DMA request will be generated;
5) If AOE is set, the MOE bit will be automatically set during the next update of event UEV.

13.3.9 Single Pulse Mode

The single pulse mode can be used to allow the microcontroller to respond to a specific event to generate a pulse after a delay. The delay and pulse width are programmable. Setting the OPM bit can make the core counter stop when the next update event UEV is generated (the counter turns over to 0).

As shown in Figure 13-5, it is necessary to detect the beginning of a rising edge on the TI2 input pin. After delaying Tdelay, a positive pulse of length Tpulse will be generated on OC1:

Figure 13-5 Single pulse generation



1) Set TI2 as trigger. Set CC2S to 01b and map TI2FP2 to TI2; set CC2P bit to 0b and set TI2FP2 to rising edge detection; set TS to 110b and set TI2FP2 as the trigger source; set SMS to 110b, and TI2FP2 is used to start the counter;
2) Tdelay is determined by the value of the compare/capture register, and Tpulse is determined by the value of the auto-reload value register and the value of the compare/capture register.

13.3.10 Encoder Mode

The encoder mode is a typical application of the timer. It can be used to access the dual-phase output of the encoder. The counting direction of the core counter is synchronized with the rotating shaft of the encoder. Each pulse outputted by the encoder will increase the core counter by adding one or subtracting one. The steps to use the encoder are: set the SMS field to 001b (counting only on TI2 edge), 010b (counting only on TI1 edge) or 011b

(counting on both TI1 and TI2 edges), and connect the encoder to compare/capture channel 1, 2 input terminals, set a value for the reload value register and this value can be set to be greater. In the encoder mode, the internal compare/capture register of timer, prescaler, repeat count register, etc. all work normally. The following table shows the relationship between the counting direction and the encoder signal.

Table 13-1 Relationship between counting direction of timer encoder mode and encoder signal

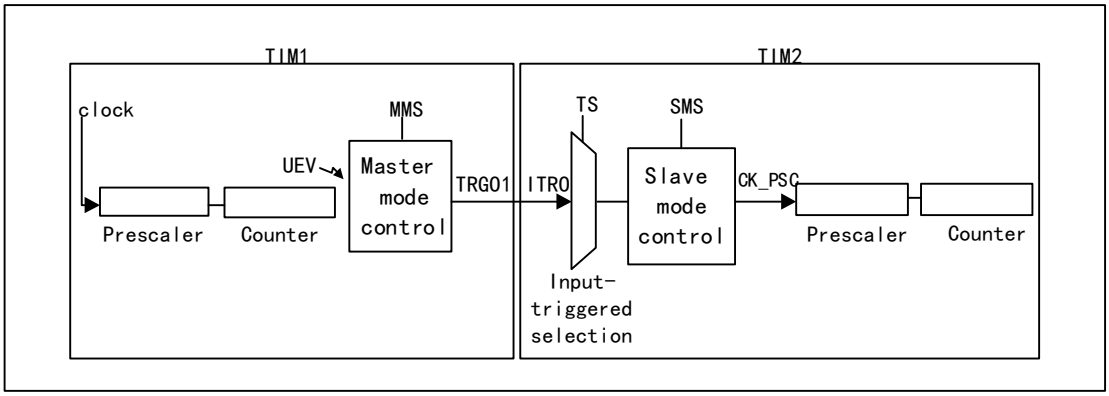
Counting active edge	Relative signal level	TI1FP1 signal edge		TI2FP2 signal	
		Rising edge	Falling edge	Rising edge	Falling edge
Only count at TI1 edge	High	Downcount	Upcount	Not count	
	Low	Upcount	Downcount		
Only count at TI2 edge	High	Not count		Upcount	Downcount
	Low			Downcount	Upcount
Count on both edges of TI1 and TI2	High	Downcount	Upcount	Upcount	Downcount
	Low	Upcount	Downcount	Downcount	Upcount

13.3.11 Timer Synchronization Mode

The TIMx timers are connected together from within to synchronize or cascade the timers. When a timer is configured in master mode, the counter of another timer configured in slave mode can be reset, started, stopped or clocked.

Using one timer as a prescaler for another timer

Figure 13-6 Master/slave timer example



For example, timer 1 can be configured as a prescaler for timer 2. To do this:

1) Configure Timer 1 in Master mode to output a periodic trigger signal every time an update event UEV occurs. If MMS=010 is written to R16_TIM1_CTLR2, TRGO1 will output a rising edge whenever an update event is generated.

2) To connect the TRGO1 output of Timer 1 to Timer 2, Timer 2 must be configured in slave mode, using ITR0 as the internal trigger. This can be selected via the TS bit in the R16_TIM2_SMCFGR register (write TS=000).

3) The slave mode controller is then set to external clock mode 1 (write SMS=111 in the R16_TIM2_SMCFGR register). In this way the clock for timer 2 will be supplied by the rising edge of the periodic trigger signal of timer 1 (corresponding to the counter overflow of timer 1).

4) Finally both timers must be enabled simultaneously by setting the corresponding CEN bits (R16_TIM1_CTLR1 register) of both timers to 1.

Note: If the OCx signal of timer 1 is selected as trigger output (MMS=1xx), the rising edge of this signal will be used to drive the counter of timer 2.

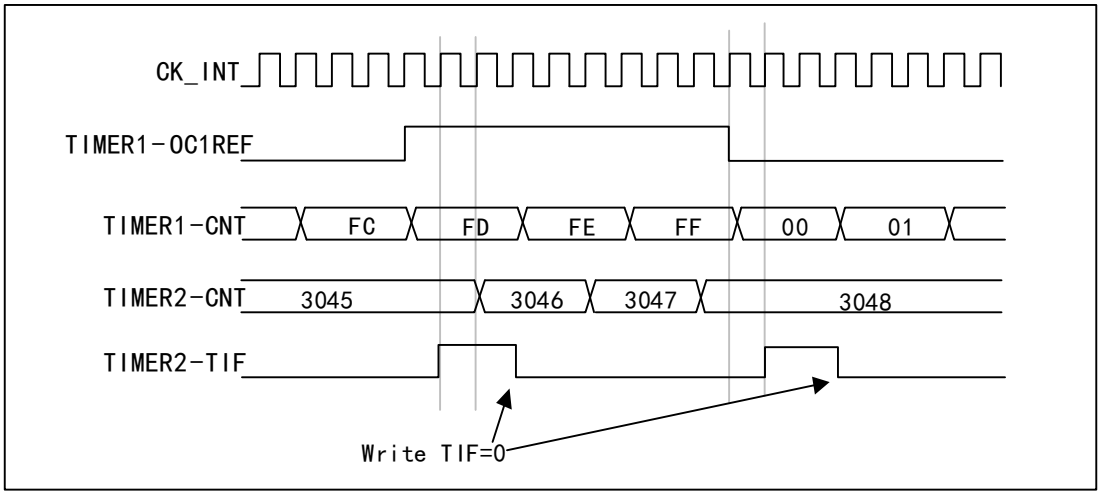
Using one timer to enable another timer

In this example Timer 2 is enabled by comparing the output of Timer 1 with 1. Timer 2 counts according to the divided internal clock only when OC1REF of Timer 1 is high. The clock frequency of both counters is based on CK_INT by prescaler performing a 3-way frequency ($f_{CK\_CNT}=f_{CK\_INT}/3$).

1) Configure timer 1 in main mode and send its output compare 1 reference signal (OC1REF) as trigger output (MMS=100 in R16_TIM1_CR2 register).
2) Configure Timer 1 for OC1REF waveform (R16_TIM1_CCMR1 register).
3) Configure Timer 2 to receive an input trigger from Timer 1 (TS=000 in R16_TIM2_SMCFGR register).
4) Configure Timer 2 to gated mode (SMS=101 in the R16_TIM2_SMCFGR register).
5) Enable Timer 2 by writing a '1' to the CEN bit (R16_TIM2_CTLR1 register).
6) Enable Timer 1 by writing '1' to the CEN bit (R16_TIM1_CTLR1 register).

Note: The clock of Counter 2 is not synchronized with Counter 1, this mode only affects the counter enable signal of Timer 2.

Figure 13-7 Gating Timer 2 using OC1REF of Timer 1



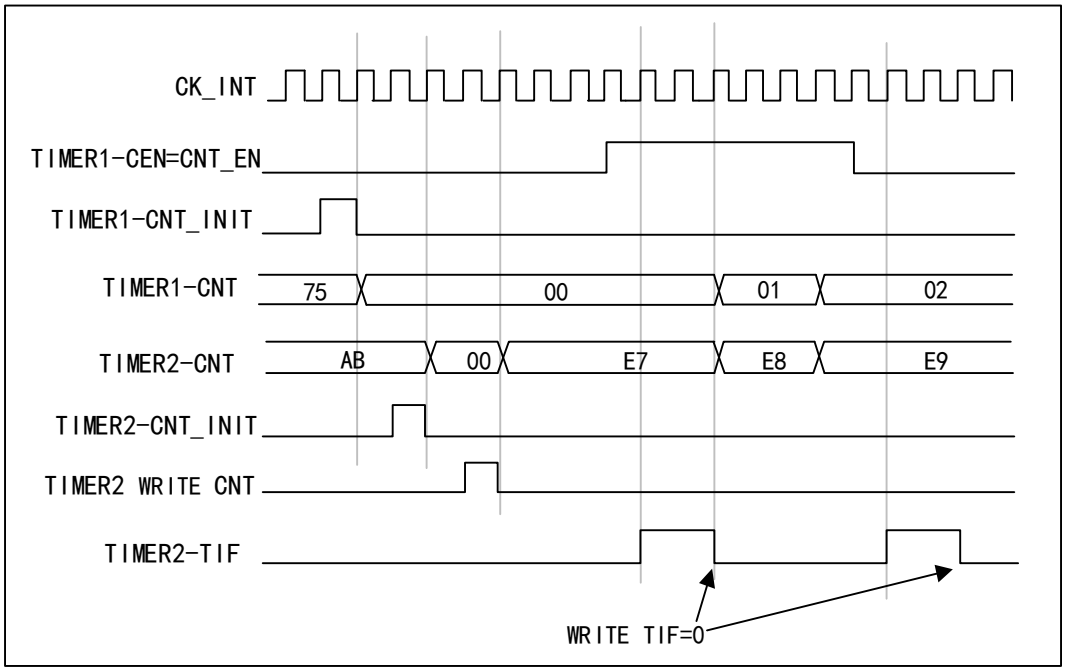
The counter and prescaler of timer 2 are not initialized before start-up. Therefore counting starts from the respective current value. Before starting Timer 1, both timers can be counted from the specified value by resetting them. This makes it possible to write any desired value into the timer counter. Both timers can be easily reset by software using the UG bit in the R16_TIM1_SWEVGR register.

In the next example, timer 1 is synchronized with timer 2. Timer 1 is in master mode and counts from 0. Timer 2 is in slave mode and counts from 0xE7. The prescaler ratios of both timers are the same. When timer 1 is disabled by writing "0" to the CEN bit in the R16_TIM1_CTLR1 register, timer 2 will stop:

1) Configure Timer 1 in main mode and send its output compare 1 reference signal (OC1REF) as the trigger output (MMS=100 in the R16_TIM1_CTLR2 register).
2) Configure Timer 1 for the OC1REF waveform (R16_TIM1_CHCTLR1 register).
3) Configure Timer 2 to receive an input trigger from Timer 1 (TS=000 in R16_TIM2_SMCFGR register).
4) Configure Timer 2 to gated mode (SMS=101 in the R16_TIM2_SMCFGR register).
5) Reset Timer 1 by writing '1' to the UG bit (R16_TIM1_SWEVGR register).

6) Reset Timer 2 by writing '1' to the UG bit (R16_TIM2_SWEVGR register).
7) Initialize Timer 2 to 0xE7 by writing '0xE7' to Timer 2's counter (R16_TIM2_CNTL).
8) Enable Timer 2 by writing '1' to the CEN bit (R16_TIM2_CTLR1 register).
9) Enable Timer 1 by writing '1' to the CEN bit (R16_TIM1_CTLR1 register).
10) Stop Timer 1 by writing '0' to the CEN bit (R16_TIM1_CTLR1 register).

Figure 13-8 Gating Timer 2 using Timer 1's enable signal

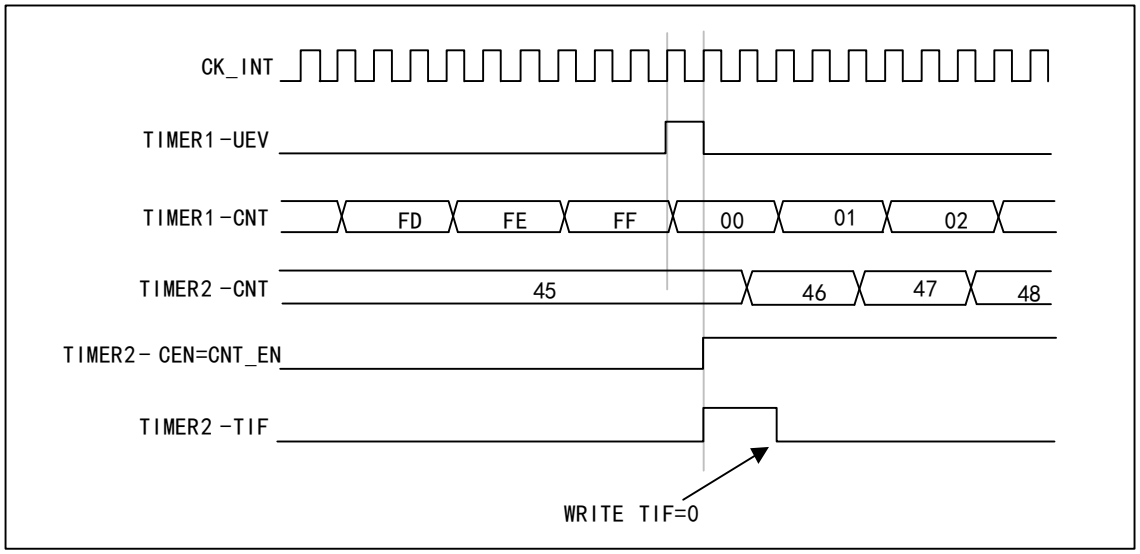


Using one timer to start another timer

This example uses the update event of Timer 1 to enable Timer 2. As soon as Timer 1 generates an update event, Timer 2 starts counting from the current value (which may not be 0) according to the internal clock after dividing the frequency. When Timer 2 receives a trigger signal, its CEN bit is automatically set to 1 and the counter starts counting until a "0" is written to the CEN bit of the R16_TIM2_CTLR1 register and the counter stops counting. The clock frequency of both counters is based on CK_INT and is divided by 3 through a prescaler ($f_{CK\_CNT}=f_{CK\_INT}/3$).

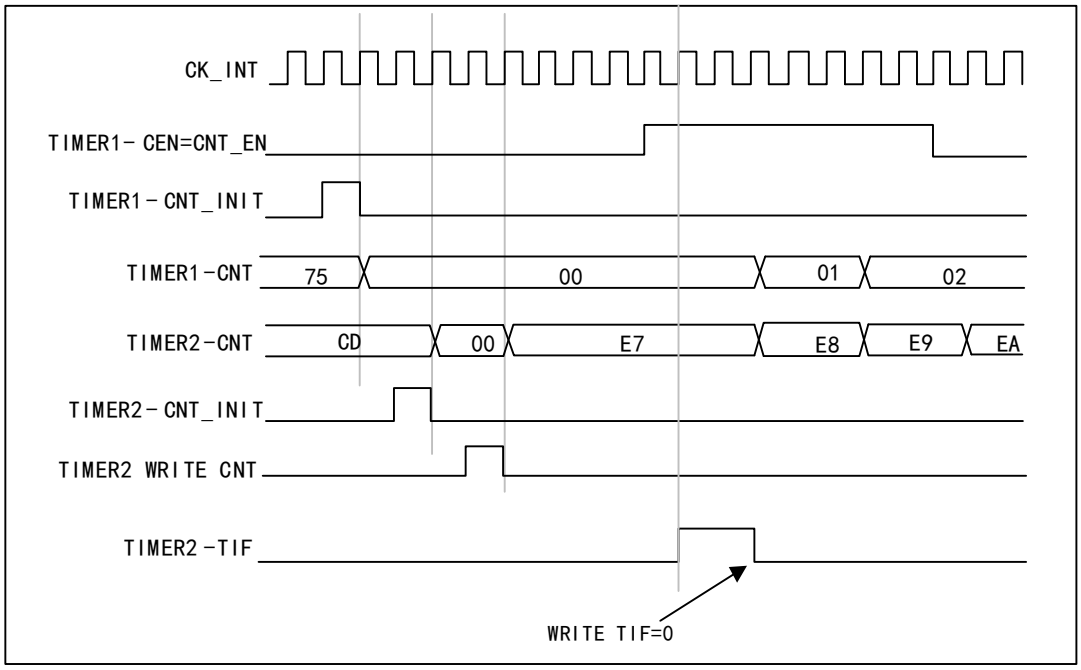
1) Configure timer 1 in main mode and send its update event (UEV) as trigger output (MMS=010 in R16_TIM1_CTLR2 register).
2) Configure the period of timer 1 (R16_TIM1_ATRLR register).
3) Configure Timer 2 to receive an input trigger from Timer 1 (TS=000 in the R16_TIM2_SMCFGR register).
4) Configure Timer 2 to trigger mode (SMS=110 in the R16_TIM2_SMCFGR register).
5) Start Timer 1 by writing '1' to the CEN bit (R16_TIM1_CTLR1 register).

Figure 13-9 Triggering Timer 2 using Timer 1 update event



As shown in the example above, the user can initialize both counters before starting to count. Figure 13-10 shows the same configuration as Figure 13-9, except that the counting behavior is in trigger mode (SMS = 110 in the R16_TIM2_SMCFGR register) rather than gate mode.

Figure 13-10 Triggering Timer 2 using Timer 1's enable signal



Using one timer as a prescaler for another timer

For example, timer 1 can be configured as a prescaler for timer 2. To do this:

1) Configure Timer 1 in main mode, sending its update event (UEV) as a trigger output (MMS=010 in the R16_TIM1_CTLR2 register). This will then output a periodic signal each time the counter overflows.

2) Configure the period of timer 1 (R16_TIM1_ATRLR register).

3) Configure Timer 2 to receive an input trigger from Timer 1 (TS=000 in the R16_TIM2_SMCFGR register).

4) Configure Timer 2 for external clock mode (SMS=111 in the R16_TIM2_SMCFGR register).

5) Start Timer 2 by writing a '1' to the CEN bit (R16_TIM2_CTLR1 register).

6) Timer 1 is started by writing '1' to the CEN bit (R16_TIM1_CTLR1 register).

Synchronized start of 2 timers using an external trigger

In this example, Timer 1 is enabled when there is a rising edge on the TI1 input of Timer 1, and Timer 2 is enabled at the same time as Timer 1. To ensure that the two counters are aligned, Timer 1 must be configured in master/slave mode (corresponding to TI1 as slave and Timer 2 as master):

1) Configure Timer 1 in Master mode, sending its enable signal as a trigger output (MMS=001 in the R16_TIM1_CTLR2 register).

2) Configure Timer 1 as Slave mode to receive the input trigger from TI1 (TS=100 in R16_TIM1_SMCFGR register).

3) Configure Timer 1 to trigger mode (SMS=110 in the R16_TIM1_SMCFGR register).

4) Configure Timer 1 to Master/Slave mode by writing MSM=1 (R16_TIMx_SMCR register).

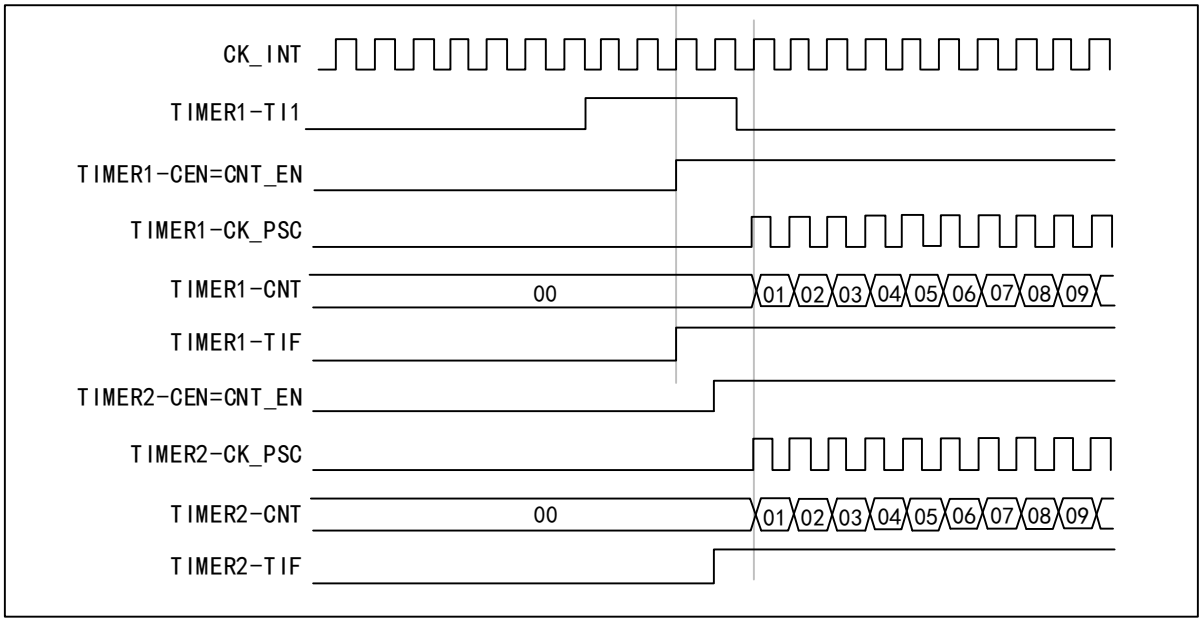
5) Configure Timer 2 to receive an input trigger from Timer 1 (TS=000 in R16_TIM2_SMCFGR register).

6) Configure Timer 2 to trigger mode (SMS=110 in the R16_TIM2_SMCFGR register).

When a rising edge occurs on TI1 (Timer 1), both counters start counting synchronously according to the internal clock and both TIF flags are set to 1.

Note: In this example, both timers are initialized (by setting their respective UG positions to 1) prior to start-up. Both counters count from 0, but an offset can easily be inserted between the two by writing to either counter register (R16_TIMx_CNT). It may be noted that the master/slave mode creates a delay between CNT_EN and CK_PSC for timer 1.

Figure 13-11 Triggering Timer 1 and Timer 2 using Timer 1's TI1 input



Timers are capable of outputting clock pulses (TRGO) and also receiving inputs from other timers (ITRx). The source of ITRx (TRGO from other timers) is different for different timers. The timer internal trigger connections are shown in Table 13-2.

Table 13-2 TIMx internal trigger connections

Slave timer	ITR0(TS=000)	ITR1(TS=001)	ITR2(TS=010)	ITR3(TS=011)
TIM1	-	TIM2	TIM3	TIM4

13.3.13 Debug Mode

When the system enters debug mode, the timer continues to run or stops according to the settings of the DBG module.

13.4 Register Description

Table 13-3 TIM1-related registers list

Name	Access address	Description	Reset value
R16_TIM1_CTLR1	0x40012C00	Control register 1	0x0000
R16_TIM1_CTLR2	0x40012C04	Control register 2	0x0000
R16_TIM1_SMCFGR	0x40012C08	Slave mode control register	0x0000
R16_TIM1_DMAINTENR	0x40012C0C	DMA/interrupt enable register	0x0000
R16_TIM1_INTFR	0x40012C10	Interrupt status register	0x0000
R16_TIM1_SWEVGR	0x40012C14	Event generation register	0x0000
R16_TIM1_CHCTLR1	0x40012C18	Compare/capture control register 1	0x0000
R16_TIM1_CHCTLR2	0x40012C1C	Compare/capture control register 2	0x0000
R16_TIM1_CCER	0x40012C20	Compare/capture enable register	0x0000
R16_TIM1_CNT	0x40012C24	Counters	0x0000
R16_TIM1_PSC	0x40012C28	Counting clock prescaler	0x0000
R16_TIM1_ATRLR	0x40012C2C	Auto-reload value register	0xFFFF
R16_TIM1_RPTCR	0x40012C30	Repeat Count Register	0x0000
R32_TIM1_CH1CVR	0x40012C34	Compare/capture register 1	0x00000000
R32_TIM1_CH2CVR	0x40012C38	Compare/capture register 2	0x00000000
R32_TIM1_CH3CVR	0x40012C3C	Compare/capture register 3	0x00000000
R32_TIM1_CH4CVR	0x40012C40	Compare/capture register 4	0x00000000
R16_TIM1_BDTR	0x40012C44	Brake and dead-time registers	0x0000
R16_TIM1_DMACFGR	0x40012C48	DMA control register	0x0000
R32_TIM1_DMAADR	0x40012C4C	DMA address register for continuous mode	0x0000

13.4.1 Control Register 1 (TIM1_CTLR1)

Offset address: 0x00

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CAP LVL	CAP OV	Reserved				CKD[1:0]		ARP E	CMS[1:0]		DIR	OPM	URS	UDIS	CEN

Bit	Name	Access	Description	Reset value
15	CAPLVL	RW	In double edge capture mode, capture level indication enables: 0: Disable the indication function. 1: Enable indicates function. *Note: after enabling, [16] of CHxCVR indicates the level corresponding to the capture value.*	0
14	CAPOV	RW	Capture value mode configuration.	0

			1: The CHxCVR value is 0xFFFF when a counter overflow is generated before capture. 0: The capture value is the value of the actual counter	
[13:10]	Reserved	RO	Reserved	0
[9:8]	CKD[1:0]	RW	These 2 bits define the division ratio between the timer clock (CK_INT) frequency, the dead-time and the sampling clock used by the dead-time generator and the digital filter (ETR, TIx). 00: Tdts=Tck_int 01: Tdts = 2 × Tck_int 10: Tdts = 4 × Tck_int 11: Reserved.	00b
7	ARPE	RW	Auto-reload preload enable bit. 1: Enable Auto-reload Value Register (ATRLR); 0: Disable Auto-reload Value Register (ATRLR).	0
[6:5]	CMS[1:0]	RW	Central alignment mode selection. 00: Edge-aligned mode. The counter counts up or down based on the direction bit (DIR). 01: Central alignment mode 1. The counter counts up and down alternately. The output compare interrupt flag bit of the channel configured as output (CCxS=00 in the CHCTLRx register) is set only when the counter counts down. 10: Central alignment mode 2. The counter counts up and down alternately. The output compare interrupt flag bit of the channel configured as output (CCxS=00 in the CHCTLRx register) is set only when the counter counts up. 11: Central alignment mode 3. The counter counts up and down alternately. The output compare interrupt flag bit of the channel configured as output (CCxS=00 in the CHCTLRx register) is set when the counter counts both up and down. *Note: When the counter is enabled (CEN=1), the transition from edge-aligned mode to center-aligned mode is not allowed.*	00b
4	DIR	RW	Counter direction: 1: The counting mode of the counter is subtractive; 0: The counting mode of the counter is increment. *Note: This bit is not valid when the counter is configured in central alignment mode or encoder mode.*	0
3	OPM	RW	Single pulse mode. 1: The counter stops (clear the CEN bit) when the	0

			next update event occurs; 0: The counter does not stop when the next update event occurs.	
2	URS	RW	Update request source, by which the software selects the source of the UEV event. 1: If an update interrupt or DMA request is enabled, only an update interrupt or DMA request is generated if the counter overflows/underflows; 0: If an update interrupt or DMA request is enabled, an update interrupt or DMA request is generated by any of the following events. -Counter overflow/underflow -Setting the UG position -Updates generated by the slave mode controller	0
1	UDIS	RW	Updates are prohibited, and the software allows / disables the generation of UEV events through this bit. 1: UEV is prohibited. No update events are generated, and the registers (ARR, PSC, CCRx) hold their values. If the UG bit is set or a hardware reset is issued from the mode controller, the counter and prescaler are reinitialized; 0: UEV is allowed. The UEV event is generated by any of the following events: -Counter overflow / underflow. -Set the UG bit. - Updates generated by the slave mode controller are loaded into their preload values in the cached registers.	0
0	CEN	RW	Enables the counter. 1: Enable the counter; 0: Disable the counter. *Note: The external clock, gated mode and encoder mode will not work until the CEN bit is set in software. Trigger mode can automatically set the CEN bit in hardware.*	0

13.4.2 Control Register 2 (TIM1_CTLR2)

Offset address: 0x04

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	OIS4	OIS3N	OIS3	OIS2N	OIS2	OIS1N	OIS1	TI1S	MMS[2:0]			CCDS	CCUS	Reserved	CCPC

Bit	Name	Access	Description	Reset value
15	Reserved	RO	Reserved	0

14	OIS4	RW	Output idle state 4. 1: When MOE=0, if OC4N is implemented, OC1=1 after dead-time; 0: When MOE=0, if OC4N is implemented, OC1=0 after dead-time. *Note: This bit cannot be modified after LOCK (TIM1_BDTR register) level 1, 2 or 3 has been set.*	0
13	OIS3N	RW	Output idle state 3. 1: When MOE = 0, OC1N = 1 after dead-time. 0: When MOE=0, OC1N=0 after dead-time. *Note: This bit cannot be modified after the LOCK (TIM1_BDTR register) level 1, 2 or 3 has been set.*	0
12	OIS3	RW	Output idle state 3, see OIS4.	0
11	OIS2N	RW	Output idle state 2, see OIS3N.	0
10	OIS2	RW	Output idle state 2, see OIS4.	0
9	OIS1N	RW	Output idle state 1, see OIS3N.	0
8	OIS1	RW	Output idle state 1, see OIS4.	0
7	TI1S	RW	TI1 selection. 1: TIM1_CH1, TIM1_CH2 and TIM1_CH3 pins connected to TI1 input after heterodyning. 0: TIM1_CH1 pin is connected directly to TI1 input.	0
[6:4]	MMS[2:0]	RW	Master mode selection: These 3 bits are used to select the synchronization information (TRGO) sent to the slave timer in master mode. The possible combinations are as follows. 000: The UG bit of the Reset-TIM1_EGR register is used as the trigger output (TRGO). In the case of a reset generated by a trigger input (from a mode controller in reset mode), there is a delay in the signal on TRGO relative to the actual reset. 001: Enable - The counter enable signal CNT_EN is used as a trigger output (TRGO). Sometimes it is necessary to start multiple timers at the same time or to control the enable from timers over a period of time. The counter enable signal is generated by the logical or of the trigger input signal in CEN control bit and gated mode. When the counter enable signal is controlled by a trigger input, there is a delay on TRGO unless master/slave mode is selected (see the description of the MSM bit in the TIM1_SMCR register). 010: Update - The update event is selected as a trigger output (TRGO). For example, the clock of a master timer may be used as a prescaler for a slave timer. 011: Comparison pulse - on the occurrence of a capture	000b

			or a successful comparison, when the CC1IF flag is to be set (even if it is already high), the trigger output sends a positive pulse (TRGO). 100: The comparison-OC1REF signal is used as a trigger output (TRGO). 101: The comparison-OC2REF signal is used as a trigger output (TRGO). 110: The comparison-OC3REF signal is used as a trigger output (TRGO). 111: The comparison -OC4REF signal is used as the trigger output (TRGO).	
3	CCDS	RW	Capture the DMA selection for comparison. 1: Send a DMA request for CHxCVR when an update event occurs. 0: Generate a DMA request for CHxCVR when CHxCVR occurs.	0
2	CCUS	RW	Compare capture control update selection bits. 1: If CCPC is set, they can be updated by setting the COM bit or a rising edge on TRGI. 0: If the CCPC is set, they can only be updated by setting the COM bit. *Note: This bit only works for channels with complementary outputs.*	0
1	Reserved	RO	Reserved	0
0	CCPC	RW	Compare capture preload control bits. 1: CCxE, CCxNE and OCxM bits are preloaded and when this bit is set they are only updated when the COM bit is set. 0: CCxE, CCxNE and OCxM bits are not preloaded. *Note: This bit only works for channels with complementary outputs.*	0

13.4.3 Slave Mode Control Register (TIM1_SMCFGR)

Offset address: 0x08

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ETP	ECE	ETPS[1:0]		ETF[3:0]				MSM	TS[2:0]			Reserved	SMS[2:0]		

Bit	Name	Access	Description	Reset value
15	ETP	RO	ETR trigger polarity selection, this bit selects whether to input ETR directly or to input the inverse of ETR. 1: Invert ETR, low or falling edge active; 0: ETR, active high or rising edge.	0
14	ECE	RW	External clock mode 2 enable selection. 1: Enables external clock mode 2.	0

			0: Disable external clock mode 2. *Note 1: Slave mode can be used simultaneously with external clock mode 2: reset mode, gated mode and trigger mode; however, TRGI cannot be connected to ETRF in this case (TS bit cannot be '111').* *Note 2: When both external clock mode 1 and external clock mode 2 are enabled, the external clock input is ETRF.*	
[13:12]	ETPS[1:0]	RW	The external trigger signal (ETRP) divides the frequency of this signal, which cannot exceed a maximum of 1/4 of the TIMxCLK frequency, and can be downconverted through this domain. 00: Prescaler off. 01: ETRP frequency divided by 2. 10: ETRP frequency divided by 4. 11: ETRP frequency divided by 8.	00b
[11:8]	ETF[3:0]	RW	Externally triggered filtering, in fact, the digital filter is an event counter, which uses a certain sampling frequency to record up to N events and then produces a jump in the output. 0001: Sampling frequency Fsampling=Fck_int, N=2. 0010: Sampling frequency Fsampling=Fck_int, N=4. 0011: Sampling frequency Fsampling=Fck_int, N=8. 0100: Sampling frequency Fsampling = Fdts/2, N = 6. 0101: Sampling frequency Fsampling = Fdts/2, N = 8. 0110: Sampling frequency Fsampling = Fdts/4, N = 6. 0111: Sampling frequency Fsampling = Fdts/4, N = 8. 1000: Sampling frequency Fsampling = Fdts/8, N = 6. 1001: Sampling frequency Fsampling = Fdts/8, N = 8. 1010: Sampling frequency Fsampling = Fdts/16, N = 5. 1011: Sampling frequency Fsampling = Fdts/16, N = 6. 1100: Sampling frequency Fsampling = Fdts/16, N = 8. 1101: Sampling frequency Fsampling = Fdts/32, N = 5. 1110: Sampling frequency Fsampling = Fdts/32, N = 6. 1111: Sampling frequency Fsampling=Fdts/32, N=8.	0000b
7	MSM	RW	Master/slave mode selection. 1: The event on the trigger input (TRGI) is delayed to allow perfect synchronization between the current timer (via TRGO) and its slave timer. This is useful when the synchronization of several timers to a single external event is required. 0: Does not function.	0
[6:4]	TS[2:0]	RW	Trigger selection field, these 3 bits select the trigger input source used to synchronize the counter.	000b

			000: Internal trigger 0 (ITR0). 001: Internal trigger 1 (ITR1). 010: Internal trigger 2 (ITR2). 011: Internal trigger 3 (ITR3). 100: Edge detector of TI1 (TI1F_ED). 101: Filtered timer input 1 (TI1FP1). 110: Filtered timer input 2 (TI2FP2). 111: External trigger input (ETRF). The above only changes when SMS is 0.	
3	Reserved	RO	Reserved.	0
[2:0]	SMS[2:0]	RW	Input mode selection field. Selects the clock and trigger mode of the core counter. 000: Driven by the internal clock CK_INT. 001: Encoder mode 1, where the core counter increments or decrements the count at the edge of TI2FP2 depending on the level of TI1FP1. 010: Encoder mode 2, where the core counter increments or decrements the count at the edge of TI1FP1, depending on the level of TI2FP2. 011: Encoder mode 3, where the core counter increments and decrements the count on the edges of TI1FP1 and TI2FP2 depending on the input level of another signal; 100: reset mode, where the rising edge of the trigger input (TRGI) will initialize the counter and generate a signal to update the registers. 101: Gated mode, when the trigger input (TRGI) is high, the counter clock is turned on; at the trigger input becomes low, the counter is stopped, and the counter starts and stops are controlled. 110: Trigger mode, where the counter is started on the rising edge of the trigger input TRGI and only the start of the counter is controlled. 111: External clock mode 1, rising edge of the selected trigger input (TRGI) drives the counter.	000b

13.4.4 DMA/Interrupt Enable Register (TIM1_DMAINTENR)

Offset address: 0x0C

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	TDE	COMDE	CC4DE	CC3DE	CC2DE	CC1DE	UDE	BIE	TIE	COMIE	CC4IE	CC3IE	CC2IE	CC1IE	UIE

Bit	Name	Access	Description	Reset value
15	Reserved	RO	Reserved	0
14	TDE	RW	Trigger the DMA request enable bit.	0

			1: Allow DMA requests to be triggered. 0: Triggering of DMA requests is disabled.	
13	COMDE	RW	DMA request enable bit of COM. 1: Allow DMA requests for COM. 0: DMA request for COM is disabled.	0
12	CC4DE	RW	Compare the DMA request enable bit of capture channel 4. 1: Allow comparison of DMA requests for capture channel 4. 0: Disable comparison of DMA requests for capture channel 4.	0
11	CC3DE	RW	Compare the DMA request enable bit of capture channel 3. 1: Allow comparison of DMA requests for capture channel 3. 0: Disable comparison of DMA requests for capture channel 3.	0
10	CC2DE	RW	Compare the DMA request enable bit of capture channel 2. 1: Allow comparison of DMA requests for capture channel 2. 0: Disable comparison of DMA requests for capture channel 2.	0
9	CC1DE	RW	Compare the DMA request enable bit of capture channel 1. 1: Allow comparison of DMA requests for capture channel 1. 0: Disable comparison of DMA requests for capture channel 1.	0
8	UDE	RW	Updated DMA request enable bit. 1: DMA requests that allow updates. 0: DMA requests for updates are disabled.	0
7	BIE	RW	Brake interrupt enable bit. 1: Allow brakes to be interrupted. 0: Brake interruption is prohibited.	0
6	TIE	RW	Trigger the interrupt enable bit. 1: Enable triggering of interrupts. 0: Trigger interrupt is disabled.	0
5	COMIE	RW	COM interrupt allow bit. 1: Allow COM interrupts. 0: COM interrupt is disabled.	0
4	CC4IE	RW	Compare capture channel 4 interrupt enable bit. 1: Allow comparison of capture channel 4 interrupts. 0: Disable compare capture channel 4 interrupt.	0

3	CC3IE	RW	Compare capture channel 3 interrupt enable bit. 1: Allow comparison of capture channel 3 interrupts. 0: Disable compare capture channel 3 interrupt.	0
2	CC2IE	RW	Compare capture channel 2 interrupt enable bit. 1: Allow comparison of capture channel 2 interrupts. 0: Disable compare capture channel 2 interrupt.	0
1	CC1IE	RW	Compare capture channel 1 interrupt enable bit. 1: Allow comparison of capture channel 1 interrupts. 0: Disable compare capture channel 1 interrupt.	0
0	UIE	RW	Update the interrupt enable bit. 1: Allow updates to be interrupted. 0: Disable update interruption.	0

13.4.5 Interrupt Status Register (TIM1_INTFR)

Offset address: 0x10

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved			CC4OF	CC3OF	CC2OF	CC1OF	Reserved	BIF	TIF	COMIF	CC4IF	CC3IF	CC2IF	CC1IF	UIF

Bit	Name	Access	Description	Reset value
[15:13]	Reserved	RO	Reserved	0
12	CC4OF	RW0	Compare capture channel 4 repeat capture flag bit.	0
11	CC3OF	RW0	Compare capture channel 3 repeat capture flag bit.	0
10	CC2OF	RW0	Compare capture channel 2 repeat capture flag bit.	0
9	CC1OF	RW0	Compare capture channel 1 repeat capture flag bit is used only when the compare capture channel is configured for input capture mode. This flag is set by hardware and a software write of 0 clears this bit. 1: The value of the counter is captured into the capture comparison register when the status of CC1IF has been set. 0: No duplicate captures are generated.	0
8	Reserved	RO	Reserved	0
7	BIF	RW0	The brake interrupt flag bit, once the brake input is valid, by hardware for this position bit, can be cleared by software. 1: A set valid level is detected on the brake pin input. 0: No braking event is generated.	0
6	TIF	RW0	Trigger interrupt flag bit, when a trigger event occurs by hardware to this location bit, by software to clear. Trigger events include the detection of a valid edge at the TRGI input from a mode other than gated, or any edge in gated mode. 1: Trigger event generation.	0

			0: No trigger event is generated.	
5	COMIF	RW0	COM interrupt flag bit, this bit is set by hardware and cleared by software once a COM event is generated. com events including CCxE, CCxNE, OCxM are updated. 1: COM event generation. 0: No COM event is generated.	0
4	CC4IF	RW0	Compare capture channel 4 interrupt flag bit.	0
3	CC3IF	RW0	Compare capture channel 3 interrupt flag bit.	0
2	CC2IF	RW0	Compare capture channel 2 interrupt flag bit.	0
1	CC1IF	RW0	Compare capture channel 1 interrupt flag bit. If the compare capture channel is configured in output mode. This bit is set by hardware when the counter value matches the comparison value, except in centrosymmetric mode. This bit is cleared by software. 1: The value of the core counter matches the value of compare capture register 1; 0: No match occurs. If compare capture channel 1 is configured as input mode. This bit is set by hardware when a capture event occurs, and it is cleared by software or by reading the compare capture register. 1: The counter value has been captured compare capture register 1. 0: No input capture is generated.	0
0	UIF	RW0	Update interrupt flag bit, this bit is set by hardware when an update event is generated and cleared by software. 1: Update interrupt generation. 0: No update event is generated. The following scenarios generate update events. If UDIS = 0, when the repeat counter value overflows or underflows. If URS = 0, UDIS = 0, when the UG bit is set, or when the counter core counter is reinitialized by software. If URS = 0, UDIS = 0, when the counter CNT is reinitialized by a trigger event.	0

13.4.6 Event Generation Register (TIM1_SWEVGR)

Offset address: 0x14

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

Bit	Name	Access	Description	Reset value
[15:8]	Reserved	RO	Reserved.	0
7	BG	WO	The brake event generation bit, which is set and cleared by software, is used to generate a brake event. 1: Generate a brake event. At this point, MOE=0, BIF=1, if the corresponding interrupt and DMA are enabled, the corresponding interrupt and DMA are generated. 0: No action.	0
6	TG	WO	The trigger event generation bit, which is set by software and cleared by hardware, is used to generate a trigger event. 1: Generate a trigger event, TIF is set, and the corresponding interrupts and DMAs are generated if enabled. 0: No action.	0
5	COMG	WO	Compare capture control update generation bit. Generates a compare capture control update event. This bit is set by software and automatically cleared by hardware. 1: when CCPC = 1, allow updating of CCxE, CCxNE, OCxM bits. 0: No action. *Note: This bit is only valid for channels with complementary outputs (channels 1, 2, 3).*	0
4	CC4G	WO	Compare capture event generation bit 4. generates compare capture event 4.	0
3	CC3G	WO	Compare capture event generation bit 3. generates compare capture event 3.	0
2	CC2G	WO	Compare capture event generation bit 2. generates compare capture event 2.	0
1	CC1G	WO	Compare capture event generation bit 1. generates compare capture event 1. This bit is set by software and cleared by hardware. It is used to generate a compare capture event. 1: Generate a compare capture event on compare capture channel 1. If compare capture channel 1 is configured as output. Set the CC1IF bit. Generate the corresponding interrupts and DMAs if they are enabled. If compare capture channel 1 is configured as input. The current core counter value is captured to compare capture register 1; set the CC1IF bit to generate the	0

			corresponding interrupts and DMAs if they are enabled; if CC1IF is already set, set the CC1OF bit. 0: No action.	
0	UG	WO	Update event generation bit to generate an update event. This bit is set by software and is automatically cleared by hardware. 1: Initialize the counter and generate an update event. 0: No action. *Note: The prescaler counter is also cleared to zero, but the prescaler factor remains unchanged. The core counter is cleared if in centrosymmetric mode or incremental counting mode; if in decremental counting mode, the core counter takes the value of the reload value register.*	0

13.4.7 Compare/Capture Control Register 1 (TIM1_CHCTLR1)

Offset address: 0x18

The channel can be used in input (capture mode) or output (compare mode), and the direction of the channel is defined by the corresponding CCxS bit. The other bits of this register have different roles in input and output modes. OCxx describes the function of the channel in output mode and ICxx describes the function of the channel in input mode.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OC2CE	OC2M[2:0]			OC2PE	OC2FE	CC2S[1:0]		OC1CE	OC1M[2:0]			OC1PE	OC1FE	CC1S[1:0]	
IC2F[3:0]				IC2PSC[1:0]				IC1F[3:0]				IC1PSC[1:0]			

Compare mode (pin direction is output).

Bit	Name	Access	Description	Reset value
15	OC2CE	RW	Compare capture channel 2 clear enable bit. 1: Clear OC2REF bit zero once ETRF input is detected high; 0: OC2REF is not affected by ETRF input.	0
[14:12]	OC2M[2:0]	RW	Compare Capture Channel 2 mode setting field. The 3 bits define the action of the output reference signal OC2REF, which determines the values of OC2, OC2N. OC2REF is active high, while the active levels of OC2 and OC2N depend on the CC2P, CC2NP bits. 000: Freeze. Comparison of the value of the capture register with the value of the comparison between the core counters does not work for OC2REF. 001: force to set to valid level. Forcing OC2REF high when the core counter has the same value as the comparison capture register 2. 010: Force to set to invalid level. Forcing OC2REF low	000b

			when the value of the core counter is the same as the comparison capture register 2. 011: Flip. Flips the level of OC2REF when the core counter is the same as the value of compare capture register 2. 100: Forced to invalid level. Forces OC2REF to low. 101: Forced to valid level. Force OC2REF to high. 110: PWM Mode 1: In up count, once the core counter is smaller than the value of the compare capture register, channel 2 is valid, otherwise it is invalid; in downward counting, once the core counter is larger than the value of the compare capture register, channel 2 is invalid (OC2REF=0), otherwise it is valid (OC2REF=1) 111: PWM mode 2: In up count, once the core counter is smaller than the value of the compare capture register, channel 2 is invalid level, otherwise it is valid level; in down count, once the core counter is larger than the value of the compare capture register, channel 2 is valid level (OC2REF=1), otherwise it is invalid level (OC2REF=0). *Note: This bit cannot be modified once the LOCK level is set to 3 and CC1S=00b. In PWM mode 1 or PWM mode 2, the OC2REF level is changed only when the comparison result is changed or when switching from freeze mode to PWM mode in the output comparison mode.*	
11	OC2PE	RW	Compare Capture Register 2 preload enable bit. 1: Enable the preload function of compare capture register 2, read and write operations only operate on the preload registers, the preload value of compare capture register 2 is loaded into the current shadow register when the update event comes; 0: Disable the preload function of compare capture register 2, compare capture register 2 can be written at any time, and the newly written value takes effect immediately. *Note: Once the LOCK level is set to 3 and CC2S=00, this bit cannot be modified; PWM mode can be used only in single pulse mode (OPM=1) without confirming the pre-load register, otherwise its action is not determined.*	0
10	OC2FE	RW	Compare Capture Channel 2 fast enable bit, this bit is used to speed up the response of the compare capture channel output to a trigger input event.	0

			1: The active edge of the input to the flipflop acts as if a comparison match has occurred. Therefore, the OC is set to the comparison level independent of the comparison result. The delay between the valid edge of the sample trigger and the output of the compare capture channel 2 is reduced to 3 clock cycles. 0: Based on the value of the counter and compare capture register 1, compare capture channel 2 operates normally, even if the flip-flop is open. The minimum delay to activate the compare capture channel 2 output is 5 clock cycles when the input of the flipflop has a valid edge. OC2FE only works when the channel is configured to PWM1 or PWM2 mode.	
[9:8]	CC2S[1:0]	RW	Compare capture channel 2 input selection fields. 00: comparison capture channel 2 is configured as an output. 01: comparison capture channel 2 is configured as an input and IC2 is mapped on TI2. 10: comparison capture channel 2 is configured as an input and IC2 is mapped on TI1. 11: Compare Capture Channel 2 is configured as an input and IC2 is mapped on TRC. This mode works only when the internal trigger input is selected (by the TS bit). *Note: Compare Capture Channel 2 is writable only when the channel is off (when CC2E is zero).*	0
7	OC1CE	RW	Compare capture channel 1 clear enable bit.	0
[6:4]	OC1M[2:0]	RW	Compare capture channel 1 mode setting field.	0
3	OC1PE	RW	Compare capture register 1 preload enable bit.	0
2	OC1FE	RW	Compare capture channel 1 fast enable bit.	0
[1:0]	CC1S[2:0]	RW	Compare capture channel 1 input selection fields.	0

Capture mode (pin direction is input).

Bit	Name	Access	Description	Reset value
[15:12]	IC2F[3:0]	RW	The input capture filter 2 configuration field, these bits set the sampling frequency of the TI1 input and the digital filter length. The digital filter consists of an event counter, which records N events and then generates a jump in the output. 0000: No filter, sampled at fDTS. 1000: Sampling frequency Fsampling = Fdts/8, N = 6. 0001: Sampling frequency Fsampling=Fck_int, N=2. 1001: Sampling frequency Fsampling = Fdts/8, N = 8.	0000b

			0010: Sampling frequency Fsampling=Fck_int, N=4. 1010: Sampling frequency Fsampling = Fdts/16, N = 5. 0011: Sampling frequency Fsampling=f=Fck_int, N=8. 1011: Sampling frequency Fsampling = Fdts/16, N = 6. 0100: Sampling frequency Fsampling = Fdts/2, N = 6. 1100: Sampling frequency Fsampling = Fdts/16, N = 8. 0101: Sampling frequency Fsampling = Fdts/2, N = 8. 1101: Sampling frequency Fsampling = Fdts/32, N = 5. 0110: Sampling frequency Fsampling = Fdts/4, N = 6. 1110: Sampling frequency Fsampling = Fdts/32, N = 6. 0111: Sampling frequency Fsampling = Fdts/4, N = 8. 1111: Sampling frequency Fsampling=Fdts/32, N=8.	
[11:10]	IC2PSC[1:0]	RW	Compare capture channel 2 prescaler configuration field, these 2 bits define the prescaler coefficient for compare capture channel 2. Once CC1E = 0, the prescaler is reset. 00: Without prescaler, one capture is triggered for each edge detected on the capture input. 01: Capture triggered every 2 events. 10: Capture triggered every 4 events. 11: Capture is triggered every 8 events.	00b
[9:8]	CC2S[1:0]	RW	Compare the capture channel 2 input selection field, these 2 bits define the direction of the channel (input/output), and the selection of the input pin. 00: Compare capture channel 1 channel is configured as an output. 01: Compare capture channel 1 channel is configured as an input and IC1 is mapped on TI1. 10: Compare capture channel 1 channel is configured as an input and IC1 is mapped on TI2. 11: Compare capture channel 1 channel is configured as an input and IC1 is mapped on TRC. This mode works only when the internal trigger input is selected (by the TS bit). *Note: CC1S is writable only when the channel is off (CC1E is 0).*	00b
[7:4]	IC1F[3:0]	RW	Input capture filter 1 configuration field.	0
[3:2]	IC1PSC[1:0]	RW	Compare the capture channel 1 prescaler configuration field.	0
[1:0]	CC1S[1:0]	RW	Compare capture channel 1 input selection fields.	0

13.4.8 Compare/Capture Control Register 2 (TIM1_CHCTLR2)

Offset address: 0x1C

The channel can be used in input (capture mode) or output (compare mode), and the direction of the channel is defined by the corresponding CCxS bit. The other bits of this register serve different purposes in input and output modes. OCxx describes the function of the channel in output mode and ICxx describes the function of the channel in input mode.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OC4CE	OC4M[2:0]			OC4PE	OC4FE	CC4S[1:0]		OC3CE	OC3M[2:0]			OC3PE	OC3FE	CC3S[1:0]	
IC4F[3:0]				IC4PSC[1:0]				IC3F[3:0]				IC3PSC[1:0]			

Compare mode (pin direction is output):

Bit	Name	Access	Description	Reset value
15	OC4CE	RW	Compare capture channel 4 clear enable bit.	0
[14:12]	OC4M[2:0]	RW	Compare capture channel 4 mode setting field.	0
11	OC4PE	RW	Compare capture register 4 preload enable bit.	0
10	OC4FE	RW	Compare capture channel 4 fast enable bit.	0
[9:8]	CC4S[1:0]	RW	Compare capture channel 4 input selection fields.	0
7	OC3CE	RW	Compare capture channel 3 clear enable bit.	0
[6:4]	OC3M[2:0]	RW	Compare capture channel 3 mode setting field.	0
3	OC3PE	RW	Compare capture register 3 preload enable bit.	0
2	OC3FE	RW	Compare capture channel 3 fast enable bit.	0
[1:0]	CC3S[1:0]	RW	Compare capture channel 3 input selection fields.	0

Capture mode (pin direction is input).

Bit	Name	Access	Description	Reset value
[15:12]	IC4F[3:0]	RW	Input capture filter 4 configuration field.	0
[11:10]	IC4PSC[1:0]	RW	Compare capture channel 4 prescaler configuration field.	0
[9:8]	CC4S[1:0]	RW	Compare capture channel 4 input selection fields.	0
[7:4]	IC3F[3:0]	RW	Input capture filter 3 configuration field.	0
[3:2]	IC3PSC[1:0]	RW	Compare capture channel 3 prescaler configuration fields.	0
[1:0]	CC3S[1:0]	RW	Compare capture channel 3 input selection fields.	0

13.4.9 Compare/Capture Enable Register (TIM1_CCER)

Offset address: 0x20

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved		CC4P	CC4E	CC3NP	CC3NE	CC3P	CC3E	CC2NP	CC2NE	CC2P	CC2E	CC1NP	CC1NE	CC1P	CC1E

Bit	Name	Access	Description	Reset value
[15:14]	Reserved	RO	Reserved	0

13	CC4P	RW	Compare the capture channel 4 output polarity setting bit.	0
12	CC4E	RW	Compare capture channel 4 output enable bit.	0
11	CC3NP	RW	Compare capture channel 3 complementary output polarity setting bit.	0
10	CC3NE	RW	Compare capture channel 3 complementary output enable bits.	0
9	CC3P	RW	Compare capture channel 3 output polarity setting bit.	0
8	CC3E	RW	Compare capture channel 3 output enable bit.	0
7	CC2NP	RW	Compare capture channel 2 complementary output polarity setting bit.	0
6	CC2NE	RW	Compare capture channel 2 complementary output enable bits.	0
5	CC2P	RW	Compare capture channel 2 output polarity setting bit.	0
4	CC2E	RW	Compare capture channel 2 output enable bit.	0
3	CC1NP	RW	Compare capture channel 1 complementary output polarity setting bit.	0
2	CC1NE	RW	Compare capture channel 1 complementary output enable bit.	0
1	CC1P	RW	Compare the output polarity setting bits of capture channel 1. The CC1 channel is configured to output: 1: OC1 low level effective. 0: OC1 high level is effective. The CC1 channel is configured to enter: The bit selects whether the inverse signal of IC1 or IC1 is used as the trigger or capture signal. 1: Inversion: capture occurs at the falling edge of the IC1; when used as an external trigger, the IC1 is inverted. 0: No inversion: capture occurs on the rising edge of the IC1; when used as an external trigger, the IC1 is not inverted. *Note: Once the LOCK level (the LOCK bit in the TIM1_BDTR register) is set to 3 or 2, the bit cannot be modified.*	0
0	CC1E	RW	Compare the output enable bits of capture channel 1. The CC1 channel is configured to output: 1: On. The OC1 signal is output to the corresponding output pin, and its output level depends on the values of MOE, OSSI, OSSR, OIS1, OIS1N and CC1NE bits. 0: Off. OC1 forbids output, so the output level of OC1 depends on the values of MOE, OSSI, OSSR, OIS1, OIS1N, and CC1NE bits. The CC1 channel is configured to enter: This bit determines whether the value of the counter can be captured into the TIM1_CCR1 register. 1: Capture enable. 0: Capture is prohibited.	0

13.4.10 Counter of Advanced-control Timer (TIM1_CNT)

Offset address: 0x24

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

Bit	Name	Access	Description	Reset value
[15:0]	CNT[15:0]	RW	The real-time value of the timer's counter.	0

13.4.11 Counting Clock Prescaler (TIM1_PSC)

Offset address: 0x28

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

Bit	Name	Access	Description	Reset value
[15:0]	PSC[15:0]	RW	The dividing factor of the prescaler of the timer; the clock frequency of the counter is equal to the input frequency of the divider/(PSC+1).	0

13.4.12 Auto-reload Value Register (TIM1_ATRLR)

Offset address: 0x2C

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

Bit	Name	Access	Description	Reset value
[15:0]	ARR[15:0]	RW	The value of this field will be loaded into the counter, see section 13.2.3 for when the ATRLR acts and updates; the counter stops when the ATRLR is empty.	0xFFFF

13.4.13 Repeat Count Value Register (TIM1_RPTCR)

Offset address: 0x30

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

Bit	Name	Access	Description	Reset value
[15:8]	Reserved	RO	Reserved	0
[7:0]	REP[7:0]	RW	The value of the repeat counter.	0

13.4.14 Compare/Capture Register 1 (TIM1_CH1CVR)

Offset address: 0x34

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															LEVEL1

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

Bit	Name	Access	Description	Reset value
[31:17]	Reserved	RO	Reserved	0
16	LEVEL1	RO	The level indicator bit corresponding to the capture value	0
[15:0]	CCR1[15:0]	RW	Compare/capture register channel 1.	0

13.4.15 Compare/Capture Register 2 (TIM1_CH2CVR)

Offset address: 0x38

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															LEVEL2

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

Bit	Name	Access	Description	Reset value
[31:17]	Reserved	RO	Reserved	0
16	LEVEL2	RO	The level indicator bit corresponding to the capture value	0
[15:0]	CCR1[15:0]	RW	Compare/capture register channel 2.	0

13.4.16 Compare/Capture Register 3 (TIM1_CH3CVR)

Offset address: 0x3C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															LEVEL3

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

Bit	Name	Access	Description	Reset value
[31:17]	Reserved	RO	Reserved	0
16	LEVEL3	RO	The level indicator bit corresponding to the capture value	0
[15:0]	CCR1[15:0]	RW	Compare/capture register channel 3.	0

13.4.17 Compare/Capture Register 4 (TIM1_CH4CVR)

Offset address: 0x40

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															LEVEL4

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

Bit	Name	Access	Description	Reset value
[31:17]	Reserved	RO	Reserved	0
16	LEVEL4	RO	The level indicator bit corresponding to the capture value	0
[15:0]	CCR1[15:0]	RW	Compare/capture register channel 4.	0

13.4.18 Brake and Dead-time Register (TIM1_BDTR)

Offset address: 0x44

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MOE	AOE	BKP	BKE	OSSR	OSSI	LOCK[1:0]		DTG[7:0]							

Bit	Name	Access	Description	Reset value
15	MOE	RW	Main output enable bit. Once the brake signal is active, it will be cleared asynchronously. 1: Allow OCx and OCxN to be set as outputs. 0: Disable the output of OCx and OCxN or force to idle state.	0
14	AOE	RW	Auto output enable. 1: MOE can be set by software or set in the next update event. 0: MOE can only be set by software.	0
13	BKP	RW	The brake input polarity setting bit. 1: Brake input active high. 0: Brake input is active low. *Note: When LOCK level 1 is set, this bit cannot be modified. A write to this bit requires an PB clock before it can take effect.*	0
12	BKE	RW	Brake function enable bit. 1: Turn on the brake input. 0: Brake input is disabled. *Note: When LOCK level 1 is set, this bit cannot be modified. A write to this bit requires an PB clock before it can take effect.*	0
11	OSSR	RW	1: When the timer is not working, once CCxE=1 or CCxNE=1, first turn on OC/OCN and outputinvalid level, then set OCx, OCxN enable output signal=1. 0: When the timer is not operating, OC/OCN output is disabled. *Note: When LOCK level 1 is set, this bit cannot be modified.*	0

10	OSSI	RW	1: when the timer is not operating, once CCxE = 1 or CCxNE = 1, OC/OCN first outputs its idle level, then OCx, OCxN enable output signal = 1. 0: When the timer is not operating, OC/OCN output is disabled. *Note: When LOCK level 1 is set, this bit cannot be modified.*	0
[9:8]	LOCK[1:0]	RW	Lock the function setting field. 00: Disable the locking function. 01: Lock level 1, no DTG, BKE, BKP, AOE, OISx and OISxN bits can be written. 10: Lock level 2, where the bits in lock level 1 cannot be written, nor the CC polarity bits, nor the OSSR and OSSI bits. 11: Lock level 3, cannot write to the bits in lock level 2, and cannot write to the CC control bits. *Note: After system reset, the LOCK bit can only be written once and cannot be modified again until reset.*	0
[7:0]	DTG[7:0]	RW	Dead-time setting bits that define the duration of the dead-time between complementary outputs. Assume that DT denotes its duration. DTG[7:5]=0xx=>DT=DTG[7:0]*Tdtg, Tdtg=TDTS； DTG[7:5]=10x=>DT=(64+DTG[5:0])*Tdtg, Tdtg=2*TDTS; DTG[7:5]=110=>DT=(32+DTG[4:0])*Tdtg, Tdtg =8×TDTS; DTG[7:5]=111=>DT=(32+DTG[4:0])*Tdtg, Tdtg =16*TDTS. *Note: Once the LOCK level (the LOCK[1:0] bit in the TIM1_BDTR register) is set to 1, 2, or 3, these bits cannot be modified.*	0

13.4.19 DMA Control Register (TIM1_DMACFGR)

Offset address: 0x48

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

Bit	Name	Access	Description	Reset value
[15:13]	Reserved	RO	Reserved	0
[12:8]	DBL[4:0]	RW	The length of the DMA continuous transmission, the actual value of which is the value of this field + 1. When the TIM1_DMAADR register is read or written, the timer performs a continuous transfer), that is: defining the number of transfers, which can be either	0

			half-words (two bytes) or bytes: 00000: 1 transmission; 00001: 2 transmissions; 00002: 3 transmissions; ….. 10001: 18 transmissions. Suppose we perform such transmission: DBL=7, DBA=TIM2_CTLR1. If DBL=7, DBA=TIM2_CTLR1 indicates the address of the data to be transferred, then the transferred address is given by the following formula: (address of TIM1_CTLR1) + DBA + (DMA index), where DMA index = D BL, where (the address of TIM1_CTLR1) + DBA plus 7, gives the address where the data will be written or read, so that the data transfer will occur in 7 registers starting from the address (the address of TIM1_CR1) + DBA. Depending on the setting of the DMA data length, the following situations may occur: 1. If the data is set to half word (16 bits), then the data will be transferred to all 7 registers. 2. If the data is set to bytes, the data will still be transferred to all 7 registers: the first register contains the first MSB byte, the second register contains the first LSB byte, and so on. So for timers, the user must specify the data width transferred by DMA.	
[7:5]	Reserved	RO	Reserved	0
[4:0]	DBA[4:0]	RW	These bits define the offset of the DMA in continuous mode from the address where control register 1 is located. 00000: TIM1_CTLR1; 00001: TIM1_CTLR2; 00010: TIM1_SMCFGR; ……	0

13.4.20 DMA Address Register in Continuous Mode (TIM1_DMAADR)

Offset address: 0x4C

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

Bit	Name	Access	Description	Reset value
[15:0]	DMAB[31:0]	RW	The address of the DMA in continuous mode.	0

			Reading or writing the TIM1_DMAADR register will result in an access operation to the register at the following address: TIM1_CTLR1 address + DBA + DMA index, where: 'TIM1_CTLR1 address' is control register 1 (TIM1 _CTLR1); 'DBA' is the base address defined in the TIM1_DMACFGR register; 'DMA index' is the offset automatically controlled by DMA, which depends on the DBL defined in the TIMx_DMACFGR register.	

Chapter 14 General-Purpose Timer (GPTM)

The general timer module consists of two 16-bit automatic reassembling timers (TIM2, TIM3) and a 32-bit automatic reinstalling timer (TIM4), which are used to measure pulse width or generate pulses and PWM waves with specific frequencies. It can be used in automatic control, power supply and other fields.

14.1 Main Features

The main features of a 16-bit general-purpose timer include:

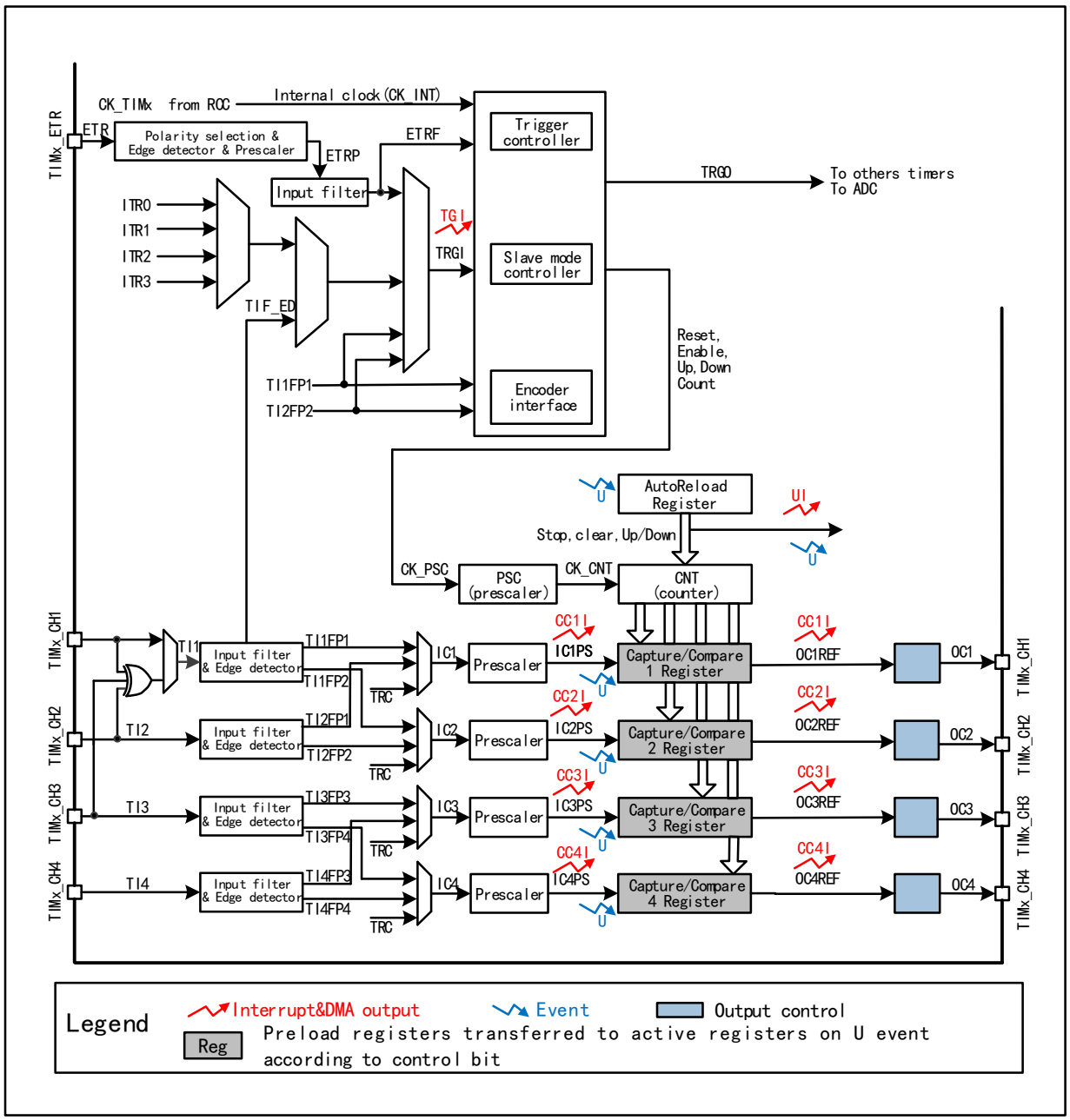
- 16-bit automatic reinstall counter, supporting increasing counting mode, decreasing counting mode and increasing and decreasing counting mode.
- 16-bit prescaler, the frequency division coefficient is dynamically adjustable from 1 to 65536.
- Support 4 independent comparison capture channels.
- Each comparison capture channel supports multiple operating modes, such as input capture, output comparison, PWM generation and single pulse output.
- Support external signal control timer.
- Support using DMA in multiple modes.
- Support incremental coding, cascading and synchronization between timers.

The main features of a 32-bit general-purpose timer include:

- 32-bit automatic reinstall counter, supporting increasing counting mode, decreasing counting mode and increasing and decreasing counting mode.
- 16-bit prescaler, the frequency division coefficient is dynamically adjustable from 1 to 65536.
- Support 4 independent comparison capture channels.
- Each comparison capture channel supports multiple operating modes, such as input capture, output comparison, PWM generation and single pulse output.
- Support external signal control timer.
- Support using DMA in multiple modes.
- Support incremental coding, cascading and synchronization between timers

14.2 Principle and Structure

Figure 14-1 Block diagram of the structure of the general-purpose timer



14.2.1 Overview

As shown in Figure 14-1, the structure of the general-purpose timer can be roughly divided into 3 parts, namely the input clock part, the core counter part and the compare capture channel part.

The clock for the general-purpose timer can come from the HB bus clock (CK_INT), from the external clock input pin (TIMx_ETR), from other timers with clock output (ITRx), and from the input of the compare capture channel (TIMx_CHx). These input clock signals become CK_PSC clocks after various set filtering and dividing operations, etc., and are output to the core counter section. In addition, these complex clock sources can also be output as TRGO to other peripherals such as timers and ADCs.

The core of the general-purpose timer is a 16-bit counter (CNT). After CK_PSC is divided by the prescaler (PSC), it becomes CK_CNT and finally lost to CNT. CNT supports up counting mode, down counting mode and up and down counting mode. In down counting mode, the CNT is reloaded with the initial value after each counting cycle. At the end of each counting period under up-counting, CNT starts counting from 0.

The general-purpose timer has 4 sets of compare capture channels, each of which can input pulses from exclusive pins or output waveforms to pins, i.e., the compare capture channels support both input and output modes. The input of each channel of the compare capture register supports filtering, dividing, edge detection, and other operations, and supports mutual triggering between channels, and can also provide clock for the core counter CNT. Each compare capture channel has a set of compare capture registers (CHxCVR) that support comparison with the main counter (CNT) to output pulses.

14.2.2 Difference between General-purpose Timer and Advanced-control Timer

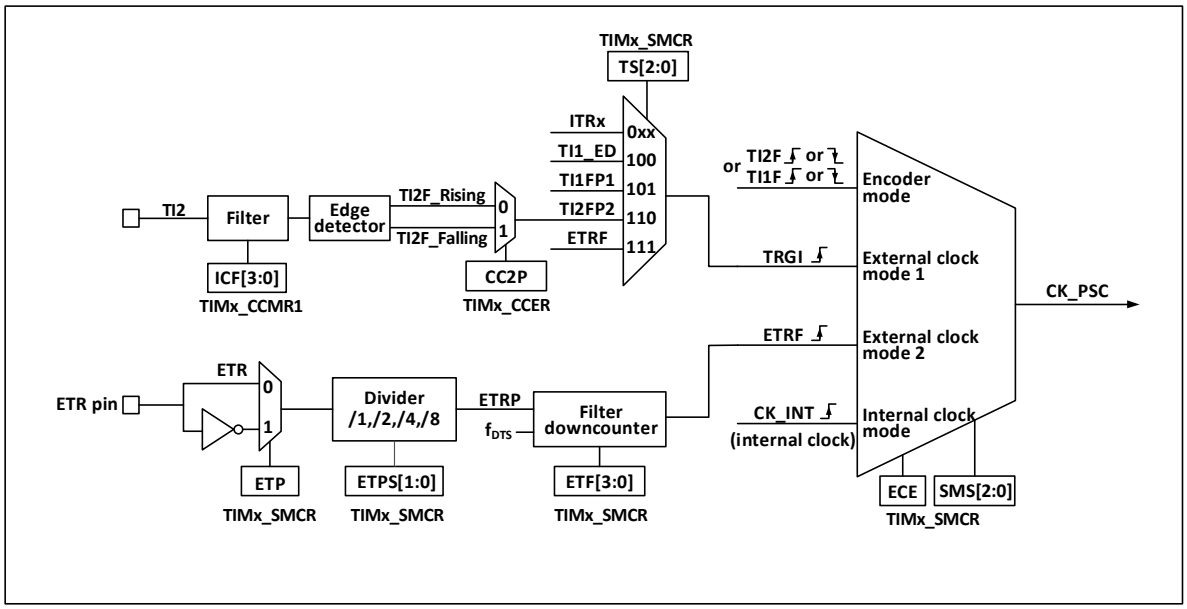
Compared with the advanced-control timer, the general-purpose timer is lack of the following functions:

1) The general-purpose timer lacks a repeated counting register that counts the count cycle of core counter.
2) The compare/capture of general-purpose timer lacks dead-time generation and has no complementary output.
3) The general-purpose timer has no break signal mechanism.

14.2.3 Clock Input

This section describes the source of CK_PSC. The clock source part of the general structure block diagram of the general-purpose timer is abstracted here.

Figure 14-2 Block diagram of general-purpose timer source



The available input clocks can be divided into 4 categories:

1) External clock pin (ETR) input: ETR→ETRP→ETRF;
2) Internal PB clock input: CK_INT;
3) From the compare/capture pin (TIMx_CHx): TIMx_CHx→TIx→TIxFPx; it is also used in encoder mode;
4) Input from other internal timers: ITRx.

The actual operation can be divided into 3 categories by determining the input pulse selection of the SMS from the CK_PSC source:

1) Select the internal clock source (CK_INT);
2) External clock source mode 1;
3) External clock source mode 2;
4) Encoder code.
The 4 clock sources mentioned above can be selected by these 4 operations.

14.2.3.1 Internal Clock Source (CK_INT)

If the general-purpose timer is started when the SMS domain is kept at 000b, then the internal clock source (CK_INT) is selected as the clock. At this moment, CK_INT is CK_PSC.

14.2.3.2 External Clock Source Mode 1

If SMS is set to 111b, the external clock source mode1 is enabled. When external clock source mode1 is enabled, TRGI is selected as the source of CK_PSC. It is worth noting that the user needs to configure TS to select the source of TRGI. For TS, the following pulses can be used as the clock source:
1) Internal Trigger (ITRx, x is 0, 1, 2, 3);
2) Signal of compare/capture1 after passing through the edge detector (TI1F_ED);
3) Signals TI1FP1 and TI2FP2 of compare/capture;
4) Signal ETRF from external clock pin.

14.2.3.3 External Clock Source Mode 2

Use external trigger mode 2 to count on every rising or falling edge of the external clock pin input. When the ECE bit is set, the external clock source mode 2 is enabled. When the external clock source mode 2 is enabled, ETRF is selected as CK_PSC. The ETR pin passes through the optional inverter (ETP) and frequency divider (ETPS) to become ETRP, and then passes through the filter (ETF) to become ETRF.

When the ECE bit is set and the SMS is set to 111b, it means that the TS selects ETRF as the input.

14.2.3.4 Encoder Mode

Set SMS to 001b, 010b and 011b to enable the encoder mode. After enable the encoder mode, you may choose to use another transition edge as a signal for signal output at a certain level in TI1FP1 and TI2FP2. This mode is used when the external encoder is used. Refer to Section 14.3.7 for the specific functions.

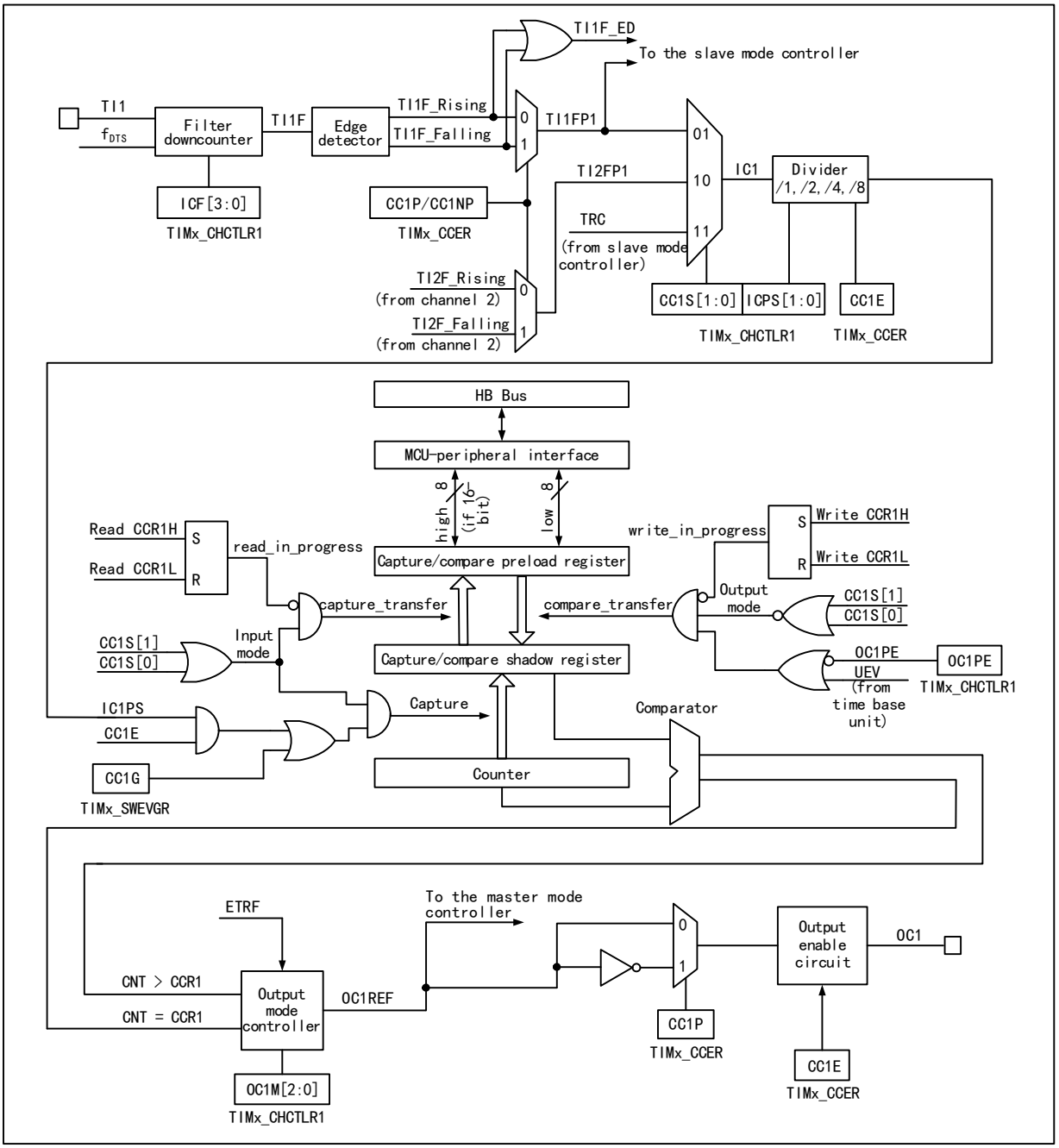
14.2.4 Counter and Periphery

CK_PSC inputs to the prescaler (PSC) for frequency division. PSC has 16 bits, and the actual frequency division factor is equivalent to the value of R16_TIMx_PSC+1. CK_PSC becomes CK_CNT through PSC. The changed value of R16_TIM1_PSC does not take effect in real time, but can be updated to the PSC after the update event. Update events include clearing and resetting the UG bit.

14.2.5 Compare/Capture Channel

The compare/capture is the core of the timer to achieve complex functions. Its core is the compare/capture register, supplemented by the digital filtering of the peripheral input part, frequency division and channel multiplexing, the output part comparator and output control. The block diagram of the compare/capture is as shown in Figure 14-3.

Figure 14-3 Structure block diagram of compare/capture channel



After the signal is input from the channel x pin, it can be selected as TIx (the source of TI1 may be more than CH1. See Figure 15-1 timer block diagram). Take TI1 as the example: TI1 passes through the filter (ICF[3:0]) to generate TI1F, and then is divided into TI1F_Rising and TI1F_Falling after passing through the edge detector. These 2 signals are selected (CC1P) to generate TI1FP1, and TI1FP1 and TI2FP1 from channel 2 are sent to CC1S together to be selected as IC1, and then sent to the compare/capture register after going through the ICPS frequency division.

The compare/capture register is composed of preload register and shadow register, and only the preload register is operated during reading and writing. In the capture mode, the capture occurs on the shadow register, and then copied to the preload register; in the comparison mode, the content of the preload register is copied to the shadow register, and then the content of the shadow register is compared with the core counter (CNT).

14.3 Function and Implementation

The general-purpose timer complex functions are implemented by the operation of compare/capture channel, clock input circuit, counter and peripheral parts of the timer. The timer's clock input can come from multiple clock sources including the input of the compare/capture. The operation of compare/capture register channel and the clock source selection directly determines its function. The compare/capture is bidirectional and can work in input and output modes.

14.3.1 Input Capture Mode

Input capture mode is one of the basic functions of timer. The principle of the input capture mode is that when a determined edge on the ICxPS signal is detected, a capture event is generated and the current value of the counter is locked in the comparison capture register (R16_TIMx_CHCTLRx). When a capture event occurs, the CCxIF (in R16_TIMx_INTFR) is set, and if the interrupt or DMA is enabled, the corresponding interrupt or DMA is generated. If the CCxIF is already set when the capture event occurs, the CCxOF bit will be set. CCxIF can be cleared by software or by hardware by reading comparison capture registers. The CCxOF is cleared by the software.

Give an example of channel 1 to illustrate the steps to use input capture mode, as follows:
1) Configure the ICx domain and select the source of the CCxS signal. For example, if you set it to 10b and select TI1FP1 as the source of the IC1, you cannot use the default setting. The CCxS domain defaults to using the comparison capture module as the output channel.
2) Configure the ICxF domain and set the digital filter for the TI signal. The digital filter will sample a certain frequency at a certain frequency, and then output a transition. This sampling frequency and number of times are determined by ICxF;
3) Configure the CCxP bit and set the polarity of TIxFPx. For example, keep the CC1P bit low and choose the rising edge jump.
4) Configure the ICxPS domain and set the ICx signal to become the frequency division coefficient between ICxPS. For example, keep the ICxPS at 00b without frequency division.
5) Configure the CCxE bit allows capturing the value of the core counter (CNT) into the compare capture register. Set CC1E bit;
6) Configure the CCxIE and CCxDE bits as needed to decide whether to allow enable interrupts or DMA.

At this point, the comparison capture channel configuration has been completed.

When the TI1 inputs a captured pulse, the value of the core counter (CNT) is recorded in the comparison capture register, the CC1IF is set, and the CCIOF bit is set when the CC1IF has been previously set. If the CC1IE bit, an interrupt is generated; if the CC1DE is set, an DMA request is generated. An input capture event can be generated by the software by writing events to generate registers (R16_TIMx_SWEVGR).

14.3.2 Compare Output Mode

The compare output mode is one of basic functions of timer. The principle of the compare output mode is to output a specific change or waveform when the value of the core counter (CNT) is consistent with the value of the compare/capture register. OCxM (in R16_TIMx_CHCTLRx) and the CCxP bit (in R16_TIMx_CCER) determine whether the output is determined high or low level or level inversion. When a compare consistent event is generated, the CCxIF bit will be also set. If the CCxIE bit is preset, an interrupt will be generated; if the CCxDE bit is preset, a DMA request will be generated.

The procedure of compare output mode configuration is as follows:

1) Configure the clock source and auto-reload value of the core counter (CNT);
2) Set the count value to be compared to the compare/capture register (R16_TIMx_CHxCVR);
3) If an interrupt needs to be generated, set the CCxIE bit;
4) Keep OCxPE as 0 and disable the preload register of the compare/capture register;
5) Set the output mode, and set OCxM and CCxP bit;
6) Enable the output and set the CCxE bit;
7) Set the CEN bit and start the timer;

14.3.3 Forced Output Mode

The output mode of the compare/capture of the timer can be forced to output a certain level by software, instead of relying on the shadow register and the core counter of the compare/capture register.

The specific means is to set OCxM to 100b, which means to force OCxREF to be low; or to set OCxM to 101b, which means setting OCxREF to a high value by force.

It shall be noted that if OCxM is set to 100b or 101b by force, the compare process between the internal main counter and the compare/capture register will be still in progress, the corresponding flag bit will be still set, and interrupts and DMA request will still be generated.

14.3.4 PWM Input Mode

The PWM input mode is used to measure the duty cycle and frequency of the PWM, which is a special case of the input capture mode. The operation is the same as the input capture mode except for the following differences: PWM occupies 2 compare/captures, and the input polarity of the 2 channels is set to opposite. One of the signals is set to trigger input, and SMS is set to reset mode.

For example, to measure the cycle and frequency of the PWM wave input from TI1, the following operations are required:
1) Set TI1 (TI1FP1) as the input of IC1 signal. Set CC1S to 01b;
2) Set TI1FP1 as the rising edge valid. Keep CC1P as 0;
3) Set TI1 (TI1FP2) as the input of IC2 signal. Set CC2S to 10b;
2) Set TI1FP2 as the falling edge valid. Set CC2P to 1;
5) The source of the clock source is TI1FP1. Set TS to 101b;
6) Set SMS to reset mode, i.e., 100b;
7) Enable the input capture. Set CC1E and CC2E bits.

Note: Since only TI1FP1 and TI2FP2 are connected to the slave mode controller, only TIMx_CH1/TIMx_CH2 can be used for PWM input mode.

14.3.5 PWM Output Mode

The PWM output mode is one of basic functions of timer. The most common method of PWM output mode is to use the reload value to determine the PWM frequency, and to use the capture comparison register to determine the duty cycle. Set 110b or 111b in OCxM to use PWM mode 1 or mode 2, set the OCxPE bit to enable the preload register, and finally set the ARPE bit. Since the value of the preload register can be sent to the shadow register when an update event occurs, it is necessary to set the UG bit to initialize all registers before the core counter starts counting. In the PWM mode, the core counter and the compare/capture register are always being compared. According to the CMS bit, the timer can output edge-aligned or center-aligned PWM signals.

- Edge alignment

When the edge alignment is used, the core counter counts up or down. In the scenario of PWM mode 1, when the value of the core counter is greater than that of the compare/capture register, OCxREF will rise to be high; when the value of the core counter is less than the compare/capture register (such as When the core counter increases to the value of R16_TIMx_ATRLR and returns to all 0s), OCxREF drops to low.

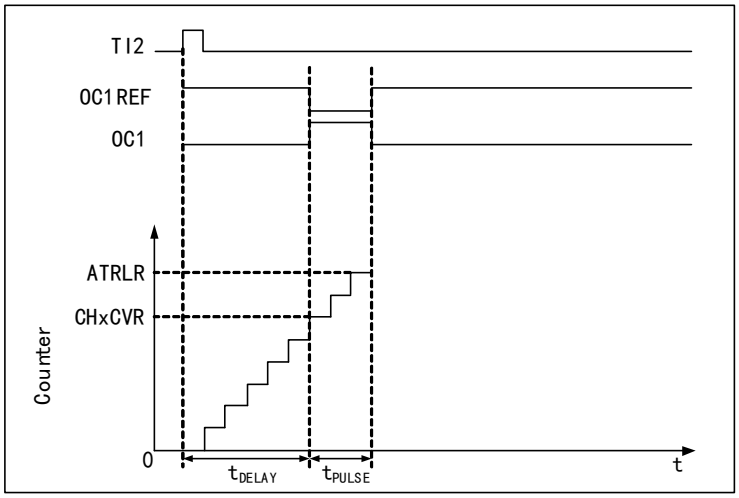
- Central alignment

When the center-aligned mode is used, the core counter will run in a mode where up counting and down counting are performed alternately, and OCxREF performs rising and falling jumps when the values of the core counter and the compare/capture register are consistent. However, in 3 types of central alignment mode, the bit setting timing of comparison flag is different somewhat. When the center-alignment mode is used, it is the best to generate a software update flag (set the UG bit) before starting the core counter.

14.3.6 Single Pulse Mode

The single pulse mode can be used to respond to a specific event to generate a pulse after a delay. The delay and pulse width are programmable. Setting the OPM bit can make the core counter stop when the next update event UEV is generated (the counter turns over to 0).

Figure 14-4 Event generation and pulse response



As shown in Figure 14-4, it is necessary to detect the beginning of a rising edge on the TI2 input pin. After delaying Tdelay, a positive pulse of length Tpulse will be generated on OC1:

1) Set TI2 as trigger. Set the CC2S field to 01b and map TI2FP2 to TI2; set the CC2P bit to 0b and set TI2FP2 to rising edge detection; set the TS field to 110b and set TI2FP2 as the trigger source; set the SMS field to 110b, and TI2FP2 is used to start the counter;

2) Tdelay is defined by the value of the compare/capture register, and Tpulse is determined by the value of the auto-reload value register and the value of the compare/capture register.

14.3.7 Encoder Mode

The encoder mode is a typical application of the timer. It can be used to access the dual-phase output of the encoder. The count direction of the core counter is synchronized with the rotating shaft of the encoder. Each pulse output by the encoder will increase the core counter by adding one or subtracting one. The steps to use the encoder are: set the SMS field to 001b (count only on TI2 edge), 010b (count only on TI1 edge) or 011b (count on both TI1 and TI2 edges), and connect the encoder to compare/capture 1, 2 inputs, set a value for the reload value register and this

value can be set to be greater. In the encoder mode, the internal compare/capture register of timer, prescaler, repeat count register and other registers all work normally. The following table shows the relationship between the counting direction and the encoder signal.

Table 14-1 Relationship between count direction of timer in encoder mode and encoder signal

Count active edge	Relative signal level	TI1FP1 signal edge		TI2FP2 signal edge	
		Rising edge	Falling edge	Rising edge	Falling edge
Only count at TI1 edge	High	Downcount	Upcount	Not count	
	Low	Upcount	Downcount		
Only count at TI2 edge	High	Not count		Upcount	Downcount
	Low			Downcount	Upcount
Count on both edges of TI1 and TI2	High	Downcount	Upcount	Upcount	Downcount
	Low	Upcount	Downcount	Downcount	Upcount

14.3.8 Timer Synchronous Mode

The timer can output clock pulses (TRGO) and can also receive input from other timers (ITRx). The sources of ITRx of different timers (TRGO of other timers) are different. Table 14-2 shows the internal trigger connection of timers.

Figure 14-2 GTPM internal trigger connection

Slave timer	ITR0(TS=000)	ITR1(TS=001)	ITR2(TS=010)	ITR3(TS=011)
TIM2	TIM1	-	TIM3	TIM4
TIM3	TIM1	TIM2	-	TIM4
TIM4	TIM1	TIM2	TIM3	-
-	TIM2	TIM3	TIM4	-

14.3.9 Debug Mode

When the system enters debug mode, the timer continues to run or stops according to the setting of the DBG module.

14.4 Register Description

Table 14-3 TIM2 registers

Name	Offset address	Description	Reset value
R16_TIM2_CTLR1	0x40000000	TIM2 control register1	0x0000
R16_TIM2_CTLR2	0x40000004	TIM2 control register2	0x0000
R16_TIM2_SMCFGR	0x40000008	TIM2 slave mode configuration register	0x0000
R16_TIM2_DMAINTENR	0x4000000C	TIM2 DMA/interrupt enable register	0x0000
R16_TIM2_INTFR	0x40000010	TIM2 interrupt flag register	0x0000
R16_TIM2_SWEVGR	0x40000014	TIM2 event generation register	0x0000
R16_TIM2_CHCTLR1	0x40000018	TIM2 compare/capture control register 1	0x0000
R16_TIM2_CHCTLR2	0x4000001C	TIM2 compare/capture control register 2	0x0000
R16_TIM2_CCER	0x40000020	TIM2 compare/capture enable register	0x0000

R16_TIM2_CNT	0x40000024	TIM2 counter	0x0000
R16_TIM2_PSC	0x40000028	TIM2 count clock prescaler	0x0000
R16_TIM2_ATRLR	0x4000002C	TIM2 auto-reload value register	0xFFFF
R32_TIM2_CH1CVR	0x40000034	TIM2 compare/capture register 1	0x00000000
R32_TIM2_CH2CVR	0x40000038	TIM2 compare/capture register 2	0x00000000
R32_TIM2_CH3CVR	0x4000003C	TIM2 compare/capture register 3	0x00000000
R32_TIM2_CH4CVR	0x40000040	TIM2 compare/capture register 4	0x00000000
R16_TIM2_DMACFGR	0x40000048	TIM2 DMA configuration register	0x0000
R16_TIM2_DMAADR	0x4000004C	TIM2 DMA address register in continuous mode	0x0000

Table 14-4 TIM3 registers

Name	Offset address	Description	Reset value
R16_TIM3_CTLR1	0x40000400	TIM3 control register 1	0x0000
R16_TIM3_CTLR2	0x40000404	TIM3 control register 2	0x0000
R16_TIM3_SMCFGR	0x40000408	TIM3 slave mode configuration register	0x0000
R16_TIM3_DMAINTENR	0x4000040C	TIM3 DMA/interrupt enable register	0x0000
R16_TIM3_INTFR	0x40000410	TIM3 interrupt flag register	0x0000
R16_TIM3_SWEVGR	0x40000414	TIM3 event generation register	0x0000
R16_TIM3_CHCTLR1	0x40000418	TIM3 compare/capture control register 1	0x0000
R16_TIM3_CHCTLR2	0x4000041C	TIM3 compare/capture control register 2	0x0000
R16_TIM3_CCER	0x40000420	TIM3 compare/capture enable register	0x0000
R16_TIM3_CNT	0x40000424	TIM3 counter	0x0000
R16_TIM3_PSC	0x40000428	TIM3 count clock prescaler	0x0000
R16_TIM3_ATRLR	0x4000042C	TIM3 auto-reload value register	0xFFFF
R32_TIM3_CH1CVR	0x40000434	TIM3 compare/capture register 1	0x000000000
R32_TIM3_CH2CVR	0x40000438	TIM3 compare/capture register 2	0x00000000
R32_TIM3_CH3CVR	0x4000043C	TIM3 compare/capture register 3	0x00000000
R32_TIM3_CH4CVR	0x40000440	TIM3 compare/capture register 4	0x00000000
R16_TIM3_DMACFGR	0x40000448	TIM3 DMA configuration register	0x0000
R16_TIM3_DMAADR	0x4000044C	TIM3 DMA address register in continuous mode	0x0000

Table 14-5 TIM4 registers

Name	Offset address	Description	Reset value
R16_TIM4_CTLR1	0x40000800	TIM4 control register1	0x0000
R16_TIM4_CTLR2	0x40000804	TIM4 control register2	0x0000
R16_TIM4_SMCFGR	0x40000808	TIM4 slave mode configuration register	0x0000
R16_TIM4_DMAINTENR	0x4000080C	TIM4 DMA/interrupt enable register	0x0000
R16_TIM4_INTFR	0x40000810	TIM4 interrupt flag register	0x0000
R16_TIM4_SWEVGR	0x40000814	TIM4 event generation register	0x0000
R16_TIM4_CHCTLR1	0x40000818	TIM4 compare/capture control register1	0x0000

R16_TIM4_CHCTLR2	0x4000081C	TIM4 compare/capture control register2	0x0000
R16_TIM4_CCER	0x40000820	TIM4 compare/capture enable register	0x0000
R16_TIM4_CNT	0x40000824	TIM4 counter	0x00000000
R16_TIM4_PSC	0x40000828	TIM4 count clock prescaler	0x0000
R16_TIM4_ATRLR	0x4000082C	TIM4 auto-reload value register	0xFFFFFFFF
R16_TIM4_CH1CVR	0x40000834	TIM4 compare/capture register1	0x00000000
R16_TIM4_CH2CVR	0x40000838	TIM4 compare/capture register2	0x00000000
R16_TIM4_CH3CVR	0x4000083C	TIM4 compare/capture register3	0x00000000
R16_TIM4_CH4CVR	0x40000840	TIM4 compare/capture register4	0x00000000
R16_TIM4_DMACFGR	0x40000848	TIM4 DMA configuration register	0x0000
R16_TIM4_DMAADR	0x4000084C	TIM4 DMA address register in continuous mode	0x00000000

14.4.1 Control Register 1 (TIMx_CTLR1) (x=2/3/4)

Offset address: 0x00

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

Bit	Name	Access	Description	Reset value
15	CAPLVL	RW	In double edge capture mode, capture level indication enables: 1: Enable indicates function; 0: Disable indication function. *Note: (1) This bit applies only to TIM2 and TIM3.* *(2) When TIM2 and TIM3 are enabled, bit [16] of CHxCVR indicates the level corresponding to the capture value.*	0
14	CAPOV	RW	Capture value mode configuration: 1: When a counter overflow occurs before capture, the CHxCVR value is 0xFFFF; 0: Capture value is the value of the actual counter.	0
[13:10]	Reserved	RO	Reserved	0
[9:8]	CKD[1:0]	RW	These 2 bits define the frequency division ratio of timer clock (CK_INT) frequency and sampling clock used for the digital filter: 00: Tdts=Tck_int; 01: Tdts= 2xTck_int; 10: Tdts= 4xTck_int; 11: Reserved.	00b
7	ARPE	RW	Auto-reload and preload enable: 1: Auto-reload value register (ATRLR) enabled; 0: Auto-reload value register (ATRLR) disabled.	0
[6:5]	CMS[1:0]	RW	Central alignment mode selection:	00b

			00: Edge alignment mode. The counter counts up or down according to the direction bit (DIR). 01: Center alignment mode 1. The counter counts up and down alternately. The output comparison interrupt flag bit of the channel configured as an output (CCxS=00 in the CHCTLRx register) is only set when the counter counts down. 10: Center alignment mode 2. The counter counts up and down alternately. The output comparison interrupt flag bit of the channel configured as an output (CCxS=00 in the CHCTLRx register) is only set when the counter counts up. 11: Center alignment mode 3. The counter counts up and down alternately. The output comparison interrupt flag bit of the channel configured as an output (CCxS=00 in the CHCTLRx register) is only set when the counter counts up and down. *Note: When the counter is enabled (CEN=1), it is not allowed to switch from edge alignment mode to center alignment mode.*	
4	DIR	RW	Counter direction: 0: Upcount; 1: Downcount. *Note: When the counter is configured in the center alignment mode or encoder mode, this bit will be invalid.*	0
3	OPM	RW	Single pulse mode. 1: The counter stops when the next update event (clear the CEN bit) occurs; 0: The counter does not stop when the next update event occurs.	0
2	URS	RW	Update request source, software selects the source of UEV events through this bit. 1: If update interrupt or DMA request is enabled, only counter overflow/underflow will generate update interrupt or DMA request; 0: If update interrupts or DMA requests are enabled, any of the following events generates an update interrupt or DMA request: -Counter overflow / underflow. -Set the UG bit. -Updates generated by the slave mode controller. Registers with caches are loaded into their preloaded values.	0
1	UDIS	RW	Updates are prohibited, and the software allows /	0

			disables the generation of UEV events through this bit. 1: UEV is prohibited. No update events are generated, and the registers (ATRLR, PSC, CHCTLRx) hold their values. If the UG bit is set or a hardware reset is issued from the mode controller, the counter and prescaler are reinitialized. 0: UEV is allowed. The UEV event is generated by any of the following events: -Counter overflow / underflow. -Set the UG bit. -Updates generated by the slave mode controller have cached registers loaded with their preloaded values.	
0	CEN	RW	Enable counter (Counter enable). 1: Enable counter; 0: Disables counters. *Note: The external clock, gating mode and encoder mode will not work until the CEN bit is set in the software. The trigger mode automatically sets the CEN bit through the hardware.*	0

14.4.2 Control Register 2 (TIMx_CTLR2) (x=2/3/4)

Offset address: 0x04

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								TI1S	MMS[2:0]			CCDS	Reserved		

Bit	Name	Access	Description	Reset value
[15:8]	Reserved	RO	Reserved	0
7	TI1S	RW	TI1 selection: 1: TIMx_CH1, TIMx_CH2 and TIMx_CH3 pins are connected to TI1 input through XOR; 0: TIMx_CH1 pin is directly connected to TI1 input.	0
[6:4]	MMS[2:0]	RW	Master mode selection: These 3 bits are used to select the synchronization information (TRGO) sent to the slave timer in the master mode. The possible combination is as follows: 000: Reset – The UG bit is used as a trigger output (TRGO). If it is a reset generated by a trigger input (the slave mode controller is in reset mode), the signal on TRGO will have a delay relative to the actual reset; 001: Enable-the counter enables signal CNT_EN to be used as a trigger output (TRGO). Sometimes, it is necessary to start multiple timers at the same time or	0

			control to enable slave timers within a period of time. The counter enable signal is generated by the logical OR of the CEN control bit and the trigger input signal in the gating mode. When the counter enable signal is controlled by the trigger input, there will be a delay on TRGO, unless the master/slave mode is selected (see the description of the MSM bit in the TIMx_SMCFGR register); 010: An update event is selected as the trigger output (TRGO). For example, the clock of a master timer can be used as a prescaler for a slave timer; 011: Comparison pulse, when a capture occurs or a comparison is successful, and the CC1IF flag is to be set (even if it is already high), the trigger output will send a positive pulse (TRGO); 100: OC1REF signal is used as trigger output (TRGO); 101: OC2REF signal is used as trigger output (TRGO); 110: OC3REF signal is used as trigger output (TRGO); 111: OC4REF signal is used as trigger output (TRGO).	
3	CCDS	RW	1: When an update event occurs, send a DMA request of CHxCVR; 0: When CHxCVR occurs, a DMA request of CHxCVR will be generated.	0
[2:0]	Reserved	RO	Reserved.	0

14.4.3 Slave Mode Control Register (TIMx_SMCFGR) (x=2/3/4)

Offset address: 0x08

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ETP	ECE	ETPS[1:0]		ETF[3:0]				MSM	TS[2:0]			Reserved	SMS[2:0]		

Bit	Name	Access	Description	Reset value
15	ETP	RO	ETR trigger polarity selection; this bit selects whether to directly input ETR or input inverted ETR. 1: ETR inverted, active at low level or falling edge; 0: ETR, active at high level or rising edge.	0
14	ECE	RW	External clock mode 2 enable. 1: External clock mode 2 enabled; 0: External clock mode 2 disabled. *Note 1: Slave mode can be used simultaneously with external clock mode 2: reset mode, gating mode and trigger mode; however, TRGI cannot be connected to ETRF at this time (TS bit cannot be 111b).* *Note 2: When both external clock mode 1 and external*	0

			clock mode 2 are enabled at the same time, the input of the external clock will be ETRF.	
[13:12]	ETPS[1:0]	RW	External trigger prescaler (ETRP); the frequency must be at most 1/4 of TIMxCLK frequency, and the frequency can be reduced through this domain. 00: Prescale OFF; 01: ETRP frequency divided by 2; 10: ETRP frequency divided by 4; 11: ETRP frequency divided by 8.	00b
[11:8]	ETF[3:0]	RW	External trigger filter. In fact, the digital filter is an event counter. N events are needed to validate a transition on the output. 0000: No filter, sampling is done at Fdts; 0001: Fsampling=Fck_int, N=2; 0010: Fsampling=Fck_int，N=4; 0011: Fsampling=Fck_int, N=8; 0100: Fsampling=Fdts/2, N=6; 0101: Fsampling=Fdts/2, N=8; 0110: Fsampling=Fdts/4, N=6; 0111: Fsampling=Fdts/4, N=8; 1000: Fsampling=Fdts/8, N=6; 1001: Fsampling=Fdts/8, N=8; 1010: Fsampling=Fdts/16, N=5; 1011: Fsampling=Fdts/16, N=6; 1100: Fsampling=Fdts/16, N=8; 1101: Fsampling=Fdts/32, N=5; 1110: Fsampling=Fdts/32, N=6; 1111: Fsampling=Fdts/32, N=8.	0
7	MSM	RW	Master/Slave mode selection: 1: The event on the trigger input (TRGI) is delayed to allow perfect synchronization between the current timer (via TRGO) and its slave timer. This is very useful when it is required to synchronize several timers to a single external event; 0: Not action.	0
[6:4]	TS[2:0]	RW	Trigger selection; these 3 bits select the trigger input source used to synchronize the counter. 000: Internal trigger 0 (ITR0); 100: Edge detector of TI1 (TI1F_ED); 001: Internal trigger 1 (ITR1); 101: Timer input 1 (TI1FP1) after filtering; 010: Internal trigger 2 (ITR2); 110: Timer input 2 (TI12FP2) after filtering; 011: Internal trigger 3 (ITR3); 111: External trigger input (ETRF);	0

			The values can be changed only when SMS is 0.	
3	Reserved	RO	Reserved	0
[2:0]	SMS[2:0]	RW	Input mode selection. Select the clock and trigger mode of the core counter. 000: Driven by the internal clock CK_INT; 001: Encoder mode1; depending on TI1FP1 level, the core counter counts up or down on edge of TI2FP2; 010: Encoder mode2; depending on TI2FP2 level, the core counter counts up or down on edge of TI1FP1; 011: Encoder mode3; depending on the input level of another signal, the core counter counts up and down on the edge of TI1FP1 and TI2FP2; 100: Reset mode; the rising edge of the trigger input (TRGI) will initialize the counter and generate a signal for updating the register; 101: Gating mode; when the trigger input (TRGI) is high, the clock of the counter will be turned on; when the trigger input becomes low, the counter will stop, and the start and stop of the counter will be controlled; 110: Trigger mode; the counter starts on the rising edge of the trigger input TRGI, and only the start of the counter is controlled; 111: External clock mode1; the rising edge of the selected trigger input (TRGI) drives the counter.	0

14.4.4 DMA/Interrupt Enable Register (TIMx_DMAINTENR) (x=2/3/4)

Offset address: 0x0C

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	TDE	Reserved	CC4DE	CC3DE	CC2DE	CC1DE	UDE	Reserved	TIE	Reserved	CC4IE	CC3IE	CC2IE	CC1IE	UIE

Bit	Name	Access	Description	Reset value
15	Reserved	RO	Reserved	0
14	TDE	RW	Trigger DMA request enable. 1: Trigger DMA request enabled; 0: Trigger DMA request disabled.	0
13	Reserved	RW	Reserved	0
12	CC4DE	RW	DMA request enable of compare/capture4. 1: DMA request of compare/capture4 enabled; 0: DMA request of compare/capture4 disabled.	0
11	CC3DE	RW	DMA request enable of compare/capture3. 1: DMA request of compare/capture3 enabled; 0: DMA request of compare/capture3 disabled.	0
10	CC2DE	RW	DMA request enable of compare/capture2.	0

			1: DMA request of compare/capture2 enabled; 0: DMA request of compare/capture2 disabled.	
9	CC1DE	RW	DMA request enable of compare/capture1. 1: DMA request of compare/capture1 enabled; 0: DMA request of compare/capture1 disabled.	0
8	UDE	RW	Update DMA request enable. 1: Update DMA request enabled; 0: Update DMA request disabled.	0
7	Reserved	RO	Reserved	0
6	TIE	RW	Trigger interrupt enable. 1: Trigger interrupt enabled; 0: Trigger interrupt disabled.	0
5	Reserved	RO	Reserved	0
4	CC4IE	RW	Interrupt enable of compare/capture4. 1: Interrupt of compare/capture4 enabled; 0: Interrupt of compare/capture4 disabled.	0
3	CC3IE	RW	Interrupt enable of compare/capture3. 1: Interrupt of compare/capture3 enabled; 0: Interrupt of compare/capture3 disabled.	0
2	CC2IE	RW	Interrupt enable of compare/capture2. 1: Interrupt of compare/capture2 enabled; 0: Interrupt of compare/capture2 disabled.	0
1	CC1IE	RW	Interrupt enable of compare/capture1. 1: Interrupt of compare/capture1 enabled; 0: Interrupt of compare/capture1 disabled.	0
0	UIE	RW	Update interrupt enable. 1: Update interrupt enabled; 0: Update interrupt disabled.	0

14.4.5 Interrupt Flag Register (TIMx_INTFR) (x=2/3/4)

Offset address: 0x10

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved			CC4OF	CC3OF	CC2OF	CC1OF	Reserved		TIF	Reserved	CC4IF	CC3IF	CC2IF	CC1IF	UIF

Bit	Name	Access	Description	Reset value
[15:13]	Reserved	RO	Reserved	0
12	CC4OF	RW0	Compare/capture4 recapture flag.	0
11	CC3OF	RW0	Compare/capture3 recapture flag.	0
10	CC2OF	RW0	Compare/capture2 recapture flag.	0
9	CC1OF	RW0	Compare/capture1 recapture flag is only used when the compare/capture is configured in the input capture mode. This flag bit is set by the hardware, write 0 by software to clear the bit.	0

			1: When the value of the counter is captured into the capture comparison register, the status of CC1IF has been set; 0: No recapture is generated.	
[8:7]	Reserved	RO	Reserved	0
6	TIF	RW0	Trigger interrupt flag. When a trigger event occurs, set by hardware and cleared by software. Trigger events include the detection of a valid edge at the TRGI input terminal from modes other than gating mode, or any edge in gating mode. 1: Trigger event occurs; 0: No trigger event occurs.	0
5	Reserved	RO	Reserved.	0
4	CC4IF	RW0	Compare/capture 4 interrupt flag.	0
3	CC3IF	RW0	Compare/capture 3 interrupt flag.	0
2	CC2IF	RW0	Compare/capture 2 interrupt flag.	0
1	CC1IF	RW0	Compare/capture 1 interrupt flag. If the compare/capture is configured as the output mode, this bit is set by hardware when the counter value matches the compare value, except in center-aligned mode. This bit is cleared by software. 1: The value of core counter matches the value of compare/capture register 1; 0: No. If the compare/capture is configured as the output mode, this bit is set by hardware when a capture event occurs, and it is cleared by software or cleared by reading the compare/capture register. 1: The counter value has been captured by the compare/capture register 1; 0: No input capture is generated.	0
0	UIF	RW0	Update interrupt flag. When an update event occurs, it is set by hardware and cleared by software. 1: Update interrupt generated; 0: No update interrupt generated. The update event generates in case of the following circumstances: For UDIS=0, when the repeated counter value overflows or underflows; For URS=0, UDIS=0, when the UG bit is set, or when the counter core is reinitialized by software; For URS=0, UDIS=0, when the counter CNT is reinitialized by a trigger event.	0

14.4.6 Event Generation Register (TIMx_SWEVGR) (x=2/3/4)

Offset address: 0x14

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved									TG	Reserved	CC4G	CC3G	CC2G	CC1G	UG

Bit	Name	Access	Description	Reset value
[15:7]	Reserved	RO	Reserved.	0
6	TG	WO	Trigger event generation. Set by software, and cleared by hardware to generate a trigger event. 1: A trigger event generated; if TIF is set and the corresponding interrupt and DMA are enabled, the corresponding interrupt and DMA will be generated; 0: No effect.	0
5	Reserved	RO	Reserved.	0
4	CC4G	WO	Compare/capture 4 generation.	0
3	CC3G	WO	Compare/capture 3 generation.	0
2	CC2G	WO	Compare/capture 2 generation.	0
1	CC1G	WO	Compare/capture1 generation. This bit is set by software and cleared by hardware. It is used to generate a compare/capture event. 1: Generate compare/capture event on channel 1: If compare/capture 1 is configured as output: Set the CC1IF bit. If the corresponding interrupt and DMA are enabled, the corresponding interrupt and DMA will be generated; If compare/capture 1 is configured as input, the current core counter value is captured to compare/capture register 1; set the CC1IF bit, if the corresponding interrupt and DMA are enabled, the corresponding interrupt and DMA will be generated. If the CC1IF bit has been set, set the CC1OF bit. 0: No effect.	0
0	UG	WO	Update event generation. This bit is set by software and cleared automatically by hardware. 1: Initialize the counter and generate an update event; 0: No effect. *Note: The counter of the prescaler is also cleared, but the prescaler factor remains unchanged. In Centro symmetric mode or up-counting mode, the core counter will be cleared; in the down-counting mode, the core counter will take the value of the reload value register.*	0

14.4.7 Compare/Capture Control Register 1 (TIMx_CHCTLR1) (x=2/3/4)

Offset address: 0x18

The channel can be used for input (capture mode) or output (compare mode), and the direction of the channel is defined by the corresponding CCxS bit. The functions of other bits of this register are different in input and output modes. OCxx describes the function of the channel in output mode, and ICxx describes the function of the channel in input mode.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OC2CE	OC2M[2:0]			OC2PE	OC2FE	CC2S[1:0]		OC1CE	OC1M[2:0]			OC1PE	OC1FE	CC1S[1:0]	
IC2F[3:0]				IC2PSC[1:0]				IC1F[3:0]				IC1PSC[1:0]			

Compare mode (pin direction is output):

Bit	Name	Access	Description	Reset value
15	OC2CE	RW	Compare/capture 2 clear enable. 1: Once the ETRF input high level is detected, clear the OC2REF bit to zero; 0: OC2REF is not affected by the ETRF input.	0
[14:12]	OC2M[2:0]	RW	Compare/capture2 mode setting. These 3 bits define the action of the output reference signal OC2REF, and OC2REF determines the value of OC2 and OC2N. OC2REF is active at high level, while the active level of OC2 and OC2N depends on the CC2P and CC2NP bits. 000: Frozen. The comparison value between the value of the compare/capture register and the core counter has no effect on OC2REF; 001: Active by force. When the core counter and compare/capture register1 have the same value, force OC2REF to be high; 010: Set as inactive level by force. When the value of the core counter is the same as compare/capture register 1, force OC2REF to be low; 011: Overturn. When the core counter and compare/capture register1 have the same value, overturn the level of OC2REF; 100: Inactive by force. Force OC2REF to be low. 101: Force to be active level. Force OC2REF to be high. 110: PWM Mode 1: In up count, once the core counter is less than the value of the compare capture register, channel 2 is valid level, otherwise it is invalid level; in downward counting, once the core counter is greater than the value of the compare capture register, channel 2 is invalid level (OC2REF=0), otherwise it is valid level (OC2REF=1).	000b

			111: PWM mode 2: In up count, once the core counter is smaller than the value of the compare capture register, channel 2 is invalid level, otherwise it is valid level; in down count, once the core counter is larger than the value of the compare capture register, channel 2 is valid level (OC2REF=1), otherwise it is invalid level (OC2REF=0). *Note: Once the LOCK level is set to 3 and CC2S=00b, this bit cannot be modified. In PWM mode1 or PWM mode2, the OC2REF level changes only when the comparison result changes or when switching from freezing mode to PWM mode in output compare mode.*	
11	OC2PE	RW	Compare/capture register 2 preload enable. 1: Enable the preload function of the compare/capture register. Read and write operations are only made on the preload register. The preload value of the compare/capture register 2 is loaded into the current shadow register when the update event arrives; 0: Disable the pre-loading function of compare/capture register 2. Compare/capture register 2 can be written at any time, and the newly written value takes effect immediately. *Note: Once the LOCK level is set to 3 and CC2S=00b, this bit cannot be modified; only in single pulse mode (OPM=1) you can use PWM mode without confirming the preload register; otherwise its action is uncertain.*	0
10	OC2FE	RW	Compare/capture 2 fast enable. It is used to speed up the response of the compare/capture output to the trigger input event. 1: The effect of the inactive edge inputted to the trigger is like a comparison match. Therefore, OC is set to the comparison level regardless of the comparison result. The delay between the valid edge of the sampling trigger and the output of compare/capture 2 is shortened to 3 clock cycles; 0: According to the value of counter and compare/capture register 2, compare/capture 2 operates normally, even if the trigger is turned on. When the input of the trigger has a valid edge, the minimum delay for activating the output of the compare/capture 2 is 5 clock cycles. OC2FE only works when the channel is configured in PWM1 or PWM2 mode.	0
[9:8]	CC2S[1:0]	RW	Compare/capture 2 input selection. 00: The compare/capture 2 is configured as output;	00b

			01: Compare/capture 2 is configured as input, and IC2 is mapped on TI2; 10: Compare/capture 2 is configured as input, and IC2 is mapped on TI1; 11: Compare/capture 2 is configured as an input, and IC2 is mapped on TRC. This mode only works when the internal trigger input is selected (selected by the TS bit). *Note: Compare/capture 2 is only writable when the channel is switched off (CC2E is zero).*	
7	OC1CE	RW	Compare/capture 1 clear enable.	0
[6:4]	OC1M[2:0]	RW	Compare/capture 1 mode setting.	0
3	OC1PE	RW	Compare/capture register 1 preload enable.	0
2	OC1FE	RW	Compare/capture 1 fast enable.	0
[1:0]	CC1S[1:0]	RW	Compare/capture 1 input selection.	0

Capture mode (pin direction is input):

Bit	Name	Access	Description	Reset value
[15:12]	IC2F[3:0]	RW	Input capture2 filter configuration. These bits set the sampling frequency and digital filter length of TI1 input. The digital filter is composed of an event counter, in which N events are needed to validate a transition on the output. 0000: No filter, sampling is done at Fdts; 1000: Fsampling=Fdts/8, N=6; 0001: Fsampling=Fck_int, N=2; 1001: Fsampling=Fdts/8, N=8; 0010: Fsampling=Fck_int，N=4; 1010: Fsampling=Fdts/16, N=5; 0011: Fsampling=f=Fck_int, N=8; 1011: Fsampling=Fdts/16, N=6; 0100: Fsampling=Fdts/2, N=6; 1100: Fsampling=Fdts/16, N=8; 0101: Fsampling=Fdts/2, N=8; 1101: Fsampling=Fdts/32, N=5; 0110: Fsampling=Fdts/4, N=6; 1110: Fsampling=Fdts/32, N=6; 0111: Fsampling=Fdts/4, N=8; 1111: Fsampling=Fdts/32, N=8.	0
[11:10]	IC2PSC[1:0]	RW	Compare/capture 2 prescaler configuration. These 2 bits define the prescaler factor of compare/capture 2. Once CC1E=0, the prescaler will be reset. 00: Prescaler OFF, each edge detected on the capture input port triggers a capture;	0

			01: Trigger a capture every 2 events; 10: Trigger a capture every 4 events; 11: Trigger a capture every 8 events.	
[9:8]	CC2S[1:0]	RW	Compare/capture2 input selection. These 2 bits define the direction of the channel (input/output) and selection of input pins. 00: Compare/capture1 is configured as output; 01: Compare/capture1 is configured as input, and IC1 is mapped on TI1; 10: Compare/capture1 is configured as input, and IC1 is mapped on TI2; 11: Compare/capture1 is configured as an input, and IC1 is mapped on TRC. This mode only works when the internal trigger input is selected (selected by the TS bit). *Note: CC1S can be written only when the channel is closed (CC1E is 0).*	0
[7:4]	IC1F[3:0]	RW	Input capture 1 filter configuration.	0
[3:2]	IC1PSC[1:0]	RW	Compare/capture 1 prescaler configuration.	0
[1:0]	CC1S[1:0]	RW	Compare/capture 1 input selection.	0

14.4.8 Compare/Capture Control Register 2 (TIMx_CHCTLR2) (x=2/3/4)

Offset address: 0x1C

The channel can be used for input (capture mode) or output (comparison mode), and the direction of the channel is defined by the corresponding CCxS bit. The functions of other bits of this register are different in input and output modes. OCxx describes the function of the channel in output mode, and ICxx describes the function of the channel in input mode.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OC4CE	OC4M[2:0]			OC4PE	OC4FE	CC4S[1:0]		OC3CE	OC3M[2:0]			OC3PE	OC3FE	CC3S[1:0]	
IC4F[3:0]				IC4PSC[1:0]				IC3F[3:0]				IC3PSC[1:0]			

Compare mode (pin direction is output):

Bit	Name	Access	Description	Reset value
15	OC4CE	RW	Compare/capture 4 clear enable	0
[14:12]	OC4M[2:0]	RW	Compare/capture 4 mode setting	0
11	OC4PE	RW	Compare/capture 4 preload enable	0
10	OC4FE	RW	Compare/capture 4 fast enable	0
[9:8]	CC4S[1:0]	RW	Compare/capture 4 input selection	0
7	OC3CE	RW	Compare/capture 3 clear enable	0
[6:4]	OC3M[2:0]	RW	Compare/capture 3 mode setting	0
3	OC3PE	RW	Compare/capture 3 preload enable	0
2	OC3FE	RW	Compare/capture 3 fast enable	0

[1:0]	CC3S[1:0]	RW	Compare/capture 3 input selection	0

Capture mode (pin direction is input):

Bit	Name	Access	Description	Reset value
[15:12]	IC4F[3:0]	RW	Input capture 4 filter configuration	0
[11:10]	IC4PSC[1:0]	RW	Compare/capture 4 prescaler configuration	0
[9:8]	CC4S[1:0]	RW	Compare/capture 4 input selection	0
[7:4]	IC3F[3:0]	RW	Input capture 3 filter configuration	0
[3:2]	IC3PSC[1:0]	RW	Compare/capture 3 prescaler configuration	0
[1:0]	CC3S[1:0]	RW	Compare/capture 3 input selection	0

14.4.9 Compare/Capture Enable Register (TIMx_CCER) (x=2/3/4)

Offset address: 0x20

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved		CC4P	CC4E	Reserved		CC3P	CC3E	Reserved		CC2P	CC2E	Reserved		CC1P	CC1E

Bit	Name	Access	Description	Reset value
[15:14]	Reserved	RO	Reserved	0
13	CC4P	RW	Compare/capture 4 output polarity	0
12	CC4E	RW	Compare/capture 4 output enable	0
[11:10]	Reserved	RO	Reserved	0
9	CC3P	RW	Compare/capture 3 output polarity	0
8	CC3E	RW	Compare/capture 3 output enable	0
[7:6]	Reserved	RO	Reserved	0
5	CC2P	RW	Compare/capture 2 output polarity	0
4	CC2E	RW	Compare/capture 2 output enable	0
[3:2]	Reserved	RO	Reserved	0
1	CC1P	RW	Compare/capture 1 output polarity CC1 channel is configured as output: 1: OC1 is active at low level; 0: OC1 is active at high level. CC1 channel configured as input: This bit selects whether IC1 or the inverted signal of IC1 is used as the trigger or capture signal. 1: Invert: Capture occurs on the falling edge of IC1; when used as an external trigger, IC1 inverts; 0: No inversion: Capture occurs on the rising edge of IC1; when used as an external trigger, IC1 does not invert.	0
0	CC1E	RW	Compare/capture 1 output enable CC1 channel is configured as output: 1: On: OC1 signal is output to the corresponding output	0

			pin; 0: Off: OC1 output is disabled. CC1 channel configured as input: This bit determines whether the counter value can be captured into the TIMx_CCR1 register. 1: Capture enabled; 0: Capture disabled.	

14.4.10 Counter of General-purpose Timer (TIMx_CNT) (x=2/3)

Offset address: 0x24

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

Bit	Name	Access	Description	Reset value
[15:0]	CNT[15:0]	RW	Real-time value of timer counter.	0

14.4.11 Counter of General-purpose Timer (TIMx_CNT) (x=4)

Offset address: 0x24

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

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

Bit	Name	Access	Description	Reset value
[31:0]	CNT[31:0]	RW	Real-time value of timer counter.	0

14.4.12 Count Clock Prescaler (TIMx_PSC) (x=2/3/4)

Offset address: 0x28

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

Bit	Name	Access	Description	Reset value
[15:0]	PSC[15:0]	RW	The frequency division factor of the timer's prescaler; the clock frequency of the counter is equal to the input frequency of the frequency divider/(PSC+1).	0

14.4.13 Auto-reload Value Register (TIMx_ATRLR) (x=2/3)

Offset address: 0x2C

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

Bit	Name	Access	Description	Reset value
[15:0]	ARR[15:0]	RW	The value of ATRLR[15:0] is loaded into the counter. Please refer to Section 15.2.4 for ATRLR acting and update time. When ATRLR is empty, the counter stops.	0xFFFF

14.4.14 Auto-reload Value Register (TIMx_ATRLR) (x=4)

Offset address: 0x2C

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

Bit	Name	Access	Description	Reset value
[31:0]	ARR[31:0]	RW	The value of ATRLR[31:0] is loaded into the counter. Please refer to Section 14.2.4 for ATRLR acting and update time. When ATRLR is empty, the counter stops.	0xFFFFFFFF

14.4.15 Compare/Capture Register 1 (TIMx_CH1CVR) (x=2/3)

Offset address: 0x34

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															LEVEL1
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR1[15：0]															

Bit	Name	Access	Description	Reset value
[31:17]	Reserved	RO	Reserved	0
16	LEVEL1	RO	The level indicator bit corresponding to the capture value	0
[15:0]	CCR1[15:0]	RW	Compare/capture register channel 1.	0

14.4.16 Compare/Capture Register 1 (TIMx_CH1CVR) (x=4)

Offset address: 0x34

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CCR1[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR1[15:0]															

Bit	Name	Access	Description	Reset value

[31:0]	CCR1[31:0]	RW	Compare/capture register channel 1.	0

14.4.17 Compare/Capture Register 2 (TIMx_CH2CVR) (x=2/3)

Offset address: 0x38

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															LEVEL2
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR2[15:0]															

Bit	Name	Access	Description	Reset value
[31:17]	Reserved	RO	Reserved	0
16	LEVEL2	RO	The level indicator bit corresponding to the capture value	0
[15:0]	CCR2[15:0]	RW	Compare/capture register channel 2.	0

14.4.18 Compare/Capture Register2 (TIMx_CH2CVR) (x=4)

Offset address: 0x38

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CCR2[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR2[15:0]															

Bit	Name	Access	Description	Reset value
[31:0]	CCR2[31:0]	RW	Compare/capture register channel 2.	0

14.4.19 Compare/Capture Register3 (TIMx_CH3CVR) (x=2/3)

Offset address: 0x3C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															LEVEL3
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR3[15:0]															

Bit	Name	Access	Description	Reset value
[31:17]	Reserved	RO	Reserved	0
16	LEVEL3	RO	The level indicator bit corresponding to the capture value	0
[15:0]	CCR3[15:0]	RW	Compare/capture register channel 3.	0

14.4.20 Compare/Capture Register3 (TIMx_CH3CVR) (x=4)

Offset address: 0x3C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CCR3[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR3[15:0]															

Bit	Name	Access	Description	Reset value
[31:0]	CCR3[31:0]	RW	Compare/capture register channel 3.	0

14.4.21 Compare/Capture Register4 (TIMx_CH4CVR) (x=2/3)

Offset address: 0x40

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															LEVEL4
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR4[15:0]															

Bit	Name	Access	Description	Reset value
[31:17]	Reserved	RO	Reserved	0
16	LEVEL4	RO	The level indicator bit corresponding to the capture value	0
[15:0]	CCR4[15:0]	RW	Compare/capture register channel 4.	0

14.4.22 Compare/Capture Register4 (TIMx_CH4CVR) (x=4)

Offset address: 0x40

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CCR4[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR4[15:0]															

Bit	Name	Access	Description	Reset value
[31:0]	CCR4[31:0]	RW	Compare/capture register channel 4.	0

14.4.23 DMA Control Register (TIMx_DMACFGR) (x=2/3/4)

Offset address: 0x48

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

Bit	Name	Access	Description	Reset value
[15:13]	Reserved	RO	Reserved	0
[12:8]	DBL[4:0]	RW	Length of data that DMA continuously transfers; the actual value is the value of this domain + 1.	0
[7:5]	Reserved	RO	Reserved.	0
[4:0]	DBA[4:0]	RW	These bits define the offset of DMA from the address of control register1 in continuous mode.	0

14.4.24 DMA Address Register in Continuous Mode (TIMx_DMAADR) (x=2/3)

Offset address: 0x4C

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

Bit	Name	Access	Description	Reset value
[31:0]	DMAB[15:0]	RW	DMA address in continuous mode.	0

14.4.25 DMA Address Register in Continuous Mode (TIMx_DMAADR) (x=4)

Offset address: 0x4C

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

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

Bit	Name	Access	Description	Reset value
[31:0]	DMAB[31:0]	RW	DMA address in continuous mode.	0

Chapter 15 Universal Synchronous Asynchronous Receiver Transmitter (USART)

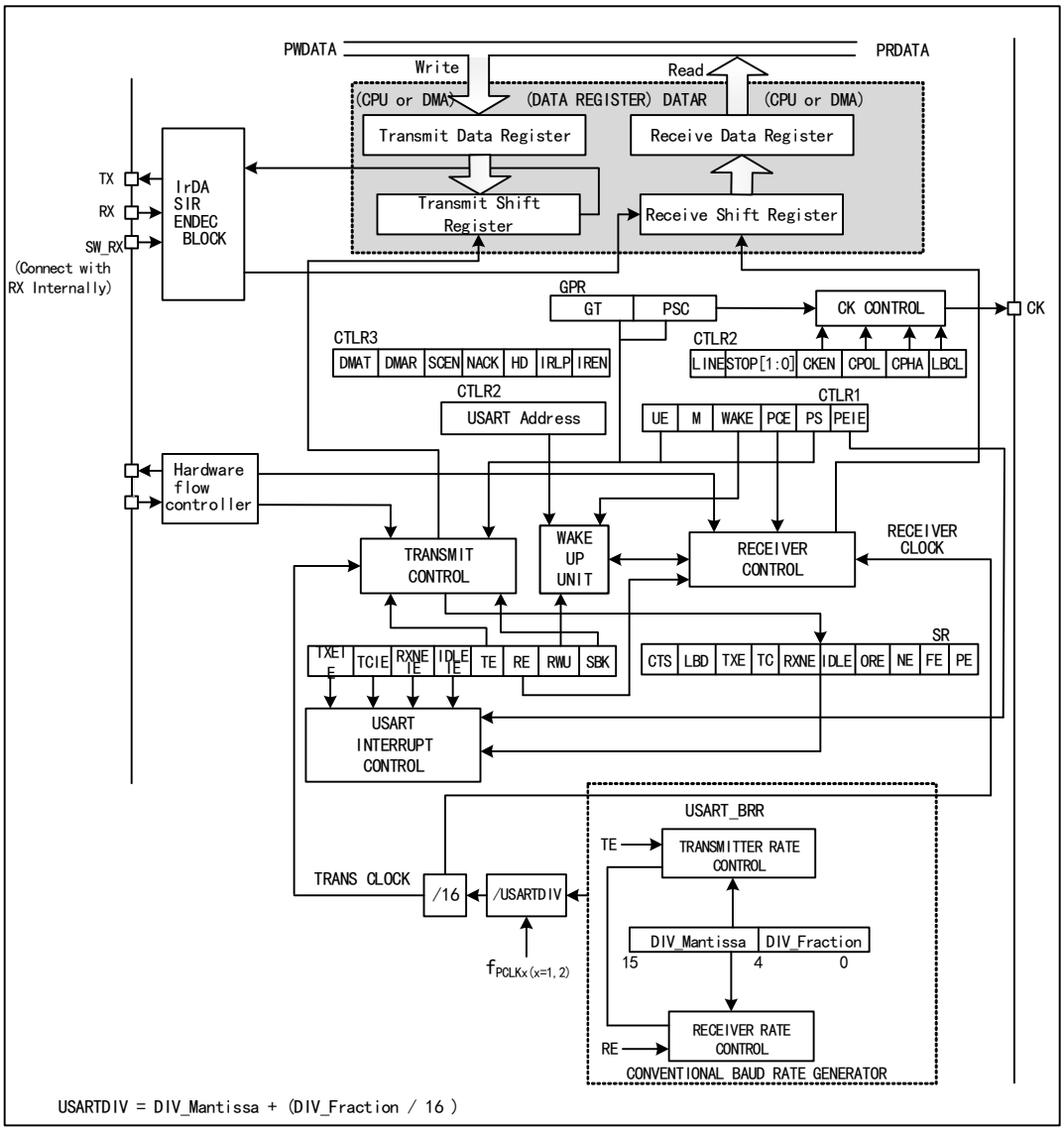
The module contains 4 universal synchronous asynchronous transceivers (USART1/2/3/4).

15.1 Main Features

- Full-duplex or half-duplex synchronous or asynchronous communication
- NRZ data format
- Fractional baud rate generator, highest 6Mbps
- Programmable data length
- Configurable stop bit
- Support LIN, IrDA encoders, smart cards
- Support DMA
- Multiple interrupt sources

15.2 Overview

Figure 15-1 Structure block diagram of universal synchronous/asynchronous receiver-transmitter



When TE (transmission enable bit) is set, the data in the transmitter shift register will be outputted on the TX pin, and the clock will be outputted on the CK pin. During transmission, the lowest significant bit is the first to be shifted out. Each data frame starts with a low-level start bit, and then the transmitter sends 8-bit data or 9-bit data according to the setting of the M (word length) bit, and finally a configurable number of stop bits. If there is a parity check bit, the last bit of the data word is the check bit. After TE is set, an idle frame is sent. The idle frame is 10-bit or 11-bit high level, including the stop bit. The break frame is a 10-bit or 11-bit low level, followed by a stop bit.

15.3 Baud Rate Generator

The baud rate of the transceiver = FCLK/(16*USARTDIV); FCLK is the clock of HBx, i.e., HCLK. The value of USARTDIV is determined according to the 2 domains: DIV_M and DIV_F in USART_BRR. The specific calculation formula is:

$$USARTDIV = DIV\_M+(DIV\_F/16)$$

It should be noted that the baud rate generated by the baud rate generator may not always generate just the baud rate

required by the user, which may be biased. In addition to taking the value as close as possible, the method to reduce the deviation can also be to increase the HB clock. For example, when setting the baud rate to 9600bps, the USARTDIV value is set to 78.125. At 12MHz, this yields a baud rate of 9600bps. However, if you require a baud rate of 115200bps, the calculated USARTDIV value is 6.51. but the closest value that can be entered into USART_BRR is 6.5. The resulting baud rate is 115384bps, resulting in an error of 0.16%.

When the serial port waveform sent by the transmitter is transmitted to the receiver, there is a certain error in the baud rate between the receiver and the sender. The error mainly comes from 3 aspects: the actual baud rate of the receiver and the sender are inconsistent; the clocks of the receiver and the sender have errors; the waveform changes in the circuit. The receiver of the peripheral module has a certain tolerance for receiving. When the sum of the total deviations generated in the above 3 aspects is less than the tolerance limit of the module, the total deviation will not affect the receiving and sending. The tolerance limit of the module is affected by the use of fractional baud rate and M bit (data field word length) or not. The use of fractional baud rate and the use of 9-bit data field length will reduce the tolerance limit, but it shall not be less than 3%.

15.4 Synchronous Mode

The synchronous mode enables the system to output clock signals when the USART module is used. When the synchronous mode is enabled to send data externally, the CK pin will output clock externally at the same time.

To enable synchronous mode, set CLKEN bit in the control register 2 (R16_USARTx_CTLR2), but you need to switch off the LIN mode, smart card mode, infrared mode and half-duplex mode at the same time, i.e., to ensure that the SCEN, HDSEL and IREN bits are in the reset status. These 3 bits are in the control register3 (R16_USARTx_CTLR3).

The main point of the synchronous mode is the output control of the clock. Attention shall be paid to the following:

The synchronous mode of the USART module only works in the master mode, i.e., the CK pin only outputs the clock and does not receive input;

The clock signal is outputted only when TX pin outputs data;

The LBCL bit determines whether the clock is outputted when the last data bit is sent. The CPOL bit determines the polarity of the clock, and the CPHA determines the phase position of the clock. These 3 bits are in the control register 2 (R16_USARTx_CTLR2). These 3 bits need to be set when TE and RE are not enabled. The specific difference is shown in Figure 15-2.

In the synchronous mode, the receiver will only sample when outputting the clock, and the slave needs to maintain a certain signal setup time and hold time, specifically as shown in Figure 15-3.

Figure 15-2 Example of USART clock timing (M=0)

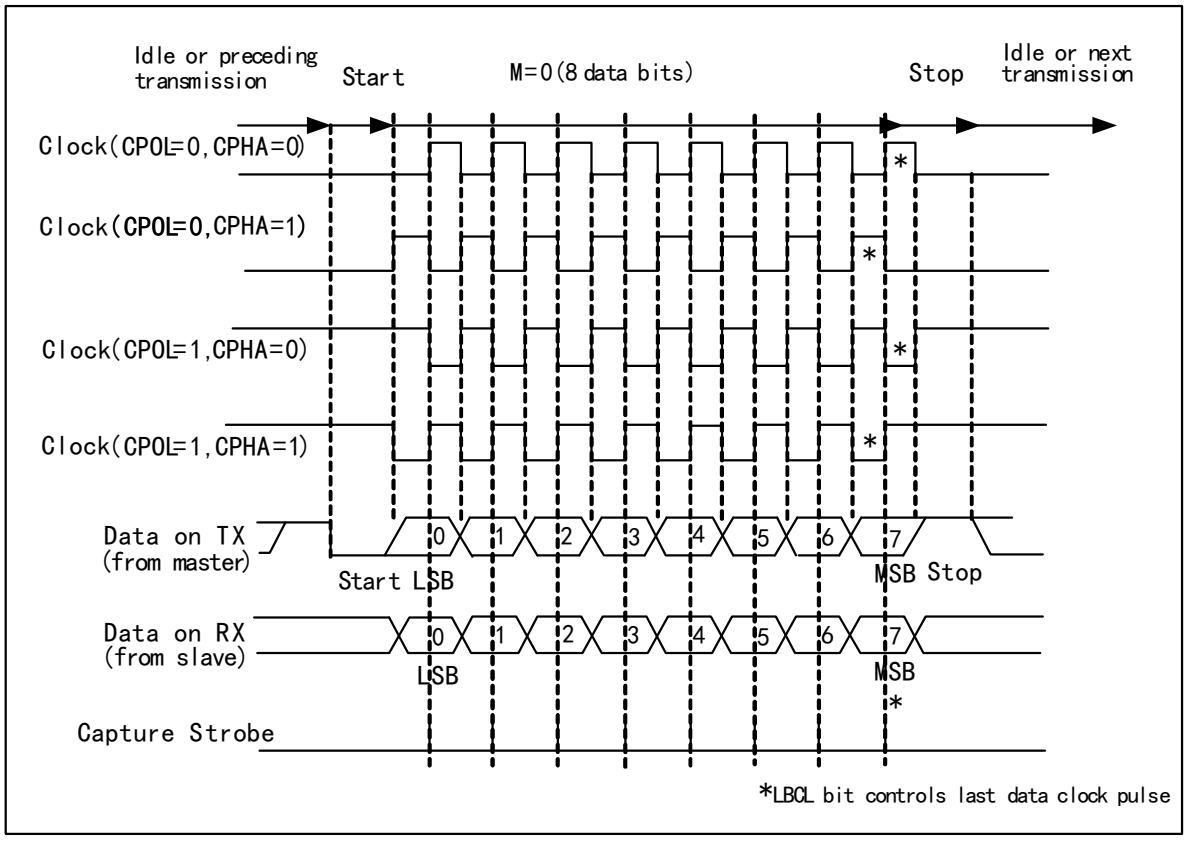
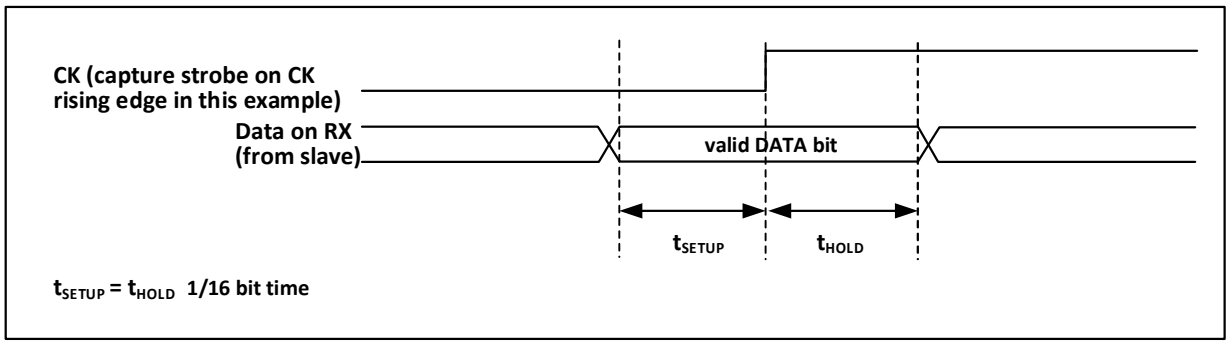


Figure 15-3 Data sample holding time



15.5 1-wire Half-duplex Mode

The half-duplex mode supports the use of a single pin (only TX pin) to receive and transmit, and the TX pin and RX pin are connected inside the chip.

To enable half-duplex mode, set HDSEL bit in the control register 3 (R16_USARTx_CTLR3), but you need to disable LIN mode, smartcard mode, infrared mode and synchronous mode at the same time, i.e., to ensure that the SCEN, CLKEN and IREN bits are in the reset status. These 3 bits are in the control register 2 and the control register 3 (R16_USARTx_CTLR2 and R16_USARTx_CTLR3).

After setting to half-duplex mode, it is needed to set the TX IO port to open-drain output high mode. When TE bit is set, the data will be sent out as long as the data is written to the data register. Special attention shall be paid to

the fact that bus conflicts may occur when multiple devices use a single bus to transmit and receive in half-duplex mode. This requires users to avoid it by software.

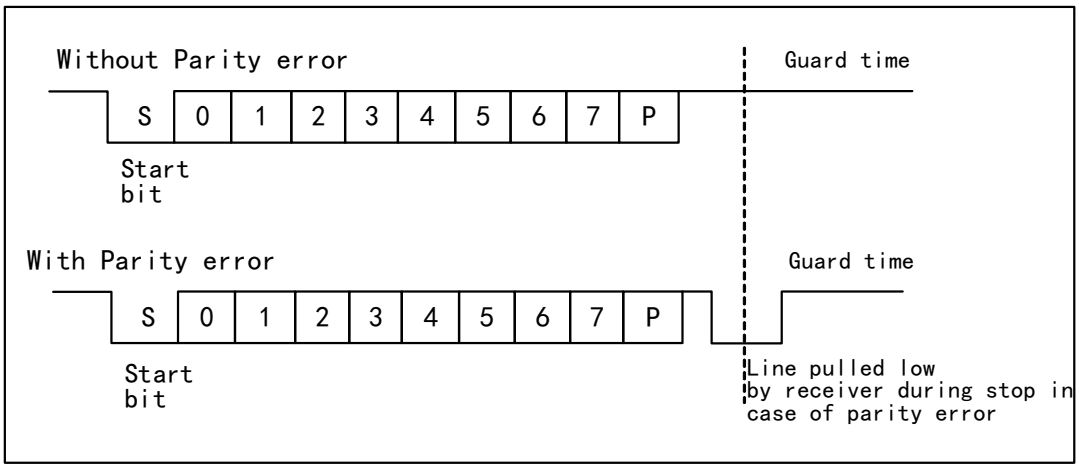
15.6 Smart Card

The smartcard mode supports ISO7816-3 protocol to access the smart card controller.

To enable smartcard mode, set the SCEN bit in the control register 3 (R16_USARTx_CTLR3), but it is needed to disable LIN mode, half-duplex mode and infrared mode at the same time, i.e., to ensure that the LINEN, HDSEL and IREN bits are in the reset status, but CLKEN can be switched on to output the clock. These 3 bits are in the control register 2 and the control register 3 (R16_USARTx_CTLR2 and R16_USARTx_CTLR3).

In order to support smartcard mode, USART shall be set to 8 data bits plus 1 check bit. It is recommended that the stop bit be configured to 1.5 bits for both sending and receiving. The smart card mode is a 1-wire half-duplex protocol, which uses TX line as the data communication and shall be configured as open drain output plus pull-up. When the receiver receives a frame of data and detects a parity check error, it will send a NACK signal at the stop bit, i.e., actively reducing a cycle of TX during the stop bit. After the sender detects the NACK signal, a frame error will be generated, and the application can resend accordingly. Figure 15-4 shows the waveforms on the TX pin under correct conditions and in the event of parity check errors. The TC flag (transmission completion flag) of the USART can delay the generation of GT (protection time) clocks, and the receiver will not recognize the NACK signal set by itself as the start bit.

Figure 15-4 (No) parity check error



In smartcard mode, the output waveform after the CK pin is enabled has nothing to do with the communication. It only provides the clock for the smart card. Its value is the PB clock and then the 5-bit settable clock frequency division (the frequency division value is double of PSC, and the highest is frequency division 62).

15.7 IrDA

USART module supports control IrDA infrared transceiver for physical layer communication. To use IrDA, the LINEN, STOP, CLKEN, SCEN and HDSEL bits must be cleared. NRZ (non-return-to-zero) coding is used between the USART module and the SIR physical layer (infrared transceiver), and the maximum support rate is 115200bps.

IrDA is a half-duplex protocol. If UASRT sends data to the SIR physical layer, the IrDA decoder ignores the newly

sent infrared signal. If the USART receives data from SIR, then SIR does not accept USART signal. The level logic sent by USART to SIR and SIR to USART is different. In SIR receive logic, '1' represents high level and '0' represents low level. However, in the SIR transmit logic, '0' represents high level and '1' represents low level.

15.8 DMA

The USART module supports DMA, and can use DMA to implement fast continuous reception and transmission. When DMA is enabled and the TXE bit is set, DMA writes data to the transmit buffer from the set memory space. When DMA is used for reception, DMA transfers the data in the receive buffer to a specific memory space each time the RXNE bit is set.

15.9 Interrupt

The USART module supports multiple interrupt sources, including transmit data register empty (TXE), CTS, transmission complete (TC), received data ready (RXNE), data overrun error (ORE), idle line (IDLE), parity check error (PE), break flag (LBD), noise (NE), multi-buffer communication overrun (ORT) and framing error (FE).

Table 15-1 Interrupts and the corresponding enable bits

Interrupt source	Enable bit
Transmit data register empty (TXE)	TXEIE
Clear to send (CTS)	CTSIE
Transmission complete (TC)	TCIE
Received data ready (RXNE)	RXNEIE
Transmit data register empty (TXE)	
Idle line (IDLE)	IDLEIE
Parity error (PE)	PEIE
Break flag (LBD)	LBDIE
Noise (NE)	EIE
Overrun error in multi-buffer communication (ORT)	
Framing error (FE) in multi-buffer communication	

15.10 Register Description

Table 15-2 USART1-related registers

Name	Access address	Description	Reset value
R32_USART1_STATR	0x40013800	UASRT1 status register	0x000000C0
R32_USART1_DATAR	0x40013804	UASRT1 data register	0x000000XX
R32_USART1_BRR	0x40013808	UASRT1 baud rate register	0x00000000
R32_USART1_CTLR1	0x4001380C	UASRT1 control register 1	0x00000000
R32_USART1_CTLR2	0x40013810	UASRT1 control register 2	0x00000000
R32_USART1_CTLR3	0x40013814	UASRT1 control register 3	0x00000000
R32_USART1_GPR	0x40013818	UASRT1 guard time and prescaler register	0x00000000

R32_USART1_CTLR4	0x4001381C	UASRT1 control register 4	0x00000000

Table 15-3 USART2-related registers

Name	Access address	Description	Reset value
R32_USART2_STATR	0x40004400	UASRT2 status register	0x000000C0
R32_USART2_DATAR	0x40004404	UASRT2 data register	0x000000XX
R32_USART2_BRR	0x40004408	UASRT2 baud rate register	0x00000000
R32_USART2_CTLR1	0x4000440C	UASRT2 control register 1	0x00000000
R32_USART2_CTLR2	0x40004410	UASRT2 control register 2	0x00000000
R32_USART2_CTLR3	0x40004414	UASRT2 control register 3	0x00000000
R32_USART2_GPR	0x40004418	UASRT2 guard time and prescaler register	0x00000000
R32_USART2_CTLR4	0x4000441C	UASRT2 control register 4	0x00000000

Table 15-4 USART3-related registers

Name	Access address	Description	Reset value
R32_USART3_STATR	0x40004800	UASRT3 status register	0x000000C0
R32_USART3_DATAR	0x40004804	UASRT3 data register	0x000000XX
R32_USART3_BRR	0x40004808	UASRT3 baud rate register	0x00000000
R32_USART3_CTLR1	0x4000480C	UASRT3 control register 1	0x00000000
R32_USART3_CTLR2	0x40004810	UASRT3 control register 2	0x00000000
R32_USART3_CTLR3	0x40004814	UASRT3 control register 3	0x00000000
R32_USART3_GPR	0x40004818	UASRT3 guard time and prescaler register	0x00000000
R32_USART3_CTLR4	0x4000481C	UASRT3 control register 4	0x00000000

Table 15-5 USART4-related registers

Name	Access address	Description	Reset value
R32_USART4_STATR	0x40004C00	UASRT4 status register	0x000000C0
R32_USART4_DATAR	0x40004C04	UASRT4 data register	0x000000XX
R32_USART4_BRR	0x40004C08	UASRT4 baud rate register	0x00000000
R32_USART4_CTLR1	0x40004C0C	UASRT4 control register 1	0x00000000
R32_USART4_CTLR2	0x40004C10	UASRT4 control register 2	0x00000000
R32_USART4_CTLR3	0x40004C14	UASRT4 control register 3	0x00000000
R32_USART4_GPR	0x40004C18	UASRT4 guard time and prescaler register	0x00000000
R32_USART4_CTLR4	0x40004C1C	UASRT4 control register 4	0x00000000

15.10.1 USART Status Register (USARTx_STATR) (x=1/2/3/4)

Offset address: 0x00

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

LPWKUP_ACT_FLAG	Reserved	MS_ERR	RX_BUSY	CTS	LBD	TXE	TC	RXNE	IDLE	ORE	NE	FE	PE

Bit	Name	Access	Description	Reset value
[31:16]	Reserved	RO	Reserved	0
15	LPWKUP_ACT_FLAG	RW1Z	Low-power wake-up start indication bit, clear by writing 1.	0
[14:12]	Reserved	RO	Reserved	0
11	MS_ERR	RO	MARK or SPACE check error flag. In receive mode, if a MARK or SPACE check error occurs, the hardware sets this bit. Reading this bit and then reading the data register resets this bit. If MS_ERRIE was set before, then setting this bit will generate a corresponding interrupt.	0
10	RX_BUSY	RO	Receive status indication bit: 1: When working in the receiving state, this bit is set; 0: In the non-receiving state.	0
9	CTS	RW0	CTS status change flag. If the CTSE bit is set, when the nCTS output state changes, the bit will be set high by the hardware. Zero is cleared by the software. If the CTSIE bit has been set, an interrupt occurs. 1: There is a change in the nCTS state line. 0: There is no change on the nCTS state line.	0
8	LBD	RW0	LIN Break detection flag. When LIN Break is detected, this bit will be set by hardware. It is cleared by the software. If the LBDIE bit has been set, an interrupt will be generated. 1: LIN Break detected; 0: No LIN Break detected.	0
7	TXE	RO	Transmission data register empty flag. When data in TDR register is transferred to shift register by hardware, this bit will be set by hardware. If TXEIE bit has been set, an interrupt will be generated, the data register will be written and this bit will be reset. 1: Data is transferred to the shift register; 0: Data is not transferred to the shift register.	1
6	TC	RW0	Transmission complete flag. When a frame containing data is sent and TXE bit is set, the hardware will set this bit. If TCIE is set, a corresponding interrupt will be generated. The software will read this bit and then write the data register to clear this bit. You can also directly write 0	1

			to clear this bit. 1: Transmission completed; 0: Transmission not completed.	
5	RXNE	RW0	Read data register not empty flag. When the data in the shift register is transferred to the data register, this bit will be set by the hardware. If the RXNEIE bit has been set, the corresponding interrupt will be generated. This bit can be cleared by the write operation of the data register. This bit can be also cleared by directly writing 0. 1: The data is received and can be read; 0: The data is not received.	0
4	IDLE	RO	Idle line flag. When an idle line is detected, the bit will be set by hardware. If IDLEIE bit has been set, the corresponding interrupt will be generated. This bit can be cleared by reading the status register and then reading the data register. 1: The bus is idle now; 0: Idle bus is not detected. *Note: This bit will not be set again until RXNE is set.*	0
3	ORE	RO	Overrun error flag. When the receiving shift register has data that needs to be transferred to the data register, but this bit will be set when there is still data that has not been read in the receiving field of the data register. If the RXNEIE bit is set, the corresponding interrupt will be generated. 1: The overrun error has occurred; 0: No overrun error has occurred. *Note: When an overrun error occurs, the value of the data register will not be lost, but the value of the shift register will be overwritten. If the EIE bit is set, the ORE flag bit will generate an interrupt in the multi-buffer communication mode.*	0
2	NE	RO	Noise error flag. When the noise error flag is detected, it will be set by hardware. This bit can be reset by reading the status register and then reading the data register. 1: The noise is detected; 0: No noise is detected. *Note: This bit will not generate the interrupt. If the EIE bit has been set, the FE flag bit will generate an interrupt in the multi-buffer communication mode.*	0
1	FE	RO	Frame error flag. When a synchronization error, excessive noise or disconnection is detected, this bit will be set by hardware. This bit can be reset by	0

			reading the bit and then reading the data register. 1: A frame error is detected; 0: No frame error is detected. *Note: This bit will not generate interrupt. If the EIE bit has been set, the FE flag bit will generate an interrupt in the multi-buffer communication mode.*	
0	PE	RO	Parity error flag. In the receiving mode, if a parity error occurs, this bit can be set by hardware. This bit can be reset by reading the bit and then reading the data register. Before this bit is cleared, the software must wait for the RXNE flag bit to be set. If PEIE bit has been set before, then the corresponding interrupt will be generated when this bit is set. 1: Parity check error occurs; 0: No parity check error occurs.	0

15.10.2 USART Data Register (USARTx_DATAR) (x=1/2/3/4)

Offset address: 0x04

31	30	29	28	27	26	25	24	23	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]								

Bit	Name	Access	Description	Reset value
[31:9]	Reserved	RO	Reserved	0
[8:0]	DR[8:0]	RW	Data register. This register is actually composed of 2 registers: receive data register (RDR) and transmit data register (TDR). The start of the read and write operations of DR is to read the receive data register (RDR) and write to the transmit data register (TDR).	X

15.10.3 USART Baud Rate Register (USARTx_BRR) (x=1/2/3/4)

Offset address: 0x08

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved												DIV_Mantissa[15:12]			

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

Bit	Name	Access	Description	Reset value
[31:20]	Reserved	RO	Reserved	0
[19:4]	DIV_Mantissa [11:0]	RW	These 16 bits define the integer portion of the divider division factor.	0

[3:0]	DIV_Fraction[3:0]	RW	These 4 bits define the decimal part of the divider division factor.	0

15.10.4 USART Control Register 1 (USARTx_CTLR1) (x=1/2/3/4)

Offset address: 0x0C

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
M_EXT[1:0]		UE	M	WAKE	PCE	PS	PEIE	TXEIE	TCIE	RXNEIE	IDLEIE	TE	RE	RWU	SBK

Bit	Name	Access	Description	Reset value
[31:16]	Reserved	RO	Reserved	0
[15:14]	M_EXT[1:0]	RW	Word length extension bits: 00: Invalid, M bit determines the data length. 01: 7 data bits; 10: 6 data bits; 11: 5 data bits.	0
13	UE	RW	USART enable: 1: Enable; 0: Disable. When this bit is cleared, the frequency divider and output of USART both stop working after the current byte transmission is completed.	0
12	M	RW	Word length. 1: 9 data bits; 0: 8 data bits.	0
11	WAKE	RW	Wake-up. This bit decides the method to wake up USART: 1: Address flag; 0: Idle line.	0
10	PCE	RW	Parity control enable: 1: Check bit enabled; 0: Check bit disabled. For the receiver, the parity of the data is performed; for the transmitter, the check bit is inserted. Once this bit is set, the parity control enable takes effect only after the current byte transmission is completed.	0
9	PS	RW	Parity selection: 1: Odd parity; 0: Even parity. After this bit is set, the parity control enable takes effect only after the current byte transmission is completed.	0

8	PEIE	RW	Parity check interrupt enable bit. 1: Allow parity error interrupt; 0: Disable parity error interrupt.	0
7	TXEIE	RW	Transmit buffer empty interrupt enable: 1: Allow the send buffer empty interrupt to be generated; 0: Disable the send buffer empty interrupt to be generated.	0
6	TCIE	RW	Transmission completion interrupt enable: 1: Allow generation of transmission completion interrupt; 0: Disable generation of transmission completion interrupt.	0
5	RXNEIE	RW	Receive buffer non-empty interrupt enable: 1: Allow the reception buffer non-empty interrupt to be generated; 0: Disable the reception buffer non-empty interrupt.	0
4	IDLEIE	RW	Bus idle interrupt enable: 1: Allow bus idle interrupt to be generated; 0: Disable bus idle interrupt to be generated.	0
3	TE	RW	Transmit enable. 1: Enable the transmitter; 0: Disable the transmitter.	0
2	RE	RW	Receive enable. 1: Enable the receiver and the receiver starts to detect the start bit on the RX pin 0: Disable the receiver.	0
1	RWU	RW	Receive wake-up. This bit decides whether the USART is in mute mode: 1: The receiver is in mute mode; 0: The receiver is in active mode. *Note 1: Before the RWU bit is set, USART needs to receive a data byte firstly. Otherwise, it cannot be woken up by the idle bus in mute mode;* *Note 2: When configured to wake up from address flag, the RWU bit cannot be modified by software when RXNE is set.*	0
0	SBK	RW	Send frame break character control. This bit is set to transmit a frame break character. For the stop bit of break frame, the bit is set by hardware. 1: Break character transmitted; 0: No break character transmitted.	0

15.10.5 USART Control Register 2 (USARTx_CTLR2) (x=1/2/3/4)

Offset address: 0x10

31	30	29	28	27	26	25	24	23	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	LINEN	STOP		CLKEN	CPOL	CPHA	LBCL	Reserved	LBDIE	LBDL	Reserved	ADD[3:0]			

Bit	Name	Access	Description	Reset value
[31:15]	Reserved	RO	Reserved	0
14	LINEN	RW	LIN mode enable. When this bit is set, the LIN mode is enabled. In LIN mode, you can use the SBK bit to send the LIN synchronization disconnection symbol and detect the LIN synchronization disconnection symbol.	0
[13:12]	STOP	RW	Stop bit setting. These bits are used to set the stop bits. 00: 1 stop bit; 01: 0.5 stop bit; 10: 2 stop bits; 11: 1.5 stop bit.	00b
11	CLKEN	RW	Clock enable. This bit is used to enable CK pin. 1: Enable; 0: Disable.	0
10	CPOL	RW	Clock polarity. In synchronous mode, this bit can be used to select the polarity of the clock output on the CK pin, and work with CPHA to generate the required clock/data sampling relationship. 1: High level is maintained on the CK pin when the bus is idle; 0: Low level is maintained on the CK pin when the bus is idle. *Note: This bit cannot be modified after enabling transmission.*	0
9	CPHA	RW	Clock phase position setting. In the synchronization mode, you can use this bit to select the phase position of the clock output on the CK pin, and work with CPOL bit to generate the required clock/data sampling relationship. 1: Data capture is performed on the second edge of the clock; 0: Data capture is performed on the first edge of the clock. *Note: This bit cannot be modified after enabling transmission.*	0
8	LBCL	RW	Last bit clock pulse control. In synchronous mode, it is used to control whether to output the clock pulse corresponding to the last data byte	0

			sent on the CK pin; 1: The clock pulse of the last bit of data is not output from CK; 0: The clock pulse of the last bit of data is output from CK. *Note: This bit cannot be modified after enabling transmission.*	
7	Reserved	RW	Reserved.	0
6	LBDIE	RW	LIN Break detection interrupt enable. This bit can enable the interrupt caused by LBD;	0
5	LBDL	RW	LIN Break detection length, used to select 11-bit or 10-bit break character detection. 1: 11-bit break detection; 0: 10-bit break detection.	0
4	Reserved	RW	Reserved.	0
[3:0]	ADD[3:0]	RW	Address of the USART node, used to set the USART node address of the device. When the data is used during mute mode in multi-processor communication, the address flag is used to wake up a certain USART device.	0

15.10.6 USART Control Register 3 (USARTx_CTLR3) (x=1/2/3/4)

Offset address: 0x14

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LPWKUP_DLY_CFG			LPWKUP_CK_SRC	LPWKUP_EN	CTSIE	CTSE	RTSE	DMAT	DMAR	SCEN	NACK	HDSEL	IRLP	IREN	EIE

Bit	Name	Access	Description	Reset value
[31:16]	Reserved	RO	Reserved	0
[15:13]	LPWKUP_DLY_CFG	RW	Low-power wake-up receive delay configuration: 000: No delay; 001: Delay of 1 low-power wake-up clock source cycle; … 111: Delay of 7 low-power wake-up clock source cycles.	0
12	LPWKUP_CK_SRC	RW	Low power wake-up clock source selection: 1: LSE; 0: LSI.	0
11	LPWKUP_EN	RW	Low-power wake-up enable: 1: Enable low-power wake-up; 0: Disable low-power wake-up.	0

10	CTSIE	RW	CTSIE interrupt enable: 1: An interrupt will be generated when CTS is set; 0: No interrupt will be generated when CTS is set.	0
9	CTSE	RW	CTS enable: 1: Enable CTS flow control; 0: Disable CTS flow control.	0
8	RTSE	RW	RTS enable: 1: Enable RTS flow control; 0: Disable RTS flow control.	0
7	DMAT	RW	DMA transmission enable: 1: Use DMA when transmitting; 0: Do not use DMA when transmitting.	0
6	DMAR	RW	DMA reception enable: 1: Use DMA when receiving; 0: Do not use DMA when receiving.	0
5	SCEN	RW	Smart card mode enable: 1: Enable smart card mode; 0: Disable smart card mode.	0
4	NACK	RW	Smart card NACK enable: 1: When a verification error occurs, NACK is sent; 0: When a verification error occurs, NACK is not sent.	0
3	HDSEL	RW	Half-duplex mode selection: 1: Enable half-duplex mode; 0: Disable half-duplex mode.	0
2	IRLP	RW	Infrared low-power selection: 1: When infrared is selected, low-power mode is enabled; 0: When infrared is selected, low-power mode is disabled.	0
1	IREN	RW	Infrared enable: 1: Enable infrared mode; 0: Disable infrared mode.	0
0	EIE	RW	Error interrupt bit enable: 1: Enable error interrupt; 0: Disable error interrupt. After setting this bit, if FE, ORE or NE is set, an interrupt will be generated under the premise that DMAR is set.	0

15.10.7 USART Guard Time and Prescaler Register (USARTx_GPR) (x=1/2/3/4)

Offset address: 0x18

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

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

Bit	Name	Access	Description	Reset value
[31:16]	Reserved	RO	Reserved	0
[15:8]	GT[7:0]	RW	Guard time domain. These bits specify the guard time in unit of baud rate clock. In smartcard mode, the transmission complete flag is set after the guard time has passed.	0
[7:0]	PSC[7:0]	RW	Prescaler domain. In infrared low-power mode, the source clock is divided by this value (all 8 bits are valid), and a value of 0 means reservation; In infrared normal mode, these bits can only be set to 1; In smartcard mode, the value (the lower 5 bits are valid) is multiplied by 2 to give the division factor of the source clock frequency, to provide the clock to the smart card. A value of 0 means reservation.	0

15.10.8 USART Control Register 4 (USARTx_CTLR4) (x=1/2/3/4)

Offset address: 0x1C

31	30	29	28	27	26	25	24	23	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												CHECK_SEL [1:0]		MS_ERRIE	SELF_OD

Bit	Name	Access	Description	Reset value
[31:4]	Reserved	RO	Reserved	0
[3:2]	CHECK_SEL[1:0]	RW	Checksum Function Selection Bits: 0x: Disable MARK and SPACE checksums. 10: Enable MARK checksum. 11: Enable SPACE checksum.	0
1	MS_ERRIE	RW	SPACE or MARK Check Enable Bit: 1: Enable SPACE or MARK check error interrupts; 0: Disable SPACE or MARK check error interrupts.	0
0	SELF_OD	RW	Open-drain output enable: 1: Open-drain output; 0: Push-pull output.	0

Chapter 16 Inter-integrated Circuit (I2C) Interface

The internal integrated circuit bus (I2C) is widely used in the communication between microcontrollers, sensors and other off-chip modules. It supports multi-master and multi-slave mode, and can communicate at both 100kHz (standard) and 400kHz (fast) speeds using only 2 wires (SDA and SCL).

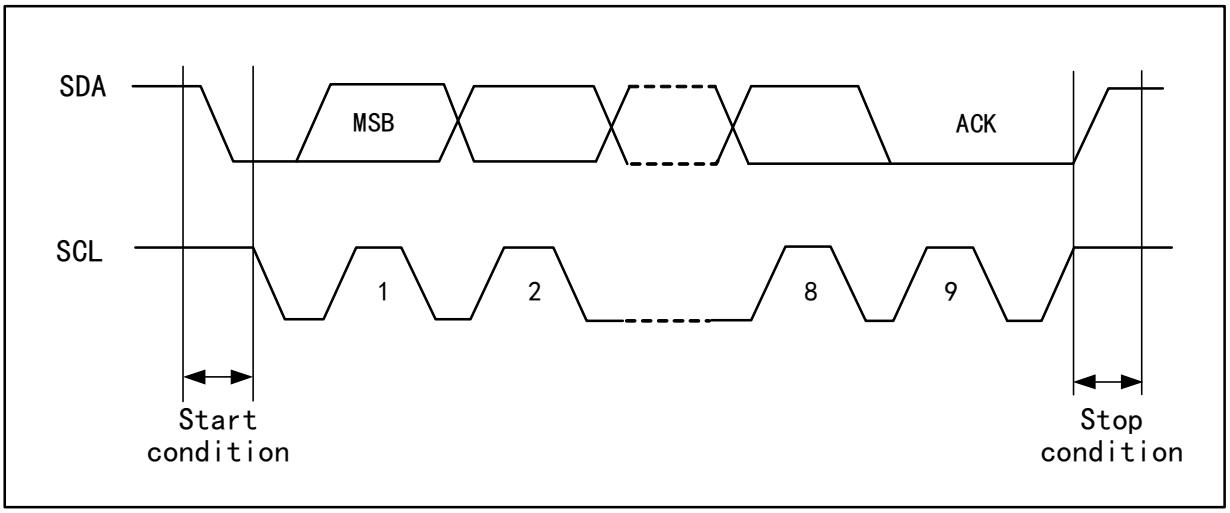
16.1 Main Features

- Master mode and slave mode
- 7-bit or 10-bit address
- Slave device supports dual 7-bit address.
- Two speed modes: 100kHz and 400kHz
- Multiple status modes, multiple error flags
- Optional clock stretching
- 2 interrupt vectors
- DMA capability
- Support PEC

16.2 Overview

I2C is a half-duplex bus, which can only run in one of the following 4 modes: master device sending mode, master device receiving mode, slave device sending mode and slave device receiving mode. The I2C module works in slave mode by default. After generating the starting condition, it will automatically switch to the master mode. When the arbitration is lost or a stop signal is generated, it will switch to the slave mode. The I2C module supports multi-host functions. When working in main mode, the I2C module will actively send out data and addresses. Data and addresses are transmitted in 8-bit units, with the high bit in front and the low bit in the back. After the start event is 1-byte (7-bit address mode) or 2-byte (10-bit address mode) address. Every time the host sends 8-bit data or address, the slave needs to reply a reply ACK, that is, pull down the SDA bus, as shown in figure 16-1.

Figure 16-1 I2C timing diagram

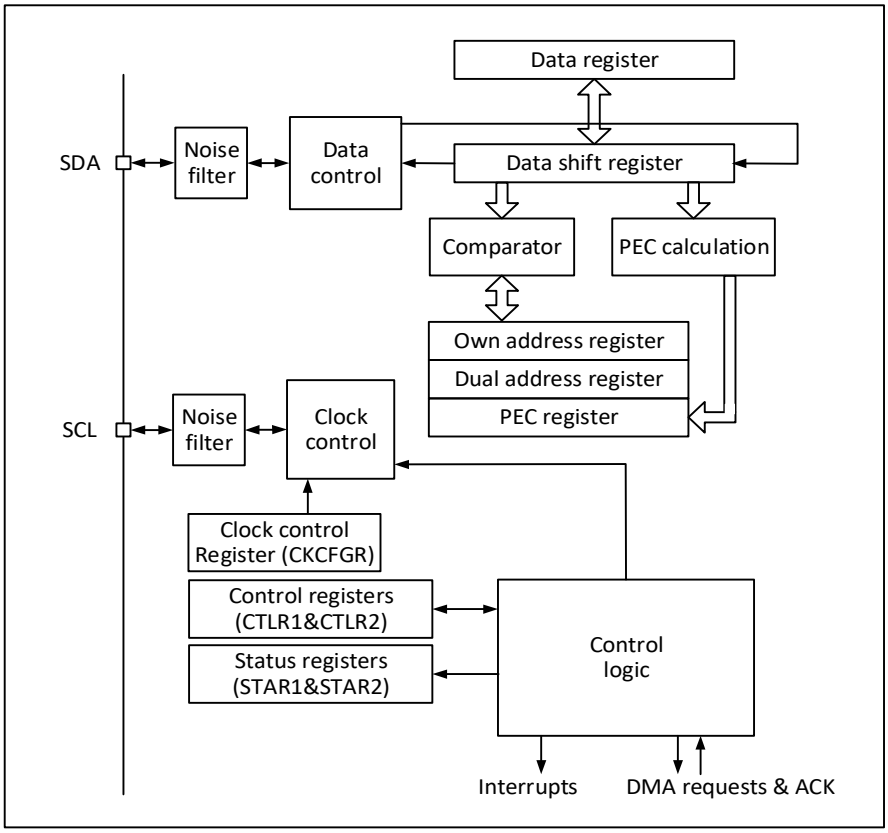


For normal use, the correct clock must be input to I2C. In the standard mode, the minimum input clock is 2MHz,

while the minimum input clock is 4MHz in the fast mode.

Figure 16-2 shows the block diagram of I2C.

Figure 16-2 I2C block diagram



16.3 Master Mode

In master mode, the I2C module leads the data transmission and outputs the clock signal. The data transfer starts with a Start event and ends with a Stop event. The following is the required operations in master mode:

Set the correct clock in the control register 2 (R16_I2Cx_CTLR2) and the clock control register (R16_I2Cx_CKCFGR).

Set the PE bit in R16_I2Cx_CTLR1 to start the peripheral.

Set the START bit in the control register (R16_I2Cx_CTLR1) to generate a start event. After the START bit is set, the I2C module automatically switches to master mode, the MSL bit is set, and a Start event is generated. After the start event is generated, the SB bit is set. If the ITEVTEN bit (in R16_I2Cx_CTLR2) is set, an interrupt is generated. In this case, it is needed to read the R16_I2Cx_STAR1 register. After the slave address is written to the data register, the SB bit is automatically cleared.

If the 10-bit address mode is enabled, then write the data register to send the header sequence (the header sequence is 11110xx0b, of which the xx bits are the highest 2 bits of the 10-bit address).

After the header sequence is transmitted, the ADD10 bit in the status register is set. If the ITEVTEN bit is already set, an interrupt will be generated. At this time, read the R16_I2Cx_STAR1 register and write the second address byte to the data register. Then, clear the ADD10 bit.

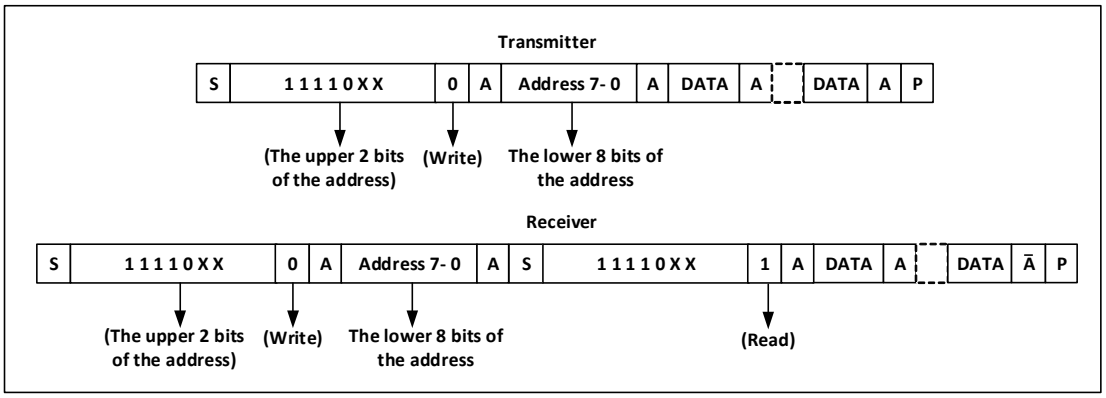
Then, write the data register to send the second address byte. After sending the second address byte, the ADDR bit in the status register is set. If the ITEVTEN bit has been set, an interrupt will be generated. Read the R16_I2Cx_STAR2 register at this time and then read R16_I2Cx_STAR1 register again to clear the ADDR bit;

If the 7-bit address mode is enabled, the write data register transmit the address byte. After the address byte is sent, the ADDR bit in the status register is set. If the ITEVTEN bit has been set, an interrupt will be generated. Read the R16_I2Cx_STAR1 register and then read the R16_I2Cx_STAR2 register again to clear the ADDR bit.

In the 7-bit address mode, the first byte sent is the address byte, the first 7 bits represent the address of the target slave device, the 8th bit determines the direction of the subsequent message, and '0' means the master device writes data to the slave device, '1' means that the master device reads information from the slave device.

In the 10-bit address mode, as shown in Figure 16-3, the first byte is 11110xx0, xx are the highest 2 bits of the 10-bit address, and the second byte is the lower 8 bits of the 10-bit address in the address transmission phase. If you subsequently enter the master transmitter mode, send data continuously. If the device enters the master receiver mode subsequently, a start event needs to be re-sent, and a byte of 11110xx1 will be sent together. Then, enter the master receiver mode.

Figure 16-3 Master receive/transmits data in 10-bit address mode



When transmitting mode, the shift register inside the main device transmits data from the data register to the SDA line. When the main device receives the ACK, the TxE of the status register 1 (R16_I2Cx_STAR1) is set, and an interrupt occurs if the ITEVTEN and ITBUFEN are set. Writing data to the data register clears the TxE bit. If the TxE bit is set and no new data is written to the data register before the last data is sent, then the BTF bit will be set, the SCL will remain low until it is cleared, and after reading the R16_I2Cx_STAR1, writing data to the data register will clear the BTF bit.

In the receiving mode, the I2C module receives data from the SDA line and writes it into the data register through the shift register. After each byte, if the ACK bit is set, the I2C module will issue a low level of reply, while the RxNe bit will be set, and if ITEVTEN and ITBUFEN are set, there will be an interruption. If the RxNE is set and the original data is not read before the new data is received, the BTF bit will be set, the SCL will remain low before clearing the BTF, and reading the R16_I2Cx_STAR1 and then reading the data register will clear the BTF bit.

When the master device ends sending data, it will actively send an end event, that is, setting the STOP bit. In the receive mode, the master device needs to NAK at the answer location of the last data bit. Note that after the NAK is generated, the I2C module will switch to slave mode.

16.4 Slave Mode

In slave mode, the I2C module can recognize its own address and broadcast call address. The software can control the identification of broadcast call addresses on or off. Once the start event is detected, the I2C module compares the SDA data through the shift register with its own address (the number of bits depends on ENDUAL and ADDMODE) or the broadcast address (when ENGC is set). If there is a mismatch, it will be ignored until a new start event is generated. If it matches the header sequence, an ACK signal is generated and waits for the address of the second byte; if the address of the second byte also matches, or if the address of the whole segment matches in the case of a 7-bit address, then:

First an ACK reply is generated; the ADDR bit is set, and if the ITEVTEN bit is set, there will be a corresponding interrupt.

If you are using dual-address mode (the ENDUAL bit is set), you also need to read the DUALF bit to determine which address the host is calling.

The slave mode defaults to receive mode, and when the last bit of the received header sequence is 1, or the last bit of the 7-bit address is 1 (depending on whether the header sequence is received for the first time or a normal 7-bit address), the I2C module will enter sender mode, and the TRA bit will indicate whether it is currently in receiver or sender mode.

When sending mode, after clearing the ADDR bit, the I2C module sends bytes from the data register to the SDA line through the shift register. Upon receipt of a reply ACK, the TXE bit is set, and an interrupt occurs if ITEVTEN and ITBUFEN are set. If TxE is set but no new data is written to the data register before the end of the next data transmission, the BTF bit will be set. The SCL will remain low until the BTF is cleared, and after reading the status register 1 (R16_I2Cx_STAR1), writing data to the data register will clear the BTF bit.

In the receiving mode, after the ADDR is cleared, the I2C module stores the data on the SDA into the data register through the shift register. After each byte is received, the I2C module will set an ACK bit and parallel the RxNE bit. If ITEVTEN and ITBUFEN are set, an interrupt will be generated. If the RxNE is set and the old data is not read out before the new data is received, then the BTF will be set. The SCL stays low until the BTF bit is cleared. Reading the status register 1 (R16_I2Cx_STAR1) and reading the data in the data register clears the BTF bit.

When the I2C module detects a stop event, the STOPF bit is set, and if the ITEVFEN bit is set, an interrupt occurs. The user needs to read the status register (R16_I2Cx_STAR1) and then write the control register (such as the reset control word SWRST) to clear.

16.5 Error

16.5.1 Bus Error (BERR)

When the I2C module detects an external start or stop event during address or data transmission, a bus error occurs. When a bus error is generated, the BERR bit is set, and an interrupt occurs if ITERREN is set. In slave mode, the data is discarded and the hardware releases the bus. If it is a start signal, the hardware will consider it as a restart signal and start waiting for the address or stop signal; if it is a stop signal, it will operate according to the normal stop condition in advance. In main mode, the hardware does not release the bus and does not affect the current transmission, and it is up to the user code to decide whether to abort the transmission.

16.5.2 Acknowledge Failure (AF)

When the I2C module detects no response after a byte, a reply error occurs. When a reply error is generated: AF will be set, and an interrupt will be generated if ITERREN is set; if an AF error is encountered, if the I2C module works in slave mode, the hardware must release the bus, and if it is in master mode, the software must generate a stop event.

16.5.3 Arbitration Lost (ARLO)

When the I2C module detects the arbitration loss, an arbitration loss error occurs. When an arbitration loss error occurs: the ARLO bit is set and an interrupt is generated if ITERREN is set; the I2C module switches to slave mode and no longer responds to slave-initiated transfers against it unless a host initiates a new start event; the hardware releases the bus.

16.5.4 Overrun/Underrun Error (OVR)

- Overload error:

In slave mode, if clock extension is prohibited, the I2C module is receiving data. If one byte of data has been received, but the last received data has not been read out, an overload error will occur. When an overload error occurs, the last received byte is discarded and the sender should resend the last sent byte.

- Underload error:

In slave mode, if clock extension is prohibited, the I2C module is sending data, and if new data is not written to the data register before the next byte clock arrives, an underload error will occur. When an underload error occurs, the data in the previous data register will be sent twice. If an underload error occurs, the receiver should discard the repeatedly received data. In order not to generate underload errors, the I2C module should write data to the data register before the first rising edge of the next byte.

16.6 Clock Extension

If clock extension is prohibited, there is a possibility of overload / underload errors. But if the clock is extended when enabled:

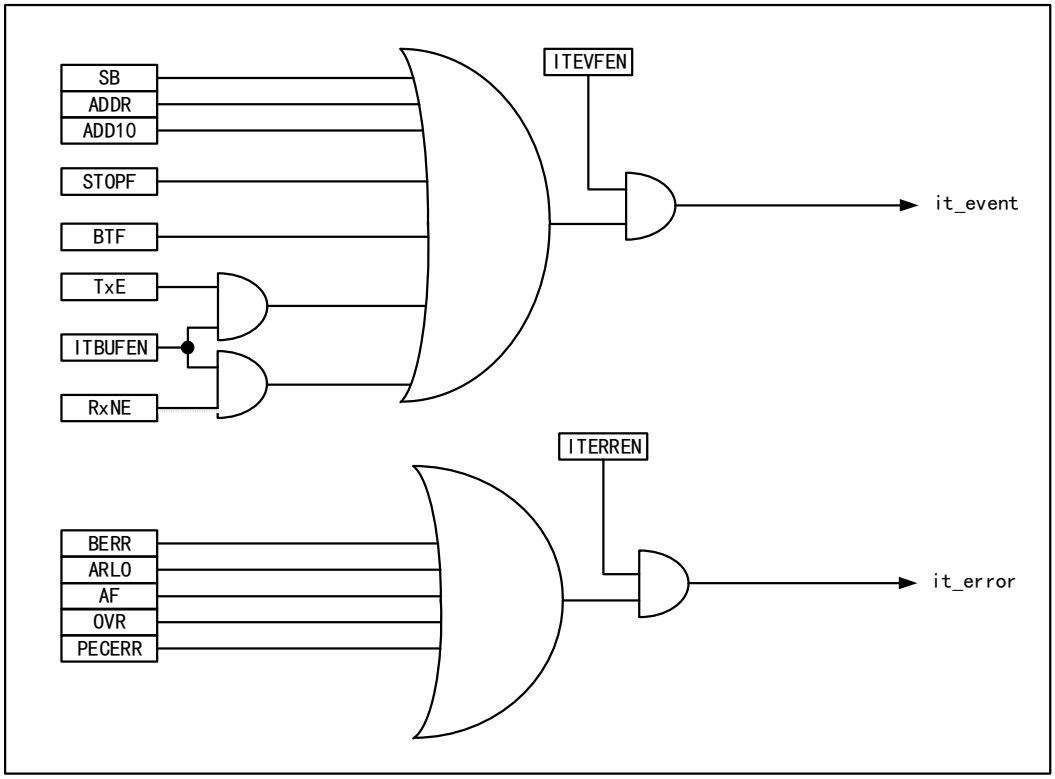
- In transmit mode, if TxE is set and BTF is set, SCL will always be low, waiting for the user to read the status register and write the data to be sent to the data register.
- In receive mode, if RxNE is set and BTF is set, the SCL will remain low after receiving the data until the user reads the status register and reads the data register.

Thus it can be seen that the extension of the enable clock can avoid overload / underload errors.

16.7 Interrupt

Each I2C module has 2 interrupt vectors: event interrupt and error interrupt. 2 types of interrupts support the interrupt sources as shown in Figure 16-4.

Figure 16-4 I2C interrupt request



16.8 DMA

DMA can be used to receive/transmit bulk data. When DMA is used, the ITBUFEN bit of the control register cannot be set.

- DMA is used for transmission

The DMA mode can be activated by setting the DMAEN bit in the CTLR2 register. As long as the TxE bit is set, the data can be loaded into the I2C data register from the set memory by DMA. The following settings are required to allocate channels for I2C.

1) Set the R16_I2Cx_DATAR register address to the DMA1_PADDRx register, and set the memory address in the DMA1_MADDRx register, so that the data will be sent from the memory to the R16_I2Cx_DATAR register after each TxE event.
2) Set the required number of transferred bytes in the DMA1_CNTRx register. After each TxE event, this value will be reduced progressively.
3) The channel priority is configured using the PL[0:1] bits in the DMA1_CFGRx register.
4) Set the DIR bit in the DMA1_CFGRx register, and it can be configured to issue an interrupt request according to application requirements when the entire transmission is half or wholly completed.
5) Activate the channel by setting the EN bit in the DMA1_CFGRx register.

When the number of data transfer bytes set in the DMA controller has been completed, the DMA controller will send an EOT/EOT_1 signal indicating the end of the transmission to the I2C interface. When the interrupt is allowed, a DMA interrupt will be generated.

- DMA is used for reception

After the DMAEN bit in the CTLR2 register is set, DMA receiver mode can be started. When DMA is used for reception, DMA transfers the data in the data register to the preset memory area. The following steps are required to allocate channels for I2C.

1) Set the I2Cx_DATAR register address to the DMA1_PADDRx register, and set the memory address in the DMA_MADDRx register, so that the data will be written into the memory from the I2Cx_DATAR register after each RxNE event.
2) Set the required number of transferred bytes in the DMA1_CNTRx register. After each RxNE event, this value will be reduced progressively.
3) The channel priority is configured by the PL[0:1] bits in the DMA1_CFGRx register.
4) Clear the DIR bit in the DMA1_CFGRx register, and it can be configured to issue an interrupt request according to application requirements when the data transmission is half or wholly completed.
5) Activate the channel by setting the EN bit in the DMA1_CFGRx register.

When the number of data transfer bytes set in the DMA controller has been completed, the DMA controller will send an EOT/EOT_1 signal indicating the end of the transmission to the I2C interface. When the interrupt is allowed, a DMA interrupt will be generated.

16.9 Packet Error Checking

Packet error checking (PEC) is a CRC8 check step added to provide the transmission reliability. Each bit of serial data can be calculated through the following polynomial:

$$C=X^8+X^2+X+1$$

PEC calculation is activated by the ENPEC bit in the control register, and all information bytes are calculated, including address and read/write bits. During transmission, enabling PEC will add a byte of CRC8 calculation result after the last byte of data. While in receiver mode, the last byte is considered to be the CRC8 check result. If it does not match the internal calculation result, it will reply with a NAK. For the master receiver, it will reply with a NAK regardless of whether the check result is correct or not.

16.10 Register Description

Table 16-1 I2C1 registers

Name	Access address	Description	Reset value
R16_I2C1_CTLR1	0x40005400	I2C1 control register 1	0x0000
R16_I2C1_CTLR2	0x40005404	I2C1 control register 2	0x0000
R16_I2C1_OADDR1	0x40005408	I2C1 address register 1	0x0000
R16_I2C1_OADDR2	0x4000540C	I2C1 address register 2	0x0000
R16_I2C1_DATAR	0x40005410	I2C1 data register	0x0000
R16_I2C1_STAR1	0x40005414	I2C1 status register 1	0x0000
R16_I2C1_STAR2	0x40005418	I2C1 status register 2	0x0000
R16_I2C1_CKCFGR	0x4000541C	I2C1 clock register	0x0000

Table 16-2 I2C2 registers

Name	Access address	Description	Reset value

R16_I2C2_CTLR1	0x40005800	I2C2 control register 1	0x0000
R16_I2C2_CTLR2	0x40005804	I2C2 control register 2	0x0000
R16_I2C2_OADDR1	0x40005808	I2C2 address register 1	0x0000
R16_I2C2_OADDR2	0x4000580C	I2C2 address register 2	0x0000
R16_I2C2_DATAR	0x40005810	I2C2 data register	0x0000
R16_I2C2_STAR1	0x40005814	I2C2 status register 1	0x0000
R16_I2C2_STAR2	0x40005818	I2C2 status register 2	0x0000
R16_I2C2_CKCFGR	0x4000581C	I2C2 clock register	0x0000

16.10.1 I2Cx Control Register 1 (R16_I2Cx_CTLR1) (x=1/2)

Offset address: 0x00

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SWRST	Reserved		PEC	POS	ACK	STOP	START	NOSTRETCH	ENGC	ENPEC	Reserved				PE

Bit	Name	Access	Description	Reset value
15	SWRST	RW	Software reset. Setting this bit by user code will reset the I2C peripheral. Before reset, make sure that the pins of the I2C bus are released and the bus is idle. *Note: This bit can reset the I2C module when no stop condition is detected on the bus but the busy bit is 1.*	0
[14:13]	Reserved	RO	Reserved.	0
12	PEC	RW	Packet error check enable. Set this bit to enable data packet error detection. This bit can be set or cleared by the user code. When the PEC is transmitted, or a start or end signal is generated, or the PE bit is cleared to 0, the bit can be cleared by hardware. 1: Provided with PEC; 0: Not provided with PEC. *Note: PEC will fa arbitration is lost.*	0
11	POS	RW	ACK and PEC position setting. This bit can be set and cleared by user code, and it can be cleared by hardware after PE is cleared; 1: The ACK bit controls the ACK or NAK of the next byte received in the shift register. The next byte received in the PEC shift register is PEC; 0: The ACK bit controls the ACK or NAK of the byte currently being accepted within the shift register. The PEC bit indicates that the current shift register byte is PEC. *Note: The usage of POS bit in 2-byte data reception is as follows: It must be configured before receiving. For the second byte of NACK, the ACK bit must be*	0

			cleared immediately after the ADDR bit is cleared; in order to detect the PEC of the second byte, the PEC bit must be set after the ADDR event occurs following the POS bit.	
10	ACK	RW	Acknowledge enable. This bit can be set or cleared by user code. When PE bit is set, this bit can be cleared by hardware; 1: Acknowledge returned after a byte is received; 0: No acknowledge is returned.	0
9	STOP	RW	Stop event generation. It can be set or cleared by user code, or cleared by hardware when a stop event is detected, or set by hardware when a timeout error is detected. In master mode: 1: A stop event is generated after the current byte transfer or the current start condition is issued; 0: No stop event occurs. In slave mode: 1: Release the SCL and SDA lines after the current byte transfer; 0: No stop event occurs.	0
8	START	RW	Start event generation. This bit can be set or cleared by the user code. When the start condition is issued or PE is cleared, it can be cleared by hardware. In master mode: 1: A start event is generated repeatedly; 0: No start event is generated. In slave mode: 1: When the bus is idle, a start event is generated; 0: No start event is generated.	0
7	NOSTRETCH	RW	Clock stretching disable. This bit is used to disable clock stretching in slave mode when the ADDR or BTF flag bit is set until it is cleared by software. 1: Clock stretching disabled; 0: Clock stretching enabled.	0
6	ENGC	RW	General call enable. Set this bit to enable the general call, and respond to general address 00h.	0
5	ENPEC	RW	PEC enable. Set this bit to enable PEC calculation.	0
[4:1]	Reserved	RO	Reserved.	0
0	PE	RW	I2C peripheral enable. 1: I2C module enabled; 0: I2C module disabled.	0

16.10.2 I2Cx Control Register 2 (R16_I2Cx_CTLR2) (x=1/2)

Offset address: 0x04

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved			LAST	DMAEN	ITBUFEN	ITEVTEN	ITERREN	Reserved		FREQ[5:0]					

Bit	Name	Access	Description	Reset value
[15:13]	Reserved	RO	Reserved	0
12	LAST	RW	Last transfer setting of DMA. 1: Next DMA EOT is the last transfer; 0: Next DMA EOT is not the last transfer. *Note: This bit is used in master receiver mode and can generate a NAK when the data is received at the last time.*	0
11	DMAEN	RW	DMA request enable. Set this bit to enable DMA request when TxE or RxEN bit is set.	0
10	ITBUFEN	RW	Buffer interrupt enable. 1: When the TxE bit is set, or when the RxEN bit is set, an event interrupt is generated; 0: When the TxE bit is set, or when the RxEN bit is set, no interrupt is generated.	0
9	ITEVTEN	RW	Event interrupt enable. Set this bit to enable event interrupt. Under the following conditions, the interrupt can be generated: SB=1 (master mode); ADDR=1(master and slave modes); ADDR10=1 (master mode); STOPF=1 (slave mode); BTF=1, but no TxE or RxEN event occurs; If ITBUFEN=1, TxE event is 1; If ITBUFEN=1, RxNE event is 1.	0
8	ITERREN	RW	Error interrupt enable. When the bit is set, the error interrupt is enabled. Under the following conditions, the interrupt can be generated: BERR=1; ARLO=1; AF=1; OVR=1; PECERR=1; TIMEOUT=1; SMBAlert=1.	0
[7:6]	Reserved	RO	Reserved.	0
[5:0]	FREQ[5:0]	RW	I2C module clock frequency domain, must input the correct clock frequency to produce the correct timing, the allowable range is between 4-60MHz. Must be set between 000100b and 111100b in MHz.	0

16.10.3 I2Cx Address Register 1 (R16_I2Cx_OADDR1) (x=1/2)

Offset address: 0x08

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDMODE	Reserved							ADD[9:8]		ADD[7:1]					ADD0

Bit	Name	Access	Description	Reset value
15	ADDMODE	RW	Addressing mode. 1: 10-bit slave address (7-bit address not acknowledged); 0: 7-bit slave address (10-bit address not acknowledged).	0
[14:10]	Reserved	RO	Reserved.	0
[9:8]	ADD[9:8]	RW	Interface address, which is 9-8 bits when using 10-bit addresses and ignored when 7-bit addresses are used.	0
[7:1]	ADD[7:1]	RW	Interface address, bit 7-1.	0
0	ADD0	RW	Interface address. Bit0 when a 10-bit address is used. It is ignored when a 7-bit address is used.	0

16.10.4 I2Cx Address Register 2 (R16_I2Cx_OADDR2) (x=1/2)

Offset address: 0x0C

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								ADD2[7:1]							ENDUAL

Bit	Name	Access	Description	Reset value
[15:8]	Reserved	RO	Reserved	0
[7:1]	ADD2[7:1]	RW	Interface address. Bit7 to bit1 of address in dual-address mode.	0
0	ENDUAL	RW	Dual addressing mode enable. When this bit is set, the ADD2 can be identified.	0

16.10.5 I2Cx Data Register (R16_I2Cx_DATAR) (x=1/2)

Offset address: 0x10

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

Bit	Name	Access	Description	Reset value
[15:8]	Reserved	RO	Reserved.	0
[7:0]	DR[7:0]	RW	Data register. This domain is used to store received data or store data to be transmitted to the bus.	0

16.10.6 I2Cx Status Register 1 (R16_I2Cx_STAR1) (x=1/2)

Offset address: 0x14

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved			PECERR	OVR	AF	ARLO	BERR	TxE	RxNE	Reserved	STOPF	ADD10	BTF	ADDR	SB

Bit	Name	Access	Description	Reset value
[15:13]	Reserved	RO	Reserved	0
12	PECERR	RW0	PEC error flag occurs during reception. This bit can be reset by user writing 0, or reset by hardware when PE becomes low. 1: PEC error. After PEC is received, NAK is returned; 0: No PEC error.	0
11	OVR	RW0	Overrun and underrun flag. 1: Overrun or underrun event occurs: In case of NOSTRETCH=1, when a new byte is received in the receiver mode and the content in the data register has not been read, the newly received byte will be lost. In the transmitter mode, no new data is written into the data register, and the same byte will be sent twice; 0: No overrun and underrun event.	0
10	AF	RW0	Acknowledge failure flag. This bit can be reset by user writing 0, or reset by hardware when PE becomes low. 1: Acknowledge error; 0: Normal acknowledge.	0
9	ARLO	RW0	Arbitration lost flag. It can be reset by user writing 0, or reset by hardware when PE becomes low. 1: Arbitration lost is detected and the module loses control of the bus; 0: Normal arbitration.	0
8	BERR	RW0	Bus error flag. It can be reset by user writing 0, or reset by hardware when PE becomes low. 1: Start or stop condition error; 0: Normal.	0
7	TxE	RO	Data register empty flag, which can be cleared by writing data to the data register, or it is automatically cleared by hardware after a start or stop bit is generated, or when PE is 0. 1: When the data is transmitted, the transmit data register is empty; 0: The data register is non-empty.	0

6	RxNE	RO	The data register is not an empty log bit, which will be cleared by reading and writing to the data register, or by the hardware when PE is 0. 1: When receiving data, the data register is not empty. 0: Normal.	0
5	Reserved	RO	Reserved.	0
4	STOPF	RO	Stop event flag. After the user reads the status register1, writing to the control register1 will clear this bit, or when PE is 0, the hardware will clear this bit. 1: After the response, the slave device will detect a stop event on the bus; 0: No stop event is detected.	0
3	ADD10	RO	10-bit address header sent flag. After the user reads the status register1, writing to the control register1 will clear this bit, or when PE is 0, the hardware will clear this bit. 1: In 10-bit address mode, the master device has sent the first address byte; 0: None	0
2	BTF	RO	Byte transmission end flag. After the user reads the status register1, reading and writing to the data register will clear this bit. During transmission, after a start or stop event is initiated, or when PE is 0, this bit will be cleared by hardware. 1: Byte transmission completed. In case of NOSTRETCH=0: when a new data is sent and the data register has not been written with new data during transmission; when a new byte is received but the data register has not been read; 0: None	0
1	ADDR	RW0	Address transmitted/matched flag. After the user reads the status register 1, the read operation of the status register2 will clear this bit, or when PE is 0, the hardware will clear this bit. Master mode: 1: End of address transmission: In 10-bit address mode, it is set when the ACK of the second byte of the address is received; in 7-bit address mode, it is set when the ACK of the address is received; 0: The address transmission is not finished. Slave mode: 1: The received address matches;	0

			0: The address does not match or no address is received.	
0	SB	RO	Start bit transmission flag. After reading the status register 1, the operation of writing the data register will clear this bit, or when PE is 0, the hardware will clear this bit. 1: The start bit has been transmitted; 0: The start bit has not been transmitted.	0

16.12.7 I2Cx Status Register 2 (R16_I2Cx_STAR2) (x=1/2)

Offset address: 0x18

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PEC[7:0]								DUALF	Reserved		GENCALL	Reserved	TRA	BUSY	MSL

Bit	Name	Access	Description	Reset value
[15:8]	PEC[7:0]	RO	Packet error checking. When PEC is enabled (ENPEC is set), this domain stores the value of PEC.	0
7	DUALF	RO	Matched detection flag. When the stop bit or start bit is generated, or when PE=0, the hardware will clear this bit. 1: The received address matched with OAR2; 0: The received address matched with OAR1.	0
[6:5]	Reserved	RO	Reserved	0
4	GENCALL	RO	General call address flag. When the stop bit or start bit is generated, or when PE=0, the hardware will clear this bit. 1: When ENGC=1, the address of general call is received; 0: No general call address is received.	0
3	Reserved	RO	Reserved	0
2	TRA	RO	Transmitter/receiver flag, cleared by hardware when a stop event (STOPF=1) is detected, repeated start condition or bus arbitration is lost (ARLO=1) or PE=0. 1: Data has been sent; 0: Data is received. This bit is determined by R/W bit of address byte.	0
1	BUSY	RO	Bus busy flag. This bit is cleared when a stop bit is detected. When the interface is disabled (PE=0), the information is still updated. 1: Busy bus: SDA or SCL has a low level; 0: The bus is idle and does not have communication.	0

0	MSL	RO	Master/slave mode indication. When the interface is in master mode (SB=1), the hardware will set this bit. When the bus detects a stop bit and the arbitration is lost, or PE=0, the hardware will clear this bit.	0

16.12.8 I2Cx Clock Register (R16_I2Cx_CKCFGR) (x=1/2)

Offset address: 0x1C

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
F/S	DUTY	Reserved		CCR[11:0]											

Bit	Name	Access	Description	Reset value
15	F/S	RW	Master mode selection. 1: Fast mode; 0: Standard mode.	0
14	DUTY	RW	Duty cycle in the fast mode: 1: $T_{Low\ level}/T_{High\ level}=16/9$; 0: $T_{Low\ level}/T_{High\ level}=2$.	0
[13:12]	Reserved	RO	Reserved.	0
[11:0]	CCR[11:0]	RW	Clock frequency division factor. These bits determine the frequency waveform of the SCL clock.	0

Chapter 17 Serial Peripheral Interface (SPI)

SPI supports data exchange in 3-wire synchronous serial mode, plus chip line selection supports hardware switching between master and slave modes, and supports communication with a single data line.

17.1 Main Features

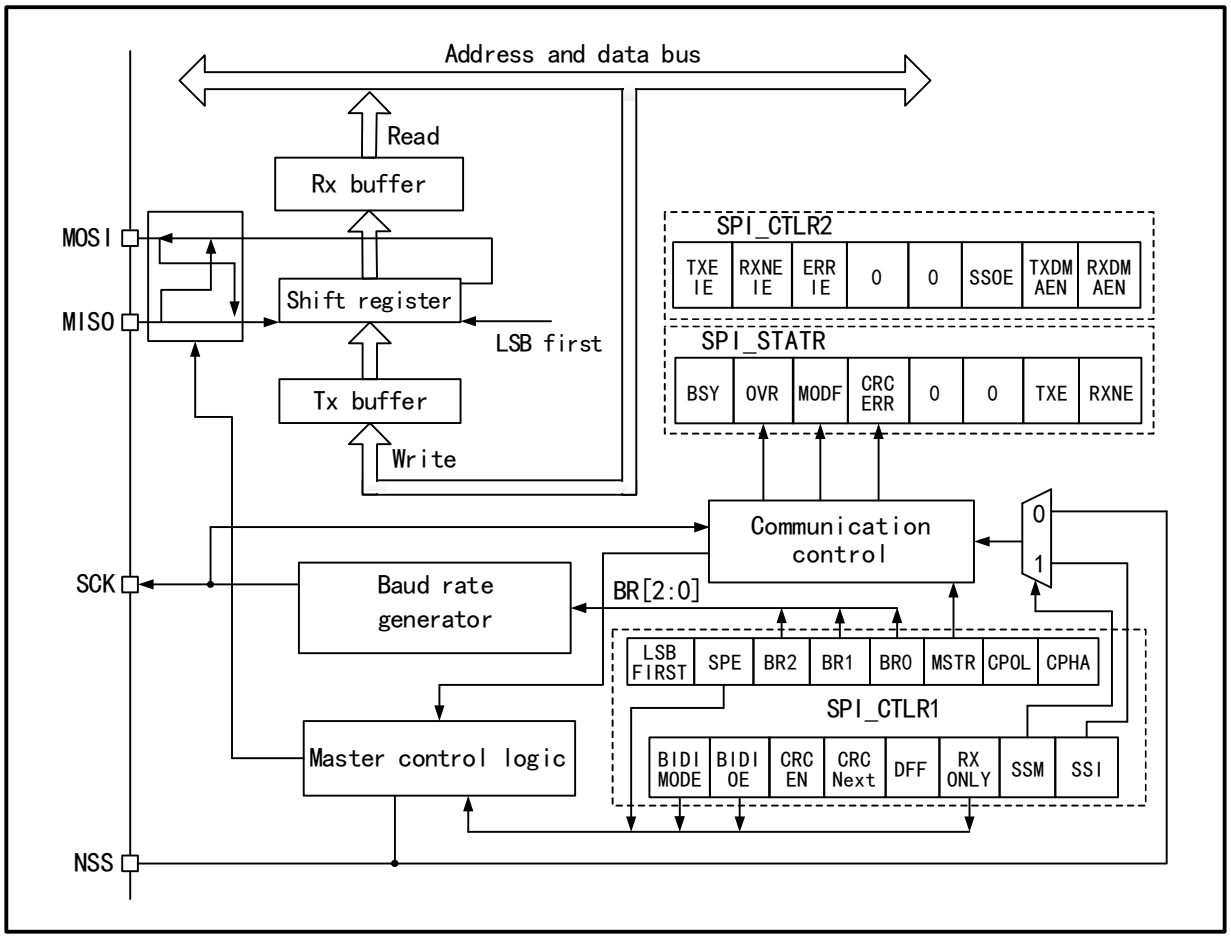
17.1.1 SPI Features

- Support full-duplex synchronous serial mode
- Support 1-wire half-duplex mode
- Support master mode and slave mode, multi-slave mode
- Support 8-bit or 16-bit data structures
- The highest clock frequency is supported to half of Fhclk
- Data ordering supports MSB or LSB first
- Support hardware or software control of the NSS pin
- Transceiver supports hardware CRC
- Transceiver buffer supports DMA transfer
- Support modifying clock phase and polarity

17.2 SPI Functional Description

17.2.1 Overview

Figure 17-1 SPI structure diagram



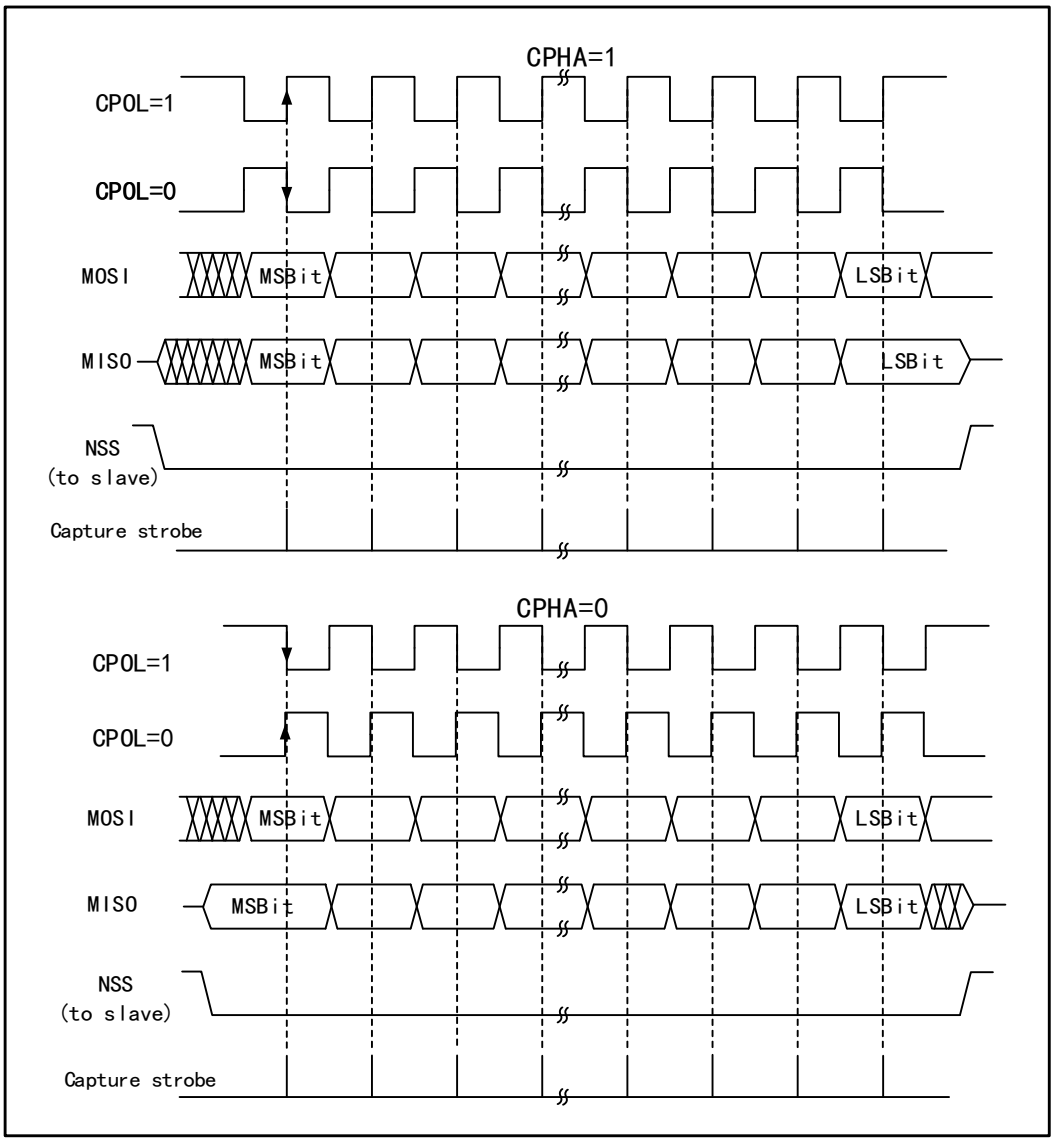
As can be seen from Figure 17-1, the 4 pins related to SPI are MISO, M0SI, SCK and NSS. The MISO pin is the data input pin when the SPI module works in the master mode; it is the data output pin when it works in the slave mode. When the MOSI pin works in master mode, it is a data output pin; when it works in slave mode, it is a data input pin. SCK is the clock pin, the clock signal is always output by the master, and the slave receives the clock signal and synchronizes data transmission and reception. The NSS pin is a chip select pin and has the following uses:

1) NSS is controlled by software: At this time, SSM is set, and the internal NSS signal is determined by SSI to output high or low. This situation is generally used in SPI master mode;

2) NSS is controlled by hardware: When the NSS output is enabled, i.e., when SSOE is set, the NSS pin will be actively pulled down when the SPI host sends the output outward, and a hardware error will be generated if the NSS pin cannot be successfully pulled down and pulled down, indicating that there are other host devices on the main line that are communicating; SSOE If it is not set, it can be used in multi-master mode. If it is pulled low, it will be forced into slave mode, and the MSTR bit will be automatically cleared.

The working mode of the SPI can be configured through CPHA and CPOL. If CPHA is set, it means that the module samples the data on the second edge of the clock, and the data is latched. If CPHA is not set, it means that the SPI module samples on the first edge of the clock, and the data is latched. CPOL indicates whether the clock remains high or low when there is no data. See Figure 17-2 below for details.

Figure 17-2 SPI mode



The host and device need to be set to the same SPI mode, and the SPE bit needs to be cleared before configuring SPI mode. The DEF bit can determine whether the single data length of the SP is 8 or 16 bits. LSBFIRST can control whether a single data word is high or low.

17.2.2 Master Mode

When the SPI module works in master mode, the serial clock is generated by SCK. To configure into master mode do the following steps:

Configure the BR[2:0] bits in the control register to determine the clock;

Configure the CPOL and CPHA bits to determine the SPI mode;

Configure DEF to determine the data word length;

Configure LSBFIRST to determine the frame format;

Configure the NSS pin, such as setting the SSOE bit to let the hardware reset NSS. You can also set the SSM bit and set the SSI bit high;

To set the MSTR bit and the SPE bit, it is necessary to ensure that NSS is already high at this time.

When you need to send data, you only need to write the data to be sent to the data register. SPI will send data from the transmit buffer to the shift register in parallel, and then send the data from the shift register according to the setting of LSBFIRST. When the data has reached the shift register, the TXE flag will be set. If it has been set TXEIE, then an interrupt will be generated. If the TXE flag bit is set, data needs to be filled into the data register to maintain a complete data flow.

When the receiver receives data, when the last sampling clock edge of the data word arrives, the data is transferred from the shift register to the receive buffer in parallel, the RXNE bit is set, and if the RXNEIE bit was previously set, it will also generate interrupt. At this time, the data register should be read as soon as possible to remove the data.

17.2.3 Slave Mode

When the SPI module works in slave mode, SCK is used to receive the clock sent by the host, and its own baud rate setting is invalid. The steps to configure into slave mode are as follows:
Configure the DEF bit to set the data bit length;
Configure CPOL and CPHA bit to match the host operating mode;
Configure LSBFIRST to match the host data frame format;
In hardware management mode, the NSS pin needs to be kept at a low level. If NSS is set to software management (SSM is set), then please keep SSI not set;

Clear the MSTR bit and set the SPE bit to enable SPI mode.

When transmitting, when the first slave receive sampling edge occurs on SCK, the slave starts to transmit. The process of sending is to move the data in the send buffer to the send shift register. When the data in the send buffer is moved to the shift register, the TXE flag will be set. If the TXEIE bit was previously set, an interrupt will be generated.

When receiving, after the last clock sampling edge, the RXNE bit is set, the byte received by the shift register is transferred to the receive buffer, and the read operation of the read data register can obtain the data in the receive buffer. An interrupt will be generated if RXNEIE is set before RXNE is set.

17.2.4 Simplex Mode

The SPI interface can work in half-duplex mode, that is, the master device uses the MOSI pin, and the slave device uses the MISO pin for communication. When using half-duplex communication, BIDIMODE needs to be set, and BIDIOE is used to control the transmission direction.

Setting the RXONLY bit in normal full-duplex mode can set the SPI module to receive-only simplex mode. After RXONLY is set, a data pin will be released. The pins released in master mode and slave mode are not the same. It is also possible to ignore the received data and set the SPI to transmit-only mode.

17.2.5 CRC

The SPI module uses CRC to ensure the reliability of full-duplex communication, and separate CRC calculators are used for data transmission and reception. The polynomial of the CRC calculation is determined by the polynomial register. For 8-bit data width and 16-bit data width, different calculation methods are used respectively.

Setting the CRCEN bit enables CRC and resets the CRC calculator. After the last data byte is sent, setting the

CRCNEXT bit will send the calculation result of the TXCRCR calculator after the current byte is sent. At the same time, if the last received value of the receive shift register is not the same as the locally calculated value of RXCRCR, then the CRCERR bit will be set. To use the CRC, you need to set the polynomial calculator and set the CRCEN bit when configuring the SPI working mode, and set the CRCNEXT bit in the last word or half word to send the CRC and check the received CRC. Note that the CRC calculation polynomials of the sender and receiver should be unified.

17.2.6 DMA

The SPI module supports the use of DMA to speed up data communication. You can use DMA to fill in data into the send buffer, or use DMA to take data from the receive buffer in time. DMA will take or send data in time with RXNE and TXE as signals. DMA can also work in simplex or CRC mode.

17.2.7 Error

- Master mode failure error

When the SPI works in the NSS pin hardware management mode, an external operation of pulling down the NSS pin occurs; or in the NSS pin software management mode, the SSI bit is cleared; or the SPE bit is cleared, causing the SPI to be turned off; or the MSTR bit is cleared, the SPI enters slave mode. An interrupt will also be generated if the ERRIE bit has been set. Clear MODF bit steps: first perform a read or write operation to R16_SPI1_STATR, and then write to R16_SPI1_CTLR1.

- Overflow error

If the master sends data and there is unread data in the slave's receive buffer, an overrun error occurs, the OVR bit is set, and an interrupt is generated if ERRIE is set. Sending an overflow error should restart the current transfer. Reading the data register followed by the status register will clear this bit.

- CRC error

When the received CRC word does not match the value of RXCRCR, a CRC error will occur, and the CRCERR bit will be set.

17.2.8 Interrupt

The interrupt of the SPI module supports 5 interrupt sources. The 2 events of the send buffer empty and the receive buffer non-empty will set TXE and RXNE respectively, and an interrupt will be generated when the TXEIE and RXNEIE bits are set respectively. In addition, the 3 errors mentioned above will also generate interrupts, namely MODF, OVR and CRCERR. After the ERRIE bit is enabled, these 3 errors will also generate error interrupts.

17.3 Register Description

Table 17-1 SPI1-related registers

Name	Access address	Description	Reset value
R16_SPI1_CTLR1	0x40013000	SPI1 control register 1	0x0000
R16_SPI1_CTLR2	0x40013004	SPI1 control register 2	0x0000
R16_SPI1_STATR	0x40013008	SPI1 status register	0x0002
R16_SPI1_DATAR	0x4001300C	SPI1 data register	0x0000
R16_SPI1_CRCR	0x40013010	SPI1 polynomial register	0x0007
R16_SPI1_RCRCR	0x40013014	SPI1 receive CRC register	0x0000
R16_SPI1_TCRCR	0x40013018	SPI1 transmit CRC register	0x0000

R16_SPI1_HSCR	0x40013024	SPI1 high-speed control register	0x0000

Table 17-2 SPI2-related registers

Name	Access address	Description	Reset value
R16_SPI2_CTLR1	0x40003800	SPI2 control register 1	0x0000
R16_SPI2_CTLR2	0x40003804	SPI2 control register 2	0x0000
R16_SPI2_STATR	0x40003808	SPI2 status register	0x0002
R16_SPI2_DATAR	0x4000380C	SPI2 data register	0x0000
R16_SPI2_CRCR	0x40003810	SPI2 polynomial register	0x0007
R16_SPI2_RCRCR	0x40003814	SPI2 receive CRC register	0x0000
R16_SPI2_TCRCR	0x40003818	SPI2 transmit CRC register	0x0000
R16_SPI2_HSCR	0x40003824	SPI2 high-speed control register	0x0000

Table 17-3 SPI3-related registers

Name	Access address	Description	Reset value
R16_SPI3_CTLR1	0x40003C00	SPI3 control register 1	0x0000
R16_SPI3_CTLR2	0x40003C04	SPI3 control register 2	0x0000
R16_SPI3_STATR	0x40003C08	SPI3 status register	0x0002
R16_SPI3_DATAR	0x40003C0C	SPI3 data register	0x0000
R16_SPI3_CRCR	0x40003C10	SPI3 polynomial register	0x0007
R16_SPI3_RCRCR	0x40003C14	SPI3 receive CRC register	0x0000
R16_SPI3_TCRCR	0x40003C18	SPI3 transmit CRC register	0x0000
R16_SPI3_HSCR	0x40003C24	SPI3 high-speed control register	0x0000

17.3.1 SPI Control Register 1 (SPIx_CTLR1) (x=1/2/3)

Offset address: 0x00

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BIDIMODE	BIDIOE	CRCEN	CRCNEXT	DFF	RXONLY	SSM	SSI	LSBFIRST	SPE	BR[2:0]			MSTR	CPOL	CPHA

Bit	Name	Access	Description	Reset value
15	BIDIMODE	RW	Unidirectional data mode enable. 1: 1-wire bidirectional mode; 0: 2-wire bidirectional mode.	0
14	BIDIOE	RW	Single-line output enable, used in conjunction with BIDIMODE. 1: Enable output, only transmit; 0: Disable output, only receive.	0
13	CRCEN	RW	Hardware CRC check enable, this bit can only be written when SPE is 0, this bit can only be used in full duplex mode.	0

			1: Enable CRC calculation; 0: Disable CRC calculation.	
12	CRCNEXT	RW	Value of the transmit CRC register after next data transfer. This bit should be set immediately after writing the last data to the data register. 1: Transmit the CRC result; 0: Continue to transmit the data of the data register.	0
11	DFF	RW	Data frame length, this bit can only be written when SPE is 0. 1: Use 16-bit data length to transmit and receive; 0: Use 8-bit data length for transmission and reception.	0
10	RXONLY	RW	Rx only in 2-wire mode. It is used in conjunction with BIDIMODE. Setting this bit allows the device to only receive and not transmit. 1: Receive only, simplex mode; 0: Full-duplex mode.	0
9	SSM	RW	CS pin management, this bit determines whether the level of the NSS pin is controlled by hardware or software. 1: Software control NSS pin; 0: Hardware controls NSS pin.	0
8	SSI	RW	CS pin control. When SSM is set, this bit determines the level of the NSS pin. 1: NSS is high level; 0: NSS is low level.	0
7	LSBFIRST	RW	Frame format control. This bit cannot be modified during communication. 1: Transmit LSB first; 0: Transmit MSB first.	0
6	SPE	RW	SPI enable. 1: Enable SPI; 0: Disable SPI.	0
[5:3]	BR[2:0]	RW	Baud rate setting domain, this domain cannot be modified during communication. When HSRXEN bit is 0, the SCK frequency is: 000: $F_{HCLK}/2$; 001: $F_{HCLK}/4$; 010: $F_{HCLK}/8$; 011: $F_{HCLK}/16$; 100: $F_{HCLK}/32$; 101: $F_{HCLK}/64$; 110: $F_{HCLK}/128$; 111: $F_{HCLK}/256$. When HSRXEN bit is 1, the SCK frequency is: 000: $F_{HCLK}/2$; 001: $F_{HCLK}/3$; 010: $F_{HCLK}/4$; 011: $F_{HCLK}/5$;	0

			100: F_{HCLK}/6; 101: F_{HCLK}/7; 110: F_{HCLK}/8; 111: F_{HCLK}/9.	
2	MSTR	RW	Master/slave setting. It cannot be modified during communication. 1: Configure as the master device; 0: Configured as the slave device.	0
1	CPOL	RW	Clock polarity selection. It cannot be modified during communication. 1: In idle state, SCK remains high; 0: In idle state, SCK remains low.	0
0	CPHA	RW	Clock phase setting. It cannot be modified during communication. 1: Data sampling starts from the second clock edge; 0: Data sampling starts from the first clock edge.	0

17.3.2 SPI Control Register 2 (SPIx_CTLR2) (x=1/2/3)

Offset address: 0x04

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								TXEIE	RXNEIE	ERRIE	Reserved		SSOE	TXDMAEN	RXDMAEN

Bit	Name	Access	Description	Reset value
[15:8]	Reserved	RO	Reserved	0
7	TXEIE	RW	Transmit buffer empty interrupt enable: 1: Allow interrupts to be generated when TXE is set; 0: Disable interrupts to be generated when TXE is set.	0
6	RXNEIE	RW	Receive buffer not empty interrupt enable: 1: Allow interrupts to be generated when RXNE is set; 0: Disable interrupts to be generated when RXNE is set.	0
5	ERRIE	RW	Error interrupt enable: 1: Allow interrupts when errors (CRCERR, OVR, MODF) occur; 0: Disable interrupts when errors (CRCERR, OVR, MODF) occur.	0
[4:3]	Reserved	RO	Reserved	0
2	SSOE	RW	SS output enable. Disable SS output to work in multi-master mode. 1: Enable SS output;	0

			0: Disable SS output in master mode.	
1	TXDMAEN	RW	Transmit buffer DMA enable. 1: Enable transmit buffer DMA; 0: Disable transmit buffer DMA.	0
0	RXDMAEN	RW	Receive buffer DMA enable. 1: Enable receive buffer DMA; 0: Disable receive buffer DMA.	0

17.3.3 SPI Status Register (SPIx_STATR) (x=1/2/3)

Offset address: 0x08

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								BSY	OVR	MODF	CRCERR	Reserved		TXE	RXNE

Bit	Name	Access	Description	Reset value
[15:8]	Reserved	RO	Reserved	0
7	BSY	RO	Busy flag. Set and reset by hardware. 1: SPI is communicating, or the transmit buffer is not empty; 0: SPI is not communicating.	0
6	OVR	RW0	Overflow flag. Set by hardware, and reset by software. 1: An overflow error occurred; 0: No overflow error occurred.	0
5	MODF	RO	Mode error flag. Set by hardware, and reset by software. 1: A mode error occurred; 0: No mode error occurred.	0
4	CRCERR	RW0	CRC error flag. Set by hardware, and reset by software. 1: The received CRC value is inconsistent with the RCRCR value; 0: The received CRC value is the same as the RCRCR value.	0
[3:2]	Reserved	RO	Reserved	0
1	TXE	RO	Transmit buffer empty flag. 1: The transmit buffer is empty; 0: The transmit buffer is not empty.	1
0	RXNE	RO	Receive buffer not empty flag. 1: The receive buffer is not empty; 0: The receive buffer is empty. *Note: Reading DATAR will automatically clear it.*	0

17.3.4 SPI Data Register (SPIx_DATAR) (x=1/2/3)

Offset address: 0x0C

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

Bit	Name	Access	Description	Reset value
[15:0]	DR[15:0]	RW	Data register. The data register is used to store the received data or pre-store the data to be sent, so the read and write of the data register is actually corresponding to the different operation area, where the read corresponds to the receiving buffer and the write corresponds to the sending buffer. The reception and transmission of data can be 8-bit or 16-bit, and how many bits of data need to be determined before transmission. When using 8 bits for data transmission, only the lower 8 bits of the data register are used, and the high 8 bits are forced to 0 when received. The use of 16-bit data structures causes all 16-bit data registers to be used.	0

17.3.5 SPI Polynomial Register (SPIx_CRCR) (x=1/2/3)

Offset address: 0x10

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

Bit	Name	Access	Description	Reset value
[15:0]	CRCPOLY[15:0]	RW	CRC polynomial. This domain defines the polynomial used for the CRC calculation.	0x0007

19.3.6 SPI Receive CRC Register (SPIx_RCRCR) (x=1/2/3)

Offset address: 0x14

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

Bit	Name	Access	Description	Reset value
[15:0]	RXCRC[15:0]	RO	Rx CRC value. Stores the result of the calculated CRC check of the received bytes. Setting CRCEN will reset this register. The calculation method uses the polynomial used by CRCPOLY. In 8-bit mode, only the lower 8 bits participate in the calculation, and in 16-bit mode, all 16 bits will participate in the calculation. This register needs	0

			to be read when BSY is 0.	

17.3.7 SPI Transmit CRC Register (SPIx_TCRCR) (x=1/2/3)

Offset address: 0x18

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

Bit	Name	Access	Description	Reset value
[15:0]	TXCRC[15:0]	RO	Send a CRC value. Stores the calculated results of the CRC check of the bytes that have been sent. Setting CRCEN resets the register. The calculation method uses the polynomials used by CRCPOLY. In 8-bit mode, only the lower 8 bits participate in the calculation, while in 16-bit mode, all 16 bits participate in the calculation. You need to read this register when BSY is 0.	0

17.3.8 SPI High-speed Control Register (SPIx_HSCR) (x=1/2/3)

Offset address: 0x24

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved													HSRXEN2	Reserved	HSRXEN

Bit	Name	Access	Description	Reset value
[15:3]	Reserved	RO	Reserved	0
2	HSRXEN2	RW	SPI high-speed read mode 2: This mode is only used when HSRXEN is turned on and SPI PCLK is greater than or equal to 120M. It is not recommended to use this bit if it is less than 120M. You can get higher host read speeds than just turning on HSRXEN. 1: Enable high-speed read mode 2; 0: Disable high-speed read mode 2.	0
1	Reserved	RO	Reserved	0
0	HSRXEN	RW	Read enable in SPI high-speed mode. 1: High-speed read mode enabled; 0: High-speed read mode disabled.	0

Chapter 18 USB High-speed Host/Device Controller (USBHS)

The USB2.0 high-speed controller functions as both a host controller and a device controller, featuring an integrated 480Mbps USB-PHY physical layer transceiver. When operating as a host controller, it supports low-speed, full-speed, and high-speed USB devices. When functioning as a device controller, it can be flexibly configured for low-speed, full-speed, or high-speed modes to accommodate diverse applications.

18.1 USB Controller Introduction

USB Controller features include:

- Support USB2.0, USB1.1, and USB1.0 protocol specifications
- Support USB Host function and USB Device function.
- Support control transfer, bulk transfer, interrupt transfer and isochronous/real-time transfer.
- Provide bus reset, suspend, wake-up and recovery functions.
- Host supports USB HUB
- Non-zero endpoints all support the maximum packet size of 1024 bytes, with built-in FIFO, interrupt and DMA support.

18.2 Register Description

The USB registers are divided into 3 parts:

- USB high-speed device control registers
- USB high-speed host control registers
- HSI calibration register

18.2.1 Device Register Description

Table 18-1 Device-ralated registers

Name	Access address	Description	Reset value
R8_USB_CTRL	0x40030000	USBHS control register	0x07
R8_USB_BASE_MODE	0x40030001	USBHS mode control register	0x00
R8_USB_INT_EN	0x40030002	USBHS interrupt enable register	0x00
R8_USB_DEV_AD	0x40030003	USBHS device address register	0x00
R8_USB_WAKE_CTRL	0x40030004	USBHS remote wakeup register	0x00
R8_USB_TEST_MODE	0x40030005	USBHS test mode register	0x00
R16_USB_LPM_DATA	0x40030006	USBHS power management register	0x8000
R8_USB_INT_FG	0x40030008	USBHS interrupt flag register	0x00
R8_USB_INT_ST	0x40030009	USBHS interrupt status register	0x00
R8_USB_MIS_ST	0x4003000A	USBHS miscellaneous status register	0x00
R16_USB_FRAME_NO	0x4003000C	USBHS frame number register	0x0000
R16_USB_BUS	0x4003000E	USBHS bus status register	0x0000
R16_UEP_TX_EN	0x40030010	USBHS endpoint transmit enable register	0x0000
R16_UEP_RX_EN	0x40030012	USBHS endpoint receive enable register	0x0000
R16_UEP_T_TOG_AUTO	0x40030014	USBHS endpoint transmit auto-flip enable register	0x0000

R16_UEP_R_TOG_AUTO	0x40030016	USBHS endpoint receive auto-flip enable register	0x0000
R8_UEP_T_BURST	0x40030018	USBHS endpoint transmit burst register	0x00
R8_UEP_T_BURST_MODE	0x40030019	USBHS endpoint transmit burst mode register	0x00
R8_UEP_R_BURST	0x4003001A	USBHS endpoint receive burst register	0x00
R8_UEP_R_RES_MODE	0x4003001B	USBHS endpoint receive burst mode register	0x00
R32_UEP_AF_MODE	0x4003001C	USBHS endpoint alternate register	0x00000000
R32_UEP0_DMA	0x40030020	Start address register of endpoint 0 receive buffer	0x000XXXXX
R32_UEP1_RX_DMA	0x40030024	Start address register of endpoint 1 receive buffer	0x000XXXXX
R32_UEP2_RX_DMA	0x40030028	Start address register of endpoint 2 receive buffer	0x000XXXXX
R32_UEP3_RX_DMA	0x4003002C	Start address register of endpoint 3 receive buffer	0x000XXXXX
R32_UEP4_RX_DMA	0x40030030	Start address register of endpoint 4 receive buffer	0x000XXXXX
R32_UEP5_RX_DMA	0x40030034	Start address register of endpoint 5 receive buffer	0x000XXXXX
R32_UEP6_RX_DMA	0x40030038	Start address register of endpoint 6 receive buffer	0x000XXXXX
R32_UEP7_RX_DMA	0x4003003C	Start address register of endpoint 7 receive buffer	0x000XXXXX
R32_UEP1_TX_DMA	0x40030040	Start address register of endpoint 1 transmit buffer	0x000XXXXX
R32_UEP2_TX_DMA	0x40030044	Start address register of endpoint 2 transmit buffer	0x000XXXXX
R32_UEP3_TX_DMA	0x40030048	Start address register of endpoint 3 transmit buffer	0x000XXXXX
R32_UEP4_TX_DMA	0x4003004C	Start address register of endpoint 4 transmit buffer	0x000XXXXX
R32_UEP5_TX_DMA	0x40030050	Start address register of endpoint 5 transmit buffer	0x000XXXXX
R32_UEP6_TX_DMA	0x40030054	Start address register of endpoint 6 transmit buffer	0x000XXXXX
R32_UEP7_TX_DMA	0x40030058	Start address register of endpoint 7 transmit buffer	0x000XXXXX
R32_UEP0_MAX_LEN	0x4003005C	Endpoint 0 maximum length packet register	0x000000XX
R32_UEP1_MAX_LEN	0x40030060	Endpoint 1 maximum length packet register	0x00000XXX
R32_UEP2_MAX_LEN	0x40030064	Endpoint 2 maximum length packet register	0x00000XXX
R32_UEP3_MAX_LEN	0x40030068	Endpoint 3 maximum length packet register	0x00000XXX
R32_UEP4_MAX_LEN	0x4003006C	Endpoint 4 maximum length packet register	0x00000XXX
R32_UEP5_MAX_LEN	0x40030070	Endpoint 5 maximum length packet register	0x00000XXX
R32_UEP6_MAX_LEN	0x40030074	Endpoint 6 maximum length packet register	0x00000XXX
R32_UEP7_MAX_LEN	0x40030078	Endpoint 7 maximum length packet register	0x00000XXX
R16_UEP0_RX_LEN	0x4003007C	Endpoint 0 receive length register	0x00XX
R16_UEP1_RX_LEN	0x40030080	Endpoint 1 single receive length register	0xXXXX
R16_UEP1_R_SIZE	0x40030082	Endpoint 1 total receive data length register	0xXXXX
R16_UEP2_RX_LEN	0x40030084	Endpoint 2 single receive length register	0xXXXX
R16_UEP2_R_SIZE	0x40030086	Endpoint 2 total receive data length register	0xXXXX
R16_UEP3_RX_LEN	0x40030088	Endpoint 3 single receive length register	0xXXXX
R16_UEP3_R_SIZE	0x4003008A	Endpoint 3 total receive data length register	0xXXXX
R16_UEP4_RX_LEN	0x4003008C	Endpoint 4 single receive length register	0xXXXX
R16_UEP4_R_SIZE	0x4003008E	Endpoint 4 total receive data length register	0xXXXX
R16_UEP5_RX_LEN	0x40030090	Endpoint 5 single receive length register	0xXXXX
R16_UEP5_R_SIZE	0x40030092	Endpoint 5 total receive data length register	0xXXXX
R16_UEP6_RX_LEN	0x40030094	Endpoint 6 single receive length register	0xXXXX
R16_UEP6_R_SIZE	0x40030096	Endpoint 6 total receive data length register	0xXXXX
R16_UEP7_RX_LEN	0x40030098	Endpoint 7 single receive length register	0xXXXX

R16_UEP7_R_SIZE	0x4003009A	Endpoint 7 total receive data length register	0xXXXX
R16_UEP0_T_LEN	0x4003009C	Endpoint 0 transmit length register	0x00XX
R8_UEP0_TX_CTRL	0x4003009E	Endpoint 0 transmit control register	0x00
R8_UEP0_RX_CTRL	0x4003009F	Endpoint 0 receive control register	0x00
R16_UEP1_T_LEN	0x400300A0	Endpoint 1 transmit length register	0x0000
R8_UEP1_TX_CTRL	0x400300A2	Endpoint 1 transmit control register	0x00
R8_UEP1_RX_CTRL	0x400300A3	Endpoint 1 receive control register	0x00
R16_UEP2_T_LEN	0x400300A4	Endpoint 2 transmit length register	0x0000
R8_UEP2_TX_CTRL	0x400300A6	Endpoint 2 transmit control register	0x00
R8_UEP2_RX_CTRL	0x400300A7	Endpoint 2 receive control register	0x00
R16_UEP3_T_LEN	0x400300A8	Endpoint 3 transmit length register	0x0000
R8_UEP3_TX_CTRL	0x400300AA	Endpoint 3 transmit control register	0x00
R8_UEP3_RX_CTRL	0x400300AB	Endpoint 3 receive control register	0x00
R16_UEP4_T_LEN	0x400300AC	Endpoint 4 transmit length register	0x0000
R8_UEP4_TX_CTRL	0x400300AE	Endpoint 4 transmit control register	0x00
R8_UEP4_RX_CTRL	0x400300AF	Endpoint 4 receive control register	0x00
R16_UEP5_T_LEN	0x400300B0	Endpoint 5 transmit length register	0x0000
R8_UEP5_TX_CTRL	0x400300B2	Endpoint 5 transmit control register	0x00
R8_UEP5_RX_CTRL	0x400300B3	Endpoint 5 receive control register	0x00
R16_UEP6_T_LEN	0x400300B4	Endpoint 6 transmit length register	0x0000
R8_UEP6_TX_CTRL	0x400300B6	Endpoint 6 transmit control register	0x00
R8_UEP6_RX_CTRL	0x400300B7	Endpoint 6 receive control register	0x00
R16_UEP7_T_LEN	0x400300B8	Endpoint 7 transmit length register	0x0000
R8_UEP7_TX_CTRL	0x400300BA	Endpoint 7 transmit control register	0x00
R8_UEP7_RX_CTRL	0x400300BB	Endpoint 7 receive control register	0x00
R16_UEP_T_ISO	0x400300BC	USBHS endpoint transmit synchronization mode enable register	0x0000
R16_UEP_R_ISO	0x400300BE	USBHS endpoint receive synchronization mode enable register	0x0000
R32_UEP1_RX_FIFO	0x400300C0	Receive FIFO address of USBHS endpoint 1	0x00000000
R32_UEP2_RX_FIFO	0x400300C4	Receive FIFO address of USBHS endpoint 2	0x00000000
R32_UEP3_RX_FIFO	0x400300C8	Receive FIFO address of USBHS endpoint 3	0x00000000
R32_UEP4_RX_FIFO	0x400300CC	Receive FIFO address of USBHS endpoint 4	0x00000000
R32_UEP5_RX_FIFO	0x400300D0	Receive FIFO address of USBHS endpoint 5	0x00000000
R32_UEP6_RX_FIFO	0x400300D4	Receive FIFO address of USBHS endpoint 6	0x00000000
R32_UEP7_RX_FIFO	0x400300D8	Receive FIFO address of USBHS endpoint 7	0x00000000
R32_UEP1_TX_FIFO	0x400300DC	Transmit FIFO address of USBHS endpoint 1	0x00000000
R32_UEP2_TX_FIFO	0x400300E0	Transmit FIFO address of USBHS endpoint 2	0x00000000
R32_UEP3_TX_FIFO	0x400300E4	Transmit FIFO address of USBHS endpoint 3	0x00000000
R32_UEP4_TX_FIFO	0x400300E8	Transmit FIFO address of USBHS endpoint 4	0x00000000
R32_UEP5_TX_FIFO	0x400300EC	Transmit FIFO address of USBHS endpoint 5	0x00000000
R32_UEP6_TX_FIFO	0x400300F0	Transmit FIFO address of USBHS endpoint 6	0x00000000
R32_UEP7_TX_FIFO	0x400300F4	Transmit FIFO address of USBHS endpoint 7	0x00000000

R32_16K_MODE_CTRL	0x400300F8	USBHS 16K mode control register	0x00000000
R32_16K_MODE_TIME	0x400300FC	USBHS 16K mode time register	0x4F391064

18.2.1.1 USBHS Control Register (R8_USB_CTRL)

Offset address: 0x00

Bit	Name	Access	Description	Reset value
7	RB_UD_LPM_EN	RW	LPM enable: 1: Enable; 0: Disable.	0
6	Reserved	RO	Reserved	0
5	RB_UD_DEV_EN	RW	USB device enable: 1: Enable; 0: Disable.	0
4	RB_UD_DMA_EN	RW	DMA transfer enable: 1: Enable; 0: Disable.	0
3	RB_UD_PHY_SUSPENDM	RW	USB PHY suspended: 1: Normal operation; 0: Suspended.	0
2	RB_UD_CLR_ALL	RW	Clear all interrupt flags: 1: Clear USB interrupt flag and FIFO; requires software reset; 0: Do not clear.	1
1	RB_UD_RST_SIE	RW	USB protocol processor reset: 1: Force reset the USB Protocol Processor (SIE), including endpoint-related registers, requiring software clearing; 0: Do not reset.	1
0	RB_UD_RST_LINK	RW	LINK layer reset: 1: USB Link layer reset; 0: No reset.	1

18.2.1.2 USBHS Mode Control Register (R8_USB_BASE_MODE)

Offset address: 0x01

Bit	Name	Access	Description	Reset value
[7:2]	Reserved	RO	Reserved	0x0
[1:0]	RB_UD_SPEED_TYPE[1:0]	RW	Desired speed mode for device: 00: Full-Speed device; 01: High-Speed device; 10: Low-Speed device; 11: Reserved.	0

18.2.1.3 USBHS Interrupt Enable Register (R8_USB_INT_EN)

Offset address: 0x02

Bit	Name	Access	Description	Reset value
7	RB_UDIE_FIFO_OVER	RW	FIFO overflow interrupt. 1: Enable interrupt; 0: Disable interrupt.	0
6	RB_UDIE_LINK_RDY	RW	USB connection interrupt enabled. 1: Enable interrupt; 0: Disable interrupt.	0
5	RB_UDIE_SOF_ACT	RW	SOF packet interrupt enable. 1: Enable interrupt; 0: Disable interrupt.	0
4	RB_UDIE_TRANSFER	RW	USB transfer complete interrupt enable. 1: Enable interrupt; 0: Disable interrupt.	0
3	RB_UDIE_LPM_ACT	RW	LPM transfer end interrupt enable. 1: Enable interrupt; 0: Disable interrupt.	0
2	RB_UDIE_BUS_SLEEP	RW	USB bus sleep interrupt enable. 1: Enable interrupt; 0: Disable interrupt.	0
1	RB_UDIE_SUSPEND	RW	USB bus interrupt enable. 1: Enable interrupt; 0: Disable interrupt.	0
0	RB_UDIE_BUS_RST	RW	USB bus reset interrupt enable. 1: Enable interrupt; 0: Disable interrupt.	0

18.2.1.4 USBHS Device Address Register (R8_USB_DEV_AD)

Offset address: 0x03

Bit	Name	Access	Description	Reset value
7	Reserved	RO	Reserved	0
[6:0]	RB_UD_DEV_ADDR[6:0]	RW	USB device address.	0

18.2.1.5 USBHS Remote Wakeup Register (R8_USB_WAKE_CTRL)

Offset address: 0x04

Bit	Name	Access	Description	Reset value
[7:1]	Reserved	RO	Reserved	0
0	RB_UD_REMOTE_WKUP	RW1Z	Remote wake-up, hardware automatically reset.	0

18.2.1.6 USBHS Test Mode Register (R8_USB_TEST_MODE)

Offset address: 0x05

Bit	Name	Access	Description	Reset value
7	RB_UD_TEST_EN	RW	Test mode enable.	0
[6:4]	Reserved	RO	Reserved	0
3	RB_UD_TEST_SE0NAK	RW	Test mode, output SE0.	0
2	RB_UD_TEST_PKT	RW	Test mode, output one packet. The packet uses the data address and length from endpoint 4, with TOG set to DATA0.	0
1	RB_UD_TEST_K	RW	Test mode, output K.	0
0	RB_UD_TEST_J	RW	Test mode, output J.	0

18.2.1.7 USBHS Power Management Register (R16_USB_LPM_DATA)

Offset address: 0x06

Bit	Name	Access	Description	Reset value
15	RB_UD_LPM_BUSY	RW	Power management busy.	1
[14:11]	Reserved	RO	Reserved	0
[10:0]	RB_UD_LPM_DATA[10:0]	RO	Power management data.	0

18.2.1.8 USBHS Interrupt Flag Register (R8_USB_INT_FG)

Offset address: 0x08

Bit	Name	Access	Description	Reset value
7	RB_UDIF_FIFO_OV	RW1Z	FIFO overflow interrupt flag bit; write 1 to clear. 1: FIFO overflow triggered; 0: No event.	0
6	RB_UDIF_LINK_RDY	RW1Z	USB connection interrupt flag bit; write 1 to clear. 1: USB connection event triggered; 0: No event.	0
5	RB_UDIF_RX_SOF	RW1Z	Receive SOF packet interrupt flag bit; write 1 to clear. 1: SOF packet event triggered; 0: No event.	0
4	RB_UDIF_RTX_ACT	RO	USB transfer complete interrupt flag: Receive cleared by RB_UEP_R_DONE; Transmit cleared by RB_UEP_T_DONE.	0
3	RB_UDIF_LPM_ACT	RW1Z	LPM transmission end interrupt flag, write 1 to clear. 1: LPM transmission end event triggered; 0: No event.	0
2	RB_UDIF_BUS_SLEEP	RW1Z	USB Bus Sleep Interrupt Flag When LPM is enabled, entering L1 (Sleep)	0

			state triggers this flag, writing 1 to clear. 1: USB bus sleep event triggered; 0: No event.	
1	RB_UDIF_SUSPEND	RW1Z	USB bus suspend interrupt flag, write 1 to clear. 1: USB suspend event triggered; 0: No event.	0
0	RB_UDIF_BUS_RST	RW1Z	USB bus reset interrupt flag, write 1 to clear. 1: USB bus reset event triggered; 0: No event.	0

18.2.1.9 USBHS Interrupt Status Register (R8_USB_INT_ST)

Offset address: 0x09

Bit	Name	Access	Description	Reset value
[7:5]	Reserved	RO	Reserved	0
4	RB_UDIS_EP_DIR	RO	Endpoint data transmission direction: 1: Endpoint IN data; 0: Endpoint OUT/SETUP data.	0
3	Reserved	RO	Reserved	0
[2:0]	RB_UDIS_EP_ID_MASK[2:0]	RO	The endpoint number where data transmission occurs.	0

18.2.1.10 USBHS Miscellaneous Status Register (R8_USB_MIS_ST)

Offset address: 0x0A

Bit	Name	Access	Description	Reset value
7	RB_UDMS_HS_MOD	RO	Whether the host is at high-speed.	0
[6:5]	Reserved	RO	Reserved	0
4	RB_UDMS_SUSP_REQ	RO	USB suspend request.	0
3	RB_UDMS_SIE_FREE	RO	USB idle state.	0
2	RB_UDMS_SLEEP	RO	USB sleep state.	0
1	RB_UDMS_SUSPEND	RO	USB suspend state. 1: The USB bus is in suspend state, with no USB activity for a period of time; 0: The USB bus is in non-suspend state.	0
0	RB_UDMS_READY	RO	USB connection status.	0

18.2.1.11 USBHS Frame Number Register (R16_USB_FRAME_NO)

Offset address: 0x0C

Bit	Name	Access	Description	Reset value
[15:13]	RB_UD_MFRAME_NO[2:0]	RO	The received micro-frame number.	0
[12:11]	Reserved	RO	Reserved	0

[10:0]	RB_UD_FRAME_NO[10:0]	RO	The received frame number.	0

18.2.1.12 USBHS Bus Status Register (R16_USB_BUS)

Offset address: 0x0E

Bit	Name	Access	Description	Reset value
[15:4]	Reserved	RO	Reserved	0
3	RB_USB_DM_ST	RO	UDM status.	0
2	RB_USB_DP_ST	RO	UDP status.	0
1	Reserved	RO	Reserved	0
0	RB_USB_WAKEUP	RO	USB wakeup (active high).	0

18.2.1.13 USBHS Endpoint Transmit Enable Register (R16_UEP_TX_EN)

Offset address: 0x10

Bit	Name	Access	Description	Reset value
[15:0]	RB_UEP_TX_EN[15:0]	RW	0~15 endpoint transmit enable register: 1: Enable; 0: Disable.	0

18.2.1.14 USBHS Endpoint Receive Enable Register (R16_UEP_RX_EN)

Offset address: 0x12

Bit	Name	Access	Description	Reset value
[15:0]	RB_UEP_RX_EN[15:0]	RW	0~15 endpoint receive enable register: 1: Enable; 0: Disable.	0

18.2.1.15 USBHS Endpoint Transmit Auto-flip Enable Register (R16_UEP_T_TOG_AUTO)

Offset address: 0x14

Bit	Name	Access	Description	Reset value
[15:8]	Reserved	RO	Reserved	0
[7:1]	RB_UEP_T_TOG_AUTO[7:1]	RW	Auto flip enable for synchronous trigger bits on endpoints 1–7: 1: Automatically flips after successful data transmission; 0: Manually controls flipping.	0
0	Reserved	RO	Reserved	0

18.2.1.16 USBHS Endpoint Receive Auto-flip Enable Register (R16_UEP_R_TOG_AUTO)

Offset address: 0x16

Bit	Name	Access	Description	Reset value
[15:8]	Reserved	RO	Reserved	0
[7:0]	RB_UEP_R_TOG_	RW	Auto flip enable for synchronous trigger	0

	AUTO[7:1]		bits on endpoints 1~7: 1: Automatically flips after successful data reception; 0: Manually controls flipping.	
0	Reserved	RO	Reserved	0

18.2.1.17 USBHS Endpoint Transmit Burst Register (R8_UEP_T_BURST)

Offset address: 0x18

Bit	Name	Access	Description	Reset value
[7:0]	RB_UEP_T_BURST_EN [7:0]	RW	0~7 endpoint burst transmit enable: 1: Enable; 0: Disable.	0

18.2.1.18 USBHS Endpoint Transmit Burst Mode Register (R8_UEP_T_BURST_MODE)

Offset address: 0x19

Bit	Name	Access	Description	Reset value
[7:0]	RB_UEP_T_BURST_ MODE[7:0]	RW	0~7 endpoint burst transmit mode: 1: Do not transmit zero-length data in burst mode; 0: Transmit zero-length data in burst mode.	0

18.2.1.19 USBHS Endpoint Receive Burst Register (R8_UEP_R_BURST)

Offset address: 0x1A

Bit	Name	Access	Description	Reset value
[7:0]	RB_UEP_R_BURST _EN[7:0]	RW	0~7 endpoint burst receive enable: 1: Enable; 0: Disable.	0

18.2.1.20 USBHS Endpoint Receive Reply Mode Register (R8_UEP_R_RES_MODE)

Offset address: 0x1B

Bit	Name	Access	Description	Reset value
[7:0]	RB_UEP_R_RES_MO DE[7:0]	RW	0~7 endpoint receive return mode: 1: NYET; 0: ACK.	0

18.2.1.21 USBHS Endpoint Alternate Register (R32_UEP_AF_MODE)

Offset address: 0x1C

Bit	Name	Access	Description	Reset value
[31:8]	Reserved	RO	Reserved	0
[7:1]	RB_UEP_T_AF[6:0]	RW	Endpoint multiplexing enable for 1~7: 1: Multiplex 9~15;	0

			0: Multiplex 1~7.	
0	Reserved	RO	Reserved	0

18.2.1.22 Start Address Register of Endpoint 0 Buffer (R32_UEP0_DMA)

Offset address: 0x20

Bit	Name	Access	Description	Reset value
[31:24]	Reserved	RO	Reserved	0
[23:0]	UEP0_DMA[23:0]	RW	Start address register of endpoint 0 buffer.	X

18.2.1.23 Start Address Register of Endpoint n Receive Buffer (R32_UEPn_RX_DMA) (n=1~7)

Offset address: 0x24 + 4*(n-1)

Bit	Name	Access	Description	Reset value
[31:24]	Reserved	RO	Reserved	0
[23:0]	UEPn_RX_DMA[23:0]	RW	Start address register of endpoint n receive buffer. The address must be 4 bytes aligned.	X

18.2.1.24 Start Address Register of Endpoint n Transmit Buffer (R32_UEPn_TX_DMA) (n=1~7)

Offset address: 0x40 + 4*(n-1)

Bit	Name	Access	Description	Reset value
[31:24]	Reserved	RO	Reserved	0
[23:0]	UEPn_TX_DMA[23:0]	RW	The start address of endpoint n transmit buffer. The address must be 4-byte aligned.	X

18.2.1.25 Endpoint 0 Maximum Length Package Register (R32_UEPn_MAX_LEN)

Offset address: 0x5C

Bit	Name	Access	Description	Reset value
[31:7]	Reserved	RO	Reserved	0
[6:0]	UEP0_MAX_LEN[6:0]	RW	The maximum offset length allowed for endpoint 0 DMA.	X

18.2.1.26 Endpoint n Maximum Length Package Register (R32_UEPn_MAX_LEN) (n=1~7)

Offset address: 0x60 + 4 (n-1)

Bit	Name	Access	Description	Reset value
[31:11]	Reserved	RO	Reserved	0
[6:0]	UEPn_MAX_LEN[10:0]	RW	The maximum offset length allowed for endpoint n DMA.	X

18.2.1.27 Endpoint 0 Receive Length Register (R16_UEP0_RX_LEN)

Offset address: 0x7C

Bit	Name	Access	Description	Reset value

[15:7]	Reserved	RO	Reserved	0
[6:0]	UEP0_RX_LEN[6:0]	RW	Endpoint 0 receive data length.	X

18.2.1.28 Endpoint n Single Receive Length Register (R16_UEPn_RX_LEN) (n=1~7)

Offset address: 0x80 + 4 (n-1)

Bit	Name	Access	Description	Reset value
[15:0]	UEPn_RX_LEN[15:0]	RW	Endpoint n single receive data length.	X

18.2.1.29 Endpoint n Total Receive Length Register (R16_UEPn_R_SIZE) (n=1~7)

Offset address: 0x82 + 4 (n-1)

Bit	Name	Access	Description	Reset value
[15:0]	UEPn_R_SIZE[15:0]	RW	Endpoint n total receive data length.	X

18.2.1.30 Endpoint 0 Transmit Length Register (R16_UEP0_T_LEN)

Offset address: 0x9C

Bit	Name	Access	Description	Reset value
[15:7]	Reserved	RO	Reserved	0
[6:0]	UEP0_T_LEN[6:0]	RW	Endpoint 0 transmit data length.	X

18.2.1.31 Endpoint 0 Transmit Control Register (R8_UEP0_TX_CTRL)

Offset address: 0x9E

Bit	Name	Access	Description	Reset value
7	RB_UEP_T_DONE	RW0	Endpoint 0 transmits the end flag, writes 0 to clear it.	0
6	RB_UEP_T_NAK_ACT	RW0	Endpoint 0 transmits NAK end flag, writes 0 to clear it.	0
[5:3]	Reserved	RO	Reserved	0
2	RB_UEP_T_TOG_MASK	RW	Synchronous trigger bit expected to be sent by endpoint 0: 1: DATA1; 0: DATA0.	0
[1:0]	RB_UEP_T_RES_MASK[1:0]	RW	Endpoint 0 response control for transmission: 00: Respond with NAK or busy; 01: Respond with STALL or error; 10: Respond with ACK; 11: Reserved.	0

18.2.1.32 Endpoint 0 Receive Control Register (R8_UEP0_RX_CTRL)

Offset address: 0x9F

Bit	Name	Access	Description	Reset value
7	RB_UEP_R_DONE	RW0	Endpoint 0 receives the end flag and	0

			writes 0 to clear it.	
6	RB_UEP_R_NAK_ACT	RW0	Endpoint 0 receives the NAK end flag and writes 0 to clear it.	0
5	RB_UEP_R_NAK_TOG	RO	Endpoint 0 For the received return NAK, packet type: 1: DATA1; 0: DATA0.	0
4	RB_UEP_R_TOG_MATCH	RO	Received synchronous trigger bit matching status with expected synchronous trigger bit: 1: Synchronized; 0: Not synchronized.	0
3	RB_UEP_R_SETUP_IS	RO	Whether endpoint 0 receives a SETUP transaction.	0
2	RB_UEP_R_TOG_MASK	RW	Expected synchronous trigger bits for endpoint 0 reception: 1: DATA1; 0: DATA0.	0
[1:0]	RB_UEP_R_RES_MASK[1:0]	RW	Endpoint 0 response control for received responses: 00: Respond with NAK or busy; 01: Respond with STALL or error; 10: Respond with ACK; 11: Reserved.	0

18.2.1.33 Endpoint n Transmit Length Register (R16_UEPn_T_LEN) (n=1~7)

Offset address: 0xA0 + 4*(n-1)

Bit	Name	Access	Description	Reset value
[15:0]	UEPn_T_LEN[15:0]	RW	The length of data sent by endpoint n.	0

18.2.1.34 Endpoint n Transmit Control Register (R8_UEPn_TX_CTRL) (n=1~7)

Offset address: 0xA2 + 4*(n-1)

Bit	Name	Access	Description	Reset value
7	RB_UEP_T_DONE	RW0	Endpoint n sends the end flag and writes 0 to clear it.	0
6	RB_UEP_T_NAK_ACT	RW0	Endpoint n sends the NAK end flag and writes 0 to clear it.	0
[5:4]	Reserved	RO	Reserved	0
[3:2]	RB_UEP_T_TOG_MASK[1:0]	RW	Expected synchronization trigger bit for transmission of endpoint n: 00: DATA0; 01: DATA1; 10: DATA2;	0

			11: MDATA. *Note: This bit can be configured in manual mode and not in automatic mode.*	
[1:0]	RB_UEP_T_RES_MASK[1:0]	RW	Endpoint n response control for transmission: 00: Respond with NAK or busy; 01: Respond with STALL or error; 10: Respond with ACK; 11: Reserved.	0

18.2.1.35 Endpoint n Receive Control Register (R8_UEPn_RX_CTRL) (n=1~7)

Offset address: 0xA3 + 4*(n-1)

Bit	Name	Access	Description	Reset value
7	RB_UEP_R_DONE	RW0	Endpoint n receives the end flag and writes 0 to clear it.	0
6	RB_UEP_R_NAK_ACT	RW0	Endpoint n receives the NAK end flag and writes 0 to clear it.	0
5	RB_UEP_R_NAK_TOG	RO	Endpoint n for the received return NAK, packet type: 1: DATA1; 0: DATA0.	0
4	RB_UEP_R_TOG_MATCH	RO	The synchronization trigger bit received by endpoint n matches the expected synchronization trigger bit: 1: Synchronized; 0: Not synchronized.	0
[3:2]	RB_UEP_R_TOG_MASK[1:0]	RW	Receiving expected synchronization trigger bit of endpoint n: 00: DATA0; 01: DATA1; 10: DATA2; 11: MDATA. *Note: This bit can be configured in manual mode and not in automatic mode.*	0
[1:0]	RB_UEP_R_RES_MASK[1:0]	RW	Endpoint n controls responses received: 00: Responds with NAK or busy; 01: Responds with STALL or error; 10: Responds with ACK; 11: Reserved.	0

18.2.1.36 Endpoint Transmit Synchronization Mode Enable Register (R16_UEP_T_ISO)

Offset address: 0xBC

Bit	Name	Access	Description	Reset value

[15:9]	RB_UEP_T_FIFO_EN [6:0]	RO	FIFO mode enable of Endpoint TX	0
8	Reserved	RO	Reserved	0
[7:1]	RB_UEPn_T_ISO_EN[6:0]	RW	The upload endpoint (IN) synchronization mode is enabled.	0
0	Reserved	RO	Reserved	0

18.2.1.37 USBHS Endpoint Receive Synchronization Mode Enable Register (R16_UEP_R_ISO)

Offset address: 0xBE

Bit	Name	Access	Description	Reset value
[15:9]	RB_UEP_R_FIFO_EN [6:0]	RO	The FIFO mode of the RX of the endpoint is enabled.	0
8	Reserved	RO	Reserved	0
[7:1]	RB_UEPn_R_ISO_EN [6:0]	RW	Downstream endpoint (OUT) synchronization mode is enabled.	0
0	Reserved	RO	Reserved	0

18.2.1.38 Received FIFO Address of USBHS Endpoint n (R32_UEPn_RX_FIFO) (n=1~7)

Offset address: 0xC0 + 4*(n-1)

Bit	Name	Access	Description	Reset value
[31:16]	RB_UEP_RX_FIFO_E	RW	FIFO end address of endpoint received.	0
[15:0]	RB_UEP_RX_FIFO_S	RW	FIFO start address of endpoint received.	0

18.2.1.39 Transmitted FIFO Address of USBHS Endpoint n (R32_UEPn_TX_FIFO) (n=1~7)

Offset address: 0xDC + 4*(n-1)

Bit	Name	Access	Description	Reset value
[31:16]	RB_UEP_TX_FIFO_E	RW	FIFO end address of endpoint transmitted.	0
[15:0]	RB_UEP_TX_FIFO_S	RW	FIFO start address of endpoint transmitted.	0

18.2.1.40 USBHS 16K Mode Control Register (R32_16K_MODE_CTRL)

Offset address: 0xF8

Bit	Name	Access	Description	Reset value
[31:8]	Reserved	RO	Reserved	0
[7:1]	RB_UEP_16KMODE_E N	RW	Endpoint 16K mode enable: 1: Enabled; 0: Disabled.	0
0	Reserved	RO	Reserved	0

18.2.1.41 USBHS 16K Mode Time Register (R32_16K_MODE_TIME)

Offset address: 0xFC

Bit	Name	Access	Description	Reset value

[31:24]	RB_TIMER_END	RW	Stop sending time windows.	0x4F
[23:16]	RB_TIMER_SEC	RW	Second delivery time window.	0x39
[15:8]	RB_TIMER_FST	RW	First delivery time window.	0x10
[7:0]	RB_TIMER_1US	RW	Basic timing of 1us.	0x64

18.2.2 Host Register Description

Table 18-2 Host-related registers

Name	Access address	Description	Reset value
R8_UH_CFG	0x40030100	USBHS host configuration register	0x07
R8_UH_INT_EN	0x40030102	USBHS host interrupt enable register	0x00
R8_UH_DEV_AD	0x40030103	USBHS host device address register	0x00
R32_UH_CONTROL	0x40030104	USBHS host control register	0x00XXXXXX
R8_UH_INT_FLAG	0x40030108	USBHS host interrupt flag register	0x00
R8_UH_INT_ST	0x40030109	USBHS host interrupt status register	0xXX
R8_UH_MIS_ST	0x4003010A	USBHS host miscellaneous status register	0xXX
R32_UH_LPM_DATA	0x4003010C	USBHS host power management data register	0x00000XXX
R32_UH_SPLIT_DATA	0x40030110	USBHS host SPLIT data register	0x000XXXXX
R32_UH_FRAME	0x40030114	USBHS host frame register	0x00000000
R32_UH_TX_LEN	0x40030118	USBHS host transmit length register	0x00000XXX
R32_UH_RX_LEN	0x4003011C	USBHS host receive length register	0x00000XXX
R32_UH_RX_MAX_LEN	0x40030120	USBHS host receive maximum length register	0x00000XXX
R32_UH_RX_DMA	0x40030124	DMA receive address register	0x000XXXXX
R32_UH_TX_DMA	0x40030128	DMA transmit address register	0x000XXXXX
R32_UH_PORT_CTRL	0x4003012C	USBHS host port control register	0x0000X000
R8_UH_PORT_CFG	0x40030130	USBHS host port configuration register	0x00
R8_UH_PORT_INT_EN	0x40030132	USBHS host port interrupt enable register	0x00
R8_UH_PORT_TEST_CT	0x40030133	USBHS host port test mode register	0x00
R16_UH_PORT_ST	0x40030134	USBHS host port status register	0x0010
R8_UH_PORT_CHG	0x40030136	USBHS host port status change register	0x00
R32_UH_BC_CTRL	0x4003013C	USBHS host BC charging control register	0x00000000

18.2.2.1 USBHS Host Control Register (R8_UH_CFG)

Offset address: 0x00

Bit	Name	Access	Description	Reset value
7	RB_UH_LPM_EN	RW	LPM enable.	0
6	RB_UH_FORCE_FS	RW	Forced to use USB_FS.	0
5	RB_UH_SOF_EN	RW	Enable SOF package transmitted.	0
4	RB_UH_DMA_EN	RW	Enable DMA transmitted.	0
3	RB_UH_PHY_SUSPENDM	RW	PHY suspended.	0
2	RB_UH_CLR_ALL	RW	Clearing the interrupt flag and FIFO requires software clearing.	1

1	RB_UH_RST_SIE	RW	The USB protocol processor is reset, which requires software clearing.	1
0	RB_UH_RST_LINK	RW	The USB connection control module is reset.	1

18.2.2.2 USBHS Host Interrupt Enable Register (R8_UH_INT_EN)

Offset address: 0x02

Bit	Name	Access	Description	Reset value
7	RB_UHIE_FIFO_OVER	RW	FIFO overflow interrupt enable	0
6	RB_UHIE_TX_HALT	RW	Transmit pause interrupt enable	0
5	RB_UHIE_SOF_ACT	RW	SOF packet transmission interrupt enable	0
4	RB_UHIE_TRANSFER	RW	USB end or transfer complete interrupt enable	0
3	RB_UHIE_RESUME_ACT	RW	Bus recovery interrupt enable	0
2	RB_UHIE_WKUP_ACT	RW	Wake-up interrupt enable	0
[1:0]	Reserved	RO	Reserved	0

18.2.2.3 USBHS Host Device Address Register (R8_UH_DEV_AD)

Offset address: 0x03

Bit	Name	Access	Description	Reset value
7	Reserved	RO	Reserved	0
[6:0]	RB_UH_DEV_ADDR[6:0]	RW	The address of the currently operating USB device.	0

18.2.2.4 USBHS Host Control Register (R32_UH_CONTROL)

Offset address: 0x04

Bit	Name	Access	Description	Reset value
[31:24]	Reserved	RO	Reserved	0
23	RB_UH_RX_NO_RES	RW	IN-DATA no response, used for synchronous transmission or high-speed SPLIT packets.	X
22	RB_UH_TX_NO_RES	RW	OUT/SETUP–DATA does not expect a response and is used for synchronous transmission or high-speed SPLIT packets.	X
21	RB_UH_RX_NO_DATA	RW	IN token packet followed by no expected data packet, used for high-speed SPLIT packets.	X
20	RB_UH_TX_NO_DATA	RW	No data packets follow the OUT/SETUP token packet, intended for high-speed SPLIT packets.	X

19	RB_UH_PRE_PID_EN	RW	Low-speed preamble packet PRE PID enable bit: When the port operates at full speed and requires transmission of low-speed packets (PRE), this bit must be enabled. 1: Enabled, used for communication with low-speed USB devices via an external full-speed HUB; 0: Disables low-speed preamble packets.	X
18	RB_UH_SPLIT_VALID	RW	Transmit SPLIT packet is valid.	X
17	RB_UH_LPM_VALID	RW	Transmit LPM packet is valid.	X
16	RB_UH_HOST_ACTION	RW	HOST performs transaction enabling, and this bit is cleared automatically after the transaction is completed.	0
[15:11]	Reserved	RO	Reserved	0
10	RB_UH_BUF_MODE	RW	Data cache control bit. 1: Use R32_UH_RX_DMA for transmission and R32_UH_TX_DMA for reception; 0: Use R32_UH_TX_DMA for transmission and R32_UH_RX_DMA for reception.	X
[9:8]	RB_UH_T_TOG_MASK[1:0]	RW	Transmit data PID: 00: PID_DATA0; 01: PID_DATA1; 10: PID_DATA2; 11: PID_MDATA。	X
[7:4]	RB_UH_T_ENDP_MASK[3:0]	RW	The endpoint number of the transaction token packet sent by the host.	X
[3:0]	RB_UH_T_TOKEN_MASK[3:0]	RW	The PID of the transaction token packet sent by the host.	X

18.2.2.5 USBHS Host Interrupt Flag Register (R8_UH_INT_FLAG)

Offset address: 0x08

Bit	Name	Access	Description	Reset value
7	RB_UHIF_FIFO_OVER	RW1Z	FIFO overflow interrupt flag.	0
6	RB_UHIF_TX_HALT	RW1Z	The transmission stop interrupt flag is used in the host mode for reading data overflow during downloading, or when the transmission is not finished at the time point of EOF2, the transmission is stopped, and the write 1 is cleared.	0
5	RB_UHIF_SOF_ACT	RW1Z	SOF packet transmission completion	0

			interrupt flag, write 1 to clear.	
4	RB_UHIF_TRANSFER	RW1Z	USB transaction transmission complete interrupt flag, write 1 to clear.	0
3	RB_UHIF_RESUME_ACT_IF	RW1Z	Bus recovery interrupt flag, write 1 to clear.	0
2	RB_UHIF_WKUP_ACT	RW1Z	Wake-up interrupt flag, write 1 to clear.	0
[1:0]	Reserved	RO	Reserved	0

18.2.2.6 USBHS Host Interrupt Status Register (R8_UH_INT_ST)

Offset address: 0x09

Bit	Name	Access	Description	Reset value
[7:5]	Reserved	RO	Reserved	X
4	RB_UHIS_PORT_RX_RESUME	RW	A bit of 1 indicates that the port has received a wake-up signal.	X
[3:0]	RB_UH_R_TOKEN_MASK [3:0]	RO	The received PID.	X

18.2.2.7 USBHS Host Miscellaneous Status Register (R8_UH_MIS_ST)

Offset address: 0x0A

Bit	Name	Access	Description	Reset value
7	RB_UHMS_BUS_SE0	RO	SE0 on USB bus.	X
6	RB_UHMS_BUS_J	RO	J on USB bus.	X
[5:4]	RB_UHMS_LINESTATE[1:0]	RO	PHY Linestate signal.	X
3	RB_UHMS_USB_WAKEUP	RO	USB bus wake status: 1: Bus wake in progress; 0: No bus wake in progress.	X
2	RB_UHMS_SOF_ACT	RO	USB bus SOF packet transmission status bit: 1: SOF packet being transmitted; 0: Transmission complete or idle.	X
1	RB_UHMS_SOF_PRE	RO	The USB bus SOF packet status indicates: 1: An SOF packet is about to be sent; 0: No SOF packet is being sent.	X
0	RB_UHMS_SOF_FREE	RO	Port enable status: 1: Port enabled; 0: Port disabled.	X

18.2.2.8 USBHS Host Power Management Data Register (R32_UH_LPM_DATA)

Offset address: 0x0C

Bit	Name	Access	Description	Reset value

[31:11]	Reserved	RO	Reserved	0
[10:0]	RB_UH_LPM_DATA[10:0]	RW	Link power management packet data content.	X

18.2.2.9 USBHS Host SPLIT Register (R32_UH_SPLIT_DATA)

Offset address: 0x10

Bit	Name	Access	Description	Reset value
[31:19]	Reserved	RO	Reserved	0
[18:0]	RB_UH_SPLIT_DATA[18:0]	RW	The data content of the SPLIT packet sent by the host endpoint, from high to low bits, is: ET, E, S, Port, SC, Hub Addr.	X

18.2.2.10 USBHS Host Frame Register (R32_UH_FRAME)

Offset address: 0x14

Bit	Name	Access	Description	Reset value
[31:26]	Reserved	RO	Reserved	0
25	RB_UH_SOF_CNT_CLR	RW	SOF count clear.	0
24	RB_UH_SOF_CNT_EN	RW	SOF count enable.	0
[23:19]	Reserved	RO	Reserved	0
[18:16]	RB_UH_MFRAME_NO[2:0]	RO	The micro-frame number of the upcoming SOF packet.	0
[15:11]	Reserved	RO	Reserved	0
[10:0]	RB_UH_FRAME_NO[10:0]	RW	The frame number of the upcoming SOF packet.	0

18.2.2.11 USBHS Host Transmit Length Register (R32_UH_TX_LEN)

Offset address: 0x18

Bit	Name	Access	Description	Reset value
[31:11]	Reserved	RO	Reserved	0
[10:0]	RB_UH_TX_LEN[10:0]	RW	Data transmission length.	X

18.2.2.12 USBHS Host Receive Length Register (R32_UH_RX_LEN)

Offset address: 0x1C

Bit	Name	Access	Description	Reset value
[31:11]	Reserved	RO	Reserved	0
[10:0]	RB_UH_RX_LEN[10:0]	RW	Received data length.	X

18.2.2.13 USBHS Host Receive Maximum Length Register (R32_UH_RX_MAX_LEN)

Offset address: 0x20

Bit	Name	Access	Description	Reset value
[31:11]	Reserved	RO	Reserved	0
[10:0]	RB_UH_RX_MAX_LEN[10:0]	RW	Maximum length of reception.	X

18.2.2.14 DMA Receive Address Register (R32_UH_RX_DMA)

Offset address: 0x24

Bit	Name	Access	Description	Reset value
[31:24]	Reserved	RO	Reserved	0
[23:0]	R32_UH_RX_DMA[23:0]	RW	Receiving address, the lower two digits are invalid, and 4 bytes alignment is required.	X

18.2.2.15 DMA Transmit Address Register (R32_UH_TX_DMA)

Offset address: 0x28

Bit	Name	Access	Description	Reset value
[31:24]	Reserved	RO	Reserved	0
[23:0]	R32_UH_TX_DMA[23:0]	RW	Transmitting address, the lower two digits are invalid, and 4 bytes alignment is required.	X

18.2.2.16 USBHS Host Port Control Register (R32_UH_PORT_CTRL)

Offset address: 0x2C

Bit	Name	Access	Description	Reset value
[31:17]	Reserved	RO	Reserved	0
16	RB_UH_BUS_RST_LONG	RW	Bus reset time selection: 1: Extended reset; 0: Normal reset.	0
[15:12]	RB_UH_PORT_SLEEP_BESL[3:0]	RW	Wake-up time control: 0000: 125us; 1000: 3ms; 0001: 150us; 1001: 4ms; 0010: 200us; 1010: 5ms; 0011: 300us; 1011: 6ms; 0100: 400us; 1100: 7ms; 0101: 500us; 1101: 8ms; 0110: 1ms; 1110: 9ms; 0111: 2ms; 1111: 10ms。	X
[11:9]	Reserved	RO	Reserved	0
8	RB_UH_CLR_PORT_SLEEP	WO	PORT exit SLEEP status (LPM).	0
[7:6]	Reserved	RO	Reserved	0
5	RB_UH_CLR_PORT_C	WO	Put the port into the port status.	0

	ONNECT			
4	RB_UH_CLR_PORT_EN	WO	PORT exits the enabled state and enters the DISABLED status.	0
3	RB_UH_SET_PORT_SLEEP	WO	PORT enter SLEEP status (LPM).	0
2	RB_UH_CLR_PORT_SUSP	WO	PORT exits the suspended status.	0
1	RB_UH_SET_PORT_SUSP	WO	PORT enters the suspended status.	0
0	RB_UH_SET_PORT_RESET	WO	Port transmit reset.	0

18.2.2.17 USBHS Host Port Configuration Register (R8_UH_PORT_CFG)

Offset address: 0x30

Bit	Name	Access	Description	Reset value
7	RB_UH_PD_EN	RW	15K resistor pull-down is enabled in host mode.	0
[6:1]	Reserved	RO	Reserved	0
0	RB_UH_HOST_EN	RW	USB port mode selection: 1: Port operates in host mode; 0: Port operates in device mode.	0

18.2.2.18 USBHS Host Port Interrupt Enable Register (R8_UH_PORT_INT_EN)

Offset address: 0x32

Bit	Name	Access	Description	Reset value
[7:6]	Reserved	RO	Reserved	0
5	RB_UHIE_PORT_SLP	RW	Port sleep state change interrupt enabled.	0
4	RB_UHIE_PORT_RESET	RW	Port reset state change interrupt enabled.	0
3	Reserved	RO	Reserved	0
2	RB_UHIE_PORT_SUSP	RW	Port suspended state change interrupt enabled.	0
1	RB_UHIE_PORT_EN	RW	Port enable state change interrupt enabled.	0
0	RB_UHIE_PORT_CONNECT	RW	Port connection state change interrupt enabled.	0

18.2.2.19 USBHS Host Port Test Mode Register (R8_UH_PORT_TEST_CT)

Offset address: 0x33

Bit	Name	Access	Description	Reset value
[7:5]	Reserved	RO	Reserved	0
4	RB_UH_TEST_SE0_NAK	RW	Test TEST_SE0_NAK.	0

3	RB_UH_TEST_PACKET	RW	Test TEST_PKT.	0
2	RB_UH_TEST_FORCE_EN	RW	Test mode enabled.	0
1	RB_UH_TEST_K	RW	Test output K.	0
0	RB_UH_TEST_J	RW	Test output J.	0

18.2.2.20 USBHS Host Port Status Register (R16_UH_PORT_ST)

Offset address: 0x34

Bit	Name	Access	Description	Reset value
[15:12]	Reserved	RO	Reserved	0
11	RB_UHIS_PORT_TEST	RO	Whether the port is in test mode.	0
10	RB_UHIS_PORT_HS	RO	Whether the port connection speed is high.	0
9	RB_UHIS_PORT_LS	RO	Whether the port connection speed is low.	0
[8:6]	Reserved	RO	Reserved	0
5	RB_UHIS_PORT_SLP	RO	Port sleep.	0
4	RB_UHIS_PORT_RST	RO	Port reset status.	1
3	Reserved	RO	Reserved	0
2	RB_UHIS_PORT_SUSP	RO	Port suspended status.	0
1	RB_UHIS_PORT_EN	RO	Port enabled.	0
0	RB_UHIS_PORT_CONNECT	RO	Port connection status.	0

18.2.2.21 USBHS Host Port Status Change Register (R8_UH_PORT_CHG)

Offset address: 0x36

Bit	Name	Access	Description	Reset value
[7:6]	Reserved	RO	Reserved	0
5	RB_UHIF_PORT_SLP	RW1Z	Port sleep state changes, write 1 to clear.	0
4	RB_UHIF_PORT_RESET	RW1Z	Port reset state changes, write 1 to clear.	0
3	Reserved	RO	Reserved	0
2	RB_UHIF_PORT_SUSP	RW1Z	Port suspend state changes, write 1 to clear.	0
1	RB_UHIF_PORT_EN	RW1Z	Port enable state changes, write 1 to clear.	0
0	RB_UHIF_PORT_CONNECT	RW1Z	Port connection status changes, write 1 to clear.	0

18.2.2.22 USBHS Host BC Charging Control Register (R32_UH_BC_CTRL)

Offset address: 0x3C

Bit	Name	Access	Description	Reset value
[31:11]	Reserved	RO	Reserved	0
10	RB_UDM_VSRC_A	RO	In automatic mode, this bit indicates UDM	0

	CT		output V_{BC_SRC}; otherwise, it is controlled by RB_UDM_BC_CMPE. *Note: This bit is set to 1 when UDP is higher than V_{BC_SRC} but UDP is not high; otherwise, it is 0.*	
9	RB_UDM_BC_VSRC	RW	UDM pin BC protocol source voltage enable: 1: UDM pin outputs BC protocol source voltage V_{BC_SRC}; 0: Output disabled.	0
8	RB_UDP_BC_VSRC	RW	UDP Pin BC Protocol Source Voltage Enable: 1: Outputs BC protocol source voltage V_{BC_SRC} via UDP pin; 0: Disables output.	0
7	Reserved	RO	Reserved	0
6	RB_BC_AUTO_MODE	RW	Auto mode enable: 1: Enabled; 0: Disabled.	0
5	RB_UDM_BC_CMPE	RW	UDM pin BC protocol comparator enable: 1: Enable; 0: Disable.	0
4	RB_UDP_BC_CMPE	RW	UDP pin BC protocol comparator enable: 1: Enable; 0: Disable.	0
[3:2]	Reserved	RO	Reserved	0
1	RB_UDM_BC_CMPO	RO	UDM pin BC protocol comparator state: 1: UDM pin voltage is higher than the BC protocol reference value V_{BC_REF}; 0: UDM pin voltage is lower than the BC protocol reference value V_{BC_REF}.	0
0	RB_UDP_BC_CMPO	RO	UDP pin BC protocol comparator state: 1: UDP pin voltage is higher than the BC protocol reference value V_{BC_REF}; 0: UDP pin voltage is lower than the BC protocol reference value V_{BC_REF}.	0

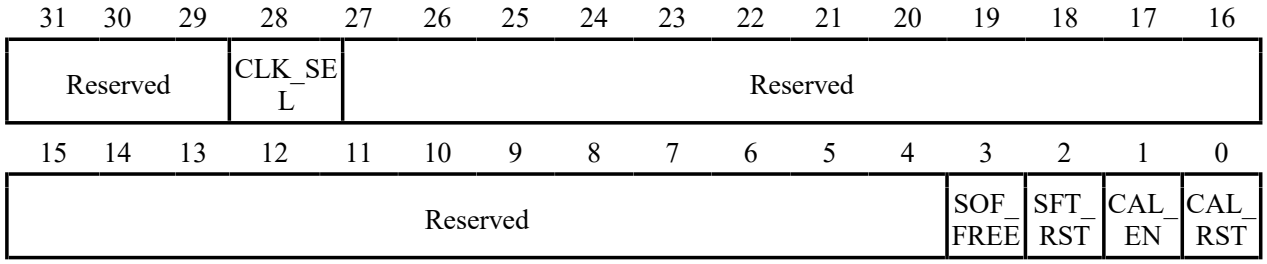
18.2.3 Host Register Description

Table 18-3 Host-related registers

Bit	Name	Access	Description
R32_HSI_CAL_CR	0x40030200	HSI calibration control register	0x0021C045

18.2.3.1 HSI Calibration Control Register (HSI_CAL_CR)

Offset address: 0x00



Bit	Name	Access	Description	Reset value
[31:29]	Reserved	RO	Reserved	0
28	CLK_SEL	RW	Clock selection: 1: 60MHz/8; 0: 60MHz.	0
[27:4]	Reserved	RO	Reserved	0x21C04
3	SOF_FREE	RW	The upstream interface enters enable mode: 1: Enabled; 0: Disabled.	0
2	SFT_RST	RW	Exit fine calibration mode, receive BUS_RESET, this bit is set, and software clears it.	1
1	CAL_EN	RW	Calibration enable: 1: Enable; 0: Disable.	0
0	CAL_RST	RW	Calibration module global reset.	1

Chapter 19 USB3.0 SuperSpeed Controller (USBSS)

The module description in this chapter is only applicable to CH32X315 microcontroller products.

The USB3.0 SuperSpeed controller has built-in device controller and SuperSpeed USB PHY physical layer transceiver, enabling USB3.0 interface function and supporting 5Gbps USB SuperSpeed signals.

The USBSS module provides application code with a link-layer register access interface for managing device connection/disconnection, bus status, and power modes. It offers separate device (DEVICE) function access interfaces to implement various data transfers and upper-layer protocols defined by the USB3.0 specification.

USBSS module peripheral base address: 0x40038000

19.1 Main Features

- Support USB3.0 protocol specifications and USB3.2 Gen1
- Support USB Device function
- Power management modes support U1/U2/U3 low-power states
- Support control transfers, bulk transfers, interrupt transfers, real-time/synchronous transfers
- All non-zero endpoints support maximum 1024-byte packets with burst mode capability
- Support direct memory access (DMA) to endpoint buffer data
- Proprietary controller and transceiver with high-speed integrated design, tested at 450Mbytes/s

19.2 Register Description

The USBSS registers are divided into 3 parts:

- USBSS global registers
- USBSS device control registers
- USBSS LINK control registers

19.2.1 Global Register Description

Table 19-1 USBSS-related registers

Name	Access address	Description	Reset value
R32_USBSS_CTRL	0x40038070	USBSS control register	0x00000006
R32_USBSS_STATUS	0x40038074	USBSS status register	0x00000000
R32_UH_TX_DMA R32_UEP0_TX_DMA	0x4003808C	Transmit buffer address register	0xXXXXXXXX
R32_UH_RX_DMA R32_UEP0_RX_DMA	0x40038090	Receive buffer address register	0xXXXXXXXX

19.2.1.1 USBSS Control Register (R32_USBSS_CTRL)

Offset address: 0x70

Bit	Name	Access	Description	Reset value
31	Reserved	RO	Reserved	0

[30:24]	RB_DEV_ADDR	RW	The address of this USB device.	0
23	RB_UIE_FIFO_RX OV	RW	Receive FIFO overflow interrupt enable: 1: Enabled; 0: Disabled.	0
22	RB_UIE_FIFO_TX OV	RW	Transmit FIFO overflow interrupt enable: 1: Enabled; 0: Disabled.	0
21	Reserved	RO	Reserved	0
20	RB_UIE_ITP	RW	Transmit ITP complete interrupt enable: 1: Enabled; 0: Disabled.	0
19	RB_UIE_RX_PING	RW	Receive PING-TP complete interrupt enable: 1: Enabled; 0: Disabled.	0
18	RB_UDIE_STATUS	RW	Device mode: Enable interrupt upon completion of STATUS transaction: 1: Enabled; 0: Disabled.	0
18	RB_UHIE_NOTIF	RW	Host mode: Enable interrupt upon completion of DEV_NOTIF-TP transaction: 1: Enabled; 0: Disabled.	0
17	RB_UDIE_SETUP	RW	Device mode: Enable interrupt upon completion of SETUP transaction: 1: Enabled; 0: Disabled.	0
17	RB_UHIE_ERDY	RW	Host mode: Enable interrupt upon completion of SETUP transaction: 1: Enabled; 0: Disabled.	0
16	RB_UIE_TRANSF ER	RW	USB transaction completion interrupt enable: 1: Enabled; 0: Disabled.	0
15	Reserved	RO	Reserved	0
14	RB_TX_ERDY_M ODE	RW	Usage in device mode: 1: Prohibit ERDY-TP insertion between two DPH packets; 0: Allow ERDY-TP insertion between two DPH packets.	0
[13:6]	Reserved	RO	Reserved	0
5	RB_SETUP_FLOW	RW	SETUP flow control 1: Upon receiving a SETUP transaction, enter flow control; device required. 0: Upon receiving a SETUP transaction, flow	0

			control is not required.	
4	Reserved	RO	Reserved	0
3	RB_DMA_MODE	RW	1: During burst transmission DPH, do not prefetch the next packet into the FIFO; 0: During burst transmission DPH, prefetch the next packet into the FIFO. *Note: Enabling prefetching minimizes the interval between burst DPH packets, increasing transmission bandwidth. Prefetching is enabled by default.*	0
2	RB_FORCE_RST	RW	Protocol layer and FIFO module reset requires software clearing: 1: Reset; 0: Do not reset.	1
1	RB_USB_CLR_ALL	RW	Reset all software configuration registers, high-order bit is valid, requires software clearing; Clear all interrupt flags, clear device addresses, clear all endpoint configurations. *Note: In device mode, this bit should be set after receiving HOT_RESET/WARM_RESET, then cleared.*	1
0	RB_DMA_EN	RW	Enable DMA: 1: Enabled, the controller can access data in SRAM; 0: Disabled, the controller cannot access data in SRAM.	0

19.2.1.2 USBSS Status Register (R32_USBSS_STATUS)

Offset address: 0x74

Bit	Name	Access	Description	Reset value
[31:13]	Reserved	RO	Reserved	0
12	RB_EP_DIR	RW	At present, the direction of the endpoint with the transmission completion interrupt flag. If there is an interrupt flag at the same time, then this bit is 1, that is, the transmission is given priority. (SETUP/STATUS independent interrupt) 1: Transmit (IN); 0: Receive (OUT).	0
11	Reserved	RO	Reserved	0
[10:8]	RB_EP_ID	RW	The endpoint number currently has a transmission completion interrupt flag. If multiple endpoints simultaneously have	0

			interrupt flags, the priority order for endpoint types is as follows (highest to lowest): synchronous transmission > control transmission > block transmission/interrupt transmission. Within the same endpoint type, lower endpoint numbers have higher priority.	
7	RB_UIF_FIFO_RXOV	RW1Z	Receive FIFO overflow interrupt flag: 1: Interrupt enabled; 0: Interrupt disabled.	0
6	RB_UIF_FIFO_TXOV	RW1Z	Transmit FIFO overflow interrupt flag: 1: Interrupt enabled; 0: Interrupt disabled.	0
5	Reserved	RO	Reserved	0
4	RB_UIF_ITP	RW1Z	Transmit ITP completed interrupt flag: 1: Interrupt enabled; 0: Interrupt disabled.	0
3	RB_UIF_RX_PING	RW1Z	Receive PING-TP completion interrupt flag: 1: Interrupt enabled; 0: Interrupt disabled.	0
2	RB_UDIF_STATUS	RW1Z	Device mode: Receive STATUS transaction completion interrupt flag: 1: Interrupt enabled; 0: Interrupt disabled.	0
1	RB_UDIF_SETUP	RW1Z	Device mode: Receive SETUP transaction completion interrupt flag: 1: Interrupt enabled; 0: Interrupt disabled.	0
0	RB_UIF_TRANSFER	RW1Z	USB transaction completion interrupt flag: 1: Interrupt enabled; 0: Interrupt disabled.	0

19.2.1.3 Transmit Buffer Address Register (R32_UH_TX_DMA/U3EP0_TX_DMA)

Offset address: 0x8C

Bit	Name	Access	Description	Reset value
[31:0]	U3EP0_TX_DMA	RW	Endpoint 0 transmits buffer start address.	X

19.2.1.4 Receive Buffer Address Register (R32_UH_RX_DMA/U3EP0_RX_DMA)

Offset address: 0x90

Bit	Name	Access	Description	Reset value
[31:0]	U3EP0_RX_DMA	RW	Start address of endpoint 0 receive buffer.	X

19.2.2 Device Register Description

Table 19-2 USBSS device registers

Name	Access address	Description	Reset value
R32_USBSS_ITP	0x40038078	Interval register in the ITP package of USB	0x00000000
R32_USBSS_ITP_ADJ	0x4003807C	Interval adaptive register in the ITP package of USB	0x00000000
R16_UEP_TX_EN	0x40038080	Endpoint transmit enable register	0x00000000
R16_UEP_RX_EN	0x40038082	Endpoint receive enable register	0x00000000
R32_UEP0_TX_CTRL	0x40038084	Endpoint 0 transmit control register	0x00000XXX
R32_UEP0_RX_CTRL	0x40038088	Endpoint 0 receive control register.	0x00000XXX

19.2.2.1 Interval Register in the ITP Package of USB (32_USBSS_ITP)

Offset address: 0x78

Bit	Name	Access	Description	Reset value
[31:14]	Reserved	RO	Reserved	0
[13:0]	REG_ITP_INTERVAL	RO	Receive Bus Interval Counter field in the ITP. *Note: Bus Interval Counter increments every 125us.*	0

19.2.2.2 Interval Adaptive register in the ITP package of USB (R32_USBSS_ITP_ADJ)

Offset address: 0x7C

Bit	Name	Access	Description	Reset value
[31:21]	Reserved	RO	Reserved	0
[20:8]	ITP_DELTA	RO	The high 13 bits (Delta) of ITS in the ITP received in device mode represent the time difference from the beginning of the current ITP packet to the previous bus interval boundary.	0
7	ITP_DELAYED	RO	In device mode, the Delayed bit in the Link Control Word of the received ITP is set to 1 when the ITP is transmitted with delay.	0
[6:0]	ITP_ADJ_CR	RO	In device mode, receiving the Bus Interval Adjustment Control field in the ITP.	0

19.2.2.3 Endpoint Transmit Enable Register (R16_UEP_TX_EN)

Offset address: 0x80

Bit	Name	Access	Description	Reset value
[15:1]	RB_EP_TX_EN	RW	Endpoints 1–15 upload enable: 1: Enable; 0: Disable.	0
0	Reserved	RO	Reserved	0

19.2.2.4 Endpoint Receive Enable Register (R16_UEP_RX_EN)

Offset address: 0x82

Bit	Name	Access	Description	Reset value
[15:1]	RB_EP_RX_EN	RW	Endpoints 1–15 download enable: 1: Enable; 0: Disable.	0
0	Reserved	RO	Reserved	0

19.2.2.5 Endpoint 0 Transmit Control Register (R32_UEP0_TX_CTRL)

Offset address: 0x84

Bit	Name	Access	Description	Reset value
31	RB_UIF_EP0_TX_ACT	RW	Upload the transaction completion interrupt flag, the software writes 0 to clear it, and the hardware sets it to 1.	0
[20:26]	Reserved	RO	Reserved	0
25	RB_EP0_TX_FLOW	RO	Sign to complete NRDY-TP transmission (response).	0
24	RB_EP0_TX_PP	RO	PP bit in the received ACK-TP.	0
23	RB_EP0_TX_ERDY	RW	Write 1 to this bit to send ERDY. After ERDY is sent, this bit is automatically cleared by hardware.	0
[22:21]	RB_EP0_TX_RES[1:0]	RW	Response to ACK-TP (nump ≠ 0): 00: NRDY-TP; 01: DPH; 10: STALL-TP; 11: No response (invalid). Automatically clears after IN transaction completion.	0
[20:16]	RB_EP0_TX_SEQ[4:0]	RW	The current serial number of the endpoint, which is automatically cleared after receiving the SETUP transaction.	0
[15:11]	Reserved	RO	Reserved	0
[10:0]	RB_EP0_TX_LEN[10:0]	RW	Endpoint send length register, with a maximum value of 512B.	X

19.2.2.6 Endpoint 0 Receive Control Register (R32_UEP0_RX_CTRL)

Offset address: 0x88

Bit	Name	Access	Description	Reset value
31	RB_UIF_EP0_RX_ACT	RW	Downloading transaction completion interrupt flag, which is cleared by software writing 0 and set by hardware.	0
[30:25]	Reserved	RO	Reserved	0

24	RB_EP0_RX_PP	RO	PP bit in the received DPH.	0
23	RB_EP0_RX_ERDY	RW	Write 1 to this bit to send ERDY. After ERDY is sent, this bit is automatically cleared by hardware.	0
[22:21]	RB_EP0_RX_RES[1:0]	RW	Response to DPH: 00: NRDY-TP; 01: ACK-TP; 10: STALL-TP; 11: No response (invalid). EPn_R_RES automatically clears after SETUP/OUT transaction completion; additionally, the SETUP packet directly responds with ACK-TP, which is unrelated to this.	0
[20:16]	RB_EP0_RX_SEQ[4:0]	RW	The sequence number that the endpoint expects to receive.	0
[15:11]	Reserved	RO	Reserved	0
[10:0]	RB_EP0_RX_LEN[10:0]	RO	Endpoint receive length register.	X

19.2.3 Device Register-Endpoints 1 ~ 4 Transmit-related Registers.

Table 19-3 Endpoint 1 transmit-related registers

Name	Access address	Description	Reset value
R8_UEP1_TX_CFG	0x400380C0	Endpoint 1 configuration register	0x86
R8_UEP1_TX_CR	0x400380C1	Endpoint 1 control register	0x10
R8_UEP1_TX_SEQ	0x400380C2	Endpoint 1 serial number register	0x00
R8_UEP1_TX_ST	0x400380C3	Endpoint 1 status register	0x00
R8_UEP1_TX_CHAIN_CR	0x400380C4	Endpoint 1 CHAIN control register	0x00
R8_UEP1_TX_CHAIN_ST	0x400380C5	Endpoint 1 CHAIN status register	0x00
R16_UEP1_TX_CHAIN_LEN	0x400380C6	Endpoint 1 CHAIN transmit last packet length	0x0000
R8_UEP1_TX_CHAIN_EXP_NUMP	0x400380C8	Endpoint 1 expects to transmit NUMP number	0x00
R8_UEP1_TX_CHAIN_NUMP	0x400380C9	Endpoint 1 has transmitted NUMP number	0x00
R16_UEP1_TX_DMA_OFS	0x400380CA	DMA offset length of endpoint 1	0x0400
R32_UEP1_TX_DMA	0x400380CC	DMA start address of endpoint 1	0x00000000

Table 19-4 Endpoint 2 transmit-related registers

Name	Access address	Description	Reset value
R8_UEP2_TX_CFG	0x400380E0	Endpoint 2 configuration register	0x86
R8_UEP2_TX_CR	0x400380E1	Endpoint 2 control register	0x10

R8_UEP2_TX_SEQ	0x400380E2	Endpoint 2 serial number register	0x00
R8_UEP2_TX_ST	0x400380E3	Endpoint 2 status register	0x00
R8_UEP2_TX_CHAIN_CR	0x400380E4	Endpoint 2 CHAIN control register	0x00
R8_UEP2_TX_CHAIN_ST	0x400380E5	Endpoint 2 CHAIN status register	0x00
R16_UEP2_TX_CHAIN_LEN	0x400380E6	Endpoint 2 CHAIN transmit last packet length	0x0000
R8_UEP2_TX_CHAIN_EXP_NUMP	0x400380E8	Endpoint 2 expects to send NUMP number	0x00
R8_UEP2_TX_CHAIN_NUMP	0x400380E9	Endpoint 2 has transmitted NUMP number	0x00
R16_UEP2_TX_DMA_OFS	0x400380EA	DMA offset length of endpoint 2	0x0400
R32_UEP2_TX_DMA	0x400380EC	DMA start address of endpoint 2	0x00000000

Table 19-5 Endpoint 3 transmit-related registers

Name	Access address	Description	Reset value
R8_UEP3_TX_CFG	0x40038100	Endpoint 3 configuration register	0x86
R8_UEP3_TX_CR	0x40038101	Endpoint 3 control register	0x10
R8_UEP3_TX_SEQ	0x40038102	Endpoint 3 serial number register	0x00
R8_UEP3_TX_ST	0x40038103	Endpoint 3 status register	0x00
R8_UEP3_TX_CHAIN_CR	0x40038104	Endpoint 3 CHAIN control register	0x00
R8_UEP3_TX_CHAIN_ST	0x40038105	Endpoint 3 CHAIN status register	0x00
R16_UEP3_TX_CHAIN_LEN	0x40038106	Endpoint 3 CHAIN transmit last packet length	0x0000
R8_UEP3_TX_CHAIN_EXP_NUMP	0x40038108	Endpoint 3 expects to send NUMP number	0x00
R8_UEP3_TX_CHAIN_NUMP	0x40038109	Endpoint 3 has transmitted NUMP number	0x00
R16_UEP3_TX_DMA_OFS	0x4003810A	DMA offset length of endpoint 3	0x0400
R32_UEP3_TX_DMA	0x4003810C	DMA start address of endpoint 3	0x00000000

Table 19-6 Endpoint 4 receives-related registers

Name	Access address	Description	Reset value
R8_UEP4_TX_CFG	0x40038120	Endpoint 4 configuration register	0x86
R8_UEP4_TX_CR	0x40038121	Endpoint 4 control register	0x10
R8_UEP4_TX_SEQ	0x40038122	Endpoint 4 serial number register	0x00
R8_UEP4_TX_ST	0x40038123	Endpoint 4 status register	0x00
R8_UEP4_TX_CHAIN_CR	0x40038124	Endpoint 4 CHAIN control register	0x00
R8_UEP4_TX_CHAIN_ST	0x40038125	Endpoint 4 CHAIN status register	0x00
R16_UEP4_TX_CHAIN_LEN	0x40038126	Endpoint 4 CHAIN transmit last packet length	0x0000
R8_UEP4_TX_CHAIN_EXP_NUMP	0x40038128	Endpoint 4 expects to send NUMP number	0x00

R8_UEP4_TX_CHAIN_NUMP	0x40038129	Endpoint 4 has transmitted NUMP number	0x00
R16_UEP4_TX_DMA_OFS	0x4003812A	DMA offset length of endpoint 4	0x0400
R32_UEP4_TX_DMA	0x4003812C	DMA start address of endpoint 4	0x00000000

19.2.3.1 Endpoint n Configuration Register (R8_UEPn_TX_CFG) (n=1~4)

Offset address: 0xC0 + (n-1)*0x20

Bit	Name	Access	Description	Reset value
7	RB_EP_TX_CHAIN_AUTO	RW	CHAIN automatically switches mode, which is recommended.	1
6	RB_EP_TX_FIFO_MODE	RW	A bit of 1 indicates that the current endpoint uses FIFO mode. Before this bit is 1, the start address and end address of FIFO should be configured first.	0
5	RB_EP_TX_FIFO_CFG	RW	1: Access offset addresses 0xC to 0xF, with the operation target being UEP_DMA[31:16]/UEP_DMA[15:0]; 0: Access offset addresses 0xC to 0xF, with the operation target being UEP_DMA.	0
4	Reserved	RO	Reserved	0
3	RB_EP_TX_EOB_MODE	RW	1: When sending a short packet, EOB/LPF in ACK-TP is 0; 0: When sending a short packet, EOB/LPF in ACK-TP is 1; *Note: The EOB_LPF of CHAIN_ST is ORed with the EOB/LPF set in this mode; this bit applies only to short packets, whereas CHAIN_ST->EOB_LPF applies to the last packet of the CHAIN and is independent of packet length.*	0
2	RB_EP_TX_ERDY_AUTO	RW	ERDY automatic mode, hardware will send ERDY without software control; The synchronization endpoint recommends clearing this bit.	1
1	RB_EP_TX_SEQ_AUTO	RW	1: Software is prohibited from writing to R16_EPn_ST->EP_SEQ_NUM; 0: Software is permitted to write to R16_EPn_ST->EP_SEQ_NUM.	1
0	RB_EP_TX_ISO_MODE	RW	A value of 1 indicates that the current endpoint is a synchronous endpoint.	0

19.2.3.2 Endpoint n Control Register (R8_UEPn_TX_CR) (n=1~4)

Offset address: 0xC1 + (n-1)*0x20

Bit	Name	Access	Description	Reset value
7	RB_EP_TX_HALT	RW	Endpoint stop, active high, endpoint stop responding to DPH STALL.	0
6	RB_EP_TX_CLR	RW1	Write 1 to clear all configuration values and states of endpoints except UEP_CFG.	0
5	RB_EP_TX_CHAIN_CLR	RW1	Write 1 to clear all CHAIN configuration values and status.	0
[4:0]	RB_EP_TX_ERDY_NUMP	RW	The hardware sends the NUMP field of ERDY-TP, and the general configuration value is the number of supported bursts; For example, it supports 16-level burst, and the configuration value of this field is 16.	0x10

19.2.3.3 Endpoint n Serial Number Register (R8_UEPn_TX_SEQ) (n=1~4)

Offset address: 0xC2 + (n-1)*0x20

Bit	Name	Access	Description	Reset value
[7:5]	Reserved	RO	Reserved	0
[4:0]	RB_TX_EP_SEQ_NUM	RW	Current serial number of endpoint, writable in non-SEQ_AUTO mode and read-only in SEQ_AUTO mode.	0

19.2.3.4 Endpoint n Status Register (R8_UEPn_TX_ST) (n=1~4)

Offset address: 0xC3 + (n-1)*0x20

Bit	Name	Access	Description	Reset value
7	RB_TX_EP_INT_FLAG	RO	Endpoint current interrupt flag. This bit is read-only. If all CHAIN_IF are 0, this bit is 0, otherwise it is 1.	0
6	RB_TX_EP_FC_ST	RW1Z	Current flow control status of endpoint, write 1 to clear.	0
5	RB_TX_EP_ERDY_REQ	RW1	Indicates that it is currently in the sending ERDY; Writing a 1 to this bit will send ERDY for use in non-ERDY_AUTO mode.	0
4	RB_TX_CHAIN_RES	RO	CHAIN response status, corresponding to 4 independent CHAINs: 1: Responds with ACK_TP; 0: Responds with NRDY-TP. *Note: In endpoint halt mode (EP_HALT=1), responds with STALL; this bit is inactive.*	0
[3:0]	RB_TX_CHAIN_EN	RO	CHAIN enabled state, corresponding to 4 independent CHAIN.	0

19.2.3.5 Endpoint n CHAIN Control Register (R8_UEPn_TX_CHAIN_CR) (n=1~4)

Offset address: 0xC4 + (n-1)*0x20

Bit	Name	Access	Description	Reset value
[7:6]	RB_TX_CUR_USE	RO	CHAIN serial number currently in use.	0
[5:4]	RB_TX_CUR_CFG	RO	CHAIN serial number currently configured.	0
3	Reserved	RO	Reserved	0
2	RB_TX_FORCE_RET	RW	This bit effectively forces the return of the selected CHAIN state machine configuration.	0
[1:0]	RB_TX_RET_SEL	RW	When FORCE_RET is valid, this bit indicates the returned CHAIN status and configuration.	0

19.2.3.6 Endpoint n CHAIN Status Register (R8_UEPn_TX_CHAIN_ST) (n=1~4)

Offset address: 0xC5 + (n-1)*0x20

Bit	Name	Access	Description	Reset value
7	RB_TX_CHAIN_EN	RO	When the previously enabled CHAIN is enabled, it is automatically set by hardware after configuring the UEP_CHAIN_NUMP register, and the CHAIN is automatically cleared after the transmission.	0
6	RB_TX_CHAIN_IF	WO	This bit is written only to release the current CHAIN_IF; The current CHAIN's NUMP_EMPTY being 1 generates this interrupt.	0
5	RB_TX_EOB_LPF	RW	Current EOB/LPF bit in the last DPH packet of CHAIN: For lot transmission, when this bit is set to 1, CHAIN transmission ends and enters flow control (End Of Block), preventing it from being linked with the next CHAIN for transmission. For synchronous transmission, when this bit is set to 1, CHAIN transmission ends, and the microframe does not send another DPH, indicating this is the last packet for this microframe (LAST PACKET FLAG).	0
4	Reserved	RO	Reserved	0
3	RB_TX_NUMP_EMPTY	RO	NUMP in the current CHAIN is 0, then this bit is set to 1.	0
2	RB_TX_DPH_PP	RW	Status of the PP bit in the currently received DPH.	0
[1:0]	RB_TX_CHAIN_NO	RW	CHAIN sequence number that currently generates the interrupt.	0

19.2.3.7 Endpoint n CHAIN Transmit Packet Length (R16_UEPn_TX_CHAIN_LEN) (n=1~4)

Offset address: 0xC6 + (n-1)*0x20

Bit	Name	Access	Description	Reset value
[15:11]	Reserved	RO	Reserved	0
[10:0]	CHAIN_TX_LEN	RW	The length of the last packet sent by the CHAIN.	0

19.2.3.8 Endpoint n Expects to Transmit NUMP Number (R8_UEPn_TX_CHAIN_EXP_NUMP) (n=1~4)

Offset address: 0xC8 + (n-1)*0x20

Bit	Name	Access	Description	Reset value
[7:0]	TX_CHAIN_EXP_NUMP	RW	The number of DPP packets that can be sent by the currently completed CHAIN.	0

19.2.3.9 Endpoint n has Transmitted NUMP Number (R8_UEPn_TX_CHAIN_NUMP) (n=1~4)

Offset address: 0xC9 + (n-1)*0x20

Bit	Name	Access	Description	Reset value
[7:0]	TX_CHAIN_NUMP	RW	Number of DPP packets currently transmitted; when CHIN_EN is 0, writing UEP_CHAIN_NUMP will automatically clear this register.	0

19.2.3.10 DMA Offset Length of Endpoint n (R16_UEPn_TX_DMA_OFS) (n=1~4)

Offset address: 0xCA + (n-1)*0x20

Bit	Name	Access	Description	Reset value
[15:0]	CHAIN_TX_DMA_OFS	RW	The offset address of the DPP in this CHAIN.	0x0400

19.2.3.11 DMA Start Address of Endpoint n (R32_UEPn_TX_DMA) (n=1~4)

Offset address: 0xCC + (n-1)*0x20

Bit	Name	Access	Description	Reset value
[31:0]	CHAIN_TX_DMA	RW	Normal mode: The DMA start address for data transmission by this CHAIN. FIFO mode: The FIFO start address corresponds to bits 16~23 of the mapped address in SRAM; The FIFO end address corresponds to bits 16~23 of the mapped address in SRAM.	0

19.2.4 Device Register-Endpoints 1 ~ 4 Receive-related Registers.

Table 19-10 Endpoint 1 receive-related registers

Name	Access address	Description	Reset value
R8_UEP1_RX_CFG	0x400380D0	Endpoint 1 configuration register	0x86

R8_UEP1_RX_CR	0x400380D1	Endpoint 1 control register	0x10
R8_UEP1_RX_SEQ	0x400380D2	Endpoint 1 serial number register	0x00
R8_UEP1_RX_ST	0x400380D3	Endpoint 1 status register	0x00
R8_UEP1_RX_CHAIN_CR	0x400380D4	Endpoint 1 CHAIN control register	0x00
R8_UEP1_RX_CHAIN_ST	0x400380D5	Endpoint 1 CHAIN status register	0x00
R16_UEP1_RX_CHAIN_LEN	0x400380D6	Endpoint 1 CHAIN receive last packet length	0x0000
R8_UEP1_RX_CHAIN_MAX_NUMP	0x400380D8	Endpoint 1 expects to receive NUMP number	0x00
R8_UEP1_RX_CHAIN_NUMP	0x400380D9	Endpoint 1 has received NUMP number	0x00
R16_UEP1_RX_DMA_OFS	0x400380DA	DMA offset length of endpoint 1	0x0400
R32_UEP1_RX_DMA	0x400380DC	DMA start address of endpoint 1	0x00000000

Table 19-11 Endpoint 2 receive-related registers

Name	Access address	Description	Reset value
R8_UEP2_RX_CFG	0x400380F0	Endpoint 2 configuration register	0x86
R8_UEP2_RX_CR	0x400380F1	Endpoint 2 control register	0x10
R8_UEP2_RX_SEQ	0x400380F2	Endpoint 2 serial number register	0x00
R8_UEP2_RX_ST	0x400380F3	Endpoint 2 status register	0x00
R8_UEP2_RX_CHAIN_CR	0x400380F4	Endpoint 2 CHAIN control register	0x00
R8_UEP2_RX_CHAIN_ST	0x400380F5	Endpoint 2 CHAIN status register	0x00
R16_UEP2_RX_CHAIN_LEN	0x400380F6	Endpoint 2 CHAIN receive last packet length	0x0000
R8_UEP2_RX_CHAIN_MAX_NUMP	0x400380F8	Endpoint 2 expects to receive NUMP number	0x00
R8_UEP2_RX_CHAIN_NUMP	0x400380F9	Endpoint 2 has received NUMP number	0x00
R16_UEP2_RX_DMA_OFS	0x400380FA	DMA offset length of endpoint 2	0x0400
R32_UEP2_RX_DMA	0x400380FC	DMA start address of endpoint 2	0x00000000

Table 19-12 Endpoint 3 receive-related registers

Name	Access address	Description	Reset value
R8_UEP3_RX_CFG	0x40038110	Endpoint 3 configuration register	0x86
R8_UEP3_RX_CR	0x40038111	Endpoint 3 control register	0x10
R8_UEP3_RX_SEQ	0x40038112	Endpoint 3 serial number register	0x00
R8_UEP3_RX_ST	0x40038113	Endpoint 3 status register	0x00
R8_UEP3_RX_CHAIN_CR	0x40038114	Endpoint 3 CHAIN control	0x00

		register	
R8_UEP3_RX_CHAIN_ST	0x40038115	Endpoint 3 CHAIN status register	0x00
R16_UEP3_RX_CHAIN_LEN	0x40038116	Endpoint 3 CHAIN receive last packet length	0x0000
R8_UEP3_RX_CHAIN_MAX_NUMP	0x40038118	Endpoint 3 expects to receive NUMP number	0x00
R8_UEP3_RX_CHAIN_NUMP	0x40038119	Endpoint 3 has received NUMP number	0x00
R16_UEP3_RX_DMA_OFS	0x4003811A	DMA offset length of endpoint 3	0x0400
R32_UEP3_RX_DMA	0x4003811C	DMA start address of endpoint 3	0x00000000

Table 19-13 Endpoint 4 receive-related registers

Name	Access address	Description	Reset value
R8_UEP4_RX_CFG	0x40038130	Endpoint 4 configuration register	0x86
R8_UEP4_RX_CR	0x40038131	Endpoint 4 control register	0x10
R8_UEP4_RX_SEQ	0x40038132	Endpoint 4 serial number register	0x00
R8_UEP4_RX_ST	0x40038133	Endpoint 4 status register	0x00
R8_UEP4_RX_CHAIN_CR	0x40038134	Endpoint 4 CHAIN control register	0x00
R8_UEP4_RX_CHAIN_ST	0x40038135	Endpoint 4 CHAIN status register	0x00
R16_UEP4_RX_CHAIN_LEN	0x40038136	Endpoint 4 CHAIN receive last packet length	0x0000
R8_UEP4_RX_CHAIN_MAX_NUMP	0x40038138	Endpoint 4 expects to receive NUMP number	0x00
R8_UEP4_RX_CHAIN_NUMP	0x40038139	Endpoint 4 has received NUMP number	0x00
R16_UEP4_RX_DMA_OFS	0x4003813A	DMA offset length of endpoint 4	0x0400
R32_UEP4_RX_DMA	0x4003813C	DMA start address of endpoint 4	0x00000000

19.2.4.1 Endpoint n Configuration Register (R8_UEPn_RX_CFG) (n=1~4)

Offset address: 0xD0 + (n-1)*0x20

Bit	Name	Access	Description	Reset value
7	RB_EP_RX_CHAIN_AUTO	RW	CHAIN automatically switches mode, which is recommended.	1
6	RB_EP_RX_FIFO_MODE	RW	A bit of 1 indicates that the current endpoint uses FIFO mode. Before this bit is 1, the start address and end address of FIFO should be configured first.	0
5	RB_EP_RX_FIFO_CFG	RW	1: Access offset addresses 0xC to 0xF, with the operation target being UEP_DMA; 0: Access offset addresses 0xC to 0xF, with the operation target being R16_EPn_FIFO_START/R16_EPn_FIFO_DE	0

			ST.	
4	RB_EP_RX_TOUT_MODE	RO	When this bit is set to 1, after receiving consecutive burst packets, if PP=1 and the packet is not a short packet, and DPH is not received within the timeout period, a TOUT_IF interrupt is generated.	0
3	RB_EP_RX_NRDY_MODE	RW	When this bit is set to 1, upon receiving a short packet or DPH (PP=0), clearing all CHAIN_EN flags will cause subsequent DPH responses to be NRDY; software reconfiguration is required.	0
2	RB_EP_RX_ERDY_AUTO	RW	ERDY automatic mode, hardware will send ERDY without software control; The synchronization endpoint recommends clearing this bit.	1
1	RB_EP_RX_SEQ_AUTO	RW	1: Software is prohibited from writing to R16_EPn_ST->EP_SEQ_NUM; 0: Software is permitted to write to R16_EPn_ST->EP_SEQ_NUM.	1
0	RB_EP_RX_ISO_MODE	RW	A value of 1 indicates that the current endpoint is a synchronous endpoint.	0

19.2.4.2 Endpoint n Control Register (R8_UEPn_RX_CR) (n=1~4)

Offset address: 0xD1 + (n-1)*0x20

Bit	Name	Access	Description	Reset value
7	RB_EP_RX_HALT	RW	Endpoint stop, active high, endpoint stop responding to DPH STALL.	0
6	RB_EP_RX_CLR	RW1Z	Write 1 to clear all configuration values and states of endpoints except UEP_CFG.	0
5	RB_EP_RX_CHAIN_CLR	RW1Z	Write 1 to clear all CHAIN configuration values and status.	0
[4:0]	RB_EP_RX_ERDY_NUMP	RW	The hardware sends the nump field of ERDY-TP, and the general configuration value is the number of supported bursts; For example, it supports 16-level burst, and the configuration value of this field is 16.	0x10

19.2.4.3 Endpoint n Serial Number Register (R8_UEPn_RX_SEQ) (n=1~4)

Offset address: 0xD2 + (n-1)*0x20

Bit	Name	Access	Description	Reset value
[7:5]	Reserved	RO	Reserved	0
[4:0]	RB_EP_RX_SEQ_NUM	RW	Current serial number of endpoint, writable in non-SEQ_AUTO mode and read-only in	0

			SEQ_AUTO mode. *Note: In general, this field is automatically controlled by hardware, and the software does not need to write EP_SEQ_NUM.*	

19.2.4.4 Endpoint n Status Register (R8_UEPn_RX_ST) (n=1~4)

Offset address: 0xD3 + (n-1)*0x20

Bit	Name	Access	Description	Reset value
7	RB_EP_RX_INT_FLAG	RO	Endpoint current interrupt flag. This bit is read-only. If all CHAIN_IF are 0, this bit is 0, otherwise it is 1.	0
6	RB_EP_RX_FC_ST	RW1Z	Current flow control status of endpoint, write 1 to clear.	0
5	RB_EP_RX_ERDY_REQ	RW1	Indicates that it is currently in the sending ERDY; Writing a 1 to this bit will send ERDY for use in non-ERDY_AUTO mode.	0
4	RB_EP_RX_CHAIN_RES	RO	CHAIN response status, corresponding to 4 independent CHAINs: 1: Responds with ACK_TP; 0: Responds with NRDY-TP. *Note: In endpoint halt mode (EP_HALT=1), responds with STALL; this bit is inactive.*	0
[3:0]	RB_EP_RX_CHAIN_EN	RO	CHAIN enabled state, corresponding to 4 independent CHAIN.	0

19.2.4.5 Endpoint n CHAIN Control Register (R8_UEPn_RX_CHAIN_CR) (n=1~4)

Offset address: 0xD4 + (n-1)*0x20

Bit	Name	Access	Description	Reset value
[7:6]	RB_EP_RX_CUR_USE	RO	CHAIN serial number currently in use.	0
[5:4]	RB_EP_RX_CUR_CFG	RO	CHAIN serial number currently configured.	0
3	Reserved	RO	Reserved	0
2	RB_EP_RX_FORCE_RET	RW	This bit effectively forces the return of the selected CHAIN state machine configuration.	0
[1:0]	RB_EP_RX_RET_SEL	RW	When FORCE_RET is valid, this bit indicates the returned CHAIN status and configuration.	0

19.2.4.6 Endpoint n CHAIN Status Register (R8_UEPn_RX_CHAIN_ST) (n=1~4)

Offset address: 0xD5 + (n-1)*0x20

Bit	Name	Access	Description	Reset value
7	RB_EP_RX_CHAI	RO	When the previously enabled CHAIN is	0

	N_EN		enabled, it is automatically set by hardware after configuring the UEP_CHAIN_NUMP register, and the CHAIN is automatically cleared after the transmission.	
6	RB_EP_RX_CHAIN_IF	RW1	This bit is written only to release the current CHAIN_IF; 1. The length of the currently received packet is less than 1024; 2. The PP bit is set to 0 in the DPH received by the asynchronous endpoint; 3. The LPF bit is set to 1 in the DPH received by the synchronous endpoint; 4. The NUMP_EMPTY bit for the current CHAIN is set to 1; 5. No DPH interrupt is received within 4 microseconds after DPH completion (optional). When receiving a short packet, the hardware may optionally trigger an automatic NRDY or disable all chains; When receiving PP=0, the hardware may optionally trigger an automatic NRDY or disable all chains; If DPP is not received after timeout, optional hardware can automatically NRDY or close all CHAIN. In synchronous mode, if the expected number of packets is not received (CRC error)	0
5	RB_EP_RX_LPF_FLAG	RO	Only the downstream endpoint is used synchronously, and the LPF status in the currently received DPH.	0
4	RB_EP_RX_ISO_PKT_ERR	RO	Only the downstream endpoint is used synchronously, and there is a CRC32 error in the currently received DPP.	0
3	RB_EP_RX_NUMP_EMPTY	RO	NUMP in the current CHAIN is 0, then this bit is set to 1.	0
2	RB_EP_RX_DPH_PP	RO	Status of the PP bit in the currently received DPH.	0
[1:0]	RB_EP_RX_CHAIN_NO	RO	CHAIN sequence number that currently generates the interrupt.	0

19.2.4.7 Endpoint n CHAIN Receive Packet Length (R16_UEPn_RX_CHAIN_LEN) (n=1~4)

Offset address: 0xD6 + (n-1)*0x20

Bit	Name	Access	Description	Reset value

[15:11]	Reserved	RO	Reserved	0
[10:0]	RB_EP_RX_CHAIN_RX_LEN	RO	The length of the last packet received by the CHAIN.	0

19.2.4.8 Endpoint n Expects to Receive NUMP Number (R8_UEPn_RX_CHAIN_MAX_NUMP) (n=1~4)

Offset address: 0xD8 + (n-1)*0x20

Bit	Name	Access	Description	Reset value
[7:0]	RX_CHAIN_MAX_NUMP	RW	The number of DPP packets that can be received by the currently completed CHAIN.	0

19.2.4.9 Endpoint n has Received NUMP Number (R8_UEPn_RX_CHAIN_NUMP) (n=1~4)

Offset address: 0xD9 + (n-1)*0x20

Bit	Name	Access	Description	Reset value
[7:0]	RX_CHAIN_NUMP	RO	Number of DPP packets that have been received at present	0

19.2.4.10 DMA Offset Length of Endpoint n (R16_UEPn_RX_DMA_OFS) (n=1~4)

Offset address: 0xDA + (n-1)*0x20

Bit	Name	Access	Description	Reset value
[15:0]	CHAIN _DMA_OFS	RW	The offset address of the DPP in this CHAIN.	0x0400

19.2.4.11 DMA Start Address of Endpoint n (R32_UEPn_RX_DMA) (n=1~4)

Offset address: 0xDC + (n-1)*0x20

Bit	Name	Access	Description	Reset value
[31:0]	CHAIN_DMA_BA	RW	Normal mode: The DMA start address for data transmission by this CHAIN. FIFO mode: The FIFO start address corresponds to bits 16–23 of the mapped address in SRAM; The FIFO end address corresponds to bits 16–23 of the mapped address in SRAM.	0

19.2.5 USBSS LINK Register Description

Table 19-14 USBSS LINK registers

Name	Access address	Description	Reset value
R32_LINK_CFG	0x40038000	LINK configuration register	0xC0000128
R32_LINK_CTRL	0x40038004	LINK control	0xXX000013
R32_LINK_INT_CTRL	0x40038008	LINK interrupt enable register	0x00000000
R32_LINK_INT_FLAG	0x4003800C	LINK interrupt flag register	0x00000000
R32_LINK_STATUS	0x40038010	LINK status register	0xX0000430

R8_LINK_ITP_PRE	0x40038017	LINK ITP timeout mode register	0x80
R8_LINK_U2_INACT_TIME	0x4003801D	LINK U2 inactivity timeout counter threshold register	0x00
R8_LINK_U1_WKUP_FILTER	0x40038028	U1 wake-up LFPS valid duration register	0x4A
R8_LINK_U2_WKUP_FILTER	0x4003802C	U2 wake-up LFPS valid duration register	0x02
R8_LINK_U3_WKUP_FILTER	0x40038030	U3 wake-up LFPS valid duration register	0x64
R16_LINK_ISO_DLY	0x40038040	LINK synchronization delay register	0x0028
R16_LINK_LPM_CR	0x40038050	Link power management register	0x0180
R32_LINK_LMP_PORT_CAP	0x40038054	PORT_CAP register	0xXX000000
R32_LINK_LMP_RX_DATA0	0x40038058	LMP receive data 0 register	0xXXXXXXXX
R32_LINK_LMP_RX_DATA1	0x4003805C	LMP receive data 1 register	0xXXXXXXXX
R32_LINK_LMP_RX_DATA2	0x40038060	LMP receive data 2 register	0xXXXXXXXX
R32_LINK_LMP_TX_DATA0	0x40038064	USB custom HP data 0 register	0xXXXXXXXX
R32_LINK_LMP_TX_DATA1	0x40038068	USB custom HP data 1 register	0xXXXXXXXX
R32_LINK_LMP_TX_DATA2	0x4003806C	USB custom HP data 2 register	0xXXXXXXXX

19.2.5.1 LINK Configuration Register (R32_LINK_CFG)

Offset address: 0x00

Bit	Name	Access	Description	Reset value
31	RB_LINK_RESET	RW	LINK reset, including resetting the state machine and all interrupt flags, is active high.	1
[30:22]	Reserved	RO	Reserved	0x100
21	RB_LINK_TOUT_MODE	RW	1: 3.2 SPEC; 0: 3.0 SPEC. *Note: Different protocol versions affect the timer duration in flow control:* *PM_LC_TIMER: 3.2_SPEC: 4us/3.0_SPEC: 3us;* *PM_ENTRY_TIMER: 3.2_SPEC: 8us/3.0_SPEC: 6us;* *PEND_HP_TIMER: 3.2_SPEC: 10us/3.0_SPEC: 3us.*	0
20	RB_LINK_U1_PING_EN	RW	U1 PING_FPFS enable: 1: Enable transmission; 0: Disable transmission.	0
[19:18]	RB_LINK_SKP_MODE	RW	SKP discard control for descrambling: 00: If the lower byte of the lower two bytes is SKP, disable descrambling for the lower two bytes; if the lower byte of the upper two bytes is SKP, disable descrambling for the upper two bytes;	0

			01/11: Both low bytes are SKP, disabling descrambling enable for the low two bytes; both high bytes are SKP, disabling descrambling enable for the high two bytes; 10: Either byte in the low two bytes is SKP, disabling descrambling enable for the low two bytes; either byte in the high two bytes is SKP, disabling descrambling enable for the high two bytes.	
17	RB_LINK_U2_ALLOW	RW	Active high: Upon receiving LGO_U2, it responds with LAU to permit entry into U2 state; otherwise, upon receiving LGO_U2, it responds with LXU to deny entry into U2 state. *Note: For the downstream port, this enable has a fixed relationship with the U2_TIMEOUT value. When U2_TIMEOUT > 0, configuring U2_ALLOW is effective; otherwise, it is ineffective.*	0
16	RB_LINK_U1_ALLOW	RW	Active high: Upon receiving LGO_U1, it responds with LAU to permit entry into U1 state; otherwise, upon receiving LGO_U1, it responds with LXU to deny entry into U1 state. *Note: For the downstream port, this enable has a fixed relationship with the U1_TIMEOUT value. When U1_TIMEOUT > 0, configuring U1_ALLOW is effective; otherwise, it is ineffective.*	0
15	RB_LINK_LTSSM_MODE	RW	The link state machine enters DISABLED mode: 1: Enter DISABLED mode according to protocol specifications upon timeout, detection count, etc.; 0: Enter DISABLED mode without adhering to protocol specifications upon timeout, detection count, etc.	0
14	RB_LINK_LOOKBACK_ACT	RW	Used in LOOPBACK mode for LOOPBACK master control pattern transmission (LOOPBACK slave maintains 0), ACT is a high-level signal sustained for a period of time. 1. When this bit is pulled high, the LOOPBACK master must send BRST to the LOOPBACK slave to reset the LOOPBACK slave's error counter (slave loopback BRST). 2. While ACT is high, the LOOPBACK master sends BDAT to the slave. The slave compares	0

			the received BDAT with a scrambled 0 to count errors (slave loopback BDAT). 3. When ACT goes low, the LOOPBACK master sends BERC to the slave (the slave returns the error count BCNT to the master).	
13	RB_LINK_LOOPBACK_EN	RW	Enable the LOOPBACK enable bit, active high. This must be used in conjunction with the LOOPBACK enable bit in TX/RX_TS_CFG[3]. Both must be active for the LINK to enter LOOPBACK mode (used to enable data loopback and error counting for the LOOPBACK slave; the LOOPBACK master remains at 0).	0
12	RB_LINK_U2_DET_EN	RW	Device detection method in U2 state: 1: Software actively initiates detection of connected devices in U2 state; 0: Hardware automatically times detection of connected devices in U2 state.	0
[11:10]	RB_LINK_CP78_SEL	RW	When using Compliance Pattern 7/8, the length of consecutive zeros or consecutive ones sent: 00: 190 bits; 01: 120 bits; 10: 50 bits; 11: 250 bits.	0
[9:8]	RB_LINK_TX_DEEMPH	RW	Transmitter De-Emphasis control: 00: -6.0 dB De-Emphasis; 01: -3.5 dB De-Emphasis (default); 10: No De-Emphasis; 11: Reserved. *Note: Normal operating mode must be 0/1 only; other values are reserved for COMPLIANCE.*	01b
7	RB_LINK_TX_SWING	RW	Transmitter signal swing control: Low swing reduces power consumption but affects transmission distance. 1: Low Swing（400mv-1200mv）; 0: Full Swing（800mv-1200mv）。	0
6	RB_LINK_RX_EQ_EN	RW	Receiver equalization enables control, and protocol provisions are optional.	0
5	RB_LINK_LFPS_RX_PD	RW	LFPS reception control, this bit is 1 to disable LFPS reception.	1
4	RB_LINK_COMPLIANCE_EN	RW	POLLING_LFPS timeout entry into COMPLIANCE mode: 1: Enter upon timeout;	0

			0: Do not enter upon timeout.	
3	RB_LINK_PHY_RESET	RW	PIPE interface reset 1: Reset; 0: Clear reset.	1
2	RB_LINK_SS_PLR_SWAP	RW	Swap SSTX and SSRX polarities as follows: 1: SSTXA(SSTXM)，SSTXB(SSTXP)；SSRXA(SSRXM)，SSRXB(SSRXP)。 0: SSTXA(SSTXP)，SSTXB(SSTXM)，SSRXA(SSRXP)，SSRXB(SSRXM)。 *Note: The difference of this adaptive Rx Polarity is that Rx Polarity only exchanges SSRXM and SSRXP.*	0
1	RB_LINK_RX_TERM_EN	RW	Terminal resistor control: 1: Connect termination resistor; 0: Remove termination resistor.	0
0	RB_LINK_DOWN_MODE	RW	Peripheral type: 1: Downstream； 0: Upstream。 In device mode, it should be configured as Upstream.	0

19.2.5.2 LINK Control Register (R32_LINK_CTRL)

Offset address: 0x04

Bit	Name	Access	Description	Reset value
[31:24]	LINK_RX_TS_CFG	RO	Received Link Control: BIT0: 1: Reset; 0: Normal transmission. BIT1: Reserved. BIT2: 1: Enable Loopback; 0: Disable Loopback. BIT3: 1: Disable scrambling; 0: Enable scrambling. BIT4~7: Reserved.	0
[23:16]	LINK_TX_TS_CFG	RW	Link configuration for sending TS1/TS2 training sequences: BIT0: 1: HOT_RESET; 0: Normal training mode. BIT1: Reserved. BIT2: 1: Enable Loopback; 0: Disable Loopback. BIT3: 1: Disable scrambling; 0: Enable scrambling. BIT4~7: Reserved. Specifically, reserved bits must be set to 0; BIT0 and BIT2 cannot be both 1	0

			simultaneously. In device mode, if a valid HOT_RESET is received, it generates a HOT_RESET interrupt flag while BIT0 is automatically set to 1 by hardware to respond to the host's HOT_RESET.	
15	LINK_TX_LGO_U3	RW	When this bit is active, send LGO_U3 (high active) to trigger hardware auto-reset. The protocol mandates that the receiver cannot refuse to enter U3 state. Querying the LINK status as U3 indicates entry into the U3 low-power state. At this point, the software should change the current power mode to P3.	0
14	LINK_TX_LGO_U2	RW	When this bit is active, send LGO_U2 (high active) to trigger hardware auto-reset. If the peer responds with LXU, it indicates refusal to enter U2 state. If the LINK status query returns U2, it confirms the peer accepts entering U2 low-power state. At this point, the software should change the current power mode to P2.	0
13	LINK_TX_LGO_U1	RW	When this bit is active, send LGO_U1 (high active) to trigger hardware auto-reset. If the peer responds with LXU, it indicates refusal to enter U1 mode. If the LINK status query returns U1, it confirms the peer accepts entering U1 low-power mode. At this point, the software should change the current power mode to P1.	0
12	LINK_POLLING_EN	RW	High active. If SS.RX_DETECT detects TERM, the POLLING handshake will proceed after this bit is set active in software. Note: When AUTO_MODE[9] is 1, if AUTO_MODE[8] is 1 and the current power supply is P2, initiate POLLING_LFPS handshake; When AUTO_MODE[9] is 1, if AUTO_MODE[7] is 1 and the current power supply is P0, initiate POLLING_LFPS handshake.	0
11	Reserved	RO	Reserved	0
10	LINK_LUP_LDN_EN	RW	U0 state: When no data is present, whether to transmit LUP/LDN packets every 10μs:	0

			1: Enable transmission; 0: Disable transmission.	
9	LINK_TX_UX_EXIT	RW1Z	Active high, hardware auto-reset: 1. If LINK is in the U3 state, it is equivalent to executing U3_WAKEUP. After the U3_WAKEUP handshake fails, the software should reinitiate U3_WAKEUP after at least 100ms. If after receiving U3_WAKEUP, the handshake signal cannot be returned for some reason, when you are ready to enter U0, you should actively send U3_WAKEUP. 2. If LINK is in U2 state, it is equivalent to executing U2_EXIT (LFPS). 3. If LINK is in U1 state, this is equivalent to executing U1_EXIT (LFPS). 4. If the U1_EXIT/U2_EXIT handshake fails, LINK enters INACTIVE.	0
8	LINK_TX_WARM_RST	RW1Z	Active high, cleared by software. This bit is valid and will send warm-reset. In device mode: Upon detecting a warm reset initiated by the host after 18ms of continuous transmission, the device controller will automatically acknowledge the warm reset and proceed with the reset. *Specifically, the protocol stipulates that only the host has the capability to actively transmit WARM_RESET; device mode can only passively respond WARM_RESET.*	0
7	LINK_GO_RX_DET	RW1Z	Before setting this bit, configure PD_MODE to P2 mode. Set LINK to enter SS.RX_DETECT. Software polls TERM_PRESENT to determine if a connection exists or has been lost. High active, auto-reset.	0
6	LINK_GO_RECOVERY	RW1Z	Set LINK to enter SS.RECOVERY mode, high active, hardware auto-reset.	0
5	LINK_GO_INACTIVE	RW1Z	Set LINK to SS.INACTIVE, high active, hardware auto-reset.	0
4	LINK_GO_DISABLED	RW	Set LINK to SS.DISABLED, high active, requiring software reset. If hardware automatically enters SS.DISABLED, this bit is automatically set to 1 by hardware. In device mode, upon receiving a USB 2.0 bus reset, the software should first set PD_MODE	1

			to P2 mode, then clear this bit. LTSSM will automatically enter SS.RX_DETECT, where the device performs 8 detections (every 12ms). If TERM is not detected, it enters DISABLE state while communicating in USB 2.0 mode. In Host Mode, the BUS_RESET signal should be sent via the USB 2.0 controller. The software should first set PD_MODE to P2 mode, then clear this bit. If TERM is detected, a POLLING_LFPS handshake will occur. Otherwise, the software should set the power mode to P3 and communicate in USB 2.0 mode.	
[3:2]	Reserved	RO	Reserved	0
[1:0]	LINK_PD_MODE	RW	Configure the PHY's current power mode, corresponding to PO/P1/P2/P3 in the PIPE: 00: PO: Normal operating mode; 01: P1: Low-power mode. In this mode, both ends of the port periodically send PING-LFPS to maintain the connection; 10: P2: Low-power mode. This mode consumes less power than P1 mode, but the time required to exit this mode also increases; 11: P3: Ultra-low power mode. In this mode, the PHY clock (125MHz) is disabled. Changing this configuration value will cause LINK_BUSY to go high. The software should check that LINK_BUSY is 0 to indicate the switchover is complete. Query PD_MODE_CURRENT in LINK_STATUS to obtain the PHY's current power mode (optional operation).	11b

19.2.5.3 LINK Interrupt Enable Register (R32_LINK_INT_CTRL)

Offset address: 0x08

Bit	Name	Access	Description	Reset value
31	LINK_IE_STATE_CHG	RW	Link state change flag interrupt enable: 1: Enable; 0: Disable.	0
30	LINK_IE_U1_TOUT	RW	U1 timeout interrupt enable: 1: Enable; 0: Disable.	0
29	LINK_IE_U2_TOUT	RW	U2 timeout interrupt enable: 1: Enable;	0

			0: Disable.	
28	LINK_IE_UX_FAIL	RW	UX conversion failure interrupt enable: 1: Enable; 0: Disable.	0
27	LINK_IE_TX_WARMRST	RW	Transmit WARM_RESET to end interrupt enable: 1: Enable; 0: Disable.	0
26	LINK_IE_UX_EXIT_FAIL	RW	Ux exit failure interrupt enable: 1: Enable; 0: Disable.	0
25	CONFIG_DONE_INT_EN	RW	Link receive or transmit to PORT_CapabilityLMP interrupt enable: 1: Enable; 0: Disable.	0
24	RX_CAP_FAIL_INT_EN	RW	The CAP of the other party received by the link and the own CAP cannot be combined to enable interrupts for the uplink port and downstream port: 1: Enable; 0: Disable.	0
23	LINK_IE_RX_LMP_TOUT	RW	Receive LMP timeout interrupt enable: 1: Enabled; 0: Disabled.	0
22	LINK_IE_TX_LMP	RW	Successful entry into U0 allows sending HP packet interrupt enable: 1: Enable; 0: Disable.	0
21	LINK_IE_RX_LMP	RW	Received link command flag interrupt enable: 1: Enable; 0: Disable.	0
20	LINK_IE_RX_DET	RW	Interrupt enable for link entering Rx.Detect state: 1: Enabled; 0: Disabled.	0
19	LINK_IE_LOOPBACK	RW	The link enters loopback mode for testing and fault isolation. Interrupt enable: 1: Enabled; 0: Disabled.	0
18	LINK_IE_COMPLIANCE	RW	Link enter compliance test, enabled via compatibility test or physical layer consistency test interrupt: 1: Enabled; 0: Disabled.	0

17	LINK_IE_HPBUF_FULL	RW	HP_BUF transmit FIFO full interrupt enable: 1: Enable; 0: Disable.	0
16	LINK_IE_HPBUF_EMPTY	RW	HP_BUF transmit FIFO empty interrupt enable: 1: Enable; 0: Disable.	0
15	LINK_IE_HOT_RST	RW	HOT RESRT, using reset interrupts from the TS1/TS2 ordered set: 1: Enable; 0: Disable.	0
14	LINK_IE_U3_WAKEUP	RW	In U3 state, upon receiving a low-frequency periodic signal (LFPS) wake-up interrupt enable: 1: Enable; 0: Disable.	0
13	LINK_IE_WARM_RST	RW	Enable warm reset (unconnected) interrupts using LPFS: 1: Enable; 0: Disable.	0
12	LINK_IE_UX_EXIT	RW	Interrupt enable for Ux exit request received: 1: Enable; 0: Disable.	0
11	LINK_IE_TXEQ	RW	LINK enters POLLING_RXEQ for receiver equalization training interrupt enable: 1: Enable; 0: Disable.	0
10	LINK_IE_TERM_PRES	RW	LINK enters RX_DETECT mode for detecting remote receiver tracking segment impedance detection interrupt enable: 1: Enable; 0: Disable.	0
9	LINK_IE_UX_REJ	RW	Transmit LGO_Ux, receive an LXU interrupt enable sent by Ux: 1: Enable; 0: Disable.	0
8	LINK_IE_U3_WK_TOUT	RW	Enable U3 command timeout interrupt request: 1: Enable; 0: Disable.	0
7	LINK_IE_GO_U0	RW	LINK is in U0 interrupt enable mode: 1: Enabled; 0: Disabled.	0
6	LINK_IE_GO_U1	RW	LINK is in U1 interrupt enable mode:	0

			1: Enabled; 0: Disabled.	
5	LINK_IE_GO_U2	RW	LINK is in U2 interrupt enable mode: 1: Enabled; 0: Disabled.	0
4	LINK_IE_GO_U3	RW	LINK is in U3 interrupt enable mode: 1: Enabled; 0: Disabled.	0
3	LINK_IE_DISABLE	RW	LINK is in DISABLED interrupt enable mode: 1: Enabled; 0: Disabled.	0
2	LINK_IE_INACTIVE	RW	LINK is in INACTIVE interrupt enable mode: 1: Enabled; 0: Disabled.	0
1	LINK_IE_RECOVERY	RW	Error LINK is in RECOVERY interrupt enable mode: 1: Enable; 0: Disable.	0
0	LINK_IE_READY	RW	LINK initialization completed, including two ports prior to (Header Sequence Number Advertisement) and (RX Header Buffer Credit Advertisement). Interrupt enable: 1: Enabled; 0: Disabled.	0

19.2.5.4 LINK Interrupt Flag Register (R32_LINK_INT_FLAG)

Offset address: 0x0C

Bit	Name	Access	Description	Reset value
31	LINK_IF_STATE_CHG	RW1Z	Link_reset is reset to 0 (write 1 to clear it): 1: Link state machine change flag; 0: Link state machine has not changed.	0
30	LINK_IF_U1_TOUT	RW1Z	U1 timeout interrupt; (Write 1 to clear): 1: Timeout occurred; 0: No timeout occurred;	0
29	LINK_IF_U2_TOUT	RW1Z	U2 timeout interrupt; (Write 1 to clear): 1: Timeout occurred; 0: No timeout occurred;	0
28	LINK_IF_UX_FAIL	RW1Z	Ux entry failure interrupt; (Write 1 to clear): 1: Ux entry failed; 0: No action.	0
27	LINK_IF_TX_WARMRST	RW1Z	Transmit WARM_RESET to terminate interrupt; (Write 1 to clear):	0

			1: Transmit WARM_RESET to terminate; 0: No action.	
26	LINK_IF_UX_EXIT_FAIL	RW1Z	Ux exit failed interrupt; (Write 1 to clear): 1: Ux exit failed; 0: No action.	0
25	CONFIG_DONE_FLAG	RW1Z	Link received or sent to PORT_Capability LMP interrupt (cleared by writing 1). 1: Link received or sent to PORT_Capability LMP; 0: No action.	0
24	RX_CAP_FAIL_FLAG	RW1Z	The CAP of the other party received by the link and the own CAP cannot be combined to form an interrupt for the uplink port and downlink port (cleared by writing 1). 1: The other party's CAP received by the link and your own CAP cannot be combined to form an upstream port and a downlink port; 0: No action.	0
23	LINK_IF_RX_LMP_TOUT	RW1Z	After LINK initialization completes, exchange Port Capabilities/Configuration LMPs timeout (20us) interrupt flags.	0
22	LINK_IF_TX_LMP	RW1Z	LINK Initialization Complete Interrupt Flag: When this signal goes high, software configures the transmission of the Port Capabilities LMP for port capability exchange.	0
21	LINK_IF_RX_LMP	RW1Z	Receive LMP interrupt flag: Control related LMP transmission during the port capabilities exchange phase: When the downstream port receives a Port Capabilities LMP, software configures and transmits a Port Configuration LMP; When the upstream port receives a Port Capabilities LMP, software configures and transmits a Port Configuration Response LMP.	0
20	LINK_IF_RX_DET	RW1Z	The RX_DETECT status interrupt flag is set. Once this flag is set to 1, the software configures PWR_MODE to P2, preparing for RX_DETECT.	0
19	LINK_IF_LOOPBACK	RW1Z	LINK enters LOOPBACK status interrupt flag.	0
18	LINK_IF_COMPLIANCE	RW1Z	LINK enters COMPLIANCE status interrupt flag.	0
17	LINK_IF_HPBUF_FULL	RW1Z	Header Packet buffer full interrupt flag.	0

16	LINK_IF_HPBUF_EMPTY	RW1Z	Header Packet buffer empty interrupt flag.	0
15	LINK_IF_HOT_RST	RW1Z	The device receives the HOT RESET interrupt flag: After this signal goes high, the software should clear TX_TS_CFG[0] to complete the HOT RESET process.	0
14	LINK_IF_WAKEUP	RW1Z	The power supply is in P3 mode, and the LFPS signal interrupt flag is detected.	0
13	LINK_IF_WARM_RST	RW1Z	Interrupt flag of WARMRESET state change (valid-> invalid or invalid-> valid) received by the device.	0
12	LINK_IF_UX_EXIT	RW1Z	Upon receiving the LFPS ready signal, LINK prepares to exit the U1/U2/U3 request interrupt flag. Once this bit is set to 1, the software should set PWR_MODE to P0. After setting PWR_MODE to P0, the Ux exit timer begins counting. Upon reaching U1_EXIT_TIMER, U2_EXIT_TIMER, or U3_WAKE_TIMER, LINK exits the Ux state and enters Recovery mode.	0
11	LINK_IF_TXEQ	RW1Z	LINK entering TXEQ State Interrupt Flag: Indicates completion of the POLLING handshake, awaiting software to set P0 to P0_MODE to enter the TXEQ phase.	0
10	LINK_IF_TERM_PRES	RW1Z	Detected TERM disconnect or connection interruption flag.	0
9	LINK_IF_UX_REJ	RW1Z	LINK refused to enter low power consumption mode (U1/U2) interrupt flag.	0
8	LINK_IF_U3_WK_TOUT	RW1Z	Wake up timeout interrupt flag from U3 (10ms).	0
7	LINK_IF_GO_U0	RW1Z	LINK enters U0 status interrupt flag.	0
6	LINK_IF_GO_U1	RW1Z	LINK enters U1 status interrupt flag.	0
5	LINK_IF_GO_U2	RW1Z	LINK enters U2 status interrupt flag.	0
4	LINK_IF_GO_U3	RW1Z	LINK enters U3 status interrupt flag.	0
3	LINK_IF_DISABLE	RW1Z	LINK enters SS.DISABLE status interrupt flag: *Note: The software is directly connected with GO_DISABLE enable (LINK_CTRL), which causes LINK to directly enter DISABLE, and the interrupt will not be pulled high.*	0
2	LINK_IF_INACTIVE	RW1Z	LINK enters SS.INACTIVE status interrupt flag. *Note: Software directly configures*	0

			GO_INACTIVE enable (LINK_CTRL), causing LINK to enter INACTIVE directly without interrupt flag being pulled high.	
1	LINK_IF_RECOVERY	RW1Z	LINK enters SS.RECOVERY state interrupt flag. After this bit is set, query LINK_ERR_STATUS to determine the cause of LINK entering SS.RECOVERY. *Note: Direct software configuration of GO_RECOVERY enable (LINK_CTRL) causes LINK to enter RECOVERY directly without raising the interrupt flag.*	0
0	LINK_IF_READY	RW1Z	LINK enters the U0 state and clears the LINK initialization interrupt flag.	0

19.2.5.5 LINK Status Register (R32_LINK_STATUS)

Offset address: 0x10

Bit	Name	Access	Description	Reset value
31	LINK_HPBUF_EMPTY	RO	HP buffer is empty.	x
30	LINK_HPBUF_FULL	RO	HP buffer is full.	x
29	LINK_HPBUF_IDLE	RO	HP_BUF transmit state of FIFO idle: 1: HP_BUF transmit FIFO has data; 0: HP_BUF transmit FIFO is empty.	x
[28:23]	RESERVE	RO	Reserved	0
22	LINK_U3_SLEEP_ALLOW	RO	The link is in U3 state, allowing sleep.	x
21	LINK_U2_SLEEP_ALLOW	RO	The link is in U2 state, allowing sleep.	x
20	LINK_RXDET_SLEEP_ALLOW	RO	The link is in RXDET state, allowing sleep.	x
19	LINK_WAKUP	RO	Link wake-up signal received.	x
18	LINK_RX_LFPS	RO	Link receives LFPS signal.	x
17	LINK_RX_DETECT	RO	Link is at P2, and RX_DETECT is in progress.	x
16	LINK_RX_UX_EXIT_REQ	RO	Received Ux exit request: 1: LINK received a valid Ux EXIT LFPS signal, preparing to exit U1/U2/U3. Software should configure the power mode to P0 (after the power mode transitions to P0, the Ux exit timer starts counting and LINK enters RECOVERY upon reaching the preset value);	0

			0: No change.	
[15:12]	Reserved	RO	Reserved	0
[11:8]	LINK_STATE	RO	Link status: 0000: STATE_U0; 0001: STATE_U1; 0010: STATE_U2; 0011: STATE_U3; 0100: STATE_DISABLED; 0101: STATE_RXDET; 0110: STATE_INACTIVE; 0111: STATE_POLLING; 1000: STATE_RECOVERY; 1001: STATE_HOTRST; 1010: STATE_COMPLIANC;; 1011: STATE_LOOPBACK; Others: Reserved.	4h
7	Reserved	RO	Reserved	0
6	LINK_TXEQ	RO	The link is in POLLING_RXEQ.	0
[5:4]	LINK_PD_MODE_MASK	RO	Link power status: 00: P0 status; 01: P1 status; 10: P2 status; 11: Default status P3.	11b
3	LINK_READY	RO	When LINK enters U0 state, this bit is set after initialization (broadcast) is completed, and this bit is automatically cleared when it exits U0 state.	0
2	LINK_BUSY	RO	LINK busy status. When switching PD_MODE, this bit is 1, and it is automatically cleared by hardware after switching.	0
1	LINK_RX_WARM_RST	RO	Received a valid WARM_RESET signal from the host (hardware automatically powers up 18ms after receiving the host's WARM_RESET signal).	0
0	LINK_RX_TERM_PRES	RO	After RX_DETECT, if a receive terminal resistor is present, this bit is set to 1.	0

19.2.5.6 LINK ITP Timeout Mode Register (R8_LINK_ITP_PRE)

Offset address: 0x17

Bit	Name	Access	Description	Reset value
[7:0]	ITP_PRE	RO	ITP timeout mode: 1: Ignore the impact of ITP transmission/reception on the link idle state for	0x80

			both downlink and uplink ports; 0: The link is considered non-idle when the downlink port transmits non-delayed ITP or the uplink port receives non-delayed ITP.	

19.2.5.7 LINK U2 Inactivity Timeout Counter Threshold Register (R8_LINK_U2_INACT_TIMER)

Offset address: 0x1D

Bit	Name	Access	Description	Reset value
[7:0]	U2_INACTIVE_TIMER	RW	Value of U2's inactivity timeout counter threshold: U2 Timeout value (this value represents the maximum allowed time in U0/U1 inactive state, and enters U2 state after timeout), value range: 0x00~0xff, 256us/LSB. *Note: When the U1/U2 Timeout value is 0xff, it means that the timeout time of the link inactive state is infinite, and the conversion request to U1/U2 will not be actively initiated when the timer reaches 0xff.*	0

19.2.5.8 U1 Wake-up LFPS Valid Duration Register (R8_LINK_U1_WKUP_FILTER)

Offset address: 0x28

Bit	Name	Access	Description	Reset value
[7:0]	U1_WKUP_FILTER	RW	The valid duration of the LFPS received during U1 exit. The port sends LFPS for handshaking. When the received LFPS reaches this duration, the received U1 EXIT is considered valid. Pull high the sent LFPS_last to indicate successful handshaking. Default is 600ns. U1_WKUP_FILTER[7]: Controls the time unit of U1_WKUP_FILTER[6:0]; 1: U1_WKUP_FILTER[6:0] unit is in us, duration is (n)us; 0: U1_WKUP_FILTER[6:0] unit is 8ns, duration is (8n)ns, where n ranges from 0 to 124; *Note: When the duration exceeds 1ns, the configuration method with U1_WKUP_FILTER[7] set to 1 must be used; otherwise, U1 cannot be awakened.*	0x4A

19.2.5.9 U2 Wake-up LFPS Valid Duration Register (R8_LINK_U2_WKUP_FILTER)

Offset address: 0x2C

Bit	Name	Access	Description	Reset value

[7:0]	U2_WKUP_FILTER	RW	The valid determination duration for LFPS received during U2 wake-up and loopback exit. The port sends LFPS for handshaking. When the received LFPS reaches this duration, the LFPS_LAST signal is pulled high, indicating successful handshaking. The received U2 EXIT can then be considered valid, with a duration of (n)us, defaulting to 2us.	0x02

19.2.5.10 U3 Wake-up LFPS Valid Duration Register (R8_LINK_U3_WKUP_FILTER)

Offset address: 0x30

Bit	Name	Access	Description	Reset value
[7:0]	U3_WKUP_FILTER	RW	The valid determination duration for LFPS received during U3 wake-up. The port sends LFPS for handshaking. When the received LFPS reaches this duration, the LFPS_LAST signal is pulled high, indicating successful handshaking. The received U3 EXIT signal is then considered valid. The duration is (n)us, with a default value of 100us.	0x64

19.2.5.11 LINK Synchronous Delay Register (R16_LINK_ISO_DLY)

Offset address: 0x40

Bit	Name	Access	Description	Reset value
[15:0]	LINK_ISOCH_DLY	RW	The delay time for serializing bitstreams into parallel data defaults to 40ns. As the device, write the delay value to this register. Note that the lower three bits of this register are invalid (since the clock frequency is 125MHz, the delay time must be set to 8*n).	0x28

19.2.5.12 Link Power Management Register (R16_LINK_LPM_CR)

Offset address: 0x50

Bit	Name	Access	Description	Reset value
[15:14]	Reserved	RO	Reserved	0
13	PHY_CHSEL_AUTO	RW	PHY layer transceiver channel type auto-Selection: 1: After device detection completes, if the connection status flag remains unchanged, switch to another channel type. The default channel type is standard USB interface. 0: PHY layer disables automatic transceiver channel type switching. Only software-	0

			configured transceiver types can be used, with standard USB interface as default.	
12	LPM_RXDET_EN	RO	When the lpm counter reaches the expected value RXDET_EXP, confirming a device is connected to the PHY, this bit is pulled high.	0
11	LPM_TERM_PRESENT	R0	The PHY layer detected a device connection.	0
10	LPM_TERM_CHG	RO	The connected device changes, and the device is inserted or pulled out.	0
9	LPM_EN	RW	CLKLPM is fixed at 12MHz. When LINK_PD_MODE=3 & LPM_EN=0 & OSCOUTEN=0 & PLLALIV=0, CLKOSCO stops and the crystal oscillator ceases operation. 1: Enables CLKLPM in L1 mode. 0: Disables. *Note: This bit is unrelated to the USBSS controller. To enter low-power mode, the crystal oscillator must be disabled by clearing this bit.*	0
8	LPM_RST	RW	Reset signal for LPM-related registers: 1: Resets LPM counter and LPM terminal detection; 0: Does not reset.	1
[7:0]	RXDET_EXP	RW	Detect external device counter register.	0x80

19.2.5.13 PORT_CAP Register (R32_LINK_LMP_PORT_CAP)

Offset address: 0x54

Bit	Name	Access	Description	Reset value
31	LINK_LMP_RX_CAP_VLD	RO	Upon receiving a valid PORT_CAP_LMP, the protocol specifies that both LINK parties exchange PORT_CAP_LMP within 20us after entering the U0 state. *Note: This flag signal is set to 0 when the LINK state changes and enters Port_State_Polling.*	0
30	LINK_LMP_TX_CAP_VLD	RO	PORT Capability configuration complete indicator (software operation, write 1 to clear 0): 1: Indicates PORT Capability configuration is incomplete; 0: Indicates PORT Capability configuration is complete (distinguished between host and device scenarios). The host clears the indicator to 0 upon	0

			receiving the device's PORT Configuration Response. The device clears the indicator to 0 upon receiving the host's PORT Configuration and replying with a PORT Configuration Response.	
[29:24]	LINK_SPEED	RO	[24] Position 1 indicates that the device supports USB 3.2 Gen 1 (5Gbps).	x
[23:20]	RX_CAP_TIEBRK	RO	Tiebreaker is used for PORT type arbitration when 2 PORTs (REG_PORT_CAP[17:16]=11) with both upstream and downstream capabilities are connected. The port with a higher value becomes the downstream port. If the tiebreakers of the 2 PORTs are equal, PORT_CAP will be re-exchanged.	X
19	RX_CAP_OTG	RO	Receives the OTG bit in the LMP of the Port Capability; indicates whether OTG functionality is available.	X
[18:17]	RX_CAP_DIR	RO	[17]: If the bit is 1, it means that the port can be configured as a downstream port; [18]: If the bit is 1, it means that the port can be configured as an upstream port.	X
16	LINK_DOWN_MODE	RO	Determine the configuration of your own port based on the PORT Capability of yourself and the other party. 1: Configure it as a downstream port; 0: Configure it as an upstream port.	0
[15:8]	RESERVED	RO	Reserved	0
[7:4]	CFG_TIEBREAKER	RW	Configure the Tiebreaker bit in the LMP of Port Capability of this port; The bit with a higher value is used as the downstream port.	0
3	CFG_UP_MODE	RW	Configure the high bit of Direction in the LMP of Port Capability of this port. 1: Supports configurability as an upstream port; 0: Does not support configurability as an upstream port.	0
2	CFG_DOWN_MODE	RW	Configure the low bit of Direction in the LMP of Port Capability of this port. 1: Supports configurability as a downstream port; 0: Does not support configurability as a downstream port.	0
1	CFG_OTG_CAPAB	RW	Configure the OTG bit in the LMP of Port	0

	LE		Capability of this port; used to enable this OTG function.	
0	AUTO_CAP	RW	When the automatic configuration of Port Capability fails, this bit is automatically cleared. 1: Hardware automatically configures Port Capability; 0: Port Capability is configured through software.	0

19.2.5.14 LMP Receive Data 0 Register (R32_LINK_LMP_RX_DATA0)

Offset address: 0x58

Bit	Name	Access	Description	Reset value
[31:0]	LINK_LMP_RX_DATA0	RO	After LMP reception is complete, HP data is stored in this register [31:0].	X

19.2.5.15 LMP Receive Data 1 Register (R32_LINK_LMP_RX_DATA1)

Offset address: 0x5C

Bit	Name	Access	Description	Reset value
[31:0]	LINK_LMP_RX_DATA1	RO	After LMP reception is complete, HP data is stored in this register [63:32].	X

19.2.5.16 LMP Receive Data 2 Register (R32_LINK_LMP_RX_DATA2)

Offset address: 0x60

Bit	Name	Access	Description	Reset value
[31:0]	LINK_LMP_RX_DATA2	RO	After LMP reception is complete, HP data is stored in this register [95:64].	X

19.2.5.17 USB Custom HP Data 0 Register (R32_LINK_LMP_TX_DATA0)

Offset address: 0x64

Bit	Name	Access	Description	Reset value
[31:0]	LINK_LMP_TX_DATA0	RW	User-defined data sent to HP [31:0].	X

19.2.5.18 USB Custom HP Data 1 Register (R32_LINK_LMP_TX_DATA1)

Offset address: 0x68

Bit	Name	Access	Description	Reset value
[31:0]	LINK_LMP_TX_DATA1	RW	User-defined data sent to HP [63:32].	X

19.2.5.19 USB Custom HP Data 2 Register (R32_LINK_LMP_TX_DATA2)

Offset address: 0x6C

Bit	Name	Access	Description	Reset value
[31:0]	LINK_LMP_TX_DATA2	RW	User-defined data sent to HP [95:64].	X

Note:

1. After writing to LINK_LMP_TX_DATA2, the hardware automatically generates a trigger signal to transmit the information from DATA0/DATA1/DATA2.

2. Custom HP transmission is used for sending undefined packets from registers such as LMP (TX_PORT_CAP/TX_PORT_CFG, etc.) and PING-TP.

Chapter 20 USB PD Controller (USBPD)

20.1 USB PD Controller Introduction

The chip has built-in USB Power Delivery controller and PD transceiver PHY, supports USB Type-C master-slave detection, automatic BMC codec and CRC, hardware edge control.

Supports USB PD 2.0, PD 3.0, and PD 3.2 Power Delivery control, supports SPR and EPR, supports 100W or 240W fast charging, supports PD sink and PD source applications, and DRP applications. Supports PDUSB, supports UFP and DFP, and DRD applications.

Supports PD packets including SOP, SOP', and SOP", supports USB PD reset signal frame hardware reset, supports a maximum packet length of 510 bytes, and supports DMA.

20.2 Register Description

Table 20-1 USBPD-related registers

Name	Access address	Description	Reset value
R32_USBPD_CONFIG	0x40024400	PD configuration register	0x00000002
R16_CONFIG	0x40024400	PD interrupt enable register	0x0002
R16_BMC_CLK_CNT	0x40024402	BMC sampling clock counter	0x0000
R32_USBPD_CONTROL	0x40024404	PD control register	0x00000000
R16_CONTROL	0x40024404	PD transceiver control register	0x0000
R8_CONTROL	0x40024404	PD transceiver enable register	0x00
R8_TX_SEL	0x40024405	PD transmit SOP selection register	0x00
R16_BMC_TX_SZ	0x40024406	PD transmit length register	0x0000
R32_USBPD_STATUS	0x40024408	PD status register	0x000000XX
R16_STATUS	0x40024408	PD interrupt and data register	0x00XX
R8_DATA_BUF	0x40024408	DMA cache data register	0xXX
R8_STATUS	0x40024409	PD interrupt flag register	0x00
R16_BMC_BYTE_CNT	0x4002440A	Byte counter	0x0000
R32_USBPD_PORT	0x4002440C	Port control register	0x00030003
R16_PORT_CC1	0x4002440C	CC1 port control register	0x0003
R16_PORT_CC2	0x4002440E	CC2 port control register	0x0003
R32_USBPD_DMA	0x40024410	DMA cache address register	0x00XXXXXX

20.2.1 PD Configuration Register (R32_USBPD_CONFIG)

Offset address: 0x00

Bit	Name
[31:16]	R16_BMC_CLK_CNT
[15:0]	R16_CONFIG

20.2.2 PD Interrupt Enable Register (R16_CONFIG)

Offset address: 0x00

Bit	Name	Access	Description	Reset value
15	IE_TX_END	RW	End-of-transmit interrupt enable: 1: Enable; 0: Disable.	0
14	IE_RX_RESET	RW	Receive reset interrupt enable: 1: Enable; 0: Disable.	0
13	IE_RX_ACT	RW	Receive completion interrupt enable: 1: Enable; 0: Disable.	0
12	IE_RX_BYTE	RW	Receive byte interrupt enable: 1: Enable; 0: Disable.	0
11	IE_RX_BIT	RW	Receive bit interrupt enable: 1: Enable; 0: Disable.	0
10	IE_PD_IO	RW	PD IO interrupt enable: 1: Enable; 0: Disable.	0
9	RX_5BIT	RO	Status indication of PD receiving 5 valid data bits: 1: 5 consecutive valid data bits have been received; 0: Not received or the data bits are discontinuous.	x
8	RX_MULTI_0	RO	Continuous reception of multiple zero indicator bits: 1: Continuous reception of multiple zeros; 0: No continuous reception of multiple zeros.	0
7	PRE_KEEP	RW	Preamble sending status keeping control bit: 1: Keep the preamble sending status; 0: Cancel the preamble sending keeping status, and exit the preamble sending status after 64 bits.	0
6	Reserved	RO	Reserved	0
5	WAKE_POLAR	RW	PD port wake-up level: 1: Active high; 0: Active low.	0
4	PD_RST_EN	RW	The PD mode reset command enable: 1: Reset; 0: Invalid.	0
3	PD_DMA_EN	RW	Enable the DMA of USBPD, which must be set to 1 in normal transfer mode:	0

			1: Enable DMA function and DMA interrupt; 0: Disable DMA.	
2	CC_SEL	RW	Select the current PD communication port: 1: Use CC2 port to communicate; 0: Use CC1 port to communicate.	0
1	PD_ALL_CLR	RW	PD mode clears all interrupt flag bits: 1: Clear the interrupt flag bit; 0: Invalid.	1
0	PD_FILT_EN	RW	Control the input filter enable for the PD pin: 1: Enable filtering; 0: Disable filtering.	0

20.2.3 BMC Sampling Clock Counter (R16_BMC_CLK_CNT)

Offset address: 0x02

Bit	Name	Access	Description	Reset value
[15:9]	Reserved	RO	Reserved	0
[8:0]	BMC_CLK_CNT	RW	BMC transmits or receives sampling clock counters.	0

20.2.4 PD Control Register (R32_USBPD_CONTROL)

Offset address: 0x04

Bit	Name
[31:16]	R16_BMC_TX_SZ
[15:0]	R16_CONTROL

20.2.5 PD Transceiver Control Register (R16_CONTROL)

Offset address: 0x04

Bit	Name
[15:8]	R8_TX_SEL
[7:0]	R8_CONTROL

20.2.6 PD Transceiver Enable Register (R8_CONTROL)

Offset address: 0x04

Bit	Name	Access	Description	Reset value
7	BMC_BYTE_HI	RO	Indicates the current half-byte status during PD data transmission/reception: 1: Processing upper 4 bits; 0: Processing lower 4 bits.	0
6	TX_BIT_BACK	RO	Indicates the bit status when the current BMC is transmitting code: 1: Indicates BMC byte transmission in progress; 0: Idle.	0
5	DATA_FLAG	RO	Cache Data Validity Flag.	0

[4:2]	RX_STATE[2:0]	RO	PD Receive Status Indicator 000: Initial receive state; 001: Start receiving SOP; 010: Receive Reset; 011: Receive SOP 100: End of receive; 101: Receive not used; 110: Receive EOP; 111: Receive byte.	0
1	BMC_START	RW	BMC transmits a start signal.	0
0	PD_TX_EN	RW	USBPD transceiver mode and transmit enable: 0: PD receive enable. 1: PD transmit enable.	0

20.2.7 PD Transmit SOP Selection Register (R8_TX_SEL)

Offset address: 0x05

Bit	Name	Access	Description	Reset value
[7:6]	TX_SEL4[1:0]	RW	Select the K-CODE4 type in PD transmitting mode: 00: SYNC2; 01: SYNC3; 1x: RST2.	0
[5:4]	TX_SEL3[1:0]	RW	Select the K-CODE3 type in PD transmitting mode: 00: SYNC1; 01: SYNC3; 1x: RST1.	0
[3:2]	TX_SEL2[1:0]	RW	Select the K-CODE2 type in PD transmitting mode: 00: SYNC1; 01: SYNC3; 1x: RST1.	0
1	Reserved	RO	Reserved	0
0	TX_SEL1	RW	Select the K-CODE1 type in PD transmitting mode: 1：RST1; 0：SYNC1.	0

20.2.8 PD Transmit Length Register (R16_BMC_TX_SZ)

Offset address: 0x06

Bit	Name	Access	Description	Reset value
[15:9]	Reserved	RO	Reserved	0
[8:0]	BMC_TX_SZ	RW	The total length transmitted in PD mode.	0

20.2.9 PD Status Register (R32_USBPD_STATUS)

Offset address: 0x08

Bit	Name
[31:16]	R16_BMC_BYTE_CNT
[15:0]	R16_STATUS

20.2.10 PD Interrupt and Data Register (R16_STATUS)

Offset address: 0x08

Bit	Name
[15:8]	R8_STATUS
[7:0]	R8_DATA_BUF

20.2.11 DMA Cache Data Register (R8_DATA_BUF)

Offset address: 0x08

Bit	Name	Access	Description	Reset value
[7:0]	DATA_BUF	RO	DMA cache data	X

20.2.12 PD Interrupt Flag Register (R8_STATUS)

Offset address: 0x09

Bit	Name	Access	Description	Reset value
7	IF_TX_END	RW1Z	Transfer completed interrupt flag, write 1 clear 0, write 0 invalid.	0
6	IF_RX_RESET	RW1Z	Receive reset interrupt flag, write 1 clear 0, write 0 invalid.	0
5	IF_RX_ACT	RW1Z	Transfer completed interrupt flag, write 1 clear 0, write 0 invalid.	0
4	IF_RX_BYTE	RW1Z	Receive byte or SOP interrupt flag, write 1 clear 0, write 0 is invalid.	0
3	IF_RX_BIT	RW1Z	Receive bit or 5bit interrupt flag, write 1 clear 0, write 0 invalid.	0
2	BUF_ERR	RW1Z	BUFFER or DMA error interrupt flag, write 1 clear 0, write 0 invalid.	0
[1:0]	BMC_AUX[1:0]	RO	Indicates the current PD status: When the PD is received or after the reception is completed, the status is as follows: 00: Receiving idle or no valid packet received; 01: SOP received i.e. SOP0; 10: SOP' received i.e. SOP1 or Hard Reset; 11: SOP' received i.e. SOP2 or Cable Reset. When PD is transmitting, the status is as follows: 00: CRC32[7:0] is being transmitted; 01: CRC32[15:8] is being transmitted; 10: CRC32[23:16] is being transmitted; 11: CRC32[31:24] is being transmitted.	0

20.2.13 Byte Counter (R16_BMC_BYTE_CNT)

Offset address: 0x0A

Bit	Name	Access	Description	Reset value
[15:9]	Reserved	RO	Reserved	0
[8:0]	BMC_BYTE_CNT	RO	Byte counter.	0

20.2.14 Port Control Register (R32_USBPD_PORT)

Offset address: 0x0C

Bit	Name
[31:16]	R16_PORT_CC2
[15:0]	R16_PORT_CC1

20.2.15 CC1 Port Control Register (R16_PORT_CC1)

Offset address: 0x0C

Bit	Name	Access	Description	Reset value
[15:8]	Reserved	RO	Reserved	0
[7:5]	CC1_CE	RW	Enable the CC1 port voltage comparator: 000: Off. 001: Reserved 010: 0.22V; 011: 0.43V; 100: 0.55V; 101: 0.66V; 110: 0.96V; 111: 1.23V.	0
4	CC1_LVE	RW	The CC1 port outputs low voltage enable: 0: Normal V_{DD} voltage drive output. 1: Low voltage drive output.	0
[3:2]	CC1_PU	RW	CC1 port pull-up current selection: 00: No pull-up current 01: 330uA; 10: 180uA; 11: 80uA.	0
1	CC1_PD	RW	CC1 port pull-down resistor Rd enable: 1: Enable Rd pull-down resistor, about 5.1KΩ; 0: Disable the pull-down resistor. *Note: Approximately 600kΩ of weak pull-down remains after shutdown.*	1
0	PA_CC1_AI	RO	The CC1 port comparator simulates input.	1

20.2.16 CC2 Port Control Register (R16_PORT_CC2)

Offset address: 0x0E

Bit	Name	Access	Description	Reset value
[15:8]	Reserved	RO	Reserved	0
[7:5]	CC2_CE	RW	Enable the CC2 port voltage comparator: 000: Off. 001: Reserved 010: 0.22V;	0

			011: 0.43V; 100: 0.55V; 101: 0.66V; 110: 0.96V; 111: 1.23V.	
4	CC2_LVE	RW	The CC2 port outputs low voltage enable: 1: Low voltage drive output; 0: Normal V_{DD} voltage drive output.	0
[3:2]	CC2_PU	RW	CC2 port pull-up current selection: 00: No pull-up current 01: 330uA; 10: 180uA; 11: 80uA.	0
1	CC2_PD	RW	CC2 port pull-down resistor Rd enable: 1: Enable Rd pull-down resistor, about 5.1KΩ; 0: Disable pull-down resistor. *Note: about 600kΩ weak pull-down remains after disabling*	1
0	PA_CC2_AI	RO	The CC2 port comparator Analog input.	1

20.2.17 DMA Cache Address Register (R32_USBPD_DMA)

Offset address: 0x10

Bit	Name	Access	Description	Reset value
[31:16]	Reserved	RO	Reserved	0
[15:0]	USBPD_DMA_ADDR	RW	USBPD_DMA cache address. The low 16-bit is valid and the address must be 4-byte aligned.	X

Chapter 21 Debug Support (DBG)

21.1 Main Features

This register allows the MCU to be configured in the debug state. Includes:

- Counters supporting Independent Watchdog (IWDG)
- Counters supporting Window Watchdog (WWDG)
- Counter supporting timer

21.2 Register Description

21.2.1 Debug MCU Configuration Register (DBGMCU_CR)

Offset address: 0x7C0(CSR)

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TIM4_STOP	TIM3_STOP	TIM2_STOP	TIM1_STOP	Reserved		WWDG_STOP	IWDG_STOP	Reserved						STOP	SLEEP

Bit	Name	Access	Description	Reset value
[31:16]	Reserved	RO	Reserved	0
15	TIM4_STOP	RW	Timer 4 debug stop bit. When the core enters debug mode, the counter stops functioning. 1: Timer 4 counter stops working; 0: Timer 4 counter continues normal operation.	0
14	TIM3_STOP	RW	Timer 3 debug stop bit. When the core enters debug mode, the counter stops functioning. 1: Timer 3 counter stops working; 0: Timer 3 counter continues normal operation.	0
13	TIM2_STOP	RW	Timer 2 debug stop bit. When the core enters debug mode, the counter stops functioning. 1: Timer 2 counter stops working; 0: Timer 2 counter continues normal operation.	0
12	TIM1_STOP	RW	Timer 1 debug stop bit. When the core enters debug mode, the counter stops functioning. 1: Timer 1 counter stops working; 0: Timer 1 counter continues normal operation.	0
[11:10]	Reserved	RO	Reserved	0
9	WWDG_STOP	RW	WWDG debug stop bit. The debug WWDG stops working when the core enters the debug state. 1: WWDG counter stops working. 0: WWDG counter continues normal operation.	0

8	IWDG_STOP	RW	IWDG debug stop bit. The debug IWDG stops working when the core enters the debug state. 1: IWDG counter stops working. 0: IWDG counter continues normal operation.	0
[7:2]	Reserved	RW	Reserved	0
1	STOP	RW	Debug stop mode bit. 1: (FCLK on, HCLK on) In Stop mode, the FCLK and HCLK clocks are provided by the internal RC oscillator. When exiting stop mode, the software must reconfigure the clock system to start the PLL, crystal oscillator, etc. (The same operation as when configuring this bit to 0); 0: (FCLK off, HCLK off) In stop mode, the clock controller disables all clocks (including HCLK and FCLK). When exiting STOP mode, the clock configuration is the same as after reset (The microcontroller is clocked by the 8MHz internal RC oscillator (HSI)). Therefore, the software must reconfigure the clock control system to start the PLL, crystal oscillator, etc.	0
0	SLEEP	RW	Debug sleep mode bit. 1: (FCLK on, HCLK on) In sleep mode, both FCLK and HCLK clocks are provided by the originally configured system clock; 0: (FCLK on, HCLK off) In sleep mode, FCLK is provided by the originally configured system clock, and HCLK is off. Since sleep mode does not reset the configured clock system, software does not need to reconfigure the clock system when exiting from sleep mode.	0

Chapter 22 Electronic Signature (ESIG)

Electronic Signature contains chip identification information: Flash memory area capacity and unique identification. It is burned into the system storage area of the memory module by the manufacturer when leaving the factory, and can be read through SWD (SDI) or application code.

22.1 Function Description

Flash memory area capacity: Indicates the size that the current chip user application can use.

Unique identification: 96-bit binary code, unique to any microcontroller, users can only read and access it and cannot modify it. This unique identification information can be used as the security password, decryption key, product serial number, etc. of the microcontroller (product) to improve the system security mechanism or indicate identity information.

Users of the above content can perform read access in 8/16/32 bits.

22.2 Register Description

Table 22-1 ESIG-related register

Name	Access address	Description	Reset value
R16_ESIG_FLACAP	0x1FFFF7E0	Flash capacity register	0xXXXX
R32_ESIG_UNIID1	0x1FFFF7E8	UID register 1	0xXXXXXXXX
R32_ESIG_UNIID2	0x1FFFF7EC	UID register 2	0xXXXXXXXX
R32_ESIG_UNIID3	0x1FFFF7F0	UID register 3	0xXXXXXXXX

22.2.1 Flash Capacity Register (ESIG_FLACAP)

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

Bit	Name	Access	Description	Reset value
[15:0]	F_SIZE[15:0]	RO	Flash memory capacity in Kbyte. Example: 0x0080 = 128 Kbytes	X

22.2.2 UID Register (ESIG_UNIID1)

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

Bit	Name	Access	Description	Reset value
[31:0]	U_ID[31:0]	RO	Bits 0-31 of the UID.	X

22.2.3 UID Register (ESIG_UNIID2)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
U_ID[63:48]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
U_ID[47:32]															

Bit	Name	Access	Description	Reset value
[31:0]	U_ID[63:32]	RO	Bits 32-63 of the UID.	X

22.2.4 UID Register (ESIG_UNIID3)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
U_ID[95:80]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
U_ID[79:64]															

Bit	Name	Access	Description	Reset value
[31:0]	U_ID[95:64]	RO	Bits 64-95 of the UID.	X

Chapter 23 Flash Memory and User Option Bytes (FLASH)

23.1 Flash Memory Organization

The flash memory organization inside the chip is as follows:

Table 23-1 Flash organizational structure

Block	Name	Address range	Size (byte)
Main memory	Page 0	0x08000000 – 0x080000FF	256
	Page 1	0x08000100 – 0x080001FF	256
	Page 2	0x08000200 – 0x080002FF	256
	Page 3	0x08000300 – 0x080003FF	256
	Page 4	0x08000400 – 0x080004FF	256
	Page 5	0x08000500 – 0x080005FF	256
	Page 6	0x08000600 – 0x080006FF	256
	Page 7	0x08000700 – 0x080007FF	256
	…	…	…
	Page 1919	0x08077F00 – 0x08077FFF	256
Information block	Boot program code	0x1FFF8000 – 0x1FFFEFFF	28K
	User option bytes	0x1FFFF800 – 0x1FFFF87F	128

Note: The above main memory area is used for the user's application storage, and the write protection is divided in 4K bytes (16 pages) units; except that the "manufacturer configuration word" area is factory locked, the user is inaccessible. Other areas can be operated by users under certain conditions.

23.2 Flash Memory Programming and Safety

23.2.1 Two Program/Erase Methods

- Standard Programming: This is the default programming mode (compatibility mode). In this mode, the CPU performs programming in a single 2-byte manner, and performs erase and whole-chip erase operations in a single 4K-byte manner.
- Fast Programming: This method employs a page operation mode (recommended). After a specific sequence is unlocked, a single 256-byte programming, 32K-byte erasing, and entire chip erasing are performed.

23.2.2 Security-preventing against Illegal Access (read, write and erase)

- Page write protection
- Read protection

When the chip is in read protection:

1) Main memory pages 0-15 (4K bytes) are automatically write-protected and are not controlled by the FLASH_WPR register; when the read-protected state is released, all main memory pages are controlled by the FLASH_WPR register.

2) The system boot code area, SWD or SDI mode, and RAM area cannot erase or program the main memory, except for full chip erasure. User selectable word area can be erased or programmed. If you try to remove read protection (program user words), the chip will automatically erase the entire user area.

23.3 FLASH Enhanced Read Mode

FLASH Enhanced Read Mode is suitable for user programs to run in FLASH, which can improve the access efficiency of FLASH. To enable this mode, set bit 1 of the EHMOD field in the FLASH_ACTLR register. Then, read the ENHANCE_STATUS bit; if bit 1 of ENHANCE_STATUS is set, the enhanced read mode is active. To disable this mode, first clear bit 1 of EHMOD, then set RSENACT to 1.

Note: When using FLASH enhanced reading mode, you should pay attention to the following points:

1) Before performing any erase or program operations on the FLASH (including user programming such as read protection removal), you must first exit Enhanced Read Mode. Failure to do so will cause erase and program operations to fail.

2) Before entering Stop Mode, you must first exit Enhanced Read Mode. Failure to do so may cause Stop Mode to malfunction.

3) After a power reset or system reset completes, the chip automatically exits Enhanced Read Mode under hardware control.

23.4 Register Description

Table 23-2 FLASH-related registers

Name	Access address	Description	Reset value
R32_FLASH_ACTLR	0x40022000	Access control register	0x0000X000
R32_FLASH_KEYR	0x40022004	FPEC key register	0xXXXXXXXX
R32_FLASH_OBKEYR	0x40022008	OBKEY register	0xXXXXXXXX
R32_FLASH_STATR	0x4002200C	Status register	0x00008000
R32_FLASH_CTLR	0x40022010	Control register	0x00008080
R32_FLASH_ADDR	0x40022014	Address register	0x00000000
R32_FLASH_OBR	0x4002201C	Option byte register	0x3FFFFFFE
R32_FLASH_WPR	0x40022020	Write protection register	0xXXXXXXXX
R32_FLASH_MODEKEYR	0x40022024	Extension key register	0xXXXXXXXX
R32_BOOT_MODEKEYR	0x40022028	BOOT key register	0xXXXXXXXX

23.4.1 Access Control Register (FLASH_ACTLR)

Offset address: 0x00

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FLASH_ST	FLASH_READY	Reserved		FLASH_RD_MD	Reserved		FLASH_LP	EHMOD	ENHANCE_STATUS	Reserved				SCK_CFG	

Bit	Name	Access	Description	Reset value
[31:16]	Reserved	RO	Reserved	0

15	FLASH_ST	RO	FLASH low-power status indicator bit: 1: FLASH is in low-power mode; 0: FLASH is in normal operating mode.	0
14	FLASH_READY	RO	FLASH ready indicator bit: 1: FLASH is ready; 0: FLASH is not ready.	X
[13:12]	Reserved	R0	Reserved	0
11	FLASH_RD_MD	RW	FLASH read mode control bit: 1: Continuous read mode; 0: Single read mode. *This bit enables standard unlock + fast unlock.*	0
[10:9]	Reserved	R0	Reserved	0
8	FLASH_LP	RW	FLASH low-power mode control bit: 1: FLASH low-power mode enabled; 0: FLASH low-power mode disabled.	0
7	EHMOD	RW	FLASH enhanced read mode control bit: This mode improves access efficiency when the program runs from FLASH. 1: Enables FLASH Enhanced Read Mode; 0: Disables FLASH Enhanced Read Mode. Requires coordination with the RSENACT bit. To exit, first clear the EHMOD bit to 0, then set RSENACT to 1. *This bit enables standard unlock + fast unlock.*	0
6	ENHANCE_STATUS	RO	Enhanced status: 1: Command transmission completed; 0: Waiting.	0
[5:2]	Reserved	RO	Reserved	0
[1:0]	SCK_CFG	RW	FLASH wait state count: 00: HCLK; FLASH read operation -- Recommended HCLK ≤ 75MHz, FLASH erase/write operation -- Recommended HCLK ≤ 60MHz. 01: HCLK/2; FLASH read operation -- Recommended HCLK ≤ 150MHz, FLASH erase/write Operation -- Recommended HCLK ≤ 120MHz. 10: HCLK/4; FLASH read operation -- HCLK maximum is 300MHz, not recommended for this setting, FLASH erase/write operation -- Recommended HCLK ≤ 240MHz.	0

			10: HCLK/8; FLASH read operation -- HCLK maximum is 312.5MHz, not recommended for this setting, FLASH erase/write operation -- Recommended HCLK ≤ 312.5MHz. *Note: The above FLASH operation clock must ensure that FLASH_RD_MD is configured to 1.*	

23.4.2 FPEC Key Register (FLASH_KEYR)

Offset address: 0x04

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

Bit	Name	Access	Description	Reset value
[31:0]	KEYR[31:0]	WO	FPEC key, the unlock key used to enter FPEC includes: RDPRT key = 0x000000A5; KEY1 = 0x45670123; KEY2 = 0xCDEF89AB.	X

23.4.3 OBKEY Register (FLASH_OBKEYR)

Offset address: 0x08

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

Bit	Name	Access	Description	Reset value
[31:0]	OBKEYR[31:0]	WO	Selection word key for entering a selection word key to disarm OBWRE.	X

23.4.4 Status Register (FLASH_STATR)

Offset address: 0x0C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BOOT_LOCK	BOOT_MODE	BOOT_STATUS	BOOT_AVA	BOOT_LOADER	Reserved					EOP	WRPRTERR	Reserved		WRBSY	BSY

Bit	Name	Access	Description	Reset value
[31:16]	Reserved	RO	Reserved	0
15	BOOT_LOCK	RW1	BOOT area lock: 1: Locked; 0: Unlocked.	1
14	BOOT_MODE	RW	When combined with BOOT_AVA, it controls switching between the user area and BOOT area: 1: After a software reset, it can switch to either the BOOT area or the user area; 0: After a software reset, it can switch to the user area.	0
13	BOOT_STATUS	RW0	Source of the currently executing program, write 0 to reset: 1: Indicates a program loaded from the BOOT area; 0: Indicates a program loaded from the user area.	0
12	BOOT_AVA	RO	Program initialization state: 1: Indicates booting from the BOOT sector; 0: Indicates booting from the user sector.	0
11	BOOT_LOADER	RO	The current program running position indication bit 1: Running in the BOOT area; 0: Running in the user area.	0
[10:6]	Reserved	RO	Reserved	0
5	EOP	RW1	Indicates the end of the operation, write 1 to clear. The hardware is set every time it is successfully erased or programmed.	0
4	WRPRTERR	RW1	Indicates write protection error, write 1 to clear. If you program a write-protected address, the hardware will set it.	0
[3:2]	Reserved	RO	Reserved	0
1	WRBSY	RO	This bit is used during fast page programming to indicate that programming data is being written. During page programming, when writing data, this bit is set to '1' and the hardware automatically clears' 0'; If this bit is '0', it means that writing the next data is allowed.	0
0	BSY	RO	Indicates busy status: 1: Indicates that a flash operation is in progress; 0: The operation is finished.	0

Note: When performing programming operations, you need to ensure that the STRT bit of the FLASH_CTLR register is 0.

23.4.5 Control Register (FLASH_CTLR)

Offset address: 0x10

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FLOCK	Reserved		EOPIE	Reserved	ERRIE	OBWRE	Reserved	LOCK	STRT	OBER	OBPG	Reserved		PER	PG

Bit	Name	Access	Description	Reset value
[31:23]	Reserved	RO	Reserved	0
22	RSENACT	WO	Exit the enhanced read mode, the hardware will automatically clear it, and it needs to be operated together with the EHMOD bit. In the exit step, first clear the EHMOD bit to 0, and then set RSENACT to 1. *This bit enables standard unlock.*	0
21	PGSTRT	WO	Start. Set to 1 to start a page programming, and the hardware will be cleared automatically. *This bit enables standard unlock.*	0
[20:19]	Reserved	RO	Reserved	0
18	BER	RW	Perform a 32KB erase. *This bit enables standard unlock + fast unlock.*	0
17	Reserved	RO	Reserved	0
16	FTPG	RW	Perform fast page programming operation. *This bit enables standard unlock + fast unlock.*	0
15	FLOCK	RW1	Fast Program Lock. Writes only '1'. When this bit is '1', quick program/erase mode is disabled. Hardware clears this bit to '0' upon detecting the correct unlock sequence. Software sets to '1' to re-lock. *This bit enables standard unlock.*	1
[14:13]	Reserved	RO	Reserved	0
12	EOPIE	RW	Interrupt control after operation completion (EOP bit set in FLASH_STATR register): 1: Enable interrupt generation; 0: Disable interrupt generation. *This bit enables standard unlock.*	0
11	Reserved	RO	Reserved	0
10	ERRIE	RW	Error status interrupt control (PGERR/WRPRTERR set in FLASH_STATR register): 1: Allows interrupt generation; 0: Disables interrupt generation. *This bit enables standard unlock.*	0
9	OBWRE	RW0	User option word lock, software reset: 1: Indicates the user-selected word is programmable. Requires writing the correct sequence to the FLASH_OBKEYR register	0

			followed by hardware setting; 0: User-selected word relocks after software reset. *This bit enables standard unlock.*	
8	Reserved	RO	Reserved	0
7	LOCK	RW1	Lock. Writable only as '1'. When this bit is '1', it indicates that FPEC and FLASH_CTLR are locked and cannot be written. After detecting the correct unlock sequence, the hardware clears this bit to '0'. Following an unsuccessful unlock operation, this bit remains unchanged until the next system reset. *This bit enables standard unlock.*	1
6	STRT	RW1	Start. Set to 1 to start an erase operation, and the hardware automatically clears 0(BSY changes to' 0'). *This bit enables standard unlock.*	0
5	OBER	RW	User option word erasure is performed. *This bit enables standard unlocking.*	0
4	OBPG	RW	Perform user option word programming. *This bit enables standard unlock.*	0
3	Reserved	RO	Reserved	0
2	MER	RW	Perform a full erase operation (erase the entire user area). *This bit enables standard unlock.*	0
1	PER	RW	Perform a sector (4KB) erase operation. *This bit enables standard unlock.*	0
0	PG	RW	Perform standard programming operations. *This bit enables standard unlock.*	0

23.4.6 Address Register (FLASH_ADDR)

Offset address: 0x14

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

Bit	Name	Access	Description	Reset value
[31:0]	FAR	RW	The flash address, which is the programming address for programming and the starting address for erasure. The register cannot be written when the BSY bit	0

			in the FLASH_STATR register is '1'.	

23.4.7 Option Byte Register (FLASH_OBR)

Offset address: 0x1C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved						DATA1								DATA0	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA0						USARTDLEN	USBHSDLEN	Reserved	START_MD	CFGRSTEN	Reserved		IWDGSW	RDPRT	OBERR

Bit	Name		Access	Description	Reset value
[31:26]	Reserved		RO	Reserved	0
[25:18]	DATA1		RO	Data byte 1	0xFF
[17:10]	DATA0		RO	Data byte 0	0xFF
9	USER	USARTDLEN	RO	BOOT enables USART keyless download functionality: 1: Enabled; 0: Disabled.	1
8		USBFSDLEN	RO	BOOT enables USBFS download functionality: 1: Enabled; 0: Disabled.	1
7		Reserved	RO	Reserved	1
6		START_MD	RO	Startup mode configuration: 1: When there is BOOT, the system starts from the BOOT area after power on; 0: When there is BOOT, the system starts from the user area after power on.	1
5		CFGRSTEN	RO	External pin reset enable: 1: Off; 0: On.	1
[4:3]		Reserved	RO	Reserved	11b
2		IWDG_SW	RO	Independent watchdog (IWDG) hardware enable bit: 1: The IWDG function is enabled by software and is prohibited from being enabled by hardware; 0: The IWDG function is enabled by hardware (determined by the HSI clock).	1
1	RDPRT		RO	Read protection status. 1: Indicates that the current read protection of	1

			the flash memory is valid; 0: Indicates that the current read protection of the flash memory is invalid.	
0	OBERR	RO	Option byte error. 1: Indicates that the option byte does not match its complement; 0: Indicates that the option byte matches its complement.	0

Note: USER and RDPRT are loaded from the user option byte area after system reset.

23.4.8 Write Protection Register (FLASH_WPR)

Offset address: 0x20

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

Bit	Name	Access	Description	Reset value
[31:0]	WRP[31:0]	RO	Flash memory write protection status. 1: Write protection disabled; 0: Write protection enabled. Each bit represents 4K bytes (16 pages) of storage write protection status.	X

Note: WPR is loaded from the user option byte area after system reset.

23.4.9 Extension Key Register (FLASH_MODEKEYR)

Offset address: 0x24

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

Bit	Name	Access	Description	Reset value
[31:0]	MODEKEYR[31:0]	WO	Enter the following sequence to unlock the fast program/erase mode: KEY1 = 0x45670123; KEY2 = 0xCDEF89AB.	X

23.4.10 BOOT Key Register (BOOT_MODEKEYR)

Offset address: 0x28

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

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

Bit	Name	Access	Description	Reset value
[31:0]	BOOTKEYR[31:0]	WO	Enter the following sequence to unlock the fast program/erase mode: KEY1 = 0x45670123; KEY2 = 0xCDEF89AB.	X

23.5 Flash Operation Procedure

23.5.1 Read Operation

Direct addressing is in the general address space, and the user can access the content of the flash memory module and get the corresponding data through any read operation of 8/16/32-bit data.

23.5.2 Flash Memory Unlock

After the system reset, the flash memory controller (FPEC) and FLASH_CTLR register will be locked and cannot be accessed. The flash memory controller module can be unlocked by writing the sequence to the FLASH_KEYR register.

Unlock sequence:
1) Write KEY1 = 0x45670123 to the FLASH_KEYR register (must operate KEY1 first);
2) Write KEY2 = 0xCDEF89AB to the FLASH_KEYR register (must operate KEY2 secondly).

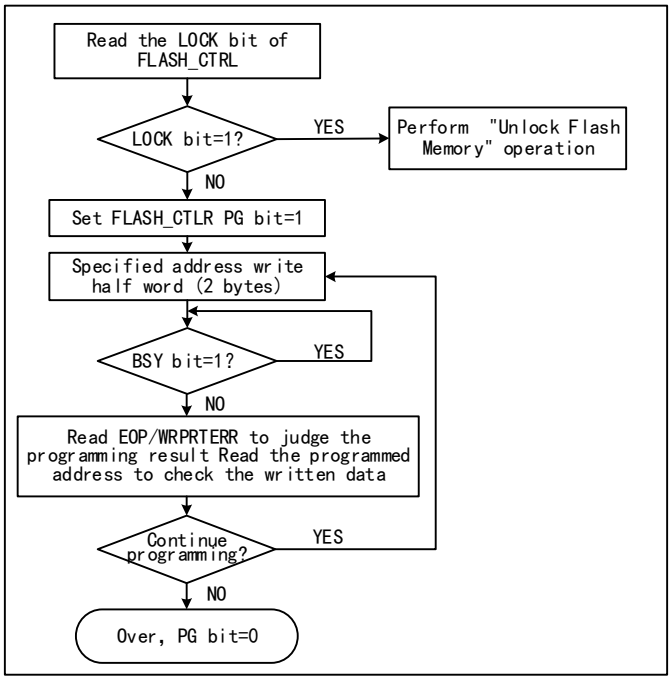
The above operations must be performed sequentially and continuously. Otherwise, it is an error operation, which will lock the FPEC module and FLASH_CTLR register and generate a bus error until the next system reset.

The flash memory controller (FPEC) and the R32_FLASH_CTLR register can be locked again by setting the "LOCK" bit in the R32_FLASH_CTLR register to 1.

23.5.3 Main Memory Standard Program

Standard programming writes 2 bytes per operation. When the PG bit in the R32_FLASH_CTLR register is set to ‘1’, each write of a half-word (2 bytes) to a flash address initiates one programming cycle. Writing any non-half-word data causes FPEC to generate a bus error. During programming, the BSY bit is ‘1’; upon completion, BSY becomes ‘0’ and EOP becomes ‘1’.
Note: When the BSY bit is ‘1’, write operations to any register are prohibited.

Figure 23-1 FLASH program



1) Check the LOCK bit in the FLASH_CTLR register. If it is 1, you need to perform the "Release Flash Memory Lock" operation.

2) Set the PEG bit in the FLASH_CTLR register to ‘1’ to enable the standard page erasure mode.

3) Write the half-word to be programmed to the specified flash memory address (even address).

4) Wait for the BSY bit to become ‘0’ or for the EOP bit in the R32_FLASH_STATR register to become ‘1’, indicating programming completion. Then clear the EOP bit to ‘0’.

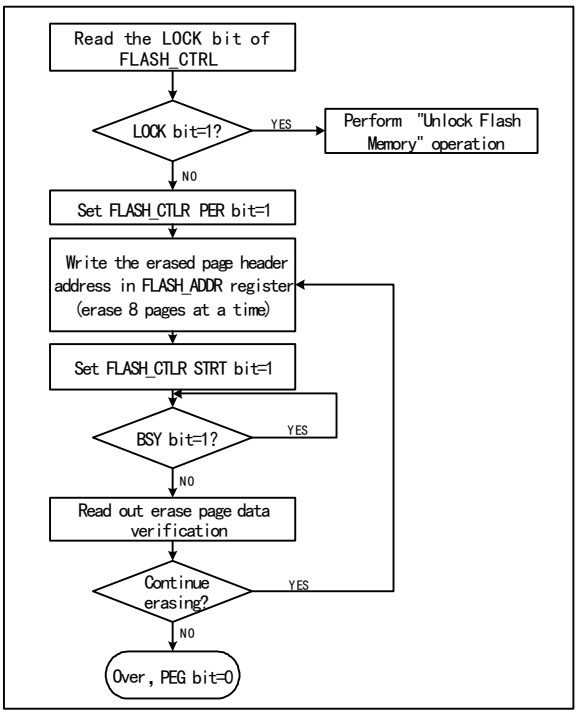
5) Check the R32_FLASH_STATR register for errors, or read the programmed address data for verification.

6) To continue programming, repeat steps 3-5. To end programming, clear the PG bit.

23.5.4 Main Memory Standard Erase

Flash memory can be erased by sectors (4K bytes) or in its entirety.

Figure 23-2 FLASH page erasure



1) Check the LOCK bit in the FLASH_CTLR register. If it is 1, you need to perform the "Release Flash Memory Lock" operation.

2) Set the PEG bit in the FLASH_CTLR register to ‘1’ to enable the sector erasure mode.

3) Write the page heading address of the page to be erased to the FLASH_ADDR register.

4) Set the STAT bit in the FLASH_CTLR register to ‘1’ to start an erase action.

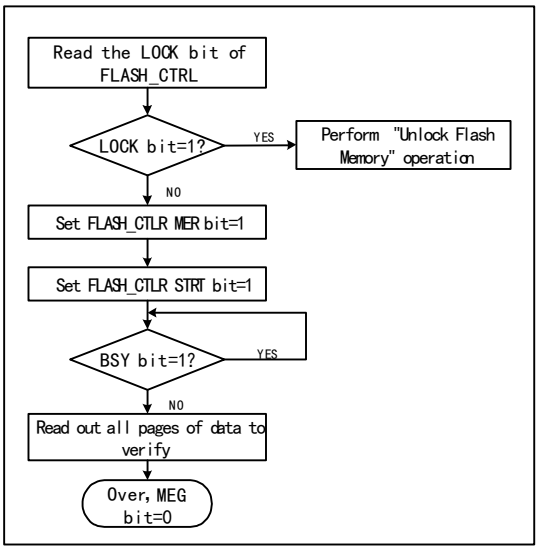
5) When the BYS bit changes to '0' or the EOP bit in the FLASH_STATR register to be '1', it indicates the end of erasure. Clear the EOP bit to 0.

6) Read the page of erasure page for verification.

7) To erase the sector continuously, you can repeat steps 3-5 to end erasing and clear the PEG bit to 0.

Note: After successful erasure, words read -0xe339e339, half words read -0xe339, even address bytes read -0x39, and odd addresses read 0xe3.

Figure 23-3 FLASH entire chip erase



1) Check the LOCK bit of the FLASH_CTLR register. If it is 1, you need to perform the "unlock flash lock" operation.
2) Set the MER bit of the FLASH_CTLR register to ‘1’ to enable the whole chip erase mode.
3) Set the STRT bit of the FLASH_CTLR register to ‘1’ to start the erase action.
4) Wait for the BSY bit to become '0' or the EOP bit of the FLASH_STATR register to be '1' to indicate the end of erasure, and clear the EOP bit to 0.
5) Read the data of the erased page for verification.
6) Clear the MER bit to 0.

23.5.5 Fast Programming Mode Unlock

Fast programming mode operation can be unlocked by writing a sequence to the R32_FLASH_MODEKEYR register. After unlocking, the block bit of the R32_FLASH_CTLR register will be cleared to 0, indicating that quick erasure and programming operations can be performed. Lock again by setting the 'FLOCK' bit of the FLASH_CTLR register to 1.

Unlock sequence:
1) Write KEY1=0x45670123 to the FLASH_MODEKEYR register.
2) Write KEY2=0xCDEF89AB to the FLASH_MODEKEYR register.
The above operations must be performed sequentially and continuously, otherwise the error operation will be locked and cannot be unlocked until the next system reset.

Note: Fast programming requires unlocking the "LOCK" and "FLOCK" layers.

23.5.6 Main Memory Fast Programming

The fast programming (256 bytes) is made according to the page.

1) Check the LOCK bit of the R32_FLASH_CTLR register. If it is '1', you need to perform the "unlock flash" operation.
2) Check the R32_FLASH_CTLR register FLOCK bit, if it is '1', you need to perform a "quick programming mode unlock" operation.
3) Check the BSY bit of the R32_FLASH_STATR register to confirm that there are no other programming

operations in progress.
4) Set the FTPG bit of the R32_FLASH_CTLR register to '1' to enable fast page programming mode.
5) Write data to a FLASH address in a 32-bit manner, such as.
* (uint32_t*) 0x80000000=0x12345678.
6) Wait for WR_BSY of FLASH_STATR register to be '0', and write the next data.
7) Repeat steps 5-6 64 times
8) Set the FTPG bit in FLASH_CTLR register to' 1' to start fast page programming.
9) Waiting for the BSY bit to change to '0' or the EOP bit of the FLASH_STATR register to '1' means that a quick page programming is completed and the EOP bit is cleared to 0.
10) Query the FLASH_STATR register to see if there are any errors, or read the programming address data check.
11) Continue fast page programming to repeat steps 5-10 and end the programming to clear the FTPG bit 0.

23.5.7 Main Memory Fast Erasure

Fast Erase erases in blocks (32K bytes).
1) Check the FLASH_CTLR register LOCK bit, if it is '1', you need to perform the "unlock flash memory" operation.
2) Check the FLASH_CTLR register FLOCK bit, if it is '1', you need to perform a "fast programming mode unlock" operation.
3) Check the BSY bit of the FLASH_STATR register to confirm that there are no other programming operations in progress.
4) Set the BER32 bit of the FLASH_CTLR register to '1' to enable the fast block erase (32K bytes) mode function.
5) Write the first address of the fast erase page to the FLASH_ADDR register.
6) Set the STRT bit of the FLASH_CTLR register to '1' to start a fast block erase (32K bytes) action.
7) Wait for the BSY bit to become '0' or the EOP bit of the FLASH_STATR register to be '1' to indicate the end of erasure, and clear the EOP bit to 0.
8) Query the FLASH_STATR register to see if there is an error, or read the erase page address data to verify.
9) Continue to quickly erase the page and repeat steps 5-8 to end the erase to clear the BER32 bit 0.
Note: After successful erasure, the byte reads -0xe339e339, the half-byte reads -0xe339, the even-address byte reads -0x39, and the odd-address byte reads 0xe3.

23.6 User Option Bytes

The User Option Bytes are solidified in FLASH and will be reloaded into the corresponding register after the system reset, and the user can erase and program at will. The user option word information block has a total of 8 bytes (4 bytes for write protection, 1 byte for read protection, 1 byte for configuration options, 2 bytes for user data storage), and each bit has the inverted code bit for checking during loading. The structure and meaning of the selected word information are described below.

Table 23-3 32-bit option bytes format division

[31:24]	[23:16]	[15:8]	[7:0]
Inverse code of option bytes 1	Option bytes 1	Inverse code of option bytes 0	Option bytes 0

Table 23-4 User option bytes information structure

Address \ Bit	[31:24]	[23:16]	[15:8]	[7:0]
0x1FFFF800	nUSER	USER	nRDPR	RDPR
0x1FFFF804	nData1	Data1	nData0	Data0
0x1FFFF808	nWRPR1	WRPR1	nWRPR0	WRPR0
0x1FFFF80C	nWRPR3	WRPR3	nWRPR2	WRPR2

Name/Byte			Description	Reset value
RDPR			Read protection control. It configures whether the code in the flash memory can be read. 0xA5: If this byte is 0xA5 (nRDP must be 0x5A), it means that the current code is in a non-read protected state and can be read; Other values: Indicates the code read protection status, which is unreadable. Pages 0-15 (4K) will be automatically write-protected and not controlled by WRPR0.	0xA5
USER	7	USARTDLEN	BOOT enable USART keyless download functionality [1]: 1: Enable; 0: Disable.	1
	6	USBFSDLEN	BOOT enable USBFS download functionality [1]: 1: Enable; 0: Disable.	1
	5	Reserved	Reserved	1
	4	START_MD	Startup mode configuration: 1: When there is BOOT, the system starts from the BOOT area after power on; 0: When there is BOOT, the system starts from the user area after power on.	1
	3	CFGRSTEN	External pin reset enable: 1: Off; 0: On.	1
	[2:1]	Reserved	Reserved	11b
	0	IWDGSW	Independent watchdog (IWDG) hardware enable bit: 1: IWDG function is enabled by software, but hardware is not allowed. 0: IWDG function is turned on by the hardware (determined by the HSI clock).	1
Data0–Data1			Stores 2 bytes of user data.	0xFFFF
WRPR0 - WRPR3			Write protection control bit. Each bit is used to control the write protection status of 1 sector (4K bytes/sector) in main memory: 1: Disable write protection.	0xFFFFFFFF

	0: Enable write protection. 4 bytes are used to protect a total of 480K bytes of main memory. WRPO: Sector 0-7 storage write protection control; WRP1: Sector 8-15 storage write protection control; WRP2: Sector 16-23 storage write protection control; WRP3: Bits 0-6 provide write protection for sectors 24-30; Bit 7 provides write protection for sectors 31-119.	

Note [1]: Please refer to the instruction manual of CH32X315 evaluation board for the specific operation mode and precautions of this function.

23.6.1 User Option Bytes Unlock

The user option bytes operation can be unlocked by writing the sequence to the FLASH_OBKEYR register. After unlocking, the OBWRE bit in the FLASH_CTLR register will be set to 1, indicating that user option bytes can be erased and programmed. By setting the OBWRE bit in the FLASH_CTLR register, it will be cleared to 0 by software to lock again.

Unlocking sequence

1) Write KEY1 = 0x45670123 to the FLASH_OBKEYR register;

2) Write KEY2 = 0xCDEF89AB to the FLASH_OBKEYR register.

Note: The user needs to unlock the 2 layers: "LOCK" and "OBWRE" for word selection.

23.6.2 User Option Bytes Programming

Only supports standard programming, writing half word (2 bytes) at a time. In the actual process, when programming the user option byte, FPEC only uses the low byte in the half-word, and automatically calculates the high byte (the high byte is the complement of the low byte), and then starts the programming operation. This will ensure that the byte in the user option byte and its complement are always correct.

1) Check the LOCK bit in the FLASH_CTLR register. If it is set to 1, perform the "Unlock Flash" operation.

2) Check the BSY bit in the FLASH_STATR register to confirm no other programming operations are in progress.

3) Check the OBWRE bit in the FLASH_CTLR register. If it is 0, perform the 'User Option Byte Unlock' operation.

4) Set the OBPG bit in the FLASH_CTLR register to '1'.

5) Write the half-word (2 bytes) to be programmed to the specified address.

6) Wait for the BSY bit to become '0' or for the EOP bit in the FLASH_STATR register to become '1', indicating programming completion. Then clear the EOP bit.

7) Read the programming address data for verification.

8) To continue programming, repeat steps 5-7. To end programming, clear the OBPG bit.

Note: when "read protected" in the modified selection word is changed to 'unprotected' state, an entire erase main storage area operation is automatically performed. If you modify a selection other than 'read protected', the whole erase operation will not occur.

23.6.3 User Option Bytes Erasure

Erase the entire 128-byte user option bytes area directly.

1) Check the LOCK bit in the FLASH_CTLR register. If it is set to 1, perform the "Unlock Flash" operation.

2) Check the BSY bit in the FLASH_STATR register to confirm no programming operation is in progress.

3) Check the OBWRE bit in the FLASH_CTLR register. If it is 0, perform the "User Option Byte Unlock" operation.
4) Set the OBER bit of the FLASH_CTLR register to '1', and then set the STRT bit of the FLASH_CTLR register to '1' to enable user-selected word erasure.
5) Wait for the BSY bit to become ‘0’ or for the EOP bit in the FLASH_STATR register to become ‘1’ indicating erase completion, then clear the EOP bit to 0.
6) Read the erased address data for verification.
7) When finished, clear the OBER bit to 0.
Note: After successful erasure, the character reads -0xe339e339, the half-character reads -0xe339, and the byte reads -0x39.

23.6.4 Read Protection Release

Whether the flash memory is read-protected or not is determined by the user option byte. Read the FLASH_OBR register. When the RDPRT bit is'1', it indicates that the current flash memory is in read-protected state, and the flash operation is protected by a series of read-protected states. The process of unprotecting read is as follows:
1) Erase the entire user option byte area, read the protection field RDPR, and the read protection is still valid.
2) The user chooses words to program and writes the correct RDPR code 0xA5 to remove the read protection of the flash memory. (This step will first cause the system to automatically erase the entire piece of flash memory.).
3) A power-on reset is performed to reload the selected bytes (including the new RDPR code), and the read protection is removed.

23.6.5 Write Protection Release

Whether the flash memory is write-protected or not is determined by the user option byte. Read the FLASH_WPR register. Each bit represents 4K bytes of flash memory space. When the bit is '1', it indicates non-write protection status, and when it is '0', it indicates write protection. The process of removing write protection is as follows:
1) Erase the entire user option byte area.
2) Write the correct RDPR code 0xA5 to allow read access.
3) Perform a system reset, reload the selected bytes (including the new WRPR[3:0] bytes), and write protection is removed.

Chapter 24 Extended Configuration (EXTEN)

24.1 Extended Configuration

The system provides an EXTEN extension configuration unit (EXTEN_CTR register). It mainly includes the following functions:

1) Peripheral DMA channel remapping;

2) Configuration of the ADCx scan cycle count output.

24.2 Register Description

Table 24-1 EXTEN-related register

Name	Access address	Description	Reset value
R32_EXTEN_CTR	0x400220C0	Configuration extended control register	0x00000000

24.2.1 Configure Extended Control Register (EXTEN_CTR)

Offset address: 0xC0

31	30	29	28	27	26	25	24	23	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						ADC_SCAN_SEL[1:0]		I2C2_DMA_RM	USART4_DMA_RM	USART3_DMA_RM	USART2_DMA_RM	USART1_DMA_RM	SPI3_DMA_RM	SPI2_DMA_RM	SPI1_DMA_RM

Bit	Name	Access	Description	Reset value
[31:10]	Reserved	RO	Reserved	0
[9:8]	ADC_SCAN_SEL [1:0]	RW	Scan cycle counter output ADC selection: 00: ADC1 scan cycle counter output; 01: ADC2 scan cycle counter output; 10: ADC3 scan cycle counter output; 11: ADC4 scan cycle counter output.	0
7	I2C2_DMA_RM	RW	I2C2 DMA channel remapping: 1: Remapping (TX/DMA1_CH6, RX/DMA1_CH7); 0: Default mapping (TX/DMA1_CH4, RX/DMA1_CH5).	0
6	USART4_DMA_RM	RW	USART4 DMA channel remapping: 1: Remapping (TX/DMA1_CH4, RX/DMA1_CH5); 0: Default mapping (TX/DMA1_CH10, RX/DMA1_CH11).	0

5	USART3_DMA_RM	RW	USART3 DMA channel remapping: 1: Remapping (TX/DMA1_CH4, RX/DMA1_CH5); 0: Default mapping (TX/DMA1_CH2, RX/DMA1_CH3).	0
4	USART2_DMA_RM	RW	USART2 DMA channel remapping: 1: Remapping (TX/DMA1_CH4, RX/DMA1_CH5); 0: Default mapping (TX/DMA1_CH7, RX/DMA1_CH6).	0
3	USART1_DMA_RM	RW	USART1 DMA channel remapping: 1: Remapping (TX/DMA1_CH8, RX/DMA1_CH9); 0: Default mapping (TX/DMA1_CH4, RX/DMA1_CH5).	0
2	SPI3_DMA_RM	RW	SPI3 DMA channel remapping: 1: Remapping (TX/DMA1_CH3, RX/DMA1_CH2); 0: Default mapping (TX/DMA1_CH9, RX/DMA1_CH8).	0
1	SPI2_DMA_RM	RW	SPI2 DMA channel remapping: 1: Remapping (TX/DMA1_CH3, RX/DMA1_CH2); 0: Default mapping (TX/DMA1_CH5, RX/DMA1_CH4).	0
0	SPI1_DMA_RM	RW	SPI1 DMA channel remapping: 1: Remapping (TX/DMA1_CH7, RX/DMA1_CH6); 0: Default mapping (TX/DMA1_CH3, RX/DMA1_CH2).	0